Relational Database Service

User Guide

Issue 01

Date 2025-03-07

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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 Working with RDS for MySQL 1
1.1 Using IAM to Grant Access to RDS 1
1.1.1 Creating a User and Granting Permissions 1
1.1.2 RDS Custom Policies 2
1.2 Buying an RDS for MySQL DB Instance 3
1.3 Instance Connection 18
1.3.1 Overview 18
1.3.2 Connecting to an RDS for MySQL Instance Through DAS (Recommended) 21
1.3.3 Connecting to an RDS for MySQL Instance Through the MySQL CLI Client 23
1.3.3.1 Using MySQL CLI to Connect to an Instance Through a Private Network 23
1.3.3.2 Using MySQL CLI to Connect to an Instance Through a Public Network 30
1.3.3.3 Installing a MySQL Client 35
1.3.4 Connecting to an RDS for MySQL Instance Through MySQL-Front 36
1.3.5 Connecting to an RDS for MySQL Instance Through JDBC 39
1.3.6 Connection Management 45
1.3.6.1 Changing a Floating IP Address 45
1.3.6.2 Changing a Private Domain Name 47
1.3.6.3 Changing a Database Port 48
1.3.6.4 Binding and Unbinding an EIP 49
1.3.6.5 Applying for and Changing a Public Domain Name 51
1.3.6.6 Configuring a Certificate 52
1.3.6.7 Configuring a Security Group Rule 53
1.4 Database Usage 56
1.4.1 Suggestions on Using RDS for MySQL 56
1.4.1.1 Instance Usage Suggestions 57
1.4.1.2 Database Usage Suggestions 59
1.4.2 Database Management 66
1.4.2.1 Creating a Database 66
1.4.2.2 Modifying Database Remarks 68
1.4.2.3 Granting Database Permissions 69
1.4.2.4 Deleting a Database 70
1.4.3 Account Management (Non-Administrator) 71
1.4.3.1 Creating a Database Account 71

1.4.3.2 Resetting a Password for a Database Account 75
1.4.3.3 Changing Permissions for a Database Account 76
1.4.3.4 Modifying Host IP Addresses 78
1.4.3.5 Deleting a Database Account 79
1.5 Database Migration 80
1.5.1 Migration Solution Overview 80
1.5.2 Migrating Data to RDS for MySQL Using mysqldump 85
1.5.3 Migrating Data to RDS for MySQL Using the Export and Import Functions of DAS 89
1.6 Version Upgrade 93
1.6.1 Upgrading a Minor Version 93
1.6.2 Upgrading an RDS for MySQL Instance from 5.7 to 8.0 96
1.6.3 Upgrading an RDS for MySQL Instance from 5.6 to 5.7 108
1.7 Instance Management 111
1.7.1 Instance Overview 111
1.7.2 Monitoring Dashboard 120
1.7.3 Instance Lifecycle 126
1.7.3.1 Buying a Same DB Instance as an Existing DB Instance 126
1.7.3.2 Stopping an Instance 127
1.7.3.3 Starting an Instance 129
1.7.3.4 Rebooting DB Instances or Read Replicas 130
1.7.3.5 Selecting Displayed Items 133
1.7.3.6 Exporting DB Instance Information 134
1.7.3.7 Deleting Pay-per-Use DB Instances or Read Replicas 135
1.7.3.8 Recycling a DB Instance 137
1.8 Instance Modifications 139
1.8.1 Changing a DB Instance Name 139
1.8.2 Changing a DB Instance Description 140
1.8.3 Changing the Replication Mode 140
1.8.4 Changing the Failover Priority 141
1.8.5 Changing Read/Write Permissions 142
1.8.6 Enabling or Disabling Event Scheduler 144
1.8.7 Changing a DB Instance Class 145
1.8.8 Changing a Storage Type 153
1.8.9 Configuring Auto Scaling of vCPUs and Memory 156
1.8.10 Scaling Up Storage Space 160
1.8.11 Configuring Storage Autoscaling 163
1.8.12 Changing the Maintenance Window 165
1.8.13 Changing a DB Instance Type from Single to Primary/Standby 167
1.8.14 Promoting a Read Replica to Primary 169
1.8.15 Manually Switching Between Primary and Standby DB Instances 170
1.8.16 Changing the AZ of a Standby DB Instance 172
1.8.17 Updating the OS of a DB Instance 173

