

Relational Database Service

User Guide

Issue	01
Date	2025-10-28



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Introduction..... 1

1.1 What Is RDS?..... 1

1.2 Basic Concepts..... 2

1.3 Advantages..... 3

1.3.1 Easy Management..... 3

1.3.2 High Security..... 4

1.3.3 High Reliability..... 5

1.3.4 Comparison Between RDS and Self-Built Databases..... 5

1.4 Product Series..... 6

1.4.1 DB Instance Introduction..... 6

1.4.2 Function Comparison..... 7

1.5 DB Instance Description..... 9

1.5.1 DB Instance Types..... 9

1.5.2 DB Instance Classes..... 9

1.5.3 DB Engines and Versions..... 11

1.5.4 DB Instance Statuses..... 12

1.6 Typical Applications..... 13

1.6.1 Read/Write Splitting..... 13

1.7 Constraints..... 13

1.7.1 MySQL Constraints..... 13

1.7.2 PostgreSQL Constraints..... 14

1.7.3 Microsoft SQL Server Constraints..... 15

1.8 Related Services..... 17

2 Getting Started with RDS for MySQL..... 18

2.1 Operation Guide..... 18

2.2 Step 1: Create a DB Instance..... 19

2.3 Step 2: Connect to a DB Instance..... 24

2.3.1 Connecting to a DB Instance..... 24

2.3.2 Connecting to a DB Instance Through a Private Network..... 26

2.3.2.1 Connecting to a DB Instance Through a Private Network..... 26

2.3.2.2 Configuring Security Group Rules..... 27

2.3.2.3 Connecting to a DB Instance Through a Private Network..... 30

2.3.3 Connecting to a DB Instance Through a Public Network..... 33

2.3.3.1 Connecting to a DB Instance Through a Public Network.....	33
2.3.3.2 Binding an EIP.....	34
2.3.3.3 Configuring Security Group Rules.....	34
2.3.3.4 Connecting to a DB Instance Through a Public Network.....	37
3 Getting Started with RDS for PostgreSQL.....	40
3.1 Operation Guide.....	40
3.2 Step 1: Create a DB Instance.....	41
3.3 Step 2: Connect to a DB Instance.....	45
3.3.1 Connecting to a DB Instance.....	46
3.3.2 Connecting to a PostgreSQL DB Instance Through a Private Network.....	47
3.3.2.1 Connecting to a DB Instance Through a Private Network.....	47
3.3.2.2 Configuring Security Group Rules.....	48
3.3.2.3 Connecting to a DB Instance Through psql.....	51
3.3.3 Connecting to a PostgreSQL DB Instance Through a Public Network.....	52
3.3.3.1 Connecting to a DB Instance Through a Public Network.....	52
3.3.3.2 Binding an EIP.....	53
3.3.3.3 Configuring Security Group Rules.....	54
3.3.3.4 Connecting to a DB Instance Through psql.....	55
4 Getting Started with RDS for SQL Server.....	57
4.1 Operation Guide.....	57
4.2 Step 1: Create a DB Instance.....	58
4.3 Step 2: Connect to a DB Instance.....	62
4.3.1 Connecting to a DB Instance.....	62
4.3.2 Connecting to a DB Instance Through a Private Network.....	64
4.3.2.1 Connecting to a DB Instance Through a Private Network.....	64
4.3.2.2 Configuring Security Group Rules.....	65
4.3.2.3 Connecting to a DB Instance Through a Private Network.....	66
4.3.3 Connecting to a DB Instance Through a Public Network.....	70
4.3.3.1 Connecting to a DB Instance Through a Public Network.....	70
4.3.3.2 Binding an EIP.....	71
4.3.3.3 Configuring Security Group Rules.....	72
4.3.3.4 Connecting to a DB Instance Through a Public Network.....	73
5 Working with RDS for MySQL.....	77
5.1 Data Migration.....	77
5.1.1 Migrating Data to RDS for MySQL Using mysqldump.....	77
5.2 Parameter Tuning.....	80
5.2.1 Suggestions on MySQL Parameter Tuning.....	80
5.3 MySQL Kernel Functions.....	82
5.3.1 Kernel Version Description.....	82
5.4 Instance Lifecycle.....	89
5.4.1 Creating a Same DB Instance as an Existing DB Instance.....	90

5.4.2 Rebooting DB Instances or Read Replicas.....	90
5.4.3 Selecting Displayed Items.....	91
5.4.4 Exporting DB Instance Information.....	92
5.4.5 Deleting a DB Instance or Read Replica.....	93
5.4.6 Recycling a DB Instance.....	94
5.5 Instance Modifications.....	95
5.5.1 Upgrading a Major Version.....	95
5.5.2 Changing a DB Instance Name.....	95
5.5.3 Changing the Failover Priority.....	96
5.5.4 Changing a DB Instance Class.....	97
5.5.5 Scaling up Storage Space.....	98
5.5.6 Changing the Maintenance Window.....	100
5.5.7 Changing a DB Instance Type from Single to Primary/Standby.....	101
5.5.8 Promoting a Read Replica Into a Single DB Instance.....	101
5.5.9 Manually Switching Between Primary and Standby DB Instances.....	102
5.5.10 Migrating a Standby DB Instance.....	103
5.6 Read Replicas.....	104
5.6.1 Introducing Read Replicas.....	104
5.6.2 Creating a Read Replica.....	105
5.6.3 Managing a Read Replica.....	107
5.7 Backups and Restorations.....	108
5.7.1 Working with Backups.....	108
5.7.2 Configuring an Automated Backup Policy.....	109
5.7.3 Setting a Cross-Region Backup Policy.....	110
5.7.4 Creating a Manual Backup.....	111
5.7.5 Downloading a Backup File.....	112
5.7.6 Downloading a Binlog Backup File.....	114
5.7.7 Setting a Local Retention Period for MySQL Binlogs.....	114
5.7.8 Restoring from Backup Files to DB Instances.....	115
5.7.9 Restoring a DB Instance to a Point in Time.....	116
5.7.10 Replicating a Backup.....	117
5.8 Parameter Template Management.....	118
5.8.1 Creating a Parameter Template.....	118
5.8.2 Modifying Parameters.....	119
5.8.3 Exporting a Parameter Template.....	121
5.8.4 Comparing Parameter Templates.....	122
5.8.5 Viewing Parameter Change History.....	123
5.8.6 Replicating a Parameter Template.....	124
5.8.7 Resetting a Parameter Template.....	125
5.8.8 Applying a Parameter Template.....	126
5.8.9 Viewing Application Records of a Parameter Template.....	127
5.8.10 Modifying a Parameter Template Description.....	127

5.8.11 Deleting a Parameter Template.....	128
5.9 Connection Management.....	128
5.9.1 Configuring and Changing a Floating IP Address.....	128
5.9.2 Binding and Unbinding an EIP.....	129
5.9.3 Changing a Database Port.....	130
5.9.4 Downloading a Certificate.....	131
5.9.5 Configuring a Security Group Rule.....	132
5.10 Database Management.....	135
5.10.1 Creating a Database.....	135
5.10.2 Granting Database Permissions.....	136
5.10.3 Deleting a Database.....	136
5.10.4 Enabling or Disabling Event Scheduler.....	137
5.11 Account Management (Non-Administrator).....	138
5.11.1 Creating a Database Account.....	138
5.11.2 Resetting a Password for a Database Account.....	139
5.11.3 Changing Permissions for a Database Account.....	140
5.11.4 Deleting a Database Account.....	141
5.12 Database Account Security.....	142
5.13 Data Security.....	142
5.13.1 Resetting the Administrator Password.....	142
5.13.2 Changing a Security Group.....	144
5.14 Metrics and Alarms.....	144
5.14.1 Configuring Displayed Metrics.....	145
5.14.2 Setting Alarm Rules.....	152
5.14.3 Viewing Monitoring Metrics.....	153
5.15 Interconnection with CTS.....	153
5.15.1 Key Operations Supported by CTS.....	154
5.15.2 Viewing Tracing Events.....	155
5.16 Log Management.....	156
5.16.1 Viewing and Downloading Error Logs.....	156
5.16.2 Viewing and Downloading Slow Query Logs.....	157
5.16.3 Viewing Failover/Switchover Logs.....	159
5.16.4 Enabling the SQL Audit Function.....	160
5.16.5 Downloading SQL Audit Logs.....	161
5.17 Task Center.....	162
5.17.1 Viewing a Task.....	162
5.17.2 Deleting a Task Record.....	164
6 Working with RDS for PostgreSQL.....	166
6.1 Data Migration.....	166
6.1.1 Migrating Data to RDS for PostgreSQL Using psql.....	166
6.2 Parameter Tuning.....	168
6.2.1 Suggestions on PostgreSQL Parameter Tuning.....	168

6.3 PostgreSQL Enhanced Edition.....	169
6.3.1 Introduction to PostgreSQL Enhanced Edition.....	169
6.3.2 Functions.....	169
6.3.3 System Views.....	176
6.3.4 Data Types.....	178
6.3.5 Implicit Type Conversion.....	179
6.3.6 Predefined Parameters.....	179
6.3.7 Macro Variables.....	180
6.3.8 Operators.....	180
6.3.9 Syntax.....	180
6.3.10 Enhanced Functions.....	182
6.3.11 Security Hardening.....	183
6.3.12 Other Functions.....	183
6.4 PostgreSQL Kernel Functions.....	183
6.4.1 RDS for PostgreSQL Kernel Release History.....	184
6.5 Instance Lifecycle.....	184
6.5.1 Creating a Same DB Instance as an Existing DB Instance.....	184
6.5.2 Rebooting DB Instances or Read Replicas.....	185
6.5.3 Selecting Displayed Items.....	186
6.5.4 Exporting DB Instance Information.....	187
6.5.5 Deleting a Pay-per-Use DB Instance or Read Replica.....	187
6.5.6 Recycling a DB Instance.....	189
6.6 Instance Modifications.....	190
6.6.1 Changing a DB Instance Name.....	190
6.6.2 Changing the Failover Priority.....	190
6.6.3 Changing a DB Instance Class.....	191
6.6.4 Scaling up Storage Space.....	192
6.6.5 Changing a DB Instance Type from Single to Primary/Standby.....	194
6.6.6 Manually Switching Between Primary and Standby DB Instances.....	195
6.6.7 Migrating a Standby DB Instance.....	196
6.7 Read Replicas.....	197
6.7.1 Introducing Read Replicas.....	197
6.7.2 Creating a Read Replica.....	198
6.7.3 Managing a Read Replica.....	200
6.8 Backups and Restorations.....	201
6.8.1 Working with Backups.....	201
6.8.2 Configuring an Automated Backup Policy.....	201
6.8.3 Set a Cross-Region Backup Policy.....	202
6.8.4 Creating a Manual Backup.....	203
6.8.5 Downloading a Full Backup File.....	205
6.8.6 Downloading an Incremental Backup File.....	206
6.8.7 Restoring from Backup Files to RDS for PostgreSQL.....	206

6.8.8 Restoring a DB Instance to a Point in Time.....	208
6.8.9 Replicating a Backup.....	209
6.9 Parameter Template Management.....	210
6.9.1 Creating a Parameter Template.....	210
6.9.2 Modifying Instance Parameters.....	212
6.9.3 Exporting a Parameter Template.....	214
6.9.4 Comparing Parameter Templates.....	215
6.9.5 Viewing Parameter Change History.....	216
6.9.6 Replicating a Parameter Template.....	217
6.9.7 Resetting a Parameter Template.....	217
6.9.8 Applying a Parameter Template.....	218
6.9.9 Viewing Application Records of a Parameter Template.....	219
6.9.10 Modifying a Parameter Template Description.....	219
6.9.11 Deleting a Parameter Template.....	220
6.10 Connection Management.....	220
6.10.1 Configuring and Changing a Floating IP Address.....	221
6.10.2 Binding and Unbinding an EIP.....	221
6.10.3 Changing a Database Port.....	223
6.10.4 Connecting to a DB Instance Through pgAdmin.....	223
6.11 Plugin Management.....	226
6.11.1 Creating or Deleting a Plugin.....	226
6.11.2 Supported Plugins.....	228
6.12 Database Account Security.....	231
6.13 Data Security.....	232
6.13.1 Resetting the Administrator Password.....	232
6.13.2 Changing a Security Group.....	234
6.14 and AlarmsMetrics and Alarms.....	234
6.14.1 Configuring Displayed Metrics.....	234
6.14.2 Setting Alarm Rules.....	238
6.14.3 Viewing Monitoring Metrics.....	238
6.15 Interconnection with CTS.....	239
6.15.1 Key Operations Supported by CTS.....	239
6.15.2 Viewing Tracing Events.....	241
6.16 Task Center.....	242
6.16.1 Viewing a Task.....	242
6.16.2 Deleting a Task Record.....	243
7 Working with RDS for SQL Server.....	244
7.1 Instance Lifecycle.....	244
7.1.1 Creating a Same DB Instance as an Existing DB Instance.....	244
7.1.2 Rebooting DB Instances or Read Replicas.....	245
7.1.3 Selecting Displayed Items.....	245
7.1.4 Exporting DB Instance Information.....	246

7.1.5 Deleting a Pay-per-Use DB Instance or Read Replica.....	247
7.1.6 Recycling a DB Instance.....	248
7.2 Instance Modifications.....	249
7.2.1 Changing a DB Instance Name.....	249
7.2.2 Changing the Failover Priority.....	250
7.2.3 Changing a DB Instance Class.....	250
7.2.4 Scaling up Storage Space.....	251
7.2.5 Changing a DB Instance Type from Single to Primary/Standby.....	253
7.2.6 Manually Switching Between Primary and Standby DB Instances.....	253
7.3 Read Replicas.....	254
7.3.1 Introducing Read Replicas.....	255
7.3.2 Creating a Read Replica.....	255
7.3.3 Managing a Read Replica.....	258
7.4 Backups and Restorations.....	258
7.4.1 Working with Backups.....	259
7.4.2 Configuring an Automated Backup Policy.....	259
7.4.3 Creating a Manual Backup.....	260
7.4.4 Downloading a Backup File.....	261
7.4.5 Restoring from Backup Files to RDS for SQL Server.....	262
7.4.6 Replicating a Backup.....	264
7.5 Parameter Template Management.....	265
7.5.1 Creating a Parameter Template.....	265
7.5.2 Modifying Parameters.....	266
7.5.3 Exporting a Parameter Template.....	269
7.5.4 Comparing Parameter Templates.....	269
7.5.5 Viewing Parameter Change History.....	270
7.5.6 Replicating a Parameter Template.....	271
7.5.7 Resetting a Parameter Template.....	272
7.5.8 Applying a Parameter Template.....	273
7.5.9 Viewing Application Records of a Parameter Template.....	274
7.5.10 Modifying a Parameter Template Description.....	274
7.5.11 Deleting a Parameter Template.....	275
7.6 Connection Management.....	275
7.6.1 Configuring and Changing a Floating IP Address.....	275
7.6.2 Binding and Unbinding an EIP.....	276
7.6.3 Changing a Database Port.....	277
7.7 Data Security.....	278
7.7.1 Resetting the Administrator Password.....	278
7.7.2 Changing a Security Group.....	280
7.8 and AlarmsMetrics and Alarms.....	280
7.8.1 Configuring Displayed Metrics.....	280
7.8.2 Setting Alarm Rules.....	285

7.8.3 Viewing Monitoring Metrics.....	286
7.9 Interconnection with CTS.....	286
7.9.1 Key Operations Supported by CTS.....	287
7.9.2 Viewing Tracing Events.....	289
7.10 Task Center.....	290
7.10.1 Viewing a Task.....	290
7.10.2 Deleting a Task Record.....	291
7.11 Usage of Stored Procedures.....	292
7.11.1 Creating a Database Account.....	292
7.11.2 Updating Information About Operators for Alerts and Jobs.....	293
7.11.3 Removing Alerts.....	296
7.11.4 Removing SQL Server Agent Notification Definitions for Specific Alerts and Operators.....	296
7.11.5 Removing Operators.....	297
8 FAQs.....	299
8.1 Product Consulting.....	299
8.1.1 What Should I Pay Attention to When Using RDS?.....	299
8.1.2 What Is the Availability of RDS DB Instances?.....	299
8.1.3 Can I Use a Template to Create DB Instances?.....	299
8.1.4 What Are the Differences Between RDS and Other Database Solutions?.....	300
8.1.5 Will My RDS DB Instances Be Affected by Other User Instances?.....	301
8.1.6 Does RDS Support Cross-AZ High Availability?.....	301
8.1.7 Can RDS Primary/Standby DB Instances Be Changed to Single DB Instances?.....	301
8.1.8 What Should I Do If Garbled Characters Are Displayed After SQL Query Results Are Exported to an Excel File?.....	301
8.1.9 How Do I Create an AD Domain?.....	301
8.1.10 What Can I Do About Slow Respond of Websites When They Use RDS?.....	307
8.1.11 How Does a Cloud Database Perform a Primary/Standby Switchover?.....	307
8.1.12 Can Multiple ECSs Connect to the Same RDS DB Instance?.....	308
8.1.13 Why Is an Error Reported When I Attempt to Delete a Database from RDS SQL Server Primary/Standby DB Instances?.....	308
8.2 Resource and Disk Management.....	309
8.2.1 Which Types of Logs and Files Occupy RDS Storage Space?.....	309
8.2.2 Which Items Occupy the Storage Space of My RDS DB Instances?.....	311
8.2.3 What Overhead Does the Storage Space Have After I Applied for an RDS DB Instance?.....	311
8.2.4 How Much Storage Space Is Required for DDL Operations?.....	311
8.2.5 How Many DB Instances Can Run on RDS?.....	311
8.2.6 How Many Databases Can Run on an RDS DB Instance?.....	311
8.3 Database Connection.....	312
8.3.1 Can an External Server Access the RDS Database?.....	312
8.3.2 What Do I Do If the Number of RDS Database Connections Reaches the Upper Limit?.....	312
8.3.3 What Is the Maximum Number of Connections to an RDS DB Instance?.....	312
8.3.4 How Can I Create and Connect to an ECS?.....	314
8.3.5 What Should I Do If an ECS Cannot Connect to an RDS DB Instance Through a Private Network?.....	314

8.3.6 What Should I Do If a Database Client Problem Causes a Connection Failure?.....	315
8.3.7 What Should I Do If an RDS Database Problem Causes a Connection Failure?.....	315
8.3.8 How Do My Applications Access an RDS DB Instance in a VPC?.....	316
8.3.9 Do Applications Need to Support Reconnecting to the RDS DB Instance Automatically?.....	316
8.3.10 How Can I Connect to a PostgreSQL Database Through JDBC?.....	316
8.3.11 What Should I Do If an RDS for SQL Server DB Instance Failed to Be Connected?.....	319
8.3.12 Can I Access an RDS DB Instance Over an Intranet Connection Across Regions?.....	320
8.3.13 Is an SSL Connection to a DB Instance Interrupted After a Primary/Standby Switchover or Failover?.....	320
8.3.14 Does MySQL Support SSL Connections?.....	320
8.3.15 Why Did the New Password Not Take Effect After I Reset the Administrator Password?.....	320
8.4 Database Migration.....	321
8.4.1 Why Do I Need to Use the mysqldump or pg_dump Tools for Migration?.....	321
8.4.2 What Types of DB Engines Does RDS Support for Importing Data?.....	321
8.5 Database Permission.....	321
8.5.1 Why Does the Root User Not Have the Super Permissions?.....	321
8.6 Database Storage.....	322
8.6.1 What Storage Engines Does the RDS for MySQL Support?.....	322
8.6.2 What Is the RDS DB Instance Storage Configuration?.....	323
8.6.3 Can I Change the Storage Type of an RDS DB Instance from Common I/O to Ultra-high I/O?.....	323
8.6.4 What Should I Do If My Data Exceeds the Available Storage of an RDS DB Instance?.....	324
8.7 Client Installation.....	324
8.7.1 How Can I Install the MySQL Client?.....	324
8.7.2 How Can I Install the PostgreSQL Client?.....	325
8.7.3 How Can I Install SQL Server Management Studio?.....	327
8.8 Backup and Restoration.....	327
8.8.1 How Long Does RDS Store Backup Data For?.....	327
8.8.2 Can My Database Be Used in the Backup Window?.....	328
8.8.3 How Can I Back Up an RDS Database to an ECS?.....	328
8.8.4 Why Has My Automated Backup Failed?.....	328
8.8.5 What Happens to Database Backups After an RDS DB Instance Is Deleted?.....	328
8.8.6 Will My Backups Be Deleted If I Delete My Cloud Account?.....	329
8.8.7 Why Is a Table or Data Missing from My Database?.....	329
8.9 Database Monitoring.....	329
8.9.1 Which DB Instance Monitoring Metrics Do I Need to Pay Attention To?.....	329
8.10 Capacity Expansion and Specification Change.....	330
8.10.1 Are My RDS DB Instances Available When Scaling?.....	330
8.10.2 Why Does the DB Instance Become Faulty After the Original Database Port Is Changed?.....	330
8.11 Database Parameter Modification.....	330
8.11.1 What Inappropriate Parameter Settings Cause Unavailability of the PostgreSQL Database?.....	331
8.11.2 Where Should I Store the NDF Files for Microsoft SQL Server?.....	331
8.11.3 How Do I Configure a Password Expiration Policy for RDS for MySQL DB Instances?.....	331
8.12 Log Management.....	333

8.12.1 How Long Is the Delay for RDS MySQL Slow Query Logs?	333
8.12.2 What's the Slow Query Threshold for Microsoft SQL Server?	333
8.12.3 How Can I Obtain Microsoft SQL Server Error Logs Using Commands?	333
8.12.4 Can I Export Statistics on RDS Slow Query Logs?	334
8.13 Network Security	334
8.13.1 What Security Protection Policies Does RDS Have?	334
8.13.2 How Can I Ensure the Security of RDS DB Instances in a VPC?	334
8.13.3 How Can Data Security Be Ensured During Transmission When I Access RDS Through an EIP?	334
8.13.4 How Can I Prevent Untrusted Source IP Addresses from Accessing RDS?	335
8.13.5 How Can I Import the Root Certificate to a Windows or Linux OS?	335
8.13.6 How Can I Identify the Validity Period of an SSL Root Certificate?	336
8.13.7 How Can I Identify Data Corruption?	336
9 Troubleshooting	337
9.1 RDS for MySQL	337
9.1.1 Backup and Restoration Issues	337
9.1.1.1 Backup Failures Due to DDL Operations	337
9.1.1.2 RDS for MySQL Backup Job Failure	337
9.1.1.3 Manual Backups Taking More Time Than Automated Full Backups	339
9.1.1.4 Automated Incremental Backup Failed Due to Full Storage	339
9.1.1.5 Precautions for Exporting Large Tables Through mysqldump	340
9.1.1.6 Commands for Exporting Data Through mysqldump	340
9.1.1.7 SQL Statements Such as SET @@SESSION.SQL_LOG_BIN Displayed After You Run mysqldump	342
9.1.1.8 Insufficient Permissions During Data Export Using mysqldump	343
9.1.2 Primary/Standby Replication Issues	343
9.1.2.1 How Primary/Standby Replication Works	343
9.1.2.2 Automatic Recovery of Extended Primary/Standby Replication Delay	345
9.1.2.3 Primary/Standby Replication Delay Scenarios and Solutions	346
9.1.2.4 Abnormal Replication Between Primary and Standby RDS DB Instances	348
9.1.2.5 Primary/Standby Replication Delay Increases Sharply and Then Decreases	348
9.1.3 Parameter-related Issues	349
9.1.3.1 long_query_time Changes Fail to Take Effect	349
9.1.3.2 Incorrect GROUP_CONCAT Results	350
9.1.3.3 [ERROR] 1071 Reported When an Index Fails to Be Created for RDS for MySQL	351
9.1.3.4 Tables Failed to Be Found After Case-Sensitivity Setting Changes for RDS for MySQL	353
9.1.3.5 Timeout Parameters	353
9.1.4 Performance Issues	355
9.1.4.1 DB Instance Becoming Read-Only Due to Full Storage	355
9.1.4.2 High Storage Usage Due to Uncleared Old Binlogs	355
9.1.4.3 Read Replica Occupies Far More Storage Than the Primary Instance	356
9.1.4.4 Slow SQL Execution Due to Hot and Cold Data Problems	357
9.1.4.5 Tablespace Bloat	357
9.1.4.6 Full Storage Caused by Complex Queries	358

9.1.4.7 Instance Class Change or Minor Version Upgrade Failure Caused by Long Transactions.....	359
9.1.4.8 Native Error 1461 Reported by an RDS for MySQL DB Instance.....	360
9.1.4.9 System Inaccessible After Field Addition to an RDS for MySQL Database Table.....	361
9.1.4.10 Storage Space Occupied by Undo Logs Due to Long Transactions.....	361
9.1.5 SQL Issues.....	362
9.1.5.1 Error 1366 Reported When Data Containing Emojis Is Updated.....	362
9.1.5.2 Invalid TIMESTAMP Default Value during Table Creation.....	362
9.1.5.3 AUTO_INCREMENT Not Displayed in the Table Structure.....	363
9.1.5.4 Slow Stored Procedure Execution Due to Inconsistent Collations.....	364
9.1.5.5 ERROR [1412] Reported for a DB Instance.....	365
9.1.5.6 Error Message "Too many keys specified" Displayed When a Secondary Index Is Created.....	365
9.1.5.7 Failed to Delete a Table with a Foreign Key.....	366
9.1.5.8 DISTINCT and GROUP BY Optimization.....	367
9.1.5.9 Character Set and Collation Settings.....	368
9.1.5.10 Server Error Message Displayed When a User Is Created for a DB Instance.....	370
9.1.5.11 Slow SQL Queries After a Large Amount of Data Is Deleted from a Large Table.....	371
9.1.5.12 Equivalent Comparison Failures with Floating-Point Numbers.....	372
9.1.5.13 ERROR[1451] Reported When a Table with Foreign Keys Cannot Be Deleted.....	374
9.1.5.14 Solution to the Failure of Converting the Field Type.....	375
9.1.5.15 "Row size too large" Reported When an RDS for MySQL Table Failed to Be Created.....	375
9.1.5.16 ERROR [1412] Reported by an RDS for MySQL DB Instance.....	376
9.1.6 Connection Issues.....	376
9.1.6.1 "Access denied" Displayed During Database Connection.....	377
9.1.6.2 Error Message "connection established slowly".....	378
9.1.6.3 SSL Connection Failed Due to Inconsistent TLS Versions.....	379
9.1.6.4 RDS for MySQL Client Automatically Disconnected from a DB Instance.....	380
9.1.6.5 RDS for MySQL DB Instance Inaccessible.....	382
9.1.6.6 Login Failed After the authentication_string Field Is Changed to Display the Password for RDS for MySQL.....	385
9.1.6.7 MySQL-server Connection Failure After a Version Upgrade of RDS for MySQL.....	385
9.1.6.8 Connection Exit Due to Improper Timeout Parameter Settings.....	387
9.1.6.9 Database Connection Through Code (php/java/python) Failed After SSL Is Enabled.....	387
9.1.7 Other Issues.....	388
9.1.7.1 Auto-increment Field Values.....	388
9.1.7.2 Starting Value and Increment of AUTO_INCREMENT.....	390
9.1.7.3 Auto-Increment Field Value Jump.....	393
9.1.7.4 Changing the AUTO_INCREMENT Value of a Table.....	398
9.1.7.5 Failed to Insert Data Because Values for the Auto-increment Primary Key Field Reach the Upper Limit.....	400
9.1.7.6 Impact of Creating an Empty Username.....	402
9.1.7.7 Connection to a Primary/Standby DB Instance Suspended Using pt-osc.....	404
9.2 RDS for SQL Server.....	406

9.2.1 Failed to Bind a New User to an RDS for SQL Server DB Instance That Was Recovered from Backups 406

1 Introduction

1.1 What Is RDS?

RDS is a cloud-based web service that is reliable, scalable, and easy to manage. RDS supports the following DB engines:

- [MySQL](#)
- [PostgreSQL](#)
- [Microsoft SQL Server](#)

RDS provides a comprehensive performance monitoring system, multi-level security protection measures, and a professional database management platform, allowing you to easily set up and scale a relational database. On the RDS console, you can perform almost all necessary tasks and no programming is required. The console simplifies operations and reduces routine O&M workloads, so you can stay focused on application and service development.

RDS for MySQL

MySQL is one of the world's most popular open-source relational databases. It works with the Linux, Apache, and Perl/PHP/Python to establish a LAMP model to provide efficient web solutions. RDS for MySQL is reliable, secure, scalable, inexpensive, and easy to manage.

- It supports various web applications and is cost-effective, preferred by small- and medium-sized enterprises.
- A web-based console provides comprehensive monitoring information, making your operations easy and visual.
- You can flexibly scale resources based on your service requirements and pay for only what you use.

For details about the versions supported by RDS for MySQL, see [DB Engines and Versions](#).

RDS for PostgreSQL

PostgreSQL is an open-source object-relational database management system that focuses on extensibility and standards compliance. It is known as the most

advanced open-source database available. RDS for PostgreSQL excels in processing complex online transaction processing (OLTP) transactions and supports NoSQL (JSON, XML, or hstore) and geographic information system (GIS) data types. It has earned a reputation for reliability and data integrity, and is widely used for websites, location-based applications, and complex data object processing.

- RDS for PostgreSQL supports the postgis plugin and provides excellent spatial performance.
- RDS for PostgreSQL is a good cost-effective solution for many different scenarios. You can flexibly scale resources based on your service requirements and pay for only what you use.

For details about the versions supported by RDS for PostgreSQL, see [DB Engines and Versions](#).

RDS for SQL Server

SQL Server is a commercial database with a mature enterprise-class architecture. One-stop deployment simplifies key O&M services and greatly reduces labor costs. It is widely used in government, finance, medical care, education, and gaming.

RDS for SQL Server is reliable, scalable, inexpensive, and easy to manage. It uses a high availability (HA) architecture, guarantees data security, and recovers from faults within seconds.

For details about the versions supported by RDS for SQL Server, see [DB Engines and Versions](#).

1.2 Basic Concepts

DB Instances

The smallest management unit of RDS is the DB instance. A DB instance is an isolated database environment on the cloud. Each DB instance runs a DB engine. For details about DB instance types, specifications, engines, versions, and statuses, see [DB Instance Description](#).

DB Engines

RDS supports the following DB engines:

- MySQL
- PostgreSQL
- Microsoft SQL Server

For details about the supported versions, see [DB Engines and Versions](#).

DB Instance Types

For details about DB instance types, see [DB Instance Introduction](#) and [Function Comparison](#).

DB Instance Classes

The DB instance class determines the compute (vCPUs) and memory capacity (memory size) of a DB instance. For details, see [DB Instance Classes](#).

Automated Backups

When you create a DB instance, an automated backup policy is enabled by default. After the DB instance is created, you can modify the policy. RDS will automatically create full backups for DB instances based on your settings.

Manual Backups

Manual backups are user-initiated full backups of DB instances. They are retained until you delete them manually.

Regions and AZs

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- Regions are defined in terms of their geographical location and network latency. Each region is completely independent, improving fault tolerance and stability. After a resource is created, its region cannot be changed.
- An AZ is a physical location using independent power supplies and networks. Faults in an AZ do not affect other AZs. A region can contain multiple AZs, which are physically isolated but interconnected through internal networks. This ensures the independence of AZs and provides low-cost and low-latency network connections.

shows the relationship between regions and AZs.

Projects

Projects are used to group and isolate OpenStack resources (compute, storage, and network resources). A project can be a department or a project team. Multiple projects can be created for a single account.

1.3 Advantages

1.3.1 Easy Management

Quick Setup

You can create a DB instance on the management console within minutes and access RDS from an ECS to reduce the application response time and avoid paying for the traffic that would be generated by regular public access.

Elastic Scaling

Cloud Eye monitors changes in the load on your database and storage capacity. You can flexibly scale resources accordingly and pay for only what you use.

High Compatibility

You use RDS database engines (DB engines) the same way as you would use a native engine. RDS is compatible with existing programs and tools.

Easy O&M

Routine RDS maintenance and management operations, including hardware and software fault handling and database patch updates, are easy to perform. With the web-based console, you can reboot DB instances, reset passwords, modify parameters, view error or slow query logs, and restore data. Additionally, the system helps you monitor DB instances in real time and generates alarms if an error occurs. You can check DB instance information at any time, including CPU usage, IOPS, database connections, and storage space usage.

1.3.2 High Security

Network Isolation

RDS uses Virtual Private Cloud (VPC) and network security groups to isolate and secure your DB instances. VPCs allow you to define what IP address range can access RDS. You can configure subnets and security groups to control access to DB instances.

Access Control

RDS controls access through the account/IAM user and security groups. When you create an RDS DB instance, an account is automatically created. To separate out specific permissions, you can create IAM users and assign permissions to them as needed. VPC security groups have rules that govern both inbound and outbound traffic for DB instances.

Transmission Encryption

RDS uses Transport Layer Security (TLS) and Secure Sockets Layer (SSL) to encrypt transmission. You can download a Certificate Agency (CA) certificate from the RDS console and upload it when connecting to a database for authentication.

Storage Encryption

RDS uses static encryption and tablespace encryption to encrypt the data to be stored. Encryption keys are managed by .

Data Deletion

When you delete an RDS DB instance, its attached disks, storage space its automated backups occupy, and all data it stores will be deleted. You can restore a deleted DB instance using a manual backup.

Anti-DDoS

When you connect to an RDS DB instance through a public network, there may be risks of a distributed denial-of-service (DDoS) attack. If the RDS security system

detects a DDoS attack, it will enable the anti-DDoS function. If the function cannot defend against the attack or the attack reaches the black hole threshold, black hole processing is triggered to ensure availability of the RDS service.

Security Protection

RDS is protected by multiple layers of firewalls to defend against various malicious attacks, such as DDoS attacks and SQL injections. For security reasons, you are advised to access RDS through a private network.

1.3.3 High Reliability

Dual-Host Hot Standby

RDS uses the hot standby architecture, in which failover upon fault occurrence takes only some seconds.

Data Backup

RDS automatically backs up data every day and transfers backup files to Object Storage Service (OBS). The backup files can be stored for 732 days and can be restored with just a few clicks. You can set a custom backup policy and create manual backups at any time.

Data Restoration

You can restore data from backups to any point in time during the backup retention period. In most scenarios, you can use backup files to restore data to a new DB instance at any time point within 732 days. After the data is verified, data can be migrated back to the primary DB instance.

1.3.4 Comparison Between RDS and Self-Built Databases

Performance

Item	Cloud Database RDS	Self-Built Database Service
Service availability	For details, see the <i>Elastic Cloud Service User Guide</i> .	Requires device procurement, primary/standby relationship setup, and RAID setup.
Data reliability	For more information, see the <i>Elastic Volume Service User Guide</i> .	Requires device procurement, primary/standby relationship setup, and RAID setup.
System security	Defends against Anti-DDoS attacks and promptly repairs database security vulnerabilities.	Requires procurement of expensive devices and software, as well as manual detection and repair of security vulnerabilities.

Item	Cloud Database RDS	Self-Built Database Service
Database backup	Supports automated backups, manual backups, and custom backup retention periods.	Requires device procurement, setup, and maintenance.
Hardware and software investment	Supports on-demand pricing and scaling without requiring hardware and software investment.	Requires large investment in database servers. The SQL Server license must be paid for separately.
System hosting	Not required.	Requires two servers for primary/standby DB instances.
Maintenance cost	Not required.	Requires large manpower investment and professional database administrator (DBA) for maintenance.
Deployment and scaling	Supports elastic scaling, fast upgrade, and on-demand enabling.	Requires procurement, deployment, and coordination of hardware that matches original devices.
Resource utilization	Bills users based on the resources actually used, resulting in 100% resource utilization.	Considers peak traffic, resulting in low resource utilization.

1.4 Product Series

1.4.1 DB Instance Introduction

Currently, RDS DB instances are classified into the following types:

- Single
- Primary/Standby
- Cluster

Different series support different DB engines and instance specifications.

Table 1-1 DB instance types

DB Instance Type	Description	Scenarios
Single	Uses a single-node architecture. More cost-effective than the mainstream primary/standby DB instances.	• Personal learning • Microsites • Development and testing environment of small- and medium-sized enterprises
Primary/Standby	Uses an HA architecture with one master node and one slave node. The primary and standby DB instances share the same IP address and can be deployed in different AZs.	• Production databases of large- and medium-sized enterprises • Applications for the Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other industries
Cluster	Uses the Microsoft Always On architecture with one master node, one slave node, and up to five read-only nodes. It features higher availability, reliability, and scalability.	• Finance industry • Internet industry • Hotel industry • Online education

1.4.2 Function Comparison

Single DB instances use a single-node architecture. Different from the primary/standby DB instances, a single DB instance contains only one node and has no slave node for fault recovery.

Advantage Comparison

- Single DB instances: support the creation of read replicas and support the queries of error logs and slow query logs. Different from primary/standby DB instances that have two database nodes, a single DB instance has only one node. If a node fails, the restoration will take a long time. Therefore, single DB instances are not recommended for sensitive services that have high requirements on database availability.
- Primary/Standby DB instances: use the slave database node only for failover and restoration. The slave database node does not provide services. The performance of single DB instances is similar to or even higher than the primary/standby DB instances.

Table 1-2 Function comparisons

Function	Single	Primary/Standby	Cluster
Number of nodes	1	2	2
Specifications	vCPUs: a maximum of 60 Memory: a maximum of 512 GB Storage: a maximum of 4,000 GB	vCPUs: a maximum of 60 Memory: a maximum of 512 GB Storage: a maximum of 4,000 GB	vCPUs: a maximum of 64 Memory: a maximum of 512 GB Storage: a maximum of 4 TB
Monitoring and alarms	Supported	Supported	Supported
Security group	Supported	Supported	Supported
Backups and restorations	Supported	Supported	Supported
Parameter settings	Supported	Supported	Supported
SSL	Supported	Supported	Supported
Log management	Supported	Supported	Supported
Read replicas (need to be created)	Supported	Supported	Supported
High-frequency monitoring	Supported	Supported	Supported
Primary/standby switchover or failover	Not supported	Supported	Supported
Standby DB instance migration	Not supported	Supported	Supported
Manual primary/standby switchover	Not supported	Supported	Supported

Function	Single	Primary/Standby	Cluster
Instance class change	Supported	Supported	Supported

1.5 DB Instance Description

1.5.1 DB Instance Types

The smallest management unit of RDS is the DB instance. A DB instance is an isolated database environment on the cloud. Each DB instance can contain multiple user-created databases, and you can access a DB instance using the same tools and applications that you use with a stand-alone DB instance. You can create and modify DB instances using the management console or APIs. RDS does not have limits on the number of running DB instances. Each DB instance has a DB instance identifier.

DB instances are classified into the following types.

Table 1-3 DB instance types

DB Instance Type	Description
Single	Uses a single-node architecture. More cost-effective than primary/standby DB instances.
Primary/Standby	Uses an HA architecture with one master node and one slave node. The instance classes of the nodes are the same.
Read replica	Uses a single-node architecture (without a standby node).
Cluster	Uses the Microsoft Always On architecture with one master node, one slave node, and up to five read-only nodes. It features higher availability, reliability, and scalability.

You can use RDS to create and manage DB instances running various DB engines.

For details about differences and function comparison between different instance types, see [DB Instance Introduction](#) and [Function Comparison](#).

1.5.2 DB Instance Classes

general-enhanced II instance classes provide robust and stable performance. They use latest-generation network acceleration engines and Data Plane Development Kit (DPDK) to provide higher network performance, meeting requirements in different scenarios.

General-enhanced II DB instances use second-generation Intel® Xeon® Scalable processors with technologies optimized and 25GE high-speed intelligent NICs to offer powerful and stable computing performance, including ultra-high network bandwidth and PPS.

Table 1-4 DB instance classes

Instance Class	vCPUs	Memory (GB)	Supported DB Engine
General-enhanced II	1	2	- MySQL - PostgreSQL
	1	4	- MySQL - PostgreSQL
	1	8	MySQL
	2	4	MySQL
	2	8	- MySQL - PostgreSQL - Microsoft SQL Server
	2	16	- MySQL - PostgreSQL - Microsoft SQL Server
	4	8	- MySQL - PostgreSQL - Microsoft SQL Server
	4	16	- MySQL - Microsoft SQL Server
	4	32	- MySQL - Microsoft SQL Server
	8	32	- MySQL - PostgreSQL - Microsoft SQL Server
	8	64	- MySQL - Microsoft SQL Server
	16	32	MySQL
	16	64	- MySQL - PostgreSQL - Microsoft SQL Server

Instance Class	vCPUs	Memory (GB)	Supported DB Engine
	16	128	- MySQL - Microsoft SQL Server
	24	192	Microsoft SQL Server
	32	64	MySQL
	32	128	- MySQL - PostgreSQL - Microsoft SQL Server
	60	128	- MySQL - PostgreSQL
	60	256	- MySQL - PostgreSQL - Microsoft SQL Server
	60	512	MySQL

The DB instance specifications vary according to site requirements.

1.5.3 DB Engines and Versions

Table 1-5 lists the DB engines and versions supported by RDS.

During new application rollout, you are advised to use the latest major version of the DB engine. For example, if the MySQL engine is used, you are advised to select MySQL 5.7. When you create a DB instance, you can select a major DB engine version only (such as MySQL 5.7). The system will automatically select an appropriate minor version (such as 5.7.31). After the DB instance is created, you can view the minor version in the **DB Engine Version** column on the **Instance Management** page. The DB engine and version vary according to site requirements.

Table 1-5 DB engines and versions

DB Engine	Single	Primary/Standby	Cluster
MySQL	8.0 5.7 5.6	8.0 5.7 5.6	Not supported

DB Engine	Single	Primary/Standby	Cluster
PostgreSQL	13 - Enhanced Edition (PostgreSQL 1.0). For more information, see PostgreSQL Enhanced Edition. 12 11 10 9.6 9.5	13 - Enhanced Edition (PostgreSQL 1.0). For more information, see PostgreSQL Enhanced Edition. 12 11 10 9.6 9.5	Not supported
Microsoft SQL Server	2019 EE 2019 SE 2017 EE 2017 Standard Edition 2016 EE 2016 SE 2014 SE 2014 EE	2019 SE 2017 EE 2017 Standard Edition 2016 EE 2016 SE 2014 EE 2014 Standard Edition	2019 EE 2017 EE

1.5.4 DB Instance Statuses

DB Instance Statuses

The status of a DB instance indicates the health of the DB instance. You can use the management console or API to view the status of a DB instance.

Table 1-6 DB instance statuses

Status	Description
Available	A DB instance is available.
Abnormal	A DB instance is abnormal.
Creating	A DB instance is being created.
Creation failed	A DB instance has failed to be created.
Switchover in progress	A standby DB instance is being switched over to the primary DB instance.

Status	Description
Changing type to primary/standby	A single DB instance is being changed to primary/standby DB instances.
Rebooting	A DB instance is being rebooted.
Changing port	A DB instance port is being changed.
Changing instance class	The CPU or memory of a DB instance is being modified.
Scaling up	Storage space of a DB instance is being scaled up.
Backing up	A DB instance is being backed up.
Restoring	A DB instance is in the process of being restored from a backup.
Restore failed	A DB instance fails to be restored.
Storage full	Storage space of a DB instance is full. Data cannot be written to databases.
Deleted	A DB instance has been deleted and will not be displayed in the instance list.
Parameter change. Pending reboot	A modification to a database parameter is waiting for an instance reboot before it can take effect.

1.6 Typical Applications

1.6.1 Read/Write Splitting

For MySQL and PostgreSQL, the primary DB instances and read replicas have independent connection addresses. A maximum of five read replicas can be created for each primary MySQL or PostgreSQL DB instance. For details about how to create a read replica, see [Creating a Read Replica](#) and [Managing a Read Replica](#).

To improve the system processing capability, you can simply create read replicas without changing your existing applications.

1.7 Constraints

1.7.1 MySQL Constraints

Table 1-7 shows the constraints designed to ensure the stability and security of RDS for MySQL.

Table 1-7 Function constraints

Function Item	Constraints
Database access	• If public accessibility is not enabled, the RDS DB instance must be in the same VPC as the ECS. • RDS read replicas must be created in the same subnet as the primary DB instance. • The security group must allow access from the ECS. By default, RDS cannot be accessed through an ECS in a different security group. You need to add an inbound rule to the RDS security group. • The default MySQL port is 3306. You can if you want to access MySQL through another port.
Deployment	ECSs in which DB instances are deployed are not visible to users. You can access the DB instances only through an IP address and a port number.
Database root permissions	Only the root user permissions are provided on the instance creation page.
Database parameter modification	Most parameters can be modified on the RDS console.
Data migration	Use MySQL CLI tools to migrate data by referring to Migrating Data to RDS for MySQL Using mysqldump .
MySQL storage engine	For details, see What Storage Engines Does the RDS for MySQL Support?
Database replication setup	RDS for MySQL uses a primary/standby dual-node replication cluster. You do not need to set up replication additionally. The standby DB instance is not visible to users and therefore you cannot access it directly.
Minor version upgrade	Currently, RDS for MySQL supports a maximum of 100,000 tables. If the number of tables is greater than 100,000, the minor version upgrade may fail.
DB instance reboot	RDS DB instances cannot be rebooted through commands. They must be rebooted through the RDS console.
RDS backup files	For details, see Downloading a Backup File .

1.7.2 PostgreSQL Constraints

[Table 1-8](#) shows the constraints designed to ensure the stability and security of RDS for PostgreSQL.

Table 1-8 Function constraints

Function Item	Constraints
Database access	• If public accessibility is not enabled, the RDS DB instance must be in the same VPC as the ECS. • RDS read replicas must be created in the same subnet as the primary DB instance. • The security group must allow access from the ECS. By default, RDS cannot be accessed through an ECS in a different security group. You need to add an inbound rule to the RDS security group. • The default RDS port is 5432. You can change it if you want to access RDS through another port.
Deployment	ECSs in which DB instances are deployed are not visible to users. You can access the DB instances only through an IP address and a port number.
Database root permissions	Only the root user permissions are provided on the instance creation page.
Database parameter modification	Most parameters can be modified on the RDS console.
Data migration	Use <code>psql</code> to import data by referring to Migrating Data to RDS for PostgreSQL Using psql .
Database replication setup	RDS for PostgreSQL uses a primary/standby dual-node replication cluster. You do not need to set up replication additionally. The standby DB instance is not visible to users and therefore you cannot access it directly.
DB instance reboot	DB instances cannot be rebooted through commands. They must be rebooted through the RDS console.
RDS backup files	For details, see Downloading a Full Backup File .

1.7.3 Microsoft SQL Server Constraints

RDS for SQL Server only supports DB instances under the License Included model and does not support "bring your own license" (BYOL). After a DB instance is created, it contains the Microsoft SQL Server software license.

[Table 1-9](#) shows the constraints designed to ensure the stability and security of RDS for SQL Server.

Microsoft SQL Server DB instances are classified into three types: single, primary/standby, and cluster. Different types support different functions.

Table 1-9 Function constraints

Function Item	Single	Primary/Standby
Maximum number of databases	100	100
Number of database accounts	Unlimited	Unlimited
Creation of user, LOGIN, or database	Supported	Supported
Database-level DDL trigger	Supported	Supported
Database permission authorization	Supported	Supported
KILL permission	Supported	Supported
SQL Profiler	Supported	Supported
Tuning Adviser	Supported	Supported
Change Data Capture (CDC)	Supported	Supported
Change tracking	Supported	Supported
Windows domain account login	Supported	Supported
SQL Server Integration Services (SSIS)	Not supported	Not supported
SQL Server Analysis Services (SSAS)	Not supported	Not supported
R Services	Not supported	Not supported
Asynchronous communication	Not supported	Not supported
Replication subscription	Not supported	Not supported
Policy management	Not supported	Not supported

1.8 Related Services

Table 1-10 Related services

Service Name	Description
Elastic Cloud Service (ECS)	Enables you to access RDS DB instances through an internal network. You can then access applications faster and you do not need to pay for public network traffic.
Virtual Private Cloud (VPC)	Isolates your networks and controls access to your RDS DB instances.
Object Storage Service (OBS)	Stores automated and manual backups of your RDS DB instances.

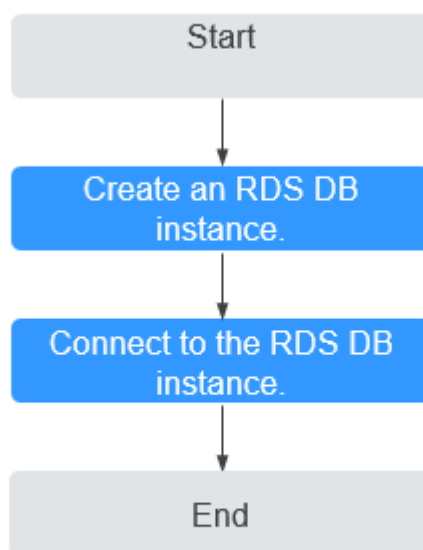
2 Getting Started with RDS for MySQL

2.1 Operation Guide

You can create and connect to DB instances on the RDS console.

Flowchart

Figure 2-1 Flowchart



Procedure

Table 2-1 Related operations and references

Operation	Reference
Creating an RDS DB instance	Step 1: Create a DB Instance
Connecting to an RDS DB Instance	Step 2: Connect to a DB Instance

2.2 Step 1: Create a DB Instance

Scenarios

This section describes how to create a DB instance on the management console.

The DB instance class and storage space you need depend on your processing power and memory requirements.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** On the **Instance Management** page, click **Create DB Instance**.
- Step 4** On the displayed page, configure information about your DB instance. Then, click **Create Now**.

Table 2-2 Basic information

Parameter	Description
Region	The region where your RDS resources will be located. You can change it on the creation page, or go back to the Instance Management page and change it in the upper left corner. NOTE Products in different regions cannot communicate with each other through a private network. After a DB instance is created, the region cannot be changed. Therefore, exercise caution when selecting a region.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
DB Engine	Set to MySQL .

Parameter	Description
DB Engine Version	For details, see DB Engines and Versions. Different DB engine versions are supported in different regions. You are advised to select the latest available version because it is more stable, reliable, and secure.
DB Instance Type	Primary/Standby: uses an HA architecture with a primary DB instance and a synchronous standby DB instance. It is suitable for production databases of large- and medium-sized enterprises in Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other sectors. The standby DB instance improves instance reliability and is invisible to you after being created. An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network. You can deploy primary and standby DB instances in a single AZ or across AZs to achieve failover and high availability. Single: uses a single-node architecture, which is more cost-effective than primary/standby DB instances. It is suitable for development and testing of microsites, and small- and medium-sized enterprises, or for learning about RDS.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. High I/O: supports a maximum throughput of 150 MB/s. Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Time Zone	Select a time zone when you are creating a DB instance, and you can change it after the DB instance is created.

Table 2-3 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and different maximum IOPS. For details about instance classes, see section DB Instance Classes. After a DB instance is created, you can change its instance class. For details, see section Changing a DB Instance Class.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Storage Space (GB)	Contains the file system overhead required for inode, reserved block, and database operation. After a DB instance is created, you can scale up its storage space. For details, see section Scaling up Storage Space. NOTE • Storage space can range in size from 40 GB to 4,000 GB and can be scaled up only by a multiple of 10 GB.

Table 2-4 Network

Parameter	Description
VPC	A dedicated virtual network in which your RDS DB instances are located. A VPC can isolate networks for different services. You can select an existing VPC or create a VPC. For details on how to create a VPC, see the "Creating a VPC" section in the <i>Virtual Private Cloud User Guide</i>. If no VPC is available, RDS allocates a VPC to you by default. NOTICE After the DB instance is created, the VPC cannot be changed.
Subnet	Improves network security by providing dedicated network resources that are logically isolated from other networks. Subnets take effect only within an AZ. The Dynamic Host Configuration Protocol (DHCP) function is enabled by default for subnets in which you plan to create RDS DB instances and cannot be disabled. A floating IP address is automatically assigned when you create a DB instance. You can also enter an unused floating IP address in the subnet CIDR block. After the DB instance is created, you can change the floating IP address.

Parameter	Description
Security Group	Enhances security by controlling access to RDS from other services. You need to add inbound rules to a security group so that you can connect to your DB instance. If no security group is available or has been created, RDS allocates a security group to you by default.

Table 2-5 Database configuration

Parameter	Description
Administrator	The default login name for the database is root .
Administrator Password	Must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#\$\$%^*_-=+?,). Enter a strong password and periodically change it for security reasons. Keep this password secure. The system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see section Resetting the Administrator Password .
Confirm Password	Must be the same as Administrator Password .
Parameter Template	Contains engine configuration values that can be applied to one or more DB instances. If you intend to create primary/standby DB instances, they use the same parameter template. NOTICE If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not delivered. Instead, the default values are used. • back_log • innodb_io_capacity_max • max_connections • innodb_io_capacity • innodb_buffer_pool_size • innodb_buffer_pool_instances You can modify the instance parameters as required after the DB instance is created. For details, see section Modifying Parameters .

Parameter	Description
Enterprise Project	If the DB instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the ProjectMan console to create a project. For details about how to create a project, see the <i>ProjectMan User Guide</i>.

Table 2-6 Batch creation

Parameter	Description
Quantity	RDS supports DB instance creation in batches. If you choose to create primary/standby DB instances and set Quantity to 1, a primary DB instance and a standby DB instance will be created synchronously. If you create multiple DB instances at a time, they will be named with four digits appended to the DB instance name. For example, if you enter instance, the first instance will be named instance-0001, the second instance-0002, and so on.

Table 2-7 Read replicas

Parameter	Description
Read Replica	By default, read replicas are named with "read" and two digits appended to the primary DB instance name. For example, if the primary instance name is instance-0001, the first read replica will be named instance-0001-read-01. The network and storage configurations are the same as those of the primary DB instance.
Read Replica AZ	By default, the primary DB instance and read replicas are deployed in different AZs. You can choose AZs as required. NOTICE Products in different regions cannot communicate with each other through a private network and you cannot change the region of a DB instance after creating the instance. Therefore, exercise caution when selecting a region.
Instance Class	Refers to the CPU and memory of a DB instance.

Parameter	Description
Read Replica Quantity	You can choose to create read replicas when you are creating a primary DB instance. After the primary DB instance is created, the system automatically triggers the creation of read replicas. You can create a maximum of five read replicas for each primary DB instance. If you intend to create primary/standby DB instances and set Read Replica Quantity to 1, a primary/standby DB instance and a read replica will be created.

 NOTE

The performance of your DB instance depends on its configurations. Hardware configuration items include the instance specifications, storage type, and storage space.

Step 5 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 6 To view and manage the DB instance, go to the **Instance Management** page.

- During the creation process, the DB instance status is **Creating**. When the creation process is complete, the instance status will change to **Available**. You can view the detailed progress and result of the task on the **Task Center** page.
- The automated backup policy is enabled by default. After the DB instance is created, you can modify the automated backup policy. An automated full backup is immediately triggered after a DB instance is created.
- The default database port is **3306**. After a DB instance is created, you can change its port.

For details, see section [Changing a Database Port](#).

----End

2.3 Step 2: Connect to a DB Instance

2.3.1 Connecting to a DB Instance

An RDS DB instance can be connected through a private network or a public network.

Table 2-8 RDS connection methods

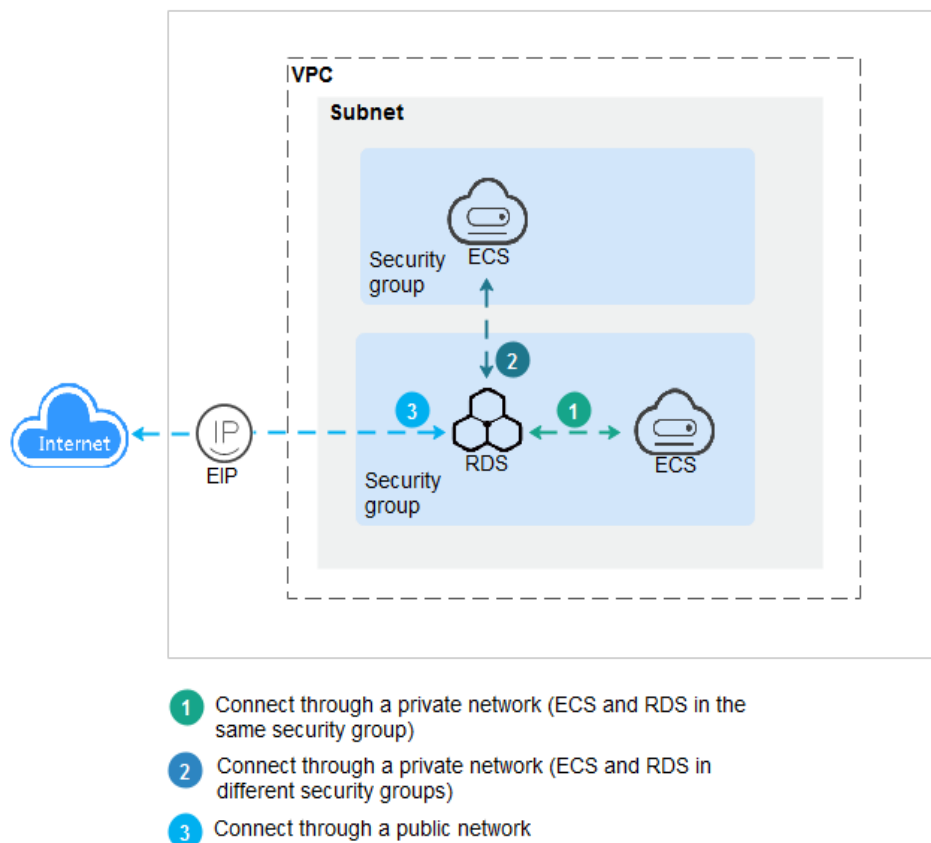
Connect Through	IP Address	Scenarios	Description
Private network	Floating IP	RDS provides a floating IP address by default. When your applications are deployed on an ECS that is in the same region and VPC as RDS, you are advised to use a floating IP address to connect to the RDS DB instance through the ECS.	- Secure and excellent performance - Recommended
Public network	EIP	If you cannot access an RDS DB instance through a floating IP address, bind an EIP to the DB instance and connect the DB instance to the ECS through the EIP.	- A relatively lower level of security compared to other connection methods - To achieve a higher transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same VPC as your RDS DB instance and use a floating IP address to access the DB instance.

 NOTE

- VPC: indicates the Virtual Private Cloud.
- ECS: indicates the Elastic Cloud Server.
- If the ECS is in the same VPC as your RDS DB instance, you do not need to apply for an EIP.

Figure 2-2 illustrates the connection over a private network or a public network.

Figure 2-2 DB instance connection



2.3.2 Connecting to a DB Instance Through a Private Network

2.3.2.1 Connecting to a DB Instance Through a Private Network

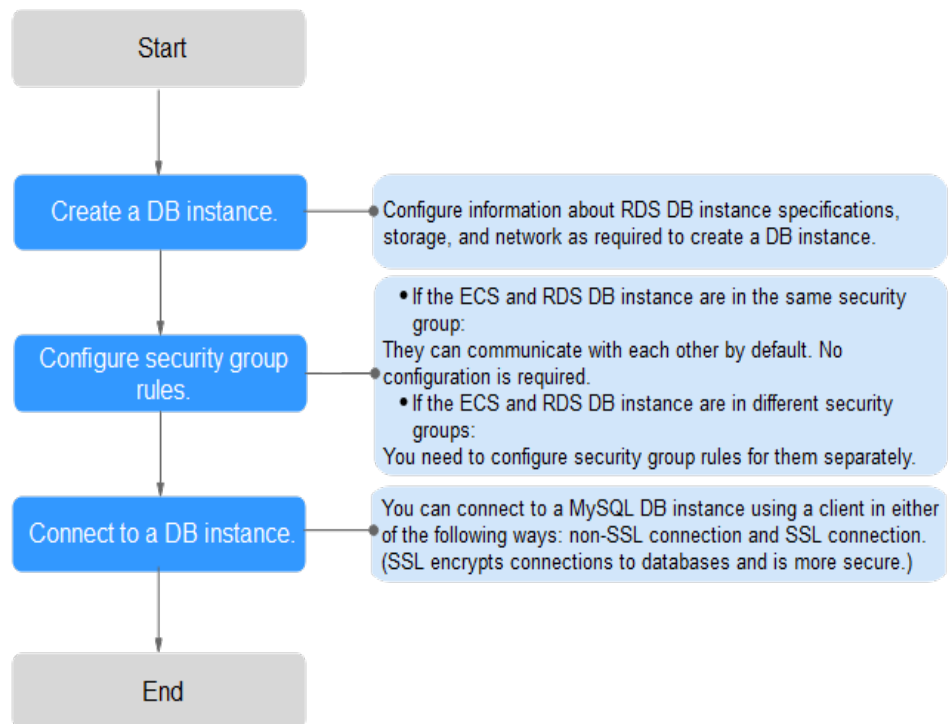
Scenarios

If you are using RDS for the first time, see the constraints described in section [MySQL Constraints](#).

Process

Figure 2-3 illustrates the process of connecting to a MySQL DB instance through a private network.

Figure 2-3 Connecting to a DB instance through a private network



2.3.2.2 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

First check whether the ECS and RDS DB instance are in the same security group.

- If the ECS and RDS DB instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured. Go to [Connecting to a DB Instance Through a Private Network](#).
- If the ECS and RDS DB instance are in different security groups, you need to configure security group rules for them, separately.
 - RDS DB instance: Configure an **inbound rule** for the security group with which the RDS DB instance is associated.
 - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security rule for the ECS. If not all outbound traffic is allowed in the security group, you need to configure an **outbound rule** for the ECS.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group.

After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, a tenant can create a maximum of 100 security groups.
- By default, each security group can have a maximum of 50 security group rules.
- One security group can be associated with only one RDS DB instance.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 On the **Instance Management** page, click the target DB instance.

Step 4 Configure a security group rule.

- In the **Connection Information** area on the **Basic Information** page, click the security group.

Figure 2-4 Connection information

Connection Information		Connection Management	
Floating IP Address	172.16.206.16  Change	Private Domain Name	
VPC	vpc-4e70	Database Port	3306  
Subnet	subnet-4f2f-16 (172.16.0.0/16)	Recommended Max. Connections	10,000
Security Group	default_securitygroup 	Read/Write Splitting Address	Apply

Step 5 On the inbound rule tab, click **Add Rule**. In the displayed dialog box, set required parameters to add an inbound rule.

You can click **+** to add more inbound rules.

Figure 2-5 Adding an inbound rule

Add Inbound Rule [Learn more](#) about security group configuration.

An inbound rule allows inbound traffic to instances in the security group.

Security Group default_securitygroup

You can import multiple rules in a batch.

Protocol & Port ?	Source ?	Operation
Custom TCP Example: 22 or 22-30	IP address 0 . 0 . 0 . 0 / 0	Operation

+

 Add Rule You can create 4,985 more security group rules. [Increase quota](#)

OK

Cancel

Table 2-9 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	Protocol: specifies the network protocol. Currently, the value can be All , TCP , UDP , ICMP , GRE , or others.	Custom TCP
	Port: specifies the port or port range over which the traffic can reach your ECS.	When connecting to the DB instance through a private network, enter the port of the target DB instance.
Source	Specifies the source of the security group rule. The value can be an IP address or a security group. For example: - xxx.xxx.xxx.xxx/32 (IPv4 address) - xxx.xxx.xxx.0/24 (subnet) 0.0.0.0/0 (any IP address)	0.0.0.0/0
Description	Provides supplementary information about the security group rule. This parameter is optional. The description can contain a maximum of 255 characters and cannot contain angle brackets (<) or (>).	N/A

----End

2.3.2.3 Connecting to a DB Instance Through a Private Network

You can connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection encrypts data and is more secure.

Prerequisites

1. You have logged in to an ECS.
 - To connect to a DB instance through an ECS, you must ensure that:
 - The ECS and DB instance must be in the same VPC.
 - The ECS must be allowed by the security group to access the DB instance.
 - If the security group with which the target DB instance is associated is the default security group, you do not need to configure security group rules.
 - If the security group with which the target DB instance is associated is not the default security group, check whether the security group rules allow the ECS to connect to the DB instance. For details, see section [Configuring Security Group Rules](#).
If the security group rules allow the access from the ECS, the ECS can connect to the DB instance.
If the security group rules do not allow the access from the ECS, you need to add a security group rule. The ECS must be allowed by the security group to access the DB instance.
2. You have installed a database client to connect to DB instances.

You can use a database client to connect to the target DB instance in the Linux or Windows OS.

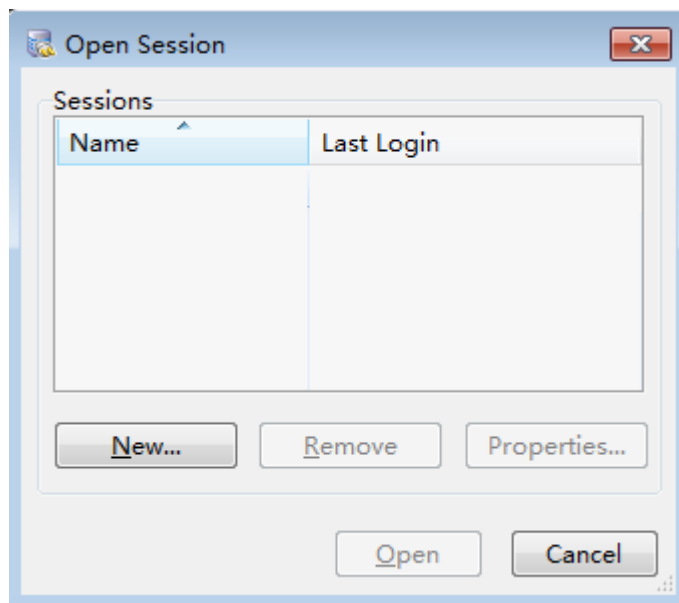
 - In the Linux OS, install the MySQL client on the device that can access RDS. It is recommended that you download a MySQL client running a version later than that of the DB instance.
For details about how to obtain and install the MySQL client, see section [How Can I Install the MySQL Client?](#)
 - In the Windows OS, you can use any common database client to connect to the target DB instance in a similar way.
The database client MySQL-Front is used as an example in [Using MySQL-Front to Connect to a DB Instance](#).

Using MySQL-Front to Connect to a DB Instance

Step 1 Start MySQL-Front.

Step 2 In the displayed dialog box, click **New**.

Figure 2-6 Connection management



Step 3 Enter the information of the DB instance to be connected and click **Ok**, as shown in [Figure 2-7](#).

Figure 2-7 Adding an account

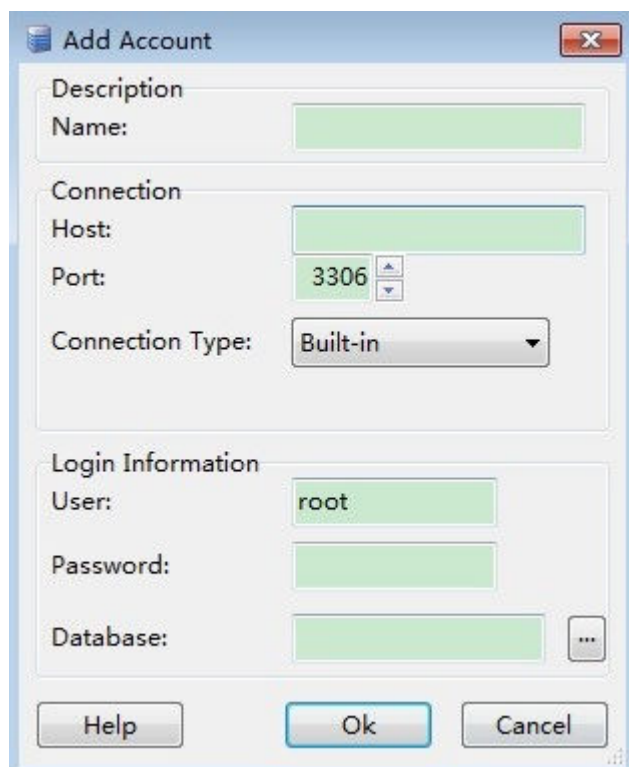
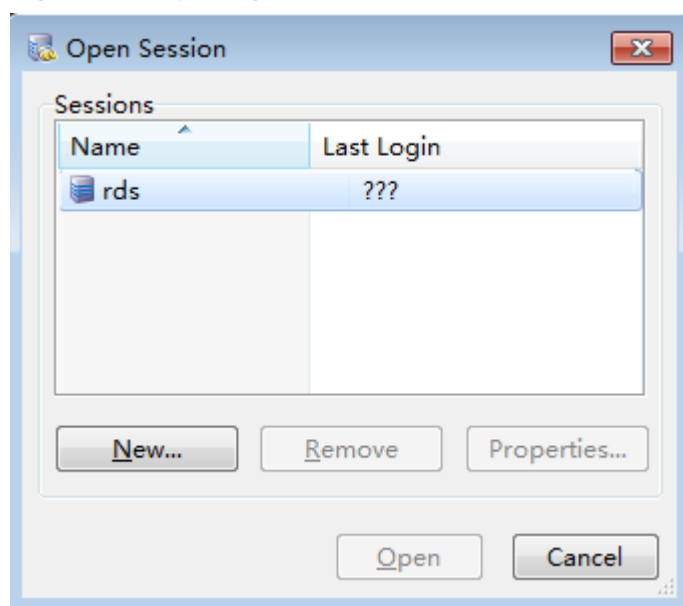


Table 2-10 Parameter description

Parameter	Description
Name	Indicates the name of the database connection task. If you do not set this parameter, it will be the same as the Host value by default.
Host	Indicates the floating IP address of the DB instance to be connected. To view the floating IP address and port of the DB instance, perform the following steps: 1. Log in to the RDS console. 2. Select the region in which the DB instance is located. 3. Click the target DB instance to enter the Basic Information page. 4. In the Connection Information area, view the floating IP address.
Port	Indicates the private network port of the DB instance.
User	Indicates the name of the user who will access the DB instance. The default user is root .
Password	Indicates the password of the RDS database account.