1.9 Data Backups ... 173
1.9.1 Introduction to Backups ... 173
1.9.2 Backup Types ... 175
1.9.3 Performing Backups ... 178
1.9.3.1 Configuring an Intra-Region Backup Policy ... 178
1.9.3.2 Configuring a Cross-Region Backup Policy ... 184
1.9.3.3 Creating a Manual Backup ... 188
1.9.3.4 Replicating a Backup ... 191
1.9.4 Managing Backups ... 192
1.9.4.1 Downloading a Full Backup File ... 192
1.9.4.2 Downloading a Binlog Backup File ... 197
1.9.4.3 Checking and Exporting Backup Information ... 201
1.9.4.4 Using mysqlbinlog to View Binlogs ... 202
1.9.4.5 Deleting a Manual Backup ... 203
1.9.5 Clearing Binlogs ... 204
1.9.5.1 Setting a Local Retention Period for RDS for MySQL Binlogs ... 204
1.9.5.2 Clearing Binlogs from DB Instances ... 206
1.10 Data Restorations ... 207
1.10.1 Restoration Solutions ... 207
1.10.2 Restoring Data to RDS for MySQL ... 210
1.10.2.1 Restoring a DB Instance from Backups ... 210
1.10.2.2 Restoring a DB Instance to a Point in Time ... 214
1.10.2.3 Restoring Databases or Tables to a Point in Time ... 219
1.10.2.4 Restoring Data Across Regions ... 223
1.10.3 Restoring Data to an On-Premises MySQL Database ... 225
1.11 Read Replicas ... 229
1.11.1 Introduction to Read Replicas ... 229
1.11.2 Creating an HA Read Replica ... 232
1.11.3 Creating a Single Read Replica ... 237
1.11.4 Changing a Single-Node Read Replica to an HA Read Replica ... 243
1.11.5 Creating Read Replicas in Batches ... 245
1.11.6 Managing a Read Replica ... 246
1.12 Database Proxy (Read/Write Splitting) ... 247
1.12.1 Introduction to RDS for MySQL Database Proxy ... 247
1.12.2 Constraints on Database Proxy ... 252
1.12.3 Using RDS for MySQL Database Proxies for Read/Write Splitting ... 254
1.12.4 Database Proxy Configurations ... 266
1.12.4.1 Configuring Transaction Splitting ... 267
1.12.4.2 Configuring Connection Pools ... 268
1.12.4.3 Modifying Read/Write Splitting Parameters ... 269
1.12.4.4 Configuring the Delay Threshold and Routing Policy ... 270
1.12.4.5 Enabling or Disabling Access Control ... 275

1.12.4.6 Changing the Read/Write Splitting Address 277
1.12.4.7 Applying for and Changing a Private Domain Name for a Database Proxy 279
1.12.4.8 Changing the Read/Write Splitting Port 280
1.12.4.9 Changing the Number of Proxy Nodes 281
1.12.4.10 Changing the Instance Class of a DB Proxy Instance 282
1.12.4.11 Configuring Multi-Statement Processing Modes 284
1.12.4.12 Changing a Proxy from Pay-per-Use to Yearly/Monthly 285
1.12.5 Database Proxy Lifecycle 286
1.12.5.1 Restarting a Database Proxy 286
1.12.5.2 Disabling Read/Write Splitting 287
1.12.6 Database Proxy Kernel Versions 288
1.12.6.1 Kernel Versions 288
1.12.6.2 Upgrading the Kernel Version of Database Proxy 290
1.12.7 Best Practices for Database Proxy 292
1.13 Problem Diagnosis and SQL Analysis 294
1.13.1 Function Overview 294
1.13.2 Performance Monitoring 296
1.13.2.1 Viewing the Overall Status of a DB Instance 296
1.13.2.2 Viewing Performance Metrics of a DB Instance 299
1.13.3 Problem Diagnosis 299
1.13.3.1 Managing Real-Time Sessions 300
1.13.3.2 Managing Disk Capacity 302
1.13.3.3 Managing Locks & Transactions 306
1.13.3.4 Managing Historical Transactions 309
1.13.3.5 Daily Reports 312
1.13.3.6 Managing Anomaly Snapshots 315
1.13.4 SQL Analysis 316
1.13.4.1 Viewing Slow Query Logs of a DB Instance 316
1.13.4.2 Viewing Top SQL Statements of a DB Instance 318
1.13.4.3 Creating a SQL Insights Task 319
1.13.4.4 Creating a Concurrency Control Rule 322
1.13.4.5 Configuring Auto Flow Control 327
1.13.5 Common Performance Problems 330
1.13.5.1 High CPU Usage of RDS for MySQL Instances 331
1.13.5.2 High Memory Usage of RDS for MySQL Instances 334
1.13.5.3 Full Storage of RDS for MySQL Instances 336
1.13.5.4 RDS for MySQL Metadata Locks 337
1.13.5.5 Troubleshooting Slow SQL Issues for RDS for MySQL DB Instances 339
1.14 Security and Encryption 341
1.14.1 Database Account Security 341
1.14.2 Resetting the Administrator Password to Restore Root Access 343
1.14.3 Changing a Security Group 345

1.14.4 Performing a Server-Side Encryption 347
1.14.5 Configuring an SSL Connection 348
1.14.6 Configuring the TDE Function 349
1.14.7 Configuring a Password Expiration Policy 353
1.14.8 Unbinding an EIP 354
1.14.9 Using the Database of the Latest Version 355
1.14.10 Using DBSS (Recommended) 356
1.15 Parameters 357
1.15.1 Modifying Parameters of an RDS for MySQL Instance 357
1.15.2 Managing Parameter Templates 365
1.15.2.1 Creating a Parameter Template 365
1.15.2.2 Applying a Parameter Template 370
1.15.2.3 Replicating a Parameter Template 372
1.15.2.4 Resetting a Parameter Template 373
1.15.2.5 Comparing Parameter Templates 374
1.15.2.6 Exporting a Parameter Template 376
1.15.2.7 Importing a Parameter Template 378
1.15.2.8 Modifying a Parameter Template Description 379
1.15.2.9 Deleting a Parameter Template 379
1.15.3 Suggestions on RDS for MySQL Parameter Tuning 380
1.16 Log Management 383
1.16.1 Log Reporting 384
1.16.2 Viewing and Downloading Error Logs 387
1.16.3 Viewing and Downloading Slow Query Logs 392
1.16.4 Viewing Failover/Switchover Logs 400
1.16.5 Enabling SQL Audit 401
1.16.6 Downloading SQL Audit Logs 408
1.17 Metrics and Alarms 410
1.17.1 Configuring Displayed Metrics 410
1.17.2 Viewing Monitoring Metrics 430
1.17.3 Setting Alarm Rules 431
1.17.4 Configuring Alarm Reporting 434
1.17.5 Configuring Monitoring by Seconds 435
1.17.6 Event Monitoring 437
1.17.6.1 Introduction to Event Monitoring 437
1.17.6.2 Viewing Event Monitoring Data 438
1.17.6.3 Creating an Alarm Rule to Monitor an Event 438
1.17.6.4 Events Supported by Event Monitoring 440
1.18 Billing Management 449
1.18.1 Renewing DB Instances 449
1.18.2 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly 451
1.18.3 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use 452