Step 4 In the displayed window, select the connection that you have created in [Step 3](#) and click **Open**. If the connection information is correct, the DB instance is successfully connected.

Figure 2-8 Opening a session



NOTE

If the connection fails, see [What Should I Do If an ECS Cannot Connect to an RDS DB Instance Through a Private Network?](#)

----End

2.3.3 Connecting to a DB Instance Through a Public Network

2.3.3.1 Connecting to a DB Instance Through a Public Network

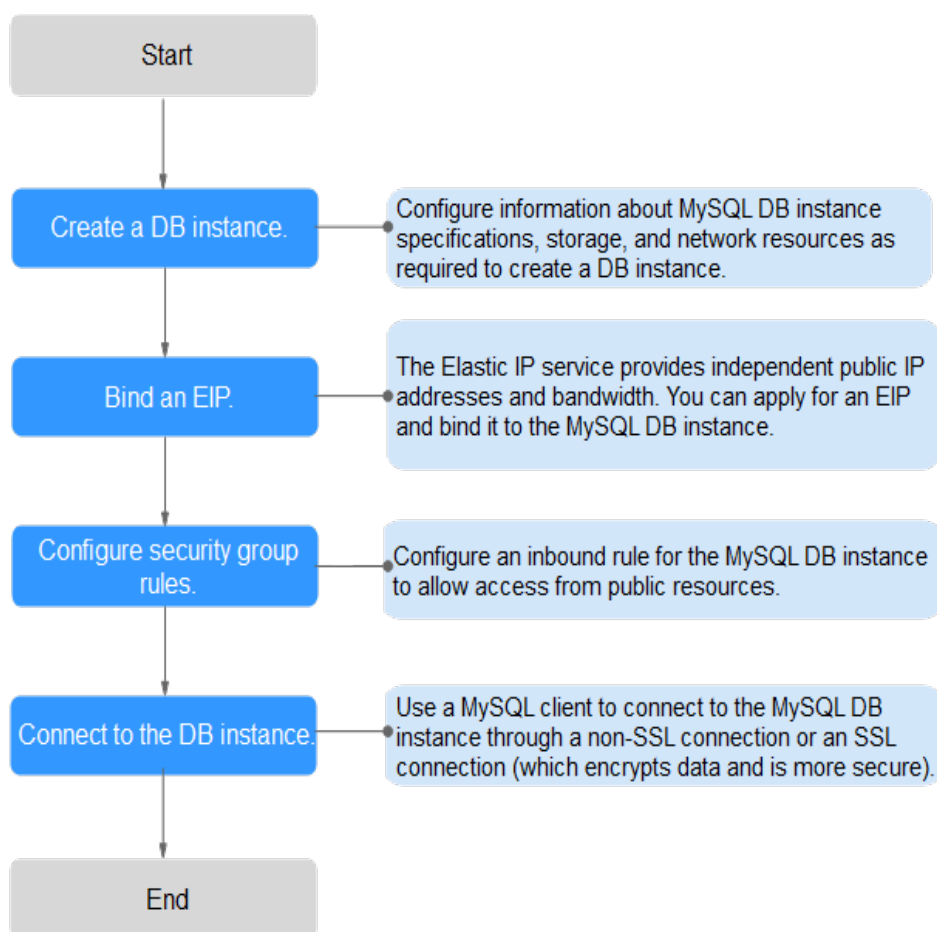
Scenarios

If you are using RDS for the first time, see the constraints described in section [MySQL Constraints](#).

Process

Figure 2-9 illustrates the process of connecting to a MySQL DB instance through a public network.

Figure 2-9 Connecting to a DB instance through a public network



2.3.3.2 Binding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility and can unbind the EIP from the DB instance as required.

Precautions

- You need to configure security groups and enable specific IP addresses and ports to access the target DB instance. Before accessing the DB instance, you need to add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see section [Configuring Security Group Rules](#).
- Public accessibility reduces the security of DB instances. Therefore, exercise caution when enabling this function. To achieve a higher transmission rate and security level, you are advised to migrate your applications to the ECS that is in the same region as RDS.

Binding an EIP

Step 1 On the **Instance Management** page, click the target DB instance.

Step 2 In the navigation pane on the left, choose **EIPs**. On the displayed page, click **Bind EIP**.

Step 3 In the displayed dialog box, select an EIP and click **OK**.

If no available EIPs are displayed, click **View EIP** to obtain an EIP.

Step 4 , view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

----End

2.3.3.3 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

When you attempt to connect to an RDS DB instance through an EIP, you need to configure an **inbound rule** for the security group associated with the DB instance.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group.

After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, a tenant can create a maximum of 100 security groups.
- By default, each security group can have a maximum of 50 security group rules.
- One security group can be associated with only one RDS DB instance.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 On the **Instance Management** page, click the target DB instance.

Step 4 Configure security group rules.

- In the **Connection Information** area on the **Basic Information** page, click the security group.

Figure 2-10 Connection information

Connection Information		Connection Management	
Floating IP Address	172.16.206.16  Change	Private Domain Name	
VPC	vpc-4e70	Database Port	3306  
Subnet	subnet-4f2f-16 (172.16.0.0/16)	Recommended Max. Connections	10,000
Security Group	default_securitygroup 	Read/Write Splitting Address	Apply

Step 5 On the inbound rule tab, click **Add Rule**. In the displayed dialog box, set required parameters to add an inbound rule.

You can click **+** to add more inbound rules.

Operation ▼

Parameter	Description	Example Value
Protocol & Port	Protocol: specifies the network protocol. Currently, the value can be All , TCP , UDP , ICMP , GRE , or others.	Custom TCP
	Port: specifies the port or port range over which the traffic can reach your ECS.	When connecting to the DB instance through a public network, enter the port of the target DB instance.
Source	Specifies the source of the security group rule. The value can be an IP address or a security group. For example: • xxx.xxx.xxx.xxx/32 (IPv4 address) • xxx.xxx.xxx.0/24 (subnet) • 0.0.0.0/0 (any IP address)	0.0.0.0/0
Description	Provides supplementary information about the security group rule. This parameter is optional. The description can contain a maximum of 255 characters and cannot contain angle brackets (<) or (>).	-

----End

2.3.3.4 Connecting to a DB Instance Through a Public Network

You can connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection encrypts data and is more secure.

Prerequisites

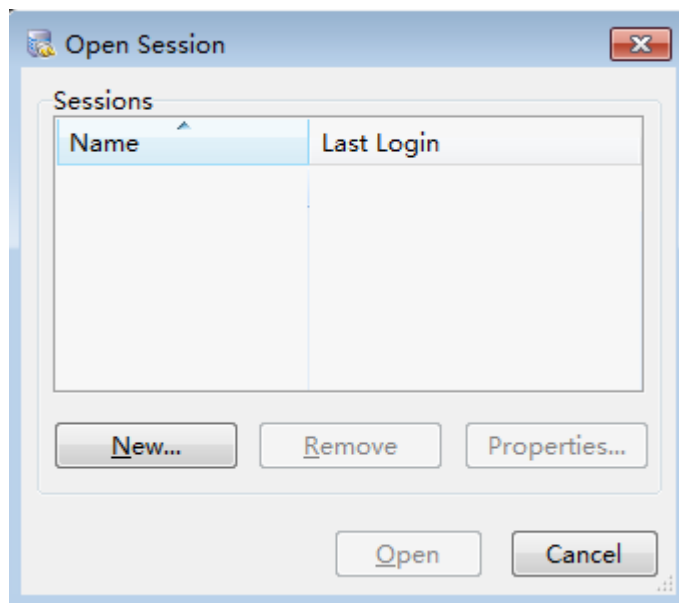
1. An EIP has been bound to the target DB instance and security group rules have been configured.
 - a. Bind an EIP to the target DB instance.
For details about how to bind an EIP, see section [Binding an EIP](#).
 - b. Obtain the IP address of the local device.
 - c. Configure security group rules.
Add the IP address obtained in [1.b](#) and the instance port to the inbound rule of the security group.
For details about how to configure a security group rule, see section [Configuring Security Group Rules](#).
 - d. Run the **ping** command to check the connectivity between the local device and the EIP that has been bound to the DB instance in [1.a](#).
2. You have installed a database client to connect to DB instances.
You can use a database client to connect to the target DB instance in the Linux or Windows operating system (OS).
 - In the Linux OS, you need to install a MySQL client on the ECS. It is recommended that you download a MySQL client running a version later than that of the DB instance.
For details about how to obtain and install the MySQL client, see section [How Can I Install the MySQL Client?](#)
 - In the Windows OS, you can use any common database client to connect to the target DB instance in a similar way.
The database client MySQL-Front is used as an example in [Using MySQL-Front to Connect to a DB Instance](#).

Using MySQL-Front to Connect to a DB Instance

Step 1 Start MySQL-Front.

Step 2 In the displayed dialog box, click **New**.

Figure 2-12 Connection management



Step 3 Enter the information of the DB instance to be connected and click **Ok**, as shown in [Figure 2-13](#).

Figure 2-13 Adding an account

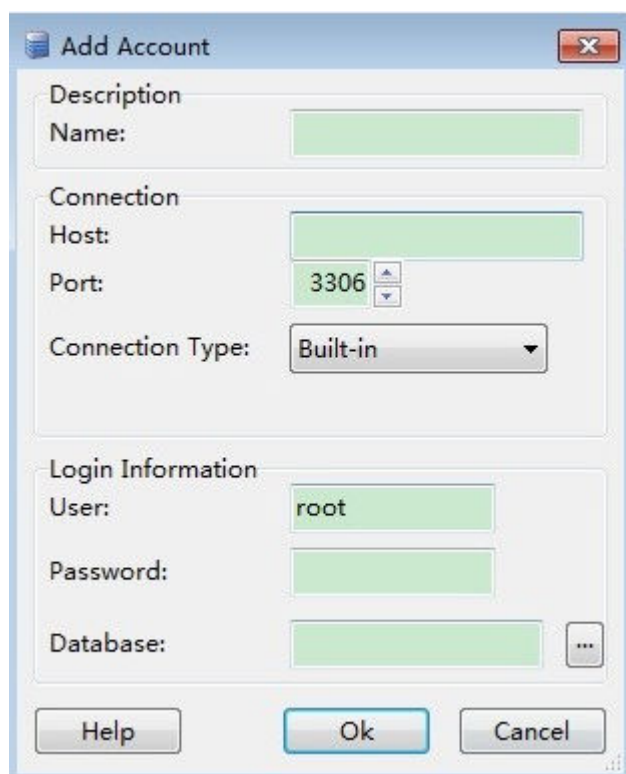


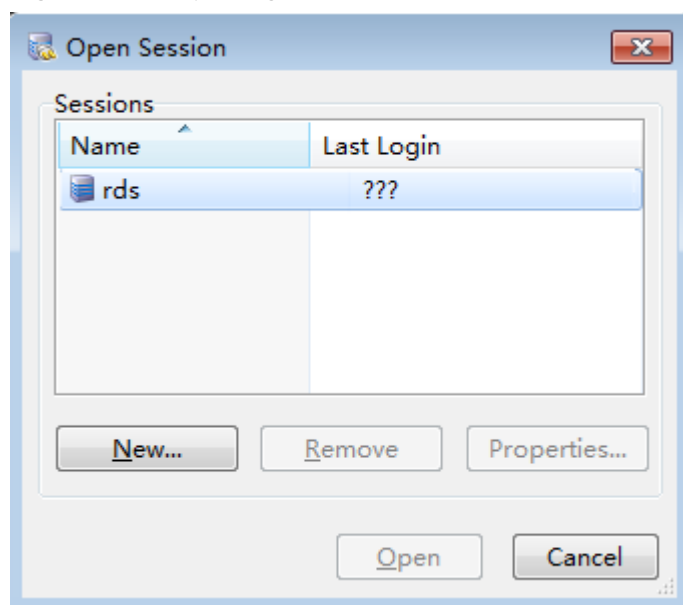
Table 2-12 Parameter description

Parameter	Description
Name	Indicates the name of the database connection task. If you do not set this parameter, it will be the same as the Host value by default.
Host	Indicates the EIP of the DB instance to be connected.
Port	Indicates the private network port of the DB instance.
User	Indicates the name of the user who will access the DB instance. The default user is root .
Password	Indicates the password of the RDS database account.

Step 4 In the displayed window, select the connection that you have created in [Figure 2-14](#) and click **Open**.

If the connection information is correct, the DB instance is successfully connected.

Figure 2-14 Opening a session



NOTE

If the connection fails, ensure that preparations have been correctly made in [Prerequisites](#) and try again.

----End

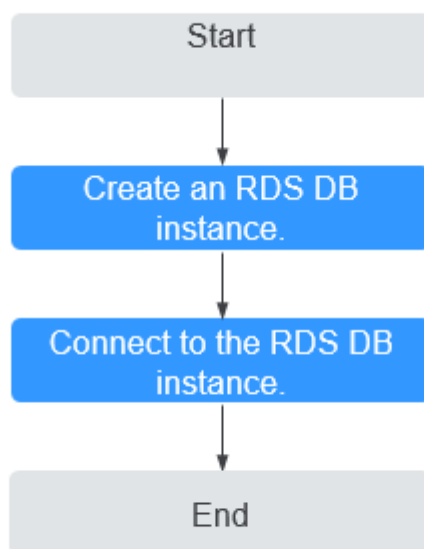
3 Getting Started with RDS for PostgreSQL

3.1 Operation Guide

You can create and connect to DB instances on the RDS console.

Flowchart

Figure 3-1 Flowchart



Procedure

Table 3-1 Related operations and references

Operation	Reference
Creating an RDS DB instance	Step 1: Create a DB Instance
Connecting to an RDS DB Instance	Step 2: Connect to a DB Instance

3.2 Step 1: Create a DB Instance

Scenarios

This section describes how to create a DB instance on the RDS console.

RDS allows you to tailor your computing resources and storage space to your business needs.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click **Create DB Instance**.
- Step 5** On the displayed page, configure information about your DB instance. Then, click **Create Now**.

Table 3-2 Basic information

Parameter	Description
Region	The region where your RDS resources will be located. You can change it on the creation page, or go back to the Instance Management page and change it in the upper left corner. NOTE Products in different regions cannot communicate with each other through a private network and you cannot change the region of a DB instance after creating the instance. Therefore, exercise caution when selecting a region.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
DB Engine	Set to PostgreSQL .
DB Engine Version	For details, see DB Engines and Versions. Different DB engine versions are supported in different regions. You are advised to select the latest available version because it is more stable, reliable, and secure.
DB Instance Type	Primary/Standby: uses an HA architecture with a primary DB instance and a synchronous standby DB instance. It is suitable for production databases of large- and medium-sized enterprises in Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other sectors. The standby DB instance improves instance reliability and is invisible to you after being created. An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network. You can deploy primary and standby DB instances in a single AZ or across AZs to achieve failover and high availability. Single: uses a single-node architecture, which is more cost-effective than primary/standby DB instances. It is suitable for development and testing of microsites, and small- and medium-sized enterprises, or for learning about RDS.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. High I/O: supports a maximum throughput of 150 MB/s. Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Time Zone	Select a time zone when you are creating a DB instance, and you can change it after the DB instance is created.

Table 3-3 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and different maximum IOPS. For details about instance classes, see DB Instance Classes. After a DB instance is created, you can change its CPU and memory. For details, see Changing a DB Instance Class.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • High I/O: supports a maximum throughput of 150 MB/s. • Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Storage Space (GB)	Contains the file system overhead required for inode, reserved block, and database operation. Storage space can range in size from 40 GB to 4,000 GB and can be scaled up only by a multiple of 10 GB. After a DB instance is created, you can scale up its storage space. For details, see Scaling up Storage Space.

Table 3-4 Network

Parameter	Description
VPC	A dedicated virtual network in which your RDS DB instances are located. A VPC can isolate networks for different services. You can select an existing VPC or create a VPC. For details on how to create a VPC, see the "Creating a VPC" section in the <i>Virtual Private Cloud User Guide</i>. If no VPC is available, RDS allocates a VPC to you by default. NOTICE After the DB instance is created, the VPC cannot be changed.
Subnet	Improves network security by providing dedicated network resources that are logically isolated from other networks. Subnets take effect only within an AZ. The Dynamic Host Configuration Protocol (DHCP) function is enabled by default for subnets in which you plan to create RDS DB instances and cannot be disabled. A floating IP address is automatically assigned when you create a DB instance. You can also enter an unused floating IP address in the subnet CIDR block. After the DB instance is created, you can change the floating IP address.

Parameter	Description
Security Group	Controls the access that traffic has in and out of a DB instance. By default, the security group associated with the DB instance is authorized. Enhances security by controlling access to RDS from other services. You need to add inbound rules to a security group so that you can connect to your DB instance. If no security group is available, RDS allocates a security group to you by default.

Table 3-5 Database configuration

Parameter	Description
Administrator	The default login name for the database is root .
Administrator Password	Must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*_+=+?,.). Enter a strong password and periodically change it for security reasons. Keep this password secure. The system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see section Resetting the Administrator Password.
Confirm Password	Must be the same as Administrator Password .
Parameter Template	Contains engine configuration values that can be applied to one or more DB instances. If you intend to create primary/standby DB instances, they use the same parameter template. NOTICE If you use a custom parameter template when creating a DB instance, the following specification-related parameters in the custom template are not delivered. Instead, the default values are used. • maintenance_work_mem • shared_buffers • max_connections • effective_cache_size You can modify the instance parameters as required after the DB instance is created. For details, see Modifying Instance Parameters.

Parameter	Description
Enterprise Project	If the DB instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the ProjectMan console to create a project. For details about how to create a project, see the <i>ProjectMan User Guide</i>.

Table 3-6 Batch creation

Parameter	Description
Quantity	RDS supports DB instance creation in batches. If you choose to create primary/standby DB instances and set Quantity to 1, a primary DB instance and a standby DB instance will be created synchronously. If you create multiple DB instances at a time, they will be named with four digits appended to the DB instance name. For example, if you enter instance, the first instance will be named instance-0001, the second instance-0002, and so on.

 NOTE

The performance of your DB instance depends on its configurations. Hardware configuration items include the instance specifications, storage type, and storage space.

Step 6 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 7 To view and manage the DB instance, go to the **Instance Management** page.

- During the creation process, the DB instance status is **Creating**. When the creation process is complete, the instance status will change to **Available**.
- The automated backup policy is enabled by default. After the DB instance is created, you can modify the policy as needed. An automated full backup is immediately triggered after a DB instance is created.
- The default database port is **5432**. After a DB instance is created, you can change the database port.

For details, see [Changing a Database Port](#).

----End

3.3 Step 2: Connect to a DB Instance

3.3.1 Connecting to a DB Instance

An RDS DB instance can be connected through a private network or a public network.

Table 3-7 RDS connection methods

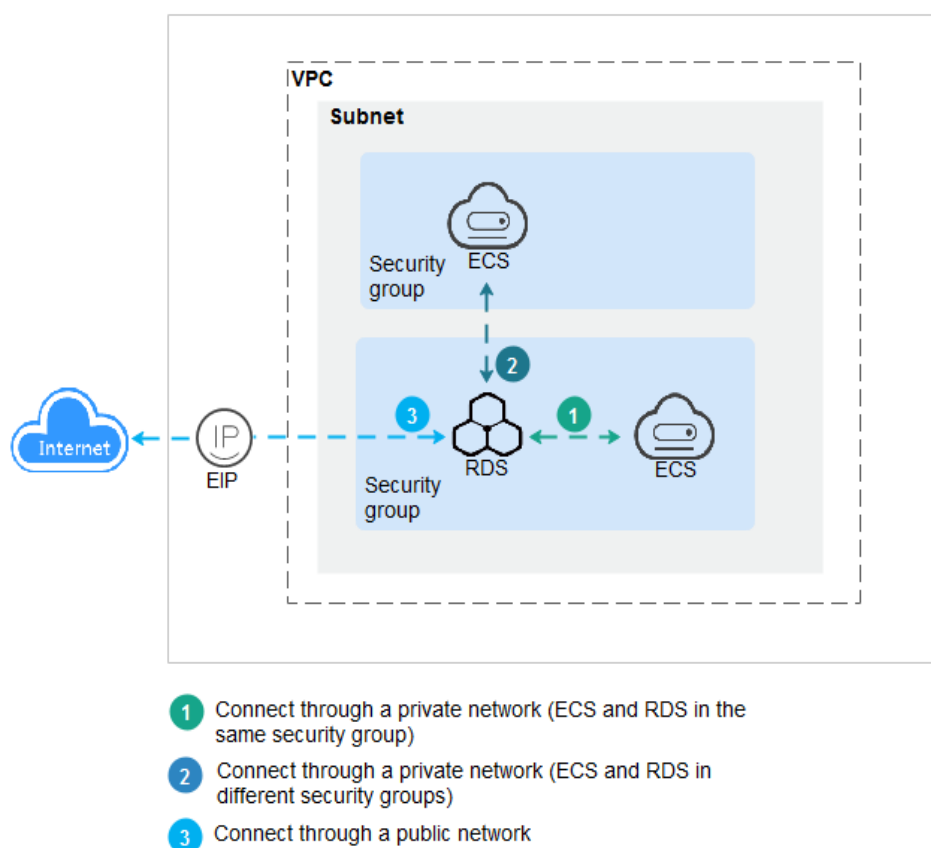
Connect Through	IP Address	Scenarios	Description
Private network	Floating IP	RDS provides a floating IP address by default. When your applications are deployed on an ECS that is in the same region and VPC as RDS, you are advised to use a floating IP address to connect to the RDS DB instance through the ECS.	- Secure and excellent performance - Recommended
Public network	EIP	If you cannot access an RDS DB instance through a floating IP address, bind an EIP to the DB instance and connect the DB instance to the ECS through the EIP.	- A relatively lower level of security compared to other connection methods - To achieve a higher transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same VPC as your RDS DB instance and use a floating IP address to access the DB instance.

NOTE

- VPC: indicates the Virtual Private Cloud.
- ECS: indicates the Elastic Cloud Server.
- If the ECS is in the same VPC as your RDS DB instance, you do not need to apply for an EIP.

Figure 3-2 illustrates the connection over a private network or a public network.

Figure 3-2 DB instance connection



3.3.2 Connecting to a PostgreSQL DB Instance Through a Private Network

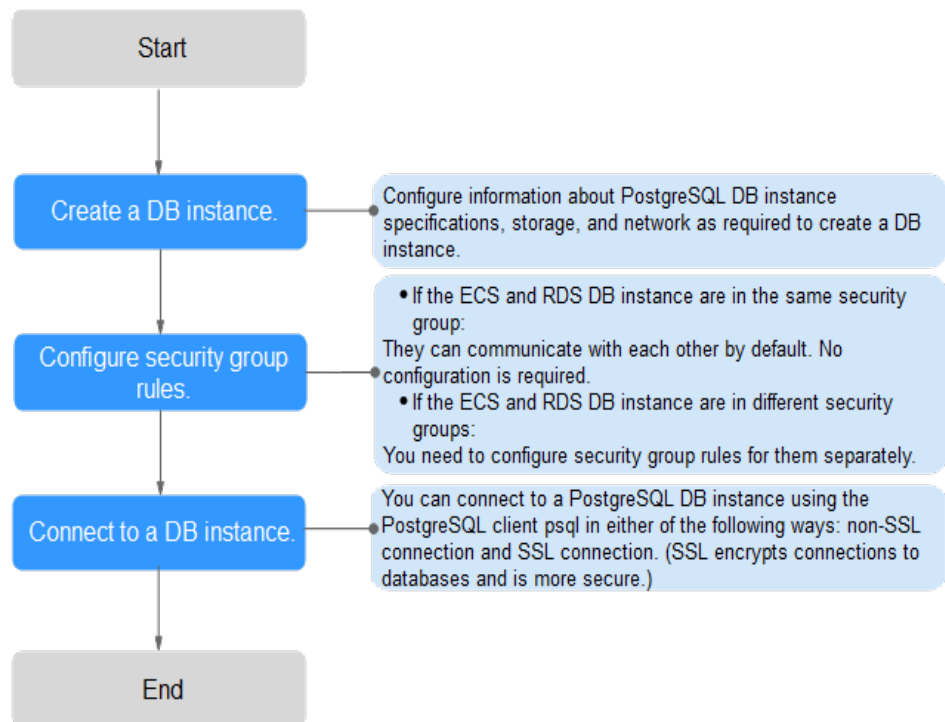
3.3.2.1 Connecting to a DB Instance Through a Private Network

If you are using RDS for the first time, see the constraints described in section [PostgreSQL Constraints](#).

Process

Figure 3-3 illustrates the process of connecting to a PostgreSQL DB instance through a private network.

Figure 3-3 Connecting to a DB instance through a private network



3.3.2.2 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

First check whether the ECS and RDS DB instance are in the same security group.

- If the ECS and RDS DB instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured. Go to [Connecting to a DB Instance Through psql](#).
- If the ECS and RDS DB instance are in different security groups, you need to configure security group rules for them, separately.
 - RDS DB instance: Configure an **inbound rule** for the security group with which the RDS DB instance is associated.
 - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security rule for the ECS. If not all outbound traffic is allowed in the security group, you need to configure an **outbound rule** for the ECS.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group.

After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, a tenant can create a maximum of 100 security groups.
- By default, each security group can have a maximum of 50 security group rules.
- One security group can be associated with only one RDS DB instance.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

 NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 On the **Instance Management** page, click the target DB instance.

Step 4 Configure security group rules.

In the **Connection Information** area, click the security group.

Figure 3-4 Connection information

Connection Information			
Floating IP Address	192.168.0.18  Change	VPC	default_vpc
Database Port	5432  	Subnet	default_subnet (192.168.0.0/24)
Recommended Max. Connections	2,048	Security Group	default_securitygroup 

Step 5 On the inbound rule tab, click **Add Rule**. In the displayed dialog box, set required parameters to add an inbound rule.

You can click **+** to add more inbound rules.

Table 3-8 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	Protocol: specifies the network protocol. Currently, the value can be All , TCP , UDP , ICMP , GRE , or others.	Custom TCP
	Port: specifies the port or port range over which the traffic can reach your ECS. The value ranges from 1 to 65535.	When connecting to the DB instance through a private network, enter the port of the target DB instance.
Source	Specifies the source of the security group rule. The value can be an IP address or a security group. For example: - xxx.xxx.xxx.xxx/32 (IPv4 address) - xxx.xxx.xxx.0/24 (subnet) 0.0.0.0/0 (any IP address)	0.0.0.0/0
Description	Provides supplementary information about the security group rule. This parameter is optional. The description can contain a maximum of 255 characters and cannot contain angle brackets (<) or (>).	-

----End

3.3.2.3 Connecting to a DB Instance Through psql

You can use the PostgreSQL client psql to connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection is encrypted and therefore more secure.

Prerequisites

1. You have logged in to the ECS.
 - To connect to a DB instance through an ECS, make sure that:
 - The ECS and DB instance must be in the same VPC.
 - The ECS must be allowed by the security group to access RDS DB instances.
 - If the security group with which the target DB instance is associated is the default security group, you do not need to configure security group rules.
 - If the security group with which the target DB instance is associated is not the default security group, check whether the security group rules allow the ECS to connect to the DB instance. For details, see [Configuring Security Group Rules](#).

If the security group rules allow the access from the ECS, the ECS can connect to the DB instance.

If the security group rules do not allow the access from the ECS, you need to add a security group rule. The ECS must be allowed by the security group to access DB instances.
2. You have installed a database client to connect to DB instances.
For details, see [How Can I Install the PostgreSQL Client?](#)

Non-SSL Connection

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Run the following command to connect to the DB instance:

```
psql --no-readline -U<user>-h<host>-p<port>-d<datastore>-W
```

Table 3-9 Parameter description

Parameter	Description
<user>	Indicates the username of the RDS database account. The default administrator is root .
<host>	Indicates the IP address of the primary DB instance. To obtain this parameter, go to the Basic Information page of the DB instance. If the DB instance is accessed through the ECS, the IP address can be found in the Floating IP Address field in the Connection Information area.

Parameter	Description
<i><port></i>	Indicates the database port in use. The default value is 5432 . To obtain this parameter, go to the Basic Information page of the DB instance. The port number can be found in the Database Port field in the Connection Information area.
<i><datastore></i>	Indicates the name of the database (the default database name is postgres).

The parameter **-W** indicates that a password must be entered for the connection. After running this command, you will be prompted to enter a password.

Example:

Run the following command as user **root** to connect to a DB instance:

```
psql --no-readline -U root -h 192.168.0.44 -p 5432 -d postgres -W
```

----End

3.3.3 Connecting to a PostgreSQL DB Instance Through a Public Network

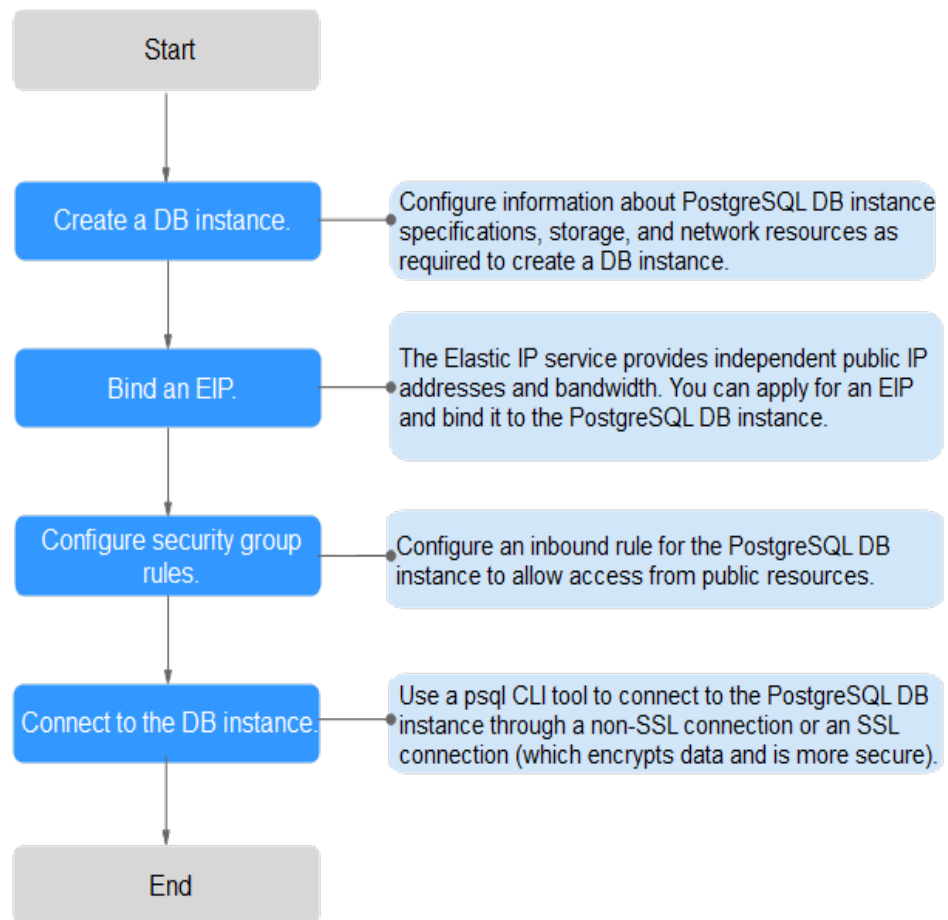
3.3.3.1 Connecting to a DB Instance Through a Public Network

If you are using RDS for the first time, see the constraints described in section [PostgreSQL Constraints](#).

Process

Figure 3-6 illustrates the process of connecting to a PostgreSQL DB instance through a public network.

Figure 3-6 Connecting to a DB instance through a public network



3.3.3.2 Binding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility and can unbind the EIP from the DB instance as required.

Precautions

- You need to configure security groups and enable specific IP addresses and ports to access the target DB instance. Before accessing the DB instance, you need to add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see section [Configuring Security Group Rules](#).
- Public accessibility reduces the security of DB instances. Therefore, exercise caution when deciding to connect to DB instances through a public network. To achieve a higher transmission rate and security level, you are advised to migrate your applications to the ECS that is in the same region as RDS.

Binding an EIP

Step 1 On the **Instance Management** page, click the target DB instance.

Step 2 In the navigation pane on the left, choose **EIPs**. On the displayed page, click **Bind EIP**.

Step 3 In the displayed dialog box, select an EIP and click **OK**.

If no available EIPs are displayed, click **View EIP** to obtain an EIP.

Step 4 On the **EIPs** page, view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

----End

3.3.3.3 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

When you attempt to connect to an RDS DB instance through an EIP, you need to configure an **inbound rule** for the security group associated with the DB instance.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group. After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, you can create a maximum of 500 security group rules.
- One security group can be associated with only one RDS DB instance.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Under **Network**, click **Virtual Private Cloud**.

- Step 2** In the navigation pane on the left, choose **Access Control > Security Groups**.
- Step 3** On the **Security Groups** page, locate the target security group and click **Manage Rule** in the **Operation** column.
- Step 4** On the displayed page, click **Add Rule**.
- Step 5** In the displayed dialog box, set required parameters to add an inbound rule.
- Step 6** Click **OK**.
- End

3.3.3.4 Connecting to a DB Instance Through psql

You can use the PostgreSQL client psql to connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection is encrypted and therefore more secure.

Prerequisites

1. An EIP has been bound to the target DB instance and security group rules have been configured.
 - a. Bind an EIP to the target DB instance.
For details about how to bind an EIP, see [Binding an EIP](#).
 - b. Obtain the IP address of a local device.
 - c. Configure security group rules.
Add the IP address obtained in [1.b](#) and the instance port to the inbound rule of the security group.
For details about how to configure security group rules, see [Configuring Security Group Rules](#).
 - d. Run the **ping** command to ping the EIP bound in [1.a](#).
2. You have installed a database client to connect to DB instances.
For details, see [How Can I Install the PostgreSQL Client?](#)

Non-SSL Connection

- Step 1** Log in to the ECS or the device that can access RDS.
- Step 2** Run the following command to connect to the DB instance:

```
psql --no-readline -U<user>-h<host>-p<port>-d<datastore>-W
```

Table 3-10 Parameter description

Parameter	Description
<user>	Indicates the username of the RDS database account. The default administrator is root .

Parameter	Description
<code><host></code>	Indicates the IP address of the primary DB instance. To obtain this parameter, go to the Basic Information page of the DB instance. The IP address can be found on the EIPs page.
<code><port></code>	Indicates the database port in use. The default value is 5432 . To obtain this parameter, go to the Basic Information page of the DB instance. The port number can be found in the Database Port field in the Connection Information area.
<code><datastore></code>	Indicates the name of the database (the default database name is postgres).

The parameter **-W** indicates that a password must be entered for the connection. After running this command, you will be prompted to enter a password.

Example:

Run the following command as user **root** to connect to a DB instance:

```
psql --no-readline -U root -h 192.168.0.44 -p 5432 -d postgres -W
```

----End

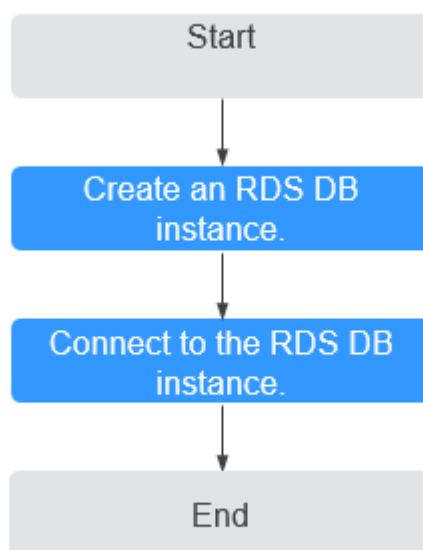
4 Getting Started with RDS for SQL Server

4.1 Operation Guide

You can create and connect to DB instances on the RDS console.

Flowchart

Figure 4-1 Flowchart



Procedure

Table 4-1 Related operations and references

Operation	Reference
Creating an RDS DB instance	Step 1: Create a DB Instance
Connecting to an RDS DB Instance	Step 2: Connect to a DB Instance

4.2 Step 1: Create a DB Instance

Scenarios

This section describes how to create a DB instance on the RDS console.

The DB instance class and storage space you need depend on your processing power and memory requirements.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click **Create DB Instance**.
- Step 5** On the displayed page, configure information about your DB instance. Then, click **Create Now**.

Table 4-2 Basic information

Parameter	Description
Region	The region where your RDS resources will be located. You can change it on the creation page, or go back to the Instance Management page and change it in the upper left corner. NOTE Products in different regions cannot communicate with each other through a private network and you cannot change the region of a DB instance after creating the instance. Therefore, exercise caution when selecting a region.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
DB Engine	Set to Microsoft SQL Server .
DB Engine Version	For details, see DB Engines and Versions. Different DB engine versions are supported in different regions. You are advised to select the latest available version because it is more stable, reliable, and secure.
DB Instance Type	Primary/Standby: uses an HA architecture with a primary DB instance and a synchronous standby DB instance. It is suitable for production databases of large- and medium-sized enterprises in Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other sectors. The standby DB instance improves instance reliability and is invisible to you after being created. An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network. You can deploy primary and standby DB instances in a single AZ or across AZs to achieve failover and high availability. Single: uses a single-node architecture, which is more cost-effective than primary/standby DB instances. It is suitable for development and testing of microsites, and small- and medium-sized enterprises, or for learning about RDS.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. High I/O: supports a maximum throughput of 150 MB/s. Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Time Zone	Select your time zone when you are creating a DB instance. After the DB instance is created, the time zone cannot be modified.
Server Collation	Defines a collation of a database or table column, or a collation cast operation when applied to character string expression. It acts as the default collation for the DB instance.

Table 4-3 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and different maximum IOPS. For details about instance classes, see DB Instance Classes. After a DB instance is created, you can change its CPU and memory. For details, see Changing a DB Instance Class.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • High I/O: supports a maximum throughput of 150 MB/s. • Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Storage Space (GB)	Contains the file system overhead required for inode, reserved block, and database operation. Storage space can range in size from 40 GB to 4,000 GB and can be scaled up only by a multiple of 10 GB. After a DB instance is created, you can scale up its storage space. For details, see section Scaling up Storage Space.

Table 4-4 Network

Parameter	Description
VPC	A dedicated virtual network in which your RDS DB instances are located. A VPC can isolate networks for different services. You can select an existing VPC or create a VPC. For details on how to create a VPC, see the "Creating a VPC" section in the <i>Virtual Private Cloud User Guide</i>. If no VPC is available, RDS allocates a VPC to you by default. NOTICE After the DB instance is created, the VPC cannot be changed.
Subnet	Improves network security by providing dedicated network resources that are logically isolated from other networks. Subnets take effect only within an AZ. The Dynamic Host Configuration Protocol (DHCP) function is enabled by default for subnets in which you plan to create RDS DB instances and cannot be disabled.

Parameter	Description
Security Group	Enhances security by controlling access to RDS from other services. You need to add inbound rules to a security group so that you can connect to your DB instance. If no security group is available, RDS allocates a security group to you by default.

Table 4-5 Database configuration

Parameter	Description
Administrator	The default login name for the database is rdsuser .
Administrator Password	Must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#\$%^*_+?,.). Enter a strong password and periodically change it for security reasons. Keep this password secure. The system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see section Resetting the Administrator Password .
Confirm Password	Must be the same as Administrator Password .
Parameter Template	Contains engine configuration values that can be applied to one or more DB instances. If you intend to create primary/standby DB instances, they use the same parameter template. NOTICE If you use a custom parameter template when creating a DB instance, the specification-related parameter max server memory (MB) in the custom template is not delivered. Instead, the default value is used. You can modify the instance parameters as required after the DB instance is created. For details, see Modifying Parameters .
Enterprise Project	If the DB instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the ProjectMan console to create a project. For details about how to create a project, see the <i>ProjectMan User Guide</i> .

Table 4-6 Batch creation

Parameter	Description
Quantity	RDS supports DB instance creation in batches. If you choose to create primary/standby DB instances and set Quantity to 1, a primary DB instance and a standby DB instance will be created synchronously. If you create multiple DB instances at a time, they will be named with four digits appended to the DB instance name. For example, if you enter instance, the first instance will be named instance-0001, the second instance-0002, and so on.

 NOTE

The performance of your DB instance depends on its configurations. Hardware configuration items include the instance specifications, storage type, and storage space.

Step 6 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 7 To view and manage the DB instance, go to the **Instance Management** page.

- During the creation process, the DB instance status is **Creating**. When the creation process is complete, the instance status will change to **Available**.
- The automated backup policy is enabled by default. An automated full backup is immediately triggered after a DB instance is created.
- The default database port number is **1433**. After a DB instance is created, you can change its port number.

For details, see [Changing a Database Port](#).

----End

4.3 Step 2: Connect to a DB Instance

4.3.1 Connecting to a DB Instance

An RDS DB instance can be connected through a private network or a public network.

Table 4-7 RDS connection methods

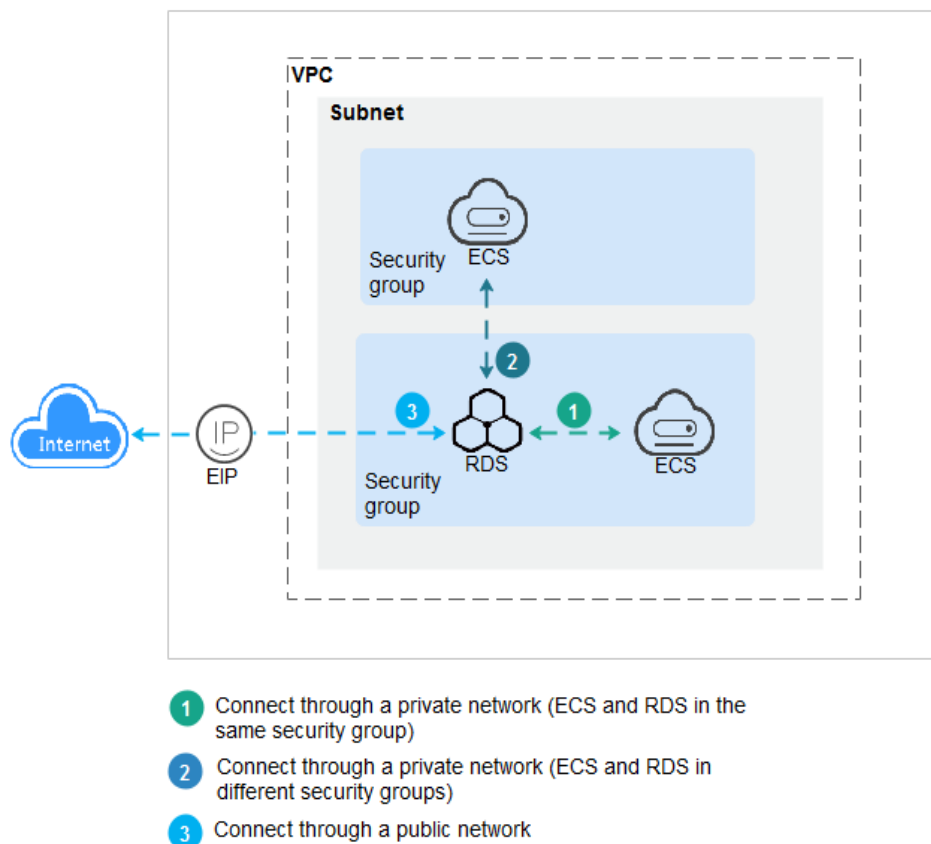
Connect Through	IP Address	Scenarios	Description
Private network	Floating IP	RDS provides a floating IP address by default. When your applications are deployed on an ECS that is in the same region and VPC as RDS, you are advised to use a floating IP address to connect to the RDS DB instance through the ECS.	- Secure and excellent performance - Recommended
Public network	EIP	If you cannot access an RDS DB instance through a floating IP address, bind an EIP to the DB instance and connect the DB instance to the ECS through the EIP.	- A relatively lower level of security compared to other connection methods - To achieve a higher transmission rate and security level, you are advised to migrate your applications to an ECS that is in the same VPC as your RDS DB instance and use a floating IP address to access the DB instance.

 NOTE

- VPC: indicates the Virtual Private Cloud.
- ECS: indicates the Elastic Cloud Server.
- If the ECS is in the same VPC as the RDS DB instance, you do not need to apply for an EIP.

Figure 4-2 illustrates the connection over a private network or a public network.

Figure 4-2 DB instance connection



4.3.2 Connecting to a DB Instance Through a Private Network

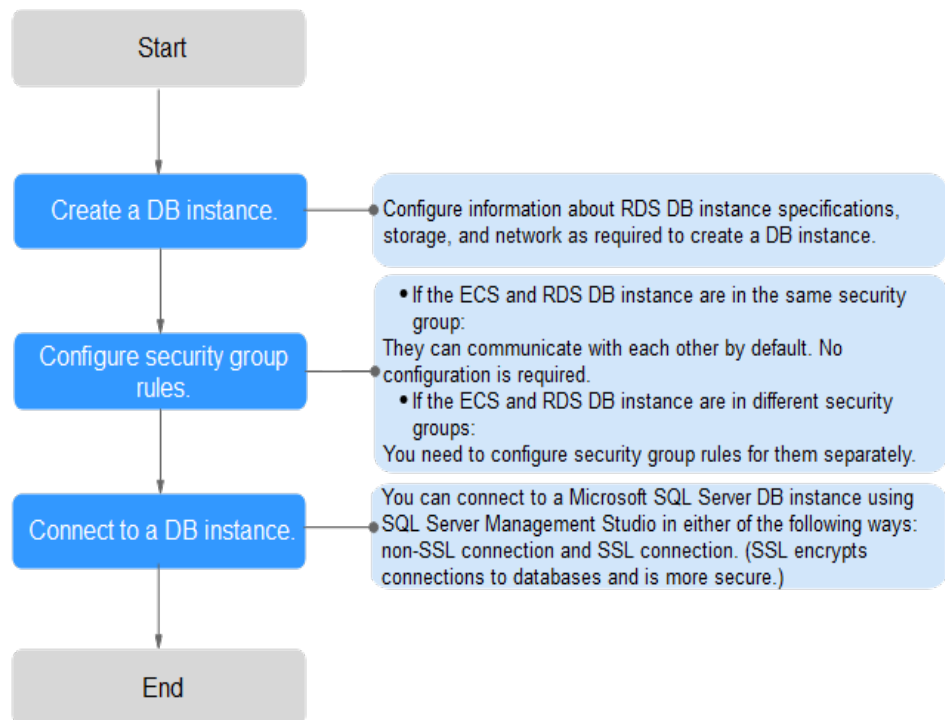
4.3.2.1 Connecting to a DB Instance Through a Private Network

If you are using RDS for the first time, see the constraints described in section [Microsoft SQL Server Constraints](#).

Process

Figure 4-3 illustrates the process of connecting to a SQL Server DB instance through a private network.

Figure 4-3 Connecting to a DB instance through a private network



4.3.2.2 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

First check whether the ECS and RDS DB instance are in the same security group.

- If the ECS and RDS DB instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured. Go to [Connecting to a DB Instance Through a Private Network](#).
- If the ECS and RDS DB instance are in different security groups, you need to configure security group rules for them, separately.
 - RDS DB instance: Configure an **inbound rule** for the security group with which the RDS DB instance is associated.
 - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security rule for the ECS. If not all outbound traffic is allowed in the security group, you need to configure an **outbound rule** for the ECS.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group.

After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, a tenant can create a maximum of 100 security groups.
- By default, each security group can have a maximum of 50 security group rules.
- One security group can be associated with only one RDS DB instance.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

 NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

- Step 1** Log in to the management console.
- Step 2** Under **Network**, click **Virtual Private Cloud**.
- Step 3** In the navigation pane on the left, choose **Access Control > Security Groups**.
- Step 4** On the **Security Groups** page, locate the target security group and click **Manage Rule** in the **Operation** column.
- Step 5** On the displayed page, click **Add Rule**.
- Step 6** In the displayed dialog box, set required parameters to add an inbound rule.
- Step 7** Click **OK**.
- End

4.3.2.3 Connecting to a DB Instance Through a Private Network

You can connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection encrypts data and is more secure.

Preparations

1. Prepare an ECS.
To connect to a DB instance through a private network, you must first create an ECS.
For details about how to create an ECS, see section [How Can I Create and Connect to an ECS?](#)
 - The ECS and RDS DB instance must be in the same VPC.
 - The ECS must be allowed by the security group to access RDS DB instances.

- If the security group with which the target DB instance is associated is the default security group, you do not need to configure security group rules.
- If the security group with which the target DB instance is associated is not the default security group, check whether the security group rules allow the ECS to connect to the DB instance.
 - 1) Log in to the management console.
 - 2) Click  in the upper left corner and select a region and a project.
 - 3) Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
 - 4) On the **Instance Management** page, click the target DB instance.
 - 5) In the **Connection Information** area, click the security group to view its rules.

If the security group rules allow the access from the ECS, the ECS can connect to the DB instance.

If the security group rules do not allow the access from the ECS, you need to add a security group rule. For details, see section [Configuring Security Group Rules](#).

2. Install the Microsoft SQL Server client.

Install the Microsoft SQL Server client on the ECS or device that was prepared in [1](#).

Non-SSL Connection

- Step 1** Log in to the ECS or device that can access RDS.
- Step 2** Start SQL Server Management Studio.
- Step 3** Choose **Connect > Database Engine**. In the displayed dialog box, enter login information.

Figure 4-4 Connecting to the server



- **Server name:** indicates the IP address and port of the DB instance. Use a comma (,) to separate them. For example: x.x.x.x,8080.
 - The IP address is the floating IP address in the **Connection Information** area on the **Basic Information** page of the DB instance.
 - The port is the database port in the **Connection Information** area on the **Basic Information** page of the DB instance.
- **Authentication:** indicates the authentication mode. Select **SQL Server Authentication**.
- **Login:** indicates the RDS database username. The default administrator is **rdsuser**.
- **Password:** indicates the password of the RDS database username.

Step 4 Click **Connect** to connect to the DB instance.

NOTE

If the connection fails, ensure that preparations have been correctly made in [Preparations](#) and try again.

----End

SSL Connection

Step 1 Download the SSL root certificate and then upload it.

1. In the **DB Information** area on the **Basic Information** page, click  in the **SSL** field to download the root certificate or certificate bundle.
2. Upload the root certificate to the ECS or save it to the device to be connected to the DB instance.
3. Import the root certificate into the Windows OS on the ECS. For details, see [How Can I Import the Root Certificate to a Windows or Linux OS?](#)

 NOTE

- Replace the old certificate before it expires to improve system security.
- After you bind an EIP to a DB instance, you must reboot the instance for the SSL connection to take effect.

Step 2 Start SQL Server Management Studio.

Step 3 Choose **Connect > Database Engine**. In the displayed dialog box, enter login information.

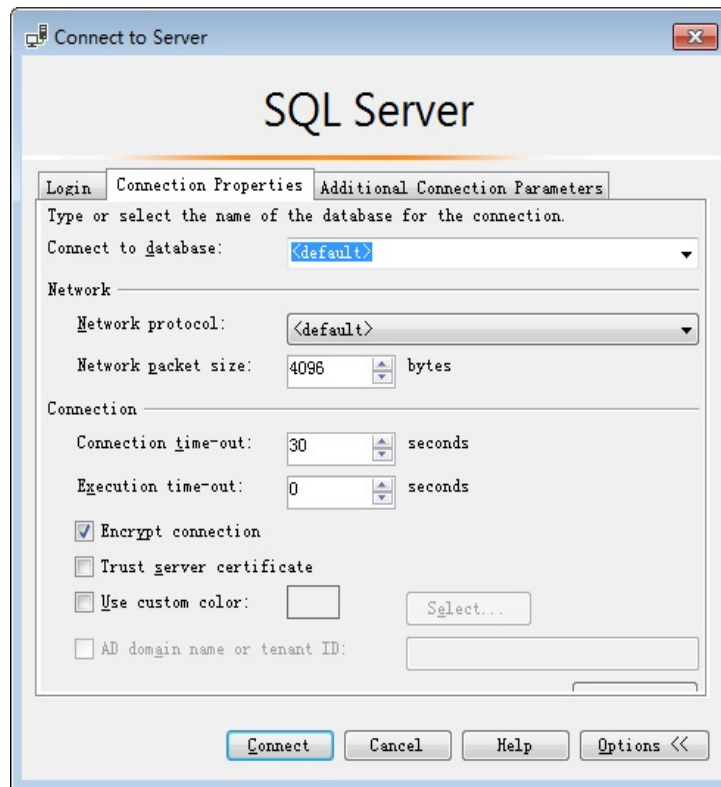
Figure 4-5 Connecting to the server



- **Server name:** indicates the IP address and port of the DB instance. Use a comma (,) to separate them. For example: x.x.x.x,8080.
 - The IP address is the floating IP address in the **Connection Information** area on the **Basic Information** page of the DB instance.
 - The port is the database port in the **Connection Information** area on the **Basic Information** page of the DB instance.
- **Authentication:** indicates the authentication mode. Select **SQL Server Authentication**.
- **Login:** indicates the RDS database username. The default administrator is **rdsuser**.
- **Password:** indicates the password of the RDS database username.

Step 4 On the **Connection Properties** page, enter related parameters and select **Encrypt connection** to enable SSL encryption. (By default, **Encrypt connection** is not selected. You need to select it manually.)

Figure 4-6 Connection properties



Step 5 Click **Connect** to connect to the DB instance.

NOTE

If the connection fails, ensure that preparations have been correctly made in [Preparations](#) and try again.

----End

4.3.3 Connecting to a DB Instance Through a Public Network

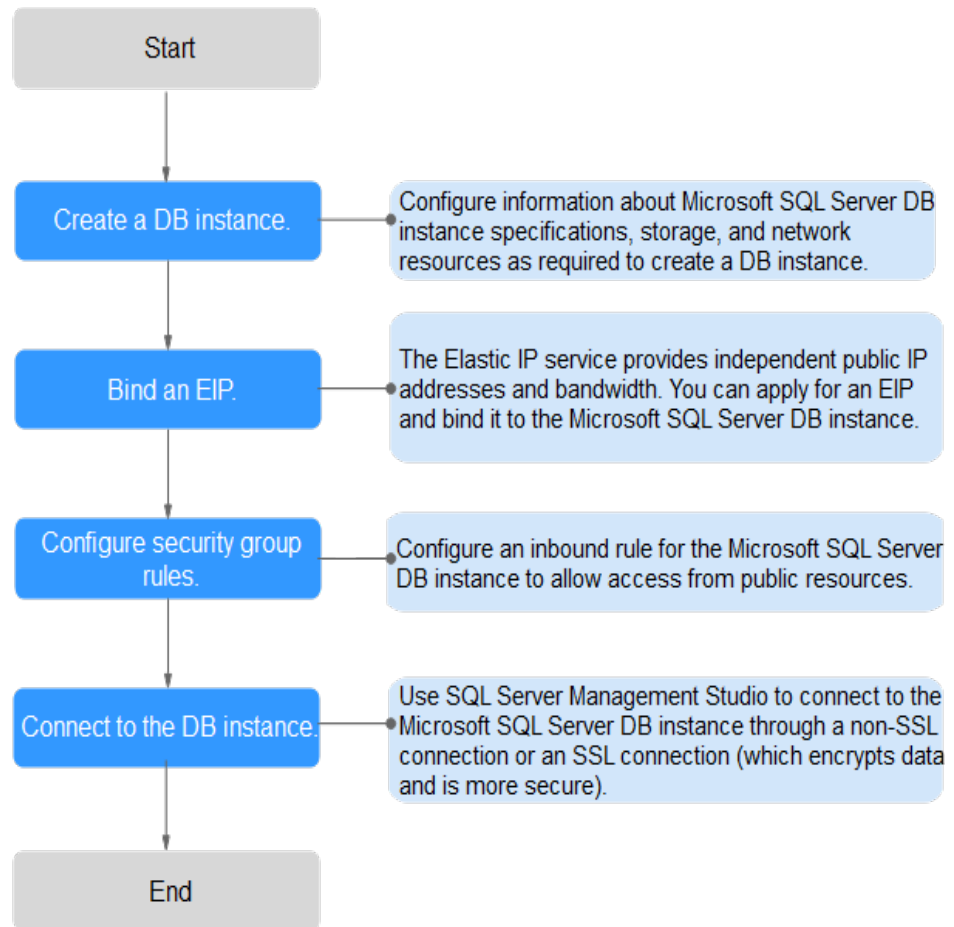
4.3.3.1 Connecting to a DB Instance Through a Public Network

If you are using RDS for the first time, see the constraints described in section [Microsoft SQL Server Constraints](#).

Process

[Figure 4-7](#) illustrates the process of connecting to a SQL Server DB instance through a public network.

Figure 4-7 Connecting to a DB instance through a public network



4.3.3.2 Binding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility and can unbind the EIP from the DB instance as required.

Precautions

- You need to configure security groups and enable specific IP addresses and ports to access the target DB instance. Before accessing the DB instance, you need to add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see section [Configuring Security Group Rules](#).

Binding an EIP

Step 1 On the **Instance Management** page, click the target DB instance.

Step 2 In the navigation pane on the left, choose **EIPs**. On the displayed page, click **Bind EIP**.

Step 3 In the displayed dialog box, select an EIP and click **OK**.

If no available EIPs are displayed, click **View EIP** and obtain an EIP.

Step 4 On the **EIPs** page of the RDS console, view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

----End

4.3.3.3 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

When you attempt to connect to an RDS DB instance through an EIP, you need to configure an **inbound rule** for the security group associated with the DB instance.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group. After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, you can create a maximum of 500 security group rules.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Network**, click **Virtual Private Cloud**.

Step 3 In the navigation pane on the left, choose **Access Control > Security Groups**.

Step 4 On the **Security Groups** page, locate the target security group and click **Manage Rule** in the **Operation** column.

Step 5 On the displayed page, click **Add Rule**.

Step 6 In the displayed dialog box, set required parameters to add an inbound rule.

Step 7 Click **OK**.

----End

4.3.3.4 Connecting to a DB Instance Through a Public Network

You can connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection encrypts data and is more secure.

Preparations

1. Install the SQL Server client.
For details, see section [How Can I Install SQL Server Management Studio?](#)
2. Bind an EIP to the target DB instance and configure security group rules.
 - a. Bind an EIP to the target DB instance.
For details about how to bind an EIP, see section [Binding an EIP](#).
 - b. Obtain the IP address of the local device.
 - c. Configure security group rules.
Add the IP address obtained in [2.b](#) and the instance port to the inbound rule of the security group.
For details about how to configure a security group rule, see section [Configuring Security Group Rules](#).
 - d. Run the **ping** command to connect the EIP that has been bound to the target DB instance in [2.a](#) to check that the local device can connect to the EIP.

Non-SSL Connection

Step 1 Start SQL Server Management Studio.

Step 2 Choose **Connect > Database Engine**. In the displayed dialog box, enter login information.

Figure 4-8 Connecting to the server



- **Server name:** indicates the IP address and port of the DB instance. Use a comma (,) to separate them. For example: x.x.x.x,8080.
 - The IP address is the EIP that has been bound to the DB instance.
 - The port is the database port in the **Connection Information** area on the **Basic Information** page of the DB instance.
- **Authentication:** indicates the authentication mode. Select **SQL Server Authentication**.
- **Login:** indicates the RDS database username. The default administrator is **rdsuser**.
- **Password:** indicates the password of the RDS database username.

Step 3 Click **Connect** to connect to the DB instance.

 NOTE

If the connection fails, ensure that preparations have been correctly made in [Preparations](#) and try again.

----End

SSL Connection

Step 1 Download the SSL root certificate and then upload it.

1. In the **DB Information** area on the **Basic Information** page, click  in the **SSL** field to download the root certificate or certificate bundle.
2. Upload the root certificate to the ECS to be connected to the DB instance.
3. Import the root certificate to the Windows OS on the ECS. For details, see [How Can I Import the Root Certificate to a Windows or Linux OS?](#)

 NOTE

- Replace the old certificate before it expires to improve system security.
- After you bind an EIP to a DB instance, you must reboot the instance for the SSL connection to take effect.

Step 2 Start SQL Server Management Studio.

Step 3 Choose **Connect > Database Engine**. In the displayed dialog box, enter login information.

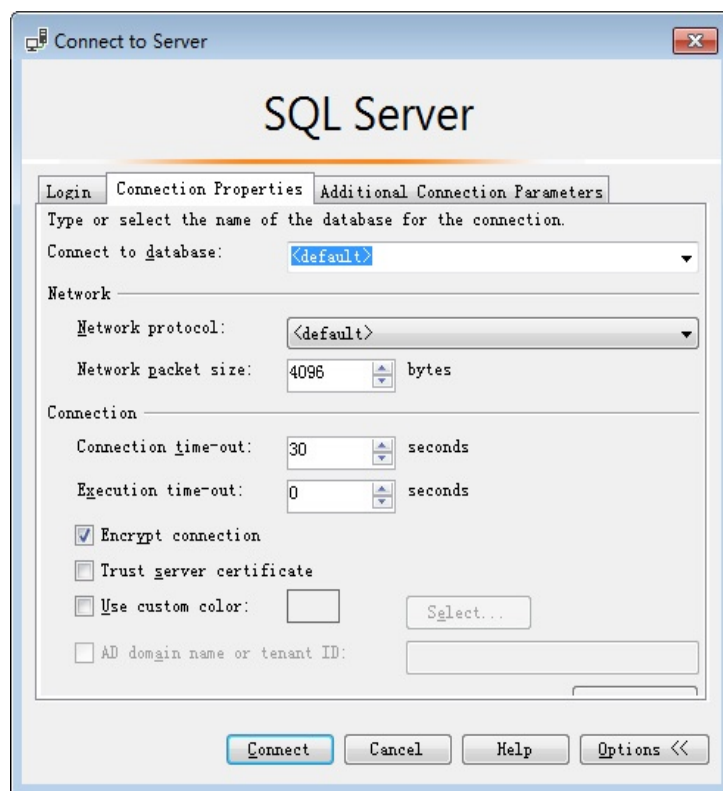
Figure 4-9 Connecting to the server



- **Server name:** indicates the IP address and port of the DB instance. Use a comma (,) to separate them. For example: x.x.x.x,8080.
 - The IP address is the EIP that has been bound to the DB instance.
 - The port is the database port in the **Connection Information** area on the **Basic Information** page of the DB instance.
- **Authentication:** indicates the authentication mode. Select **SQL Server Authentication**.
- **Login:** indicates the RDS database username. The default administrator is **rdsuser**.
- **Password:** indicates the password of the RDS database username.

Step 4 On the **Connection Properties** page, enter related parameters and select **Encrypt connection** to enable SSL encryption. (By default, **Encrypt connection** is not selected. You need to select it manually.)

Figure 4-10 Connection properties



Step 5 Click **Connect** to connect to the DB instance.

NOTE

If the connection fails, ensure that preparations have been correctly made in [Preparations](#) and try again.

----End

5 Working with RDS for MySQL

5.1 Data Migration

5.1.1 Migrating Data to RDS for MySQL Using mysqldump

Preparing for Data Migration

You can access RDS DB instances through an EIP or through an ECS.

1. Prepare an ECS for accessing DB instances in the same VPC or prepare a device for accessing RDS through an EIP.
 - To connect to a DB instance through an ECS, you need to create an ECS first.
 - To connect to a DB instance through an EIP, you must:
 - i. Bind an EIP to a DB instance. For details, see [Binding an EIP](#).
 - ii. Ensure that the local device can access the EIP.
2. Install the MySQL client on the prepared ECS or device.

NOTE

The MySQL client version must be the same as the version of RDS for MySQL. The MySQL database or client will provide mysqldump and mysql.

After data is migrated to RDS, you may need to change the IP address. For details, see [Configuring and Changing a Floating IP Address](#).

Exporting Data

Before migrating data to RDS, you need to export data first.

NOTICE

- The export tool must match the DB engine version.
 - Database migration is performed offline. Before the migration, you must stop any applications using the source database.
-

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Use the mysqldump tool to export metadata into an SQL file.

NOTICE

The MySQL database is required for RDS management. When exporting metadata, do not specify **--all-database**. Otherwise, the MySQL database will be unavailable.

```
mysqldump --databases <DB_NAME> --single-transaction --order-by-primary  
--hex-blob --no-data --routines --events --set-gtid-purged=OFF -u <DB_USER>  
-p -h <DB_ADDRESS> -P <DB_PORT> |sed -e 's/DEFINER[ ]*=[ ]*[^\]*\*/' -e 's/  
DEFINER[ ]*.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*.*PROCEDURE/  
PROCEDURE/' -e 's/DEFINER[ ]*.*TRIGGER/TRIGGER/' -e 's/  
DEFINER[ ]*.*EVENT/EVENT/' > <BACKUP_FILE>
```

- *DB_NAME* indicates the name of the database to be migrated.
- *DB_USER* indicates the database username.
- *DB_ADDRESS* indicates the database address.
- *DB_PORT* indicates the database port.
- *BACKUP_FILE* indicates the name of the file to which the data will be exported.

Enter the database password when prompted.

Example:

```
mysqldump --databases rdsdb --single-transaction --order-by-primary --hex-  
blob --no-data --routines --events --set-gtid-purged=OFF -u root -p -h  
192.168.151.18 -P 3306 |sed -e 's/DEFINER[ ]*=[ ]*[^\]*\*/' -e 's/  
DEFINER[ ]*.*FUNCTION/FUNCTION/' -e 's/DEFINER[ ]*.*PROCEDURE/  
PROCEDURE/' -e 's/DEFINER[ ]*.*TRIGGER/TRIGGER/' -e 's/  
DEFINER[ ]*.*EVENT/EVENT/' > dump-defs.sql
```

Enter password:

 NOTE

If you use mysqldump with a version earlier than 5.6, remove **--set-gtid-purged=OFF** before running this command.

After this command is executed, a **dump-defs.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll dump-defs.sql  
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 dump-defs.sql
```

Step 3 Use the mysqldump tool to export data into an SQL file.

NOTICE

The MySQL database is required for RDS management. When exporting metadata, do not specify **--all-database**. Otherwise, the MySQL database will be unavailable.

```
mysqldump --databases <DB_NAME> --single-transaction --hex-blob --set-
gtid-purged=OFF --no-create-info --skip-triggers -u <DB_USER> -p -h
<DB_ADDRESS> -P <DB_PORT> -r <BACKUP_FILE>
```

For details on the parameters in the preceding command, see [Step 2](#).

Enter the database password when prompted.

Example:

```
mysqldump --databases rdsdb --single-transaction --hex-blob --set-gtid-
purged=OFF --no-create-info --skip-triggers -u root -p -h 192.168.151.18 -P
8635 -r dump-data.sql
```

NOTE

If you use mysqldump with a version earlier than 5.6, remove **--set-gtid-purged=OFF** before running this command.

After this command is executed, a **dump-data.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll dump-data.sql
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 dump-data.sql
```

----End

Importing Data

You can connect your client to RDS and import exported SQL files into RDS.

NOTICE

If the source database calls triggers, stored procedures, functions, or events, you must set **log_bin_trust_function_creators** to **ON** on the destination database before importing data.

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Import metadata into RDS.

```
# mysql -f -h <RDS_ADDRESS> -P <DB_PORT> -u root -p <<BACKUP_DIR>/
dump-defs.sql
```

- *RDS_ADDRESS*: indicates the IP address of the RDS DB instance.
- *DB_PORT*: indicates the RDS DB instance port.
- *BACKUP_DIR*: indicates the directory where **dump-defs.sql** is stored.

Example:

```
# mysql -f -h 172.16.66.198 -P 3306 -u root -p < dump-defs.sql
```

Enter password:

NOTE

If you intend to import SQL statements of a table to RDS, you are advised to specify a database. Otherwise, the error message "No database selected" may be displayed. For example, if you intend to import SQL statements of a table to database **mydb**, run the following command:

```
# mysql -f -h 172.16.66.198 -P 3306 -u root -p mydb < dump-defs.sql
```

Enter password:

Step 3 Import data into RDS.

```
# mysql -f -h <RDS_ADDRESS> -P <DB_PORT> -u root -p < <BACKUP_DIR>/  
dump-data.sql
```

- *RDS_ADDRESS*: indicates the IP address of the RDS DB instance.
- *DB_PORT*: indicates the RDS DB instance port.
- *BACKUP_DIR*: indicates the directory where **dump-data.sql** is stored.

Example:

```
# mysql -f -h 172.16.66.198 -P 3306 -u root -p < dump-data.sql
```

Enter password:

NOTE

If you intend to import SQL statements of a table to RDS, you are advised to specify a database. Otherwise, the error message "No database selected" may be displayed. For example, if you intend to import SQL statements of a table to database **mydb**, run the following command:

```
# mysql -f -h 172.16.66.198 -P 3306 -u root -p mydb < dump-defs.sql
```

Enter password:

Step 4 View the import result.

```
mysql> show databases;
```

The following result indicates that database **rdsdb** has been imported.

```
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| rdsdb |  
| mysql |  
| performance_schema |  
+-----+  
4 rows in set (0.00 sec)
```

----End

5.2 Parameter Tuning

5.2.1 Suggestions on MySQL Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect the stable running of databases. This section

describes some important parameters for your reference. For details, visit the [MySQL official website](#).

For details on how to modify MySQL parameters on the console, see [Modifying Parameters](#).

Sensitive Parameters

The following parameters can result in system security and stability issues if set improperly:

- **lower_case_table_names**

Default value: **1**

Function: Controls whether database and tables stored on disks are case sensitive. The value **1** indicates that database and table names are case-insensitive and are lowercase by default.

 NOTE

MySQL 8.0 does not support this parameter.

Impact: Changing this parameter value may cause primary/standby replication exceptions. Exercise caution when performing this operation.

- If you want to change this parameter value from **1** to **0**, change it on read replicas and reboot them first, and then repeat the operations on the primary DB instance.
- If you want to change this parameter value from **0** to **1**, change it on the primary DB instance and reboot it first, and then run **SELECT @@GLOBAL.GTID_EXECUTED** on read replicas. Wait until the result set is at least the same as the primary DB instance and then change this parameter value on read replicas and reboot them.

- **innodb_flush_log_at_trx_commit**

Default value: **1**

Function: Controls the balance between strict ACID compliance for commit operations and higher performance. The default setting of **1** is required for full ACID compliance. Logs are written and flushed to disks at each transaction commit. If the value is set to **0**, logs are written and flushed to disks once per second. If the value is set to **2**, logs are written at each transaction commit and flushed to disks every two seconds.

Impact: If this parameter is not set to **1**, data security is not guaranteed. If the system fails, data may be lost.

- **sync_binlog**

Default value: **1**

Function: Controls how often the MySQL server synchronizes binary logs to the disk. The default setting of **1** requires synchronization of the binary log to the disk at each transaction commit. If the value is set to **0**, synchronization of the binary log to the disk is not controlled by the MySQL server but relies on the OS to flush the binary log to the disk. This setting provides the best performance, but in the event of a power failure or OS crash, all binary log information in **binlog_cache** is lost.

Impact: If this parameter is not set to **1**, data security is not guaranteed. If the system fails, data may be lost.

- **innodb_large_prefix**

Default value: **OFF**

Function: Specifies the maximum length of a single-column index in an InnoDB table.

 NOTE

This parameter is available only for MySQL 5.6.

Impact: Changing this parameter value during DDL execution may cause primary/standby replication exceptions. Exercise caution when performing this operation.

- If you want to change this parameter value from **OFF** to **ON**, change it on read replicas first and then on the primary DB instance.
- If you want to change this parameter value from **ON** to **OFF**, change it on the primary DB instance first and then on read replicas.

Performance Parameters

The following parameters can affect database performance:

- The values of **innodb_spin_wait_delay** and **query_alloc_block_size** are determined by the DB instance specifications. If you increase their values, database performance may be affected.
- If **key_buffer_size** is set to a value smaller than **4096**, the parameter modification will fail.
- If **max_connections** is set to a small value, database access will be affected.
- The default values of the following parameters are determined by the DB instance specifications: **innodb_buffer_pool_size**, **max_connections**, and **back_log**. These parameter values are **default** before being specified.
- The values of **innodb_io_capacity_max** and **innodb_io_capacity** are determined by the storage type. These parameter values are **default** before being specified.

5.3 MySQL Kernel Functions

5.3.1 Kernel Version Description

This section describes the kernel version updates of RDS for MySQL.

RDS for MySQL 8.0

Table 5-1 RDS for MySQL 8.0 version description

Date	Description
2022-12-09	• New features and performance optimized – Connection per thread was supported for killing sessions. – Constraints on memory were added for the Performance Schema. – The performance of SQL Explorer was optimized based on specific scenarios. – Database performance was optimized in specific scenarios when the value of internal_tmp_mem_storage_engine was set to MEMORY. – The compiler was upgraded to GCC 10.3. • Issues resolved – Errors caused by writes to a temporary file were resolved. – Unexpected responses to Common Table Expression (CTE) queries were resolved. • Security hardening – The following security vulnerabilities were fixed: CVE-2021-2417, CVE-2021-2339, CVE-2021-2425, CVE-2021-2426, CVE-2021-2427, CVE-2021-2424, CVE-2021-2383, CVE-2021-2384, and CVE-2021-2410.
2022-09-01	• New features and performance optimized – The kernel version was upgraded to 8.0.25. – SQL statement concurrency control was supported. – The compiler was upgraded to GCC 9.3. • Issues resolved – Replication interruption caused by a single oversized binlog was resolved. – Inaccurate innodb_row_lock_current_waits statistics were fixed. – Unexpected restarts due to the use of BLOB fields were resolved. • Security hardening – The following security vulnerabilities were fixed: CVE-2021-2307, CVE-2021-2180, and CVE-2021-2194.

Date	Description
2021-11-07	- New features and performance optimized - The static link library for the thread pool improved the performance. - Profile-Guided Optimization (PGO) was enabled. - The MySQL hash algorithm was optimized. - Remarks can be added to a database. - Protection from being modified by DDL was provided for the system database. - The innodb_total_tablespace parameter was added to calculate the number of InnoDB tablespaces. - The InnoDB lock view was provided in information_schema. - The OpenSSL, jemalloc, and curl open-source components were upgraded. - Issues resolved - The issue that XA transactions may restart abnormally after binlog rotation was resolved. - The issue that precompiled SQL statement types were not recorded by SQL Explorer was resolved. - Incorrect execution time statistics for FLUSH PRIVILEGES was resolved. - The issue that audit logs were incorrectly written into other files was resolved.
2021-07-13	- The issue that XA transactions may be lost after unexpected database restart was resolved. - The adaptive hash segmentation algorithm was optimized. - The kernel version was upgraded to 8.0.21. - Security hardening The following security vulnerabilities were fixed: CVE-2020-14697, CVE-2020-14680, CVE-2020-14678, CVE-2020-14663, CVE-2021-2020, CVE-2020-14619, CVE-2020-14591, CVE-2020-14576, and CVE-2020-14539.
2021-04-26	- SQL statement concurrency control was optimized. - Full SQL collection was optimized.
2021-03-31	Performance optimized The compiler was upgraded to GCC 9.
2021-03-01	- Performance optimized The efficiency of collecting additional information about slow query logs was improved. - Issues resolved The issue that replication on the standby DB instance might be interrupted during the rollback of distributed transactions was resolved.

Date	Description
2021-02-06	- Issues resolved - Timing error caused by gettimeofday in multiple threads was fixed.
2020-12-21	- Disconnection details were recorded in error logs. - Index hints were supported.
2020-11-03	- The execution time and waiting time of large transactions were displayed. - Independent connection control was used to manage users. - SQL filter was used to restrict the execution frequency of SQL statements during peak hours. - Kernel performance was optimized.

RDS for MySQL 5.7

Table 5-2 RDS for MySQL 5.7 version description

Date	Description
2022-12-09	- New features and performance optimized - The kernel version was upgraded to 5.7.38 through backporting. - The compiler was upgraded to GCC 10.3. - Connection per thread was supported for killing sessions. - The threshold for slow query logs can be set based on the lock wait duration. - ALT security was hardened. - Issues resolved - Recovery security of crashed XA transactions on the primary instance was enhanced. - Abnormal instance reboot when Database Proxy is enabled was resolved. - Abnormal reboot caused by failed memory request for plugins was resolved. - Security hardening - The following security vulnerabilities were fixed: CVE-2022-21454, CVE-2022-21417, CVE-2022-21427, CVE-2022-21451, CVE-2022-21444, and CVE-2022-21460.

Date	Description
2022-09-01	• New features and performance optimized – The kernel version was upgraded to 5.7.37. – The compiler was upgraded to GCC 9.3. – The OpenSSL and curl open-source components were upgraded. • Issues resolved – Replication interruption caused by a single oversized binlog was resolved. – Unexpected restarts caused by concurrent playback grants to the slave node were resolved. – Replication interruptions caused by hidden auto-increment keys were resolved. – Unexpected restarts during rollback of tables with virtual columns were resolved. – Unexpected restarts during recovery of encrypted tables were resolved. – Inaccurate Seconds Behind Master values in specific scenarios were resolved. • Security hardening – The following security vulnerabilities were fixed: CVE-2022-21367, CVE-2022-21304, and CVE-2022-21344.
2022-04-26	• New features and performance optimized – Hidden auto-increment keys were supported. • Issues resolved – Replication exceptions caused by repeated commitment of XA transactions were resolved. – Inaccurate innodb_row_lock_current_waits statistics were fixed.
2022-02-26	• New features and performance optimized – Restrictions on the length of a single record for SQL Explorer were removed. – Application Lossless and Transparent (ALT) Phase I was supported for RDS for MySQL. • Issues resolved – Memory problems of the thread pool in extreme scenarios were resolved. – Occasional suspension of XA transaction playbacks on a standby node was resolved.

Date	Description
2021-11-07	• New features and performance optimized – The static link library for the thread pool improved the performance. – Profile-Guided Optimization (PGO) was enabled. – The MySQL hash algorithm was optimized. – Remarks can be added to a database. – Protection from being modified by DDL was provided for the system database. – I/O latency information can be queried in error logs. – Minidumps were supported. – The kernel version was upgraded to 5.7.33. – The OpenSSL, jemalloc, and curl open-source components were upgraded. • Issues resolved – The issue that the replication on the standby node may be interrupted due to playback order preserving was resolved. – The issue that XA transactions may restart abnormally after binlog rotation was resolved. – The issue that precompiled SQL statement types may not be recorded by SQL Explorer was resolved. – Incorrect execution time statistics for FLUSH PRIVILEGES was resolved. – The issue that audit logs were incorrectly written into other files was resolved. • Security hardening The following security vulnerabilities were fixed: CVE-2021-2011, CVE-2021-2178, and CVE-2021-2202.
2021-07-13	• The issue that XA transactions may be lost after unexpected database restart was resolved. • The adaptive hash segmentation algorithm was optimized. • The kernel version was upgraded to 5.7.32.
2021-04-26	• New features The real client address can be displayed when proxy is used. • Issues resolved Full SQL collection was optimized. The issue that revoking permissions might cause permission inconsistency between primary and standby DB instances was resolved. MySQL 8.0 optimizations on Instant Add Column was incorporated.

Date	Description
2021-03-31	- Performance optimized - The efficiency of collecting additional information about slow query logs was improved. - The compiler was upgraded to GCC 9. - Issues resolved The issue that replication on the standby DB instance might be interrupted during the rollback of distributed transactions was resolved.
2021-03-01	The conflict frequency between fil_sys mutual exclusions was reduced.
2021-02-06	- New features - The compiler was optimized. - The UTF-8 encoding efficiency was improved on non-Arm platforms. - Issues resolved Timing error caused by gettimeofday in multiple threads was fixed.
2020-12-21	- The kernel version was upgraded to 5.7.31. - SQL filter was used to restrict the execution frequency of SQL statements during peak hours.
2020-11-03	- Kernel performance was optimized. - The recycle bin was supported. - Issues that may occur during local disk data cleanup were resolved.

RDS for MySQL 5.6

Table 5-3 RDS for MySQL 5.6 version description

Date	Description
2022-12-09	- New features and performance optimized - Connection per thread was supported for killing sessions. - Issues resolved - Abnormal instance reboot when Database Proxy is enabled was resolved. - main.proxy_connect buffer overflow issues were resolved. - Abnormal reboot caused by failed memory request for plugins was resolved.
2022-09-01	Inaccurate innodb_row_lock_current_waits statistics were fixed.

Date	Description
2021-11-07	• New features – Remarks can be added to a database. – Protection from being modified by DDL was provided for the system database. – The OpenSSL and jemalloc open-source components were upgraded. • Issues resolved – The issue that the synchronization may be interrupted after password change was resolved. – The issue that audit logs were incorrectly written into other files was resolved.
2021-07-13	• The issue that the replication on the standby node may be interrupted due to playback order preserving was resolved. • The kernel version was upgraded to 5.6.51. • Security hardening The vulnerability patches for MySQL 5.6 Community Edition are no longer released because the patches for this version are not released anymore.
2021-04-26	• New features The real client address can be displayed when proxy is used. • Issues resolved The issue that a syntax error was reported during the select 1 for update execution was resolved. Full SQL collection was optimized.
2021-03-31	SQL filter was reconstructed to improve usability.
2021-02-06	The kernel version was upgraded to 5.6.50.
2020-12-23	SQL filter was used to restrict the execution frequency of SQL statements during peak hours.
2020-11-03	The kernel version was upgraded to 5.6.49.
2020-10-09	• Local disk logs whose size exceeds the local disk capacity were stored on cloud disks. • Users' operation history can be recorded in error logs.

5.4 Instance Lifecycle

5.4.1 Creating a Same DB Instance as an Existing DB Instance

Scenarios

This section describes how to quickly create a DB instance with the same configurations as the selected one.

NOTE

- You can create DB instances with the same configurations numerous times.
- This function is unavailable for read replicas.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Create Same DB Instance** in the **Operation** column.

Step 5 On the displayed page, the configurations are the same as those of the selected DB instance. You can change them as required. Then, click **Create Now**.

For details about MySQL DB instance configurations, see section [Step 1: Create a DB Instance](#).

Step 6 Confirm the instance specifications.

Step 7 Refresh the DB instance list and view the status of the DB instance. If the status is **Available**, it has been created successfully.

You can manage the DB instance on the **Instance Management** page.

----End

5.4.2 Rebooting DB Instances or Read Replicas

Scenarios

You may need to reboot a DB instance during maintenance. For example, after you modify some parameters, a reboot is required for the modifications to take effect. You can reboot a primary DB instance or a read replica on the management console.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- Rebooting DB instances will cause service interruptions. During the reboot process, the DB instance status is **Rebooting**.

- Rebooting DB instances will cause instance unavailability and clear cached memory. To prevent traffic congestion during peak hours, you are advised to reboot DB instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance, or click  and then locate the target read replica. Choose **More > Reboot** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page to go to the **Basic Information** page. In the upper right corner, click **Reboot**.

For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 5 In the displayed dialog box, click **Yes**.

Step 6 Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has rebooted successfully.

----End

5.4.3 Selecting Displayed Items

Scenarios

You can customize which instance items are displayed on the **Instance Management** page.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click  to edit columns displayed in the DB instance list.

- The following items are displayed by default: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, and operation. These default items cannot be deselected.
- In a single project, you can select up to 9 items: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, creation time, database port, storage type, and operation.

- For multiple projects, if you have enabled the ProjectMan permissions, you can select up to 9 items: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, creation time, database port, storage type, and operation.

----End

5.4.4 Exporting DB Instance Information

Scenarios

You can export information about all or selected DB instances to view and analyze DB instance information.

Exporting Information About All DB Instances

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click  in the upper right corner of the page. By default, information about all DB instances are exported. In the displayed dialog box, you can select the items to be exported and click **Export**.

Step 5 Find a .csv file locally after the export task is completed.

----End

Exporting Information About Selected DB Instances

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, filter DB instances by DB engine, DB instance name, DB instance ID, or floating IP address, or select the DB instances to be exported, and click  in the upper right corner of the page. In the displayed dialog box, select the items to be exported and click **Export**.

Step 5 Find a .csv file locally after the export task is completed.

----End

5.4.5 Deleting a DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas as required on the **Instance Management** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If you delete a DB instance, its automated backups are also deleted and you are no longer billed for them. Manual backups, however, are still retained and will generate additional costs.

NOTICE

- If you delete a primary DB instance, its standby DB instance and read replicas (if any) are also deleted automatically. Exercise caution when performing this operation.
 - Deleted DB instances cannot be recovered and resources are released. Exercise caution when performing this operation. If you want to retain data, [create a manual backup](#) first before deleting the DB instance.
 - You can use a manual backup to restore a DB instance. For details, see [Restoring from Backup Files to DB Instances](#).
-

Deleting a DB Instance

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.
- Step 5** In the displayed dialog box, click **Yes**.
- Step 6** Refresh the DB instance list later to confirm that the deletion was successful.
- End

Deleting a Read Replica

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

- Step 4** On the **Instance Management** page, locate the target DB instance and click . All the read replicas created for the DB instance are displayed.
- Step 5** Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.
- Step 6** In the displayed dialog box, click **Yes**.
- Step 7** Refresh the DB instance list later to check that the deletion is successful.

----End

5.4.6 Recycling a DB Instance

Scenarios

Deleted DB instances can be moved to the recycle bin. You can rebuild DB instances from the recycle bin to restore data. The DB engine, DB engine version, and storage type of the new DB instance are the same as those of the original DB instance. Other parameters can be reconfigured. DB instances that were deleted up to 7 days ago can be restored.

Constraints

- Read replicas cannot be moved to the recycle bin.
- The recycle bin is enabled by default and cannot be disabled.

Modifying Recycling Policy

NOTICE

A new recycling policy only applies to DB instances that were put in the recycle bin after the new policy was put into effect. For DB instances that were in the recycle bin before the modification, the original recycling policy takes effect.

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** In the navigation pane on the left, choose **Recycling Management**.
- Step 5** On the **Recycling Management** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances from 1 day to 7 days.
- Step 6** Then, click **OK**.

----End

Rebuilding a DB Instance

You can rebuild the primary DB instances in the recycle bin within the retention period.

- Step 1** In the navigation pane on the left, choose **Recycling Management**.
- Step 2** On the **Recycling Management** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.
- Step 3** On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see [Restoring from Backup Files to DB Instances](#).

----End

5.5 Instance Modifications

5.5.1 Upgrading a Major Version

Major versions cannot be upgraded on the management console. You can use Data Replication Service (DRS) to migrate databases from RDS for MySQL 5.6 to RDS for MySQL 5.7 smoothly. Before the migration, create a DB instance of the target version.

On the **Instance Management** page, click the instance you want to migrate. On the displayed **Basic Information** page, click **Migrate Database** in the upper right corner.

Table 5-4 MySQL database version information

Source Database Version	Destination Database Version	Migration Type
RDS for MySQL/Self-built MySQL/MySQL in other clouds • 5.5.x • 5.6.x • 5.7.x • 8.0.x	RDS for MySQL • 5.6.x • 5.7.x • 8.0.x	Version upgrade

NOTE

DRS only supports migration from an earlier version to a later version.

5.5.2 Changing a DB Instance Name

Scenarios

You can change the name of a primary DB instance or read replica.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click  next to it to edit the DB instance name. Then, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Instance Name** field in the **DB Information** area, click  to edit the DB instance name.

The DB instance name must start with a letter and consist of 4 to 64 characters. Only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_) are allowed.

- To submit the change, click .
- To cancel the change, click .

Step 5 View the results on the **Basic Information** page.

----End

5.5.3 Changing the Failover Priority

Scenarios

You can configure the failover priority for reliability or for availability, depending on your service requirements.

- **Reliability:** This option is selected by default. Data consistency is preferentially ensured during a primary/standby failover. This is recommended for users whose highest priority is data consistency.
- **Availability:** Database availability is preferentially ensured during a primary/standby failover. This is recommended for users who require their databases to provide uninterrupted online services.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary/standby DB instances.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click **Change** in the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.

Step 6 View the results on the **Basic Information** page.

----End

5.5.4 Changing a DB Instance Class

Scenarios

You can change the instance class (CPU or memory) of a DB instance as required. If the status of a DB instance changes from **Changing instance class** to **Available**, the change is successful.

Constraints

- A DB instance cannot be deleted when its instance class is being changed.
- After you change instance classes, the DB instances will reboot and services will be interrupted. Therefore, you are advised to change instance classes during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Change Instance Class** in the **Operation** column.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click **Change** in the **Instance Class** field.

Step 5 On the displayed page, specify the new instance class and click **Next**.

Step 6 View the DB instance class change result.

Changing the DB instance class takes 5–15 minutes. During this period, you can view the execution progress of **Changing a MySQL DB instance class** on the **Task Center** page. The status of the DB instance on the **Instance Management** page is **Changing instance class**. After a few minutes, click the DB instance and view the instance class on the displayed **Basic Information** page to check that the change is successful.

NOTICE

After you change a MySQL instance class, the values of the following parameters will also be changed accordingly: **back_log**, **innodb_buffer_pool_size**, **innodb_log_buffer_size**, **innodb_log_files_in_group**, **max_connections**, **innodb_page_cleaners**, **innodb_buffer_pool_instances**, **threadpool_size**, and **slave_parallel_workers**.

-----End

5.5.5 Scaling up Storage Space

Scenarios

If the original storage space is insufficient as your services grow, scale up storage space of your DB instance.

If your storage space usage reaches up to 95% for a disk less than 1 TB or the remaining space becomes 50 GB for a disk no less than 1 TB, the DB instance status becomes **Storage full** and data cannot be written to databases. In this case, scale up storage space to make the DB instance preserve at least 15% of its capacity to work properly.

You are advised to set alarm rules for the storage space usage by referring to [Setting Alarm Rules](#).

For details about the causes and solutions of insufficient storage space, see section [What Should I Do If My Data Exceeds the Available Storage of an RDS DB Instance?](#)

RDS allows you to scale up storage space of DB instances but you cannot change the storage type. During the scale-up period, services are not interrupted.

Constraints

- The maximum allowed storage is 4,000 GB. There is no limit on the number of scale-ups. If you want to increase the storage upper limit to 10,000 GB, contact customer service.
- The DB instance is in **Scaling up** state when its storage space is being scaled up and the backup services are not affected.
- For primary/standby DB instances, scaling up the primary DB instance will cause the standby DB instance to also be scaled up accordingly.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up, not down.

Scaling up a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the target DB instance to enter the **Basic Information** page. In the **Storage Space** area, click **Scale**.

Step 5 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.
- If your settings are correct, click **Submit**.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During the time period, the status of the DB instance on the **Instance Management** page will be **Scaling up**. Click the DB instance and view the utilization on the displayed **Basic Information** page to verify that the scale-up is successful.

If the DB instance is running the MySQL DB engine, you can view the detailed progress and result of the task on the **Task Center** page. For details, see section [Task Center](#).

----End

Scaling up a Read Replica

Scaling up the storage space of a read replica does not affect that of the primary DB instance. Therefore, you can separately scale read replicas to meet service requirements. New storage space of read replicas after scaling up must be greater than or equal to that of the primary DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click  in front of it. Locate a read replica to be scaled and choose **More > Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the target DB instance to enter the **Basic Information** page. In the **Storage Space** area, click **Scale**.

Step 5 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.
- If your settings are correct, click **Submit**.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During the time period, the status of the read replica on the **Instance Management** page will be **Scaling up**. Click the read replica and view the utilization on the displayed **Basic Information** page to verify that the scale-up is successful.

If the read replica is running the MySQL DB engine, you can view the detailed progress and result of the task on the **Task Center** page. For details, see section [Task Center](#).

----End

5.5.6 Changing the Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required. To prevent service interruptions, you are advised to set the maintenance window to off-peak hours.

Precautions

- During the maintenance window, the DB instance will be intermittently disconnected for one or two times. Ensure that your applications support automatic reconnection.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance. In the **DB Information** area on the **Basic Information** page, click **Change** in the **Maintenance Window** field.

Step 5 In the displayed dialog box, select a maintenance window and click **OK**.

NOTE

Changing the maintenance window does not affect the execution time of the scheduled tasks in the original maintenance period.

----End

5.5.7 Changing a DB Instance Type from Single to Primary/Standby

Scenarios

- RDS enables you to change single DB instances to primary/standby DB instances to improve instance reliability. This operation does not affect the services running on the primary DB instance.
- Primary/standby DB instances support automatic failover. If the primary DB instance fails, the standby DB instance takes over services quickly. You are advised to deploy primary and standby DB instances in different AZs for high availability and disaster recovery.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate a single DB instance and choose **More > Change Type to Primary/Standby** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  on the left to change the instance type from single to primary/standby.

Step 5 Select a standby AZ. Other configurations are the same as those of the primary DB instance by default. Confirm the configurations and click **Submit**.

Step 6 After a single DB instance is changed to primary/standby instance, you can view and manage it on the **Instance Management** page.

- The DB instance is in the **Changing type to primary/standby** status. You can view the progress on the **Task Center** page. For details, see section [Task Center](#).
- In the upper right corner of the DB instance list, click  to refresh the list. After the DB instance type is changed to primary/standby, the instance status will change to **Available** and the instance type will change to **Primary/Standby**.

----End

5.5.8 Promoting a Read Replica Into a Single DB Instance

Scenarios

RDS enables you to promote a read replica into a single DB instance. When you promote a read replica, replication is stopped. After the promotion is complete, the read replica is available as a single DB instance. This operation does not affect the performance of the original DB instance.

 NOTE

- This function is available only for MySQL 5.7 and 8.0.
- This function is unavailable for DB instances with proxy enabled.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target read replica and choose **More > Promote to Primary** in the **Operation** column.

Step 5 View the read replica status on the **Instance Management** page.

- During the promotion, the read replica status is **Promoting to primary**.
- Refresh the DB instance list by clicking  to see if the promotion is complete. After the promotion is complete, the read replica is disassociated from the original DB instance and is available as a single DB instance.
- The billing mode on the new DB instance remains unchanged.

----End

5.5.9 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create primary/standby DB instances, RDS will create a primary DB instance and a synchronous standby DB instance in the same region. You can access the primary DB instance only. The standby instance serves as a backup. You can manually promote the standby DB instance to the new primary instance for failover support.

Constraints

- A DB instance is running properly.
- The replication between the primary and standby instances is normal.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary/standby DB instance.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** in the **DB Instance Type** field.

Alternatively, click  in the DB instance topology on the **Basic Information** page to perform a primary/standby switchover.

NOTICE

A primary/standby switchover may cause service interruptions for some seconds or minutes (depending on the replication delay). If the replication delay is too long, a small amount of data may be lost. To prevent traffic congestion, you are advised to perform a switchover during off-peak hours.

Step 6 In the **Switch Primary/Standby Instances** dialog box, click **Yes** to switch between the primary and standby DB instances.

If the replication status is **Available** and the replication delay is greater than 300s, the primary/standby switchover task cannot be delivered.

Step 7 After a switchover is successful, you can view and manage the DB instance on the **Instance Management** page.

- During the switchover process, the DB instance status is **Switchover in progress**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is successful, the DB instance status will become **Available**.

----End

5.5.10 Migrating a Standby DB Instance

Scenarios

You can migrate a standby DB instance to another AZ in the same region as the original AZ.

NOTE

- Only primary/standby DB instances running MySQL 5.6 or 5.7 support standby DB instance migration to another AZ. DB instances whose storage type is local SSD cannot be migrated.
- DDL operations and scheduled events will be suspended during migration. To prevent service interruptions, perform the migration during off-peak hours.

Procedure

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and choose **More > Migrate Standby DB Instance** in the **Operation** column.
- Step 5** On the displayed page, select a target AZ and click **Submit**.
- Step 6** After the migration is complete, you can view and manage the DB instance on the **Instance Management** page.
- During the migration process, the DB instance status is **Migrating standby DB instance**. You can view the progress on the **Task Center** page. For details, see section [Task Center](#).
 - In the upper right corner of the DB instance list, click  to refresh the list. After the migration is complete, the DB instance status will become **Available**.
 - In the **DB Information** on the **Basic Information** page, you can view the AZ hosting the standby DB instance.
- End

5.6 Read Replicas

5.6.1 Introducing Read Replicas

Introduction

RDS for MySQL supports read replicas.

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To expand the DB instance read ability to offload read pressure on the database, you can create read replicas in a region. These read replicas can process a large number of read requests and increase application throughput.

A read replica uses a single-node architecture (without a slave node). Changes to the primary DB instance are also automatically synchronized to all associated read replicas through the native MySQL replication function. The synchronization is not affected by network latency. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Functions

- Read replica specifications can be different from primary DB instance specifications. It is recommended that the read replica specifications be greater than or equal to the primary DB instance specifications to prevent long delay and high load.

- Read replicas support system performance monitoring.
RDS provides up to 20 monitoring metrics, including storage space, IOPS, the number of database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on DB instances.
- Read replicas do not support automated backups or manual backups.
- Read replicas do not support restoration from backups to new, existing, or original read replicas.
- Data cannot be migrated to read replicas.
- Read replicas do not support database creation and deletion.
- Read replicas do not support database account creation.

Constraints

A maximum of five read replicas can be created for each primary DB instance.

Creating and Managing a Read Replica

- [Creating a Read Replica](#)
- [Managing a Read Replica](#)

5.6.2 Creating a Read Replica

Scenarios

Read replicas are used to enhance the read capabilities and reduce the load on primary DB instances.

After DB instances are created, you can create read replicas for them.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click **Create Read Replica** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  under the primary DB instance to create read replicas.

Step 5 On the displayed page, configure information about the DB instance and click **Next**.

Table 5-5 Basic information

Parameter	Description
Region	By default, read replicas are in the same region as the primary DB instance.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
DB Engine	Same as the DB engine of the primary DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of the primary DB instance by default and cannot be changed.
AZ	RDS allows you to deploy primary DB instances and read replicas in a single AZ or across AZs to improve reliability.

Table 5-6 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS. For details about instance classes, see section DB Instance Classes. After a DB instance is created, you can change its instance class (CPU and memory). For details, see section Changing a DB Instance Class.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • High I/O: supports a maximum throughput of 150 MB/s. • Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Storage Space	Contains the file system overhead required for inode, reserved block, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.

Table 5-7 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.
Subnet	Same as the primary DB instance's subnet. A floating IP address is automatically assigned when you create a read replica. You can also enter an unused floating IP address in the subnet CIDR block. After the read replica is created, you can change the floating IP address.
Security Group	Same as the primary DB instance's VPC.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 7 After a read replica is created, you can view and manage it.

For details about how to manage read replicas, see [Managing a Read Replica](#).

You can view the detailed progress and result of the task on the **Task Center** page.

----End

Follow-up Operations

[Managing a Read Replica](#)

5.6.3 Managing a Read Replica

Entering the Management Interface Through a Read Replica

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 In the DB instance list, click  to expand the DB instance details and click the target read replica to go to the **Basic Information** page.

----End

Entering the Management Interface Through a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.
- Step 5** In the DB instance topology, click the target read replica. You can view and manage it in the displayed pane.
- End

Deleting a Read Replica

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** In the DB instance list, click  in front of a DB instance, locate the read replica to be deleted, and choose **More > Delete** in the **Operation** column.
- End

5.7 Backups and Restorations

5.7.1 Working with Backups

RDS supports backups and restorations to ensure data reliability.

Automated Backups

Automated backups are created during the backup time window for your DB instances. RDS saves automated backups based on a retention period you specify. If necessary, you can restore a DB instance to any point in time during your backup retention period. For details, see [Configuring an Automated Backup Policy](#).

Manual Backups

Manual backups are user-initiated full backups of DB instances. They are retained until you delete them manually. For details, see [Creating a Manual Backup](#).

Downloading a Backup File

You can download a full or an incremental backup file for local data backup or restoration. For details, see [Downloading a Backup File](#) and [Downloading a Binlog Backup File](#).

5.7.2 Configuring an Automated Backup Policy

Scenarios

When you create a DB instance, an automated backup policy is enabled by default. For security purposes, the automated backup policy cannot be disabled. After the DB instance is created, you can customize the automated backup policy as required and then RDS backs up data based on the automated backup policy you configure.

RDS backs up data at the DB instance level, rather than the database level. If a database is faulty or data is damaged, you can restore it from backups. Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects database read and write performance, the automated backup time window should be set to off-peak hours.

The automated backup policy is enabled by default as follows:

- Retention period: 7 days
- Time window: An hour within 24 hours, such as 01:00-02:00 or 12:00-13:00. The backup time is configured based on UTC time and is adjusted for daylight saving time, which changes at different times depending on the time zone.
- Backup cycle: Each day of the week by default

Constraints

- The number of tables in a DB instance affects the backup speed. The maximum number of tables is 500,000.
- The system verifies the connection to the DB instance when starting a full backup task. If either of the following conditions is met, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
 - DDL operations are being performed on the DB instance.
 - The backup lock failed to be obtained from the DB instance.

Modifying an Automated Backup Policy

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Backups & Restorations** page, click **Modify Backup Policy**.

- **Retention Period** is the number of days that your automated backups are saved for. Increasing the retention period will improve data reliability.
- If you shorten the retention period, the new backup policy takes effect for all backup files. Any backup files that have expired, based on a newly configured retention period, will be deleted.
- The backup retention period is the number of days you want automated full backups and binlog backups of your DB instance to be saved for. It ranges

from 1–732 days. The backup time window is one hour. You are advised to select an off-peak time window for automated backups. By default, each day of the week is selected for **Backup Cycle** and you can change it. At least one day must be selected.

Step 6 Click **OK**.

----End

5.7.3 Setting a Cross-Region Backup Policy

Scenarios

RDS can store backup files in the storage space that is in a different region from the DB instance for disaster recovery. If a DB instance in a region is faulty, you can use the backup files in another region to restore data to a new DB instance.

After you enable cross-region backup, the backup files are automatically stored in the region you specify. On the **Backup Management** page of the RDS console, you can click **View Backup** in the **Operation** column and manage cross-region backup files.

Enabling or Modifying a Cross-Region Backup Policy

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Set Cross-Region Backup Policy**.

- If you enable **Cross-Region Full Backup**, automated full backup files of the DB instance are stored in OBS in the region you specify.
- If you enable **Cross-Region Log Backup**, binlog (incremental) backup files of the DB instance are stored in OBS in the region you specify.
- Cross-region backup files can be retained from 1 to 1,825 days.
- After the cross-region log backup function is enabled, you can restore a DB instance to a specified time point only after the next automated full backup replication is complete. The specified time point must be later than the time when the automated full backup is complete.

Step 6 Click **OK**.

Step 7 On the **Cross-Region Backups** tab of the **Backup Management** page, manage cross-region backup files.

- To modify the cross-region backup policy, click **Set Cross-Region Backup** in the **Operation** column.

- To view generated cross-region backup files, click **View Cross-Region Backup** in the **Operation** column. If a DB instance fails, you can use the cross-region backup files to restore data to a new DB instance.

----End

Disabling a Cross-Region Backup Policy

Step 1 On the **Backup Management** page, click the **Cross-Region Backups** tab.

Step 2 Locate a target DB instance and click **Set Cross-Region Backup** in the **Operation** column. On the displayed page, disable the cross-region backup policy.

Step 3 Click **OK**.

----End

5.7.4 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups of a running primary DB instance. You can use these backups to restore data.

NOTE

When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.

Constraints

- The number of tables in a DB instance affects the backup speed. The maximum number of tables is 500,000.
- The system verifies the connection to the DB instance when starting a full backup task. If either of the following conditions is met, the verification fails and a retry is automatically performed. If the retry fails, the backup will fail.
 - DDL operations are being performed on the DB instance.
 - The backup lock failed to be obtained from the DB instance.

Method 1

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Create Backup** in the **Operation** column.

Step 5 In the displayed dialog box, enter a backup name and description. Then, click **OK**. If you want to cancel the backup creation task, click **Cancel**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =
- The time required for creating a manual backup depends on the amount of data.

Step 6 After a manual backup has been created, you can view and manage it on the **Backup Management** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.

----End

Method 2

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Backups & Restorations** page, click **Create Backup**. In the displayed dialog box, enter a backup name and description and click **OK**. If you want to cancel the backup creation task, click **Cancel**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =
- The time required for creating a manual backup depends on the amount of data.

Step 6 After a manual backup has been created, you can view and manage it on the **Backup Management** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.

----End

5.7.5 Downloading a Backup File

Scenarios

This section describes how to download a manual or an automated backup file to a local device and restore data from the backup file.

RDS for MySQL enables you to download full backup files.

Method 1: Using Current Browser

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Backup Management** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.
- Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.
- Step 5** In the displayed dialog box, select **Use Current Browser** for **Download Method** and click **OK**.

----End

Method 2: Using Download URL

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Backup Management** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.
- Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.
- Step 5** In the displayed dialog box, select **Use Download URL** for **Download Method**, click  to copy the URL, and click **OK**.

A valid URL for downloading the backup data is displayed.

- You can use various download tools to download backup files.
- You can also run the following command to download backup files:

```
wget -O FILE_NAME --no-check-certificate "DOWNLOAD_URL"
```

The parameters in the command are as follows:

FILE_NAME: indicates the new backup file name after the download is successful. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with **wget** to rename the backup file.

DOWNLOAD_URL: indicates the location of the backup file to be downloaded. If the location contains special characters, escape is required.

----End

5.7.6 Downloading a Binlog Backup File

Scenarios

This section describes how to download a manual or an automated backup to a local device and restore data from the backup file.

RDS for MySQL enables you to download binlog backup files.

Downloading a Binlog Backup File

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance. The **Basic Information** page is displayed.
- Step 5** In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

You can also select the binlog backups to be downloaded and click **Download** above the list.
- Step 6** After the download is complete, you can view the binlog backups locally.

----End

5.7.7 Setting a Local Retention Period for MySQL Binlogs

Scenarios

RDS for MySQL deletes local binlogs after they are backed up to OBS. You can set the local retention period for binlogs as required.

NOTE

Binlog is enabled for RDS by default and uses the row-based logging.

Binlogs can be retained from 0 to 168 (7x24) hours locally.

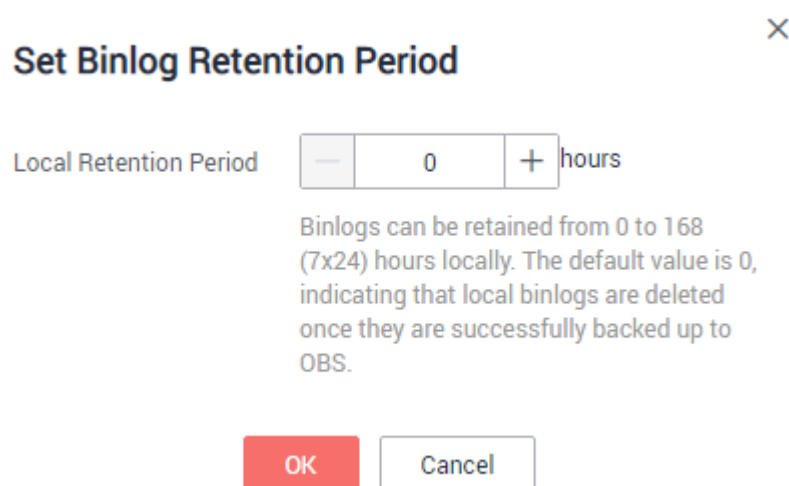
The binlogs used for replication between primary DB instances and standby DB instances or read replicas are not deleted, even when the retention period is set to 0.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.

- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, click **Set Binlog Retention Period**.
- Step 6** In the displayed dialog box, set the local retention period and click **OK**.

Figure 5-1 Setting the binlog retention period



----End

5.7.8 Restoring from Backup Files to DB Instances

Scenarios

This section describes how to use an automated or manual backup to restore a DB instance to the status when the backup was created. The restoration is at the DB instance level.

When you restore a DB instance from a backup file, a full backup file is downloaded from OBS and then restored to the DB instance at an average speed of 40 MB/s.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Backup Management** page, select the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page. On the displayed page, choose **Backups & Restorations**. On the displayed page, select the target backup to be restored and click **Restore** in the **Operation** column.

Step 5 Select a restoration method and click **OK**.

- Create New Instance

The **Create New Instance** page is displayed.

- The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
- Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.
- Other settings are the same as those of the original DB instance by default and can be modified. For details, see [Step 1: Create a DB Instance](#).

Step 6 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

After the new instance is created, a full backup will be automatically triggered.

----End

5.7.9 Restoring a DB Instance to a Point in Time

Scenarios

You can restore from automated backups to a specified point in time.

When you enter the time point that you want to restore the DB instance to, RDS downloads the most recent full backup file from OBS to the DB instance. Then, incremental backups are also restored to the specified point in time on the DB instance. Data is restored at an average speed of 30 MB/s.

Restoring a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Restore to Point in Time**.

- Step 6** Select the restoration date and time range, enter a time point within the selected time range, and select a restoration method. Then, click **OK**.
- Create New Instance
The **Create New Instance** page is displayed.
 - The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
 - Other settings are the same as those of the original DB instance by default and can be modified. For details, see [Step 1: Create a DB Instance](#).
- Step 7** View the restoration result. The result depends on which restoration method was selected:
- Create New Instance
A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.
The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.
After the new DB instance is created, a full backup will be automatically triggered.

----End

5.7.10 Replicating a Backup

Scenarios

This section describes how to replicate a manual or an automated backup. The new backup name must be different from the original backup name.

Constraints

You can replicate backups and use them only within the same region.

Backup Retention Policy

- If you want to retain the automated backups for a long time, you can replicate them to generate manual backups, which will be always retained until you delete them.
- If the storage occupied by manual backups exceeds the provisioned free backup storage, additional storage costs may incur.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

- Step 4** On the **Instance Management** page, click the target DB instance. On the **Backups & Restorations** page, locate the target backup to be replicated and click **Replicate** in the **Operation** column.
- Step 5** In the displayed dialog box, enter a new backup name and description and click **OK**.
- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
 - The description consists of a maximum of 256 characters and cannot contain the following special characters: > ! < " & ' =
- Step 6** After the new backup has been created, you can view and manage it on the **Backup Management** page.
- End

5.8 Parameter Template Management

5.8.1 Creating a Parameter Template

You can use database parameter templates to manage the DB engine configuration. A database parameter template acts as a container for engine configuration values that can be applied to one or more DB instances.

This default template contains DB engine defaults and system defaults that are configured based on the engine, compute class, and allocated storage of the instance. Default parameter templates cannot be modified, but you can create your own parameter template to change parameter settings.

NOTICE

Not all of the DB engine parameters in a custom parameter template can be changed.

If you want to use a custom parameter template, you simply create a parameter template and select it when you create a DB instance or apply it to an existing DB instance following the instructions provided in section [Applying a Parameter Template](#).

When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template following the instructions provided in section [Replicating a Parameter Template](#).

The following are the key points you should know when using parameters in a parameter template:

- When you change a dynamic parameter value in a parameter template and save the change, the change takes effect immediately. When you change a static parameter value in a parameter template and save the change, the

change will take effect only after you manually reboot the DB instances that the parameter template applies to.

- Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter template. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.

 NOTE

RDS does not share parameter template quotas with DDS.

You can create a maximum of 100 parameter templates for RDS DB instances. All RDS DB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, configure required information and click **OK**.

- Select a DB engine for the parameter template.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =

----End

5.8.2 Modifying Parameters

You can modify parameters in a custom parameter template to optimize RDS database performance.

You can only change parameter values in custom parameter templates. You cannot change the parameter values in default parameter templates.

Global parameters must be modified on the console. Session-level parameters can be modified using SQL statements. When you modify a parameter, the time when the modification takes effect depends on the type of the parameter.

The RDS console displays the statuses of DB instances that the parameter template applies to. For example, if the DB instance has not yet used the latest modifications made to its parameter template, its status is **Parameter change. Pending reboot**. Manually reboot the DB instance for the latest modifications to take effect for that DB instance.

 NOTE

RDS has default parameter templates whose parameter values cannot be changed. You can view these parameter values by clicking the default parameter templates. If a custom parameter template is set incorrectly, the database startup may fail. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Modifying Parameter Template Parameters

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the **Parameters** page, modify parameters as required.

For parameter description details, see [Suggestions on MySQL Parameter Tuning](#).

Available operations are as follows:

Figure 5-2 Modifying parameters in a parameter template

Parameters

Parameters

Change History

Save

Cancel

Previous

Parameter Name	Effective upon reload	Value	Allowed Values	Description
auto_increment_increment	No	1	1-65,535	auto_increment_increment and auto_increment_offset are intended for use with master-to-master replication, and can...
auto_increment_offset	No	1	1-65,535	auto_increment_increment and auto_increment_offset are intended for use with master-to-master replication, and can...
back_log	Yes	default	1-65,535	The number of outstanding connection requests MySQL can have. This comes into play when the main MySQL thread...
binlog_cache_size	No	32768	4096-16,777,216	The size of the cache to hold the SQL statements for the binary log during a transaction. The parameter value must be...
binlog_checksum	No	CRC32	NONE, CRC32	When enabled, this variable causes the master to write a checksum for each event in the binary log.
binlog_stmt_cache_size	No	32768	4096-16,777,216	This variable determines the size of the cache for the binary log to hold nontransactional statements issued during a r...
block_encryption_mode	No	aes-128-ctr	aes-128-ctr, aes-192-ctr, aes-256-ctr, aes-128-ecb...	Controls the block encryption mode for block based algorithms such as AES. It affects encryption for AES_ENCRYPT()...
bulk_insert_buffer_size	No	8388608	0-10,485,744,073,709,551,013	Limits the size of the MyISAM cache table in bytes per thread.
character_set_server	No	utf8	utf8, latin1, gbk, utf8mb4	The server's default character set.
collation_server	No	utf8_general_ci	utf8_general_ci, utf8mb4_general_ci, utf8mb4_unicode_ci, utf8mb4_0900_ai_ci	The server's default collation. The value must be equal to or a subset of the value character_set_server (otherwise i...

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 6 After the parameter values are modified, you can click **View Change History** to view the details.

Step 7 The modifications do not take effect until you apply the parameter template to your DB instances. For details, see [Applying a Parameter Template](#).

Step 8 View the status of the DB instance to which the parameter template was applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/

standby DB instances, the parameter modifications are also applied to the standby DB instance.)

- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

Modifying Instance Parameters

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, modify parameters as required.

NOTICE

Check the value in the **Effective upon Reboot** column.

- If the value is **Yes** and the DB instance status on the **Instance Management** page is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
 - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
 - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.
- If the value is **No**, the modifications take effect immediately.

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

After parameters are modified, you can click **Change History** to view parameter modification details.

----End

5.8.3 Exporting a Parameter Template

Scenarios

- You can export a parameter template of a DB instance for future use. You can also apply the exported parameter template to DB instances by referring to [Applying a Parameter Template](#).

- You can export the parameter template information (parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analysis.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

- Exporting to a custom template

In the displayed dialog box, configure required information and click **OK**.

NOTE

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =

After the parameter template is exported, a new template is generated in the list on the **Parameter Template Management** page.

- Exporting to a file

The parameter template information (parameter names, values, and descriptions) of a DB instance is exported to a CSV file. In the displayed dialog box, enter the file name and click **OK**.

NOTE

The file name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

----End

5.8.4 Comparing Parameter Templates

Scenarios

You can compare DB instance parameters with a parameter template that uses the same DB engine to understand the differences of parameter settings.

You can also compare default parameter templates that use the same DB engine to understand the differences of parameter settings.

Comparing Instance Parameters with a Parameter Template

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.
- Step 6** In the displayed dialog box, select a parameter template to be compared and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

Comparing Parameter Templates

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.
- Step 5** In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

5.8.5 Viewing Parameter Change History

Scenarios

You can view the change history of DB instance parameters or custom parameter templates.

NOTE

An exported or custom parameter template has initially a blank change history.

Viewing Change History of a DB Instance

- Step 1** Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.
- You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.
- You can apply the parameter template to DB instances as required by referring to [Applying a Parameter Template](#).
- End

Viewing Change History of a Parameter Template

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** Choose **Parameter Template Management** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.
- Step 5** On the displayed page, choose **Change History** in the navigation pane on the left.
- You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.
- End

5.8.6 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template.

After a parameter template is replicated, it takes about 5 minutes before the new template is displayed.

Default parameter templates cannot be replicated, but you can create parameter templates based on the default ones.

Procedure

- Step 1** Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.
- Alternatively, click the target DB instance on the **Instance Management** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.
- Step 5** In the displayed dialog box, configure required information and click **OK**.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
 - The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =
- After the parameter template is replicated, a new template is generated in the list on the **Parameter Template Management** page.
- End

5.8.7 Resetting a Parameter Template

Scenarios

You can reset all parameters in a custom parameter template to their default settings.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and choose **More > Reset** in the **Operation** column.
- Step 5** Click **Yes**.
- Step 6** The modifications take effect only after you apply the parameter template to DB instances. For details, see [Applying a Parameter Template](#).
- Step 7** View the status of the DB instance to which the parameter template is applied.
- If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.

- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

5.8.8 Applying a Parameter Template

Scenarios

You can apply parameter templates to DB instances as needed.

- The parameter **innodb_buffer_pool_size** is determined by the memory. DB instances of different specifications have different value ranges. If this parameter value is out of range of the DB instance that the parameter template applies to, the maximum value within the range is used.
- A parameter template can be applied only to DB instances of the same DB engine version.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, perform the following operations based on the type of the parameter template to be applied:

- If you intend to apply a default parameter template to DB instances, click **Default Templates**, locate the target parameter template, and click **Apply** in the **Operation** column.
- If you intend to apply a custom parameter template to DB instances, click **Custom Templates**, locate the target parameter template, and choose **More** > **Apply** in the **Operation** column.

A parameter template can be applied to one or more DB instances.

Step 5 In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

After the parameter template is successfully applied, you can view the application records by referring to section [Viewing Application Records of a Parameter Template](#).

----End

5.8.9 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left.

Step 5 On the **Default Templates** page, locate the target parameter template and click **View Application Record** in the **Operation** column. Alternatively, on the **Custom Templates** page, choose **More > View Application Record** in the **Operation** column.

You can view the name or ID of the DB instance that the parameter template applies to, as well as the application status, application time, and failure cause (if failed).

----End

5.8.10 Modifying a Parameter Template Description

Scenarios

You can modify the description of a parameter template you have created.

NOTE

You cannot modify the description of a default parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**.
Locate the target parameter template and click  in the **Description** column.

Step 5 Enter a new description. You can click  to submit or to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain the following special characters: > ! < " & ' =

- After the modification is successful, you can view the new description in the **Description** column of the parameter template list.

----End

5.8.11 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer in use.

NOTICE

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template to be deleted and choose **More > Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

----End

5.9 Connection Management

5.9.1 Configuring and Changing a Floating IP Address

Scenarios

You can change floating IP addresses after migrating on-premises databases or other cloud databases to RDS.

Constraints

After a floating IP address is changed, the domain name needs to be resolved again. This operation takes several minutes and may interrupt database connections. Therefore, you are advised to change a floating IP address during off-peak hours.

Configuring a Floating IP Address

You can use an automatically-assigned IP address when creating a DB instance.

Changing a Floating IP Address

You can change the floating IP address of an existing DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the **Connection Information** area on the **Basic Information** page, click **Change** in the **Floating IP Address** field.

Step 6 In the displayed dialog box, enter a new floating IP address and click **OK**.

In the in-use IP address list, the IP addresses whose statuses are **To be used** are occupied and cannot be reused.

----End

5.9.2 Binding and Unbinding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility, and you can unbind the EIP from the DB instance later if needed.

NOTICE

To ensure that the DB instance is accessible, the security group associated with the instance must allow access over the database port. For example, if the database port is 8635, ensure that the security group allows access over the 8635 port.

Prerequisites

- You have assigned an EIP on the VPC console.
- You can bind an EIP to a primary DB instance or a read replica only.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the DB instance first before binding a new EIP to it.

Binding an EIP

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **EIPs**. On the displayed page, click **Bind EIP**.
- Step 6** In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **OK**. If no available EIPs are displayed, click **View EIP** to obtain one.
- Step 7** On the **EIPs** page, view the EIP that has been bound to the DB instance.
- You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.
- To unbind the EIP from the DB instance, see [Unbinding an EIP](#).
- End

Unbinding an EIP

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the DB instance that has an EIP bound.
- Step 5** On the **EIPs** page, locate the target EIP to be unbound and click **Unbind**. In the displayed dialog box, click **Yes**.
- Step 6** On the **EIPs** page, view the unbinding result.
- You can also view the progress and result of unbinding an EIP from a DB instance on the **Task Center** page.
- To bind an EIP to the DB instance again, see [Binding an EIP](#).
- End

5.9.3 Changing a Database Port

Scenarios

This section describes how to change the database port of a primary DB instance or a read replica. For primary/standby DB instances, changing the database port of the primary DB instance will cause the database port of the standby DB instance to also be changed accordingly.

Constraints

When the database port of a DB instance is being changed, you cannot:

- Bind an EIP to the DB instance.
- Delete the DB instance.
- Create a backup for the DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance or click  first and then click the target read replica.

Step 5 In the **Connection Information** area on the **Basic Information** page, click  in the **Database Port** field.

NOTE

A MySQL database can use ports 2100 to 9500.

- To submit the change, click .
 - In the dialog box, click **Yes**.
 - i. If you change the database port of the primary DB instance, that of the standby DB instance will also be changed and both DB instances will be rebooted.
 - ii. If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will be rebooted.
 - iii. This process takes 1-5 minutes.
 - In the dialog box, click **No** to cancel the modification.
- To cancel the change, click .

Step 6 View the results on the **Basic Information** page.

----End

5.9.4 Downloading a Certificate

RDS for MySQL allows you to download a certificate.

Downloading a Certificate

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance. The **Basic Information** page is displayed.

Step 5 In the **DB Information** area, click  in the **SSL** field to download the root certificate or certificate bundle.

 NOTE

- Since April 2017, RDS has offered a new root certificate that has a 20-year validation period. The new certificate takes effect after DB instances are rebooted. Replace the old certificate before it expires to improve system security.

For details, see section [How Can I Identify the Validity Period of an SSL Root Certificate?](#)

- You can also download the certificate bundle, which contains both the new certificate provided since April 2017 and the old certificate.

----End

5.9.5 Configuring a Security Group Rule

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted in a VPC.

This section describes how to create a security group to enable specific IP addresses and ports to access RDS.

- When you attempt to connect to an RDS DB instance through an EIP, you need to configure an inbound rule for the security group associated with the DB instance.
- Check whether the ECS and RDS DB instance are in the same security group.
 - If the ECS and RDS DB instance are in the same security group, they can communicate with each other by default. No security group rule needs to be configured.
 - If the ECS and RDS DB instance are in different security groups, you need to configure security group rules for them, separately.
 - RDS DB instance: Configure an inbound rule for the security group with which the RDS DB instance is associated.
 - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security rule for the ECS. If not all outbound traffic is allowed in the security group, you need to configure an **outbound rule** for the ECS.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are deployed in the same security group. After a security group is created, you can configure security group rules to control access from and to the DB instances in the security group.

- By default, a tenant can create a maximum of 100 security groups.
- By default, each security group can have a maximum of 50 security group rules.

- One security group can be associated with only one RDS DB instance.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for each security group.
- To enable access to an RDS DB instance from resources outside the security group, you need to configure an **inbound rule** for the security group associated with the RDS DB instance.

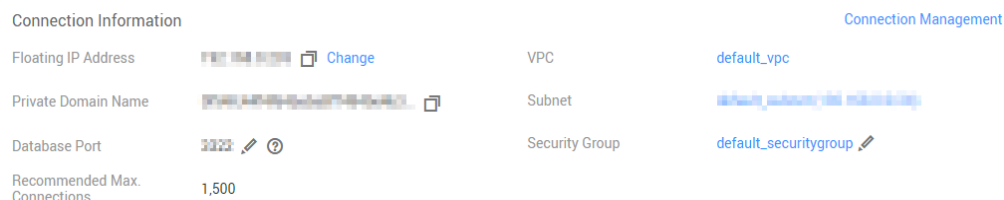
 NOTE

If you use **0.0.0.0/0**, RDS DB instances in the security group can be accessed from any IP address.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** Configure security group rules.
- In the **Connection Information** area on the **Basic Information** page, click the security group.

Figure 5-3 Connection information



- Step 6** On the inbound rule tab, click **Add Rule**. In the displayed dialog box, set required parameters to add an inbound rule.
- You can click **+** to add more inbound rules.

Figure 5-4 Adding an inbound rule

Add Inbound Rule

[Learn more](#) about security group configuration.

×

An inbound rule allows inbound traffic to instances in the security group.

Security Group

default_securitygroup

You can import multiple rules in a batch.

Protocol & Port ?	Source ?	Operation
Custom TCP	IP address	
Example: 22 or 22-30	0 . 0 . 0 . 0 / 0	Operation

+

 Add Rule You can create 4,985 more security group rules. [Increase quota](#)

OK

Cancel

Table 5-8 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	Protocol: specifies the network protocol. Currently, the value can be All , TCP , UDP , ICMP , GRE , or others.	Custom TCP
	Port: specifies the port or port range over which the traffic can reach your ECS.	When connecting to the DB instance through a private network, enter the port of the target DB instance.
Source	Specifies the source of the security group rule. The value can be an IP address or a security group. For example: - xxx.xxx.xxx.xxx/32 (IPv4 address) - xxx.xxx.xxx.0/24 (subnet) 0.0.0.0/0 (any IP address)	0.0.0.0/0
Description	Provides supplementary information about the security group rule. This parameter is optional. The description can contain a maximum of 255 characters and cannot contain angle brackets (<) or (>).	-

----End

5.10 Database Management

5.10.1 Creating a Database

Scenarios

After a DB instance is created, you can create databases on it.

Constraints

- Databases cannot be created for DB instances that are in the process of being restored.
- This section applies only to the MySQL DB engine.

Creating a Database through RDS

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Databases** page, click **Create Database**. In the displayed dialog box, enter a database name and remarks, select a character set, and authorize permissions for users. Then, click **OK**.

- The database name consists of 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and dollar signs (\$) are allowed. MySQL 8.0 does not support dollar signs (\$). The total number of hyphens (-) and dollar signs (\$) cannot exceed 10.
- The default character set is **utf8**. You can click **More** and select another one.

- Select unauthorized users and click  to authorize permissions (mandatory) or select authorized users and click  to revoke permissions. If there are no unauthorized users, you can create one by referring to section [Creating a Database Account](#).

Step 6 After the database is created, you can manage it on the **Databases** page of the selected DB instance.

NOTICE

The **AUTO_PK_ROW_ID** column name is a reserved column name for the RDS for MySQL database and cannot be created by users.

----End

5.10.2 Granting Database Permissions

Scenarios

You can grant permissions to database users you have created to use specific databases or revoke permissions from specific database users.

Constraints

Permissions cannot be granted to database users for a DB instance that is in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the target database and click **Authorize** in the **Operation** column.

Step 6 In the displayed dialog box, select unauthorized users and click  to authorize them or select authorized users and click  to revoke permissions.

If there are no unauthorized users, you can create one by referring to [Creating a Database Account](#).

Step 7 In the dialog box, click **OK**.

----End

5.10.3 Deleting a Database

Scenarios

You can delete databases that you have created.

NOTICE

Deleted databases cannot be recovered. Exercise caution when performing this operation.

Constraints

Custom databases cannot be deleted from DB instances that are in the process of being restored.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** On the **Databases** page, locate the target database and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

----End

5.10.4 Enabling or Disabling Event Scheduler

Scenarios

Event scheduler manages the scheduling and execution of events. The MySQL built-in event scheduler cannot guarantee the consistency of event statuses between primary and standby DB instances. If a failover or switchover occurs, events will not be scheduled. RDS for MySQL resolves this issue. With RDS for MySQL, even if there is a failover or switchover, the event status will still be properly scheduled. You can simply enable or disable the event scheduler on the RDS console.

- By default, the event scheduler is disabled after a DB instance is created.
- After a primary/standby failover or switchover, the event status remains unchanged. The **event_scheduler** is **on** for the original primary DB instance and **off** for the original standby DB instance.
- After a restoration to a new DB instance, the event status is the same as that of the original DB instance.
- After a single DB instance is changed to primary/standby DB instances, the event status is the same as that of the primary DB instance.

Constraints

- Only MySQL kernel 5.6.43.2, 5.7.25.2, and later versions are supported.
- Event scheduler cannot be enabled for read replicas.

Enabling Event Scheduler

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target primary/standby DB instances.
- Step 5** In the **DB Information** area on the displayed **Basic Information** page, click  in the **Event Scheduler** field.

NOTICE

After you enable the event scheduler on the RDS console, log in to the DB instance and check that the event status has been set to **enable**.

----End

Disabling Event Scheduler

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target primary/standby DB instances.
- Step 5** In the **DB Information** area on the displayed **Basic Information** page, click  in the **Event Scheduler** field.

----End

5.11 Account Management (Non-Administrator)

5.11.1 Creating a Database Account

Scenarios

When you create a DB instance, account **root** is created at the same time by default. You can create other database accounts as needed.

Constraints

Database accounts cannot be created for DB instances that are in the process of being restored.

Creating a Database Account Through RDS

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Accounts** page, click **Create Account**. In the displayed dialog box, enter a username, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

- The username consists of 1 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
 - If the database version is MySQL 5.6, the username consists of 1 to 16 characters.
 - If the database version is MySQL 5.7 or 8.0, the username consists of 1 to 32 characters.

- Select unauthorized databases and click  to authorize them or select authorized databases and click  to revoke permissions.

If there are no unauthorized databases, you can create one by referring to section [Creating a Database](#). You can also modify the permissions after the database creation by referring to section [Changing Permissions for a Database Account](#).

- The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#\$\$%^*_-=+?,).

Step 6 After the account is created, you can manage it on the **Accounts** page of the selected DB instance.

----End

5.11.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect against brute force hacking attempts and ensure system security, change your password periodically, such as every three or six months.

Constraints

Passwords cannot be reset for DB instances that are in the process of being restored.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **Reset Password** in the **Operation** column.
- Step 6** In the displayed **Reset Password** dialog box, enter and confirm a new password, and click **OK**.
 - The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#\$%^*_-=+?,).
 - The password you entered in the **Confirm Password** text box must be the same as that you entered in the **New Password** text box.
 - After the password is reset, the database will not be rebooted and permissions will not be changed.

----End

5.11.3 Changing Permissions for a Database Account

Scenarios

You can authorize database users you have created to specific databases or revoke permissions from authorized database users.

Constraints

Permissions cannot be changed for DB instances that are in the process of being restored.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and choose **More > Change Permission** in the **Operation** column.

Step 6 In the displayed dialog box, select unauthorized databases and click  to authorize them. You can also select authorized databases and click  to revoke permissions.

- If there are no unauthorized databases, you can create one by referring to [Creating a Database](#).

Step 7 Click **OK**.

----End

5.11.4 Deleting a Database Account

Scenarios

You can delete database accounts you have created.

NOTICE

Deleted database accounts cannot be restored. Exercise caution when deleting an account.

Constraints

Accounts cannot be deleted from DB instances that are in the process of being restored.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the target username and choose **More > Delete** in the **Operation** column.

Step 6 In the displayed dialog box, click **Yes**.

----End

5.12 Database Account Security

Password Strength Requirements

RDS has a password security policy for user-created database accounts. Passwords must:

- Consist of at least eight characters.
- Contain at least one uppercase letter, one lowercase letter, one digit, and one special character.

When you are creating a DB instance, the password strength is checked. You can modify the password strength as user **root**. For security reasons, you are advised to use a password that is at least as strong as the default one.

Account Description

To provide O&M services, the system automatically creates system accounts when you create MySQL DB instances. These accounts cannot be used by users.

NOTICE

Attempting to delete, rename, and change passwords or permissions for these accounts will result in an error. Exercise caution when performing these operations.

- **rdsAdmin**: indicates the management account, which has the superuser permissions and is used to query and modify DB instance information, rectify faults, migrate data, and restore data.
- **rdsRepl**: indicates the replication account, which is used to synchronize data from primary DB instances to standby DB instances or read replicas.
- **rdsBackup**: indicates the backup account, which is used for backend backup.
- **rdsMetric**: indicates the metric monitoring account, which is used by watchdog to collect database status data.
- **rdsProxy**: indicates the proxy account, which is automatically created when read/write splitting is enabled and is used for authentication when a database is connected through a read/write splitting address.

5.13 Data Security

5.13.1 Resetting the Administrator Password

Scenarios

You can reset the administrator password of a primary instance.

You can also reset the password of your database account when using RDS.

Precautions

- If you have changed the administrator password of the primary DB instance, the administrator passwords of the standby DB instance and read replicas (if any) will also be changed accordingly.
- The time required for the new password to take effect depends on the amount of service data currently being processed by the primary DB instance.
- To protect against brute force hacking attempts and ensure system security, change your password periodically.

Method 1

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and choose **More > Reset Password** in the **Operation** column.
- Step 5** Enter and confirm a new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@\$#%^*_-=+?,.). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

-----End

Method 2

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the **DB Information** area on the **Basic Information** page, click **Reset Password** in the **Administrator** field.

Step 6 Enter and confirm a new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@\$#%^*_-=+?,.). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

5.13.2 Changing a Security Group

Scenarios

This section describes how to change the security group of a primary DB instance or read replica. For primary/standby DB instances, changing the security group of the primary DB instance will cause the security group of the standby DB instance to also be changed accordingly.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary DB instance or read replica.

Step 5 In the **Connection Information** area on the **Basic Information** page, click  in the **Security Group** field.

- To submit the change, click .
- To cancel the change, click .

Step 6 Changing the security group takes 1 to 3 minutes. Click  in the upper right corner on the **Basic Information** page to view the results.

----End

5.14 Metrics and Alarms

5.14.1 Configuring Displayed Metrics

The RDS Agent monitors RDS DB instances and collects monitoring metrics only.

Description

This section describes the RDS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS.

Namespace

SYS.RDS

DB Instance Monitoring Metrics

[Table 5-9](#) lists the performance metrics of MySQL databases.

Table 5-9 Database performance metrics

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100%	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100%	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds006_conn_count	Total Connections	Total number of connections that attempt to connect to the MySQL server	≥ 0 counts	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds007_conn_active_count	Current Active Connections	Number of current active connections	≥ 0 counts	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds008_qps	QPS	Query times of SQL statements (including stored procedures) per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds009_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0 transactions/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds010_innodb_buf_usage	Buffer Pool Usage	Ratio of idle pages to the total number of buffer pool pages in the InnoDB buffer	0–1	Monitored object: database Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds011_innod_b_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0–1	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds012_innod_b_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to used pages in the InnoDB buffer	0–1	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds013_innod_b_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥ 0 bytes/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds014_innod_b_writes	InnoDB Write Throughput	Number of write bytes per second in the InnoDB buffer	≥ 0 bytes/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds015_innod_b_read_count	InnoDB File Read Frequency	Number of times that InnoDB reads data from files per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds016_innod_b_write_count	InnoDB File Write Frequency	Number of times that InnoDB writes data to files per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds017_innodb_log_write_requests_per_second	InnoDB Log Write Requests per Second	Number of InnoDB log write requests per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds018_innodb_log_physical_write_frequency	InnoDB Log Physical Write Frequency	Number of InnoDB physical write times to log files per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds019_innodb_log_fsync_write_frequency	InnoDB Log fsync() Write Frequency	Number of completed fsync() write times to InnoDB log files per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds021_myisam_buffer_usage	Key Buffer Usage	MyISAM key buffer usage	0–1	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds022_myisam_buffer_write_hit_ratio	Key Buffer Write Hit Ratio	MyISAM key buffer write hit ratio	0–1	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds023_myisam_buffer_read_hit_ratio	Key Buffer Read Hit Ratio	MyISAM key buffer read hit ratio	0–1	Monitored object: database Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds024_myisam_disk_write_count	MyISAM Disk Write Frequency	Number of times that indexes are written to disks per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds025_myisam_disk_read_count	MyISAM Disk Read Frequency	Number of times that indexes are read from disks per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds026_myisam_buf_write_count	MyISAM Buffer Pool Write Requests per Second	Number of requests for writing indexes into the MyISAM buffer pool per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds027_myisam_buf_read_count	MyISAM Buffer Pool Read Requests per Second	Number of requests for reading indexes from the MyISAM buffer pool per second	≥ 0 counts/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds028_com_delete_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds029_com_insert_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds030_comd_ml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds031_comd_ml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds032_comd_ml_rep_sel_count	REPLACE_SELECT Statements per Second	Number of REPLACE_SELECT statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds033_comd_ml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds034_comd_ml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0 queries/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds036_innod_b_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds037_innod_b_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds038_innod_b_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥ 0 rows/s	Monitored object: database Monitored instance type: MySQL instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0-100%	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	Monitored object: ECS Monitored instance type: MySQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: MySQL instance	1 minute

Dimension

Key	Value
rds_instance_id	MySQL DB instance ID

5.14.2 Setting Alarm Rules

Scenarios

- You can set alarm rules by referring to [Setting Alarm Rules](#) to customize the monitored objects and notification policies and keep track of the RDS running status.
The RDS alarm rules include alarm rule names, resource types, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Setting Alarm Rules

Step 1 Log in to the management console.

Step 2 Click **Service List** and choose **Management & Governance > Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Alarm Management > Alarm Rules**.

Step 4 On the displayed **Alarm Rules** page, click **Create Alarm Rule**.

----End

5.14.3 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS monitoring metrics on the management console.

Monitored data takes some time before it can be displayed. The RDS status displayed on the Cloud Eye console is about 5 to 10 minutes delayed. When you create a new RDS DB instance, it takes 5 to 10 minutes before monitoring data is displayed on Cloud Eye.

Prerequisites

- RDS is running properly.
Monitoring metrics of the RDS DB instances that are faulty or have been deleted are not displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to normal.

NOTE

If an RDS DB instance has been faulty for 24 hours, Cloud Eye considers it to no longer exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- RDS has been running properly for about 10 minutes.
For a newly created RDS DB instance, you need to wait a bit before you can view the metrics.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click **View Metric** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metric** in the upper right corner of the page to go to the Cloud Eye console.

Step 5 On the Cloud Eye console, view monitoring metrics of the primary DB instance.

Cloud Eye can monitor performance metrics from the last 1 hour, 3 hours, 12 hours, 1 day, and 7 days.

----End

5.15 Interconnection with CTS

5.15.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 5-10 RDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or a read replica, or restoring data to a new DB instance	instance	createInstance
Scaling up storage space and changing instance class	instance	instanceAction
Rebooting a DB instance	instance	instanceRestart
Restoring data to the original DB instance	instance	instanceRestore
Renaming a DB instance	instance	instanceRename
Resetting a password	instance	resetPassword
Setting database version parameters	instance	setDBParameters
Resetting database version parameters	instance	resetDBParameters
Enabling, modifying, or disabling a backup policy	instance	setBackupPolicy
Changing a database port	instance	changeInstancePort
Binding or unbinding an EIP	instance	setOrResetPublicIP
Modifying a security group	instance	modifySecurityGroup
Adding a tag	instance	createTag
Deleting a tag	instance	deleteTag
Editing a tag	instance	modifyTag
Deleting a DB instance	instance	deleteInstance
Performing a primary/standby switchover	instance	instanceFailOver
Changing the replication mode	instance	instanceFailOver-Mode
Changing a failover priority	instance	instanceFailOver-Strategy

Operation	Resource Type	Trace Name
Changing a DB instance type from single to primary/standby	instance	modifySingleToHaIn-stance
Creating a backup	backup	createManualSnap-shot
Replicating a backup	backup	copySnapshot
Downloading a backup (using OBS)	backup	downLoadSnapshot
Downloading a backup (using a browser)	backup	backupsDownLoad
Deleting a backup	backup	deleteManualSnap-shot
Downloading a merged backup	backup	packBackupsDown-Load
Creating a parameter template	parameterGroup	createParameterGrou-p
Modifying parameters in a parameter template	parameterGroup	updateParameterGro-up
Deleting a parameter template	parameterGroup	deleteParameterGrou-p
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Applying a parameter template	parameterGroup	applyParameterGrou-p
Saving parameters in a parameter template	parameterGroup	saveParameterGroup
Deleting a frozen DB instance	all	rdsUnsubscribeIn-stance
Freezing a DB instance	all	rdsfreezeInstance

5.15.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

This section describes how to query the operation records of last 7 days on the CTS console.