1.18.4 Unsubscribing from a Yearly/Monthly Instance....453
1.19 Interconnection with CTS....457
1.19.1 Key Operations Supported by CTS....457
1.19.2 Viewing Tracing Events....459
1.20 Task Center....459
1.20.1 Viewing a Task....459
1.20.2 Deleting a Task Record....461
1.21 RDS for MySQL Tags....462
1.22 RDS for MySQL Quotas....464
1.23 RDS Memory Acceleration....465
1.23.1 Memory Acceleration Overview....465
1.23.2 Enabling and Using Memory Acceleration....466
1.23.3 Modifying and Deleting a Memory Acceleration Rule....474
1.23.4 Viewing and Removing Mappings....475
2 Working with RDS for MariaDB....477
2.1 Suggestions on Using RDS for MariaDB....477
2.1.1 Instance Usage Suggestions....477
2.1.2 Database Usage Suggestions....479
2.2 Instance Connection....482
2.2.1 Connecting to an RDS for MariaDB Instance....482
2.2.2 Connecting to an RDS for MariaDB Instance Through the MySQL CLI Client....483
2.2.2.1 Using MySQL CLI to Connect to an Instance Through a Private Network....483
2.2.2.2 Using MySQL CLI to Connect to an Instance Through a Public Network....486
2.2.3 Connecting to an RDS for MariaDB Instance Through JDBC....489
2.2.4 Connecting to an RDS for MariaDB Instance Through DAS....495
2.3 Performance Tuning....496
2.3.1 What Is the Maximum Number of IOPS Supported by RDS?....496
2.3.2 How Do I Improve the Query Speed of My RDS Database?....496
2.3.3 Identifying Why CPU Usage of RDS for MariaDB Instances Is High and Providing Solutions....496
2.3.4 RDS for MariaDB Memory Usage Too High....497
2.3.5 What Should I Do If an RDS DB Instance Is Abnormal Due to Full Storage Space?....498
2.3.6 Troubleshooting Slow SQL Issues for RDS for MariaDB Instances....498
2.3.7 Resolving Insufficient Storage Issues for RDS for MariaDB Instances....500
2.4 Permissions Management....502
2.4.1 Creating a User and Granting Permissions....502
2.4.2 RDS Custom Policies....503
2.5 Instance Lifecycle....504
2.5.1 Rebooting DB Instances or Read Replicas....504
2.5.2 Selecting Displayed Items....506
2.5.3 Exporting DB Instance Information....506
2.5.4 Deleting a Pay-per-Use DB Instance or Read Replica....507
2.5.5 Modifying Recycling Policy....509

2.5.6 Rebuilding a DB Instance 510
2.6 Instance Modifications 511
2.6.1 Upgrading a Minor Version 511
2.6.2 Changing a DB Instance Name 512
2.6.3 Changing a DB Instance Description 513
2.6.4 Changing the Replication Mode 514
2.6.5 Changing the Failover Priority 515
2.6.6 Changing a DB Instance Class 516
2.6.7 Scaling Up Storage Space 517
2.6.8 Storage Autoscaling 519
2.6.9 Manually Switching Between Primary and Standby DB Instances 521
2.6.10 Changing the Maintenance Window 522
2.7 Read Replicas 523
2.7.1 Introducing Read Replicas 524
2.7.2 Creating a Read Replica 525
2.7.3 Creating Read Replicas in Batches 528
2.7.4 Managing a Read Replica 529
2.8 Data Backups 530
2.8.1 Backup Solutions 530
2.8.2 Configuring an Intra-Region Backup Policy 532
2.8.3 Creating a Manual Backup 535
2.8.4 Checking and Exporting Backup Information 537
2.8.5 Downloading a Full Backup File 537
2.8.6 Downloading a Binlog Backup File 543
2.8.7 Setting a Local Retention Period for RDS for MariaDB Binlogs 543
2.8.8 Replicating a Backup 545
2.8.9 Deleting a Manual Backup 546
2.9 Data Restorations 547
2.9.1 Restoration Solutions 547
2.9.2 Restoring a DB Instance from a Backup 547
2.9.3 Restoring a DB Instance to a Point in Time 551
2.10 Parameter Templates 553
2.10.1 Creating a Parameter Template 553
2.10.2 Modifying RDS for MariaDB Instance Parameters 555
2.10.3 Exporting a Parameter Template 557
2.10.4 Importing a Parameter Template 558
2.10.5 Comparing Parameter Templates 559
2.10.6 Viewing Parameter Change History 561
2.10.7 Replicating a Parameter Template 562
2.10.8 Resetting a Parameter Template 564
2.10.9 Applying a Parameter Template 565
2.10.10 Viewing Application Records of a Parameter Template 566