Procedure

- Step 1** Log in to the management console.
- Step 2** Choose **Management & Governance > Cloud Trace Service**.
- Step 3** Choose **Trace List** in the navigation pane on the left.
- Step 4** Filter conditions to query traces. The details are described as follows:
- **Trace Source, Resource Type, and Search By:** Select a filter from the drop-down list.
When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
 - **Operator:** Select a specific operator from the drop-down list.
 - **Trace Status:** Available options include **All trace statuses, Normal, Warning, and Incident**. You can only select one of them.
 - In the upper right corner of the page, you can specify a time range for querying traces.
- Step 5** Select the search criteria, and click **Query**.
- Step 6** Click  on the left of the required trace to expand its details.
- Step 7** Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.
- Step 8** Click **Export** on the right. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to traces on the management console.
- For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.
- End

5.16 Log Management

5.16.1 Viewing and Downloading Error Logs

RDS log management allows you to view database-level logs, including error logs and slow SQL query logs.

Viewing Log Details

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Log Details** to view details about error logs.

- You can select a log level in the upper right corner to view logs of the selected level.

 NOTE

For MySQL DB instances, the following levels of logs are displayed:

- ERROR
- WARNING
- NOTE
- You can click  in the upper right corner to view error logs generated in different time segments.
- If the description of a log is truncated, locate the log and move your pointer over the description in the **Description** column to view details.

----End

Downloading an Error Log

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Download**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
 - When the log is being prepared for download, the log status is **Preparing**.
 - When the log is ready for download, the log status is **Preparation completed**.
 - If the preparation for download fails, the log status is **Abnormal**.
- Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

----End

5.16.2 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed **long_query_time** (1 second by default). You can view log details and statistics to identify statements that are

executing slowly and optimize the statements. You can also download slow query logs for service analysis.

RDS supports the following statement types:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE

Viewing Log Details

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Log Details** to view details about slow query logs.

- You can view the slow query log records of a specified execution statement type or a specific time period.
- The **long_query_time** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **long_query_time** is changed from 1s to 0.1s, none of the previously recorded logs that do not meet the new threshold are deleted. For example, a 1.5s SQL statement that was recorded when the threshold was 1s will not be deleted now that the new threshold is 2s.

----End

Viewing Statistics

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Statistics** to view details.

 NOTE

- On the **Statistics** page, only one of the SQL statements of the same type is displayed as an example. For example, if two `select sleep(N)` statements, **`select sleep(1)`** and **`select sleep(2)`**, are executed in sequence, only **`select sleep(1)`** will be displayed.
- The **`long_query_time`** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **`long_query_time`** is changed from 1s to 0.1s, none of the previously recorded logs that do not meet the new threshold are deleted. For example, a 1.5s SQL statement that was recorded when the threshold was 1s will not be deleted now that the new threshold is 2s.

----End

Downloading a Slow Query Log

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Download**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
 - When the log is being prepared for download, the log status is **Preparing**.
 - When the log is ready for download, the log status is **Preparation completed**.
 - If the preparation for download fails, the log status is **Abnormal**. Logs in the **Preparing** or **Abnormal** status cannot be downloaded.
- The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. If you need to download the log, click **OK**.

----End

5.16.3 Viewing Failover/Switchover Logs

You can view failover or switchover logs of MySQL DB instances to evaluate the impact on services.

 NOTE

Only failover and switchover logs within 30 days are displayed.

Procedure

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Logs**. On the displayed page, click **Failover/Switchover Logs** to view log details.
- End

5.16.4 Enabling the SQL Audit Function

After you enable the SQL audit function, all SQL operations will be recorded in log files. You can [download](#) audit logs to view log details.

By default, SQL audit is disabled because enabling this function may affect database performance. This section describes how to enable, modify, or disable SQL audit.

NOTE

- After you enable the SQL audit function, the system records all SQL operations and uploads logs every half an hour or when the size is accumulated to 100 MB.
- After SQL audit is enabled, log files will occupy your backup space.

Constraints

Only the versions listed below support SQL audit. .

- MySQL 5.6.43 and later versions for DB instances using cloud disks, MySQL 5.6.47.3 and later versions for DB instances using local disks
- MySQL 5.7.23 and later versions for DB instances using cloud disks, MySQL 5.7.29.3 and later versions for DB instances using local disks
- MySQL 8.0

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **SQL Audits**. On the displayed page, click **Set SQL Audit** above the list. In the displayed dialog box, configure information as required and click **OK**.

Enabling or setting SQL audit

- To retain SQL audit logs, set  (disabled) to  (enabled).

- Audit logs are retained for 7 days by default but can be retained from anywhere from 1 to 732 days if needed.

Disabling SQL audit

To disable SQL audit, toggle  (enabled) to  (disabled).

- If you select the check box "I acknowledge that after audit log is disabled, all audit logs are deleted.", all audit logs will be deleted.

NOTICE

Deleted audit logs cannot be recovered. Exercise caution when performing this operation.

- If you do not select the check box "I acknowledge that after audit log is disabled, all audit logs are deleted.", the audit logs will be retained.

----End

5.16.5 Downloading SQL Audit Logs

If you [enable the SQL audit function](#), all SQL operations will be logged, and you can download audit logs to view details. By default, the SQL audit function is disabled. Enabling this function may affect performance.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **SQL Audits**.

Step 6 On the displayed page, select a time range in the upper right corner, select SQL audit logs to be downloaded in the list, and click **Download** above the list to download SQL audit logs in batches.

Alternatively, select an audit log and click **Download** in the **Operation** column to download an individual SQL audit log.

Step 7 The following figure shows the SQL audit log content. For field descriptions, see [Table 5-11](#).

Figure 5-5 MySQL audit logs

```
"6","408543","1159","Connect",2020-03-20T03:35:05 UTC,"","","","","","",""
"7","408543","0","Quit",2020-03-20T03:35:05 UTC,"","","","","","",""
"8","408544","1159","Connect",2020-03-20T03:35:20 UTC,"","","","","","",""
"9","408544","0","Quit",2020-03-20T03:35:20 UTC,"","","","","","",""
"10","408546","1159","Connect",2020-03-20T03:35:35 UTC,"","","","","","",""
"11","408546","0","Quit",2020-03-20T03:35:35 UTC,"","","","","","",""
"12","408547","1159","Connect",2020-03-20T03:35:50 UTC,"","","","","","",""
"13","408547","0","Quit",2020-03-20T03:35:50 UTC,"","","","","","",""
```

Table 5-11 Audit log field description

Parameter	Description
record_id	ID of a single audit log record.
connection_id	ID of the session executed by the record, which is the same as the ID in the show processlist command output.
connection_status	Session status, which is usually the returned error code of a statement. If a statement is successfully executed, the value 0 is returned.
name	Recorded type name. Generally, DML and DDL operations are QUERY, connection and disconnection operations are CONNECT and QUIT, respectively.
timestamp	Recorded UTC time.
command_class	SQL command type. The value is the parsed SQL type, for example, select or update. (This field does not exist if the connection is disconnected.)
sqltext	Executed SQL statement content. (This field does not exist if the audit connection is disconnected.)
user	Login account.
host	Login host. The value is localhost for local login and is empty for remote login.
external_user	External username.
ip	IP address of the remotely-connected client. The local IP address is empty.
default_db	Default database on which SQL statements are executed.

----End

5.17 Task Center

5.17.1 Viewing a Task

You can view the progress and results of scheduled and instant tasks on the **Task Center** page.

 NOTE

RDS allows you to view and manage the following tasks:

- Creating DB instances
- Creating read replicas
- Scaling up storage space
- Changing single DB instances to primary/standby
- Switching primary/standby DB instances
- Rebooting DB instances
- Binding EIPs to DB instances
- Unbinding EIPs from DB instances
- Restoring data to new DB instances
- Migrating a standby MySQL DB instance

Viewing an Instant Task

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. Locate the target task and view its details on the displayed **Instant Tasks** page.

- To identify the target task, you can use the task name or DB instance name/ID, or simply enter the target task name in the search box in the upper right corner.
- You can view the progress and status of tasks in a specific period. The default period is seven days.
The task list shows tasks that have been executed in the past 30 days.
- You can view instant tasks in the following statuses:
 - Running
 - Completed
 - Failed
- You can view the task creation and completion time.

----End

Viewing a Scheduled Task

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, view the task progress and results.

- To identify the target task, you can use the DB instance name/ID or enter the target DB instance ID in the search box in the upper right corner.
- You can view the scheduled tasks in the following statuses:
 - Running
 - Completed
 - Failed
 - Canceled
 - To be executed
 - To be authorized

----End

5.17.2 Deleting a Task Record

You can delete task records so that they are no longer displayed in the task list. This operation only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Deleting an Instant Task Record

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed **Instant Tasks** page, locate the target task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

You can delete the records of instant tasks in any of the following statuses:

- Running
- Completed
- Failed

----End

Deleting a Scheduled Task Record

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the target task record to be deleted and check whether the task status is **To be executed** or **To be authorized**.

- If yes, go to [Step 5](#).
- If no, go to [Step 6](#).

Step 5 Click **Cancel** in the **Operation** column. In the displayed dialog box, click **Yes** to cancel the task. Then, click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the task record.

You can delete the records of scheduled tasks in any of the following statuses:

- Running
- Completed
- Failed
- Canceled
- To be executed
- To be authorized

----End

6 Working with RDS for PostgreSQL

6.1 Data Migration

6.1.1 Migrating Data to RDS for PostgreSQL Using psql

Preparing for Data Migration

PostgreSQL supports logical backups. You can use the `pg_dump` logical backup function to export backup files and then import them to RDS using `psql`.

You can access RDS DB instances through an EIP or through an ECS.

Preparations

1. Prepare an ECS for accessing DB instances in the same VPC or prepare a device for accessing RDS through an EIP.
 - To connect to a DB instance through an ECS, you need to create an ECS first.
 - To connect to a DB instance through an EIP, you must:
 - i. Bind an EIP to a DB instance. For details, see [Binding an EIP](#).
 - ii. Ensure that the local device can access the EIP that has been bound to the DB instance.
2. Install the PostgreSQL client on the prepared ECS or device.

NOTE

The PostgreSQL client version must be the same as the version of RDS for PostgreSQL. The PostgreSQL database or client will provide `pg_dump` and `psql`.

Exporting Data

Before migrating an existing PostgreSQL database to RDS, you need to export the PostgreSQL database.

NOTICE

- The export tool must match the DB engine version.
- Database migration is performed offline. Before the migration, you must stop any applications using the source database.

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Use the `pg_dump` tool to export the source database into an SQL file.

```
pg_dump --username=<DB_USER> --host=<DB_ADDRESS> --port=<DB_PORT> --format=plain --file=<BACKUP_FILE> <DB_NAME>
```

- **DB_USER** indicates the database username.
- **DB_ADDRESS** indicates the database address.
- **DB_PORT** indicates the database port.
- **BACKUP_FILE** indicates the name of the file to which the data will be exported.
- **DB_NAME** indicates the name of the database to be migrated.

Enter the database password when prompted.

Example:

```
$ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db
```

Password for user root:

After this command is executed, a **backup.sql** file will be generated as follows:

```
[rds@localhost ~]$ ll backup.sql
```

```
-rw-r-----. 1 rds rds 2714 Sep 21 08:23 backup.sql
```

-----End

Importing Data

Step 1 Log in to the ECS or the device that can access RDS.

Step 2 Ensure that the destination database to which data is to be imported exists.

If the destination database does not exist, run the following command to create a database:

```
# psql --host=<RDS_ADDRESS> --port=<DB_PORT> --username=root --dbname=postgres -c "create database <DB_NAME>;"
```

- **RDS_ADDRESS** indicates the IP address of the RDS DB instance.
- **DB_PORT** indicates the RDS DB instance port.
- **DB_NAME** indicates the name of the database to be imported.

Step 3 Import the exported file to RDS.

```
# psql --host=<RDS_ADDRESS> --port=<DB_PORT> --username=root --dbname=<DB_NAME> --file=<BACKUP_DIR>/backup.sql
```

- **RDS_ADDRESS** indicates the IP address of the RDS DB instance.
- **DB_PORT** indicates the RDS DB instance port.
- **DB_NAME** indicates the name of the database to which data is to be imported. Ensure that the database exists.
- **BACKUP_DIR** indicates the directory where the **backup.sql** file is stored.

Enter the password for the RDS DB instance when prompted.

Example:

```
# psql --host=172.16.66.198 --port=5432 --username=root --dbname=my_db --file=backup.sql
```

Password for user root:

Step 4 View the import result.

```
my_db=> \l my_db
```

In this example, the database named **my_db** has been imported.

```
my_db=> \l my_db
List of databases
Name | Owner | Encoding | Collate | Ctype | Access privileges
-----+-----+-----+-----+-----+-----
my_db | root | UTF8 | en_US.UTF-8 | en_US.UTF-8 |
(1 row)
```

----End

6.2 Parameter Tuning

6.2.1 Suggestions on PostgreSQL Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect the stable running of databases. This section describes some important parameters for your reference. For details, visit the [PostgreSQL official website](#).

For details on how to modify PostgreSQL parameters on the console, see [Modifying Instance Parameters](#).

Sensitive Parameters

The following parameters can result in system security and stability issues if set improperly:

- The **search_path** parameter must be set to a schema sequence where schemas are separated by commas (.). Ensure that the schemas exist. Otherwise, the database performance will be affected.
- If you enable the parameter **log_duration**, SQL statements containing sensitive information may be recorded in logs. You are advised to disable this parameter.
- **log_min_duration_statement** specifies how many milliseconds a query has to run before it has to be logged. The unit is millisecond. Setting this parameter

to **0** means that all statements are recorded. Setting this parameter to **-1** means that no statement is recorded.

- The parameters **nls_timestamp_format** and **nls_date_format** control the input and output formats of timestamp and date. They are available only for PostgreSQL Enhanced Edition.
 - The **nls_timestamp_format** parameter defines the timestamp format. The **nls_date_format** defines the date format.
- The **temp_file_limit** parameter specifies the maximum amount of disk space (in KB) that a session can use for temporary files. It supports PostgreSQL 11 and 12 only. Changing this parameter value is a high-risk operation. Exercise caution when deciding to perform this operation.
 - If the parameter value exceeds the threshold, the DB instance will become unavailable.
 - If the parameter value is changed to a larger value for temporary use but is not changed to the original value after the use, the disk space will be continuously used to store temporary files. If the disk space is used up, services will be interrupted and the DB instance will become unavailable.

Performance Parameters

The following parameters can affect database performance:

- If **log_statement** is set to **ddl**, **mod**, or **all**, the operations for creating and deleting database users (including passwords and other sensitive information) are recorded. This operation affects database performance. Exercise caution when setting this parameter.
- Enabling the following parameters will affect the database performance: **log_hostname**, **log_duration**, **log_connections**, and **log_disconnections**. Exercise caution when enabling these parameters.

6.3 PostgreSQL Enhanced Edition

6.3.1 Introduction to PostgreSQL Enhanced Edition

PostgreSQL Enhanced Edition is developed from [PostgreSQL 11](#). It retains all functions of the PostgreSQL databases.

PostgreSQL Enhanced Edition supports system views, PL/SQL, data types, advanced functions, SQL syntax, and null value processing.

It helps you reduce costs for migration to the cloud and provides a comprehensive solution with high security, high availability, and high performance.

6.3.2 Functions

This section describes the built-in functions and advanced function packages added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 6-1 Built-in functions

Built-in Function	Description
add_months(date,integer)	Returns the date plus integer months. The return type is DATE.
appendchildxml(XMLType_instance, XPath_string, value_expr[, namespace_string])	Appends the value_expr node onto XPath_string specified by XMLType_instance. The namespace_string provides namespace information for the XPath_string.
asciistr(string)	Returns an ASCII version of the string in the database character set. Non-ASCII characters are not supported.
bin_to_num(expr_list)	Converts a binary string in expr_list to its equivalent decimal number. The return type is NUMBER.
bitand(number1,number2)	Returns the bitwise 'AND' for two supplied integers number1 and number2. The return type is BIT.
convert(char, dest_char_set[, source_char_set])	Converts char in the source_char_set to the dest_char_set encoding format. This function takes effect only on the server.
cosh(n)	Returns the hyperbolic cosine of argument n.
decode(expr,search1, result1[[,search2, result2],.....] [, default])	Compares expr to each search value (search1, search2, etc). If expr is equal to a search, then Oracle Database returns the corresponding result. If no match is found, then Oracle returns default. If default is omitted, then Oracle returns null.
empty_blob()	Returns an empty BLOB.
hextoraw(char)	Converts a hexadecimal string to a raw value.
instrb(string, substring[, position[, occurrence]])	Searches a string for a substring using characters and return the position in the string that is the first character of a specified occurrence of the substring. The functions vary in how they determine the position of the substring to return.
last_day(date)	Returns the date of the last day of the month that contains date.
lengthb(char)	Returns the length of char. Char can be any of the data types (CHAR, VARCHAR2, NCHAR, or NVARCHAR2), or types (such as integer) that can be implicitly converted into character strings.

Built-in Function	Description
listagg(measure_expr[, 'delimiter']) within group(order_by_clause) [over query_partition_clause]	Sorts the values of the column expression measure_expr in the query_partition_clause group based on the order_by_clause rule and aggregates them into one row. Values are separated by delimiter.
lnnvl(condition)	Returns a value of condition expression. The return type is BOOLEAN.
mod(n2, n1)	Returns the remainder of n2 divided by n1. Returns n2 if n1 is 0.
months_between(date1, date2)	Returns the number of months between dates date1 and date2. If date1 is earlier than date2, then the result is negative.
nanvl(n2, n1)	Returns n1 if the single- or double-precision floating point number input value n2 is NAN. If the input value n2 is not NAN, n2 is returned.
nchr(number)	Returns the character having the binary equivalent to number in the national character set.
new_time(date, timezone1, timezone2)	Returns the date and time in time zone timezone2 when date and time in time zone timezone1 are date. The return type is DATE.
next_day(date, char)	Returns the date of the first weekday named by char that is later than the date (including workdays, weekends, and holidays). The return type is DATE.
numtodsinterval(n, interval_unit)	Converts n to an INTERVAL DAY TO SECOND literal. The value for interval_unit specifies the unit of n and must resolve to 'DAY', 'HOUR', 'MINUTE', and 'SECOND'.
numtoyminterval(n, 'interval_unit')	Converts n to an INTERVAL YEAR TO MONTH literal. The value for interval_unit can be YEAR or MONTH.
nlssort(char[, nlsparam])	Sorts the char string according to the sorting character set specified by nlsparam. By default, char is used for sorting.
nls_upper(char[, nlsparam])	Converts all alphabetic characters in the character string char to uppercase letters based on the sort sequence specified by nlsparam. The character string type is CHAR, VARCHAR2, NCHAR, NVARCHAR2, CLOB, or NCLOB, and nlsparam is in the form of NLS_SORT = sort.

Built-in Function	Description
<code>nls_lower(char[, nlsparam])</code>	Converts all alphabetic characters in the character string <code>char</code> to lowercase letters based on the sort sequence specified by <code>nlsparam</code> . The character string type is <code>CHAR</code> , <code>VARCHAR2</code> , <code>NCHAR</code> , <code>NVARCHAR2</code> , <code>CLOB</code> , or <code>NCLOB</code> , and <code>nlsparam</code> is in the form of <code>NLS_SORT = sort</code> .
<code>nvl(expr1, expr2)</code>	Returns the first non-null value in <code>expr1</code> and <code>expr2</code> .
<code>rawtohex(raw)</code>	Converts <code>raw</code> to a character value containing its hexadecimal representation.
<code>regexp_count(source_char, pattern, position, match_param)</code>	Returns the number of times a pattern occurs in a source string starting from the position that indicates the <code>source_char</code> character where the database begins the search. The <code>match_param</code> parameter is a text literal that lets you change the default matching behavior of the function. For example, <code>match_param='i'</code> specifies case-insensitive matching.
<code>regexp_instr(source_char, pattern[, position[, occurrence[, return_opt[, match_param[, subexpr]]]])</code>	Extends the <code>INSTR</code> function and allows regular expression matching. The return type is <code>INTEGER</code>. <code>position</code>: indicates the start position of the search. <code>occurrence</code>: indicates the sequence number of the pattern in <code>source_char</code>. <code>return_opt</code>: - The value 0 indicates the start position of the return mode. - The value 1 indicates the end position of the return mode. <code>match_param</code>: indicates the control parameter of the regular expression, such as case sensitive. <code>subexpr</code>: indicates the group number of the regular expression group.
<code>regexp_like(source_char, pattern[, match_param])</code>	<code>source_char</code> is a character expression. Pattern is the regular expression. The <code>match_param</code> parameter is a text literal that lets you change the default matching behavior of the function.

Built-in Function	Description
<code>regexp_substr(source_char, pattern[, position[, occurrence[, match_param[, subexpr]]]])</code>	Matches the character string in the <code>source_char</code> string based on the regular expression. <code>source_char</code> is the text expression that is searched. Supports all character strings, including CHAR, VARCHAR2, NCHAR, or NVARCHAR2, or types (such as integer) that can be implicitly converted into character strings. <code>pattern</code> is the text expression to search for. <code>position</code> is a nonzero integer indicating the character of <code>source_char</code> where the function begins the search. <code>occurrence</code> is an integer indicating which occurrence of <code>pattern</code> the function should search for. <code>match_parameter</code> is a text expression that lets you change the default matching behavior of the function. <code>subexpr</code> is a nonnegative integer from 0 to 9 indicating which subexpression in <code>pattern</code> is to be returned by the function.
<code>raise_application_error(errnum, errmsg)</code>	Sends the error code <code>errnum</code> and error message <code>errmsg</code> to the client.
<code>remainder(n2, n1)</code>	Returns the remainder of <code>n2</code> divided by <code>n1</code> . The remainder function is similar to <code>mod</code> except that <code>mod</code> uses floor in its formula, whereas <code>remainder</code> uses ROUND. The return type is NUMERIC or double-precision floating-point number (determined by the input parameter type).
<code>round(n, precision)</code>	Returns <code>n</code> rounded to integer places to the right of the decimal point. The precision is the number of digits in a number.
<code>scn_to_timestamp(number)</code>	Returns the approximate timestamp associated with a system change number (SCN).
<code>sinh(n)</code>	Returns the hyperbolic sine of <code>n</code> . If <code>n</code> is BINARY_FLOAT, the return type is BINARY_DOUBLE. Otherwise, the return type is NUMERIC.

Built-in Function	Description
substr(char,position[,substring_length])	Returns a portion of string, beginning at a specified position in the string. The functions vary in how they calculate the length of the substring to return. If substring_length is not specified, the function returns all characters to the end of string.
substrb(char, position[, substring_length])	Returns a portion of char, beginning at a specified position in the string. The functions vary in how they calculate the length of the substring to return. If substring_length is not specified, the function returns all characters to the end of string.
sys_context(namespace, parameter)	Returns the value of parameter associated with the context namespace. The return type is VARCHAR2.
sys_guid()	Returns a globally unique identifier (RAW value).
sys_connect_by_path(column, char)	Is valid only in CONNECT BY queries and returns the path of a column value from root to node.
tanh(n)	Returns the hyperbolic tangent of argument n.
to_blob(char)	Converts char strings to BLOB values. Char can be any of the data types (CHAR, VARCHAR2, NCHAR, or NVARCHAR2), or types (such as integer) that can be implicitly converted into character strings.
to_binary_float(expr)	Converts expr to the single-precision float type.
to_binary_double(expr)	Converts expr to the double-precision float type.
to_clob(char)	Converts char to the CLOB data type.
to_char(char)	Supports char types: char, character, and varchar.
to_date(char[,fmt])	Converts char of the CHAR, VARCHAR2, NCHAR, NVARCHAR2, or TIMESTAMP data type to a value of the DATE data type according to the fmt format. If fmt is omitted, char must use the default format of the DATE data type.

Built-in Function	Description
to_dsinterval('sql_format' 'ds_iso_format')	Converts the time character string of the SQL standard (such as '100 00:00:00') or ISO standard (such as 'P100DT05H') to the INTERVAL DAY TO SECOND data type.
to_multi_byte(char)	Converts a single-byte character char into a multi-byte character.
to_number(expr)	Converts expr to a value of NUMBER data type.
to_number(expr, fmt, 'nlsparam')	Converts expr to a value of NUMBER data type in the format specified by fmt. The nlsparam is an international language parameter and supports the following parameters: NLS_NUMERIC_CHARACTERS, NLS_CURRENCY, and NLS_ISO_CURRENCY.
to_timestamp(char[,fmt])	Converts char of the CHAR, VARCHAR2, NCHAR, NVARCHAR2, or TIMESTAMP data type to a value of the timestamp data type according to the fmt format. If fmt is omitted, char must use the default format of the TIMESTAMP data type.
to_single_byte(char)	Converts multibyte characters to their corresponding single-byte characters.
to_ymininterval('sql_format' 'ym_iso_format')	Converts the time character string of the SQL standard (such as '01-02') or ISO standard (such as 'P1Y2M') to the INTERVAL MONTH TO YEAR data type.
timestamp_to_scn(timestamp)	Returns the approximate system change number (SCN) associated with a timestamp.
trunc(date[, fmt])	Truncates date according to the date format specified by fmt. The return type is DATE. If fmt is omitted, the default date format is 'DDD'.
tz_offset({time_zone_name '{+ -}hh:mi'})	Returns the specified time zone offset. The return type is VARCHAR2. The parameter is a character string in the time_zone_name or '{+ -}hh:mi' format.
value(correlation_variable)	Returns the recorded row associated with correlation_variable in object table mode. The return type is the object table associated with correlation_variable.

Table 6-2 Advanced function packages

Advanced Function Package	Description
DBMS_OUTPUT.PUT(item)	Places the item string in the local buffer. Item indicates all types that can be converted into character strings.
DBMS_OUTPUT.PUT_LINE(item)	Places the item string in the local buffer and outputs all the content in the local buffer. Item indicates all types that can be converted into character strings.
DBMS_RANDOM.SEED(val)	Val is the seed number used to generate a random number. It can be a character string or a digit.
DBMS_RANDOM.VALUE([low,high])	Returns a 16-digit random number between low and high. If the range of low and high is not specified, the default value range is 0-1.
dbms_lob.getlength(lob_loc {clob blob})	Returns the LOB length specified by lob_loc.
dbms_lob.read(lob_loc, amount, offset, buffer)	Returns the specified amount into the buffer parameter, starting from an absolute offset from the beginning of the LOB.
dbms_lob.write(lob_loc, amount, offset, buffer)	Writes the buffer content to the large object lob_loc buffer (the referenced large object is not affected) starting at offset. The amount represents the size.
utl_raw.cast_to_raw(char)	Converts char of the VARCHAR2 data type to RAW. The return type is RAW.
utl_raw.length(raw)	Returns the length of the raw data type. The return type is NUMBER.
utl_raw.cast_from_binary_integer(n, endianness)	Converts the integer n to the RAW type based on the memory alignment mode specified by endianness. The values of endianness are as follows: • 1: big_endian • 2: little_endian • 3: machine_endian

6.3.3 System Views

This section describes the system views added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 6-3 System views

Super Administrator	DBA	USER
ALL_ALL_TABLES	DBA_ALL_TABLES	-
ALL_COL_COMMENTS	-	USER_COL_COMMENTS
-	DBA_DATA_FILES	-
ALL_DIRECTORIES	DBA_DIRECTORIES	-
ALL_INDEXES	DBA_INDEXES	USER_INDEXES
ALL_JOBS	DBA_JOBS	USER_JOBS
ALL_OBJECTS	-	USER_OBJECTS
ALL_PROCEDURES	DBA_PROCEDURES	USER_PROCEDURES
ALL_SOURCE	DBA_SOURCE	USER_SOURCE
ALL_SEQUENCES	DBA_SEQUENCES	USER_SEQUENCES
ALL_TABLES	DBA_TABLES	USER_TABLES
-	DBA_TABLESPACES	USER_TABLESPACE
ALL_TAB_COLUMNS	DBA_TAB_COLUMNS	USER_TAB_COLUMNS
-	DBA_TRIGGERS	USER_TRIGGERS
ALL_USERS	DBA_USERS	-
ALL_VIEWS	DBA_VIEWS	USER_VIEWS
ALL_IND_COLUMNS	DBA_IND_COLUMNS	USER_IND_COLUMNS
ALL_TAB_PARTITIONS	DBA_TAB_PARTITIONS	USER_TAB_PARTITIONS
ALL_PART_TABLES	DBA_PART_TABLES	USER_PART_TABLES
ALL_PART_KEY_COLUMNS	DBA_PART_KEY_COLUMNS	USER_PART_KEY_COLUMNS
ALL_PART_INDEXES	DBA_PART_INDEXES	USER_PART_INDEXES
ALL_TAB_SUBPARTITIONS	DBA_TAB_SUBPARTITIONS	USER_TAB_SUBPARTITIONS
ALL_SUBPART_KEY_COLUMNS	DBA_SUBPART_KEY_COLUMNS	USER_SUBPART_KEY_COLUMNS

Table 6-4 Common view

View Name	Description
V\$SESSION	Displays information related to the current session, such as SID and username.
NLS_SESSION_PARAMETERS	Shows the NLS parameters and values of the current session.
V\$SESSION_LONGOPS	Displays the status of database operations that have been running for more than 6 seconds.

6.3.4 Data Types

This section describes the data types added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 6-5 Data types

Name	Data Type
Variable-length character type	VARCHAR2 and NVARCHAR2
Decimal floating point type	DECIMAL
Double precision binary floating point type	BINARY_DOUBLE
Binary data type	RAW
Binary large object type	BLOB
Character large object type	CLOB
National character large object	NCLOB
Number type	NUMBER
Variable-length character type	NVARCHAR
Unicode character data type	NCHAR
32-bit floating point data type	BINARY_FLOAT
Long integer	LONG
XML data type	XMLType
Timestamp with local time zone	TIMESTAMP WITH LOCAL TIME ZONE
PL/SQL integer numeric data	BINARY_INTEGER
PL/SQL integer numeric data	PLS_INTEGER

6.3.5 Implicit Type Conversion

This section describes the implicit type conversion added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

- Conversion between the fixed-length character string type CHARACTER and NUMERIC, INT4, INT8, FLOAT4, and FLOAT8
- Conversion between variable-length character string type VARCHAR and NUMERIC, INT4, INT8, FLOAT4, and FLOAT8
- Conversion between the text type TEXT and NUMERIC, INT2, INT4, INT8, FLOAT4, and FLOAT8
- Conversion from short int INT2 to CHARACTER and VARCHAR
- Conversion between binary large object BLOB and binary RAW

6.3.6 Predefined Parameters

This section describes the predefined parameters added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 6-6 Predefined parameters

Predefined Parameter	Description
NLS_DATE_FORMAT	Defines the date format.
NLS_DATE_LANGUAGE	Defines the date language.
NLS_DUAL_CURRENCY	Defines the local currency symbols for the currencies of specific territories or countries.
NLS_CURRENCY	Defines the currency symbol.
NLS_TIME_FORMAT	Defines the time format without the time zone.
NLS_TIME_TZ_FORMAT	Defines the time format without the time zone.
NLS_TIMESTAMP_FORMAT	Defines the timestamp format without the time zone.
NLS_TIMESTAMP_TZ_FORMAT	Defines the timestamp format with the time zone.
NLS_NUMERIC_CHARACTERS	Defines the characters used as group separator and decimal character.
NLS_ISO_CURRENCY	Defines the ISO currency symbols for the currencies of specific territories or countries.
NLS_TERRITORY	Resets the values of NLS_CURRENCY, NLS_ISO_CURRENCY, and NLS_NUMERIC_CHARACTERS based on the regional currency and displayed number format.
NLS_LANGUAGE	Defines the default language of the database.

Predefined Parameter	Description
NLS_LENGTH_SEMANTICS	Defines the default length semantics of character strings. The value is BYTE or CHAR.
NLS_SORT	Defines the collating sequence for local characters.
NLS_COMP	Defines the collation behavior of database sessions.

6.3.7 Macro Variables

This section describes the macro variables added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

- SYSDATE: indicates the current system time.
- SYSTIMESTAMP: indicates the current system timestamp.
- DBTIMEZONE: indicates the current database time zone.
- SESSIONTIMEZONE: indicates the current session time zone.
- ROWNUM: indicates the tuple number in the query results.

6.3.8 Operators

This section describes the following operators added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition:

- Arithmetic operator: MINUS
- Equality operator: ^=

NOTE

Blank characters (including spaces and tab keys) are allowed in the following operators: inequality (^=, <>, and !=), greater than or equal to (>=), and less than or equal to (<=).

6.3.9 Syntax

This section describes the syntax added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition. The following are supported:

- CREATE SEQUENCE
- CREATE/ALTER DATABASE
- CREATE/ALTER VIEW
- CREATE TABLE
- CREATE TABLESPACE
- CLUSTER
- FORALL
- CREATE/DROP DIRECTORY
- ALTER TABLE ADD CONSTRAINT USING INDEX
- Table names or table aliases for target columns in the INSERT INTO statement

- ROWNUM in non-partitioned tables
- CREATE INDEX ON COLUMN_EXPR
- ALTER TABLE MODIFY
- Specifying length units for VARCHAR and CHARACTER data types
- TYPE/NAME/VERSION/VALUE/INTERVAL alias
- Stored procedures
- DATE
- HASH-, RANGE-, and LIST-partitioned table creation
- MERGE
MERGE [HINT] INTO table_name USING ({subquery | table_name | view_name}) alias ON (condition) merge_update_clause merge_insert_clause;
- Time interval operation:
INTERVAL YEAR TO MONTH,INTERVAL DAY (I) TO SECOND (P);
- CREATE TRIGGER with BODY:
CREATE TRIGGER name... {DECLARE ... BEGIN | BEGIN} body END;
- Stored procedure cursor syntax:
CURSOR cursor_name [parameter_list] IS select_statement, TYPE type_name IS REF CURSOR;
- Stored procedure cursor variables:
SQL%ISOPEN,SQL%FOUND,SQL%NOTFOUND,SQL%ROWCOUNT,cursor%ISOPEN,cursor%FOUND,cursor%NOTFOUND,cursor%ROWCOUNT;
- Scheduled task advanced package:
DBMS_JOB.SUBMIT,DBMS_JOB.ISUBMIT,DBMS_JOB.REMOVE,DBMS_JOB.BROKEN,DBMS_JOB.CHANGE,DBMS_JOB.WHAT,DBMS_JOB.NEXT_DATE,DBMS_JOB.INTERVAL;
- CREATE USER:
{DEFAULT COLLATION | DEFAULT TABLESPACE | [LOCAL] TEMPORARY TABLESPACE} Clause;
- Session attribute modification:
ALTER SESSION SET param_name = value;
- Anonymous blocks
- Cross-mode access to stored procedures
- SQLCODE built-in variables in stored procedures
- Enhanced syntax compatibility in stored procedures: stored procedure names can be used as end tags; FOR VAR IN SELECT-CLAUSE is supported; end tags can be specified for LOOP statements; default value of IN can be specified.
- Subqueries with no alias specified
- NOCYCLE in CREATE SEQUENCE
- Replacing PASSWORD with IDENTIFIED BY in CREATE/ALTER USER
- Specifying table names or alias in UPDATE SET
- (columnname)=(value) in UPDATE SET
- ALTER TABLE support for MODIFY NOT NULL and ENABLE
- Null character string equivalent to NULL
- sequencCURRVAL and sequencNEXTVAL
- Creating users and schemas with same names at the same time
- Deleting FROM from the table record syntax

- XML data type pseudo column COLUMN_VALUE
- OUTER JOIN (+)
- Operators between the data types INTERVAL and number: +, -, >, <, >=, <=, and <>
- Partition table DML operations: SELECT, INSERT, UPDATE, and DELETE
- Composite partitioning of partition tables
- Expressions used as partition boundaries
- Trigger DDL: schema
- Time format: IYY
- CREATE/ALTER MATERIALIZED VIEW
- CREATE TYPE
- CREATE PROFILE
- Enable/disable syntax for column constraints
- Tablespace options specified by partitioned tables
- DROP TABLE tablename [CASCADE CONSTRAINTS] [PURGE]
- Stored procedure dynamic SQL syntax EXECUTE IMMEDIATE. The current edition does not support dynamic execution of anonymous blocks with DECLARE.
- FUNCTION definition
- CONNECT BY queries: LEVEL, CONNECT_BY_ROOT, and CONNECT_BY_ISLEAF pseudo columns; sys_connect_by_path, CONNECT_BY_ROOT, and ORDER SIBLINGS
- TIME data type precision
- Supported for virtual columns: column_name datatype [GENERATED ALWAYS] AS (expression) [VIRTUAL]
- One-dimensional array definition: CREATE OR REPLACE TYPE array_name AS VARRAY (len) OF typename
- One-dimensional array: array_name.extend, array_name.count, array_name.first, array_name.last
- ROLLUP, CUBE, and GROUPING SETS Group By supported for grouping_id([expr1[, expr2[, ...exprn]]]) and group_id()
- Sorting query statements returned by non-grouping fields: SELECT SUM(colname) FROM tbl ORDER BY colname

6.3.10 Enhanced Functions

This section describes the built-in plugin PG_PERMISSIONS, table partitioning optimization, and parallel query optimization added to PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table Partitioning Optimization

- Supported for partitions based on hash keys
- Supported for partition tables PRIMARY KEY, FOREIGN KEY, INDEXES, and TRIGGERS
- Supported for creation of "default" partitions

- Supported for INSERT ON CONFLICT
- Supported for automatic movement of records affected by UPDATE to the correct partition
- Supported for intelligent parallel JOIN (enable_partitionwise_join must be enabled)

Parallel Query Optimization

PostgreSQL Enhanced Edition improves the parallel query performance, as well as the performance of parallel sequential scan, hash connection, and partition data scan. If underlying queries are unparallelized, PostgreSQL can run UNION in parallel for SELECT queries. The following functions are supported:

- Supported for parallel processing of creating B-tree indexes
- CREATE TABLE ... AS, CREATE MATERIALIZED VIEW, and some queries supported for parallel UNION
- Supported for parallel hash join

6.3.11 Security Hardening

This section describes the newly added security hardening of PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

- Sensitive information such as passwords cannot be printed in logs.
- User group rights management is optimized: Only group members can authorize the rights of the user group.
- The SHA256 encryption is used to authenticate the password for connection between the client and server.
- The security vulnerability CVE-2018-16850 is resolved.
- The permission control for viewing pg_stat_statements statistics is optimized.

6.3.12 Other Functions

This section describes the newly added functions of PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

- Supports the pseudo table DUAL.
`SELECT * FROM DUAL;`
- Adds the **sql_format** parameter with the default value **postgresql**.
- Optimizes the path management in the CREATE TABLESPACE cloud application scenario, making users being unaware of the actual path.
- Introduces the pg_hint_plan plugin to enhance query plans and optimize maintenance methods.
- Provides the data file integrity check tool: pg_verify_checksums.
- Supports scheduled task management.
- Enhances the pg_stat_statements plugin.

6.4 PostgreSQL Kernel Functions

6.4.1 RDS for PostgreSQL Kernel Release History

The following table lists the release dates of the community versions supported by RDS for PostgreSQL.

Table 6-7 Community versions supported by RDS for PostgreSQL

Release Date	Community Versions
2022-08	• 14.4 • 13.7 • 12.11 • 11.16 • 10.21
2022-05	• 13.6 • 12.10 • 11.15 • 10.20 • 9.6.24
2021-05	• 13.2 • 12.6 • 11.11 • 10.16 • 9.5.25
2020-05	• 12.2 • 11.5 • 10.10 • 9.6.15 • 9.5.19

6.5 Instance Lifecycle

6.5.1 Creating a Same DB Instance as an Existing DB Instance

Scenarios

This section describes how to quickly create a DB instance with the same configurations as the selected one.

 NOTE

- You can create DB instances with the same configurations numerous times.
- This function is unavailable for read replicas.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Create Same DB Instance** in the **Operation** column.

Step 5 On the displayed page, the configurations are the same as those of the selected DB instance. You can change them as required. Then, click **Next**.

For details about PostgreSQL DB instance configurations, see section [Step 1: Create a DB Instance](#).

Step 6 Confirm the instance specifications.

Step 7 Refresh the DB instance list and view the status of the DB instance. If the status is **Available**, it has been created successfully.

You can manage the DB instance on the **Instance Management** page.

----End

6.5.2 Rebooting DB Instances or Read Replicas

Scenarios

You may need to reboot a DB instance during maintenance. For example, after you modify some parameters, a reboot is required for the modifications to take effect. You can reboot a primary DB instance or a read replica on the management console.

Constraints

- If the DB instance status is **Abnormal**, the reboot may fail.
- Rebooting a DB instance will cause service interruptions. During this period, the DB instance status is **Rebooting**.
- Rebooting DB instances will cause instance unavailability and clear cached memory. To prevent traffic congestion during peak hours, you are advised to reboot DB instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance, or click  in the front of a DB instance and then locate the target read replica. Choose **More > Reboot** in the **Operation** column.
- Alternatively, click the target DB instance on the **Instance Management** page to go to the **Basic Information** page. In the upper right corner, click **Reboot**.
- For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.
- Step 5** In the displayed dialog box, select a scheduled time, and click **Yes**.
- Step 6** Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has rebooted successfully.

----End

6.5.3 Selecting Displayed Items

Scenarios

You can customize which instance items are displayed on the **Instance Management** page.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click  to edit columns displayed in the DB instance list.
- The following items are displayed by default: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, and operation. These default items cannot be deselected.
 - In a single project, you can select up to 9 items: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, creation time, database port, storage type, and operation.
 - For multiple projects, if you have enabled the ProjectMan permissions, you can select up to 9 items: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, creation time, database port, storage type, and operation.

----End

6.5.4 Exporting DB Instance Information

Scenarios

You can export information about all or selected DB instances to view and analyze DB instance information.

Exporting Information About All DB Instances

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click  in the upper right corner of the page. By default, information about all DB instances are exported. In the displayed dialog box, you can select the items to be exported and click **Export**.
- Step 5** Find a .csv file locally after the export task is completed.
- End

Exporting Information About Selected DB Instances

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, filter DB instances by DB engine, DB instance name, DB instance ID, or floating IP address, or select the DB instances to be exported, and click  in the upper right corner of the page. In the displayed dialog box, select the items to be exported and click **Export**.
- Step 5** Find a .csv file locally after the export task is completed.
- End

6.5.5 Deleting a Pay-per-Use DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas as required on the **Instance Management** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.

- If you delete a DB instance, its automated backups are also deleted and you are no longer billed for them. Manual backups, however, are still retained and will generate additional costs.

NOTICE

- If you delete a primary DB instance, its standby DB instance and read replicas (if any) are also deleted automatically. Exercise caution when performing this operation.
 - Deleted DB instances cannot be recovered and resources are released. Exercise caution when performing this operation. If you want to retain data, **create a manual backup** first before deleting the DB instance.
 - You can use a manual backup to restore a DB instance. For details, see [Restoring from Backup Files to DB Instances](#).
-

Deleting a DB Instance

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.
- Step 5** In the displayed dialog box, click **Yes**.
- Step 6** Refresh the DB instance list later to confirm that the deletion was successful.
- End

Deleting a Read Replica

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and click . All the read replicas created for the DB instance are displayed.
- Step 5** Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.
- Step 6** In the displayed dialog box, click **Yes**.
- Step 7** Refresh the DB instance list later to check that the deletion is successful.
- End

6.5.6 Recycling a DB Instance

Scenarios

Deleted DB instances can be moved to the recycle bin. You can rebuild DB instances from the recycle bin to restore data. The DB engine, DB engine version, and storage type of the new DB instance are the same as those of the original DB instance. Other parameters can be reconfigured. DB instances that were deleted up to 7 days ago can be restored.

Constraints

- Read replicas cannot be moved to the recycle bin.
- The recycle bin is enabled by default and cannot be disabled.

Modifying Recycling Policy

NOTICE

A new recycling policy only applies to DB instances that were put in the recycle bin after the new policy was put into effect. For DB instances that were in the recycle bin before the modification, the original recycling policy takes effect.

- Step 1** Log in to the management console.
 - Step 2** Click  in the upper left corner and select a region and a project.
 - Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
 - Step 4** In the navigation pane on the left, choose **Recycling Management**.
 - Step 5** On the **Recycling Management** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances from 1 day to 7 days.
 - Step 6** Then, click **OK**.
- End

Rebuilding a DB Instance

You can rebuild the primary DB instances in the recycle bin within the retention period.

- Step 1** In the navigation pane on the left, choose **Recycling Management**.
 - Step 2** On the **Recycling Management** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.
 - Step 3** On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see [Restoring from Backup Files to DB Instances](#).
- End

6.6 Instance Modifications

6.6.1 Changing a DB Instance Name

Scenarios

You can change the name of a primary DB instance or read replica.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click  next to it to edit the DB instance name. Then, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Instance Name** field in the **DB Information** area, click  to edit the DB instance name.

The DB instance name must start with a letter and consist of 4 to 64 characters. Only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_) are allowed.

- To submit the change, click .
- To cancel the change, click .

Step 5 View the results on the **Basic Information** page.

----End

6.6.2 Changing the Failover Priority

Scenarios

You can configure the failover priority for reliability or for availability, depending on your service requirements.

- **Reliability:** This option is selected by default. Data consistency is preferentially ensured during a primary/standby failover. This is recommended for users whose highest priority is data consistency.
- **Availability:** Database availability is preferentially ensured during a primary/standby failover. This is recommended for users who require their databases to provide uninterrupted online services.

Procedure

- Step 1** Log in to the management console.
 - Step 2** Click  in the upper left corner and select a region and a project.
 - Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
 - Step 4** On the **Instance Management** page, click the target primary/standby DB instances.
 - Step 5** In the **DB Information** area on the displayed **Basic Information** page, click **Change** in the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.
 - Step 6** View the results on the **Basic Information** page.
- End

6.6.3 Changing a DB Instance Class

Scenarios

You can change the instance class (CPU or memory) of a DB instance as required. If the status of a DB instance changes from **Changing instance class** to **Available**, the change is successful.

Constraints

- A DB instance cannot be deleted when its instance class is being changed.
- If the primary DB instance has a read replica, the new DB instance class must be less than or equal to the read replica class. When changing the read replica class, ensure that the selected class is greater than or equal to the current primary instance class.
- After you change instance classes, the DB instances will be rebooted and service will be interrupted. Therefore, you are advised to change instance classes during off-peak hours.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and choose **More > Change Instance Class** in the **Operation** column.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click **Change** in the **Instance Class** field.
- Step 5** On the displayed page, specify the new instance class and click **Next**.

Step 6 View the DB instance class change result.

Changing the DB instance class takes 5–15 minutes. During this period, the status of the DB instance on the **Instance Management** page is **Changing instance class**. After a few minutes, click the DB instance and view the instance class on the displayed **Basic Information** page to check that the change is successful.

NOTICE

After the CPU or memory of a PostgreSQL DB instance is changed, the system will change the values of the following parameters accordingly:

- **shared_buffers**
- **max_connections**
- **maintenance_work_mem**
- **effective_cache_size**

----End

6.6.4 Scaling up Storage Space

Scenarios

If the original storage space is insufficient as your services grow, scale up storage space of your DB instance.

The DB instance needs to preserve at least 15% of its capacity to work properly. The new minimum storage space required to make this instance available has been automatically calculated for you. You are advised to set alarm rules for the storage space usage by referring to [Setting Alarm Rules](#).

For details about the causes and solutions of insufficient storage space, see [What Should I Do If My Data Exceeds the Available Storage of an RDS DB Instance?](#)

RDS allows you to scale up storage space of DB instances but you cannot change the storage type. During the scale-up period, services are not interrupted.

Constraints

- The maximum allowed storage is 4,000 GB. There is no limit on the number of scale-ups.
- For primary/standby DB instances, scaling up the primary DB instance will cause the standby DB instance to also be scaled up accordingly.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up.

Scaling up a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the target DB instance to enter the **Basic Information** page. In the **Storage Space** area, click **Scale**.

Step 5 On the displayed page, specify the new storage space and click **Next**.

The minimum start value of each scaling is 10 GB. A DB instance can be scaled up only by a multiple of 10 GB. The allowed maximum storage space is 4,000 GB.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.
- If your settings are correct, click **Submit**.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During the time period, the status of the DB instance on the **Instance Management** page will be **Scaling up**. Click the DB instance and view the utilization on the displayed **Basic Information** page to verify that the scale-up is successful.

----End

Scaling up a Read Replica

Scaling up the storage space of a read replica does not affect that of the primary DB instance. Therefore, you can separately scale read replicas to meet service requirements. New storage space of read replicas after scaling up must be greater than or equal to that of the primary DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click  in front of it. Locate a read replica to be scaled and choose **More > Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the target DB instance to enter the **Basic Information** page. In the **Storage Space** area, click **Scale**.

Step 5 On the displayed page, specify the new storage space and click **Next**.

The minimum start value of each scaling is 10 GB. A read replica can be scaled up only by a multiple of 10 GB. The allowed maximum storage space is 4,000 GB.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.
- If your settings are correct, click **Submit**.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time, the status of the read replica on the **Instance Management** page will be **Scaling up**. Click the read replica and view the utilization on the displayed **Basic Information** page to verify that the scale-up is successful.

-----End

6.6.5 Changing a DB Instance Type from Single to Primary/Standby

Scenarios

- RDS enables you to change single DB instances to primary/standby DB instances to improve instance reliability.
- Primary/standby DB instances support automatic failover. If the primary DB instance fails, the standby DB instance takes over services quickly. You are advised to deploy primary and standby DB instances in different AZs for high availability and disaster recovery.
- Anti-affinity deployment is supported for primary/standby DB instances to prevent the entire instance unavailability due to the failure of a single host.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate a single DB instance and choose **More > Change Type to Primary/Standby** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  on the left to change the instance type from single to primary/standby.

Step 5 Select a standby AZ. Other configurations are the same as those of the primary DB instance by default. Confirm the configurations and click **Submit**.

Step 6 After a single DB instance is changed to primary/standby instance, you can view and manage it on the **Instance Management** page.

- The DB instance is in the **Changing type to primary/standby** status. You can view the progress on the **Task Center** page. For details, see [Task Center](#).
- In the upper right corner of the DB instance list, click  to refresh the list. After the DB instance type is changed to primary/standby, the instance status

will change to **Available** and the instance type will change to **Primary/Standby**.

----End

6.6.6 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create primary/standby DB instances, RDS will create a primary DB instance and a synchronous standby DB instance in the same region. You can access the primary DB instance only. The standby instance serves as a backup. You can manually promote the standby DB instance to the new primary instance for failover support.

Constraints

- A DB instance is running properly.
- The replication between the primary and standby instances is normal.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary/standby DB instance.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** in the **DB Instance Type** field.

Alternatively, click  in the DB instance topology on the **Basic Information** page to perform a primary/standby switchover.

NOTICE

A primary/standby switchover may cause service interruptions for some seconds or minutes (depending on the replication delay). If the replication delay is too long, a small amount of data may be lost. To prevent traffic congestion, you are advised to perform a switchover during off-peak hours.

Step 6 In the **Switch Primary/Standby Instances** dialog box, click **Yes** to switch between the primary and standby DB instances.

If the replication status is **Available** and the replication delay is greater than 300s, the primary/standby switchover task cannot be delivered.

Step 7 After a switchover is successful, you can view and manage the DB instance on the **Instance Management** page.

- During the switchover process, the DB instance status is **Switchover in progress**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is successful, the DB instance status will become **Available**.

----End

6.6.7 Migrating a Standby DB Instance

Scenarios

You can migrate a standby DB instance to another AZ in the same region as the original AZ.

NOTE

- DDL operations and scheduled events will be suspended during migration. To prevent service interruption, perform the migration during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Migrate Standby DB Instance** in the **Operation** column.

Step 5 On the displayed page, select a target AZ and click **Submit**.

Step 6 After the migration is complete, you can view and manage the DB instance on the **Instance Management** page.

- During the migration process, the DB instance status is **Migrating standby DB instance**. You can view the progress on the **Task Center** page. For details, see section [Task Center](#).
- In the upper right corner of the DB instance list, click  to refresh the list. After the migration is complete, the DB instance status will become **Available**.
- In the **DB Information** on the **Basic Information** page, you can view the AZ hosting the standby DB instance.

----End

6.7 Read Replicas

6.7.1 Introducing Read Replicas

Introduction

RDS for PostgreSQL supports read replicas.

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To expand the DB instance read ability to offload read pressure on the database, you can create one or more read replicas in a region. These read replicas can process a large number of read requests and increase application throughput. You need to separately configure connection addresses of the primary DB instance and each read replica on your applications so that all read requests can be sent to read replicas and write requests to the primary DB instance.

A read replica uses a single-node architecture (without a slave node). Changes to the primary DB instance are also automatically synchronized to all associated read replicas through the native PostgreSQL replication function. The synchronization is not affected by network latency. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Functions

- Read replica specifications can be different from primary DB instance specifications. It is recommended that the read replica specifications be greater than or equal to the primary DB instance specifications to prevent long delay and high load.
- You do not need to maintain separate database accounts or databases. They are synchronized from the primary DB instance.
- Read replicas support system performance monitoring. RDS provides up to 20 monitoring metrics, including storage space, IOPS, the number of database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on DB instances.
- Read replicas do not support automated backups or manual backups.
- Read replicas do not support restoration from backups to new, existing, or original read replicas.
- Data cannot be migrated to read replicas.
- Read replicas do not support database creation and deletion.
- Read replicas do not support database account creation.
- The specifications of read replicas must be greater than or equal to the specifications of the current primary DB instance.

Constraints

- A maximum of five read replicas can be created for each primary DB instance.

Creating and Managing a Read Replica

- [Creating a Read Replica](#)
- [Managing a Read Replica](#)

6.7.2 Creating a Read Replica

Scenarios

Read replicas are used to enhance the read capabilities and reduce the load on primary DB instances.

You can create read replicas as needed.

NOTE

A maximum of five read replicas can be created for a primary DB instance.

The specifications of read replicas must be greater than or equal to the specifications of the current primary DB instance.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click **Create Read Replica** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  under the primary DB instance to create read replicas.

Step 5 On the displayed page, configure information about the DB instance and click **Next**.

Table 6-8 Basic information

Parameter	Description
Region	By default, read replicas are in the same region as the primary DB instance.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
DB Engine	Same as the DB engine of the primary DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of the primary DB instance by default and cannot be changed.

Parameter	Description
AZ	RDS allows you to deploy primary DB instances and read replicas in a single AZ or across AZs to improve reliability.

Table 6-9 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS. For details about instance classes, see section DB Instance Classes. After a DB instance is created, you can change its CPU and memory. For details, see section Changing a DB Instance Class.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • High I/O: supports a maximum throughput of 150 MB/s. • Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Storage Space	Contains the file system overhead required for inode, reserved block, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.

Table 6-10 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.
Subnet	Same as the primary DB instance's subnet. A floating IP address is automatically assigned when you create a read replica. You can also enter an unused floating IP address in the subnet CIDR block. After the read replica is created, you can change the floating IP address.
Security Group	Same as the primary DB instance's VPC.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

Step 7 After a read replica has been created, you can view and manage it on the **Instance Management** page by clicking  on the left of the DB instance to which it belongs.

Alternatively, click the target DB instance. In the DB instance topology, click the target read replica. You can view and manage it in the displayed pane.

----End

Follow-up Operations

Managing a Read Replica

6.7.3 Managing a Read Replica

Entering the Management Interface Through a Read Replica

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 In the DB instance list, click  to expand the DB instance details and click the target read replica to go to the **Basic Information** page.

----End

Entering the Management Interface Through a Primary DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.

Step 5 In the DB instance topology, click the target read replica. You can view and manage it in the displayed pane.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 In the DB instance list, click  in front of a DB instance, locate the read replica to be deleted, and choose **More > Delete** in the **Operation** column.

----End

6.8 Backups and Restorations

6.8.1 Working with Backups

RDS supports backups and restorations to ensure data reliability.

Automated Backups

Automated backups are created during the backup time window for your DB instances. RDS saves automated backups based on a retention period you specify. If necessary, you can restore to any point in time during your backup retention period. For details, see [Configuring an Automated Backup Policy](#).

Manual Backups

Manual backups are user-initiated full backups of DB instances. They are retained until you delete them manually. For details, see [Creating a Manual Backup](#).

Downloading a Backup File

You can download a full or an incremental backup file for local data backup or restoration. For details, see [Downloading a Full Backup File](#) and [Downloading an Incremental Backup File](#).

6.8.2 Configuring an Automated Backup Policy

Scenarios

When you create a DB instance, an automated backup policy is enabled by default. For security purposes, the automated backup policy cannot be disabled. After the DB instance is created, you can customize the automated backup policy as required and then RDS backs up data based on the automated backup policy you configure.

RDS backs up data at the DB instance level, rather than the database level. If a database is faulty or data is damaged, you can restore it from backups. Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects database read and write performance, the automated backup time window should be set to off-peak hours.

The automated backup policy is enabled by default as follows:

- Retention period: 7 days. Backup files that exceed the retention period will be deleted and cannot be restored.

- Time window: An hour within 24 hours, such as 01:00-02:00 or 12:00-13:00. The backup time is configured based on UTC time and is adjusted for daylight saving time, which changes at different times depending on the time zone.
- Backup cycle: Each day of the week

Modifying an Automated Backup Policy

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Backups & Restorations** page, click **Modify Backup Policy**.

- **Retention Period** is the number of days that your automated backups are saved for. Increasing the retention period will improve data reliability.
- If you shorten the retention period, the new backup policy takes effect for all backup files. The backup files that have expired will be deleted.
- The backup retention period is the number of days you want automated full and incremental backups of your DB instance to be saved for. It ranges from 1–732 days. The backup time window is one hour. You are advised to select an off-peak time window for automated backups. By default, each day of the week is selected for **Backup Cycle** and you can change it. At least one day must be selected.

Step 6 Click **OK**.

----End

6.8.3 Set a Cross-Region Backup Policy

Scenarios

RDS can store backup files in the storage space that is in a different region from the DB instance for disaster recovery. If a DB instance in a region is faulty, you can use the backup files in another region to restore data to a new DB instance.

After you enable cross-region backup, the backup files are automatically stored in the region you specify. On the **Backup Management** page of the RDS console, you can click **View Backup** in the **Operation** column and manage cross-region backup files.

Enabling or Modifying a Cross-Region Backup Policy

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Set Cross-Region Backup Policy**.

- If you enable **Cross-Region Full Backup**, automated full backup files of the DB instance are stored in OBS in the region you specify.
- If you enable **Cross-Region Log Backup**, full and incremental backup files of the DB instance are stored in OBS in the region you specify.
- Cross-region backup files can be retained from 1 to 1,825 days.
- After the cross-region log backup function is enabled, you can restore a DB instance to a specified time point only after the next automated full backup replication is complete. The specified time point must be later than the time when the automated full backup is complete.

Step 6 Click **OK**.

Step 7 Choose **Backup Management** in the left navigation pane and click **Cross-Region Backups** to manage cross-region backup files.

- To modify the cross-region backup policy, click **Set Cross-Region Backup** in the **Operation** column.
- To view generated cross-region backup files, click **View Cross-Region Backup** in the **Operation** column. If a DB instance fails, you can use the cross-region backup files to restore data to a new DB instance.

----End

Disabling a Cross-Region Backup Policy

Step 1 On the **Backup Management** page, click the **Cross-Region Backups** tab.

Step 2 Locate a target DB instance and click **Set Cross-Region Backup** in the **Operation** column. On the displayed page, disable the cross-region backup policy.

Step 3 Click **OK**.

----End

6.8.4 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups of a running primary DB instance. You can use these backups to restore data.

NOTE

When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.

Method 1

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and choose **More > Create Backup** in the **Operation** column.
- Step 5** In the displayed dialog box, enter a backup name and description. Then, click **OK**. If you want to cancel the backup creation task, click **Cancel**.
- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
 - The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =
 - The time required for creating a manual backup depends on the amount of data.
- Step 6** After a manual backup has been created, you can view and manage it on the **Backup Management** page.
- Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.
- End

Method 2

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** On the **Backups & Restorations** page, click **Create Backup**. In the displayed dialog box, enter a backup name and description and click **OK**. If you want to cancel the backup creation task, click **Cancel**.
- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
 - The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =
 - The time required for creating a manual backup depends on the amount of data.
- Step 6** After a manual backup has been created, you can view and manage it on the **Backup Management** page.
- Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.
- End

6.8.5 Downloading a Full Backup File

Scenarios

This section describes how to download a manual or an automated backup file to a local device and restore data from the backup file.

RDS for PostgreSQL enables you to download full backup files.

Method 1: Using Current Browser

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Backup Management** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 In the displayed dialog box, select **Use Current Browser** for **Download Method** and click **OK**.

Step 6 Restore data locally as required.

----End

Method 2: Using Download URL

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Backup Management** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 In the displayed dialog box, select **Use Download URL** for **Download Method**, click  to copy the URL, and click **OK**.

A valid URL for downloading the backup data is displayed.

- You can use various download tools to download backup files.
- You can also run the following command to download backup files:
wget -O FILE_NAME --no-check-certificate "DOWNLOAD_URL"

The parameters in the command are as follows:

FILE_NAME: indicates the new backup file name after the download is successful. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with **wget** to rename the backup file.

DOWNLOAD_URL: indicates the location of the backup file to be downloaded. If the location contains special characters, escape is required.

Step 6 Restore data locally as required.

----End

6.8.6 Downloading an Incremental Backup File

Scenarios

This section describes how to download a manual or an automated backup file to a local device and restore data from the backup file.

RDS for PostgreSQL enables you to download incremental backup files.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance. Choose **Backups & Restorations** in the navigation pane on the left. On the **Incremental Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

You can also select the incremental backups to be downloaded and click **Download** above the list.

Step 5 After the download is complete, you can view the incremental backups locally.

----End

6.8.7 Restoring from Backup Files to RDS for PostgreSQL

Scenarios

This section describes how to use an automated or manual backup to restore a DB instance to the status when the backup was created. The restoration is at the DB instance level.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Backup Management** page, select the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page. On the displayed page, choose **Backups & Restorations**. On the displayed page, select the target backup to be restored and click **Restore** in the **Operation** column.

Step 5 Select a restoration method and click **OK**.

- Create New Instance

The **Create New Instance** page is displayed.

- The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
- Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.
- Other settings are the same as those of the original DB instance by default and can be modified. For details, see [Step 1: Create a DB Instance](#).

- Restore to Existing

- Select "I acknowledge that restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable during the restoration." and click **Next**.
- Confirm the information and click **OK**.

NOTICE

- If the target existing DB instance has been deleted, data cannot be restored to it.
- Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.
- To restore backup data to an existing DB instance, the selected DB instance must be in the same VPC as the original DB instance and must have the same DB engine and the same or later version than the original DB instance.
- Ensure that the storage space of the selected DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.

Step 6 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

After the new instance is created, a full backup will be automatically triggered.

- Restore to Existing

On the **Instance Management** page, the status of the target existing DB instance changes from **Restoring** to **Available**. If the target existing DB instance contains read replicas, the read replica status is the same as the target existing DB instance status.

After the restoration is complete, a full backup will be automatically triggered.

----End

6.8.8 Restoring a DB Instance to a Point in Time

Scenarios

You can restore from automated backups to a specified point in time.

When you enter the time point that you want to restore the DB instance to, RDS downloads the most recent full backup file from OBS to the DB instance. Then, incremental backups are also restored to the specified point in time on the DB instance. Data is restored at an average speed of 30 MB/s.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Restore to Point in Time**.

Step 6 Select the restoration date and time range, enter a time point within the selected time range, and select a restoration method. Then, click **OK**.

- Create New Instance

The **Create New Instance** page is displayed.

- The DB engine and version of the new DB instance are the same as those of the original DB instance and cannot be changed.
- Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.
- Other settings are the same as those of the original DB instance by default and can be modified. For details, see section [Step 1: Create a DB Instance](#).

- Restore to Existing
 - a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable during the restoration." and click **Next**.
 - b. Confirm the information and click **OK**.

NOTICE

- Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.
 - To restore backup data to an existing DB instance, the selected DB instance must be in the same VPC as the original DB instance and must have the same DB engine and the same or later version than the original DB instance.
 - Ensure that the storage space of the selected DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.
-

Step 7 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

The new DB instance is independent from the original one. If you need read replicas to offload read pressure, create one or more for the new DB instance.

After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

On the **Instance Management** page, the status of the DB instance changes from **Restoring** to **Available**.

After the restoration is complete, a full backup will be automatically triggered.

----End

6.8.9 Replicating a Backup

Scenarios

This section describes how to replicate a manual or an automated backup. The new backup name must be different from the original backup name.

Constraints

You can replicate backups and use them only within the same region.

Backup Retention Policy

- If you want to retain the automated backups for a long time, you can replicate them to generate manual backups, which will be always retained until you delete them.
- If the storage occupied by manual backups exceeds the provisioned free backup storage, additional storage costs may incur.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance. On the **Backups & Restorations** page, locate the target backup to be replicated and click **Replicate** in the **Operation** column.

Step 5 In the displayed dialog box, enter a new backup name and description and click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain the following special characters: > ! < " & ' =

Step 6 After the new backup has been created, you can view and manage it on the **Backup Management** page.

----End

6.9 Parameter Template Management

6.9.1 Creating a Parameter Template

You can use database parameter templates to manage the DB engine configuration. A database parameter template acts as a container for engine configuration values that can be applied to one or more DB instances.

This default template contains DB engine defaults and system defaults that are configured based on the engine, compute class, and allocated storage of the instance. Default parameter templates cannot be modified, but you can create your own parameter template to change parameter settings.

NOTICE

Not all of the DB engine parameters in a custom parameter template can be changed.

If you want to use a custom parameter template, you simply create a parameter template and select it when you create a DB instance or apply it to an existing DB instance following the instructions provided in section [Applying a Parameter Template](#).

When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template following the instructions provided in section [Replicating a Parameter Template](#).

The following are the key points you should know when using parameters in a parameter template:

- When you change a dynamic parameter value in a parameter template and save the change, the change takes effect immediately. When you change a static parameter value in a parameter template and save the change, the change will take effect only after you manually reboot the DB instances that the parameter template applies to.
- Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter template. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.

NOTE

RDS does not share parameter template quotas with DDS.

You can create a maximum of 100 parameter templates for RDS DB instances. All RDS DB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, configure required information and click **OK**.

- Select a DB engine for the parameter template.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =

----End

6.9.2 Modifying Instance Parameters

You can modify parameters in a custom parameter template to optimize RDS database performance.

You can change parameter values in custom parameter templates only and cannot change parameter values in default parameter templates.

When you modify a parameter, the time when the modification takes effect is determined by the type of the parameter.

The RDS console displays the statuses of DB instances that the parameter template applies to. For example, if the DB instance has not yet used the latest modifications made to its parameter template, its status is **Parameter change. Pending reboot**. Manually reboot the DB instance for the latest modifications to take effect for that DB instance.

NOTE

RDS has default parameter templates whose parameter values cannot be changed. You can view these parameter values by clicking the default parameter templates. If a custom parameter template is set incorrectly, the database startup may fail. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Modifying Parameters of a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, modify parameters as required.

Available operations are **Save**, **Cancel**, and **Preview**:

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

NOTICE

In the **Effective upon Reboot** column:

- If the value is **Yes** and the DB instance status on the **Instance Management** page is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
 - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
 - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.
- If the value is **No**, the modifications take effect immediately.

After parameters are modified, you can view parameter change history by referring to section [Viewing Parameter Change History](#).

----End

Modifying Parameter Template Parameters

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the **Parameters** page, modify parameters as required.

Available operations are **Save**, **Cancel**, and **Preview**:

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 6 After the parameter values are modified, you can click **View Change History** to view the modification details.

Step 7 The modifications take effect only after you apply the parameter template to DB instances. For details, see [Applying a Parameter Template](#).

Step 8 View the status of the DB instance to which the parameter template was applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- A DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/

standby DB instances, the parameter modifications are also applied to the standby DB instance.)

- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

6.9.3 Exporting a Parameter Template

Scenarios

- You can export a parameter template of a DB instance for future use. You can also apply the exported parameter template to DB instances by referring to [Applying a Parameter Template](#).
- You can export the parameter template information (parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analysis.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

- Exporting to a custom template

In the displayed dialog box, configure required information and click **OK**.

NOTE

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

After the parameter template is exported, a new template is generated in the list on the **Parameter Template Management** page.

- Exporting to a file

The parameter template information (parameter names, values, and descriptions) of a DB instance is exported to a CSV file. In the displayed dialog box, enter the file name and click **OK**.

NOTE

The file name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

----End

6.9.4 Comparing Parameter Templates

Scenarios

You can compare DB instance parameters with a parameter template that uses the same DB engine to understand the differences of parameter settings.

You can also compare default parameter templates that use the same DB engine to understand the differences of parameter settings.

Comparing Instance Parameters with a Parameter Template

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.
- Step 6** In the displayed dialog box, select a parameter template to be compared and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

Comparing Parameter Templates

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.
- Step 5** In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

6.9.5 Viewing Parameter Change History

Scenarios

You can view the change history of DB instance parameters or custom parameter templates.