2.10.11 Modifying a Parameter Template Description 566
2.10.12 Deleting a Parameter Template 567
2.11 Connection Management 568
2.11.1 Viewing and Changing a Floating IP Address 568
2.11.2 Binding and Unbinding an EIP 569
2.11.3 Changing a Database Port 571
2.11.4 Downloading a Certificate 572
2.11.5 Configuring a Security Group Rule 573
2.12 Database Management 576
2.12.1 Creating a Database 576
2.12.2 Granting Database Permissions 577
2.12.3 Deleting a Database 578
2.12.4 Enabling or Disabling Event Scheduler 579
2.13 Account Management (Non-Administrator) 580
2.13.1 Creating a Database Account 580
2.13.2 Resetting a Password for a Database Account 583
2.13.3 Changing Permissions for a Database Account 585
2.13.4 Modifying Host IP Addresses for a Database Account 586
2.13.5 Deleting a Database Account 587
2.14 Account and Network Security 588
2.14.1 Database Account Security 588
2.14.2 Resetting the Administrator Password to Restore root Access 590
2.14.3 Configuring an SSL Connection 592
2.14.4 Configuring a Password Expiration Policy 594
2.14.5 Unbinding an EIP 595
2.14.6 Using DBSS (Recommended) 596
2.15 Metrics and Alarms 596
2.15.1 Configuring Displayed Metrics 597
2.15.2 Viewing Monitoring Metrics 610
2.15.3 Setting Alarm Rules 611
2.15.4 Event Monitoring 614
2.15.4.1 Introduction to Event Monitoring 614
2.15.4.2 Viewing Event Monitoring Data 614
2.15.4.3 Creating an Alarm Rule to Monitor an Event 615
2.15.4.4 Events Supported by Event Monitoring 617
2.16 Interconnection with CTS 626
2.16.1 Key Operations Supported by CTS 626
2.16.2 Viewing Traces 628
2.17 Log Management 629
2.17.1 Viewing and Downloading Error Logs 629
2.17.2 Viewing and Downloading Slow Query Logs 631
2.17.3 Enabling or Disabling SQL Audit 633

2.17.4 Downloading SQL Audit Logs 635
2.18 DBA Assistant 637
2.18.1 Function Overview 637
2.18.2 Viewing the Overall Status of a DB Instance 639
2.18.3 Managing Real-Time Sessions 641
2.18.3.1 Viewing Session Statistics 641
2.18.3.2 Setting a Slow Session Threshold 642
2.18.4 Viewing Performance Metrics 643
2.18.5 Subscribing to Intelligent O&M 644
2.18.6 Viewing Storage Usage 645
2.18.7 Viewing Table Diagnosis Results 648
2.18.8 Setting a Diagnosis Threshold 648
2.18.9 Viewing Top Databases and Tables by Physical File Size 650
2.18.10 Viewing Slow Query Logs 651
2.18.11 Concurrency Control 653
2.18.12 Auto Flow Control 656
2.18.13 Managing Diagnosis Reports 658
2.18.13.1 Viewing Diagnosis Reports 658
2.18.13.2 Subscribing to Diagnosis Reports 660
2.19 Task Center 661
2.19.1 Viewing a Task 661
2.19.2 Deleting a Task Record 662
2.20 Managing Tags 663
2.21 Managing Quotas 665

3 Working with RDS for PostgreSQL 667

3.1 Using IAM to Grant Access to RDS 667
3.1.1 Creating a User and Granting Permissions 667
3.1.2 RDS Custom Policies 668
3.2 Buying an RDS for PostgreSQL DB Instance 669
3.3 Instance Connection 680
3.3.1 Overview 680
3.3.2 Logging In to an RDS for PostgreSQL Instance and Creating a Database Through DAS (Recommended) 683
3.3.3 Connecting to an RDS for PostgreSQL Instance Through the psql CLI Client 689
3.3.3.1 Connecting to a DB Instance from a Linux ECS over a Private Network 690
3.3.3.2 Connecting to a DB Instance from a Linux ECS over a Public Network 697
3.3.4 Connecting to an RDS for PostgreSQL Instance Using pgAdmin 704
3.3.5 Connecting to an RDS for PostgreSQL Instance Through JDBC 713
3.3.6 Connecting to an RDS for PostgreSQL Instance Using Python 716
3.3.7 Connection Management 718
3.3.7.1 Viewing and Changing a Floating IP Address 719
3.3.7.2 Changing a Private Domain Name 720