NOTE

An exported or custom parameter template has initially a blank change history.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

You can apply the parameter template to DB instances as required by referring to section [Applying a Parameter Template](#).

----End

Viewing Change History of a Parameter Template

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

----End

6.9.6 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template.

After a parameter template is replicated, it takes about 5 minutes before the new template is displayed.

Default parameter templates cannot be replicated, but you can create parameter templates based on the default ones.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

Step 5 In the displayed dialog box, configure required information and click **OK**.

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Template Management** page.

-----End

6.9.7 Resetting a Parameter Template

Scenarios

You can reset all parameters in a custom parameter template to their default settings.

Procedure

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and choose **More > Reset** in the **Operation** column.
- Step 5** Click **Yes**.
- Step 6** The modifications take effect only after you apply the parameter template to DB instances. For details, see [Applying a Parameter Template](#).
- Step 7** View the status of the DB instance to which the parameter template is applied.
- If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
 - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
 - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

6.9.8 Applying a Parameter Template

Scenarios

You can apply parameter templates to DB instances as needed. A parameter template can be applied only to DB instances of the same version.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, perform the following operations based on the type of the parameter template to be applied:
- If you intend to apply a default parameter template to DB instances, click **Default Templates**, locate the target parameter template, and click **Apply** in the **Operation** column.
 - If you intend to apply a custom parameter template to DB instances, click **Custom Templates**, locate the target parameter template, and choose **More > Apply** in the **Operation** column.

A parameter template can be applied to one or more DB instances.

- Step 5** In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

After the parameter template is successfully applied, you can view the application records by referring to section [Viewing Application Records of a Parameter Template](#).

----End

6.9.9 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** Choose **Parameter Template Management** in the navigation pane on the left.
- Step 5** On the **Default Templates** page, locate the target parameter template and click **View Application Record** in the **Operation** column. Alternatively, on the **Custom Templates** page, choose **More > View Application Record** in the **Operation** column.

You can view the name or ID of the DB instance that the parameter template applies to, as well as the application status, application time, and failure cause (if failed).

----End

6.9.10 Modifying a Parameter Template Description

Scenarios

You can modify the description of a parameter template you have created.

NOTE

You cannot modify the description of a default parameter template.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.

- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click  in the **Description** column.
- Step 5** Enter a new description. You can click  to submit or to cancel the modification.
- The description consists of a maximum of 256 characters and cannot contain the following special characters: > ! < " & ' =
 - After the modification is successful, you can view the new description in the **Description** column of the parameter template list.

----End

6.9.11 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer in use.

NOTICE

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template to be deleted and choose **More > Delete** in the **Operation** column.
- Step 5** In the displayed dialog box, click **Yes**.

----End

6.10 Connection Management

6.10.1 Configuring and Changing a Floating IP Address

Scenarios

You can change floating IP addresses after migrating on-premises databases or other cloud databases to RDS.

Constraints

After a floating IP address is changed, the domain name needs to be resolved again. This operation takes several minutes and may interrupt database connections. Therefore, you are advised to change a floating IP address during off-peak hours.

Configuring a Floating IP Address

You can use an automatically-assigned IP address when creating a DB instance.

Changing a Floating IP Address

You can change the floating IP address of an existing DB instance.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the **Connection Information** area on the **Basic Information** page, click **Change** in the **Floating IP Address** field.

Step 6 In the displayed dialog box, enter a new floating IP address and click **OK**.

In the in-use IP address list, the IP addresses whose statuses are **To be used** are occupied and cannot be used.

----End

6.10.2 Binding and Unbinding an EIP

Scenarios

NOTICE

To ensure that the DB instance is accessible, the security group associated with the instance must allow access over the database port. For example, if the database port is 5432, ensure that the security group allows access over the 5432 port.

Prerequisites

- You can bind an EIP to a primary DB instance or a read replica only.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the DB instance first before binding a new EIP to it.

Binding an EIP

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **EIPs** page, click **Bind EIP**.

Step 6 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **OK**. If no available EIPs are displayed, click **View EIP** and obtain an EIP.

Step 7 On the **EIPs** page of the RDS console, view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

To unbind the EIP from the DB instance, see [Unbinding an EIP](#).

----End

Unbinding an EIP

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the DB instance that has an EIP bound.

Step 5 On the **EIPs** page, locate the target EIP to be unbound and click **Unbind**. In the displayed dialog box, click **Yes**.

Step 6 On the **EIPs** page, view the unbinding result.

You can also view the progress and result of unbinding an EIP from a DB instance on the **Task Center** page.

To bind an EIP to the DB instance again, see [Binding an EIP](#).

----End

6.10.3 Changing a Database Port

Scenarios

This section describes how to change the database port of a primary DB instance or a read replica. For primary/standby DB instances, changing the database port of the primary DB instance will cause the database port of the standby DB instance to also be changed accordingly.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance or click  first and then click the target read replica.
- Step 5** In the **Connection Information** area on the **Basic Information** page, click  in the **Database Port** field.

NOTE

A PostgreSQL database can use ports 2100 to 9500.

- To submit the change, click .
 - In the dialog box, click **Yes**.
 - i. If you change the database port of the primary DB instance, that of the standby DB instance will also be changed and both DB instances will be rebooted.
 - ii. If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will be rebooted.
 - iii. This process takes 1-5 minutes.
 - In the dialog box, click **No** to cancel the modification.
- To cancel the change, click .

- Step 6** View the results on the **Basic Information** page.

----End

6.10.4 Connecting to a DB Instance Through pgAdmin

You can use the pgAdmin client to connect to an RDS DB instance.

NOTICE

The pgAdmin version must be 4 or later.

Preparations

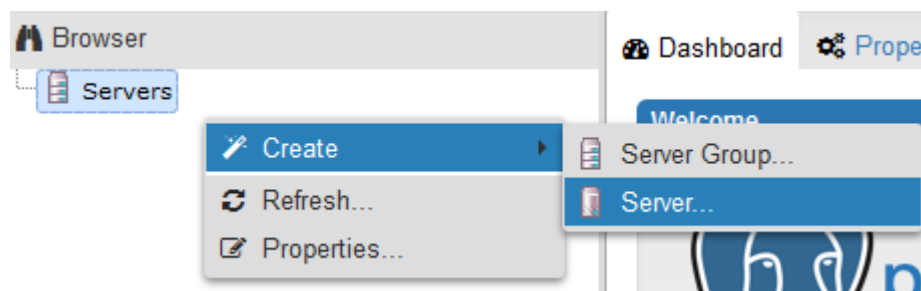
1. Prepare an **ECS** or a device that can access RDS DB instances.
To connect to a DB instance through a floating IP address, you must:
 - Ensure that the ECS and DB instance must be in the same VPC.
 - Ensure that the ECS must be allowed by the security group to access RDS DB instances.To connect to a DB instance through an EIP, you must:
 - a. Ensure that the local device can access the EIP that has been bound to the DB instance.
2. Install the pgAdmin client on the prepared ECS or device.

Procedure

Step 1 Start pgAdmin.

Step 2 In the displayed login window, choose **Servers > Create > Server**.

Figure 6-1 Creation



Step 3 On the **General** page, specify **Name**. On the **Connection** page, specify information about the DB instance to be connected. Then, click **Save**.

Figure 6-2 General page

The screenshot shows the 'Create - Server' dialog box with the 'General' tab selected. The 'Name' field is empty and has a red border. The 'Server group' dropdown menu is set to 'Servers'. The 'Connect now?' checkbox is checked. The 'Comments' field is empty. A red error message at the bottom states 'Name must be specified.' The bottom bar contains an information icon, a help icon, and 'Save', 'Cancel', and 'Reset' buttons.

Create - Server	
General	Connection
Name	
Server group	Servers
Connect now?	☒
Comments	
Name must be specified.	
[i] [?] [Save] [Cancel] [Reset]	

Figure 6-3 Connection page

The screenshot shows the 'Create - Server' dialog box with the 'Connection' tab selected. The 'Host name/address' field is empty and has a red border. The 'Port' field contains '5432'. The 'Maintenance database' field contains 'postgres'. The 'User name' field is empty and has a red border. The 'Password' field is empty. The 'Save password?' checkbox is unchecked. The 'Role' field is empty. The 'SSL mode' dropdown menu is set to 'Prefer'. A red error message at the bottom states 'Name must be specified.' The bottom bar contains an information icon, a help icon, and 'Save', 'Cancel', and 'Reset' buttons.

Create - Server	
General	Connection
Host name/address	
Port	5432
Maintenance database	postgres
User name	
Password	
Save password?	☐
Role	
SSL mode	Prefer
Name must be specified.	
[i] [?] [Save] [Cancel] [Reset]	

Parameter description:

- **Host name/address:** indicates the IP address of the DB instance you want to connect to. If you connect to a DB instance through a floating IP address, enter the floating IP address displayed in the **Connection Information** area on the **Basic Information** page of your DB instance. If you connect to a DB instance through an EIP, enter the EIP of your DB instance.
- **Port:** indicates the database port. By default, the value is **5432**.
- **User name:** indicates the username. By default, the value is **root**.
- **Password:** indicates the password of the target database username.

Step 4 In the login window, check that the connection information is correct. The target DB instance is successfully connected.

----End

6.11 Plugin Management

6.11.1 Creating or Deleting a Plugin

RDS provides the PostgreSQL plugin management solution for user **root**. Except the following plugins, you need to manually create other plugins by referring to this section.

- auto_explain
- passwordcheck
- pg_profile_pro
- pg_sql_history
- wal2json
- test_decoding

NOTE

The PostgreSQL plugin takes effect at the database level, not globally. You need to manually create it on corresponding databases.

The latest minor versions of PostgreSQL 11, 12, and Enhanced Edition allow the **root** user to create plugins (create extension) or delete plugins (drop extension).

Creating a Plugin

Step 1 Connect to the database **database1** as user **root** and use **template1** to create a database that can support the plugin.

```
# psql --host=<RDS_ADDRESS> --port=<DB_PORT> --dbname=database1 --
username=root -c "create database <DB_NAME> template template1;"
```

- **RDS_ADDRESS** indicates the IP address of the RDS DB instance.
- **DB_PORT** indicates the RDS DB instance port.
- **DB_NAME** indicates the name of the database to be created.

Enter the password of user **root** when prompted.

Create a database named *my_extension_db* that can support the plugin. Example:

```
# psql --host=192.168.6.141 --port=5432 --dbname=database1 --
username=root -c "create database my_extension_db template template1;"
```

```
Password for user root:
CREATE DATABASE
```

Note: If you are creating a database as a common user, log in to the created database as the common user and run the following command to grant all rights to user **root**:

```
GRANT ALL ON DATABASE db1 TO root;
```

Step 2 Connect to the created database as user **root** and create a plugin.

```
# psql --host=<RDS_ADDRESS> --port=<DB_PORT> --dbname=<DB_NAME> --
username=root -c "select control_extension('create',<EXTENSION_NAME>);"
```

- *RDS_ADDRESS* indicates the IP address of the RDS DB instance.
- *DB_PORT* indicates the RDS DB instance port.
- *DB_NAME* indicates the name of the database to be created.
- *EXTENSION_NAME* indicates the plugin name. For more information, see [Supported Plugins](#).

Enter the password of user **root** when prompted.

Create the postgis plugin in the database *my_extension_db*. Example:

```
# psql --host=192.168.6.141 --port=5432 --dbname=my_extension_db --
username=root -c "select control_extension('create','postgis');"
```

```
Password for user root:
control_extension
-----
create postgis successfully.
(1 row)
```

----End

Deleting a Plugin

Connect to the database with a plugin created as user **root** and delete the plugin.

```
# psql --host=<RDS_ADDRESS> --port=<DB_PORT> --username=root --
dbname=<DB_NAME> -c "select control_extension
('drop',<EXTENSION_NAME>);"
```

- *RDS_ADDRESS* indicates the IP address of the RDS DB instance.
- *DB_PORT* indicates the RDS DB instance port.
- *DB_NAME* indicates the name of the database to be created.
- *EXTENSION_NAME* indicates the plugin name. For more information, see [Supported Plugins](#).

Enter the password of user **root** when prompted.

Example:

```
# psql --host=192.168.6.141 --port=5432 --dbname=my_extension_db --
username=root -c "select control_extension('drop','postgis');"
```

```
Password for user root:
control_extension
-----
drop postgis successfully.
(1 row)
```

6.11.2 Supported Plugins

NOTE

The following table lists the plugins supported by the latest minor versions of RDS for PostgreSQL. You can use **SELECT name FROM pg_available_extensions;** to view the plugins supported by your DB instance.

Before using the `mysql_fdw`, `oracle_fdw`, `pgsql-ogr-fdw`, `postgres_fdw`, and `tds_fdw` plugins, ensure that the server IP addresses of the two PostgreSQL DB instances are in the same VPC and subnet.

Table 6-11 Supported plugins

Plugin Name	Postgre SQL 9.5	Postgre SQL 9.6	Postgre SQL 10	Postgre SQL 11	Postgre SQL Enhanc ed Edition (1.0)	Postgr eSQL 12
address_standar dizer	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0
address_standar dizer_data_us	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0
btree_gin	1.0	1.0	1.2	1.3	1.3	1.3
btree_gist	1.1	1.2	1.5	1.5	1.5	1.5
citext	1.1	1.3	1.4	1.5	1.5	1.6
cube For details, see cube .	1.0	1.2	1.2	1.4	1.4	1.4
dblink	1.1	1.2	1.2	1.2	1.2	1.2
dict_int	1.0	1.0	1.0	1.0	1.0	1.0
dict_xsyn	1.0	1.0	1.0	1.0	1.0	1.0
earthdistance For details, see earthdistance .	1.0	1.1	1.1	1.1	1.1	1.1
fuzzystrmatch	1.0	1.1	1.1	1.1	1.1	1.1
hll	2.12	2.12	2.12	2.12	2.12	2.14
hstore	1.3	1.4	1.4	1.5	1.5	1.6

Plugin Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL Enhanced Edition (1.0)	PostgreSQL 12
intagg	1.0	1.1	1.1	1.1	1.1	1.1
intarray	1.0	1.2	1.2	1.2	1.2	1.2
isn	1.0	1.1	1.1	1.2	1.2	1.2
ltree	1.0	1.1	1.1	1.1	1.1	1.1
oracle_fdw	-	-	1.1	1.1	1.1	1.1
orafce	3.8	3.8	3.8	3.8	3.8	3.8
pg_cron	-	-	-	-	-	1.2
pg_hint_plan	1.1.5	1.2.0	1.3.0	1.3.0	1.3.0	-
pg_jieba	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0
pg_pathman	1.5	1.5	1.5	1.5	1.5	1.5
pg_repack	-	-	-	1.4.6	1.4.6	1.4.6
pgrouting	-	-	-	3.1.0	3.1.0	3.1.0
pg_stat_statements	1.3	1.4	1.6	1.6	1.6	1.7
pg_trgm	1.1	1.3	1.3	1.4	1.4	1.4
pgcrypto	1.2	1.3	1.3	1.3	1.3	1.3
pgstattuple	-	-	1.5	1.5	1.5	1.5
plpgsql	1.0	1.0	1.0	1.0	1.0	1.0
postgis For details, see postgis .	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0
postgres_fdw	-	-	1.0	1.0	1.0	1.0
postgis_raster	Integrated to postgis	Integrated to postgis	Integrated to postgis	Integrated to postgis	Integrated to postgis	3.0.0
postgis_sfcgal	1.3.9	1.3.9	1.3.9	1.3.9	1.3.9	1.3.9
postgis_tiger_geocoder	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0
postgis_topology	2.5.1	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0

Plugin Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL Enhanced Edition (1.0)	PostgreSQL 12
powa-archivist	-	-	-	4.1.2	-	4.1.2
pg_qualstats	-	-	-	2.0.2	-	2.0.2
pg_stat_kcache	-	-	-	2.2.0	-	2.2.0
pg_track_settings	-	-	-	2.0.1	-	2.0.1
pg_wait_sampling	-	-	-	1.1.3	-	1.1.3
hypopg	-	-	-	1.2.0	-	1.2.0
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0
tds_fdw	-	-	-	2.0.1	2.0.1	2.0.1
timescaledb For details, see timescaledb .	-	1.3.2	1.3.2	1.3.2	1.3.2	1.7.0
unaccent	1.0	1.1	1.1	1.1	1.1	1.1
uuid-oss	1.0	1.1	1.1	1.1	1.1	1.1
wal2json	-	-	-	2.3	2.3	2.3
zhparser	1.0	1.0	1.0	1.0	1.0	1.0

Plugin Description

- **postgis**
 - When postgis is created, the following plugins are created at the same time:
postgis
postgis_topology
fuzzystrmatch
postgis_tiger_geocoder
address_standardizer
address_standardizer_data_us
 - After the postgis plugin is created on the primary DB instance, you need to disconnect the connection from the standby DB instance first and re-establish a connection to update the **search_path** setting.

- For PostgreSQL Enhanced Edition DB instances, you need to set **empty_is_null** to **OFF** on the console before creating the postgis plugin.
- **earthdistance**
To install the earthdistance plugin, you must install the cube plugin first.
- **cube**
If the earthdistance plugin has been installed, deleting the cube plugin will cause the earthdistance plugin to be unavailable.
- **timescaledb**
The timescaledb plugin does not support the TLS protocol. For more information, see [APIs Not Supported by the timescaledb Plugin](#).
- **wal2json**
This plugin is a logical replication plugin. You can directly use it without installing it through control_extension.

APIs Not Supported by the timescaledb Plugin

- add_compress_chunks_policy
- add_drop_chunks_policy
- add_reorder_policy
- alter_job_schedule
- compress_chunk
- decompress_chunk
- drop_chunks
- interpolate
- locf
- move_chunk
- remove_compress_chunks_policy
- remove_drop_chunks_policy
- remove_reorder_policy
- reorder_chunk
- set_integer_now_func
- time_bucket_gapfill

6.12 Database Account Security

Password Strength Requirements

- RDS has a password security policy for user-created database accounts. Passwords must:
 - Consist of at least eight characters.
 - Contain letters, digits, and special characters.
 - Not contain the username.

Suggestions for Creating Users

When you run **CREATE USER** or **CREATE ROLE**, you are advised to specify a password expiration time with the **VALID UNTIL 'timestamp'** parameter (**timestamp** indicates the expiration time).

Suggestions for Accessing Databases

When you access a database object, you are advised to specify the schema name of the database object to prevent [trojan-horse attacks](#).

Account Description

To provide O&M services, the system automatically creates system accounts when you create PostgreSQL DB instances. These accounts cannot be used by users.

NOTICE

Attempting to delete, rename, and change passwords or permissions for these accounts will result in an error.

- **rdsAdmin**: management account, which has the superuser permissions and is used to query and modify DB instance information, rectify faults, migrate data, and restore data.
- **rdsRepl**: replication account, which is used to synchronize data from primary DB instances to standby DB instances or read replicas.
- **rdsBackup**: backup account, which is used for backend backup.
- **rdsMetric**: metric monitoring account, which is used by watchdog to collect database status data.
- **rdsProxy**: database proxy account, which is used for authentication when the database is connected through the read/write splitting address. This account is automatically created when you enable read/write splitting.

6.13 Data Security

6.13.1 Resetting the Administrator Password

Scenarios

You can reset the administrator password of a primary instance.

You can also reset the password of your database account when using RDS.

 NOTE

- If you have changed the administrator password of the primary DB instance, the administrator passwords of the standby DB instance and read replicas (if any) will also be changed accordingly.
- The time required for the new password to take effect depends on the amount of service data currently being processed by the primary DB instance.
- To protect against brute force hacking attempts and ensure system security, change your password periodically, such as every three or six months.

Method 1

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and choose **More > Reset Password** in the **Operation** column.
- Step 5** Enter and confirm a new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*_-=+?,.). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

Method 2

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the **DB Information** area on the **Basic Information** page, click **Reset Password** in the **Administrator** field.
- Step 6** Enter and confirm a new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*_+=+?,.). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

6.13.2 Changing a Security Group

Scenarios

This section describes how to change the security group of a primary DB instance or read replica. For primary/standby DB instances, changing the security group of the primary DB instance will cause the security group of the standby DB instance to also be changed accordingly.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary DB instance or read replica.

Step 5 In the **Connection Information** area on the **Basic Information** page, click  in the **Security Group** field.

- To submit the change, click .
- To cancel the change, click .

Step 6 Changing the security group takes 1 to 3 minutes. Click  in the upper right corner on the **Basic Information** page to view the results.

----End

6.14 and Alarms Metrics and Alarms

6.14.1 Configuring Displayed Metrics

The RDS Agent monitors RDS DB instances and collects monitoring metrics only.

Description

This section describes the metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS.

Namespace

SYS.RDS

DB Instance Monitoring Metrics

- [Table 6-12](#) lists the performance metrics of PostgreSQL databases.

Table 6-12 Database performance metrics

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100 %	Monitored object: Monitored instance type: PostgreSQL instance	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100 %	Monitored object: Monitored instance type: PostgreSQL instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	Monitored object: Monitored instance type: PostgreSQL instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	Monitored object: Monitored instance type: PostgreSQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	Monitored object: Monitored instance type: PostgreSQL instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0-100 %	Monitored object: Monitored instance type: PostgreSQL instance	1 minute
rds040_transaction_logs_usage	Transaction Logs Usage	Storage space usage of transaction logs	≥ 0 MB	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds041_replication_slot_usage	Replication Slot Usage	Storage space usage of replication slot files	≥ 0 MB	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds042_database_connections	Database Connections in Use	Number of database connections in use	≥ 0 counts	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds043_maximum_used_transaction_ids	Maximum Used Transaction IDs	Maximum number of transaction IDs that have been used	≥ 0 counts	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds044_transaction_logs_generations	Transaction Logs Generation	Size of transaction logs generated per second	≥ 0 MB/s	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds045_oldest_replication_slot_lag	Oldest Replication Slot Lag	Lagging size of the most lagging replica in terms of WAL data received	≥ 0 MB	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds046_replication_lag	Replication Lag	Replication lag	≥ 0 ms	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	Monitored object: database Monitored instance type: PostgreSQL instance	1 minute

Metric	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	Monitored object: Monitored instance type: PostgreSQL instance	1 minute

Dimension

Key	Value
postgresql_instance_id	PostgreSQL DB instance ID

6.14.2 Setting Alarm Rules

Scenarios

You can set alarm rules by referring to [Setting Alarm Rules](#) to customize the monitored objects and notification policies and keep track of the RDS running status.

The RDS alarm rules include alarm rule names, services, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Setting Alarm Rules

- Step 1** Log in to the management console.
- Step 2** Click **Service List** and choose **Management & Governance > Cloud Eye**.
- Step 3** In the navigation pane on the left, choose **Alarm Management > Alarm Rules**.
- Step 4** On the displayed **Alarm Rules** page, click **Create Alarm Rule**.

----End

6.14.3 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS monitoring metrics on the management console.

Monitored data takes some time before it can be displayed. The RDS status displayed on the Cloud Eye console is about 5 to 10 minutes delayed. When you create a new RDS DB instance, it takes 5 to 10 minutes before monitoring data is displayed on Cloud Eye.

Prerequisites

- RDS is running properly.
Monitoring metrics of the RDS DB instances that are faulty or have been deleted are not displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to normal.

NOTE

If an RDS DB instance has been faulty for 24 hours, Cloud Eye considers it to no longer exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- RDS has been running properly for about 10 minutes.
For a newly created RDS DB instance, you need to wait a bit before you can view the metrics.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click **View Metric** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metric** in the upper right corner of the page to go to the Cloud Eye console.

Step 5 On the Cloud Eye console, view monitoring metrics of the primary DB instance.

Cloud Eye can monitor performance metrics from the last 1 hour, 3 hours, 12 hours, 1 day, and 7 days.

----End

6.15 Interconnection with CTS

6.15.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 6-13 RDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or a read replica, or restoring data to a new DB instance	instance	createInstance
Scaling up storage space and changing instance class	instance	instanceAction
Rebooting a DB instance	instance	instanceRestart
Restoring data to the original DB instance	instance	instanceRestore
Renaming a DB instance	instance	instanceRename
Resetting a password	instance	resetPassword
Setting database version parameters	instance	setDBParameters
Resetting database version parameters	instance	resetDBParameters
Enabling, modifying, or disabling a backup policy	instance	setBackupPolicy
Changing a database port	instance	changeInstancePort
Binding or unbinding an EIP	instance	setOrResetPublicIP
Modifying a security group	instance	modifySecurityGroup
Adding a tag	instance	createTag
Deleting a tag	instance	deleteTag
Editing a tag	instance	modifyTag
Deleting a DB instance	instance	deleteInstance
Performing a primary/standby switchover	instance	instanceFailOver
Changing the replication mode	instance	instanceFailOver-Mode
Changing a failover priority	instance	instanceFailOver-Strategy
Changing a DB instance type from single to primary/standby	instance	modifySingleToHaInstance
Creating a backup	backup	createManualSnapshot
Replicating a backup	backup	copySnapshot

Operation	Resource Type	Trace Name
Downloading a backup (using OBS)	backup	downloadSnapshot
Downloading a backup (using a browser)	backup	backupsDownload
Deleting a backup	backup	deleteManualSnapshot
Downloading a merged backup	backup	packBackupsDownload
Creating a parameter template	parameterGroup	createParameterGroup
Modifying parameters in a parameter template	parameterGroup	updateParameterGroup
Deleting a parameter template	parameterGroup	deleteParameterGroup
Replicating a parameter template	parameterGroup	copyParameterGroup
Resetting a parameter template	parameterGroup	resetParameterGroup
Applying a parameter template	parameterGroup	applyParameterGroup
Saving parameters in a parameter template	parameterGroup	saveParameterGroup
Deleting a frozen DB instance	all	rdsUnsubscribeInstance
Freezing a DB instance	all	rdsfreezeInstance

6.15.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

This section describes how to query the operation records of last 7 days on the CTS console.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Management & Governance > Cloud Trace Service**.

Step 3 Choose **Trace List** in the navigation pane on the left.

Step 4 Filter conditions to query traces. The details are described as follows:

- **Trace Source, Resource Type, and Search By:** Select a filter from the drop-down list.
When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
- **Operator:** Select a specific operator from the drop-down list.
- **Trace Status:** Available options include **All trace statuses, Normal, Warning, and Incident**. You can only select one of them.
- In the upper right corner of the page, you can specify a time range for querying traces.

Step 5 Select the search criteria, and click **Query**.

Step 6 Click  on the left of the required trace to expand its details.

Step 7 Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.

Step 8 Click **Export** on the right. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to traces on the management console.

For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.

----End

6.16 Task Center

6.16.1 Viewing a Task

You can view the detailed progress and result of the task on the **Task Center** page.

NOTE

You can view and manage the following tasks:

- Creating DB instances
- Creating read replicas
- Changing single DB instances to primary/standby
- Scaling up storage space
- Binding EIPs to DB instances
- Unbinding EIPs from DB instances
- Switching primary/standby DB instances
- Rebooting DB instances
- Restoring data to new DB instances

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Task Center** page, locate the target task and view its details.
- End

6.16.2 Deleting a Task Record

You can delete task records so that they are no longer displayed in the task list. This operation only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** Choose **Task Center** in the navigation pane on the left. On the displayed page, locate the target task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.
- You can delete the records of instant tasks in any of the following statuses:
- Running
 - Completed
 - Failed
- End

7 Working with RDS for SQL Server

7.1 Instance Lifecycle

7.1.1 Creating a Same DB Instance as an Existing DB Instance

Scenarios

This section describes how to quickly create a DB instance with the same configurations as the selected one.

NOTE

- You can create DB instances with the same configurations numerous times.
- This function is unavailable for read replicas.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Create Same DB Instance** in the **Operation** column.

Step 5 On the displayed page, the configurations are the same as those of the selected DB instance. You can change them as required. Then, click **Create Now**.

Step 6 Confirm the instance specifications.

Step 7 Refresh the DB instance list and view the status of the DB instance. If the status is **Available**, it has been created successfully.

You can manage the DB instance on the **Instance Management** page.

----End

7.1.2 Rebooting DB Instances or Read Replicas

Scenarios

You may need to reboot a DB instance during maintenance. For example, after you modify some parameters, a reboot is required for the modifications to take effect. You can reboot a primary DB instance or a read replica on the management console.

Constraints

- If the database service status is abnormal, you can forcibly reboot the DB instance, but this will interrupt uncommitted transactions.
- Rebooting DB instances will cause service interruptions. During the reboot process, the DB instance status is **Rebooting**.
- Rebooting DB instances will cause instance unavailability and clear cached memory. To prevent traffic congestion during peak hours, you are advised to reboot DB instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance, or click  and then locate the target read replica. Choose **More > Reboot** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page to go to the **Basic Information** page. In the upper right corner, click **Reboot**.

For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 5 In the displayed dialog box, select a reboot mode. If you select **Forceful**, select the checkbox before **Confirm forcible reboot** and click **Yes**.

Step 6 Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has rebooted successfully.

----End

7.1.3 Selecting Displayed Items

Scenarios

You can customize which instance items are displayed on the **Instance Management** page.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click  to edit columns displayed in the DB instance list.
- The following items are displayed by default: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, and operation. These default items cannot be deselected.
 - In a single project, you can select up to 9 items: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, creation time, database port, storage type, and operation.
 - For multiple projects, if you have enabled the ProjectMan permissions, you can select up to 9 items: DB instance name/ID, DB instance type, DB engine version, status, floating IP address, creation time, database port, storage type, and operation.
- End

7.1.4 Exporting DB Instance Information

Scenarios

You can export information about all or selected DB instances to view and analyze DB instance information.

Exporting Information About All DB Instances

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click  in the upper right corner of the page. By default, information about all DB instances are exported. In the displayed dialog box, you can select the items to be exported and click **Export**.
- Step 5** Find a .csv file locally after the export task is completed.
- End

Exporting Information About Selected DB Instances

- Step 1** Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, filter DB instances by DB engine, DB instance name, DB instance ID, or floating IP address, or select the DB instances to be exported, and click  in the upper right corner of the page. In the displayed dialog box, select the items to be exported and click **Export**.
- Step 5** Find a .csv file locally after the export task is completed.
- End

7.1.5 Deleting a Pay-per-Use DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas as required on the **Instance Management** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If you delete a DB instance, its automated backups are also deleted and you are no longer billed for them. Manual backups, however, are still retained and will generate additional costs.

NOTICE

- If you delete a primary DB instance, its standby DB instance and read replicas (if any) are also deleted automatically. Exercise caution when performing this operation.
 - Deleted DB instances cannot be recovered and resources are released. Exercise caution when performing this operation. If you want to retain data, [create a manual backup](#) first before deleting the DB instance.
 - You can use a manual backup to restore a DB instance. For details, see [Restoring from Backup Files to DB Instances](#).
-

Deleting a DB Instance

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the primary DB instance to be deleted and click **More > Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

Step 6 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Read Replica

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click . All the read replicas created for the DB instance are displayed.

Step 5 Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.

Step 6 In the displayed dialog box, click **Yes**.

Step 7 Refresh the DB instance list later to check that the deletion is successful.

----End

7.1.6 Recycling a DB Instance

Scenarios

Deleted DB instances can be moved to the recycle bin. You can rebuild DB instances from the recycle bin to restore data. The DB engine, DB engine version, and storage type of the new DB instance are the same as those of the original DB instance. Other parameters can be reconfigured. DB instances that were deleted up to 7 days ago can be restored.

Constraints

- Read replicas cannot be moved to the recycle bin.
- The recycle bin is enabled by default and cannot be disabled.

Modifying Recycling Policy

NOTICE

A new recycling policy only applies to DB instances that were put in the recycle bin after the new policy was put into effect. For DB instances that were in the recycle bin before the modification, the original recycling policy takes effect.

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** In the navigation pane on the left, choose **Recycling Management**.
- Step 5** On the **Recycling Management** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances from 1 day to 7 days.
- Step 6** Then, click **OK**.
- End

Rebuilding a DB Instance

You can rebuild the primary DB instances in the recycle bin within the retention period.

- Step 1** In the navigation pane on the left, choose **Recycling Management**.
- Step 2** On the **Recycling Management** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.
- Step 3** On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see [Restoring from Backup Files to DB Instances](#).
- End

7.2 Instance Modifications

7.2.1 Changing a DB Instance Name

Scenarios

You can change the name of a primary DB instance or read replica.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and click  next to it to edit the DB instance name. Then, click **OK**.
- Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Instance Name** field in the **DB Information** area, click  to edit the DB instance name.

The DB instance name must start with a letter and consist of 4 to 64 characters. Only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_) are allowed.

- To submit the change, click .
- To cancel the change, click .

Step 5 View the results on the **Basic Information** page.

----End

7.2.2 Changing the Failover Priority

Scenarios

You can configure the failover priority for reliability or for availability, depending on your service requirements.

- **Reliability:** This option is selected by default. Data consistency is preferentially ensured during a primary/standby failover. This is recommended for users whose highest priority is data consistency.
- **Availability:** Database availability is preferentially ensured during a primary/standby failover. This is recommended for users who require their databases to provide uninterrupted online services.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary/standby DB instances.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click **Change** in the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.

Step 6 View the results on the **Basic Information** page.

----End

7.2.3 Changing a DB Instance Class

Scenarios

You can change the instance class (CPU or memory) of a DB instance as required. If the status of a DB instance changes from **Changing instance class** to **Available**, the change is successful.

Constraints

- A DB instance cannot be deleted when its instance class is being changed.
- After you change instance classes, the DB instances will be rebooted and service will be interrupted. Therefore, you are advised to change instance classes during off-peak hours.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Change Instance Class** in the **Operation** column.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click **Change** in the **Instance Class** field.

Step 5 On the displayed page, specify the new instance class and click **Next**.

Step 6 View the DB instance class change result.

Changing the DB instance class takes 5–15 minutes. During this period, the status of the DB instance on the **Instance Management** page is **Changing instance class**. After a few minutes, click the DB instance and view the instance class on the displayed **Basic Information** page to check that the change is successful.

NOTICE

After you change a SQL Server DB instance class, the value of **max server memory** will also be changed accordingly. You are advised to set it to a size equal to the memory size minus 520 MB. For example, if there is 1 GB of memory (1,024 MB), you are advised to set **max server memory** to 504 MB (1,024 MB – 520 MB).

----End

7.2.4 Scaling up Storage Space

Scenarios

If the original storage space is insufficient as your services grow, scale up storage space of your DB instance.

The DB instance needs to preserve at least 15% of its capacity to work properly. The new minimum storage space required to make this instance available has been automatically calculated for you. You are advised to set alarm rules for the storage space usage by referring to [Setting Alarm Rules](#).

For details about the causes and solutions of insufficient storage space, see [What Should I Do If My Data Exceeds the Available Storage of an RDS DB Instance?](#)

RDS allows you to scale up storage space of DB instances but you cannot change the storage type. During the scale-up period, services are not interrupted.

Constraints

- DB instances can be scaled up numerous times.
- The maximum allowed storage is 4,000 GB. There is no limit on the number of scale-ups.
- For primary/standby DB instances, scaling up the primary DB instance will cause the standby DB instance to also be scaled up accordingly.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Scale Storage Space** in the **Operation** column.

You can also perform the following operations to scale up storage space:

- Click the target DB instance to enter the **Basic Information** page. In the **Storage Space** area, click **Scale**.

Step 5 On the displayed page, specify the new storage space and click **Next**.

The minimum start value of each scaling is 10 GB. A DB instance can be scaled up only by a multiple of 10 GB. The allowed maximum storage space is 4,000 GB.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.
- If your settings are correct, click **Submit**.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time, the status of the DB instance on the **Instance Management** page will be **Scaling up**. Click the DB instance and view the utilization on the displayed **Basic Information** page to verify that the scale-up is successful.

----End

7.2.5 Changing a DB Instance Type from Single to Primary/Standby

Scenarios

- RDS enables you to change single DB instances to primary/standby DB instances to improve instance reliability.
- Primary/standby DB instances support automatic failover. If the primary DB instance fails, the standby DB instance takes over services quickly. You are advised to deploy primary and standby DB instances in different AZs for high availability and disaster recovery.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate a single DB instance and choose **More > Change Type to Primary/Standby** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  on the left to change the instance type from single to primary/standby.

Step 5 Select a standby AZ and enter the original administrator password. Other configurations are the same as those of the primary DB instance by default. Confirm the configurations and click **Submit**.

Step 6 After a single DB instance is changed to primary/standby instance, you can view and manage it on the **Instance Management** page.

- The DB instance is in the **Changing type to primary/standby** status. You can view the progress on the **Task Center** page. For details, see [Task Center](#).
- In the upper right corner of the DB instance list, click  to refresh the list. After the DB instance type is changed to primary/standby, the instance status will change to **Available** and the instance type will change to **Primary/Standby**.

----End

7.2.6 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create primary/standby DB instances, RDS will create a primary DB instance and a synchronous standby DB instance in the same region. You can access the primary DB instance only. The standby instance serves as a backup. You

can manually promote the standby DB instance to the new primary instance for failover support.

Constraints

- A DB instance is running properly.
- The replication between the primary and standby instances is normal.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary/standby DB instance.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** in the **DB Instance Type** field.

Alternatively, click  in the DB instance topology on the **Basic Information** page to perform a primary/standby switchover.

NOTICE

A primary/standby switchover may cause service interruptions for some seconds or minutes (depending on the replication delay). If the replication delay is too long, a small amount of data may be lost. To prevent traffic congestion, you are advised to perform a switchover during off-peak hours.

Step 6 In the **Switch Primary/Standby Instances** dialog box, click **Yes** to switch between the primary and standby DB instances.

If the replication status is **Available** and the replication delay is greater than 300s, the primary/standby switchover task cannot be delivered.

Step 7 After a switchover is successful, you can view and manage the DB instance on the **Instance Management** page.

- During the switchover process, the DB instance status is **Switchover in progress**.
- In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is successful, the DB instance status will become **Available**.

----End

7.3 Read Replicas

7.3.1 Introducing Read Replicas

Introduction

Only RDS for SQL Server 2019 Enterprise Edition and 2017 Enterprise Edition support read replicas.

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To expand the DB instance read ability to offload read pressure on the database, you can create one or more read replicas in a region. These read replicas can process a large number of read requests and increase application throughput. You need to separately configure connection addresses of the primary DB instance and each read replica on your applications so that all read requests can be sent to read replicas and write requests to the primary DB instance.

A read replica uses a single-node architecture (without a slave node). Changes to the primary DB instance are also automatically synchronized to all associated read replicas through the native SQL Server replication function. The synchronization is not affected by network latency. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Functions

- Specifications of read replicas can be different from those of the primary DB instance, and can be changed at any time.
- You do not need to maintain separate database accounts or databases. They are synchronized from the primary DB instance.
- Read replicas support system performance monitoring.
RDS provides up to 20 monitoring metrics, including storage space, IOPS, the number of database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on DB instances.
- Read replicas do not support automated backups or manual backups.
- Read replicas do not support restoration from backups to new, existing, or original read replicas.
- Data cannot be migrated to read replicas.

Constraints

A maximum of five read replicas can be created for each primary DB instance.

Creating and Managing a Read Replica

- [Creating a Read Replica](#)
- [Managing a Read Replica](#)

7.3.2 Creating a Read Replica

Scenarios

Read replicas are used to enhance the read capabilities and reduce the load on primary DB instances.

After a DB instance has been created, you can create read replicas for it.

 NOTE

A maximum of five read replicas can be created for a primary DB instance.

Only RDS for SQL Server 2019 Enterprise Edition and 2017 Enterprise Edition support read replicas.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Create Read Replica** in the **Operation** column.

Alternatively, click the target DB instance. In the DB instance topology, click  under the primary DB instance to create read replicas.

Step 5 On the displayed page, configure information about the DB instance and click **Next**.

Table 7-1 Basic information

Parameter	Description
Region	By default, read replicas are in the same region as the primary DB instance.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
DB Engine	Same as the DB engine of the primary DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of the primary DB instance by default and cannot be changed.
AZ	RDS allows you to deploy primary DB instances and read replicas in a single AZ or across AZs to improve reliability.

Table 7-2 Instance specifications

Parameter	Description
Instance Class	Refers to the CPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS. For details about instance classes, see DB Instance Classes. After a DB instance is created, you can change its CPU and memory. For details, see Changing a DB Instance Class.
Storage Type	Determines the DB instance read/write speed. The higher the maximum throughput is, the higher the DB instance read/write speed can be. • High I/O: supports a maximum throughput of 150 MB/s. • Ultra-high I/O: supports a maximum throughput of 350 MB/s.
Storage Space	Contains the file system overhead required for inode, reserved block, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.

Table 7-3 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.
Subnet	Same as the primary DB instance's subnet. A floating IP address is automatically assigned when you create a read replica. You can also enter an unused floating IP address in the subnet CIDR block. After the read replica is created, the floating IP address cannot be changed.
Security Group	Same as the primary DB instance's VPC.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- Otherwise, click **Submit**.

Step 7 After a read replica has been created, you can view and manage it on the **Instance Management** page by clicking on the left of the DB instance to which it belongs.

Alternatively, click the target DB instance. In the DB instance topology, click the target read replica. You can view and manage it in the displayed pane.

----End

Follow-up Operations

Managing a Read Replica

7.3.3 Managing a Read Replica

Entering the Management Interface Through a Read Replica

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** In the DB instance list, click  to expand the DB instance details and click the target read replica to go to the **Basic Information** page.
- End

Entering the Management Interface Through a Primary DB Instance

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.
- Step 5** In the DB instance topology, click the target read replica. You can view and manage it in the displayed pane.
- End

Deleting a Read Replica

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** In the DB instance list, click  in front of a DB instance, locate the read replica to be deleted, and choose **More > Delete** in the **Operation** column.
- End

7.4 Backups and Restorations

7.4.1 Working with Backups

RDS supports DB instance backups and restorations to ensure data reliability.

Automated Backups

Automated backups are created during the backup period of your DB instances. RDS saves automated backups based on the retention period you specified. If necessary, you can restore to any point in time during your backup retention period. For details, see [Configuring an Automated Backup Policy](#).

Manual Backups

Manual backups are user-initiated full backups of DB instances. They are retained until you delete them manually. For details, see [Creating a Manual Backup](#).

Downloading a Backup File

You can download a full backup file for local data backup or restoration. For details, see [Downloading a Backup File](#).

7.4.2 Configuring an Automated Backup Policy

Scenarios

When you create a DB instance, an automated backup policy is enabled by default. For security purposes, the automated backup policy cannot be disabled. After the DB instance is created, you can customize the automated backup policy as required and then RDS backs up data based on the automated backup policy you have set.

RDS automatically backs up data at the DB instance level. If a database is faulty or data is damaged, you can restore it from backups to ensure data reliability. Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects the database read and write performance, you are advised to set the automated backup time window to off-peak hours.

The default automated backup policy is as follows:

- Retention period: 7 days
- Time window: An hour within 24 hours, such as 01:00-02:00 or 12:00-13:00. The backup time is configured based on UTC time and is adjusted for daylight saving time, which changes at different times depending on the time zone.
- Backup cycle: Each day of the week

Modifying an Automated Backup Policy

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Backups & Restorations** page, click **Modify Backup Policy**.

- **Retention Period** refers to the number of days that your automated backups can be retained. Increasing the retention period will improve data reliability.
- If you shorten the retention period, the new backup policy takes effect for all backup files. The backup files that have expired will be deleted.
- The backup retention period indicates the number of days you want automated full and incremental backups of your DB instance to be retained. It ranges from 1–732 days. The backup time window is one hour. You are advised to select an off-peak time window for automated backups. By default, each day of the week is selected for **Backup Cycle** and you can change it. At least one day must be selected.

Step 6 Click **OK**.

----End

7.4.3 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups of a running primary DB instance. You can use these backups to restore data.

NOTE

When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.

Method 1

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and choose **More > Create Backup** in the **Operation** column.

Step 5 In the displayed dialog box, enter a backup name, select a target database for which to create the backup, and enter a description. Then, click **OK**. If you want to cancel the backup creation task, click **Cancel**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- The time required for creating a manual backup depends on the amount of data.

 NOTE

System databases are backed up by default.

Step 6 After a manual backup has been created, you can view and manage it on the **Backup Management** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.

----End

Method 2

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 On the **Backups & Restorations** page, click **Create Backup**. In the displayed dialog box, enter a backup name, select a target database for which to create the backup, and enter a description. Then, click **OK**.

- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- The time required for creating a manual backup depends on the amount of data.

 NOTE

System databases are backed up by default.

Step 6 After a manual backup has been created, you can view and manage it on the **Backup Management** page.

Alternatively, click the target DB instance. On the **Backups & Restorations** page, you can view and manage the manual backups.

----End

7.4.4 Downloading a Backup File

Scenarios

This section describes how to download a manual backup or an automated backup to a local device and restore data from the backup file.

Using Download URL

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Backup Management** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 In the displayed dialog box, select a method to download backup data.

In the displayed dialog box, select **Use Download URL** for **Download Method**, click  to copy the URL, and click **OK**.

For Microsoft SQL Server DB instances, the URLs of all the backup files are displayed. You can download the backup files of a specific database.

- You can use other download tools to download backup files.
- You can also run the following command to download backup files:

```
wget -O FILE_NAME --no-check-certificate "DOWNLOAD_URL"
```

Variables in the commands are as follows:

FILE_NAME: indicates the new backup file name after the download is successful. The original backup file name may be too long and exceed the maximum characters allowed by the client file system. You are advised to use the **-O** argument with **wget** to rename the backup file.

DOWNLOAD_URL: indicates the path of the backup file to be downloaded. If the path contains special characters, escape is required.

----End

7.4.5 Restoring from Backup Files to RDS for SQL Server

Scenarios

This section describes how to use an automated or manual backup to restore a DB instance to the status when the backup was created.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Backups & Restorations** page, locate the target backup and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page. On the displayed page, choose **Backups & Restorations**. On the displayed page, select the target backup to be restored and click **Restore** in the **Operation** column.

Step 5 In the displayed dialog box, configure required information and click **OK**.

1. Select a restoration method and click **OK**.

- Create New Instance

The **Create New Instance** page is displayed.

- The DB engine of the new DB instance is the same as that of the original DB instance and cannot be changed.
- Storage space of the new DB instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.

- Restore to Original

NOTICE

- If the DB instance for which the backup is created has been deleted, data cannot be restored to the original DB instance.
- Restoring to the original DB instance will overwrite data on it and cause the original DB instance to be unavailable during the restoration.

- Restore to Existing

NOTICE

- Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable.
- To restore backup data to an existing DB instance, the selected DB instance must be in the same VPC as the original DB instance and must have the same DB engine and the same or later version than the original DB instance.
- Ensure that the storage space of the selected DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.
- DB instances with the TDE function enabled cannot be restored from backups to existing DB instances.

Select an existing DB instance and click **Next**.

2. Select the databases to be restored. You can rename these databases as required. If you do not enter a new name, the original database name will be used.

 NOTE

- The new database names must be different from each other and must be different from the original database names.
- The new database names cannot contain the following fields (case-insensitive): rdsadmin, master, msdb, tempdb, model, and resource.
- Each database name consists of 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Step 6 View the restoration result. The result depends on which restoration method was selected:

- Create New Instance

A new DB instance is created using the backup data. The status of the DB instance changes from **Creating** to **Available**.

The new DB instance is independent from the original one.

- Restore to Original

On the **Instance Management** page, the status of the target existing DB instance changes from **Restoring** to **Available**. If the target existing DB instance contains read replicas, the read replica status is the same as the target existing DB instance status.

After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

On the **Instance Management** page, the status of the DB instance changes from **Restoring** to **Available**.

----End

7.4.6 Replicating a Backup

Scenarios

This section describes how to replicate a manual or an automated backup. The new backup name must be different from the original backup name.

Constraints

You can replicate backups and use them only within the same region.

Backup Retention Policy

- If you want to retain the automated backups for a long time, you can replicate them to generate manual backups, which will be always retained until you delete them.
- If the storage occupied by manual backups exceeds the provisioned free backup storage, additional storage costs may incur.

Procedure

Step 1 Log in to the management console.

- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance. On the **Backups & Restorations** page, locate the target backup to be replicated and click **Replicate** in the **Operation** column.
- Step 5** In the displayed dialog box, enter a new backup name and description and click **OK**.
- The backup name must consist of 4 to 64 characters and start with a letter. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).
 - The description consists of a maximum of 256 characters and cannot contain the following special characters: > ! < " & ' =
- Step 6** After the new backup has been created, you can view and manage it on the **Backup Management** page.
- End

7.5 Parameter Template Management

7.5.1 Creating a Parameter Template

You can use database parameter templates to manage the DB engine configuration. A database parameter template acts as a container for engine configuration values that can be applied to one or more DB instances.

This default template contains DB engine defaults and system defaults that are configured based on the engine, compute class, and allocated storage of the instance. Default parameter templates cannot be modified, but you can create your own parameter template to change parameter settings.

NOTICE

Not all of the DB engine parameters in a custom parameter template can be changed.

If you want to use a custom parameter template, you simply create a parameter template and select it when you create a DB instance or apply it to an existing DB instance following the instructions provided in section [Applying a Parameter Template](#).

When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template following the instructions provided in section [Replicating a Parameter Template](#).

The following are the key points you should know when using parameters in a parameter template:

- When you change a dynamic parameter value in a parameter template and save the change, the change takes effect immediately. When you change a static parameter value in a parameter template and save the change, the change will take effect only after you manually reboot the DB instances that the parameter template applies to.
- Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when modifying database parameters and you need to back up data before modifying parameters in a parameter template. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.

 NOTE

RDS does not share parameter template quotas with DDS.

You can create a maximum of 100 parameter templates for RDS DB instances. All RDS DB engines share the parameter template quota.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Create Parameter Template**.

Step 5 In the displayed dialog box, configure required information and click **OK**.

- Select a DB engine for the parameter template.
- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

----End

7.5.2 Modifying Parameters

You can modify parameters in a custom parameter template to optimize RDS database performance.

You can only change parameter values in custom parameter templates. You cannot change the parameter values in default parameter templates.

When you modify a parameter, the time when the modification takes effect depends on the type of the parameter.

The RDS console displays the statuses of DB instances that the parameter template applies to. For example, if the DB instance has not yet used the latest modifications made to its parameter template, its status is **Parameter change**.

Pending reboot. Manually reboot the DB instance for the latest modifications to take effect for that DB instance.

 NOTE

RDS has default parameter templates whose parameter values cannot be changed. You can view these parameter values by clicking the default parameter templates. If a custom parameter template is set incorrectly, the database startup may fail. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Modifying Parameter Template Parameters

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the **Parameters** page, modify parameters as required.

Relevant parameters are as follows:

- Set **remote access** to **0** (default value) to prevent locally stored procedures from running on a remote server and remotely stored procedures from running on a local server.
- The **max server memory (MB)** parameter indicates the server memory. The default value equals to the OS memory (MB) minus 520 (MB). Its minimum value is 1024 MB.

Available operations are as follows:

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

Step 6 After the parameter values are modified, you can click **View Change History** to view the modification details.

Step 7 The modifications take effect only after you apply the parameter template to DB instances. For details, see [Applying a Parameter Template](#).

Step 8 View the status of the DB instance to which the parameter template was applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- A DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)

- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

Modifying Instance Parameters

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, modify parameters as required.

Relevant parameters are as follows:

- Set **remote access** to **0** (default value) to prevent locally stored procedures from running on a remote server and remotely stored procedures from running on a local server.
- The **max server memory (MB)** parameter indicates the server memory. The default value equals to the OS memory (MB) minus 520 (MB). Its minimum value is 1024 MB.

Available operations are **Save**, **Cancel**, and **Preview**.

- To save the modifications, click **Save**.
- To cancel the modifications, click **Cancel**.
- To preview the modifications, click **Preview**.

NOTICE

In the **Effective upon Reboot** column:

- If the value is **Yes** and the DB instance status on the **Instance Management** page is **Parameter change. Pending reboot**, a reboot is required the modifications to take effect.
 - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
 - If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.
- If the value is **No**, the modifications take effect immediately.

After parameters are modified, you can view parameter change history by referring to section [Viewing Parameter Change History](#).

----End

7.5.3 Exporting a Parameter Template

Scenarios

- You can export a parameter template of a DB instance for future use. You can also apply the exported parameter template to DB instances by referring to [Applying a Parameter Template](#).
- You can export the parameter template information (parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analysis.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

- Exporting to a custom template
In the displayed dialog box, configure required information and click **OK**.

NOTE

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: > ! < " & ' =

After the parameter template is exported, a new template is generated in the list on the **Parameter Template Management** page.

- Exporting to a file
The parameter template information (parameter names, values, and descriptions) of a DB instance is exported to a CSV file. In the displayed dialog box, enter the file name and click **OK**.

NOTE

The file name must start with a letter and consist of 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

----End

7.5.4 Comparing Parameter Templates

Scenarios

You can compare DB instance parameters with a parameter template that uses the same DB engine to understand the differences of parameter settings.

You can also compare default parameter templates that use the same DB engine to understand the differences of parameter settings.

Comparing Instance Parameters with a Parameter Template

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.
- Step 6** In the displayed dialog box, select a parameter template to be compared and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

Comparing Parameter Templates

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.
- Step 5** In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

7.5.5 Viewing Parameter Change History

Scenarios

You can view the change history of DB instance parameters or custom parameter templates.

 NOTE

An exported or custom parameter template has initially a blank change history.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

You can apply the parameter template to DB instances as required by referring to section [Applying a Parameter Template](#).

----End

Viewing Change History of a Parameter Template

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the displayed page, choose **Change History** in the navigation pane on the left.

You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

----End

7.5.6 Replicating a Parameter Template

Scenarios

You can replicate a parameter template you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template.

After a parameter template is replicated, it takes about 5 minutes before the new template is displayed.

Default parameter templates cannot be replicated, but you can create parameter templates based on the default ones.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

Alternatively, click the target DB instance on the **Instance Management** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

Step 5 In the displayed dialog box, configure required information and click **OK**.

- The template name must consist of 1 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.).
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Template Management** page.

----End

7.5.7 Resetting a Parameter Template

Scenarios

You can reset all parameters in a custom parameter template to their default settings.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template and choose **More > Reset** in the **Operation** column.

Step 5 Click **Yes**.

Step 6 The modifications take effect only after you apply the parameter template to DB instances. For details, see [Applying a Parameter Template](#).

Step 7 View the status of the DB instance to which the parameter template is applied.

If the DB instance status is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.

- The DB instance reboot caused by instance class changes will not make parameter modifications take effect.
- If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications to take effect. (For primary/standby DB instances, the parameter modifications are also applied to the standby DB instance.)
- If you have modified parameters of a read replica, you need to reboot the read replica for the modifications to take effect.

----End

7.5.8 Applying a Parameter Template

Scenarios

You can apply parameter templates to DB instances as needed. A parameter template can be applied only to DB instances of the same version.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, perform the following operations based on the type of the parameter template to be applied:

- If you intend to apply a default parameter template to DB instances, click **Default Templates**, locate the target parameter template, and click **Apply** in the **Operation** column.
- If you intend to apply a custom parameter template to DB instances, click **Custom Templates**, locate the target parameter template, and choose **More > Apply** in the **Operation** column.

A parameter template can be applied to one or more DB instances.

Step 5 In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

After the parameter template is successfully applied, you can view the application records by referring to section [Viewing Application Records of a Parameter Template](#).

----End

7.5.9 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Parameter Template Management** in the navigation pane on the left.

Step 5 On the **Default Templates** page, locate the target parameter template and click **View Application Record** in the **Operation** column. Alternatively, on the **Custom Templates** page, choose **More > View Application Record** in the **Operation** column.

You can view the name or ID of the DB instance that the parameter template applies to, as well as the application status, application time, and failure cause (if failed).

----End

7.5.10 Modifying a Parameter Template Description

Scenarios

You can modify the description of a parameter template you have created.

NOTE

You cannot modify the description of a default parameter template.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**.
Locate the target parameter template and click  in the **Description** column.

Step 5 Enter a new description. You can click  to submit or to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain the following special characters: > ! < " & ' =

- After the modification is successful, you can view the new description in the **Description** column of the parameter template list.

----End

7.5.11 Deleting a Parameter Template

Scenarios

You can delete a custom parameter template that is no longer in use.

NOTICE

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
- Default parameter templates cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Parameter Template Management** page, click **Custom Templates**. Locate the target parameter template to be deleted and choose **More > Delete** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

----End

7.6 Connection Management

7.6.1 Configuring and Changing a Floating IP Address

Scenarios

You can change floating IP addresses after migrating on-premises databases or other cloud databases to RDS.

Constraints

After a floating IP address is changed, the domain name needs to be resolved again. This operation takes several minutes and may interrupt database connections. Therefore, you are advised to change a floating IP address during off-peak hours.

Configuring a Floating IP Address

You can use a self-configured floating IP address when creating a DB instance.

Changing a Floating IP Address

You can change the floating IP address of an existing DB instance.

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the **Connection Information** area on the **Basic Information** page, click **Change** in the **Floating IP Address** field.
- Step 6** In the displayed dialog box, enter a new floating IP address. Click **OK**.

In the in-use IP address list, the IP addresses whose statuses are **To be used** are occupied and cannot be used.

----End

7.6.2 Binding and Unbinding an EIP

Scenarios

By default, a DB instance is not publicly accessible (not bound with an EIP) after being created. You can bind an EIP to a DB instance for public accessibility, and you can unbind the EIP from the DB instance later if needed.

NOTICE

To ensure that the DB instance is accessible, the security group associated with the instance must allow access over the database port. For example, if the database port is 8635, ensure that the security group allows access over the 8635 port.

Prerequisites

- You can bind an EIP to a primary DB instance or a read replica only.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the DB instance first before binding a new EIP to it.

Binding an EIP

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.

- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** On the **EIPs** page, click **Bind EIP**.
- Step 6** In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **OK**. If no available EIPs are displayed, click **View EIP** and obtain an EIP.
- Step 7** On the **EIPs** page of the RDS console, view the EIP that has been bound to the DB instance.

You can also view the progress and result of binding an EIP to a DB instance on the **Task Center** page.

To unbind the EIP from the DB instance, see [Unbinding an EIP](#).

----End

Unbinding an EIP

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the DB instance that has an EIP bound.
- Step 5** On the **EIPs** page, locate the target EIP to be unbound and click **Unbind**. In the displayed dialog box, click **Yes**.
- Step 6** On the **EIPs** page, view the unbinding result.

You can also view the progress and result of unbinding an EIP from a DB instance on the **Task Center** page.

To bind an EIP to the DB instance again, see [Binding an EIP](#).

----End

7.6.3 Changing a Database Port

Scenarios

This section describes how to change the database port of a primary DB instance or a read replica. For primary/standby DB instances, changing the database port of the primary DB instance will cause the database port of the standby DB instance to also be changed accordingly.

Procedure

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target DB instance or click  first and then click the target read replica.

Step 5 In the **Connection Information** area on the **Basic Information** page, click  in the **Database Port** field.

 NOTE

The DB instance port is 1433 by default or it can be set to a value from 2100 to 9500, excluding 5355 and 5985.

- To submit the change, click .
 - In the dialog box, click **Yes**.
 - i. If you change the database port of the primary DB instance, that of the standby DB instance will also be changed and both DB instances will be rebooted.
 - ii. If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will reboot.

 NOTE

For RDS for SQL Server, only 2017 Enterprise Edition supports read replicas.

- iii. This process takes 1-5 minutes.
 - In the dialog box, click **No** to cancel the modification.

Step 6 View the results on the **Basic Information** page.

-----End

7.7 Data Security

7.7.1 Resetting the Administrator Password

Scenarios

You can reset the administrator password only through the primary DB instance.

You can also reset the password of your database account when using RDS.

 NOTE

- If you have changed the administrator password of the primary DB instance, the administrator passwords of the standby DB instance and read replicas (if any) will also be changed accordingly.
- The time required for the new password to take effect depends on the amount of service data currently being processed by the primary DB instance.
- To protect against brute force hacking attempts and ensure system security, change your password periodically, such as every three or six months.

Method 1

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, locate the target DB instance and choose **More > Reset Password** in the **Operation** column.
- Step 5** Enter and confirm a new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#\$%^*_+?,.). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.
- To cancel the reset operation, click **Cancel**.

----End

Method 2

- Step 1** Log in to the management console.
- Step 2** Click  in the upper left corner and select a region and a project.
- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the **DB Information** area on the **Basic Information** page, click **Reset Password** in the **Administrator** field. In the displayed dialog box, enter and confirm a new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

The new password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#\$%^*_+?,.). Enter a strong password and periodically change it for security reasons.

- To submit the new password, click **OK**.

- To cancel the reset operation, click **Cancel**.

----End

7.7.2 Changing a Security Group

Scenarios

This section describes how to change the security group of a primary DB instance or read replica. For primary/standby DB instances, changing the security group of the primary DB instance will cause the security group of the standby DB instance to also be changed accordingly.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, click the target primary DB instance or read replica.

Step 5 In the **Connection Information** area on the **Basic Information** page, click  in the **Security Group** field.

- To submit the change, click .
- To cancel the change, click .

Step 6 Changing the security group takes 1 to 3 minutes. Click  in the upper right corner on the **Basic Information** page to view the results.

----End

7.8 Metrics and Alarms

7.8.1 Configuring Displayed Metrics

The RDS Agent monitors RDS DB instances and collects monitoring metrics only.

Description

This section describes the metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS.

Namespace

SYS.RDS

DB Instance Monitoring Metrics

- [Table 7-4](#) lists details about ECS metrics.

Table 7-4 ECS performance metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0%–100%	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0%–100%	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0%–100%	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds053_avg_disk_queue_length	Average Disk Queue Length	Number of processes to be written into the monitored object	≥ 0	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds054_db_connections_in_use	Database Connections in Use	Number of database connections in use	≥0 counts	Monitored object: database Monitored instance type: Microsoft SQL Server instance	1 minute

- **Table 7-5** lists the performance metrics of Microsoft SQL Server databases.

Table 7-5 Database performance metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100%	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-1	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds005_bytes_output	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0–1	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
rds053_avg_disk_queue_length	Average Disk Queue Length	Number of processes to be written into the monitored object	≥ 0	Monitored object: ECS Monitored instance type: Microsoft SQL Server instance	1 minute
rds054_db_connections_in_use	Database Connections in Use	Number of database connections in use	≥ 0 counts	Monitored object: database Monitored instance type: Microsoft SQL Server instance	1 minute

Dimension

Key	Value
rds_instance_sqlserver_id	Microsoft SQL Server DB instance node ID

7.8.2 Setting Alarm Rules

Scenarios

You can set alarm rules by referring to [Setting Alarm Rules](#) to customize the monitored objects and notification policies and keep track of the RDS running status.

The RDS alarm rules include alarm rule names, services, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Setting Alarm Rules

Step 1 Log in to the management console.

Step 2 Click **Service List** and choose **Management & Governance > Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Alarm Management > Alarm Rules**.

Step 4 On the displayed **Alarm Rules** page, click **Create Alarm Rule**.

----End

7.8.3 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS monitoring metrics on the management console.

Monitored data takes some time before it can be displayed. The RDS status displayed on the Cloud Eye console is about 5 to 10 minutes delayed. When you create a new RDS DB instance, it takes 5 to 10 minutes before monitoring data is displayed on Cloud Eye.

Prerequisites

- RDS is running properly.
Monitoring metrics of the RDS DB instances that are faulty or have been deleted are not displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to normal.

NOTE

If an RDS DB instance has been faulty for 24 hours, Cloud Eye considers it to no longer exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- RDS has been running properly for about 10 minutes.
For a newly created RDS DB instance, you need to wait a bit before you can view the metrics.

Procedure

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 On the **Instance Management** page, locate the target DB instance and click **View Metric** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metric** in the upper right corner of the page to go to the Cloud Eye console.

Step 5 On the Cloud Eye console, view monitoring metrics of the primary DB instance.

Cloud Eye can monitor performance metrics from the last 1 hour, 3 hours, 12 hours, 1 day, and 7 days.

----End

7.9 Interconnection with CTS

7.9.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 7-6 RDS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or a read replica, or restoring data to a new DB instance (using the console, Trove APIs, or open APIs)	instance	createInstance
Scaling up storage space and changing instance class (using the console, Trove APIs, or open APIs)	instance	instanceAction
Rebooting a DB instance (using the console, Trove APIs, or open APIs)	instance	instanceRestart
Restoring to the original DB instance (using the console, Trove APIs, or open APIs)	instance	instanceRestore
Renaming a DB instance (using the console)	instance	instanceRename
Resetting the password (using the console)	instance	resetPassword
Setting the database version parameters (using open APIs)	instance	setDBParameters
Resetting the database version parameters (using open APIs)	instance	resetDBParameters
Enabling, modifying, or disabling the backup policy (using the console or open APIs)	instance	setBackupPolicy
Changing a database port (using the console)	instance	changeInstancePort
Binding and unbinding an EIP (using the console)	instance	setOrResetPublicIP
Modifying a security group (using the console)	instance	modifySecurityGroup
Adding a tag (using the console or open APIs)	instance	createTag

Operation	Resource Type	Trace Name
Deleting a tag (using the console or open APIs)	instance	deleteTag
Modifying a tag (using the console or open APIs)	instance	modifyTag
Deleting a DB instance (using the console, Trove APIs, or open APIs)	instance	deleteInstance
Enabling TDE for a Microsoft SQL Server DB instance (using the console)	instance	sqlserverOpenTDE
Performing a primary/standby switchover (using the console)	instance	instanceFailOver
Changing the replication mode (using the console)	instance	instanceFailOver-Mode
Changing the failover priority (using the console)	instance	instanceFailOver-Strategy
Changing a DB instance type from single to primary/standby (using the console, Trove APIs, or open APIs)	instance	modifySingleToHaIn-stance
Creating a backup (using the console or open APIs)	backup	createManualSnap-shot
Replicating a backup (using the console)	backup	copySnapshot
Download a backup (using the console or open APIs)	backup	downLoadSnapshot
Deleting a backup (using the console or open APIs)	backup	deleteManualSnap-shot
Creating a parameter template (using the console or Trove APIs)	parameterGroup	createParameterGrou-p
Modifying parameters in a parameter template (using the console or Trove APIs)	parameterGroup	updateParameterGro-up
Deleting a parameter template (using the console or Trove APIs)	parameterGroup	deleteParameterGrou-p
Replicating a parameter template (using the console)	parameterGroup	copyParameterGroup
Resetting a parameter template (using the console)	parameterGroup	resetParameterGroup

Operation	Resource Type	Trace Name
Comparing parameter templates (using the console)	parameterGroup	compareParameterGroup
Applying a parameter template (using the console)	parameterGroup	applyParameterGroup
Saving parameters in a parameter template (using the console)	parameterGroup	saveParameterGroup
Deleting a frozen DB instance (using the console)	all	deleteInstance
Freezing a DB instance (using the console)	all	rdsfreezeInstance

7.9.2 Viewing Tracing Events

Scenarios

After CTS is enabled, operations on cloud resources are recorded. You can view the operation records of the last 7 days on the CTS console.

This section describes how to query the operation records of last 7 days on the CTS console.

Procedure

- Step 1** Log in to the management console.
- Step 2** Choose **Management & Governance > Cloud Trace Service**.
- Step 3** Choose **Trace List** in the navigation pane on the left.
- Step 4** Filter conditions to query traces. The details are described as follows:
 - **Trace Source, Resource Type, and Search By:** Select a filter from the drop-down list.
When you select **Resource ID** for **Search By**, you also need to select or enter a resource ID.
 - **Operator:** Select a specific operator from the drop-down list.
 - **Trace Status:** Available options include **All trace statuses**, **Normal**, **Warning**, and **Incident**. You can only select one of them.
 - In the upper right corner of the page, you can specify a time range for querying traces.
- Step 5** Select the search criteria, and click **Query**.
- Step 6** Click  on the left of the required trace to expand its details.
- Step 7** Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.

Step 8 Click **Export** on the right. CTS exports traces collected in the past seven days to a CSV file. The CSV file contains all information related to traces on the management console.

For details about key fields in the trace structure, see sections "Trace Structure" and "Trace Examples" in the *Cloud Trace Service User Guide*.

----End

7.10 Task Center

7.10.1 Viewing a Task

You can view the progress and results of tasks on the **Task Center** page.

NOTE

You can view and manage the following tasks:

- Creating DB instances
- Creating read replicas
- Changing single DB instances to primary/standby
- Scaling up storage space
- Binding EIPs to DB instances
- Unbinding EIPs from DB instances
- Switching primary/standby DB instances
- Rebooting DB instances
- Restoring data to new DB instances

Viewing an Instant Task

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. Locate the target task and view its details on the displayed **Instant Tasks** page.

- To identify the target task, you can use the task name and DB instance name/ID or enter the target task name in the search box in the upper right corner.
- You can click  in the upper right corner to view the progress and status of tasks in a specific period. The default period is seven days.
The task list shows tasks that have been executed in the past 30 days.
- You can view instant tasks in the following statuses:
 - Running
 - Completed
 - Failed

- View the task creation and completion time.

----End

Viewing a Scheduled Task

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, view the task progress and results.

- To identify the target task, you can use the DB instance name/ID or enter the target DB instance ID in the search box in the upper right corner.
- You can view the scheduled tasks in the following statuses:
 - Running
 - Completed
 - Failed
 - Canceled
 - To be executed
 - To be authorized

----End

7.10.2 Deleting a Task Record

You can delete task records so that they are no longer displayed in the task list. This operation only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

NOTICE

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Deleting an Instant Task Record

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed **Instant Tasks** page, locate the target task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

You can delete the records of instant tasks in any of the following statuses:

- Running
- Completed
- Failed

----End

Deleting a Scheduled Task Record

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the target task record to be deleted and check whether the task status is **To be executed** or **To be authorized**.

- If yes, go to [Step 5](#).
- If no, go to [Step 6](#).

Step 5 Click **Cancel** in the **Operation** column. In the displayed dialog box, click **Yes** to cancel the task. Then, click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the task record.