3.3.7.3 Configuring SSL Encryption 721
3.3.7.4 Binding and Unbinding an EIP 724
3.3.7.5 Changing a Database Port 726
3.4 Database Usage 727
3.4.1 Suggestions on Using RDS for PostgreSQL 727
3.4.1.1 Instance Usage Suggestions 727
3.4.1.2 Database Usage Suggestions 730
3.4.2 Databases 731
3.4.2.1 Creating a Database 731
3.4.2.2 Modifying Database Remarks 733
3.4.2.3 Deleting a Database 733
3.4.3 Accounts (Non-Administrator) 734
3.4.3.1 Creating a Database Account 734
3.4.3.2 Resetting a Password for a Database Account 737
3.4.3.3 Modifying Remarks of a Database Account 738
3.4.3.4 Deleting a Database Account 739
3.4.3.5 Modifying pg_hba.conf 739
3.4.3.6 Viewing the pg_hba.conf Change History 743
3.4.4 Tablespace Management 744
3.5 Database Migration 746
3.5.1 Migration Solution Overview 746
3.5.2 Migrating Data to RDS for PostgreSQL Using psql 749
3.5.3 Migrating Data to RDS for PostgreSQL Using the Export and Import Functions of DAS 753
3.6 Version Upgrade 757
3.6.1 Upgrading a Minor Version 757
3.6.2 Upgrading the Major Version of a DB Instance Using SQL Commands 759
3.6.3 Upgrading the Major Version of a DB Instance on the Console 762
3.7 Instance Management 767
3.7.1 Instance Lifecycle 767
3.7.1.1 Buying a Same DB Instance as an Existing DB Instance 767
3.7.1.2 Stopping an Instance 768
3.7.1.3 Starting an Instance 769
3.7.1.4 Rebooting DB Instances or Read Replicas 770
3.7.1.5 Selecting Displayed Items 772
3.7.1.6 Exporting DB Instance Information 772
3.7.1.7 Deleting a Pay-per-Use DB Instance or Read Replica 773
3.7.1.8 Recycling a DB Instance 775
3.8 Instance Modifications 777
3.8.1 Changing a DB Instance Name 777
3.8.2 Changing a DB Instance Description 778
3.8.3 Changing the Replication Mode 778
3.8.4 Changing the Failover Priority 779

3.8.5 Changing a DB Instance Class 780
3.8.6 Changing a Storage Type 785
3.8.7 Scaling Storage Space 786
3.8.8 Configuring Storage Autoscaling 790
3.8.9 Changing the Maintenance Window 792
3.8.10 Changing a DB Instance Type from Single to Primary/Standby 794
3.8.11 Manually Switching Between Primary and Standby DB Instances 796
3.8.12 Changing the AZ of a Standby DB Instance 797
3.8.13 Updating the OS of a DB Instance 798
3.9 Data Backups 799
3.9.1 Introduction to Backups 799
3.9.2 Backup Types 801
3.9.3 Instance-Level Backups 804
3.9.3.1 Configuring an Intra-Region Backup Policy 804
3.9.3.2 Creating a Manual Backup 806
3.9.3.3 Replicating a Backup 808
3.9.4 Creating a Database-Level Backup 809
3.9.5 Managing Backups 811
3.9.5.1 Downloading an Instance-Level Backup 811
3.9.5.2 Downloading a Database-Level Backup 816
3.9.5.3 Downloading an Incremental Backup File 819
3.9.5.4 Checking and Exporting Backup Information 820
3.9.5.5 Deleting a Manual Backup 821
3.9.5.6 Stopping a Backup 821
3.10 Data Restorations 822
3.10.1 Restoration Solutions 822
3.10.2 Restoring Data to RDS for PostgreSQL 824
3.10.2.1 Restoring a DB Instance from Backups 824
3.10.2.2 Restoring a DB Instance to a Point in Time 828
3.10.2.3 Restoring Databases or Tables to a Point in Time 831
3.10.2.4 Restoring Databases from Database-Level Backups 834
3.10.3 Restoring Data to an On-Premises PostgreSQL Database from a Full Backup 835
3.11 Read Replicas 839
3.11.1 Introducing Read Replicas 839
3.11.2 Creating a Read Replica 841
3.11.3 Managing a Read Replica 846
3.11.4 Configuring Replication Delay for a Read Replica 847
3.12 DR Management 851
3.12.1 Creating a DR Relationship 851
3.12.2 Promoting a DR Instance to Primary 854
3.12.3 Removing a DR Relationship 854
3.13 Extension Management 855

3.13.1 Installing and Uninstalling an Extension on the RDS Console 855
3.13.2 Installing and Uninstalling an Extension Using SQL Commands 858
3.13.3 Supported Extensions 860
3.13.4 pg_profile_pro 867
3.13.5 pg_repack 871
3.13.6 pgl_ddl_deploy 873
3.13.7 pgvector 876
3.13.8 pgAudit 878
3.13.9 pglogical 881
3.13.10 pg_stat_statements 884
3.13.11 rds_hwdrs_ddl 886
3.13.12 rds_hwdrs_privs 887
3.13.13 HypoPG 889
3.13.14 pg_cron 891
3.13.15 dblink 895
3.13.16 rds_pg_sql_ccl 897
3.14 Problem Diagnosis and SQL Analysis 904
3.14.1 Function Overview 904
3.14.2 Performance Monitoring 906
3.14.2.1 Viewing the Overall Status of a DB Instance 906
3.14.2.2 Viewing Performance Metrics of a DB Instance 908
3.14.3 Problem Diagnosis 909
3.14.3.1 Killing Sessions 909
3.14.3.2 Managing Real-Time Sessions 910
3.14.4 SQL Analysis 911
3.14.4.1 Viewing Slow Query Logs of a DB Instance 911
3.14.4.2 Creating a SQL Insights Task 913
3.14.4.3 Creating a Concurrency Control Rule 915
3.14.5 Common Performance Problems 918
3.14.5.1 Troubleshooting High CPU Usage 918
3.14.5.2 Troubleshooting High Memory Usage 921
3.14.5.3 Troubleshooting Database Age Increase Problem 924
3.14.5.4 Troubleshooting High Storage Space Usage 928
3.14.5.5 Troubleshooting Abnormal Connections and Active Connections 933
3.14.5.6 Troubleshooting Long-Running Transactions 935
3.14.5.7 Troubleshooting Inactive Logical Replication Slots 936
3.14.5.8 Troubleshooting High Oldest Replication Slot Lag or Replication Lag 938
3.14.5.9 Troubleshooting SQL Statements That Have Been Executed for 3s or 5s 939
3.15 Security and Encryption 942
3.15.1 Database Account Security 942
3.15.2 Resetting the Administrator Password to Restore Root Access 944
3.15.3 Changing a Security Group 946

3.15.4 Performing a Server-Side Encryption 947
3.15.5 Using DBSS (Recommended) 948
3.16 Parameters 949
3.16.1 Modifying Parameters of an RDS for PostgreSQL Instance 949
3.16.2 Managing Parameter Templates 952
3.16.2.1 Creating a Parameter Template 952
3.16.2.2 Applying a Parameter Template 954
3.16.2.3 Resetting a Parameter Template 955
3.16.2.4 Replicating a Parameter Template 956
3.16.2.5 Comparing Parameter Templates 958
3.16.2.6 Importing a Parameter Template 960
3.16.2.7 Exporting a Parameter Template 961
3.16.2.8 Modifying a Parameter Template Description 963
3.16.2.9 Deleting a Parameter Template 964
3.16.2.10 Viewing Parameter Change History 964
3.16.2.11 Viewing Application Records of a Parameter Template 966
3.16.3 Suggestions on RDS for PostgreSQL Parameter Tuning 966
3.17 Log Management 967
3.17.1 Log Reporting 968
3.17.2 Viewing and Downloading Error Logs 970
3.17.3 Viewing and Downloading Slow Query Logs 973
3.17.4 Enabling SQL Audit 977
3.17.5 Downloading SQL Audit Logs 980
3.18 Metrics and Alarms 982
3.18.1 Configuring Displayed Metrics 982
3.18.2 Viewing Monitoring Metrics 1008
3.18.3 Setting Alarm Rules 1009
3.18.4 Event Monitoring 1012
3.18.4.1 Introduction to Event Monitoring 1012
3.18.4.2 Viewing Event Monitoring Data 1013
3.18.4.3 Creating an Alarm Rule to Monitor an Event 1013
3.18.4.4 Events Supported by Event Monitoring 1015
3.19 Billing Management 1021
3.19.1 Renewing DB Instances 1022
3.19.2 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly 1023
3.19.3 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use 1025
3.19.4 Unsubscribing from a Yearly/Monthly Instance 1026
3.20 Interconnection with CTS 1029
3.20.1 Key Operations Supported by CTS 1029
3.20.2 Viewing Tracing Events 1031
3.21 Task Center 1031
3.21.1 Viewing a Task 1031

3.21.2 Deleting a Task Record....1033
3.22 RDS for PostgreSQL Tags....1034
3.23 RDS for PostgreSQL Quotas....1036
3.24 RDS for PostgreSQL Enhanced Edition....1037
3.24.1 Introduction to RDS for PostgreSQL Enhanced Edition....1037
3.24.2 Functions....1037
3.24.3 System Views....1044
3.24.4 Data Types....1046
3.24.5 Implicit Type Conversion....1047
3.24.6 Predefined Parameters....1047
3.24.7 Macro Variables....1048
3.24.8 Operators....1048
3.24.9 Syntax....1048
4 Working with RDS for SQL Server....1051
4.1 Suggestions on Using RDS for SQL Server....1051
4.2 Instance Connection....1054
4.2.1 Connecting to an RDS for SQL Server Instance....1054
4.2.2 Connecting to an RDS for SQL Server Instance Through DAS (Recommended)....1054
4.2.3 Connecting to an RDS for SQL Server Instance Through the SQL Server Management Studio Client....1055
4.2.3.1 Connecting to a DB Instance from a Windows ECS....1055
4.2.3.2 Connecting to a DB Instance from a Windows Server....1061
4.2.3.3 Installing SQL Server Management Studio....1064
4.3 Database Migration....1064
4.3.1 Migration Solution Overview....1065
4.3.2 Migrating Data to RDS for SQL Server Using the Export and Import Functions of DAS....1066
4.4 Performance Tuning....1071
4.4.1 High CPU Usage of RDS for SQL Server Instances....1071
4.4.2 Full Storage of RDS for SQL Server Instances....1071
4.5 Instance Lifecycle....1073
4.5.1 Buying a Same DB Instance as an Existing DB Instance....1073
4.5.2 Stopping an Instance....1074
4.5.3 Starting an Instance....1076
4.5.4 Rebooting DB Instances or Read Replicas....1077
4.5.5 Selecting Displayed Items....1080
4.5.6 Exporting DB Instance Information....1080
4.5.7 Deleting a Pay-per-Use DB Instance or Read Replica....1081
4.5.8 Recycling a DB Instance....1083
4.6 Instance Modifications....1085
4.6.1 Changing a DB Instance Name....1085
4.6.2 Changing a DB Instance Description....1086
4.6.3 Changing the Failover Priority....1086