You can delete the records of scheduled tasks in any of the following statuses:

- Running
- Completed
- Failed
- Canceled
- To be executed
- To be authorized

----End

7.11 Usage of Stored Procedures

7.11.1 Creating a Database Account

Scenarios

You can use a stored procedure to create a login account. This account has all permissions of the rdsuser user on Microsoft SQL Server databases.

 NOTE

- The stored procedure can be executed only by the **rdsuser** user or the created account.
- The password must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#%^*_+?,.).

Prerequisites

An RDS for SQL Server DB instance has been connected.

Procedure

Run the following command to create an account. After the command is executed successfully, you can use the created account to log in.

```
EXEC master.dbo.rds_create_major_login @login='loginName',  
@password='password' ;
```

- **loginName**: login name of the created account.
- **password**: password of the created account.

Example

Run the following command to create an account whose name is **rdsuser1** and password is *********:

```
EXEC master.dbo.rds_create_major_login @login='rdsuser1', @password='*****';
```

7.11.2 Updating Information About Operators for Alerts and Jobs

Scenarios

You can use a stored procedure to update information about an operator (notification recipient) for use with alerts and jobs.

Prerequisites

An RDS Microsoft SQL Server DB instance has been connected. Connect to the DB instance through the Microsoft SQL Server client.

Procedure

Run the following commands to update information about the operator for the alert and job:

```
EXEC [msdb].[dbo].[rds_update_operator]  
  
@name ='name',  
  
@new_name = 'new_name',  
  
@enabled=enabled,
```

```

@email_address='email_address',
@pager_address= 'pager_number',
@weekday_pager_start_time= weekday_pager_start_time,
@weekday_pager_end_time= weekday_pager_end_time,
@saturday_pager_start_time= saturday_pager_start_time,
@saturday_pager_end_time= saturday_pager_end_time,
@sunday_pager_start_time= sunday_pager_start_time,
@sunday_pager_end_time= sunday_pager_end_time,
@pager_days= pager_days,
@netsend_address ='netsend_address',
@category_name='category'

```

Table 7-7 Parameter description

Parameter	Description
'name'	The name of the operator to modify. This name must be unique and cannot contain the percent (%) character. name is sysname , with no default.
'new_name'	The new name for the operator. This name must be unique. new_name is sysname , with a default of NULL .
enabled	The current status of the operator. enabled is tinyint , with a default of 1 (enabled). If the value is 0 , the operator is not enabled and does not receive notifications.
'email_address'	The e-mail address of the operator. This string is passed directly to the e-mail system. email_address is nvarchar(100) , with a default of NULL .
'pager_number'	The pager address of the operator. This string is passed directly to the e-mail system. pager_number is nvarchar(100) , with a default of NULL .
weekday_pager_start_time	The time after which SQL Server Agent sends pager notification to the specified operator on the weekdays, from Monday through Friday. weekday_pager_start_time is int , with a default of 090000 , which indicates 9:00 A.M. on a 24-hour clock, and must be entered using the form HHMMSS.
weekday_pager_end_time	The time after which SQLServerAgent service no longer sends pager notification to the specified operator on the weekdays, from Monday through Friday. weekday_pager_end_time is int , with a default of 180000 , which indicates 6:00 P.M. on a 24-hour clock, and must be entered using the form HHMMSS.

Parameter	Description
saturday_pager_start_time	The time after which SQLServerAgent service sends pager notification to the specified operator on Saturdays. saturday_pager_start_time is int , with a default of 090000 , which indicates 9:00 A.M. on a 24-hour clock, and must be entered using the form HHMMSS.
saturday_pager_end_time	The time after which SQLServerAgent service no longer sends pager notification to the specified operator on Saturdays. saturday_pager_end_time is int , with a default of 180000 , which indicates 6:00 P.M. on a 24-hour clock, and must be entered using the form HHMMSS.
sunday_pager_start_time	The time after which SQLServerAgent service sends pager notification to the specified operator on Sundays. sunday_pager_start_time is int , with a default of 090000 , which indicates 9:00 A.M. on a 24-hour clock, and must be entered using the form HHMMSS.
sunday_pager_end_time	The time after which SQLServerAgent service no longer sends pager notification to the specified operator on Sundays. sunday_pager_end_time is int , with a default of 180000 , which indicates 6:00 P.M. on a 24-hour clock, and must be entered using the form HHMMSS.
pager_days	A number that indicates the days that the operator is available for pages (subject to the specified start/end times). pager_days is tinyint , with a default of 0 , indicating the operator is never available to receive a page. Valid values are from 0 through 127 . pager_days is calculated by adding the individual values for the required days. For example, from Monday through Friday is $2+4+8+16+32 = 62$. The following lists the value for each day of the week: • 1: indicates Sunday. • 2: indicates Monday. • 4: indicates Tuesday. • 8: indicates Wednesday. • 16: indicates Thursday. • 32: indicates Friday. • 64: indicates Saturday.
'netsend_address'	The network address of the operator to whom the network message is sent. netsend_address is nvarchar(100) , with a default of NULL .
'category'	The name of the category for this operator. category is sysname , with a default of NULL .

After the command is executed, the system displays the following information.

Commands completed successfully.

7.11.3 Removing Alerts

Scenarios

You can use a stored procedure to remove an alert.

Prerequisites

An RDS Microsoft SQL Server DB instance has been connected. Connect to the DB instance through the Microsoft SQL Server client.

Procedure

Run the following commands to remove an alert:

```
EXEC [msdb].[dbo].[rds_delete_alert]
```

```
@name='name'
```

Table 7-8 Parameter description

Parameter	Description
'name'	The name of the alert. name is sysname , with no default.

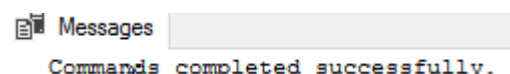
After the command is executed, the system displays the following information.

Commands completed successfully.

Example

```
EXEC [msdb].[dbo].[rds_delete_alert]
    @name='test'
```

The command output is as follows.



Messages
Commands completed successfully.

7.11.4 Removing SQL Server Agent Notification Definitions for Specific Alerts and Operators

Scenarios

You can use a stored procedure to remove a SQL Server Agent notification definition for a specific alert and operator.

Prerequisites

An RDS Microsoft SQL Server DB instance has been connected. Connect to the DB instance through the Microsoft SQL Server client.

Procedure

Run the following commands to remove the SQL Server Agent notification definition for a specific alert and operator:

```
EXEC [msdb].[dbo].[rds_delete_notification]
```

```
@alert_name = 'alert',
```

```
@operator_name = 'operator';
```

Table 7-9 Parameter description

Parameter	Description
'alert'	The name of the alert. alert is sysname , with no default.
'operator'	The name of the operator. operator is sysname , with no default.

After the command is executed, the system displays the following information.

Commands completed successfully.

Example

```
EXEC [msdb].[dbo].[rds_delete_notification]
@alert_name = 'alert',
@operator_name = N'TestOperator';
```

The command output is as follows.

Messages
Commands completed successfully.

7.11.5 Removing Operators

Scenarios

You can use a stored procedure to remove an operator.

Prerequisites

An RDS Microsoft SQL Server DB instance has been connected. Connect to the DB instance through the Microsoft SQL Server client.

Procedure

Run the following commands to remove an operator:

```
EXEC [msdb].[dbo].[rds_delete_operator]
```

```
@name='name'
```

@reassign_to_operator = 'reassign_operator';

Table 7-10 Parameter description

Parameter	Description
'name'	The name of the operator to delete. name is sysname , with no default.
'reassign_operator'	The name of an operator to whom the specified operator's alerts can be reassigned. reassign_operator is sysname , with a default of NULL .

After the command is executed, the system displays the following information.

Commands completed successfully.

Example

```
EXEC [msdb].[dbo].[rds_delete_operator]
    @name = N'HWTest01';
```

The command output is as follows.

 Messages
Commands completed successfully.

8 FAQs

8.1 Product Consulting

8.1.1 What Should I Pay Attention to When Using RDS?

1. DB instance operating systems (OSs) are invisible to you. Your applications can access a database only through an IP address and a port.
2. The backup files stored in and the Elastic Cloud Server (ECS) used by RDS are invisible to you. They are visible only to the RDS instance management system.
3. Before viewing the DB instance list, ensure that the region is the same as the region where the DB instance is purchased.
4. Precautions after purchasing RDS:
After purchasing RDS DB instances, you do not need to perform basic database O&M operations, such as applying HA and security patches. However, you must still pay attention to:
 - a. Whether the CPU, input/output operations per second (IOPS), and space of the RDS DB instance are sufficient. If any of these becomes insufficient, change the CPU/Memory or scale up the DB instance.
 - b. Whether the performance of the RDS DB instances is adequate, a large number of slow query SQL statements exist, SQL statements need to be optimized, or any indexes are redundant or missing.

8.1.2 What Is the Availability of RDS DB Instances?

Calculation formula for RDS DB instance availability:

DB instance availability = (1 – Failure duration/Total service duration) × 100%

8.1.3 Can I Use a Template to Create DB Instances?

Currently, DB instance cannot be created using a template.

8.1.4 What Are the Differences Between RDS and Other Database Solutions?

Table 8-1 Differences between RDS and other database solutions

Function	RDS	Self-Built Database Service
Service availability	For details, see Elastic Cloud Service User Guide.	Requires self-guarantee, primary/standby relationship setup, and RAID setup.
Data reliability	For details, see <i>Elastic Volume Service User Guide</i> .	Requires self-guarantee, primary/standby relationship setup, and RAID setup.
System security	Defends against Anti-DDoS attacks and promptly repairs database security vulnerabilities.	Requires procurement of expensive devices and software, as well as manual detection and repair of security vulnerabilities.
Database backup	Automated backup	You need to find backup storage space by yourself and periodically check whether backup data can be restored.
Hardware and software investment	Supports on-demand pricing without hardware and software investment.	Requires large investment in database servers.
System hosting	Not required.	The hosting cost is high.
Maintenance cost	Not required.	Full-time Database Administrators (DBAs) are required for maintenance, leading to high manpower costs.
Deployment and scaling	Supports elastic scaling, fast deployment, and on-demand enabling.	Requires procurement, deployment, and coordination of hardware.
Resource utilization	Bills users based on the resources actually used, resulting in high resource utilization.	Peak resource utilization is considered, leading to low resource usage.

8.1.5 Will My RDS DB Instances Be Affected by Other User Instances?

No. Your RDS DB instances and resources are isolated from other users' DB instances.

8.1.6 Does RDS Support Cross-AZ High Availability?

Yes. RDS supports cross-AZ high availability. When you create primary/standby DB instances, you can select different AZs for them.

8.1.7 Can RDS Primary/Standby DB Instances Be Changed to Single DB Instances?

No. RDS single DB instances can be changed to primary/standby DB instances, but not the other way around.

8.1.8 What Should I Do If Garbled Characters Are Displayed After SQL Query Results Are Exported to an Excel File?

The default code is utf8. You need to convert the default code to Unicode in the exported Excel file.

8.1.9 How Do I Create an AD Domain?

This section describes how to set up an Active Directory (AD) domain and how to add an RDS for SQL Server instance to the domain.

Procedure

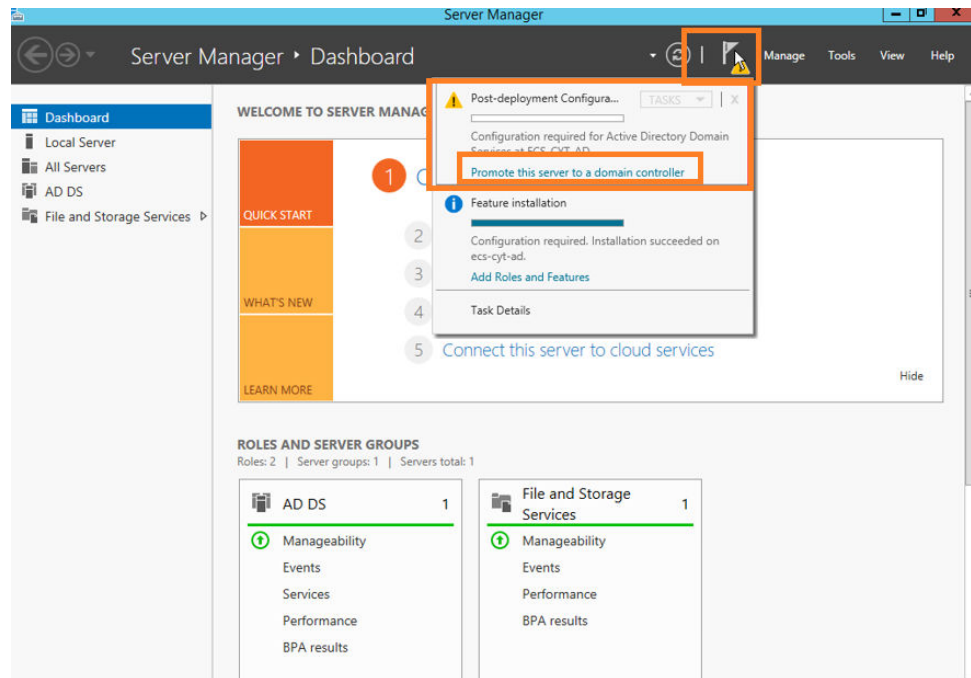
This section describes how to use Windows Server 2012 R2 to create a domain server.

NOTE

1. When you configure an AD domain information during the DB instance creation, do not configure or disable Group Policy Object (GPO) for your domain controller server, or the DB instance creation will fail.
2. If GPO is required, create an ECS and set up a new domain controller server with GPO disabled. Then, establish trust between your domain controller server and the new domain controller server. For details, contact customer service.

Step 1 Install an AD domain controller.

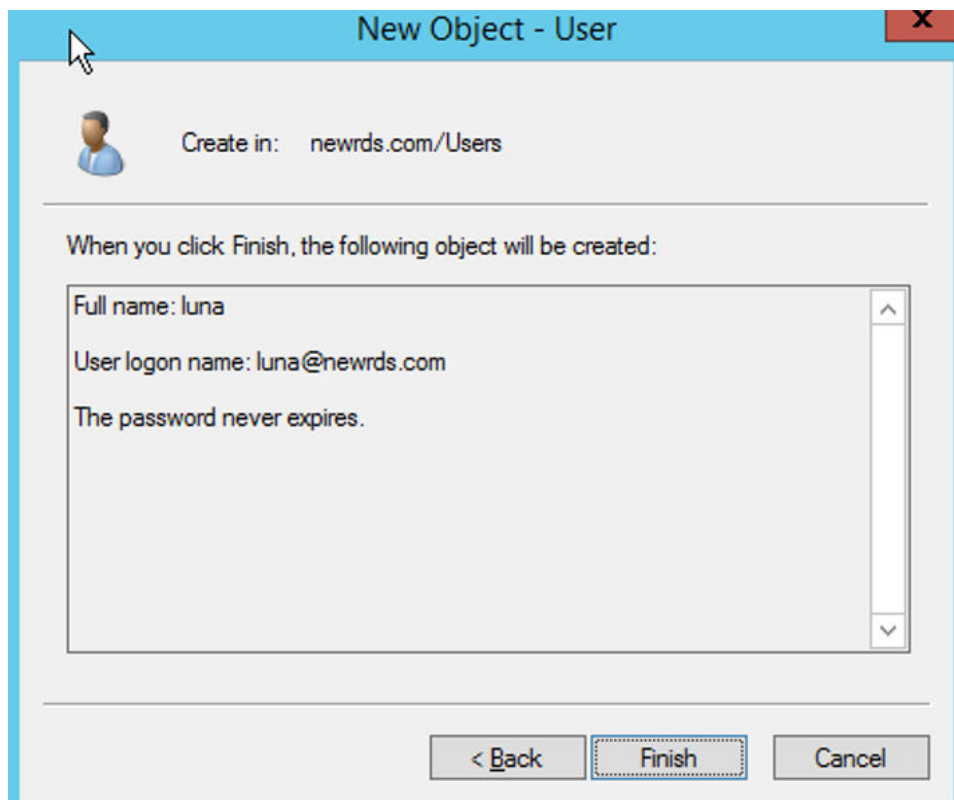
1. Open Server Manager and choose **Manage > Add Roles and Features**.
2. In the **Add Roles and Features Wizard** dialog box, click **Next** until the **Select server roles** page is displayed. Select **Active Directory Domain Services** and click **Add Features** in the displayed box.
3. Click **Next** until the **Confirm installation selections** page is displayed. Click **Install** to start the role installation process.
4. After the installation is complete, a yellow triangle icon is displayed. Click **Promote this server to a domain controller**. The **Active Directory Domain Services Configuration Wizard** window is displayed.



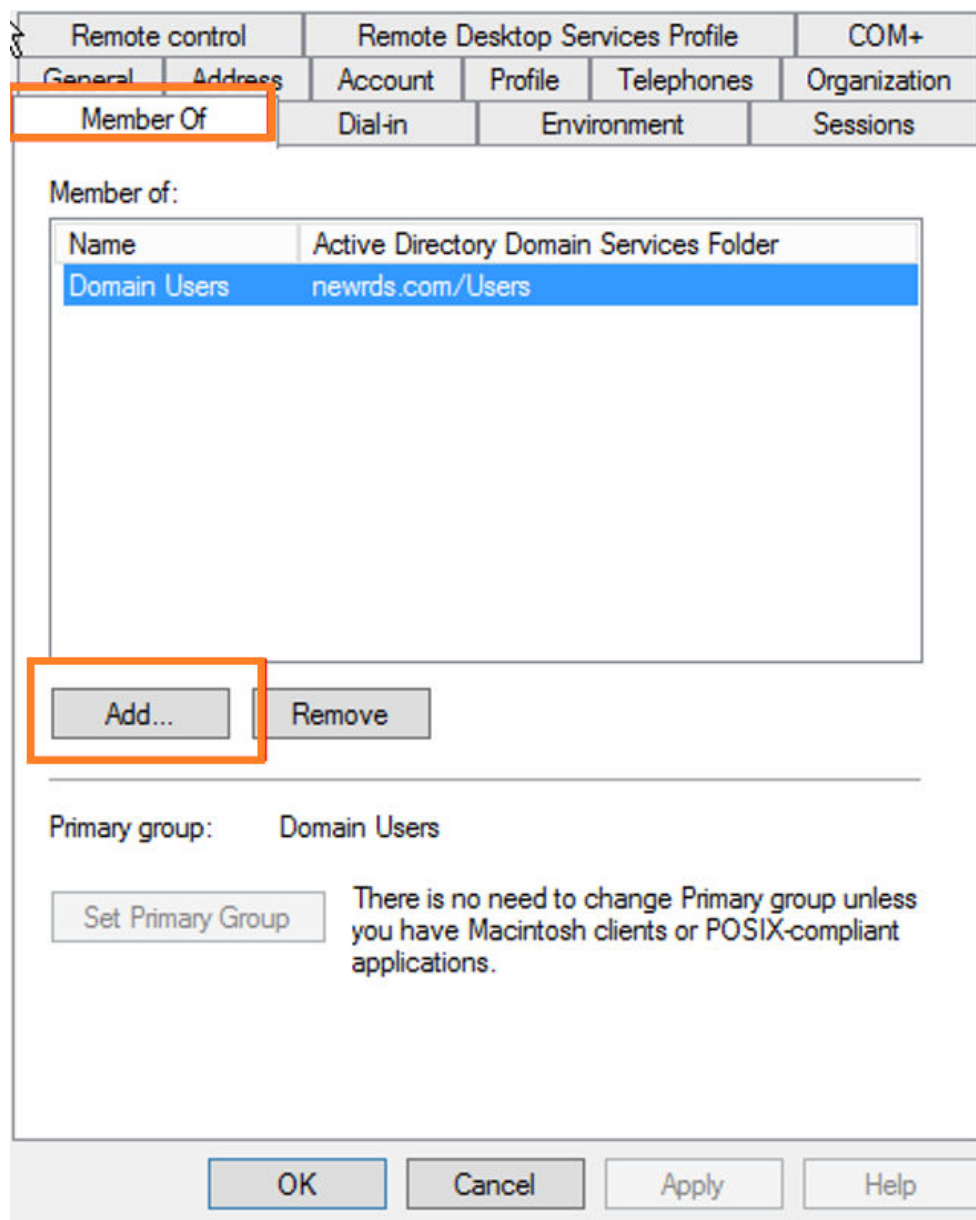
5. On the **Deployment Configuration** page, select **Add a new forest** and set a domain name, such as newrds.com.
6. Click **Next**. On the displayed page, enter the DSRM password (non-domain user).
7. Click **Next** until the **Prerequisites Check** page displayed. Click **Install**. After the installation is complete, the server automatically reboots.
8. Modify the DNS configuration of the network interface. Set the IP address of the active DNS server to the server's private IP address (such as 192.168.0.133).

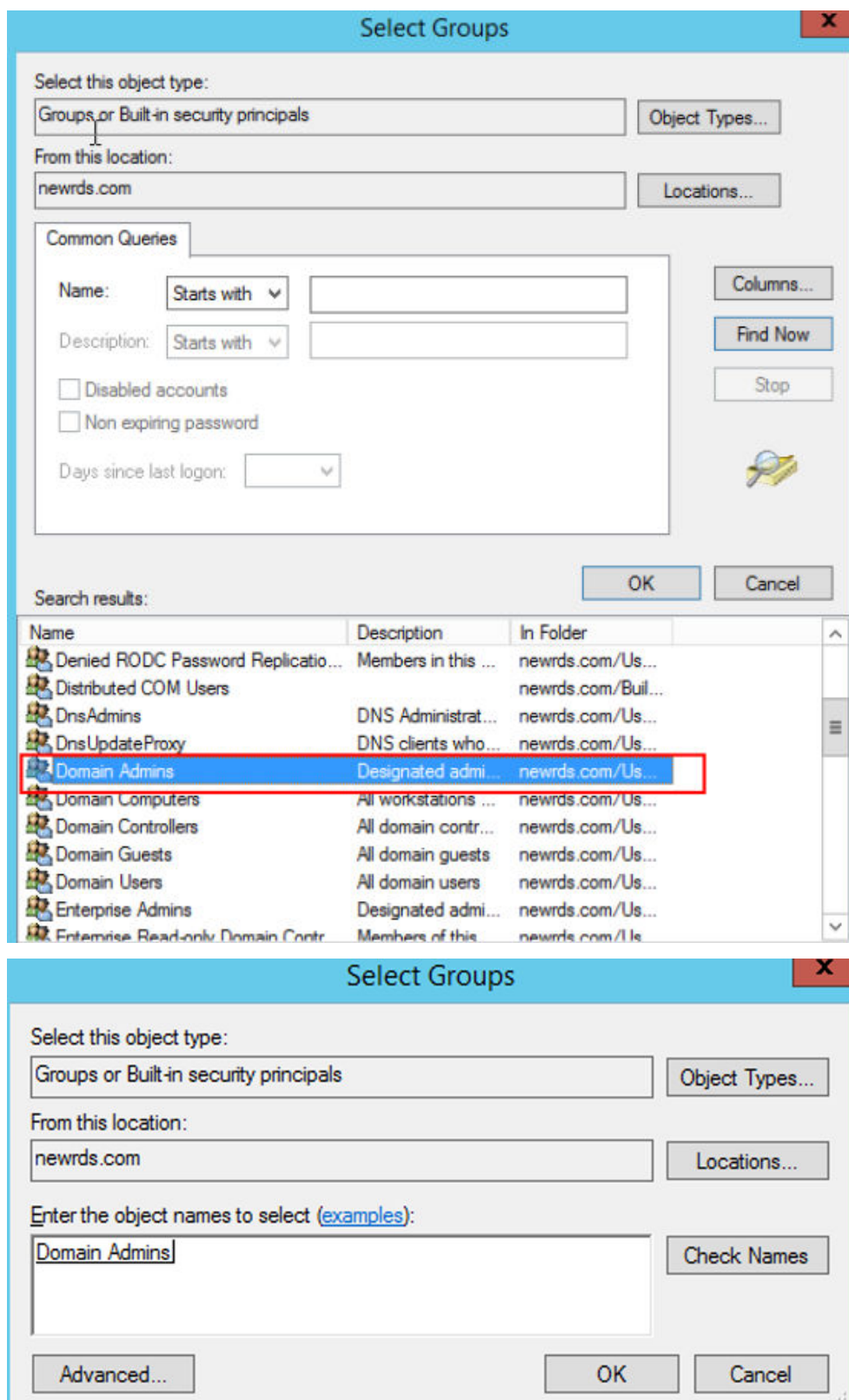
Step 2 Create and add a domain account.

1. Open **Active Directory Users and Computers**, right-click on the **Users** and choose **New > User**. Enter the username and click **Next**.
2. Enter the first name, last name, and a user logon name, such as luna@newrds.com.
3. Enter the password and then confirm it. Deselect all check boxes (do not change the password at the first login).
4. After the user is added, a figure similar to the following is displayed. You can add domain accounts to user groups for permission control.



5. Add the domain account for logging in to RDS to the AD Admin group.





The screenshot shows the 'Remote Desktop Services Profile' configuration window. The 'Member Of' tab is active, displaying a table of domain groups. The 'Domain Users' group is selected. Below the table are 'Add...' and 'Remove' buttons. At the bottom, the 'Primary group' is set to 'Domain Users', with a 'Set Primary Group' button and a note: 'There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.' The window has standard 'OK', 'Cancel', 'Apply', and 'Help' buttons at the bottom.

Member of:	
Name	Active Directory Domain Services Folder
Domain Admins	newrds.com/Users
Domain Users	newrds.com/Users

Buttons: Add..., Remove

Primary group: Domain Users

Set Primary Group

There is no need to change Primary group unless you have Macintosh clients or POSIX-compliant applications.

Buttons: OK, Cancel, Apply, Help

Step 3 Add an RDS DB instance to the domain.

On the instance creation page, click **Configure** to configure the AD domain and then complete the DB instance creation. Once the instance is created, the AD domain is ready for use.

Table 8-2 AD domain parameters

Parameter	Description
Directory Address	Enter the IP address of an ECS that supports the AD domain. For example: 192.168.x.x. NOTE Ensure that the ECS IP address can connect to the RDS DB instance.

Parameter	Description
Domain Name	A fully qualified domain name, such as DBStest.com, must: 1. Be the same as the ECS domain name. 2. Be no more than 48 characters long. 3. Include letters, digits, periods (.), and hyphens (-) only. 4. Include a domain extension, which contains at least 2 characters and cannot contain digits, for example, .com.
Directory Administrator	Enter the domain administrator username.
Directory Administrator Password	Password of the directory administrator. Keep this password secure. The system cannot retrieve it.

----End

8.1.10 What Can I Do About Slow Respond of Websites When They Use RDS?

To solve this problem:

- Check the performance of RDS DB instances on the RDS console.
- Compare the database connection status of local databases and RDS DB instances. This problem depends on web applications.

8.1.11 How Does a Cloud Database Perform a Primary/Standby Switchover?

RDS provides the high availability. Select the primary/standby mode.

Failover

Also called an unplanned handover. If the primary DB instance fails, the system will automatically switch to the standby DB instance within 5 minutes. No human intervention is required. The connection IP address remains unchanged. DB instances cannot be accessed during the failover. You need to configure automatic reconnections between applications and RDS DB instances to ensure near-continuous availability.

Switchover (Manual)

Also called a planned handover. When a DB instance is running properly, you can manually perform a primary/standby switchover if it is needed.

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner and select a region and a project.

- Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
- Step 4** On the **Instance Management** page, click the target DB instance.
- Step 5** In the **DB Information** area on the displayed **Basic Information** page, click **Switch** in the **DB Instance Type** field.

Alternatively, click  in the DB instance topology on the **Basic Information** page to manually trigger a primary/standby switchover.

NOTICE

A primary/standby switchover may cause service interruption for some seconds or minutes (depending on the replication delay). If the replication delay is too long, a small amount of data may be lost. To prevent traffic congestion, perform a switchover during off-peak hours.

- Step 6** In the **Switch Primary/Standby Instances** dialog box, click **Yes** to switch between the primary and standby DB instances.

If the replication status is **Available** and the replication delay is greater than 300s, the primary/standby switchover task cannot be delivered.

- Step 7** After a switchover is successful, you can view and manage the DB instance on the **Instance Management** page.
- During the switchover process, the DB instance status is **Switchover in progress**.
 - In the upper right corner of the DB instance list, click  to refresh the list. After the switchover is complete, the DB instance status will become **Available**.

----End

8.1.12 Can Multiple ECSs Connect to the Same RDS DB Instance?

Multiple ECSs can connect to the same RDS DB instance as long as the capability limits of a database are not exceeded.

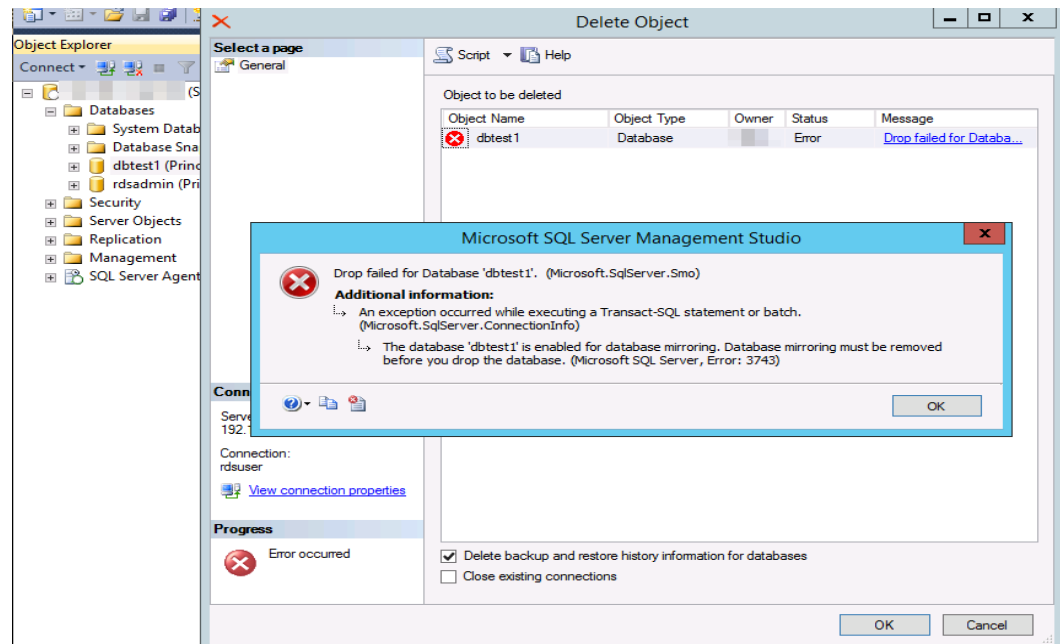
8.1.13 Why Is an Error Reported When I Attempt to Delete a Database from RDS SQL Server Primary/Standby DB Instances?

Symptom

An error shown in [Figure 8-1](#) is reported on SQL Server Management Studio when a database is being deleted from RDS SQL Server primary/standby DB instances.

The database 'xxxx' is enabled for database mirroring. Database mirroring must be removed before you drop the database. Error: 3743

Figure 8-1 Error information



Possible Causes

The error details indicate that the SQL Server DB instance type is primary/standby and database mirroring is enabled for the standby DB instance. As a result, the database cannot be deleted.

Solution

Before deleting the database, run the following commands to disable the mirroring:

Use master

go

ALTER DATABASE [Database_Name] SET PARTNER OFF;

GO

Once database mirroring is disabled, the database can be deleted.

8.2 Resource and Disk Management

8.2.1 Which Types of Logs and Files Occupy RDS Storage Space?

The following logs and files occupy RDS storage space.

Table 8-3 MySQL database file types

DB Engine	File Type
MySQL	Log files: database undo-log, redo-log, and binlog files
	Data files: database content files and index files
	Other files: ibdata, ib_logfile0, and temporary files

Table 8-4 PostgreSQL database file types

DB Engine	File Type
PostgreSQL	Log files: database error log and transaction log files
	Data files: database content, index, replication slot data, transaction status data, and database configuration files
	Other files: temporary files

Table 8-5 Microsoft SQL Server database file types

DB Engine	File Type
Microsoft SQL Server	Log files: database error log, transaction log, and trace files
	Data files: database content files

Solution

1. If the original storage space is insufficient as your services grow, scale up storage space of your DB instance.
2. If data occupies too much storage space, run **DROP**, **TRUNCATE**, or **DELETE +OPTIMIZE TABLE** to delete useless historical table data to release storage space. If no historical data can be deleted, scale up your storage space.
3. If temporary files generated by sorting queries occupy too much storage space, optimize your SQL query statements.
 - a. A large number of temporary files are generated if there are a large number of sorting queries executed by applications.
 - b. A large number of binlog files are generated and occupy space if there are large amounts of insert, delete, and update operations in a short period.
 - c. A large number of binlog files are generated if there are a large number of transactions and write operations.
4. Use to monitor the size, usage, and utilization of storage space of your DB instance and set alarm policies.

8.2.2 Which Items Occupy the Storage Space of My RDS DB Instances?

Both your common data (excluding backup data) and the data required for the operation of your DB instances (such as system database data, rollback logs, redo logs, and indexes) occupies storage space on your purchased RDS DB instances. The following RDS log files also occupy storage space:

- Binlog files generated by MySQL databases
- Logs files generated by PostgreSQL database servers
- Log files, including Microsoft SQL Server logs, default Microsoft SQL Server Trace logs, and Microsoft SQL Server Agent logs, generated by Microsoft SQL Server databases.

These files ensure the stability of RDS DB instances.

8.2.3 What Overhead Does the Storage Space Have After I Applied for an RDS DB Instance?

The storage space you applied for will include space for the file system overhead required for inode, reserved block, and database operations.

8.2.4 How Much Storage Space Is Required for DDL Operations?

Data Definition Language (DDL) operations may increase storage space usage sharply. To ensure that services are running properly, do not perform DDL operations during peak hours. If DDL operations are required, ensure that storage space is 10 GB greater than or equal to twice the size of the tablespace. For example, if your tablespace is 500 GB, ensure that storage space is greater than or equal to 1,010 GB ($500 \text{ GB} \times 2 + 10 \text{ GB}$).

8.2.5 How Many DB Instances Can Run on RDS?

There are no limitations on the number of DB instances running on RDS.

8.2.6 How Many Databases Can Run on an RDS DB Instance?

The maximum number of databases that can run on an RDS DB instance depends on the DB engine settings.

If there are enough CPU, memory, and storage resources, there are no limitations to the number of databases running on a DB instance. However, the number of tables in the databases affects the backup speed. If the number of tables exceeds 500,000, the backup will fail.

- MySQL allows you to create numerous databases and tables. For details, see the official MySQL documentation.
- PostgreSQL allows you to create numerous databases and database accounts.
- Microsoft SQL Server allows you to create a maximum of 100 databases and numerous database accounts.

8.3 Database Connection

8.3.1 Can an External Server Access the RDS Database?

DB Instance Bound with an EIP

For a DB instance that has an EIP bound, you can access it through the EIP.

DB Instance Not Bound with an EIP

- Enable a VPN in a VPC and use the VPN to connect to the RDS DB instance.
- Create an RDS and an ECS in the same VPC and access RDS through the ECS.

8.3.2 What Do I Do If the Number of RDS Database Connections Reaches the Upper Limit?

The number of database connections indicates the number of applications that can be simultaneously connected to a database, and is irrelevant to the maximum number of users allowed by your applications or websites.

If there is an excessive number of database connections, applications may fail to be connected, and the full and incremental backups may fail, affecting services.

Fault Locating

1. Check whether applications are connected, optimize the connections, and release unnecessary connections.
2. Check the specifications and scale them up if needed.
3. Check whether any metrics are abnormal and whether any alarms are generated on the Cloud Eye console. Cloud Eye monitors database metrics, such as the CPU usage, memory usage, storage space usage, and database connections, and allows you to set alarm policies to identify risks in advance if any alarms are generated. For details, see the *Cloud Eye User Guide*.

8.3.3 What Is the Maximum Number of Connections to an RDS DB Instance?

RDS does not have constraints on the number of connections. This number is determined by the default value and value range of the DB engine. For example, you can set **max_connections** and **max_user_connections** in a parameter template to configure the maximum number of connections for an RDS for MySQL DB instance.

Changing the Maximum Number of Connections

You can change the maximum number of connections using commands.

1. Query the maximum number of connections.
show global variables like 'max_connections';
2. Change the value of **max_connections** under **mysqld** in the **my.cnf** file.
[mysqld]
max_connections = 1000

Setting the Maximum Number of RDS for MySQL Connections to an Appropriate Value

- In addition to the value of **max_connections**, the maximum number of concurrent client connections allowed by MySQL is also limited by the maximum number of files that can be opened by a single process in the operating system. For example, if the maximum number of files that can be opened by each process is set to **100** in the operating system, the **max_connections** parameter does not take effect even if it is set to **200**.

Check the maximum number of files that can be opened by a single process in the operating system. The default value is **1024**.

ulimit -n

The parameter **open_files_limit** indicates the maximum number of files that can be opened by a single process, which is read from the operating system during MySQL startup.

Check the value of **open_files_limit**.

show variables like 'open_files_limit';

- Suggestions
The maximum number of MySQL connections can be customized provided that it is allowed by your instance specifications. The maximum number of connections is closely related to the instance memory. For details, see [About max_connections](#).

In actual scenarios, set the maximum number of connections to an appropriate value because more connections lead to large resource consumption.

About max_connections

max_connections: maximum number of clients that can be connected at the same time. If this parameter is set to **default**, it is related to the memory (unit: GB) of the DB instance. The calculation formula is as follows:

Estimated value of max_connections = Available node memory/Estimated memory occupied by a single connection

NOTE

- Available node memory = Total memory – Memory occupied by the buffer pool – 1 GB (mysqld process/OS/monitoring program)
- Estimated memory usage of a single connection (single_thread_memory) = thread_stack (256 KB) + binlog_cache_size (32 KB) + join_buffer_size (256 KB) + sort_buffer_size (256 KB) + read_buffer_size (128 KB) + read_rnd_buffer_size (256 KB) ≈ 1 MB

The following table lists the default values of **max_connections** for different memory specifications.

Table 8-6 Max_connections for different memory specifications

Memory (GB)	Connections
512	100,000
384	80,000
256	60,000
128	30,000
64	18,000
32	10,000
16	5,000
8	2,500
4	1,500
2	800

8.3.4 How Can I Create and Connect to an ECS?

- For details about how to create an ECS, see the *Elastic Cloud Server User Guide*.
 - If you connect to an RDS DB instance through a private network, ensure that the ECS and DB instance are in the same VPC. If you connect to an RDS DB instance through a public network, the ECS and DB instance can be in different VPCs.
 - Configure a security group to allow the ECS to access the RDS DB instance through the IP address.
- For details on how to connect to the ECS, see section "Logging In to an ECS" in the *Elastic Cloud Server User Guide*.

8.3.5 What Should I Do If an ECS Cannot Connect to an RDS DB Instance Through a Private Network?

Perform the following steps to identify the problem:

- Step 1** Check whether the ECS and RDS DB instances are located in the same VPC.
- If they are in the same VPC, go to [Step 2](#).
 - If they are in different VPCs, create an ECS in the VPC in which the RDS DB instance is located.
- Step 2** Check whether a security group has been added to the ECS.
- If a security group has been added, check whether the rules are configured correctly.
- For MySQL DB instances, see the security group description in [Step 1: Create a DB Instance](#). Then, go to [Step 3](#).

For PostgreSQL DB instances, see the security group description in [Step 1: Create a DB Instance](#). Then, go to [Step 3](#).

For Microsoft SQL Server DB instances, see the security group description in [Step 1: Create a DB Instance](#). Then, go to [Step 3](#).

- If no security group has been added, go to the VPC console from the ECS details page and click **Security Groups** to add a security group.

Step 3 On the ECS, check whether the RDS DB instance port can be connected.

The default port of RDS for MySQL is **3306**.

The default port of RDS for PostgreSQL is **5432**.

The default RDS for SQL Server port number is **1433**.

```
telnet <IP address> {port number}
```

- If the ECS can connect to the RDS DB instance port, the network between the ECS and the RDS DB instance is normal and no further action is required.
- If the ECS still cannot connect to the port, contact technical support.

----End

8.3.6 What Should I Do If a Database Client Problem Causes a Connection Failure?

Troubleshoot RDS connection failures caused by a client problem by checking the following items:

1. ECS Security Policy

In Windows, check whether the RDS instance port is enabled in the Windows security policy. In Linux, run **iptables** to check whether the RDS DB instance port is enabled in firewall settings.

2. Application Configuration

Check whether the connection address, port parameter configuration, and JDBC connection parameter configuration are correct.

3. Username or Password

Check whether the username or password is correct if an error similar to the following occurs during RDS DB connection:

- [Warning] Access denied for user 'username'@'yourIp' (using password: NO)
- [Warning] Access denied for user 'username'@'yourIp' (using password: YES)
- Login failed for user 'username'

 NOTE

If the problem persists, contact post-sales technical support.

8.3.7 What Should I Do If an RDS Database Problem Causes a Connection Failure?

Check whether any of the following problems occurred on the RDS DB instance.

1. The RDS DB instance is not properly connected.

Solution: Check the connection. If you connect to the RDS DB instance through a private network, the ECS and DB instance must be in the same VPC and the DB instance can be accessed only through an ECS in the same VPC. If you connect to the RDS DB instance through a public network, the ECS and DB instance can be in different VPCs.

2. The maximum number of connections has been reached.

Solution: Use RDS resource monitoring to check if the CPU usage and the number of current connections are abnormal. If either of them has reached the maximum, reboot, disconnect, or scale up the class of the RDS DB instance.

3. The DB instance is abnormal. For example, the RDS DB instance fails to be rebooted, the system is faulty, or the instance or table is locked.

Solution: Reboot the RDS DB instance to see if the problem is resolved. If the problem persists, contact post-sales technical support.

8.3.8 How Do My Applications Access an RDS DB Instance in a VPC?

Ensure that the ECS in which your applications are located is in the same VPC as the RDS DB instance. If the ECS and the RDS DB instance are in different VPCs, modify the VPC route table and network access control list (ACL) to ensure that the ECS can access the RDS DB instance.

8.3.9 Do Applications Need to Support Reconnecting to the RDS DB Instance Automatically?

It is recommended that your applications support automatic reconnections to the database. After a database reboot, your applications will automatically reconnect to the database to increase service availability and continuity.

To reduce resource consumption and improve performance, configure your applications to connect to the database using a persistent connection.

8.3.10 How Can I Connect to a PostgreSQL Database Through JDBC?

Although the SSL certificate is optional if you choose to connect to a database through Java database connectivity (JDBC), download an SSL certificate to encrypt the connections for security.

Prerequisites

Familiarize yourself with:

- Computer basics.
- Java programming language.
- JDBC knowledge.

Obtaining and Using JDBC

- JDBC driver download address: <https://jdbc.postgresql.org/download/>
- JDBC Interface: <https://jdbc.postgresql.org/documentation/>

Connection with the SSL Certificate

NOTE

Download the SSL certificate and verify the certificate before connecting to databases.

In the **DB Information** area on the **Basic Information** page, click  in the **SSL** field to download the root certificate or certificate bundle.

Step 1 Connect to the RDS PostgreSQL DB instance through JDBC.

```
jdbc:postgresql://<instance_ip>:<instance_port>/<database_name>?sslmode=verify-  
full&sslrootcert=<ca.pem>
```

Table 8-7 Parameter description

Parameter	Description
<instance_ip>	If you attempt to access the RDS DB instance through an ECS, set <i>instance_ip</i> to the floating IP address displayed on the Basic Information page of the DB instance to which you intend to connect.
	If you attempt to access the RDS DB instance through an EIP, set <i>instance_ip</i> to the EIP that has been bound to the DB instance.
<instance_port>	Enter the database port displayed on the Basic Information page. Default value: 5432
<database_name>	Enter the name of the database to which you intend to connect. Default value: postgres
sslmode	Enter the SSL connection mode. Default value: verify-full
sslrootcert	Enter the directory of the CA certificate for the SSL connection. The certificate should be stored in the directory where the command is executed.

Example script in Java:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;
        // set sslmode here.
        // with ssl certificate and path.
        String url = "jdbc:postgresql://192.168.0.225:5432/my_db_test?sslmode=verify-  
full&sslrootcert=/home/Ruby/ca.pem";
```

```
try {
    Class.forName("org.postgresql.Driver");
    conn = DriverManager.getConnection(url, "root", "password");
    System.out.println("Database connected");

    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM mytable WHERE columnfoo = 500");
    while (rs.next()) {
        System.out.println(rs.getString(1));
    }

    rs.close();
    stmt.close();
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
    System.out.println("Test failed");
} finally {
    // release resource ....
}
}
```

----End

Connection Without the SSL Certificate

NOTE

You do not need to download the SSL certificate because the certificate verification on the server is not required.

Step 1 Connect to the RDS PostgreSQL DB instance through JDBC.

```
jdbc:postgresql://<instance_ip>:<instance_port>/<database_name>?sslmode=disable
```

Table 8-8 Parameter description

Parameter	Description
<instance_ip>	If you attempt to access the RDS DB instance through an ECS, set <i>instance_ip</i> to the floating IP address displayed on the Basic Information page of the DB instance to which you intend to connect.
	If you attempt to access the RDS DB instance through an EIP, set <i>instance_ip</i> to the EIP that has been bound to the DB instance.
<instance_port>	Enter the database port displayed on the Basic Information page. Default value: 5432
<database_name>	Enter the name of the database to which you intend to connect. Default value: postgres
sslmode	Enter the SSL connection mode. disable means data is not encrypted.

Example script in Java:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;
        // set sslmode here.
        // no ssl certificate, so do not specify path.
        String url = "jdbc:postgresql://192.168.0.225:5432/my_db_test?sslmode=disable";
        try {
            Class.forName("org.postgresql.Driver");
            conn = DriverManager.getConnection(url, "root", "password");
            System.out.println("Database connected");

            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM mytable WHERE columnfoo = 500");
            while (rs.next()) {
                System.out.println(rs.getString(1));
            }
            rs.close();
            stmt.close();
            conn.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Test failed");
        } finally {
            // release resource ....
        }
    }
}
```

----End

8.3.11 What Should I Do If an RDS for SQL Server DB Instance Failed to Be Connected?

Fault Location

- Check whether the ECS can connect to the RDS DB instance.
If the ECS cannot connect to the RDS DB instance, check whether the ECS and RDS DB instance are located in the same VPC and security group.
In private network connection mode, the ECS and RDS DB instance must be in the same VPC and the DB instance can be accessed only through an ECS in the same VPC. In public network connection mode, the ECS and DB instance can be in different VPCs.
- Check whether the IP address and port are correct.
Use a colon to separate the IP address and port.
- Check whether the RDS service is running properly.
- Check whether the username and password are correct. You can reset the password.
- Reboot the RDS DB instance and check whether it can be connected through an ECS.

Solution

- Step 1** Log in to the management console.
 - Step 2** Click  in the upper left corner and select a region and a project.
 - Step 3** Click **Service List**. Under **Database**, click **Relational Database Service**. The RDS console is displayed.
 - Step 4** On the **Instance Management** page, click the target DB instance. On the **Basic Information** and **Backups & Restorations** pages, check connection and backup information.
 - Step 5** On the **Basic Information** page, check the administrator.
 - Step 6** Download an SQL Server Management Studio installation package and install it on an ECS.
 - Step 7** Connect to the RDS DB instance through an ECS.
- End

8.3.12 Can I Access an RDS DB Instance Over an Intranet Connection Across Regions?

By default, RDS DB instances cannot be accessed over an intranet across regions. Cloud services in different regions cannot communicate with each other over an intranet. You can use Cloud Connect (CC) or Virtual Private Network (VPN) to connect to RDS DB instances across regions.

8.3.13 Is an SSL Connection to a DB Instance Interrupted After a Primary/Standby Switchover or Failover?

For DB instances connected using SSL, a primary/standby switchover or failover does not interrupt the connection because the SSL certificate is still valid for both the primary and standby DB instances.

8.3.14 Does MySQL Support SSL Connections?

MySQL supports SSL connections. Different from other vendors, RDS for MySQL enables the SSL connection on the database server by default. When you use a client to connect to MySQL DB instances, you can determine whether to enable SSL as required.

For details about SSL, see section [Connecting to a DB Instance Through a Public Network](#).

8.3.15 Why Did the New Password Not Take Effect After I Reset the Administrator Password?

Possible Causes

You may have restored from a backup before you reset the administrator password.

Locating Method

Check whether the DB instance was restored after you reset the administrator password.

Solution

Log in to the RDS console and reset the administrator password again. For details, see section [Resetting the Administrator Password](#).

8.4 Database Migration

8.4.1 Why Do I Need to Use the mysqldump or pg_dump Tools for Migration?

The mysqldump or pg_dump tool is easy to use for data migration. However, when you use this tool, the server is stopped for a long period of time during data migration. Only use these tools when there is not much data to migrate or if stopping the server for a long period of time is not an issue.

RDS is compatible with source database services. The procedure for migrating data from your database to RDS is similar to the procedure for migrating data from one database server to another.

8.4.2 What Types of DB Engines Does RDS Support for Importing Data?

- Exporting or importing data between DB engines of the same type is called homogeneous database export or import.
- Exporting or importing data between DB engines of different types is called heterogeneous database export or import. For example, import data from Oracle to DB engines supported by RDS.

Generally, data cannot be exported or imported between heterogeneous databases due to the different data formats involved. However, if the data formats are compatible, table data can, in theory, be migrated between them.

Third-party software is usually required for data replication to export and import between heterogeneous databases. For example, you can use a third-party tool to export table records from Oracle into a .txt file. Then, you can use LOAD statements to import the exported table records to a DB engine supported by RDS.

8.5 Database Permission

8.5.1 Why Does the Root User Not Have the Super Permissions?

Most relational database cloud service platforms do not provide super permissions for the **root** user. The super permissions allow users to execute many

management commands, such as reset master, set global, kill, and reset slave. These operations may cause primary/standby replication errors. This is a major difference between databases and on-premises MySQL databases. To ensure stable running of DB instances, RDS does not provide the super permission for the **root** user.

If you need to perform actions that normally require super permissions, RDS provides alternative methods.

For example:

1. You can modify parameter values only on the RDS console. You cannot run the following command on an RDS DB database to modify parameter values.
set global parameter name=*Parameter value*;
If the script contains the **set global** command, delete the **set global** command and modify parameter values through the RDS console.
2. An error is reported after you run the following command because the **root** user does not have super permissions. You can delete **definer='root'** from the command to solve the problem.
create definer='root'@'%' trigger(procedure)...
You can import data using mysqldump. For operation details, see [Migrating Data to RDS for MySQL Using mysqldump](#).
3. You can create PostgreSQL plugins by referring to [Creating or Deleting a Plugin](#).

8.6 Database Storage

8.6.1 What Storage Engines Does the RDS for MySQL Support?

The database storage engine is a core service for **storing, processing, and protecting data**. It can be used to control access permissions and rapidly process transactions to meet enterprise requirements.

For MySQL databases, only InnoDB supports backup and restoration functions and is therefore recommended.

For versions later than MySQL 5.6.40 and 5.7.22, some storage engines are no longer supported.

RDS for MySQL does not currently support MyISAM because:

- MyISAM engine tables do not support transactions and support only table-level locks. As a result, read and write operations conflict with each other.
- MyISAM has a defect in protecting data integrity, which may cause database data damage or even data loss.
- If data is damaged, MyISAM does not support data restoration provided by RDS for MySQL and requires manual restoration.
- Data can be transparently migrated from MyISAM to InnoDB, which does not require code modification for tables.

RDS for MySQL does not currently support FEDERATED, because:

- If the primary/standby DB instances support FEDERATED, the same DML operations are repeatedly executed on remote databases, resulting in data disorder.
- During the PITR restoration, data on remote databases is not restored to the status when the full backup is performed after the full restoration is complete. Applying data during the incremental restoration will disorder FEDERATED table data.

RDS for MySQL does not currently support MEMORY for the following reasons:

- If a memory table becomes empty after a restart, the database generates a DELETE event to the binlog when the table is opened. If primary/standby DB instances use memory tables and the standby database (or read-only database) is restarted, a GTID is generated, which is inconsistent with that of the primary database. As a result, the standby database is rebuilt.
- Using memory tables may cause out-of-memory (OOM) errors and even service terminations.

8.6.2 What Is the RDS DB Instance Storage Configuration?

RDS uses Elastic Volume Service (EVS) disks for storage. For details about EVS disks, see *Elastic Volume Service Product Introduction*.

The RDS backup data is stored in OBS and does not occupy the database storage space. For details on the RDS DB instance storage configuration, see the *Object Storage Service User Guide*.

8.6.3 Can I Change the Storage Type of an RDS DB Instance from Common I/O to Ultra-high I/O?

No. After an RDS DB instance is created, the storage type cannot be changed.

Table 8-9 Items that cannot be changed

Item	Change Direction
Storage type	• From common I/O to ultra-high I/O • From ultra-high I/O to common I/O • From high I/O to common I/O The preceding descriptions are examples only. The storage type cannot be changed.

8.6.4 What Should I Do If My Data Exceeds the Available Storage of an RDS DB Instance?

Scenario

There is not enough storage available for an RDS DB instance and the DB instance has entered read-only status, or applications can no longer read from or write to the databases, interrupting services.

Causes

Possible causes include:

- Too much data being stored.
- Too many MySQL binlog files being generated due to a large number of transactions and write operations.
- A large number of temporary files being generated due to a large number of sorting queries executed by applications.

Solution

1. If your DB instance becomes read-only because there is too much data, contact technical support to cancel the read-only status and run statements to delete useless historical table data to free up space. To delete data of an entire table, run **DROP** or **TRUNCATE**. To delete part of table data, run **DELETE** and **OPTIMIZE TABLE**. If the instance still supports read and write operations, directly delete useless historical table data. If no historical data can be deleted, scale up your storage space.
2. If MySQL binlog files are taking up too much space, contact technical support to delete local binlog files to free up storage space.
3. If temporary files generated by sorting queries are taking up too much space, optimize your SQL query statements.
4. If none of these solutions work, use database and table sharding.

8.7 Client Installation

8.7.1 How Can I Install the MySQL Client?

MySQL provides client installation packages for different OSs on its official website. MySQL 5.6 is used as an example. Download the [MySQL 5.6 client installation package](#) or [other versions of the packages](#). The following procedure illustrates how to obtain the required installation package and install the MySQL client into a Red Hat Linux system.

Procedure

Step 1 Obtain the installation package.

Find the [link](#) to the required version on the download page. MySQL-client-5.6.31-1.el6.x86_64.rpm is used as an example in the following figure.

Figure 8-2 Download**MySQL Product Archives**

MySQL Community Server (Archived Versions)

Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Community Server, please visit [MySQL Downloads](#).

Product Version:
Operating System:
OS Version:

RPM Bundle (MySQL-5.6.31-1.el6.x86_64.rpm-bundle.tar)	May 19, 2016	215.4M	Download
RPM Package, MySQL Server (MySQL-server-5.6.31-1.el6.x86_64.rpm)	May 18, 2016	54.3M	Download
RPM Package, Client Utilities (MySQL-client-5.6.31-1.el6.x86_64.rpm)	May 18, 2016	17.9M	Download
RPM Package, Development Libraries (MySQL-devel-5.6.31-1.el6.x86_64.rpm)	May 18, 2016	3.2M	Download

Step 2 Upload the installation package to the ECS.

1. When you create an ECS, select an OS, such as Red Hat 6.6, and bind an EIP to it.
2. Use a remote connection tool to connect to the ECS through the bound EIP and upload the installation package to the ECS.

Step 3 Run the following command to install the MySQL client:

```
sudo rpm -ivh MySQL-client-5.6.31-1.el6.x86_64.rpm
```

NOTE

- If there are any conflicts during the installation, add the **replacefiles** parameter to the command and try to install the client again. Example:

```
rpm -ivh --replacefiles MySQL-client-5.6.31-1.el6.x86_64.rpm
```
- If a message is displayed prompting you to install a dependency package, you can add the **nodeps** parameter to the command and install the client again. Example:

```
rpm -ivh --nodeps MySQL-client-5.6.31-1.el6.x86_64.rpm
```

----End

8.7.2 How Can I Install the PostgreSQL Client?

PostgreSQL provides [client installation methods](#) for different OSs on its official website.

The following describes how to install a PostgreSQL 12 client in CentOS.

Procedure

Step 1 Log in to an ECS.

1. When you create an ECS, select an OS, such as CentOS 7, and bind an EIP to it.
2. Use a remote connection tool to connect to the ECS through the EIP.

Step 2 Open the [client installation page](#).**Step 3** Select a DB engine version, OS, and OS architecture, and run the following commands on the ECS to install a PostgreSQL client.

```
sudo yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm  
sudo yum install -y postgresql12-server
```

Figure 8-3 Installing a client

To use the PostgreSQL Yum Repository, follow these steps:

- Select version:

12
- Select platform:

Red Hat Enterprise, CentOS, Scientific or Oracle version 7
- Select architecture:

x86_64
- Copy, paste and run the relevant parts of the setup script:


```
# Install the repository RPM:
sudo yum install -y https://download.postgresql.org/pub/repos/yum/reposms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm

# Install PostgreSQL:
sudo yum install -y postgresql12-server

# Optionally initialize the database and enable automatic start:
sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
sudo systemctl enable postgresql-12
sudo systemctl start postgresql-12
```

Copy Script

- Select a DB engine version that is consistent with that of your RDS for PostgreSQL instance.
- Select an OS that is consistent with that of the ECS.
- Select an OS architecture that is consistent with that of the .

Figure 8-4 Installing the RPM package

```
root@ecs-d685 ~]# sudo yum install -y https://download.postgresql.org/pub/repos/yum/reposms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
Loaded plugins: fastestmirror
pgdg-redhat-repo-latest.noarch.rpm                                | 8.6 kB  00:00:00
Examining /var/tmp/yum-root-ZonITG/pgdg-redhat-repo-latest.noarch.rpm: pgdg-redhat-repo-42.0-28.noarch
Marking /var/tmp/yum-root-ZonITG/pgdg-redhat-repo-latest.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package pgdg-redhat-repo.noarch 0:42.0-28 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                arch             Version          Repository        Size
=====
Installing:
pgdg-redhat-repo                        noarch           42.0-28          /pgdg-redhat-repo-latest.noarch 13 k
Transaction Summary
=====
Install 1 Package

Total size: 13 k
Installed size: 13 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : pgdg-redhat-repo-42.0-28.noarch                                1/1
Verifying  : pgdg-redhat-repo-42.0-28.noarch                                1/1

Installed:
pgdg-redhat-repo.noarch 0:42.0-28

Complete!
```

Figure 8-5 Client installed

```
Total
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
Importing GPG key 0x442DF0F8:
Userid : "PostgreSQL RPM Building Project <pgsql-pkg-yum@postgresql.org>"
Fingerprint: 60c9 e2b9 1a37 d136 fe74 d176 1f16 d2e1 442d f0f8
Package : pgdg-redhat-repo-12.0-28.noarch (@pgdg-redhat-repo-latest.noarch)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-PGDG
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : libicu-58.2-4.el7_7.x86_64
Installing : postgresql12-libs-12.13-1PGDG.rhel7.x86_64
Installing : postgresql12-12.13-1PGDG.rhel7.x86_64
Installing : postgresql12-server-12.13-1PGDG.rhel7.x86_64
Verifying : postgresql12-libs-12.13-1PGDG.rhel7.x86_64
Verifying : postgresql12-12.13-1PGDG.rhel7.x86_64
Verifying : postgresql12-server-12.13-1PGDG.rhel7.x86_64
Verifying : libicu-58.2-4.el7_7.x86_64

Installed:
 postgresql12-server.x86_64 0:12.13-1PGDG.rhel7

Dependency Installed:
 libicu.x86_64 0:58.2-4.el7_7 postgresql12.x86_64 0:12.13-1PGDG.rhel7 postgresql12-libs.x86_64 0:12.13-1PGDG.rhel7

Complete!
```

Step 4 Connect to the RDS for PostgreSQL instance.

Figure 8-6 Connection successful

```
[root@ecs-d605 ~]# psql -h [redacted] -d postgres -U root
Password for user root:
psql (12.13, server 12.11)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

postgres=>
```

----End

8.7.3 How Can I Install SQL Server Management Studio?

The Microsoft SQL Server official website provides the SQL Server Management Studio installation package. SQL Server Management Studio applications can run in the Windows OS only.

Procedure

Step 1 Obtain the SQL Server Management Studio installation package.

Visit the [Microsoft website](#) and download the installation package, for example, SQL Server Management Studio 18.0.

Step 2 Upload the installation package to the ECS.

Step 3 Double-click the installation package and complete the installation as instructed.

----End

8.8 Backup and Restoration

8.8.1 How Long Does RDS Store Backup Data For?

Automated backup data is kept based on the backup retention period you specified. There is no limit for the manual backup retention period. You can delete manual backups as needed.

The backup data is stored in OBS and does not occupy the database storage space.

8.8.2 Can My Database Be Used in the Backup Window?

A backup window is a user-specified time during which RDS DB instances are backed up. With these periodic data backups, RDS allows you to restore DB instances to the backups during a retention period. This backup process does not affect services. However, you cannot reboot these DB instances on the RDS console.

8.8.3 How Can I Back Up an RDS Database to an ECS?

You can back up data to an ECS the same way you export SQL statements. The ECS service does not have restrictions on the types of data to be backed up as long as the data complies with local laws and regulations. You can store RDS backup data on an ECS, but using an ECS is not recommended. You are advised to store RDS backup data to OBS for higher data reliability and service assurance.

8.8.4 Why Has My Automated Backup Failed?

Automated backups may fail for the following reasons:

1. The network environment is unstable due to issues such as network delay or interruption. RDS will detect these problems and trigger another automated backup half an hour later. You can also perform a manual backup before then.
2. Multi-task executions are complicated, resulting in problems such as task waiting or interruptions. RDS will detect these problems and trigger another automated backup half an hour later. You can also perform a manual backup.
3. A DB instance status is unavailable, possibly because the DB instance is faulty or being modified. RDS will trigger an automated backup when the DB instance status becomes available. You can also perform a manual backup before then.
4. The backup speed depends on how many tables there are in the databases. If the number of tables exceeds 500,000, the backup will fail.
5. A parameter change was incorrect. For example, a DB instance may become faulty after a parameter template containing incorrect parameters was applied to it. You can check whether original and current values are correct, check whether any related parameters also need to be changed, reset the parameter template, or reboot the DB instance.
6. An error occurred during data import.
For example, system table records get lost due to inappropriate data import.
7. If the problem persists, contact technical support.

8.8.5 What Happens to Database Backups After an RDS DB Instance Is Deleted?

When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.

8.8.6 Will My Backups Be Deleted If I Delete My Cloud Account?

If your cloud account is deleted, both your automated and manual backups are deleted.

8.8.7 Why Is a Table or Data Missing from My Database?

RDS does not delete or perform any operations on any user data. If this problem occurs, check if there have been any misoperations and restore the data from backup files, if necessary.

Check for misoperations: If , you can view data execution records in audit logs.

Restore data using backup files:

- Use the RDS restoration function.
- Import the backup data to RDS through an ECS.

8.9 Database Monitoring

8.9.1 Which DB Instance Monitoring Metrics Do I Need to Pay Attention To?

You need to pay attention to CPU, memory, and storage space usage.

You can configure the system to report alarms based on service requirements and take measures to handle any reported alarms.

Configuration examples:

- Configure RDS to report alarms to Cloud Eye if its CPU utilization reaches or exceeds a specific value (for example, 90%) multiple times (for example, 3 times) within a set period (for example, 5 minutes).
- Configure RDS to report alarms to Cloud Eye if its memory utilization reaches or exceeds a specific value (for example, 90%) multiple times (for example, 4 times) within a set period (for example, 5 minutes).
- Configure RDS to report alarms to Cloud Eye if its storage utilization reaches or exceeds a specific value (for example, 85%) multiple times (for example, 5 times) within a set period (for example, 5 minutes).

NOTE

For details on Cloud Eye alarm configuration, see "Creating an Alarm Rule" in the *Cloud Eye User Guide*.

Measures:

- If a CPU or memory alarm is reported, you can scale up the vCPUs or memory by changing the DB instance class.
- If a storage space usage alarm is reported, you can:

- Check the storage space consumption to see if any space can be freed up by deleting data from DB instances or by dumping the data to another system.
- Scale up the storage space.

8.10 Capacity Expansion and Specification Change

8.10.1 Are My RDS DB Instances Available When Scaling?

Currently, you can scale up storage space and change the CPU or memory of a DB instance.

- When storage space is being scaled up, RDS DB instances are still available and services are not affected. However, you cannot delete or reboot DB instances that are being scaled.
- When the CPU or memory of DB instances is being changed, the network is intermittently disconnected once or twice within a few seconds. (For Microsoft SQL Server 2017 Enterprise Edition, you need to stop services first and then change the CPU or memory of DB instances.) For primary/standby DB instances, a failover may occur and services may be briefly interrupted. Changing the CPU or memory takes 5 to 15 minutes.

After you change the CPU or memory, the DB instances will reboot and services will be interrupted. Therefore, you are advised to change instance classes during off-peak hours.

8.10.2 Why Does the DB Instance Become Faulty After the Original Database Port Is Changed?

Symptom

- The DB instance is in **Faulty** state after the original database port is changed.
- The DB instance cannot be connected using the new database port.

Possible Causes

The submitted database port is occupied.

Procedure

- If the database port is changed successfully, the previous change failed because the submitted database port was occupied.
- If the original database port still fails to be changed, contact technical support.

8.11 Database Parameter Modification

8.11.1 What Inappropriate Parameter Settings Cause Unavailability of the PostgreSQL Database?

In the following cases, inappropriate parameter settings cause the database to be unavailable:

- Parameter value ranges are related to DB instance specifications.
The maximum values of **shared_buffers** and **max_connections** are related to the DB instance physical memory. If you set these parameters inappropriately, the database will be unavailable.
- Parameter association is incorrect.
 - If **log_parser_stats**, **log_planner_stats**, or **log_executor_stats** is enabled, you must disable **log_statement_stats**. Otherwise, the database is unavailable.
 - **max_connections**, **autovacuum_max_workers**, and **max_worker_processes** must meet the following requirements. Otherwise, the database is unavailable.
$$\text{max\_connections value} + \text{autovacuum\_max\_workers value} + \text{max\_worker\_processes value} + 1 < 8388607$$

NOTE

For additional details, visit the [PostgreSQL official website](#).

Solution:

1. Log in to the RDS console and query the logs to locate the incorrectly configured parameters.
2. On the **Configuration** page, change parameters to default values and reboot the database.
3. Configure the incorrect parameter values and restore other parameters to their original default values.

8.11.2 Where Should I Store the NDF Files for Microsoft SQL Server?

When you add NDF files of the custom database and the tempdb database, do not place them in the C drive. If you place them in the C drive, the system disk space will be used up and services may be interrupted. You need to store the NDF auxiliary file of the custom database in **D:\RDSDBDATA\DATA** and the NDF auxiliary file of the tempdb database in **D:\RDSDBDATA\Temp**.

8.11.3 How Do I Configure a Password Expiration Policy for RDS for MySQL DB Instances?

In MySQL 5.7 and 8.0, you can set the global variable **default_password_lifetime** to control the default validity period of a user password.