4.6.4 Cloning a DB Instance....1087
4.6.5 Changing a DB Instance Class....1089
4.6.6 Scaling Up Storage Space....1091
4.6.7 Configuring Autoscaling....1093
4.6.8 Changing the Maintenance Window....1095
4.6.9 Changing a DB Instance Type from Single to Primary/Standby....1096
4.6.10 Manually Switching Between Primary and Standby DB Instances....1097
4.6.11 Updating the DB Engine and OS of a DB Instance....1098
4.7 Read Replicas....1098
4.7.1 Managing a Read Replica....1098
4.8 Data Backups....1099
4.8.1 Backup Solutions....1099
4.8.2 Configuring an Intra-Region Backup Policy....1101
4.8.3 Creating a Manual Backup....1104
4.8.4 Downloading a Backup File....1105
4.8.5 Checking and Exporting Backup Information....1111
4.8.6 Replicating a Backup....1111
4.8.7 Deleting a Manual Backup....1113
4.9 Data Restorations....1113
4.9.1 Restoration Solutions....1113
4.9.2 Restoring from Backup Files to RDS for SQL Server Instances....1114
4.9.3 Restoring from Backup Files to a Self-Built SQL Server Database Using SSMS....1116
4.9.4 Restoring a DB Instance to a Point in Time....1126
4.10 Parameters....1128
4.10.1 Creating a Parameter Template....1128
4.10.2 Modifying RDS for SQL Server Instance Parameters....1130
4.10.3 Exporting a Parameter Template....1133
4.10.4 Comparing Parameter Templates....1134
4.10.5 Viewing Parameter Change History....1135
4.10.6 Replicating a Parameter Template....1137
4.10.7 Resetting a Parameter Template....1138
4.10.8 Applying a Parameter Template....1139
4.10.9 Viewing Application Records of a Parameter Template....1140
4.10.10 Modifying a Parameter Template Description....1141
4.10.11 Deleting a Parameter Template....1141
4.11 Connection Management....1142
4.11.1 Viewing and Changing a Floating IP Address....1142
4.11.2 Applying for and Changing a Private Domain Name....1144
4.11.3 Applying for and Changing a Public Domain Name....1145
4.11.4 Binding and Unbinding an EIP....1146
4.11.5 Changing a Database Port....1148
4.12 Accounts (Non-Administrator)....1150

4.12.1 Creating a Database Account ... 1150
4.12.2 Resetting a Password for a Database Account ... 1151
4.12.3 Deleting a Database Account ... 1152
4.13 Databases ... 1153
4.13.1 Creating a Database ... 1153
4.13.2 Granting Database Permissions ... 1155
4.13.3 Deleting a Database ... 1156
4.13.4 Copying a Database ... 1156
4.13.5 Viewing Database Properties ... 1157
4.14 Security and Encryption ... 1158
4.14.1 Database Account Security ... 1158
4.14.2 Resetting the Administrator Password ... 1159
4.14.3 Changing a Security Group ... 1162
4.14.4 Performing a Server-Side Encryption ... 1163
4.14.5 Configuring the TDE Function ... 1164
4.14.6 Using DBSS (Recommended) ... 1168
4.15 Distributed Transactions ... 1169
4.16 SQL Server Integration Services (SSIS) ... 1173
4.17 Metrics and Alarms ... 1177
4.17.1 Configuring Displayed Metrics ... 1177
4.17.2 Viewing Monitoring Metrics ... 1185
4.17.3 Setting Alarm Rules ... 1186
4.17.4 Event Monitoring ... 1189
4.17.4.1 Introduction to Event Monitoring ... 1189
4.17.4.2 Viewing Event Monitoring Data ... 1189
4.17.4.3 Creating an Alarm Rule to Monitor an Event ... 1190
4.17.4.4 Events Supported by Event Monitoring ... 1192
4.18 Interconnection with CTS ... 1198
4.18.1 Key Operations Supported by CTS ... 1198
4.18.2 Viewing Tracing Events ... 1201
4.19 Log Management ... 1201
4.19.1 Viewing and Downloading System Logs ... 1201
4.19.2 Viewing and Downloading Audit Logs ... 1203
4.19.3 Viewing and Downloading Slow Query Logs ... 1208
4.20 DBA Assistant ... 1211
4.20.1 Function Overview ... 1211
4.20.2 Sessions ... 1212
4.20.3 Storage Analysis ... 1213
4.20.4 Real-Time Top SQL ... 1217
4.20.5 Slow Query Log ... 1220
4.20.6 Deadlocks ... 1222
4.21 Publications and Subscriptions ... 1224