The value of **default_password_lifetime** indicates how many days until a password expires. The default value is **0**, indicating that the created user password will never expire.

```
mysql> show variables like 'default_password_lifetime';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| default_password_lifetime | 0 |
+-----+-----+
1 row in set (0.00 sec)
```

Changing the Global Automatic Password Expiration Policy

- Change the value of the **default_password_lifetime** parameter on the RDS console.
- Run the following command to change the value of **default_password_lifetime**:
mysql> set global default_password_lifetime=0;

Checking the Password Expiration Date of All Users

Run the following command:

```
mysql> select
user,host,password_expired,password_last_changed,password_lifetime from
user;
```

```
mysql> select user,host,password_expired,password_last_changed,password_lifetime from user;
+-----+-----+-----+-----+-----+
| user | host | password_expired | password_last_changed | password_lifetime |
+-----+-----+-----+-----+-----+
| mysql.session | localhost | N | 2020-01-17 15:02:23 | NULL |
| mysql.sys | localhost | N | 2020-01-17 15:02:23 | NULL |
| rdsAdmin | localhost | N | 2020-01-17 15:02:30 | 0 |
| root | % | N | 2020-03-05 14:23:54 | NULL |
| rdsRepl | 192.168.% | N | 2020-01-17 15:02:45 | 0 |
| rdsMetric | 192.168.% | N | 2020-01-17 15:02:30 | 0 |
| rdsBackup | localhost | N | 2020-01-17 15:02:30 | 0 |
| u_test01 | % | N | 2020-03-05 14:28:10 | 30 |
| u_test02 | % | N | 2020-03-05 14:28:38 | NULL |
| jeffrey | localhost | N | 2020-03-05 15:23:17 | NULL |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Checking the Password Expiration Policy of a Specified User

Run the following command:

```
mysql> show create user jeffrey@'localhost';
```

```
mysql> show create user jeffrey@'localhost';
+-----+-----+
| CREATE USER for jeffrey@localhost |
+-----+-----+
| CREATE USER 'jeffrey@localhost' IDENTIFIED WITH 'mysql_native_password' AS '*1369F151658FC90255853119A9CBB0554DB807F' REQUIRE NONE PASSWORD EXPIRE DEFAULT ACCOUNT UNLOCK |
+-----+-----+
1 row in set (0.00 sec)
```

EXPIRE DEFAULT indicates that the password follows the global expiration policy.

Configuring the Password Expiration Policy for a Specified User

- Configuring the password expiration policy during user creation

```
create user 'script'@'localhost' identified by '*****' password expire interval 90 day;
```

- Configuring the password expiration policy after user creation
ALTER USER 'script'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
- Setting the password to be permanently valid
mysql> CREATE USER 'mike'@'%' PASSWORD EXPIRE NEVER;
mysql> ALTER USER 'mike'@'%' PASSWORD EXPIRE NEVER;
- Setting the password to follow the global expiration policy
mysql> CREATE USER 'mike'@'%' PASSWORD EXPIRE DEFAULT;
mysql> ALTER USER 'mike'@'%' PASSWORD EXPIRE DEFAULT;

8.12 Log Management

8.12.1 How Long Is the Delay for RDS MySQL Slow Query Logs?

Generally, the delay is 5 minutes. If the size of slow query logs reaches 10 MB within 5 minutes, the logs will be uploaded to OBS.

8.12.2 What's the Slow Query Threshold for Microsoft SQL Server?

The slow query threshold is 5 seconds.

8.12.3 How Can I Obtain Microsoft SQL Server Error Logs Using Commands?

Step 1 Log in to the Microsoft SQL Server client as user **rdsuser**.

Step 2 Run the following statement to query error logs:

```
EXECUTE master.dbo.rds_read_errorlog  
FileID,LogType,FilterText,FilterBeginTime,FilterEndTime
```

- *FileID*: indicates the ID of an error log. The value **0** indicates the latest logs.
- *LogType*: indicates the log type. The value **1** indicates error logs and value **2** indicates agent logs.
- *FilterText*: indicates a keyword, which can be **NULL**.
- *FilterBeginTime*: indicates the start time in queries, which can be **NULL**.
- *FilterEndTime*: indicates the completion time in queries, which can be **NULL**.

Example:

```
EXEC master.dbo.rds_read_errorlog 0,1,'FZYUN','2018-06-14 14:30','2018-06-14 14:31'
```

Figure 8-7 shows the query results.

Figure 8-7 Example query results

	LogDate	ProcessInfo	Text
1	2018-06-14 14:30:47.490	spid64	Starting up database 'FZYUN032020'.
2	2018-06-14 14:30:47.430	spid64	CHECKDB for database 'FZYUN029029' finished wit...
3	2018-06-14 14:30:47.400	spid64	Starting up database 'FZYUN029029'.
4	2018-06-14 14:30:47.330	spid64	CHECKDB for database 'FZYUN029027' finished wit...
5	2018-06-14 14:30:47.290	spid64	Starting up database 'FZYUN029027'.
6	2018-06-14 14:30:47.220	spid64	CHECKDB for database 'FZYUN02' finished without...
7	2018-06-14 14:30:47.180	spid64	Starting up database 'FZYUN02'.
8	2018-06-14 14:30:47.110	spid64	CHECKDB for database 'FZYUN' finished without e...
9	2018-06-14 14:30:47.080	spid64	Starting up database 'FZYUN'.
10	2018-06-14 14:30:46.840	spid64	Starting up database 'FZYUN032020'.

----End

8.12.4 Can I Export Statistics on RDS Slow Query Logs?

Sorry, statistics on RDS slow query logs cannot be exported.

8.13 Network Security

8.13.1 What Security Protection Policies Does RDS Have?

Network

- RDS runs your DB instances in a VPC, ensuring that the DB instances are isolated from other services.
- RDS uses security groups to ensure that only trusted sources can access your DB instances.
- RDS supports SSL connections to encrypt data during transmission.

Management

You can use the Identity and Access Management (IAM) service to manage RDS permissions.

8.13.2 How Can I Ensure the Security of RDS DB Instances in a VPC?

The VPC security group helps ensure the security of RDS in a VPC. In addition, an ACL can be used to allow or reject I/O network traffic for each subnet.

8.13.3 How Can Data Security Be Ensured During Transmission When I Access RDS Through an EIP?

When you access RDS through an EIP, service data will be transmitted on the public network. To prevent any potential data breaches, you are advised to use SSL to encrypt data transmitted on a public network. You can also use the Direct Connect or VPN services to encrypt data transmission.

8.13.4 How Can I Prevent Untrusted Source IP Addresses from Accessing RDS?

- If you enable public accessibility, your EIP DNS and database port may be vulnerable to hacking. To protect information such as your EIP, DNS, database port, database account, and password, you are advised to set the range of source IP addresses in the RDS security group to ensure that only trusted source IP addresses can access your DB instances.
- To prevent your database password from being cracked, set a strong password and periodically change it.
- RDS for SQL Server includes defense against brute force cracking. If malicious individuals have obtained your EIP DNS, database port, or database login information and attempt a brute force attack, your service connections may be delayed. In this case, you can restrict the source connections and change the database username and password to prevent further damage.

NOTE

RDS for MySQL and RDS for PostgreSQL do not include defense against brute force attacks.

For RDS for SQL Server, defense against brute force attacks is enabled by default and cannot be disabled.

8.13.5 How Can I Import the Root Certificate to a Windows or Linux OS?

Importing the Root Certificate to the Windows OS

1. Click **Start** and choose **Run**. In the displayed **Run** dialog box, enter **MMC** and press **Enter**.
2. On the displayed console, choose **File > Add/Remove Snap-in**.
3. In the left **Available snap-ins** pane of the displayed **Add or Remove Snap-ins** dialog box, select **Certificates** and click **Add**.
4. In the displayed **Certificates snap-in** dialog box, select **Computer account** and click **Next**.
5. In the displayed **Select Computer** dialog box, click **Finish**.
6. In the **Add or Remove Snap-ins** dialog box, click **OK**.
7. On the console, double-click **Certificates**.
8. Right-click **Trusted Root Certification Authorities** and choose **All Tasks > Import**.
9. In the displayed **Certificate Import Wizard** dialog box, click **Next**.
10. Click **Browse** to change the file type to **All Files (*.*)**.
11. Locate the downloaded root certificate ca.pem file and click **Open**. Then, click **Next**.

NOTICE

You must change the file type to **All Files (*.*)** because **.pem** is not a standard certificate extension name.

12. Click **Next**.
13. Click **Finish**.
14. Click **OK** to complete the import of the root certificate.

Importing the Root Certificate to the Linux OS

You can use a connection tool (such as WinSCP or PuTTY) to upload the certificate to any directory on a Linux OS.

8.13.6 How Can I Identify the Validity Period of an SSL Root Certificate?

When you connect to an RDS MySQL DB instance using an SSL connection, run the following command to check whether the certificate has expired:

```
show status like '%ssl_server%';
```

Update the root certificate to the latest version before it expires:

1. In the **DB Information** area on the **Basic Information** page, click  in the **SSL** field to download the root certificate or certificate bundle.
2. Reboot the DB instance for the new certificate to take effect.
3. Connect to the DB instance using the new certificate or certificate bundle.

NOTE

If a certificate is about to expire, replace it with an officially issued certificate to improve system security.

8.13.7 How Can I Identify Data Corruption?

- **Data tampering**
Lots of security measures are provided to ensure that only authenticated users have permissions to perform operations on database table records. Database tables can be accessed only through specific database ports.
Verifying package during primary/standby synchronization can prevent data tampering. MySQL uses the InnoDB storage engine to prevent data from being damaged.
- **DB instance servers may be powered off suddenly, causing database page corruption and database rebooting failures.**
If a primary DB instance becomes faulty, RDS switches to the standby DB instance within 1 to 5 minutes to provide services for you. Databases cannot be accessed during a failover. You must configure automatic reconnection between your applications and RDS to make sure that your applications are available after the failover.

9 Troubleshooting

9.1 RDS for MySQL

9.1.1 Backup and Restoration Issues

9.1.1.1 Backup Failures Due to DDL Operations

Scenario

Backups failed to be created for a DB instance in two consecutive days because there were DDL operations performed during the time window defined in the backup policy.

Possible Causes

RDS for MySQL uses XtraBackup to create full backups. To ensure data consistency, there are metadata lock conflicts between full backups and DDL operations. As a result, backups are blocked and fail due to timeout.

Run the **show processlist** command to check whether any DDL operation was performed in the backup time window.

Solution

1. Stop the DDL operations and perform a manual backup.
2. Do not perform DDL operations in the backup time window.

9.1.1.2 RDS for MySQL Backup Job Failure

Scenario

When a user ran the `mysqldump` command to back up RDS for MySQL data to an ECS that is in a different subnet from RDS, the backup job failed after running for 300 seconds.

Possible Causes

Replace the ECS where the backup job is executed with an ECS that is in the same subnet as RDS. The backup job is successfully executed.

- Network: There are no differences on latency and bandwidth between the two ECSs.
- Database: The **net_write_timeout** parameter is set to **300** on the RDS for MySQL database. The connection between the ECS and RDS for MySQL is interrupted after 300s regardless of whether data writes have been completed.

```
| net_write_timeout | 300 |
```

Procedure

Step 1 Understand the backup data flow, protocol, and port.

mysqldump uses TCP to connect to port 8635 of RDS. After the connection is established, the backup job starts.

Step 2 Compare the hardware configuration and OS version of the two ECSs.

1. Both of them use the same hardware configuration: two cores and 6 GB of memory.
2. Both of them use the same OS version: CentOS 7.4.

Step 3 Check whether the NIC rates are the same.

Step 4 Check whether the kernel parameter settings are the same. The result shows that the network parameters on the ECS where the backup job failed are not optimized.

```
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
vm.swappiness = 8
net.ipv4.neigh.default.gc_stale_time=120
net.ipv4.conf.all.rp_filter=0
net.ipv4.conf.default.rp_filter=0
net.ipv4.conf.default.arp_announce = 2
net.ipv4.conf.lo.arp_announce=2
net.ipv4.conf.all.arp_announce=2
net.ipv4.tcp_syncookies = 1
net.ipv4.tcp_synack_retries = 2
vm.swappiness=5
net.ipv4.tcp_fack=1
net.ipv4.tcp_tw_reuse=1
net.ipv4.tcp_tw_recycle=1
net.ipv4.tcp_fin_timeout=30
net.ipv4.tcp_keepalive_time=600
net.ipv4.tcp_keepalive_probes=5
net.ipv4.tcp_keepalive_intvl=15
net.ipv4.tcp_max_syn_backlog=4096
ipr2.ipv4.tcp_max_syn_backlog = 5000
net.core.rmem_default=434176
net.core.rmem_max=1048576
net.core.wmem_max=1048576
kernel.shmmax=5153960755
kernel.sem= 512 524288 32 1024
kernel.msgmni=128
net.core.somaxconn=20000
fs.file-max=800000
net.ipv4.tcp_mcm = 524288 706432 1048576
net.ipv4.tcp_wmem = 4096 16384 65536
net.ipv4.tcp_rmem = 4096 83780 65536
```


- Step 5** Set the kernel parameters of the ECS where the backup job failed to the same as those of the ECS where the backup job succeeded. Start a backup job again. The backup job is successful.

```
[root@szst-test01 data1# tail -fn 20 database_dump.20180413.sql
/*!40014 SET UNIQUE_CHECKS=OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=OLD_SQL_NOTES */;

-- Dump completed on 2018-04-13 13:41:40
```

----End

Solution

There is a large volume of data writes during the backup process across networks. The data write capability and TCP buffer on the backup end do not match the sending capability of the RDS. When the timeout period reaches the preset threshold (300s), the backup job failed. You can increase the TCP buffer by modifying the ECS kernel parameters to resolve this issue.

9.1.1.3 Manual Backups Taking More Time Than Automated Full Backups

Scenario

When 20 GB of data needed to be backed up, a full backup automatically performed by RDS took less time than a full backup performed manually.

Possible Causes

The disk throughput during the manual backup was lower than that during the automated backup.

Check the disk reads and writes on the Cloud Eye console.

Backups are stored in OBS buckets. The automated backup was performed during off-peak hours of OBS, and therefore took a short period of time. The manual backup was performed during peak hours of OBS, and therefore took much more time.

Solution

Perform full backups during OBS off-peak hours.

9.1.1.4 Automated Incremental Backup Failed Due to Full Storage

Scenario

Automated incremental backup failed.

Possible Causes

As your service data grows, the load on your DB instance is too heavy and the original storage space may be insufficient. When the DB instance status is **Storage full**, data cannot be written to databases, causing the failure of incremental backup.

Solution

Scale up storage space for your RDS instance. On the **Instance Management** page, locate the target DB instance and choose **More > Scale Storage Space** in the **Operation** column. Perform a full backup after the storage space is successfully scaled up and the incremental backup will be successful.

9.1.1.5 Precautions for Exporting Large Tables Through mysqldump

If the **-q** or **--quick** parameter is added when you use mysqldump to export data, the results of SELECT statements are not buffered in memory but directly exported. If this parameter is disabled, the results of SELECT statements are buffered in memory and then sent to the client.

- If you use mysqldump to back up only a small amount of data which can be stored in the idle memory buffer, disabling **-q** increases the export speed.
- Buffering a large amount of data may consume a large amount of memory, causing a memory swapping. If you use mysqldump to back up a large amount of data which cannot be stored in the memory buffer, enable **-q**. If **-q** is not enabled, a large amount of memory will be consumed and may even cause the database to break down due to out of memory.

Therefore, you are advised to enable the **-q** parameter when using mysqldump to back up data.

Example command:

```
mysqldump -uroot -p-P8635 -h 192.168.0.199 --set-gtid-purged=OFF --single-transaction --flush-logs -q test t1>t1.sql
```

9.1.1.6 Commands for Exporting Data Through mysqldump

Background

mysqldump is the most commonly used tool for importing and exporting MySQL data.

mysqldump Options

Table 9-1 Option description

Option Name	Description
add-drop-table	Adds the DROP TABLE statement before each data table is created.

Option Name	Description
events, E	Exports events.
routines, R	Exports stored procedures and customized functions.
flush-logs	Updates logs before the logs are exported.
no-create-db, n	Exports only data without adding of the CREATE DATABASE statement.
add-drop-database	Adds the DROP DATABASE statement before each database is created.
no-create-info, t	Exports only data without adding of the CREATE TABLE statement.
no-data, d	Exports only the database table structure.
set-gtid-purged=OFF	Does not export GTID statements.
hex-blob	Exports binary string fields in hexadecimal format.

Examples for Using mysqldump

- Export all data of databases **db1** and **db2**.

```
>db12.sqldb1 db2--hex-blob --set-gtid-purged=OFF --single-transaction --order-by-primary --flush-logs -q --databases 192.168.0.199 -h8635mysqldump -uroot -p -P
```
- Export the **t1** and **t2** tables of database **db1**.

```
.sqlt1_t2>t1 t2 --tables db1 --hex-blob --set-gtid-purged=OFF --single-transaction --order-by-primary --flush-logs -q --databases 192.168.0.199 -h8635mysqldump -uroot -p -P
```
- Export data whose id equals **1** from table **t1** in database **db1**.

```
.sqlt1_id>id=1 --where='t1 --tables db1 --hex-blob --set-gtid-purged=OFF --single-transaction --order-by-primary --flush-logs -q --databases 192.168.0.199 -h8635mysqldump -uroot -p -P
```
- Export all table structures in database **db1** without exporting data.

```
_table.sqldb1>db1set-gtid-purged=OFF --single-transaction --order-by-primary -n --flush-logs -q --databases --no-data -- 192.168.0.199 -h8635mysqldump -uroot -p -P
```
- Export all data excluding the tables and data in database **db1**.

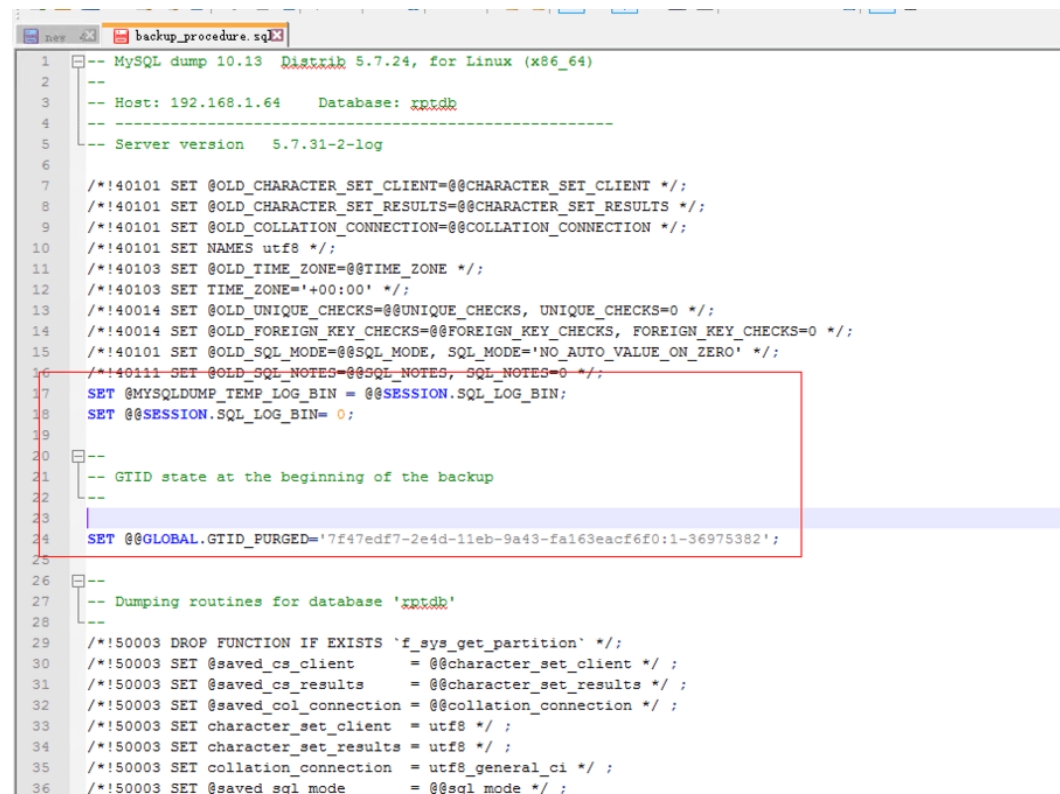
```
> others.sql db1 --set-gtid-purged=OFF -F -n -t -d -E -R8635 -P 192.168.0.199mysqldump -uroot -p -h
```

9.1.1.7 SQL Statements Such as SET @@SESSION.SQL_LOG_BIN Displayed After You Run mysqldump

Scenario

When you run mysqldump on a database, the following code is displayed:

Figure 9-1 Code



```
1  -- MySQL dump 10.13 Distrib 5.7.24, for Linux (x86_64)
2  --
3  -- Host: 192.168.1.64 Database: xptdb
4  --
5  -- Server version 5.7.31-2-log
6
7  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
8  /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
9  /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
10 /*!40101 SET NAMES utf8 */;
11 /*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
12 /*!40103 SET TIME_ZONE='+00:00' */;
13 /*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
14 /*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
15 /*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
16 /*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
17 SET @MYSQLDUMP_TEMP_LOG_BIN = @@SESSION.SQL_LOG_BIN;
18 SET @@SESSION.SQL_LOG_BIN= 0;
19
20 --
21 -- GTID state at the beginning of the backup
22 --
23
24 SET @@GLOBAL.GTID_PURGED='7f47edf7-2e4d-11eb-9a43-fa163eacf6f0:1-36975382';
25
26 --
27 -- Dumping routines for database 'xptdb'
28 --
29 /*!50003 DROP FUNCTION IF EXISTS `f_sys_get_partition` */;
30 /*!50003 SET @saved_cs_client      = @@character_set_client */;
31 /*!50003 SET @saved_cs_results     = @@character_set_results */;
32 /*!50003 SET @saved_col_connection = @@collation_connection */;
33 /*!50003 SET character_set_client  = utf8 */;
34 /*!50003 SET character_set_results = utf8 */;
35 /*!50003 SET collation_connection  = utf8_general_ci */;
36 /*!50003 SET @saved_sql_mode       = @@sql_mode */;
```

Fault Analysis

The parameter **gtid-mode** is set to **ON**.

If GTID is enabled for a database, you can use mysqldump to back up or dump all Global Transaction Identifiers (GTIDs) in the database or even to back up the whole RDS for MySQL database.

Solution

When the primary and standby RDS for MySQL databases are exported for backup and restoration, check whether GTID is enabled.

If GTID is enabled, add **--set-gtid-purged=OFF** to the **mysqldump** command during data dump.

9.1.1.8 Insufficient Permissions During Data Export Using mysqldump

Scenario

When you export database data with mysqldump using a specified user account, the error message "Access denied; you need (at least one of) the PROCESS privilege(s)" is displayed.

```
mysql> mysqldump -h192.168.1.100 -P3306 -uadmin -p123456 --set-gtid-purged=off --skip-lock-tables zzkj > zzkj.sql
mysqldump: [warning] using a password on the command line interface can be insecure.
mysqldump: Error: 'Access denied; you need (at least one of) the PROCESS privilege(s) for this operation' when trying to dump tablespaces
```

Possible Causes

The user account does not have the PROCESS permission.

Solution

Grant the PROCESS permission to the user account as the administrator.

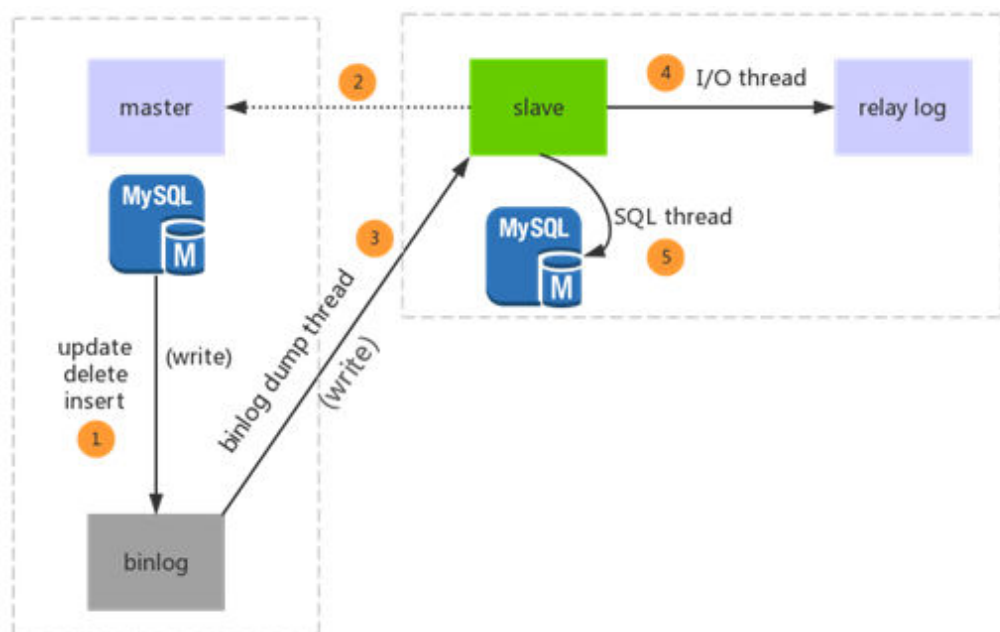
```
GRANT SELECT, PROCESS ON *.* TO 'dump_user'@'%';
FLUSH PRIVILEGES;
```

9.1.2 Primary/Standby Replication Issues

9.1.2.1 How Primary/Standby Replication Works

RDS for MySQL standby instances, read replicas, self-built standby databases, and DR instances established through DRS all use MySQL Primary-Secondary Replication. This section describes how MySQL Primary-Secondary Replication works.

Primary-Secondary Replication Process



- When data is updated on the master node, the update is written to the binlog of the master node as an event. The event types include INSERT, DELETE, UPDATE, and QUERY.
- When slave nodes are connected to the master node, a binlog dump thread is created for each slave node.
- When the binlog of the master node changes, the binlog dump threads notify all the slave nodes of the change and push the binlog updates to the slave nodes.
- After receiving the binlog updates, the I/O thread of each slave node writes the updates to the local relay log.
- The SQL thread of each slave node reads the relay log and replays the operation (such as DML and DDL) based on the event in the relay log.

Seconds_Behind_Master Calculation Method

Seconds_Behind_Master indicates the primary/standby replication delay, which can be obtained by running **show slave status**. The following pseudocode shows how the Seconds_Behind_Master value is calculated.

```
if (SQL thread is running)
//If the SQL thread is started
{
    if (SQL thread processed all the available relay log)
        //If the binlog pulled by the I/O thread from the master node is the same as that read by the SQL
        thread from the relay log
        {
            if (IO thread is running)
                //If the I/O thread is started, Seconds_Behind_Master is set to 0.
                print 0;
            else
                //If the I/O thread is not started, Seconds_Behind_Master is set to NULL.
                print NULL;
        }
    else
        //If the SQL thread did not process all events written by the I/O thread, the value of
        Seconds_Behind_Master needs to be calculated.
        Calculate the value of Seconds_Behind_Master using the formula;
}
else
    //If the SQL thread is not started, Seconds_Behind_Master is set to NULL.
    print NULL;
```

In the pseudocode, the following formula is used to calculate the value of **Seconds_Behind_Master**:

$$\text{Seconds\_Behind\_Master} = \text{time}(0) - \text{last\_master\_timestamp} - \text{clock\_diff\_with\_master}$$

Explanations of the variables:

1. **time(0)**: The system time of the current slave server.
2. **clock_diff_with_master**: The difference between the system time of the slave server and that of the master server. Generally, the value is 0. If the system time of the slave server is different from that of the master server, the calculated value of **Seconds_Behind_Master** is inaccurate.
3. **last_master_timestamp**: The execution time of an event on the master node that is updated by the SQL thread. This variable is calculated and updated when the slave node replays the event in the relay log. The update time in the

multi-threaded slave (MTS) replication is different from that in single-threaded replication. MTS is enabled by default.

- MTS: The SQL thread of the slave node updates the value of **last_master_timestamp** after each transaction is executed. The update is performed by transaction. Therefore, large transactions and DDL operations may cause a long primary/standby replication delay. For details, see [Automatic Recovery of Extended Primary/Standby Replication Delay](#).
- Single-threaded replication: After the SQL thread of the slave node reads a transaction from the relay log, **last_master_timestamp** is updated before the transaction is executed. The update is performed by transaction.

In conclusion, the formula for calculating **Seconds_Behind_Master** can be understood as follows:

$$\text{Seconds\_Behind\_Master} = \text{System time of the current slave server} - \text{Execution time of the transaction on the master node that is updated by the SQL thread} - \text{Difference between the system time of the slave server and that of the master server}$$

9.1.2.2 Automatic Recovery of Extended Primary/Standby Replication Delay

Scenario

The primary/standby replication delay of a DB instance was long, kept increasing for a period of time, and then automatically recovered.

Possible Causes

According to [Primary/Standby Replication Delay Scenarios and Solutions](#) and [How Primary/Standby Replication Works](#), this problem is caused by large transactions or DDL operations.

You can analyze full logs or slow query logs to check whether there are large transactions or DDL operations.

As shown in the following figure, if a DDL operation for adding an index was recorded in the slow query logs, the table contained hundreds of millions of data records, and the execution took about one day, the replication delay kept increasing when the DDL operation was replayed on the read replica or standby node. After the DDL operation was replayed, the replication delay dropped back to the normal range.

```
# Time: 2022-04-14 10:00:00.734954+08:00
# User@Host: root@ [192.168.0.185] Id: 20167391
# Query_time: 82714.322375 Lock_time: 0.000173 Rows_sent: 1 Rows_examined: 999999999 Thread_id: 20167391 Schema:
# Read_last: 0 Read_key: 0 Read_next: 0 Read_prev: 0 Read_rnd: 0 Read_rnd_next: 493972 Sort_merge_passes: 0
# Created_tmp_tables: 0 Start: 2022-04-14 10:00:00.734954+08:00 End: 2022-04-14 10:00:00.734954+08:00
# QC_Hit: No Full_scan: No Full_join: No Tmp_table: No Tmp_table_on_disk: No Filesort: No Filesort_on_disk: No
SET timestamp=1649777004;
alter table t1 add index ind1(name,no);
```

Solution

- Wait until the DDL operation is complete.

- Add indexes during off-peak hours.

9.1.2.3 Primary/Standby Replication Delay Scenarios and Solutions

RDS for MySQL standby instances, read replicas, self-built standby databases, and DR instances established through DRS all use MySQL Primary-Secondary Replication. Primary/standby replication can be implemented in asynchronous or semi-synchronous mode. In either mode, primary/standby replication delay is inevitable for some statements.

Symptom: The replication delay of the standby instance or read replica is too long, or even an alarm is generated for too long replication delay.

Scenario 1: Large Transaction Executed on the Primary Instance

A large transaction refers to a transaction containing a large quantity of data update operations. For example, a transaction includes tens of thousands of DML (INSERT, UPDATE, and DELETE) operations, or an SQL statement updates tens of thousands of rows of data in batches. The execution of a large transaction usually takes a long time (minutes). After a large transaction is executed on the primary instance, a large number of binlogs are generated. It takes much more time for the standby instance or read replica to obtain these binlogs than for a common transaction. In addition, replaying the transaction also takes much time, at least the same as the time for the transaction being executed on the primary instance. As a result, replication delay occurs on the standby node or read replica.

Troubleshooting:

- For a large transaction that contains a large number of DML statements, run the following command to find the statement that is executed for a long time:
select t.*,to_seconds(now())-to_seconds(t.trx_started) idle_time from INFORMATION_SCHEMA.INNODB_TRX t \G;
- For a large transaction where an SQL statement executes a large amount of data updates, run **show full processlist** to check whether any DELETE or UPDATE statements are executed for a long time.
- Analyze full logs or slow query logs to check whether there are any large transactions.

Solution:

- To ensure data consistency between the primary and standby instances, wait until the large transaction is complete.
- Split a large transaction into small transactions and execute them in batches. For example, you can use the WHERE condition or LIMIT statement to limit the amount of data to be updated each time.

Scenario 2: Table Without a Primary Key Updated

RDS for MySQL binlogs use the row-based logging. When data in a row is updated, an event record in row format is generated in the binlog. For example, if an UPDATE statement updates 100 rows of data in a table, 100 rows of update records will be generated in the binlog. During playback on the standby instance or read replica, 100 single-row updates are executed.

When replaying binlog events of the primary instance, the standby instance or read replica searches for rows to be modified based on the primary key or secondary index of the table. If no primary key is created for the table, a large number of full table scans are generated. As a result, the application speed of the binlog is reduced and replication delay occurs.

Troubleshooting:

Use **show create table xxx** to find out the table with the UPDATE or DELETE statement executed slowly and check whether the table has a primary key.

Solution:

Add a primary key for the table that does not have a primary key, and a secondary index for the table that does not have a secondary index.

Scenario 3: DDL Operation Performed

A DDL operation usually takes a long time, especially when the table contains a large amount of data. Generally, the time for replaying a DDL operation on the standby instance or read replica is almost the same as that for executing the DDL operation on the primary instance. Therefore, after a DDL operation is performed on a large table in the primary instance, the replication time will increase when the standby instance or read replica replays the operation.

Solution:

Wait until the DDL execution is complete. Perform DDL operations during off-peak hours.

Scenario 4: Read Replica Waiting for a Metadata Lock

When a long transaction is being executed on a table in a read replica, the DDL operation of the table synchronized from the primary instance will be blocked due to a metadata lock, and subsequent binlog playback of the table is also blocked. As a result, the replication delay increases.

Troubleshooting:

1. Log in to the read replica and run the following command to check whether there are any transactions that have been executed for a long time:
select t.*,to_seconds(now())-to_seconds(t.trx_started) idle_time from INFORMATION_SCHEMA.INNODB_TRX t \G;
2. View the metadata lock view of the read replica to check whether a metadata lock conflict occurs.
select * from information_schema.metadata_lock_info;
Find the blocked session based on the thread ID in the metadata lock view.
For more information.

Solution:

Kill the long transaction that blocks DDL operations on the read replica, or commit the long transaction on the application.

Scenario 5: Read Replica Specifications Smaller Than the Primary Instance

If the specifications of the read replica or DR instance established through DRS are smaller than those of the primary instance and the write load of the primary instance increases to a certain extent, the read replica or DR instance cannot replay binlogs in a timely manner due to insufficient resources. As a result, the replication delay increases.

Solution:

Scale up the specifications of the read replica or DR instance to match the specifications of the primary instance.

9.1.2.4 Abnormal Replication Between Primary and Standby RDS DB Instances

Scenario

The replication relationship between primary and standby RDS DB instances is abnormal. A possible cause is that the default security group rule is deleted.

Solution

- Step 1** On the **Instances** page, click the target DB instance.
- Step 2** On the **Basic Information** page, click the security group name.
- Step 3** On the **Inbound Rules** tab, click **Add Rule**. In the displayed dialog box, select **All** for **Protocol & Port** and **Security group** for **Source**. In the drop-down list, select the same security group as the displayed one on the **Basic Information** page.
- Step 4** Wait until the rule is added. Check that the replication relationship between primary and standby DB instances is restored.

----End

9.1.2.5 Primary/Standby Replication Delay Increases Sharply and Then Decreases

Scenario

The metric **Real-Time Replication Delay** of a read replica increased sharply and then fell back on the Cloud Eye console.

Possible Causes

- This problem is related to the method of calculating the value of **Seconds_Behind_Master**. For details about how to calculate this field, see [How Primary/Standby Replication Works](#).
- The replication delay peak occurred because the I/O thread of the read replica received a new binlog file, but the SQL thread did not start to replay the new binlog file. As a result, the value of **last_master_timestamp** for calculating **Seconds_Behind_Master** is the time when the previous binlog transaction

was executed on the primary node, which is different from the system time (**time (0)**) of the read replica. When the SQL thread started to parse the new binlog file, the replication delay immediately decreased.

- This problem occurs occasionally and does not affect your workloads.

The following symptom was found in the binlog files generated during the period when the replication delay increased sharply and then fell back:

The difference between the start time of the first transaction in the new binlog file and the end time of the last transaction in the previous binlog file was the same as the difference between the sudden increase and fallback times.

```
# at 63746
#210923 16:06:03 server id 3559516408 end_log_pos 65793 CRC32 0x8cccd7df Rotate to mysql-bin.011018 pos: 4
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;

#210923 16:06:03 server id 3559516408 end_log_pos 126 CRC32 0xd7e526a7 Start: binlog v 4, server v 5.7.32-2-log created 210923 16:06:03
# at 126
#210923 16:06:03 server id 3559516408 end_log_pos 197 CRC32 0x722a15a4 Previous-GTIDs
# 353f4e30-e856-11eb-b9c4-fa163ef03ae7:1-4640966
# at 197
#210923 16:09:16 server id 3559516408 end_log_pos 262 CRC32 0x6ec61958 GTID last_committed=0 sequence_number=1 rbr_only=yes
/*!50718 SET TRANSACTION ISOLATION LEVEL READ COMMITTED'/*!*/;
SET @@SESSION.GTID_NEXT= '353f4e30-e856-11eb-b9c4-fa163ef03ae7:4640967'/*!*/;

# at 65746
#210923 16:11:03 server id 3559516408 end_log_pos 65793 CRC32 0xelf2a4fb Rotate to mysql-bin.011019 pos: 4
SET @@SESSION.GTID_NEXT= 'AUTOMATIC' /* added by mysqlbinlog */ /*!*/;
DELIMITER ;
# End of log file
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;

#210923 16:11:03 server id 3559516408 end_log_pos 126 CRC32 0x6bcc3575 Start: binlog v 4, server v 5.7.32-2-log created 210923 16:11:03
# at 126
#210923 16:11:03 server id 3559516408 end_log_pos 197 CRC32 0xc8cb1f82 Previous-GTIDs
# 353f4e30-e856-11eb-b9c4-fa163ef03ae7:1-4640967
# at 197
#210923 16:14:16 server id 3559516408 end_log_pos 262 CRC32 0xc9ab23b GTID last_committed=0 sequence_number=1 rbr_only=yes
/*!50718 SET TRANSACTION ISOLATION LEVEL READ COMMITTED'/*!*/;
SET @@SESSION.GTID_NEXT= '353f4e30-e856-11eb-b9c4-fa163ef03ae7:4640968'/*!*/;
```

Solution

No action is required. This scenario is normal and occasionally occurs.

9.1.3 Parameter-related Issues

9.1.3.1 long_query_time Changes Fail to Take Effect

Scenario

After the value of **long_query_time** was changed from 0.1s to 0.2s on the console, slow query logs still contained slow SQL statements whose execution duration was less than 0.2s.

Possible Causes

Changes to **long_query_time** on the console take effect globally. After the value of **long_query_time** is changed, new connections use the new value, but old connections still use the original value (0.1s in this case). Therefore, slow SQL queries whose execution duration was less than 0.2s were generated for old connections.

This problem is caused by the MySQL engine. Changing any of the parameters that take effect on all databases may also have this problem.

Solution

If you require certain session connections to use the new value after **long_query_time** is changed, close the session connections and re-establish new connections.

9.1.3.2 Incorrect GROUP_CONCAT Results

Scenario

When the GROUP_CONCAT() function was used in an SQL statement, the result did not meet the expectation.

Possible Causes

The GROUP_CONCAT() function returned a string result consisting of concatenated values in the group. However, the **group_concat_max_len** parameter limited the result length of this function.

Example:

```
mysql> show variables like 'group_concat_max_len';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| group_concat_max_len | 1024 |
+-----+-----+
1 row in set (0.01 sec)

mysql> select GROUP_CONCAT(c1,c2,c3) from dis;
+-----+
| GROUP_CONCAT(c1,c2,c3) |
+-----+
| 111,222,322            |
+-----+
```

```
mysql> set session group_concat_max_len=8;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like 'group_concat_max_len';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| group_concat_max_len | 8     |
+-----+-----+
1 row in set (0.01 sec)

mysql> select GROUP_CONCAT(c1,c2,c3) from dis;
+-----+
| GROUP_CONCAT(c1,c2,c3) |
+-----+
| 111,222,                |
+-----+
```

Solution

Change the value of **group_concat_max_len** to adapt to the result length of the GROUP_CONCAT() function.

9.1.3.3 [ERROR] 1071 Reported When an Index Fails to Be Created for RDS for MySQL

Scenario

The index failed to be created because its length exceeds the upper limit. The following error was reported:

[ERROR] 1071 - Specified key was too long; max key length is 3072 bytes

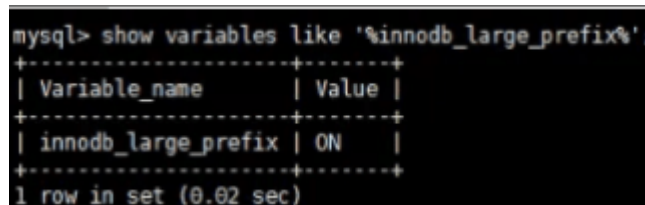
This problem may occur in MySQL-8.0.20.5.

Fault Analysis

The InnoDB table engine has a length limit on index prefixes.

By default, an index prefix can contain a maximum of 767 bytes, but if **innodb_large_prefix** is set to **ON**, the index prefix length is increased to 3,072 bytes.

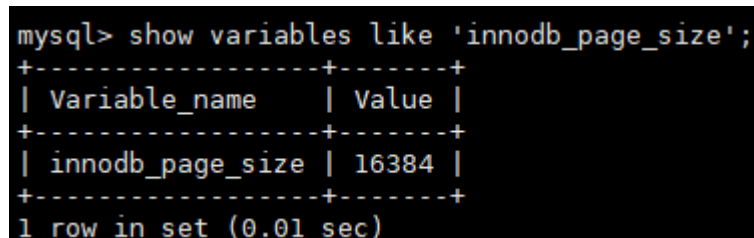
```
SHOW VARIABLES LIKE '%innodb_large_prefix%';
```



```
mysql> show variables like '%innodb_large_prefix%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_large_prefix | ON    |
+-----+-----+
1 row in set (0.02 sec)
```

The length of an index prefix also depends on the InnoDB page size. If the parameter **innodb_page_size** is set to its default value 16 KB, the maximum index prefix length is 3,072 bytes, but if this parameter is set to 8 KB, the maximum index prefix length is 1,536 bytes. If this parameter is set to 4 KB, the maximum length of the index prefix is 768 bytes.

```
SHOW VARIABLES LIKE '%innodb_page_size%';
```



```
mysql> show variables like 'innodb_page_size';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| innodb_page_size | 16384 |
+-----+-----+
1 row in set (0.01 sec)
```

Check the structure of the problem table and query all supported character sets and their byte usage:

```
SHOW CHARACTER SET;
```

```
mysql> SHOW CHARACTER SET;
```

Charset	Description	Default collation	Maxlen
armscii8	ARMSSCII-8 Armenian	armscii8_general_ci	1
ascii	US ASCII	ascii_general_ci	1
big5	Big5 Traditional Chinese	big5_chinese_ci	2
binary	Binary pseudo charset	binary	1
cp1250	Windows Central European	cp1250_general_ci	1
cp1251	Windows Cyrillic	cp1251_general_ci	1
cp1256	Windows Arabic	cp1256_general_ci	1
cp1257	Windows Baltic	cp1257_general_ci	1
cp850	DOS West European	cp850_general_ci	1
cp852	DOS Central European	cp852_general_ci	1
cp866	DOS Russian	cp866_general_ci	1
cp932	SJIS for Windows Japanese	cp932_japanese_ci	2
dec8	DEC West European	dec8_swedish_ci	1
eucjpms	UJIS for Windows Japanese	eucjpms_japanese_ci	3
euckr	EUC-KR Korean	euckr_korean_ci	2
gb18030	China National Standard GB18030	gb18030_chinese_ci	4
gb2312	GB2312 Simplified Chinese	gb2312_chinese_ci	2
gbk	GBK Simplified Chinese	gbk_chinese_ci	2
geostd8	GEOSTD8 Georgian	geostd8_general_ci	1
greek	ISO 8859-7 Greek	greek_general_ci	1
hebrew	ISO 8859-8 Hebrew	hebrew_general_ci	1
hp8	HP West European	hp8_english_ci	1
keybcs2	DOS Kamenicky Czech-Slovak	keybcs2_general_ci	1
koi8r	KOI8-R Relcom Russian	koi8r_general_ci	1
koi8u	KOI8-U Ukrainian	koi8u_general_ci	1
latin1	cp1252 West European	latin1_swedish_ci	1
latin2	ISO 8859-2 Central European	latin2_general_ci	1
latin5	ISO 8859-9 Turkish	latin5_turkish_ci	1
latin7	ISO 8859-13 Baltic	latin7_general_ci	1
macce	Mac Central European	macce_general_ci	1
macroman	Mac West European	macroman_general_ci	1
sjis	Shift-JIS Japanese	sjis_japanese_ci	2
swe7	7bit Swedish	swe7_swedish_ci	1
tis620	TIS620 Thai	tis620_thai_ci	1
ucs2	UCS-2 Unicode	ucs2_general_ci	2
ujis	EUC-JP Japanese	ujis_japanese_ci	3
utf16	UTF-16 Unicode	utf16_general_ci	4
utf16le	UTF-16LE Unicode	utf16le_general_ci	4
utf32	UTF-32 Unicode	utf32_general_ci	4
utf8	UTF-8 Unicode	utf8_general_ci	3
utf8mb4	UTF-8 Unicode	utf8mb4_0900_ai_ci	4

When the problem table uses the utf8mb4 character set, each character occupies 4 bytes. This means that the index prefix only contains 768 ($3072/4 = 768$) characters if the index prefix has 3072 bytes. Therefore, you only need to set the index prefix length in the CREATE TABLE statement to **768** or modify the index field to keep it less than 3072 bytes.

Solutions

Change the length of the index field.

9.1.3.4 Tables Failed to Be Found After Case-Sensitivity Setting Changes for RDS for MySQL

Scenario

After the RDS for MySQL parameter **lower_case_table_names** was set to case sensitive, a table containing uppercase letters was created. However, after the parameter was set to case insensitive, the table containing uppercase letters, such as tbl_newsTalking, cannot be found.

Case: When a backup is restored to a new instance, restoration will fail if the parameter case-sensitive setting of the new instance is different from that of the original instance.

Solution

- Step 1** Change the value of **lower_case_table_names** to **0**, indicating that table names are case sensitive.
- Step 2** Reboot the database.
- Step 3** Change uppercase letters in table names to corresponding lowercase letters.
- Step 4** Change the value of **lower_case_table_names** to **1**, indicating that table names are case insensitive.
- Step 5** Reboot the database, or export the entire database and import it again.

----End

NOTE

- The names of databases, tables, and variables, and table alias must be case sensitive.
- Column names and aliases are always case insensitive. Field values are case insensitive by default.
- For RDS for MySQL 5.7, you can specify case sensitivity for table names when creating an instance on the console or using APIs, or set the **lower_case_table_names** parameter after an instance is created.
- For RDS for MySQL 8.0, you can only specify case sensitivity for table names when creating an instance on the console or using APIs.

9.1.3.5 Timeout Parameters

The following table lists the RDS for MySQL timeout parameters.

Table 9-2 Parameter description

Parameter	Reboot Required	Description
connect_timeout	No	Number of seconds that the mysqld server waits for a connection packet before responding with Bad handshake . If the network quality is poor, you can increase the value of this parameter.
idle_readonly_transaction_timeout	No	Time in seconds that the server waits for idle read-only transactions before killing the connection. (Supported in versions later than 5.7.23)
idle_transaction_timeout	No	Time in seconds that the server waits for idle transactions before killing the connection. If this parameter is set to the default value 0 , connections are never killed. (Supported in versions later than 5.7.23)
idle_write_transaction_timeout	No	Time in seconds that the server waits for idle read-write transactions before killing the connection. If this parameter is set to the default value 0 , connections are never killed. (Supported in versions later than 5.7.23)
innodb_lock_wait_timeout	No	Length of time in seconds an InnoDB transaction waits for a row lock before giving up.
innodb_rollback_on_timeout	Yes	InnoDB rolls back only the last statement on a transaction timeout by default. If innodb_rollback_on_timeout is specified, a transaction timeout causes InnoDB to abort and roll back the entire transaction.
lock_wait_timeout	No	Timeout in seconds for attempts to acquire metadata locks
net_read_timeout	No	Number of seconds to wait for more data from a connection before aborting the read.
net_write_timeout	No	Number of seconds to wait for a network packet to be written to a TCP connection before aborting the write.

Parameter	Reboot Required	Description
interactive_timeout	No	Number of seconds the server waits for activity on an interactive connection before closing it.
wait_timeout	No	Number of seconds the server waits for activity on a noninteractive connection before closing it.

9.1.4 Performance Issues

9.1.4.1 DB Instance Becoming Read-Only Due to Full Storage

Scenario

The following error information was displayed for an RDS for MySQL instance:

The MySQL server is running with the --read-only option so it cannot execute this statement

Troubleshooting

1. Go to the instance details page and check whether the storage space is full.
2. Log in to the database and check the **read_only** variable.
show variables like 'read_only';
3. From the previous steps, you can find that the storage space is full and the instance status has changed to read-only. As a result, the SQL statement fails to be executed.

Solution

[Scaling up Storage Space.](#)

9.1.4.2 High Storage Usage Due to Uncleared Old Binlogs

Scenario

The storage usage of a read replica or primary instance was high. By running **SHOW BINARY LOGS** or **SHOW MASTER LOGS**, it was found that a large number of old binlogs were not cleared.

Possible Causes

In most cases, if the [binlog retention period](#) is configured, the binlogs that have been backed up to OBS but exceed the retention period will be automatically cleared. If binlogs are not cleared within a long period of time, check whether it is caused by replication exceptions.

Troubleshooting:

1. **Check error logs of the instance** for any records about binlog purging failures.
2022-01-18T05:39:03.139207+08:00 29 [Warning] file ./mysql-bin.106259 was not purged because it was being read by thread number 27490757
2. Check whether a local standby database exists or whether tools such as Canal are used to listen to binlogs of the instance. If the primary database does not receive the information that the binlogs have been obtained by the standby database or tool, the binlogs will not be deleted. As a result, binlogs are stacked.
3. Based on the records obtained in [1](#), analyze the logs of the local standby database or Canal and check network connectivity to find out the cause.

Solution

- Temporarily stop the binlog listening task of the instance so that the instance can automatically clear binlogs.
- If there is a local standby database, re-establish the replication relationship.
- If a tool such as Canal is used, re-create a binlog pull task.

9.1.4.3 Read Replica Occupies Far More Storage Than the Primary Instance

Scenario

The storage usage of an RDS for MySQL read replica was 195 GB higher than that of the primary DB instance.

Troubleshooting

Check the transactions running on the read replica.

```
trx_waited_time: 0
***** 12. row *****
      trx_id: 422175775626376
      trx_state: RUNNING
      trx_started: 2022-07-22 11:00:26
      trx_requested_lock_id: NULL
      trx_wait_started: NULL
      trx_weight: 0
      trx_mysql_thread_id: 575769551
      trx_query: SELECT 'movie_id' FROM 'hg_app_pay_movie' WHERE 'member_id' = '12831953'
      trx_operation_state: fetching rows
      trx_tables_in_use: 1
      trx_tables_locked: 0
      trx_lock_structs: 0
      trx_lock_memory_bytes: 1136
      trx_rows_locked: 0
      trx_rows_modified: 0
      trx_concurrency_tickets: 0
      trx_isolation_level: REPEATABLE READ
      trx_unique_checks: 1
      trx_foreign_key_checks: 1
      trx_last_foreign_key_error: NULL
      trx_adaptive_hash_latched: 0
      trx_adaptive_hash_timeout: 0
      trx_is_read_only: 1
      trx_autocommit_non_locking: 1
      trx_schedule_weight: NULL
      trx_waited_time: 0
***** 13. row *****
      trx_id: 422175775614616
      trx_state: RUNNING
      trx_started: 2022-07-21 15:27:39
      trx_requested_lock_id: NULL
      trx_wait_started: NULL
      trx_weight: 0
      trx_mysql_thread_id: 546262844
      trx_query: SELECT COUNT(*) AS tp_count FROM 'hg_member_history' 'a' LEFT JOIN 'hg_member_info' 'b' ON 'a'.'member_id'='b'.'id' LEFT JOIN 'hg_app_movie'
      trx_operation_state: starting index read
      trx_tables_in_use: 3
      trx_tables_locked: 0
      trx_lock_structs: 0
      trx_lock_memory_bytes: 1136
      trx_rows_locked: 0
      trx_rows_modified: 0
      trx_concurrency_tickets: 0
```

As shown in the preceding figure, a long transaction started one day ago and was not committed yet. The undo logs accumulated for almost a whole day and were not cleared. Therefore, the storage usage was high.

Solution

- Method 1: Wait until the transaction is committed. After that, the undo logs will be cleared, releasing storage space.
- Method 2: Kill the corresponding session to stop the long transaction.

9.1.4.4 Slow SQL Execution Due to Hot and Cold Data Problems

Scenario

When you migrate data from a self-built MySQL database or a peer vendor's MySQL database to an RDS for MySQL instance, the execution speed of the same SQL statement is much lower than that of the source database.

Possible Causes

The execution speed of the same SQL statement differs greatly when it is executed for the first time and the second time. This is determined by the MySQL buffer pool mechanism.

- When the statement is executed for the first time, data is stored on the disk, which is called cold data. Reading cold data takes a certain period of time.
- The data you have queried is then cached in the buffer pool of the memory. It is called hot data and can be quickly accessed in the memory. When you execute the same statement for the second time, data is read from the buffer pool, which is much faster than reading data from disks.

In this troubleshooting case, the data you queried in the source database is frequently accessed data, that is, hot data. Thus, it can be read at a high speed. After the data is migrated to the RDS for MySQL instance, when you execute the same SQL statement on the new database for the first time, the data you expect to query is probably cold data. This time, the access speed is slow. If you run the statement again, the data access speed will greatly improve.

Solution

This issue is not an exception. In the same database, it usually takes much time to execute a statement for the first time, but when the statement is executed again, it gets much faster. The access speed improves because reading hot data from the buffer pool is much faster than reading cold data from disks.

9.1.4.5 Tablespace Bloat

Scenario

Tablespace bloat often occurs in RDS for MySQL instances. For example, a table contains only 11,774 rows of data but occupies 49.9 GB of storage space. After the table is exported to a local directory, it occupies only 800 MB.

Possible Causes

Cause 1: Parallel Migration During DRS Full Migration

During full migration, DRS uses row-level parallel migration to ensure migration performance and transmission stability. If the source database data is compact, table bloat may occur after data is migrated to RDS for MySQL. As a result, storage usage is much greater than that of the source database.

Cause 2: Table Fragmentation After a Large Number of Deletions

When data is deleted, RDS for MySQL does not reclaim the storage occupied by the deleted data. Instead, it only marks the deletion and fills the space with new data if any. If there is no data to fill, tablespace bloat occurs, causing table fragmentation.

You can run the following SQL statement to query detailed information about a table. The **DATA_FREE** field indicates the size of tablespace fragmentation.

```
select * from information_schema.tables where table_schema='db_name' and table_name = 'table_name'\G
```

```
mysql> select * from information_schema.tables where table_schema='mall19wo' and table_name='deliveryman_track'\G
***** 1. row *****
TABLE_CATALOG: def
TABLE_SCHEMA: mall19wo
TABLE_NAME: deliveryman_track
TABLE_TYPE: BASE TABLE
ENGINE: InnoDB
VERSION: 10
ROW_FORMAT: Dynamic
TABLE_ROWS: 11968
AVG_ROW_LENGTH: 4479273
DATA_LENGTH: 53607940096
MAX_DATA_LENGTH: 0
INDEX_LENGTH: 802816
DATA_FREE: 54668558336
AUTO_INCREMENT: 94507
CREATE_TIME: 2022-06-28 23:39:00
UPDATE_TIME: 2022-07-07 11:03:22
CHECK_TIME: NULL
TABLE_COLLATION: utf8mb4_general_ci
CHECKSUM: NULL
CREATE_OPTIONS: row_format=DYNAMIC
TABLE_COMMENT:
1 row in set (0.00 sec)
```

Solution

Run the following SQL command to optimize the tablespace:

```
optimize table table_name;
```

Note: The **optimize table** command locks tables for a short period of time. Therefore, you are advised to optimize tablespaces during off-peak hours.

9.1.4.6 Full Storage Caused by Complex Queries

Scenario

The storage usage of a primary instance or read replica is occasionally high or reaches 100%, while the storage usage of the standby instance or other read replicas is within a normal range.

Possible Causes

When you run complex queries on data of an RDS for MySQL database, RDS creates temporary tables to store the data and operations such as GROUP BY, ORDER BY, DISTINCT, and UNION are executed on the data in the temporary tables. When memory is insufficient, storage space is consumed.

Troubleshooting:

1. Check the storage usage of the standby instance and other read replicas. If the storage usage of such instances is normal, the high storage usage of the primary instance or read replica is related to SQL queries running on it.
2. Check the instance slow query logs to find whether any slow-running queries occurred when the storage usage was high.
3. If there is a slow-running query, run the **explain** *[slow SQL statement]* command to analyze the SQL statement.
4. Check whether the **extra** column in the command output contains **using temporary** or **using filesort**. If yes, the statement uses temporary tables or temporary files. If a large amount of data is queried, the storage usage is high.

Solution

1. Optimize the query statement by adopting the following measures:
 - Add a proper index.
 - Use the WHERE condition.
 - Rewrite the SQL statement to optimize the execution plan.
 - If temporary tables are necessary, reduce the number of concurrent requests.
2. Workaround: Scale up storage space temporarily. Optimizing complex query statements cannot reduce the storage usage right away.

9.1.4.7 Instance Class Change or Minor Version Upgrade Failure Caused by Long Transactions

Scenario

An instance class change or a minor version upgrade fails due to long transactions.

Possible Causes

- To minimize the impact on workloads, an instance class change or a minor version upgrade is performed in rolling mode and therefore a primary/standby switchover is required.
- To ensure data consistency, the primary instance is set to read-only before the switchover.
- If there are long transactions executing on the primary instance, setting the primary instance to read only will time out or fail. As a result, the instance class change or minor version upgrade fails.

Solution

1. Run the **show processlist** command to view the transactions that are being executed. Then run the following command to find out the long transaction:
select t.*,to_seconds(now())-to_seconds(t.trx_started) idle_time from INFORMATION_SCHEMA.INNODB_TRX t \G;
 Example output:

```
mysql> select t.*,to_seconds(now())-to_seconds(t.trx_started) idle_time from INFORMATION_SCHEMA.INNODB_TRX t \G
***** 1. POW *****
      trx_id: 6168
      trx_state: RUNNING
      trx_started: 2021-09-16 11:08:27
      trx_requested_lock_id: NULL
      trx_wait_started: NULL
      trx_weight: 3
      trx_mysql_thread_id: 231
      trx_query: NULL
      trx_operation_state: NULL
      trx_tables_in_use: 0
      trx_tables_locked: 1
      trx_lock_structs: 3
      trx_lock_memory_bytes: 1136
      trx_rows_locked: 2
      trx_rows_modified: 0
      trx_concurrency_tickets: 0
      trx_isolation_level: REPEATABLE READ
      trx_unique_checks: 1
      trx_foreign_key_checks: 1
      trx_last_foreign_key_error: NULL
      trx_adaptive_hash_latched: 0
      trx_adaptive_hash_timeout: 0
      trx_is_read_only: 0
      trx_autocommit_non_locking: 0
      idle_time: 220
```

In the preceding command output, **idle_time** is the calculated time the transaction stays idle. **trx_mysql_thread_id** indicates the thread ID of the transaction, which is the thread ID queried by running **show processlist**.

The value of **trx_query** is **NULL**, but it does not mean that the transaction is not executed. If the transaction contains multiple SQL statements and all of them have been executed, no statement is displayed in the **trx_query** field. If the transaction is being executed, InnoDB cannot determine whether there are any subsequent SQL statements and when the transaction is committed. In this case, **trx_query** cannot determine which SQL statement is being executed, so **NULL** is displayed.

2. Kill the long transaction, and then change the instance class or upgrade the minor version.
3. Do not execute any long transaction when you are changing the instance class or upgrading the minor version.

9.1.4.8 Native Error 1461 Reported by an RDS for MySQL DB Instance

Scenario

The following error information is displayed when there are large amounts of concurrent read and write requests, large amounts of SQL statements, or in data migration scenarios:

```
mysql_stmt_prepare failed! error(1461)Can't create more than
max_prepared_stmt_count statements (current value: 16382)
```

Fault Analysis

The **max_prepared_stmt_count** value ranges from 0 to 1048576. The default value is **16382**. This parameter limits the total number of prepared statements in all sessions on mysqld. The current value exceeds the value range of this parameter.

Solution

Set `max_prepared_stmt_count` to a larger value. The recommended value is 65535.

9.1.4.9 System Inaccessible After Field Addition to an RDS for MySQL Database Table

Description

After a field was added to an RDS for MySQL database table, the system becomes inaccessible.

Solution

The database performance is affected due to the addition of table fields. A possible reason is that indexes are not added to the new table fields. As a result, a large amount of data consumes a large number of CPU resources. You are advised to:

- Add indexes and primary keys.
- Optimize slow SQL statements.

9.1.4.10 Storage Space Occupied by Undo Logs Due to Long Transactions

Scenario

An instance triggered an alarm due to full storage. After a period of time, the alarm was automatically cleared.

Possible Causes

- According to the multiversion concurrency control (MVCC) mechanism, RDS generates undo logs when you update data in database tables and stores the undo logs in your storage. Only after the transactions of all your sessions are committed or rolled back, the undo logs can be removed from the storage.
- If a long transaction exists in one of the sessions, the undo logs generated for the table update sessions cannot be cleared as long as the long transaction is not committed. As a result, the storage usage keeps increasing.

Troubleshooting:

- Run the following statement to check whether any transaction is not committed for a long period of time:
`select t.*,to_seconds(now())-to_seconds(t.trx_started) idle_time from INFORMATION_SCHEMA.INNODB_TRX t \G;`
- View audit logs or slow query logs to check whether a large amount of data is inserted at a time by a large transaction.

Solution

- Kill the long transaction.
- Do not execute long transactions or insert a large amount of data when the storage space is insufficient.
- Scale up storage space in advance.

9.1.5 SQL Issues

9.1.5.1 Error 1366 Reported When Data Containing Emojis Is Updated

Scenario

Error 1366 was reported when data containing emojis was inserted or updated.

```
java.sql.SQLException: Incorrect string value: '\xF0\x9F\x90\xB0\xE5\xA4...' for column 'username' at row 1 ;
uncategorized SQLException for SQL []; SQL state [HY000]; error code [1366];
Incorrect string value: '\xF0\x9F\x90\xB0\xE5\xA4...' for column 'username' at row 1;
```

Possible Causes

The character set of the RDS for MySQL instance is incorrectly configured.

- An emoji is a special character and needs to be stored in a 4-byte character set.
- In this error scenario, the MySQL character set is utf-8, which supports a maximum of three bytes. You need to change the character set to utf8mb4 to support four bytes.

Solution

1. Change the character set for the field that stores emojis to utf8mb4.
If a large number of tables and fields are involved, you are advised to set the encoding format of the tables and databases to utf8mb4. Sample commands:
ALTER DATABASE database_name CHARACTER SET= utf8mb4 COLLATE= utf8mb4_unicode_ci;
ALTERTABLE table_name CONVERTTOCHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
ALTERTABLE table_name MODIFY Field name VARCHAR(128) CHARSET utf8mb4 COLLATE utf8mb4_unicode_ci;
2. If the character set for the field is already utf8mb4, set the character sets of the client and RDS for MySQL server to utf8mb4.

9.1.5.2 Invalid TIMESTAMP Default Value during Table Creation

Scenario

The CREATE TABLE statement failed to be executed, and the error message "ERROR 1067: Invalid default value for 'session_start'" was displayed.

```
CREATE TABLE cluster_membership
(
```



```
...  
session_start TIMESTAMP DEFAULT '1970-01-01 00:00:01',  
...  
);
```

Possible Causes

RDS for MySQL converts the value inserted to the TIMESTAMP column from the current time zone to the UTC time for storage. During query, it returns the value by converting the UTC time to the current time zone.

The time range for the TIMESTAMP column is from '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC. For details, see [The DATE, DATETIME, and TIMESTAMP Types](#).

The **TIMESTAMP** data type is used for values that contain both date and time parts. **TIMESTAMP** has a range of '1970-01-01 00:00:01' UTC to '2038-01-19 03:14:07' UTC.

Run the following command to check the time zone:

show variables like "%zone%";

The UTC+8 time zone is used. Therefore, the valid range for the default value starts from 1970-01-01 08:00:01.

```
mysql> show variables like "%zone%";  
+-----+-----+  
| Variable_name | Value |  
+-----+-----+  
| system_time_zone |      |  
| time_zone       | +08:00 |  
+-----+-----+  
2 rows in set, 1 warning (0.00 sec)
```

Solution

Change the default value of the TIMESTAMP column.

```
session_start TIMESTAMP DEFAULT '1970-01-01 08:00:01',
```

9.1.5.3 AUTO_INCREMENT Not Displayed in the Table Structure

Scenario

When a table was created, **AUTO_INCREMENT** was set to 1. After **show create table** was executed, **AUTO_INCREMENT** was not displayed in the table structure.

A table was created:

```
mysql> CREATE TABLE test (
-> id int(10) NOT NULL AUTO_INCREMENT COMMENT 'id',
-> account_id bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT 'account_id',
-> account_name varchar(128) NOT NULL DEFAULT '' COMMENT 'account_name',
-> identity varchar(64) NOT NULL DEFAULT '' COMMENT 'identity',
-> mobile varchar(32) NOT NULL DEFAULT '' COMMENT 'mobile',
-> score float(6,2) NOT NULL DEFAULT '0.00' COMMENT 'score',
-> utime bigint(20) unsigned NOT NULL DEFAULT '0' COMMENT 'utime',
-> PRIMARY KEY (id),
-> UNIQUE KEY unique_index_account_id (account_id)
-> ) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8mb4;
Query OK, 0 rows affected (0.01 sec)
```

After **show create table xxx** was executed, **AUTO_INCREMENT** was not displayed in the table structure:

Possible Causes

sql_mode was set to **NO_FIELD_OPTIONS**.

Valid values for **sql_mode** are as follows:

- **NO_FIELD_OPTIONS**: Do not print MySQL-specific column options in the output of SHOW CREATE TABLE.
- **NO_KEY_OPTIONS**: Do not print MySQL-specific index options in the output of SHOW CREATE TABLE.
- **NO_TABLE_OPTIONS**: Do not print MySQL-specific table options (such as ENGINE) in the output of SHOW CREATE TABLE.

Solution

Change the value of **sql_mode**.

9.1.5.4 Slow Stored Procedure Execution Due to Inconsistent Collations

Scenario

It took more than 1 minute to process a small amount of data using a stored procedure in an RDS for MySQL instance. However, executing the SQL statement in the stored procedure was much faster.

Possible Causes

The collation of the stored procedure is inconsistent with that of the related table and database. As a result, a large number of characters need to be converted in the query result, and the execution is slow.

Troubleshooting:

Run the following commands to view the definitions of the stored procedure and related table and check whether the collations are the same:

```
SHOW CREATE PROCEDURE xxx;
SHOW CREATE TABLE xxx
```

Example:

```
mysql> SHOW CREATE PROCEDURE testProc \G
***** 1. row *****
Procedure: showstuscore
sql_mode: STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
Create Procedure: xxx
character_set_client: utf8mb4
collation_connection: utf8mb4_general_ci
Database Collation: utf8_general_ci
1 row in set (0.01 sec)
```

The collation of the stored procedure is **utf8mb4_general_ci**, but the collation of the database is **utf8_general_ci** by default. The collations are inconsistent, which may cause performance issues.

Solution

Change the collation of the stored procedure to be the same as that of the related table and database.

9.1.5.5 ERROR [1412] Reported for a DB Instance

Scenario

When an SQL statement was executed on an RDS for MySQL instance, the following error message was displayed:

```
ERROR[1412]:Table definition has changed, please retry transaction``
```

Possible Causes

After a transaction with consistent snapshot was started, another session was executing DDL statements. Procedure for reproducing the problem:

1. Session 1 starts a transaction with consistent snapshot.

```
mysql> start transaction with consistent snapshot;  
Query OK, 0 rows affected (0.00 sec)
```

2. Session 2 executes a DDL statement to modify the table structure.

```
mysql> alter table t_sec_user add test int;  
Query OK, 0 rows affected (0.01 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

3. Session 1 executes a common query statement.

```
mysql> select count(*) from t_sec_user;  
ERROR 1412 (HY000): Table definition has changed, please retry transaction  
mysql>
```

You can also analyze binlogs or audit logs to check whether a DDL statement and transaction with consistent snapshot are executed concurrently on the same table.

Solution

Do not execute a DDL statement and transaction with consistent snapshot concurrently on the same table.

9.1.5.6 Error Message "Too many keys specified" Displayed When a Secondary Index Is Created

Scenario

A secondary index failed to be created, and the error message "Too many keys specified; max 64 keys allowed" was displayed.

Fault Analysis

RDS for MySQL limits the maximum number of secondary indexes in each InnoDB table to 64. If the number of secondary indexes exceeds 64, the error message

"Too many keys specified; max 64 keys allowed" will be displayed. For details, see [InnoDB Limits](#).

[MySQL 8.0 Reference Manual](#) / [The InnoDB Storage Engine](#) / [InnoDB Limits](#)

15.22 InnoDB Limits

This section describes limits for `InnoDB` tables, indexes, tablespaces, and other aspects of the `InnoDB` storage engine.

- A table can contain a maximum of 1017 columns. Virtual generated columns are included in this limit.
- A table can contain a maximum of 64 [secondary indexes](#).
- The index key prefix length limit is 3072 bytes for `InnoDB` tables that use `DYNAMIC` or `COMPRESSED` row format.

Solution

Do not create too many indexes for a single table.

NOTE

Other restrictions on `InnoDB` tables:

1. A table can contain a maximum of 1017 columns (including virtual generated columns).
2. The index key prefix limit is 3072 bytes for `InnoDB` tables that use the `DYNAMIC` or `COMPRESSED` row format.
3. A maximum of 16 columns is permitted for multicolumn indexes. Exceeding the limit returns an error.

9.1.5.7 Failed to Delete a Table with a Foreign Key

Scenario

When an RDS for MySQL table with a foreign key is deleted, the following error message will be displayed, which is irrelevant to user permissions:

```
ERROR 1451 (23000): Cannot delete or update parent row: a foreign key constraint fails .....
```

Possible Causes

A foreign key relationship exists between this table and another table. A link is established between the data in the two tables. To prevent foreign key constraints from being violated, data in the tables cannot be updated or deleted.

You can set `FOREIGN_KEY_CHECKS` to `off` to remove the foreign key relationship. For details, see [FOREIGN KEY Constraints](#).

Solution

Set `FOREIGN_KEY_CHECKS` to `off`.

```
set session foreign_key_checks=off;
drop table table_name;
```

9.1.5.8 DISTINCT and GROUP BY Optimization

Scenario

The execution of the DISTINCT or GROUP BY statement is slow.

Possible Causes

In most cases, DISTINCT can be converted into an equivalent GROUP BY statement. In RDS for MySQL, DISTINCT is mainly used to remove duplicate records from database tables and fetch only the unique records.

The DISTINCT statement groups data first, and then fetches a piece of data from each group and returns the data to the client. There are two scenarios for grouping data:

- All DISTINCT fields are included in the same index. In this scenario, RDS for MySQL directly uses the index to group data, obtains a piece of data from each group, and returns the data.
- Not all DISTINCT fields are included in the index. In this scenario, qualified data is written to a temporary table and grouped in the temporary table. Using temporary tables causes extra overhead, deteriorating the performance.

In conclusion, when using DISTINCT or GROUP BY, set an index that contains all dependent fields. The following is an optimization example:

- No proper index is available. As a result, temporary tables are used.

```
mysql> show create table test;
+-----+-----+
| Table | Create Table |
+-----+-----+
| test  | CREATE TABLE `test` (
  `id` int NOT NULL,
  `c1` int DEFAULT NULL,
  `c2` int DEFAULT NULL,
  `c3` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `c1` (`c1`),
  KEY `c2` (`c2`),
  KEY `c3` (`c3`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> explain select distinct c1,c2,c3 from test;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | test | NULL       | ALL  | NULL         | NULL | NULL    | NULL | 1    | 100.00   | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.00 sec)
```

```
mysql> explain select c1,c2,c3 from test group by c1,c2,c3;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE      | test | NULL       | ALL  | NULL         | NULL | NULL    | NULL | 1    | 100.00   | Using temporary |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.03 sec)
```

- A proper index is available, and temporary tables are not required.

```
mysql> alter table test add key(c1,c2,c3);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show create table test;
+-----+-----+
| Table | Create Table |
+-----+-----+
| test  | CREATE TABLE `test` (
  `id` int NOT NULL,
  `c1` int DEFAULT NULL,
  `c2` int DEFAULT NULL,
  `c3` int DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `c1` (`c1`),
  KEY `c2` (`c2`),
  KEY `c3` (`c3`),
  KEY `c1_2` (`c1`,`c2`,`c3`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+

mysql> explain select c1,c2,c3 from test group by c1,c2,c3;
+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+
| 1 | SIMPLE      | test  | NULL       | index | c1_2          | c1_2 | 15      | NULL | 1 | 100.00 | Using index |
+-----+-----+
1 row in set, 1 warning (0.00 sec)

mysql> explain select distinct c1,c2,c3 from test;
+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+
| 1 | SIMPLE      | test  | NULL       | index | c1_2          | c1_2 | 15      | NULL | 1 | 100.00 | Using index |
+-----+-----+
1 row in set, 1 warning (0.01 sec)
```

Solution

When using DISTINCT or GROUP BY, create an index that contains all dependent fields.

9.1.5.9 Character Set and Collation Settings

Related Variables

By default, **character_set_server** is set to **utf8** and **collation_server** to **utf8_general_ci** for your DB instance. You can change the values on the RDS console.

Configuring Character Sets and Collations for Databases, Tables, and Fields

- If the character set and collation are not explicitly specified for a database during database creation, the values of **character_set_server** and **collation_server** are used for the database. If the character set and collation are explicitly specified, the specified character set and collation are used for the database.
- If the character set and collation are not explicitly specified for a table during table creation, the character set and collation of the database hosting the table are used for the table. If the character set and collation are explicitly specified, the specified character set and collation are used for the table.

- If the character set and collation are not explicitly specified for a field during table creation, the character set and collation of the table hosting the field are used for the field. If the character set and collation are explicitly specified, the specified character set and collation are used for the field.

Example 1: Create a database and table without explicitly specifying the character set and collation.

```
mysql> show variables like 'character_set_server';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| character_set_server | utf8 |
+-----+-----+
1 row in set (0.01 sec)

mysql> show variables like 'collation_server';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| collation_server | utf8_general_ci |
+-----+-----+
1 row in set (0.01 sec)

mysql> create database test_default;
Query OK, 1 row affected (0.26 sec)

mysql> show create database test_default;
+-----+-----+
| Database | Create Database |
+-----+-----+
| test_default | CREATE DATABASE `test_default` /*!40100 DEFAULT CHARACTER SET utf8 */ /*!80016 DEFAULT ENCRYPTION='N' */ |
+-----+-----+
1 row in set (0.00 sec)

mysql> create table test_default.t_default(name varchar(20));
Query OK, 0 rows affected (0.23 sec)

mysql> show create table test_default.t_default;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_default | CREATE TABLE `t_default` (
  `name` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.01 sec)
```

Example 2: Create a database with the character set and collation explicitly specified.

```
mysql> create database test_define CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci;
Query OK, 1 row affected (0.00 sec)

mysql> show create database test_define;
+-----+-----+
| Database | Create Database |
+-----+-----+
| test_define | CREATE DATABASE `test_define` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_0900_ai_ci */ /*!80016 DEFAULT ENCRYPTION='N' */ |
+-----+-----+
1 row in set (0.00 sec)

mysql> create table test_define.t_default(name varchar(20));
Query OK, 0 rows affected (0.08 sec)

mysql> show create table test_define.t_default;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_default | CREATE TABLE `t_default` (
  `name` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.01 sec)
```

Example 3: Create a table with the character set and collation explicitly specified for the table.

```
mysql> create table test_define.t_define(name varchar(20)) CHARACTER SET utf8 COLLATE utf8_bin;
Query OK, 0 rows affected, 2 warnings (0.05 sec)

mysql> show create table test_define.t_define;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_define | CREATE TABLE `t_define` (
  `name` varchar(20) COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin |
+-----+-----+
1 row in set (0.01 sec)
```

Example 4: Create a table with the character set and collation explicitly specified for a field.

```
mysql> create table test_define.t_v_define(name varchar(20) CHARACTER SET gbk COLLATE gbk_bin, str char(32)) CHARACTER SET utf8 COLLATE utf8_bin;
Query OK, 0 rows affected, 2 warnings (0.06 sec)

mysql> show create table test_define.t_v_define;
+-----+-----+
| Table | Create Table |
+-----+-----+
| t_v_define | CREATE TABLE 't_v_define' (
  'name' varchar(20) CHARACTER SET gbk COLLATE gbk_bin DEFAULT NULL,
  'str' char(32) COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin |
+-----+-----+
1 row in set (0.01 sec)
```

9.1.5.10 Server Error Message Displayed When a User Is Created for a DB Instance

Scenario

A user account disappeared from the console, but the account and its password could still be used to connect to the instance.

When a new account with the same name as the disappeared account was created, the following error information was displayed:

```
ERROR 1396 (HY000): Operation CREATE USER failed for xxx
```

Possible Causes

1. The disappeared account has been deleted from the **mysql.user** table and therefore was not displayed on the console.
2. Because the account and its password could still be used to log in to the instance, the account was deleted using **delete from mysql.user**. If you use **delete from mysql.user** to delete an account, you also need to run the **flush privileges** command to delete related data from the memory. Then, the account cannot log in to the instance any more.
3. The reason why a new account with the same name as the disappeared account could not be created is that there was still related data about the disappeared account in the memory.

```
mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
Query OK, 0 rows affected (0.03 sec)

mysql> DELETE FROM mysql.user WHERE Host='localhost'AND User='test1';
Query OK, 1 row affected (0.02 sec)

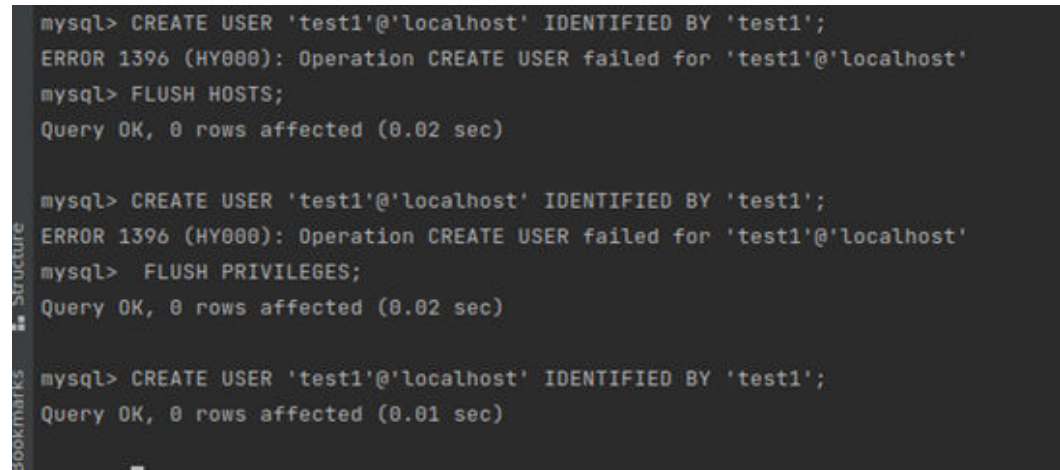
mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test1'@'localhost'
```

The correct way to delete an account is using the **drop user** statement. When running **drop user**, pay attention to that:

- **drop user** can be used to delete one or more users and revoke their permissions.
- **drop user** requires the DELETE permission or the global CREATE USER permission on the RDS for MySQL instance.
- If the host name of the account is not specified in the **drop user** statement, the host name % is used by default.

Troubleshooting example:

After an account is created, the **delete** statement is used to delete the account. When a new account with the same name as the disappeared account is created, error 1396 is reported. After the **flush privileges** command is executed, an account with the same name can be created.



```
mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test1'@'localhost'
mysql> FLUSH HOSTS;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
ERROR 1396 (HY000): Operation CREATE USER failed for 'test1'@'localhost'
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE USER 'test1'@'localhost' IDENTIFIED BY 'test1';
Query OK, 0 rows affected (0.01 sec)
```

Solution

- Method 1 (recommended): During off-peak hours, run the **drop user user_name** command as the administrator to delete the disappeared account and then create an account with the same name.
- Method 2: During off-peak hours, run the **flush privileges** command as the administrator and then create an account with the same name. You are advised to enable SQL Audit to locate which client deletes the user.

9.1.5.11 Slow SQL Queries After a Large Amount of Data Is Deleted from a Large Table

Scenario

After multiple wide columns of data (the length of each record is about 1 GB) are deleted at a time, performing an INSERT, DELETE, UPDATE, or SELECT operation on the same table again takes an extended period of time. After about 20 minutes, the problem is resolved.

Problem Reproduction

1. Assume that the value of **max_allowed_packet** is **1073741824**.
2. Create a table.

```
CREATE TABLE IF NOT EXISTS zstest1
(
  id int PRIMARY KEY not null,
  c_longtext LONGTEXT
);
```
3. Insert data to the table.

```
insert into zstest1 values(1, repeat('a', 1073741800));
insert into zstest1 values(2, repeat('a', 1073741800));
insert into zstest1 values(3, repeat('a', 1073741800));
insert into zstest1 values(4, repeat('a', 1073741800));
insert into zstest1 values(5, repeat('a', 1073741800));
insert into zstest1 values(6, repeat('a', 1073741800));
```

```
insert into zstest1 values(7, repeat('a', 1073741800));
insert into zstest1 values(8, repeat('a', 1073741800));
insert into zstest1 values(9, repeat('a', 1073741800));
insert into zstest1 values(10, repeat('a', 1073741800));
```

4. Delete data from the table.

```
delete from zstest1;
```

5. Execute a query statement.

```
select id from zstest1; //The execution is slow.
```

Possible Causes

After the DELETE operation is performed, the background purge thread clears the records marked with delete mark. Due to the large amount of data to be deleted, the purge thread obtains the SX lock of the index root node where the page is located when traversing and releasing the page. As a result, the SELECT statement cannot obtain the RW lock of the root page and keeps waiting.

Solution

- This phenomenon is normal. After the purge operation is complete, the fault is rectified.
- Scale up the instance specifications to improve the purge efficiency.
- Do not delete a large amount of data at a time. To delete all data from a table, use the **truncate table** statement.

9.1.5.12 Equivalent Comparison Failures with Floating-Point Numbers

Possible Causes

Equivalent comparison of floating-point numbers is a common problem. In computing, floating-point numbers are stored as approximate values instead of exact values. Therefore, unexpected problems may occur during equivalent comparison and mathematical operations.

In RDS for MySQL, FLOAT and DOUBLE are floating-point types. The following figure shows an example for equivalent comparison using floating-point numbers:

```
mysql> create table f(fnum float, dnum double);
Query OK, 0 rows affected (0.26 sec)

mysql> insert into f values(1.1, 1.2);
Query OK, 1 row affected (0.07 sec)

mysql> insert into f values(2.1, 2.2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into f values(2.1, 3.2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into f values(3.1, 3.2);
Query OK, 1 row affected (0.03 sec)

mysql> select * from f;
+-----+-----+
| fnum | dnum |
+-----+-----+
| 1.1 | 1.2 |
| 2.1 | 2.2 |
| 2.1 | 3.2 |
| 3.1 | 3.2 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from f where fnum = 1.1;
Empty set (0.03 sec)

mysql> select * from f where fnum < 2;
+-----+-----+
| fnum | dnum |
+-----+-----+
| 1.1 | 1.2 |
+-----+-----+
1 row in set (0.00 sec)
```

Solution

1. Decide on an acceptable tolerance for differences between the field and the value and then do the comparison against the tolerance value. Example:

```
mysql> select * from f where fnum = 0.01;
Empty set (0.00 sec)

mysql> select * from f where abs(fnum - 1.1) < 0.01;
+-----+-----+
| fnum | dnum |
+-----+-----+
| 1.1 | 1.2 |
+-----+-----+
1 row in set (0.00 sec)
```

2. Use the fixed-point number type (DECIMAL) to replace the floating-point number type. Example:

```
mysql> create table d(d1 DECIMAL(5,2), d2 DECIMAL(5,2));
Query OK, 0 rows affected (0.09 sec)

mysql> insert into d values(1.1, 1.2);
Query OK, 1 row affected (0.02 sec)

mysql> insert into d values(2.1, 2.2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into d values(3.1, 3.2);
Query OK, 1 row affected (0.01 sec)

mysql> select * from d;
+-----+-----+
| d1    | d2    |
+-----+-----+
| 1.10  | 1.20  |
| 2.10  | 2.20  |
| 3.10  | 3.20  |
+-----+-----+
3 rows in set (0.00 sec)

mysql> select * from d where d1 = 1.1;
+-----+-----+
| d1    | d2    |
+-----+-----+
| 1.10  | 1.20  |
+-----+-----+
1 row in set (0.00 sec)
```

9.1.5.13 ERROR[1451] Reported When a Table with Foreign Keys Cannot Be Deleted

Scenario

The **root** user does not have the permissions required to delete or modify tables in the database. Error message is as follows:

**ERROR[1451] -Cannot deleteorupdatea parent row:
aforeignkeyconstraintfails (...)**

Fault Analysis

The FRM file also exists in sys_tables. This table has foreign key relationships with other tables and cannot be deleted directly.

The foreign key association has been set in the RDS for MySQL DB instance. As a result, data cannot be updated or deleted. You can set **foreign_key_checks** to avoid this problem.

Solution

```
set session foreign_key_checks=off;
drop table table_name;
```

To delete the table, set **foreign_key_checks** to **off**.

9.1.5.14 Solution to the Failure of Converting the Field Type

Scenario

The **varchar** field is read using the char data type, and cannot be converted through the char() function.

Fault Analysis

The char() function cannot be used to convert data types.

Solution

The CAST() function and CONVERT() function of RDS for MySQL can be used to obtain values of one type and generate values of another type. The syntax is as follows:

```
CAST(value as type);  
CONVERT(value, type);
```

That is, CAST(xxx AS type) and CONVERT(xxx, type).

NOTE

The data types that can be converted are limited. The supported data types are as follows:

- BINARY: the same as adding a binary prefix to a data string
- CHAR(): characters. You can specify the length of the characters.
- DATE: calendar date
- Time: time of day
- DATETIME: date and time
- DECIMAL: floating point number
- SIGNED: integer
- UNSIGNED: unsigned integer

9.1.5.15 "Row size too large" Reported When an RDS for MySQL Table Failed to Be Created

Scenario

An RDS for MySQL table failed to be created and the following information is displayed:

Row size too large. The maximum row size for the used table type, not counting BLOBs, is 65535. This includes storage overhead, check the manual. You have to change some columns to TEXT or BLOBs

Fault Analysis

The total length of the **varchar** fields exceeds 65535, resulting in a table creation failure.

Solution

1. Reduce the length.

```
CREATE TABLE t1 (a VARCHAR(10000),b VARCHAR(10000),c VARCHAR(10000),d VARCHAR(10000),e VARCHAR(10000),f VARCHAR(10000) ) ENGINE=MyISAM CHARACTER SET latin1;
```
2. Change a column to **TEXT** by referring to the [official document](#).

9.1.5.16 ERROR [1412] Reported by an RDS for MySQL DB Instance

Scenario

The following error information is displayed:

ERROR[1412]:Table definition has changed, please retry transaction

This problem may occur in MySQL-5.7.31.2.

Fault Analysis

Cause 1: A transaction is started using START TRANSACTION WITH CONSISTENT SNAPSHOT.

Scenario 1

```
mysql> start transaction with consistent snapshot;
Query OK, 0 rows affected (0.00 sec)
```

Scenario 2

```
mysql> alter table t_sec_user add test int;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Scenario 3

```
mysql> select count(*) from t_sec_user;
ERROR 1412 (HY000): Table definition has changed, please retry transaction
mysql>
```

Cause 2: DDL operations are performed for binlog files.

```
# at 13793936
#210325 18:44:06 server id 1605802263 end_log_pos 13794065 CRC32 0x4df2db06 Query thread_id=19314688 exec_time=1 error_code=0
use `zzkj`/*!*/;
SET TIMESTAMP=1616669046/*!*/;
SET @session.sql_mode=1346371584/*!*/;
/*!C utf8mb4 *//*!*/;
SET @session.character_set_client=45,@session.collation_connection=45,@session.collation_server=33/*!*/;
ALTER TABLE bas_cabinet_box ADD lock_status tinyint(4)
/*!*/;
```

Solution

If the error is caused by any of the preceding causes, the fault needs to be rectified on the service side.

9.1.6 Connection Issues

9.1.6.1 "Access denied" Displayed During Database Connection

Scenario

A client failed to connect to a database, and the error message "Error 1045: Access denied for user xxx" was displayed.

Handling Methods

1. An incorrect host is connected.

Cause: An incorrect database host is connected, and the user or client IP address does not have the access permission.

Solution: Ensure that the host name of the database to be connected is correctly specified.

2. The user does not exist.

Cause: The user account used for connecting to the database does not exist.

Solution:

- Log in to the database as an administrator and run the following command to check whether the target user exists:

```
SELECT User FROM mysql.user WHERE User='xxx';
```

- If the user does not exist, create the user.

```
CREATE USER 'xxx'@'xxxxxx' IDENTIFIED BY 'xxxx';
```

3. The client IP address does not have the access permission.

Cause: The user used by the client exists, but the client IP address is not allowed to access the database.

Solution:

- Log in to the database as an administrator and run the following command to check which client IP addresses are allowed to connect to the database for the target user:

```
SELECT Host, User FROM mysql.user WHERE User='xxx';
```

- If the client IP address is not within the allowed network segment, assign the access permission to the client IP address. For example, run the following command to grant the **test** user the permission to access the **192.168.0** network segment:

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'192.168.0.%' IDENTIFIED BY 'password' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

4. The password is incorrect.

Cause: The password of the user is incorrect.

Solution:

- Check whether the target password is correct. Because the password is used for identity authentication, the user password cannot be read from RDS for MySQL in plain text. However, you can compare the hash string with the **PASSWORD** function value of the target password to check whether the target password is correct. The following is an example of SQL statements:

```
mysql> SELECT Host, User, authentication_string, PASSWORD('12345') FROM mysql.user
WHERE User='test';
```

Host	User	authentication_string	PASSWORD('12345')

```
| % | test | *6A23DC5E7446019DC9C1778554ED87BE6BA61041 |  
*00A51F3F48415C7D4E8908980D443C29C69B60C9 |
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+  
2 rows in set, 1 warning (0.00 sec)
```

The preceding example shows that the hash value of **PASSWORD('12345')** does not match the **authentication_string** field, indicating that the target password **12345** is incorrect.

- To reset the user password, run the following SQL statement:
set password for 'test'@'%' = 'new_password';

5. The password contains special characters and is escaped by Bash.

Cause: In the default Bash environment of Linux, when the CLI is used to connect to a database, special characters in the password will be escaped by the environment. As a result, the password becomes invalid.

For example, in the Bash environment, the password of user **test** is **test\$123**. When you run the **mysql -hxxx -u test -ptest\$123** command to connect to a database, the error message "ERROR 1045 (28000): Access denied" will be displayed.

Solution: Enclose the password in single quotation marks to prevent Bash from interpreting special characters.

```
mysql -hxxx -u test -p'test$123'
```

6. **REQUIRE SSL** is configured for the user, but the client uses a non-SSL connection.

Troubleshooting:

- Run the **show create user 'xxx'** command to check whether the user must use the SSL connection. If the **REQUIRE SSL** attribute is displayed, the user must use the SSL connection.
- Check whether statements similar to the following have been used to grant permissions to the user:
GRANT ALL PRIVILEGES ON . TO 'ssluser'@'localhost' IDENTIFIED BY 'zdh1234' REQUIRE SSL;
- Check the **ssl_type** value of the target user. If the value is not empty, the user must use SSL.
SELECT User, Host, ssl_type FROM mysql.user WHERE User='xxx';

Solution:

- Connect the client to the database in SSL mode. For details, see [Connecting to a DB Instance Using a MySQL Client](#).
- Run the **ALTER USER 'test'@'xxxxx' REQUIRE NONE;** command to remove the SSL permission from the user.

9.1.6.2 Error Message "connection established slowly"

Scenario

During peak hours, the connection between a client and an RDS for MySQL instance often times out. As a result, it takes more than 10 seconds to log in to the instance.

Possible Causes

1. [Check error logs of the instance](#) to check whether the information "connection xxx is established slowly" is displayed. Example:


```

2021-01-28 09:29:09 94016 [Warning] Access denied for user 'XXXXXXXXXXXXXXXXXXXX@165' (using password: YES)
2021-01-28 09:29:09 94016 [Warning] Aborted connection 27103 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:15 94016 [Note] This connection(rdsAdmin@localhost) is established slowly(Connected=yes)
2021-01-28 09:29:16 94016 [Warning] Access denied for user 'XXXXXXXXXXXXXXXXXXXX@165' (using password: YES)
2021-01-28 09:29:18 94016 [Warning] Aborted connection 27188 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:27 94016 [Warning] Aborted connection 27192 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:28 94016 [Warning] Aborted connection 27193 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:38 94016 [Warning] Aborted connection 27195 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:38 94016 [Warning] Aborted connection 27194 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:41 94016 [Warning] Aborted connection 27196 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:41 94016 [Warning] Aborted connection 27197 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:47 94016 [Note] This connection(XXXXXXXXXXXX@165) is established slowly(Connected=yes)
2021-01-28 09:29:49 94016 [Warning] Aborted connection 27199 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:51 94016 [Warning] Aborted connection 27201 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:29:54 94016 [Warning] Aborted connection 27202 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:30:00 94016 [Warning] Aborted connection 27204 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:30:05 94016 [Warning] Aborted connection 27206 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:30:09 94016 [Warning] Aborted connection 27207 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:30:13 94016 [Warning] Aborted connection 27208 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:30:16 94016 [Warning] Aborted connection 27210 to db: 'unconnected' user: 'rdsmetric' host: '
2021-01-28 09:30:17 94016 [Note] This connection(null@165) is established slowly(Connected=no).The time of total and launch is 3165018 and 3164825 microseconds respectively.
  
```

If yes, some connections have timed out and have not been processed by the RDS for MySQL instance yet. When the connection between a client and the instance exceeds the specified timeout duration, an error is reported.

2. Check the thread pool configuration (enabled by default) on the console.

In the preceding figure, **threadpool_size** is set to **1**, **threadpool_stall_limit** is set to **500 ms**, and **threadpool_oversubscribe** is set to **3**. The wait time for the thread pool to process connections is mainly related to the three parameters.

- When all threads in the thread pool are busy, the scheduling thread in the thread pool creates a new thread every 500 ms (**threadpool_stall_limit**). In this case, each thread group can process a new connection every 500 ms on average. If the queue is too long, the client may time out.
 - If all threads are busy, the number of working threads reaches the total number of threads in the thread pool. When a large number of connections are established, the total number of threads is calculated as $\text{threadpool\_size} \times (\text{threadpool\_oversubscribe} + 1)$.
3. Therefore, if the value of **threadpool_size** is improper or too small, new connections will be established slowly during peak hours.

Solution

If there are a large number of new connections, increase the values of **threadpool_size** and **threadpool_oversubscribe** to increase the total number of threads and decrease the value of **threadpool_stall_limit** to reduce the wait time for new connections. This reduces the overhead of repeated thread creation and destruction, and limits the number of running threads to protect the system against avalanche.

In normal cases, the thread pool is used when there are a large number of short connections. If persistent connections are used and there are a few connections (for example, the client uses a connection pool), the thread pool is not so helpful. In this case, adjust the values of **threadpool_size** and **threadpool_oversubscribe** to increase the total number of threads, or close the thread pool.

9.1.6.3 SSL Connection Failed Due to Inconsistent TLS Versions

Scenario

A client failed to connect to an RDS for MySQL instance using SSL, but could connect to a self-built MySQL database using SSL.

Possible Causes

Troubleshooting:

1. [View error logs of the instance](#). The following error information was displayed:

```
2021-07-09T10:30:58.476586+08:00 212539 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:30:58.476647+08:00 212539 [Note] Bad
handshake2021-07-09T10:32:43.535738+08:00 212631 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:32:43.535787+08:00 212631 [Note] Bad
handshake2021-07-09T10:50:03.401100+08:00 213499 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:50:03.401161+08:00 213499 [Note] Bad
handshake2021-07-09T10:53:44.458404+08:00 213688 [Warning] SSL errno: 337678594, SSL errmsg:
error:14209102:SSL routines:tls_early_post_process_client_hello:unsupported
protocol2021-07-09T10:53:44.458475+08:00 213688 [Note] Bad handshake
```

2. According to **unsupported protocol** in the error information, the problem may be related to the TLS version. Run the following command to check the TLS versions of the RDS for MySQL instance and self-built MySQL database:

show variables like '%tls_version%';

It was found that the RDS for MySQL instance used TLS v1.2 and the self-built MySQL database used TLS v1.1. The TLS version of the client was the same as that of the self-built MySQL database. Therefore, the self-built MySQL database is successfully connected, but the RDS for MySQL instance failed to be connected.

Solution

Upgrade the TLS version of the client to TLS v1.2.

If the official JDBC driver **mysql-connector/J** is used, see [Connecting Securely Using SSL](#) for the configuration method.

TLS versions: The allowable versions of TLS protocol can be restricted using the connection properties `tlsVersions` and, for X DevAPI connections and for release 8.0.19 and later, `xdevapi.tls-versions` (when `xdevapi.tls-versions` is not specified, it takes up the value of `tlsVersions`). If no such restrictions have been specified, Connector/J attempts to connect to the server with the TLSv1.2 and TLSv1.3.

9.1.6.4 RDS for MySQL Client Automatically Disconnected from a DB Instance

Description

The RDS for MySQL client was automatically disconnected from the DB instance. The following error information is displayed: ERROR 2013: Lost connection to MySQL server during query.

Solution

ERROR 2013 is usually caused by incorrect configuration.

- **wait_timeout**: indicates the number of seconds the server waits for activity on a non-interactive connection before closing it.
- **interactive_timeout**: indicates the number of seconds the server waits for activity on an interactive connection before closing it.

Step 1 Check whether the DB instance is available.

If the DB instance is available, check for other possible causes.

Step 2 View error logs.

Step 3 Use the RDS for MySQL command-line client to connect to the target database. Run **status** to check whether the DB instance has been rebooted frequently.

```
mysql> status
-----
mysql Ver 14.14 Distrib 5.6.34, for Linux (x86_64) using EditLine wrapper

Connection id:          16288
Current database:
Current user:           root@192.168.0.5
SSL:                    Not in use
Current pager:          stdout
Using outfile:          ''
Using delimiter:        ;
Server version:         5.6.34-log MySQL Community Server (GPL)
Protocol version:       10
Connection:             192.168.0.24 via TCP/IP
Server characterset:    utf8
Db characterset:        utf8
Client characterset:    utf8
Conn. characterset:     utf8
TCP port:               8635
Uptime:                 5 hours 5 min 34 sec

Threads: 2  Questions: 62118  Slow queries: 0  Opens: 70  Flush tables: 2  Open tables: 0  Queries per second avg: 3.388
-----
```

Uptime indicates the running time of the DB instance. The command output shows that the database has not been restarted frequently. Therefore, the client disconnection is not caused by a database restart.

Step 4 Check parameters. If the values of **wait_timeout** and **interactive_timeout** are too small, the RDS for MySQL client automatically stops connections that timed out.

Step 5 You can change the values of **wait_timeout** and **interactive_timeout** based on service requirements without the need of rebooting the DB instance.

Step 6 After about 10 minutes, run the **show databases** command to check whether the connection is normal.

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)

mysql>
```

If information similar to the preceding figure is returned, the connection is normal.

----End

9.1.6.5 RDS for MySQL DB Instance Inaccessible

Scenario

When the MySQL client attempted to connect to a database, the following error information is displayed:

- Fault 1
ERROR 1045 (28000): Access denied for user 'root'@'192.168.0.30' (using password:YES)
- Fault 2
ERROR 1226 (42000): User 'test' has exceeded the 'max_user_connections' resource (current value:10)
- Fault 3
ERROR 1129 (HY000): Host '192.168.0.111' is blocked because of many connection errors; unblock with 'mysqladmin flush-hosts'

Fault 1

Step 1 Check whether the password of user **root** is correct.

ERROR 1045 (28000) is usually caused by an incorrect password so you need to check the password first.

```
select password('Test1i@123');  
select host,user,Password from mysql.user where user='test1';
```

```
mysql> select host,user,Password from mysql.user where user='test1';  
+-----+-----+-----+  
| host      | user  | Password |  
+-----+-----+-----+  
| 192.168.0.74 | test1 | *094C0901889F4D899028A83C061EF44272590E22 |  
+-----+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> select password('test1i@123');  
+-----+  
| password('test1i@123') |  
+-----+  
| *094C0901889F4D899028A83C061EF44272590E22 |  
+-----+  
1 row in set (0.01 sec)
```

An incorrect password will cause a login failure as shown in the following figure.

```
[root@ecs-lwt-0921 ~]# mysql -utest1 -ptest1i@321 -P 8635 -h192.168.0.73  
Warning: Using a password on the command line interface can be insecure.  
ERROR 1045 (28000): Access denied for user 'test1'@'192.168.0.74' (using password: YES)
```

Step 2 Check whether the ECS or device has the permission to connect to the DB instance.

```
select user,host from mysql.user where user='username';
```

```
mysql> select user,host from mysql.user where user='test1';  
+-----+-----+  
| user  | host      |  
+-----+-----+  
| test1 | 192.168.0.74 |  
+-----+-----+  
1 row in set (0.00 sec)
```

If you want to log in to the database from another ECS or device, log in to the database as user **root** and grant permission to the user.

For example, if the IP address of the ECS or device is 192.168.0.76, run the following command:

GRANT all privileges ON test.* TO 'test1'@'192.168.0.76' identified by 'Test1i@123';

flush privileges;

```
[root@ecs-lwt-0921 ~]# mysql -uroot -ptest1i@123 -P 8635 -h192.168.0.73
Warning: Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6961
Server version: 5.6.35-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, 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> GRANT all privileges ON test.* TO 'test1'@'192.168.0.76' identified by 'test1i@123';
Query OK, 0 rows affected (0.01 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

Step 3 Check whether the RDS for MySQL client can ping the DB instance VIP and use telnet to check the connection to the port.

If the ping and telnet operations are successful, ERROR 1045 (28000) is not caused by the DB instance VIP or port.

Step 4 Check the security group and rules of the DB instance.

Step 5 Query the user table and check user information.

```
mysql> select user,host,max_connections,max_user_connections,password_expired from mysql.user order by user;
+-----+-----+-----+-----+-----+
| user      | host      | max_connections | max_user_connections | password_expired |
+-----+-----+-----+-----+-----+
| mysql.sys | localhost | 0               | 0                     | N                |
| rdsAdmin  | localhost | 100000          | 100000                | N                |
| rdsBackup | localhost | 100000          | 100000                | N                |
| rdsMetric | localhost | 100000          | 100000                | N                |
| rdsRepl   | 172.16.% | 100000          | 100000                | N                |
| root      | %         | 0               | 0                     | N                |
| root      | 192.168.% | 0               | 0                     | N                |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

The command output indicates that two **root** users exist.

When the client is in the network segment 192.168.***.***, RDS for MySQL authenticates the user **root@'192.168.%'**. However, the password of user **root@'%'** is used to log in to the database, which causes a login failure. Therefore, ERROR 1045 (28000) is caused by an incorrect password.

NOTE

User **root@'%'** is used for setting a password when you create a DB instance on the RDS console.

----End

Fault 2

- Step 1** Check whether the **max_user_connections** option is set when you create an RDS for MySQL user, which limits the number of connections.

```
select user,host,max_user_connections from mysql.user where user='test';
```

```
mysql> select user,host,max_user_connections from mysql.user where user='test';
+-----+-----+-----+
| user | host      | max_user_connections |
+-----+-----+-----+
| test | 192.168.0.100 | 10 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

The command output indicates that the connection fails because the **max_user_connections** option is set.

- Step 2** Increase the maximum number of connections for the user.

```
alter user test@'192.168.0.100' with max_user_connections 15
```

- Step 3** Query the modification result and check whether the database can be connected to.

```
mysql> alter user test@'192.168.0.100' with max_user_connections 15;
Query OK, 0 rows affected (0.01 sec)

mysql> select user,host,max_user_connections from mysql.user where user='test';
+-----+-----+-----+
| user | host      | max_user_connections |
+-----+-----+-----+
| test | 192.168.0.100 | 15 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

----End

Fault 3

- Step 1** Check whether the number of failed connection attempts (not caused by incorrect passwords) of the RDS for MySQL client exceeds the value of **max_connection_errors**.

- Step 2** Log in to the MySQL database as user **root** and run **flush hosts**.

Alternatively, run the following command:

```
mysqladmin flush-hosts -u<user> -p<password> -h<ip> -P<port>
```

- Step 3** Use the MySQL client to connect to the database again.

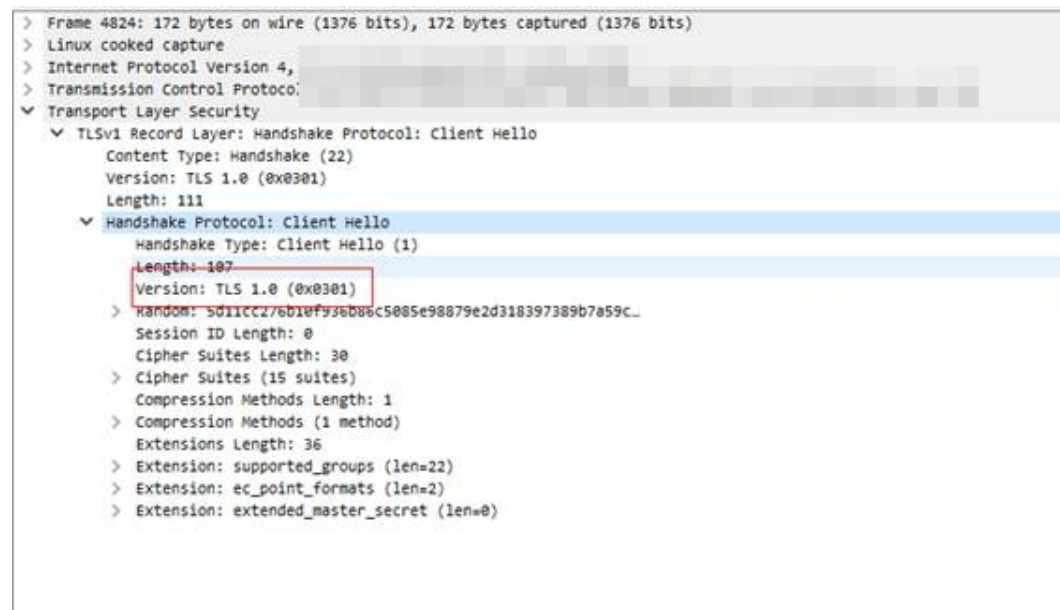
----End

Caused by: javax.net.ssl.SSLException: Received fatal alert: protocol_version

MySQL-server connection failed after RDS for MySQL 5.7.23 is upgraded to 5.7.25. [Figure 9-2](#) shows the captured packet.

The TLS version sent from the client to the server during the TLS handshake is 1.0. A total number of 15 supported cipher suites are provided.

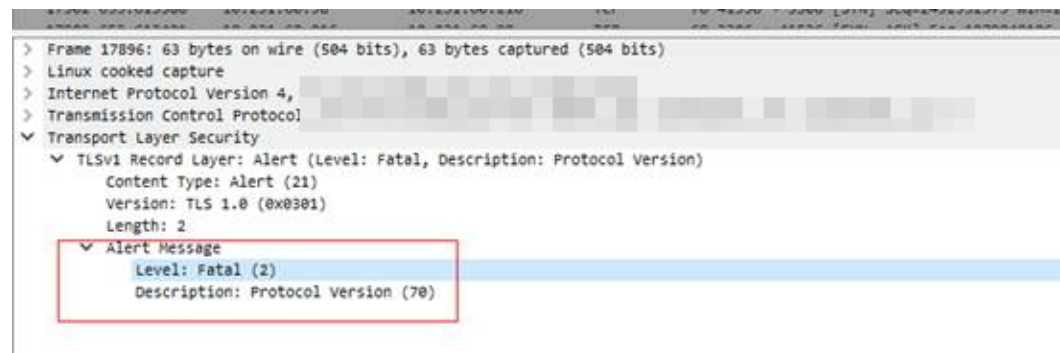
Figure 9-2 Packets captured when connection failed



Fault Analysis

As shown in the MySQL-server response in [Figure 9-3](#), the server rejects the client connection because OpenSSL has been upgraded to 1.1.1a on MySQL 5.7.25, resulting in the rejection of the insecure TLS version and password suite.

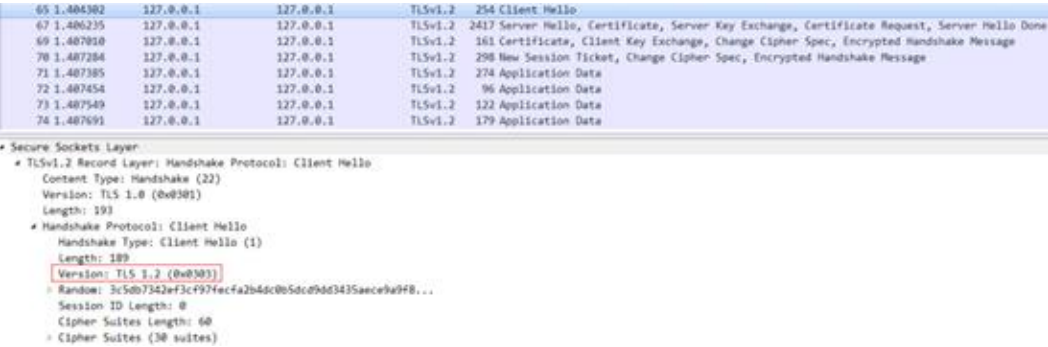
Figure 9-3 MySQL-server response



Solution

Upgrade your JDK client to **JDK 8 or a later version**. By default, TLS 1.2 is supported and 30 cipher suites are provided. [Figure 9-4](#) shows a normal captured packet.

Figure 9-4 Packets captured when connection is normal



9.1.6.8 Connection Exit Due to Improper Timeout Parameter Settings

Scenario

A database connection exit often occurs. As a result, subsequent statements fail to be executed.

Possible Causes

When a connector or API is used to connect to a database, the client has some default parameter settings. The settings of some important parameters, such as **socketTimeout** and **connectTimeout**, determine the client connection timeout duration. If the wait time of a connection exceeds the value of one of these parameters, the connection will be interrupted.

Solution

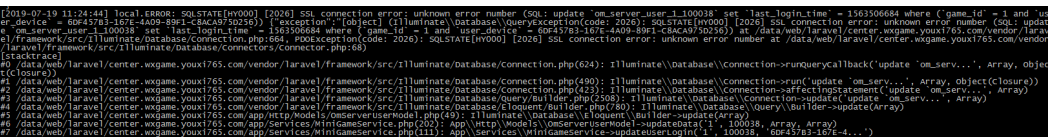
- Change the default values of parameters such as **socketTimeout** and **connectTimeout** to appropriate values.
- Pay attention to the reconnection function in the program.
- Using connection pools is recommended.

9.1.6.9 Database Connection Through Code (php/java/python) Failed After SSL Is Enabled

Scenario

After SSL is enabled, an error message is displayed when a database is connected through commands.

Figure 9-5 Connection failure



Fault Analysis

If SSL is enabled, only an SSL connection can be used for login.

Solution

Enable SSL and use an SSL connection to connect to a database. For details, see [Connecting to a DB Instance Through a Private Network](#).

9.1.7 Other Issues

9.1.7.1 Auto-increment Field Values

RDS for MySQL uses the following methods to assign values to an auto-increment field:

```
# Table structure
CREATE TABLE animals (
  id MEDIUMINT NOT NULL AUTO_INCREMENT,
  name CHAR(30) NOT NULL,
  PRIMARY KEY (id)
);
```

1. If no value is specified for the auto-increment field, RDS for MySQL automatically enters the value of **AUTO_INCREMENT** to the field.

```
mysql> INSERT INTO animals (name) VALUES ('fish'),('cat'),('penguin'),('lax'),('whale'),('ostrich');
```

```
Query OK, 6 rows affected (0.01 sec)
```

```
Records: 6 Duplicates: 0 Warnings: 0
```

```
mysql> select * from animals;
```

```
+-----+-----+
```

```
| id | name |
```

```
+-----+-----+
```

```
| 1 | fish |
```

```
| 2 | cat |
```

```
| 3 | penguin |
```

```
| 4 | lax |
```

```
| 5 | whale |
```

```
| 6 | ostrich |
```

```
+-----+-----+
```

```
6 rows in set (0.00 sec)
```

```
mysql> show create table animals;
```

```
+-----+-----+
```

```
| Table | Create Table |
```

```
+-----+-----+
```

```
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT, `name`  
char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT  
CHARSET=utf8 |
```

```
+-----+-----+
```

2. If **0** or **NULL** is specified for the auto-increment field, RDS for MySQL automatically enters the value of **AUTO_INCREMENT** to the field.

```
mysql> INSERT INTO animals (id,name) VALUES(0,'groundhog');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO animals (id,name) VALUES(NULL,'squirrel');
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from animals;
```

```
+-----+-----+
```

```
| id | name |
```

```
+-----+-----+
```

```
| 1 | fish |
```

```
| 2 | cat |
```

```
| 3 | penguin |
```

```
| 4 | lax |
```

```
| 5 | whale |
```

```
| 6 | ostrich |
| 7 | groundhog |
| 8 | squirrel |
+----+-----+8
rows in set (0.00 sec)
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8 |
+-----+-----+
```

3. If the value X that is greater than the value of **AUTO_INCREMENT** is specified for the auto-increment field, RDS for MySQL inserts X to the field and changes **AUTO_INCREMENT** to $X + 1$.

```
mysql> INSERT INTO animals (id,name) VALUES(100,'rabbit');
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+----+-----+
| id | name |
+----+-----+
| 1 | fish |
| 2 | cat |
| 3 | penguin |
| 4 | tax |
| 5 | whale |
| 6 | ostrich |
| 7 | groundhog |
| 8 | squirrel |
| 100 | rabbit |
+----+-----+
9 rows in set (0.00 sec)
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |
+-----+-----+
```

4. If a value less than the value of **AUTO_INCREMENT** is specified for the auto-increment field, RDS for MySQL enters the value to the field and **AUTO_INCREMENT** remains unchanged.

```
mysql> INSERT INTO animals (id,name) VALUES(50,'middle');
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+----+-----+
| id | name |
+----+-----+
| 1 | fish |
| 2 | cat |
| 3 | penguin |
| 4 | tax |
| 5 | whale |
| 6 | ostrich |
| 7 | groundhog |
| 8 | squirrel |
| 50 | middle |
| 100 | rabbit |
+----+-----+
10 rows in set (0.00 sec)
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT, `name`
```

```
char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT
CHARSET=utf8 |
+-----+-----+
```

5. If a negative value is specified for the auto-increment field, RDS for MySQL enters the value to the field and **AUTO_INCREMENT** remains unchanged.

```
mysql> INSERT INTO animals (id,name) VALUES(-50,'-middle');
```

```
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from animals;
```

```
+-----+-----+
| id | name |
+-----+-----+
```

```
| -50 | -middle |
| 1 | fish |
| 2 | cat |
| 3 | penguin |
| 4 | tax |
| 5 | whale |
| 6 | ostrich |
| 7 | groundhog |
| 8 | squirrel |
| 50 | middle |
| 100 | rabbit |
+-----+-----+
```

```
11 rows in set (0.00 sec)
```

```
mysql> show create table animals;
```

```
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` ( `id` mediumint NOT NULL AUTO_INCREMENT,
`name` char(30) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT
CHARSET=utf8 |
+-----+-----+
```

9.1.7.2 Starting Value and Increment of AUTO_INCREMENT

The starting value and increment of **AUTO_INCREMENT** are determined by the **auto_increment_offset** and **auto_increment_increment** parameters.

- **auto_increment_offset** determines the starting point for the **AUTO_INCREMENT** column value.
- **auto_increment_increment** controls the interval between successive column values.
- When the value of **auto_increment_offset** is greater than that of **auto_increment_increment**, the value of **auto_increment_offset** is ignored.
- When the value of **auto_increment_offset** is less than or equal to that of **auto_increment_increment**, the value of **AUTO_INCREMENT** is calculated as **auto_increment_offset + N x auto_increment_increment** (*N* indicates the number of inserted data records).

In RDS for MySQL, the values of **auto_increment_increment** and **auto_increment_offset** are both **1** by default. You can change them on the RDS console. For details, see [Modifying RDS for MySQL Instance Parameters](#).

Example:

- If both **auto_increment_offset** and **auto_increment_increment** are set to **1**, the starting value is 1 and the increment is 1.

```
mysql> show variables like 'auto_inc%';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 1 |
+-----+-----+
```

```
| auto_increment_offset | 1 |
+-----+
mysql> create table auto_test1(id int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`));
Query OK, 0 rows affected (0.09 sec)
mysql> show create table auto_test1;
+-----+
| Table | Create Table |
+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`) ) ENGINE=InnoDB DEFAULT
CHARSET=utf8 |
+-----+
mysql> insert into auto_test1 values(0), (0), (0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
3 rows in set (0.01 sec)
mysql> show create table auto_test1;
+-----+
| Table | Create Table |
+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=4
DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.00 sec)
```

- If **auto_increment_increment** is set to 2, the increment is 2.

```
mysql> set session auto_increment_offset=2;
Query OK, 0 rows affected (0.02 sec)
mysql> show variables like 'auto_inc%';
+-----+
| Variable_name | Value |
+-----+
| auto_increment_increment | 2 |
| auto_increment_offset | 1 |
+-----+
mysql> insert into auto_test1 values(0), (0), (0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 4 |
| 6 |
| 8 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+
| Table | Create Table |
+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
`id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.01 sec)
```

- If **auto_increment_offset** is set to **10** and **auto_increment_increment** is set to **2**, the starting value is 2 (because the value of **auto_increment_offset** is greater than that of **auto_increment_increment**) and the increment is 2.

```
mysql> set session auto_increment_offset=10;
mysql> set session auto_increment_increment=2;
mysql> show variables like 'auto_inc%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 2 |
| auto_increment_offset | 10 |
+-----+-----+
mysql> create table auto_test2(id int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`)); Query OK,
0 rows affected (0.08 sec)
mysql> show create table auto_test2;
+-----+
-----+
| Table | Create Table |
+-----+
+-----+
| auto_test2 | CREATE TABLE `auto_test2` ( `id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY
(`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+
+-----+
1 row in set (0.01 sec)
mysql> insert into auto_test2 values(0), (0), (0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test2;
+----+
| id |
+----+
| 2 |
| 4 |
| 6 |
+----+
3 rows in set (0.01 sec)
mysql> show create table auto_test2;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test2 | CREATE TABLE `auto_test2` (
`id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=8
DEFAULT CHARSET=utf8 |
+-----+-----+
```

- If **auto_increment_offset** is set to **5** and **auto_increment_increment** is set to **10**, the starting value is 5 and the increment is 10.

```
mysql> set session auto_increment_offset=5; mysql> set session auto_increment_increment=10;
mysql> show variables like 'auto_inc%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| auto_increment_increment | 10 |
| auto_increment_offset | 5 |
+-----+-----+
mysql> create table auto_test3(id int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`));
mysql> show create table auto_test3;
+-----+
-----+
| Table | Create Table |
+-----+
+-----+
+-----+
| auto_test3 | CREATE TABLE `auto_test3` ( `id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY
```

```
(`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+
+-----+
+-----+
mysql> insert into auto_test3 values(0), (0), (0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test3;
+----+
| id |
+----+
| 5 |
| 15 |
| 25 |
+----+
mysql> show create table auto_test3;
+-----+
| Table | Create Table |
+-----+
| auto_test3 | CREATE TABLE `auto_test3` (
  `id` int NOT NULL AUTO_INCREMENT, PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=35 DEFAULT CHARSET=utf8 |
+-----+
+-----+
```

9.1.7.3 Auto-Increment Field Value Jump

If the values of the auto-increment field are discontinuous, the possible causes are as follows:

- If the increment is not 1, the values of the auto-increment field are discontinuous.

```
mysql> show variables like 'auto_inc%';
+-----+
| Variable_name | Value |
+-----+
| auto_increment_increment | 2 |
| auto_increment_offset | 1 |
+-----+
mysql> select * from auto_test1;
+----+
| id |
+----+
| 2 |
| 4 |
| 6 |
| 8 |
+----+
```

- The value of **AUTO_INCREMENT** is changed.

```
mysql> select * from animals;
+----+
| id | name |
+----+
| 1 | fish |
| 2 | cat |
| 3 | penguin |
+----+
mysql> show create table animals;
+-----+
| Table | Create Table |
+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT,
  `name` char(30) NOT NULL,
  PRIMARY KEY (`id`) )
ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 |
+-----+
mysql> alter table animals AUTO_INCREMENT=100;
```

```
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+
| Table | Create Table |
+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT,
  `name` char(30) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=100 DEFAULT CHARSET=utf8 |
+-----+
mysql> INSERT INTO animals (id,name) VALUES(0,'rabbit');
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+-----+
| id | name |
+-----+
| 1 | fish |
| 2 | cat |
| 3 | penguin |
| 100 | rabbit |
+-----+
9 rows in set (0.00 sec)
```

- The value of the auto-increment field is specified when data is inserted.

```
mysql> select * from animals;
+-----+
| id | name |
+-----+
| 1 | fish |
| 2 | cat |
| 3 | penguin |
+-----+
mysql> INSERT INTO animals (id,name) VALUES(100,'rabbit');
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+-----+
| id | name |
+-----+
| 1 | fish |
| 2 | cat |
| 3 | penguin |
| 100 | rabbit |
+-----+
9 rows in set (0.00 sec)
```

- If a transaction is not committed or is rolled back, the value of **AUTO_INCREMENT** increases but does not decrease after the transaction is rolled back. When data is inserted again, the value of the auto-increment field jumps.

```
mysql> show create table auto_test1;
+-----+
| Table | Create Table |
+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 |
+-----+
1 row in set (0.00 sec)
mysql> select * from auto_test1;
+-----+
| id |
+-----+
| 1 |
| 2 |
| 3 |
+-----+
```



```
mysql> begin;
Query OK, 0 rows affected (0.02 sec)
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test1 |
CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> rollback;
Query OK, 0 rows affected (0.05 sec)
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
3 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 7 |
| 8 |
| 9 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
```

```
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

- After data is inserted, the value of **AUTO_INCREMENT** changes. But when the corresponding data row is deleted, the value of **AUTO_INCREMENT** does not decrease. When data is inserted again, the value of the auto-increment field jumps.

```
mysql> show create table auto_test1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Create Table |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Create Table |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
mysql> delete from auto_test1 where id>3;
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
+----+
3 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Table | Create Table |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
```

```
+-----+-----+
mysql> insert into auto_test1 values (0),(0),(0);
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test1;
+----+
| id |
+----+
| 1 |
| 2 |
| 3 |
| 7 |
| 8 |
| 9 |
+----+
6 rows in set (0.00 sec)
mysql> show create table auto_test1;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test1 | CREATE TABLE `auto_test1` (
  `id` int NOT NULL AUTO_INCREMENT,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8 |
+-----+-----+
```

- If data insertion fails due to some reasons (for example, unique key conflict), the value of **AUTO_INCREMENT** may jump.

```
mysql> create table auto_test7(`id` int NOT NULL AUTO_INCREMENT, cred_id int UNIQUE, PRIMARY
KEY (`id`));
Query OK, 0 rows affected (0.64 sec)
mysql> insert into auto_test7 values(null, 1);
Query OK, 1 row affected (0.03 sec)
mysql> show create table auto_test7;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test7 | CREATE TABLE `auto_test7` ( `id` int NOT NULL AUTO_INCREMENT, `cred_id` int
DEFAULT NULL, PRIMARY KEY (`id`), UNIQUE KEY `cred_id` (`cred_id`)) ENGINE=InnoDB
AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> insert into auto_test7 values(null, 1);
ERROR 1062 (23000): Duplicate entry '1' for key 'auto_test7.cred_id'
mysql> show create table auto_test7;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test7 | CREATE TABLE `auto_test7` ( `id` int NOT NULL AUTO_INCREMENT, `cred_id` int
DEFAULT NULL, PRIMARY KEY (`id`), UNIQUE KEY `cred_id` (`cred_id`)) ENGINE=InnoDB
AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 |
+-----+-----+
```

- When data is inserted in batches (such as **insert...select** and **load file**), the auto-increment key is requested in batches. Two to the power of n sequence numbers are requested in each batch. If the sequence numbers are not used up, the sequence numbers will not be returned. As a result, the value of **AUTO_INCREMENT** may jump.

```
mysql> create table auto_test5_tmp(id tinyint not null AUTO_INCREMENT, name varchar(8), PRIMARY
KEY (`id`));
Query OK, 0 rows affected (0.08 sec)
mysql> select * from auto_test5;
+-----+
| id | name |
+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | X |
```

```
| 5 | Y |
| 6 | Z |
| 8 | A |
| 9 | B |
| 10 | C |
| 11 | X |
| 12 | Y |
| 13 | Z |
+----+-----+
12 rows in set (0.00 sec)
mysql> insert into auto_test5_tmp select 0,name from auto_test5;
Query OK, 12 rows affected (0.01 sec)
Records: 12 Duplicates: 0 Warnings: 0
mysql> select * from auto_test5_tmp;
+----+-----+
| id | name |
+----+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | X |
| 5 | Y |
| 6 | Z |
| 7 | A |
| 8 | B |
| 9 | C |
| 10 | X |
| 11 | Y |
| 12 | Z |
+----+-----+
12 rows in set (0.00 sec)
mysql> show create table auto_test5_tmp;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test5_tmp | CREATE TABLE `auto_test5_tmp` ( `id` tinyint NOT NULL AUTO_INCREMENT,
`name` varchar(8) DEFAULT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=16
DEFAULT CHARSET=utf8 |
+-----+-----+
```

9.1.7.4 Changing the AUTO_INCREMENT Value of a Table

The methods are as follows:

- If the value of **AUTO_INCREMENT** is greater than the maximum value of the auto-increment column in the table, **AUTO_INCREMENT** can be changed to a larger value within the value range.

```
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
`id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> select * from animals;
+----+-----+
| id | name |
+----+-----+
| -50 | -middle |
| 1 | fish |
| 2 | cat |
| 50 | middle |
| 100 | rabbit |
+----+-----+
11 rows in set (0.00 sec)
mysql> alter table animals AUTO_INCREMENT=200;
```

```
Query OK, 0 rows affected (0.22 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+
| Table | Create Table |
+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=200 DEFAULT CHARSET=utf8 |
+-----+
```

- If the new value of **AUTO_INCREMENT** is still greater than the maximum value of the auto-increment column in the table, the value change is successful. Otherwise, the value is changed to the maximum value of the auto-increment column plus 1 by default.

```
mysql> select * from animals;
+-----+
| id | name |
+-----+
| -50 | -middle |
| 1 | fish |
| 2 | cat |
| 50 | middle |
| 100 | rabbit |
+-----+
mysql> show create table animals;
+-----+
| Table | Create Table |
+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=200 DEFAULT CHARSET=utf8 |
+-----+
mysql> alter table animals AUTO_INCREMENT=150;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+
| Table | Create Table |
+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=150 DEFAULT CHARSET=utf8 |
+-----+
mysql> alter table animals AUTO_INCREMENT=50;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+
| Table | Create Table |
+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=101 DEFAULT CHARSET=utf8 |
+-----+
mysql> delete from animals where id=100;
Query OK, 1 row affected (0.00 sec)
mysql> select * from animals;
+-----+
| id | name |
+-----+
| -50 | -middle |
| 1 | fish |
| 2 | cat |
| 50 | middle |
+-----+
10 rows in set (0.00 sec)
mysql> alter table animals AUTO_INCREMENT=50;
Query OK, 0 rows affected (0.04 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table animals;
+-----+-----+
| Table | Create Table |
+-----+-----+
| animals | CREATE TABLE `animals` (
  `id` mediumint NOT NULL AUTO_INCREMENT, `name` char(30) NOT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=51 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

- The value of **AUTO_INCREMENT** cannot be changed to a negative number.
mysql> alter table animals AUTO_INCREMENT=-1;
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 '-1' at line 1

9.1.7.5 Failed to Insert Data Because Values for the Auto-increment Primary Key Field Reach the Upper Limit

Scenario

The error message "ERROR 1062 (23000): Duplicate entry 'xxx' for key 'xxx'" was displayed when data was inserted into a table.

Possible Causes

The values for the auto-increment primary key field reaches the upper limit and cannot be increased. As a result, the auto-increment primary key value generated for the newly inserted data is the same as that of the previous data record in the table. Since the auto-increment primary key value cannot be duplicate, an error is reported.

```
mysql> create table auto_test5(id tinyint not null AUTO_INCREMENT, name varchar(8), PRIMARY KEY
(`id`));
Query OK, 0 rows affected (0.06 sec) mysql> insert into auto_test5(name) values('A'),('B'),('C');
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test5;
+----+-----+
| id | name |
+----+-----+
| 1 | A |
| 2 | B |
| 3 | C |
+----+-----+
3 rows in set (0.00 sec)
mysql> alter table auto_test5 AUTO_INCREMENT=125;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> show create table auto_test5;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test5 | CREATE TABLE `auto_test5` (
  `id` tinyint NOT NULL AUTO_INCREMENT, `name` varchar(8) DEFAULT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=125 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> insert into auto_test5(name) values('X'),('Y'),('Z');
Query OK, 3 rows affected (0.00 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from auto_test5;
+----+-----+
| id | name |
+----+-----+
```

```

| 1 | A |
| 2 | B |
| 3 | C |
|125| X |
|126| Y |
|127| Z |
+-----+
6 rows in set (0.00 sec)
mysql> show create table auto_test5;
+-----+
| Table      | Create Table
+-----+
| auto_test5 | CREATE TABLE `auto_test5` ( `id` tinyint NOT NULL AUTO_INCREMENT, `name` varchar(8)
DEFAULT NULL, PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=127 DEFAULT CHARSET=utf8 |
+-----+
mysql> insert into auto_test5(name) values('D');
ERROR 1062 (23000): Duplicate entry '127' for key 'auto_test5.PRIMARY'

```

Solution

- If there are many data changes and the actual data volume in the table is far less than the capacity of the auto-increment primary key, import all data in the table to a new table, delete the original table, and change the name of the new table to the original table name. (There are multiple methods for importing and exporting data. The following is only an example.)

```

mysql> create table auto_test5_tmp(id tinyint not null AUTO_INCREMENT, name varchar(8),
PRIMARY KEY (`id`));

```

Query OK, 0 rows affected (0.07 sec)

```

mysql> insert into auto_test5_tmp select 0,name from auto_test5;

```

Query OK, 6 rows affected (0.01 sec)

Records: 6 Duplicates: 0 Warnings: 0

```

mysql> select * from auto_test5_tmp;

```

```

+-----+
| id | name |
+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | X |
| 5 | Y |
| 6 | Z |
+-----+

```

```

mysql> drop table auto_test5;

```

```

mysql> rename table auto_test5_tmp to auto_test5;

```

Query OK, 0 rows affected (0.12 sec)

```

mysql> select * from auto_test5;

```

```

+-----+
| id | name |
+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 4 | X |
| 5 | Y |
| 6 | Z |
+-----+

```

6 rows in set (0.01 sec)

```

mysql> show create table auto_test5;

```

```

+-----+
| Table      | Create Table
+-----+
| auto_test5 | CREATE TABLE `auto_test5` (
`id` tinyint NOT NULL AUTO_INCREMENT, `name` varchar(8) DEFAULT NULL,
PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8 |
+-----+

```

- If the values for the auto-increment primary key are not enough, change the field type of the auto-increment primary key to store more data.

```
mysql> select * from auto_test6;
+----+-----+
| id | name |
+----+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 125 | X |
| 126 | Y |
| 127 | Z |
+----+-----+
6 rows in set (0.00 sec)
mysql> show create table auto_test6;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test6 | CREATE TABLE `auto_test6` (
  `id` tinyint NOT NULL AUTO_INCREMENT, `name` varchar(8) DEFAULT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=127 DEFAULT CHARSET=utf8 |
+-----+-----+
mysql> alter table auto_test6 modify column id int NOT NULL AUTO_INCREMENT;
Query OK, 6 rows affected (0.15 sec)
Records: 6 Duplicates: 0 Warnings: 0
mysql> show create table auto_test6;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test6 | CREATE TABLE `auto_test6` (
  `id` int NOT NULL AUTO_INCREMENT, `name` varchar(8) DEFAULT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=128 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
mysql> insert into auto_test6(name) values('D');
Query OK, 1 row affected (0.01 sec)
mysql> select * from auto_test6;
+----+-----+
| id | name |
+----+-----+
| 1 | A |
| 2 | B |
| 3 | C |
| 125 | X |
| 126 | Y |
| 127 | Z |
| 128 | D |
+----+-----+
7 rows in set (0.00 sec)
mysql> show create table auto_test6;
+-----+-----+
| Table | Create Table |
+-----+-----+
| auto_test6 | CREATE TABLE `auto_test6` (
  `id` int NOT NULL AUTO_INCREMENT, `name` varchar(8) DEFAULT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO_INCREMENT=129 DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.01 sec)
```

9.1.7.6 Impact of Creating an Empty Username

The username "" is allowed in RDS for MySQL instances, but using such an empty username has negative impacts on instances.

When you perform operations on an RDS for MySQL instance using an empty username, any username can be matched in RDS. This brings security and functional impacts on your instance. You are advised not to use empty usernames in actual scenarios.

- Security impact

- Your instance can be connected using any username if an empty username exists.
- Your database can be logged in using any username and the password of the empty username and the login user will obtain all permissions of the empty username. Example:

#If there is no empty username created and the invalid username abcd is used to connect to the instance, the connection fails.

```
mysql> select user,host from mysql.user;
```

```
+-----+-----+
| user      | host      |
+-----+-----+
| root      | %         |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys  | localhost |
+-----+-----+
```

```
mysql -uabcd -h127.0.0.1 -P3306 -pTest_1234
```

```
mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
ERROR 1045 (28000): Access denied for user 'abcd'@'localhost' (using password: YES)
```

#If an empty username has been created and the invalid username abcd and the password of the empty username are used to connect to the instance, the connection is successful.

```
mysql> create user ''@'localhost' IDENTIFIED BY 'Test_1234';
```

```
mysql> select user,host from mysql.user;
```

```
+-----+-----+
| user      | host      |
+-----+-----+
| root      | %         |
|           | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys  | localhost |
+-----+-----+
```

```
mysql -uabcd -h127.0.0.1 -P3306 -pTest_1234
```

```
mysql: [Warning] Using a password on the command line interface can be insecure.
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 37Server version: 8.0.22-debug Source distribution
```

```
Copyright (c) 2000, 2020, 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>
```

- If the empty user does not have a password, you can use any username to log in to the instance without a password and obtain all permissions of the empty user. Example:

#If there is an empty username that does not have a password, the database can be logged in using any username without a password.

```
mysql> create user ''@'localhost';
```

```
Query OK, 0 rows affected (8.87 sec)
```

```
mysql> select user,host from mysql.user;
```

```
+-----+-----+
| user      | host      |
+-----+-----+
| root      | %         |
|           | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys  | localhost |
+-----+-----+
```

```
mysql -uabcd -h127.0.0.1 -P3306
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

```
Your MySQL connection id is 39Server version: 8.0.22-debug Source distribution
```

```
Copyright (c) 2000, 2020, 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>
#-----
mysql -usdhsjkdshk -h127.0.0.1 -P3306
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 40Server version: 8.0.22-debug Source distribution
Copyright (c) 2000, 2020, 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>
```

- Functional impact

If an empty username exists, the database cannot be logged in using a correct username due to a name matching error.

Example: If the host of an empty user overlaps that of the **root** user, the **root** user cannot log in to the database using its password or it can log in to the database using the password of the empty username but cannot obtain the **root** user permissions.

```
mysql> create user ''@'localhost';
Query OK, 0 rows affected (8.87 sec)
mysql> select user,host from mysql.user;
+-----+-----+
| user      | host      |
+-----+-----+
| root      | %         |
|           | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys   | localhost |
+-----+-----+

#The database cannot be logged in using the password of the root user.
mysql -uroot -h127.0.0.1 -P3306 -pTest_root
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
#The user who logs in to the database using the password of the empty user (password-free) is
actually an empty user so the user does not have the root permissions.
mysql -uroot -h127.0.0.1 -P3306
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45Server version: 8.0.22-debug Source distribution
Copyright (c) 2000, 2020, 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> select user,host from mysql.user;
ERROR 1142 (42000): SELECT command denied to user ''@'localhost' for table 'user'
mysql>
```

9.1.7.7 Connection to a Primary/Standby DB Instance Suspended Using pt-osc

Scenario

The pt-online-schema-change (pt-osc for short) tool can be used to perform online DDL operations on a local single instance, but cannot be used on an RDS for MySQL primary/standby instance and the connection is suspended, as shown in the following figure.

```
[e]# pt-online-schema-change --user=root --password= --host= --port=3306
,D=apppalaxy_test,t=T_RULEMODIFYRECORD --charset=utf8 --alter="ENGINE=InnoDB" --nocheck-replication-filters --alter-f
oreign-keys-method=auto --execute > /software/169.log
```

Possible Causes

How pt-osc works:

1. Create an empty table with the same structure as the original table. The table name is suffixed with **_new**.
2. Modify the structure of the empty table created in 1.
3. Add three triggers to the original table: **delete**, **update**, and **insert**. The triggers are used to execute the statements to be executed in the original table in the new table during data copy.
4. Copy the data in the original table to the new table in the form of data chunks.
5. Rename the original table, change the name of the new table to that of the original table, and delete the original table.
6. Delete the triggers.

A large amount of data needs to be copied. There will be a replication delay between the RDS for MySQL primary and standby instances. Workloads running on the standby instance may be affected. Considering the replication delay, the pt-osc provides the following control options:

- **--max-lag**
- **--check-interval**
- **--recursion-method**
- **--check-slave-lag**

If the replication delay of the standby instance exceeds the value of **max-lag**, the tool pauses the data copy and sleeps for **check-interval** seconds. If you specify **check-slave-lag**, the tool only examines that server for replication delay, not all servers. **recursion-method** is used to control exactly which servers the tool monitors. Its values include **processlist** (default value, monitoring the primary/standby replication delay), **hosts**, **dsn**, and **none** (ignoring the primary/standby replication delay). For more information, see [pt-online-schema-change](#).

In this case:

- When pt-osc is used to connect to the RDS for MySQL primary/standby instance, the connection is suspended because there is a primary/standby replication delay and the tool stops copying data. You can add **--recursion-method=none** to solve the problem.
- If the primary/standby replication delay is ignored, data copy becomes fast. To minimize the impact on workloads, you can set the **--max-load** configuration item.

Solution

Add the **--recursion-method=none** configuration item to the pt-osc to ignore the replication delay.

Common commands of pt-osc:

- Adding a field

```
pt-online-schema-change --user=root --password=xxx --host=xxx --alter "ADD
COLUMN content text" D=aaa,t=tmp_test --no-check-replication-filters --
alter-foreign-keys-method=auto --recursion-method=none --print --execute
```

- Deleting a field

```
pt-online-schema-change --user=root --password=xxx --host=xxx --alter
"DROP COLUMN content " D=aaa,t=tmp_test --no-check-replication-filters --
alter-foreign-keys-method=auto --recursion-method=none --quiet --execute
```

- Modifying a field

```
pt-online-schema-change --user=root --password=xxx --host=xxx --alter
"MODIFY COLUMN age TINYINT NOT NULL DEFAULT 0" D=aaa,t=tmp_test --
no-check-replication-filters --alter-foreign-keys-method=auto --recursion-
method=none --quiet --execute
```

- Renaming a field

```
pt-online-schema-change --user=root --password=xxx --host=xxx --alter
"CHANGE COLUMN age address varchar(30)" D=aaa,t=tmp_test --no-check-
alter --no-check-replication-filters --alter-foreign-keys-method=auto --
recursion-method=none --quiet --execute
```

- Adding an index

```
pt-online-schema-change --user=root --password=xxx --host=xxx --alter "ADD
INDEX idx_address(address)" D=aaa,t=tmp_test --no-check-alter --no-check-
replication-filters --alter-foreign-keys-method=auto --recursion-method=none
--print --execute
```

- Deleting an index

```
pt-online-schema-change --user=root --password=xxx --host=xxx --alter
"DROP INDEX idx_address" D=aaa,t=tmp_test --no-check-alter --no-check-
replication-filters --alter-foreign-keys-method=auto --recursion-method=none
--print --execute
```

If the primary/standby replication delay is important for your workloads, adjust the following parameters as required: **max-lag**, **check-interval**, **recursion-method**, and **check-slave-lag**. For more information, see [pt-online-schema-change](#).

9.2 RDS for SQL Server

9.2.1 Failed to Bind a New User to an RDS for SQL Server DB Instance That Was Recovered from Backups

Scenarios

An RDS for SQL Server DB instance is recovered from backup, and binding a newly created user to this instance fails. An error message "User, group, or role 'ecology' already exists in the current database" is displayed.

Possible Causes

After the DB instance recovers, the user already exists but is not displayed on the console.

Solution

Run the following commands as user **rdsuser** to connect to the DB instance and change its username (*ecology* is just an example):

```
use [DBName];  
Exec sp_change_users_login 'AUTO_FIX','ecology';
```