4.21.1 Creating a Publication ... 1224
4.21.2 Creating a Subscription ... 1230
4.21.3 Checking Jobs and Links ... 1233
4.22 Task Center ... 1235
4.22.1 Viewing a Task ... 1235
4.22.2 Deleting a Task Record ... 1237
4.22.3 Authorizing a Task ... 1238
4.23 Billing Management ... 1239
4.23.1 Unsubscribing from a Yearly/Monthly Instance ... 1239
4.24 Enabling or Disabling FileStream ... 1242
4.25 CLR Integration ... 1244
4.26 Default Language Setting for RDS for SQL Server ... 1249
4.27 Usage of Stored Procedures ... 1250
4.27.1 Creating a Database Account ... 1251
4.27.2 Granting SSIS Permissions to a Domain Account ... 1251
4.27.3 Deploying an SSIS Project ... 1252
4.27.4 Changing Custom Database Names ... 1253
4.27.5 Viewing Error Logs ... 1254
4.27.6 Tracing Flags ... 1254
4.27.7 Capturing Change Data ... 1255
4.27.8 Removing a Custom Database from an Availability Group ... 1256
4.27.9 Replicating Databases ... 1257
4.27.10 Granting Database Permissions to Subaccounts ... 1258
4.27.11 Deleting Custom Databases ... 1259
4.27.12 Updating Database Statistics ... 1260
4.27.13 Cycling SQL Server Agent Error Logs ... 1261
4.27.14 Cycling SQL Server Error Logs ... 1261
4.27.15 Creating Alerts ... 1262
4.27.16 Setting Up Notifications for Alert ... 1264
4.27.17 Creating Operators for Alerts and Jobs ... 1266
4.27.18 Updating Alert Settings ... 1268
4.27.19 Updating Alert Notification Methods ... 1272
4.27.20 Updating Information About Operators for Alerts and Jobs ... 1274
4.27.21 Removing Alerts ... 1276
4.27.22 Removing SQL Server Agent Notification Definitions for Specific Alerts and Operators ... 1277
4.27.23 Removing Operators ... 1278
4.27.24 Shrinking Databases ... 1279
4.27.25 Changing the Permission to View All Databases ... 1281
4.27.26 Granting Permissions of Database-Level db_owner Role ... 1282
4.28 RDS for SQL Server Tags ... 1283
4.29 RDS for SQL Server Quotas ... 1285

1 Working with RDS for MySQL

1.1 Using IAM to Grant Access to RDS

1.1.1 Creating a User and Granting Permissions

This chapter describes how to use **Identity and Access Management (IAM)** for fine-grained permissions management for your RDS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing RDS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your RDS resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

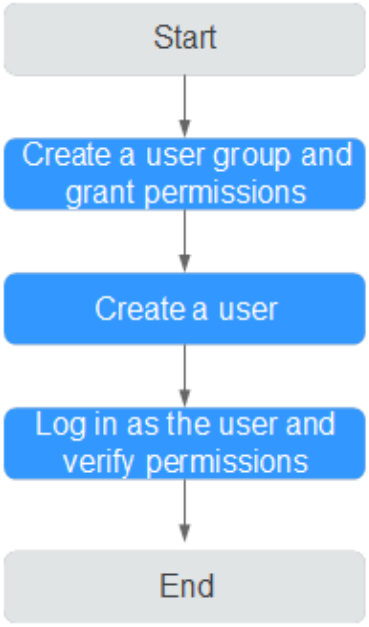
This section describes the procedure for granting permissions (see **Figure 1-1**).

Prerequisites

Learn about the permissions (see **Permissions Management**) supported by RDS and choose policies or roles according to your requirements. For the system policies of other services, see **System Permissions**.

Process Flow

Figure 1-1 Process for granting RDS permissions



1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console, and attach the **RDS ReadOnlyAccess** policy to the group.

   NOTE

   To use some interconnected services, you also need to configure permissions of such services.

   For example, to connect to your DB instance through the console, configure the **DAS FullAccess** permission of Data Admin Service (DAS) besides **RDS ReadOnlyAccess**.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the RDS console by using the created user, and verify that the user only has read permissions for RDS.

   - Choose **Service List** > **Relational Database Service** and click **Buy DB Instance**. If a message appears indicating that you have insufficient permissions to perform the operation, the **RDS ReadOnlyAccess** policy has already been applied.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **RDS ReadOnlyAccess** policy has already taken effect.

1.1.2 RDS Custom Policies

Custom policies can be created to supplement the system policies of RDS. For the actions supported for custom policies, see **Permissions Policies and Supported Actions**.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains examples of common RDS custom policies.

Example Custom Policies

- Example 1: Allowing users to create RDS DB instances

```
{
   "Version": "1.1",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["rds:instance:create"]
   }]
}
```

- Example 2: Denying RDS DB instance deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user include both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **RDS FullAccess** policy to a user but you want to prevent the user from deleting RDS DB instances. Create a custom policy for denying RDS DB instance deletion, and attach both policies to the group the user belongs to. Then, the user can perform all operations on RDS DB instances except deleting RDS DB instances. The following is an example deny policy:

```
{
   "Version": "1.1",
   "Statement": [{
      "Action": ["rds:instance:delete"],
      "Effect": "Deny"
   }]
}
```

1.2 Buying an RDS for MySQL DB Instance

Scenarios

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

RDS for MySQL supports the yearly/monthly and pay-per-use billing modes. RDS allows you to tailor your compute resources and storage space to your business needs.

You can create multiple read replicas when you are buying single or primary/standby DB instances.

Precautions

RDS for MySQL supports encryption of data in transit during primary/standby replication. To use this function, you need to request required permissions by

choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console before purchasing an instance. After a DB instance is created, **manually enable SSL** for it.

Procedure

Step 1 **Sign up for a HUAWEI ID and enable Huawei Cloud services**.

Step 2 Before purchasing DB instances, ensure that your account balance is sufficient. **Top up your account** if required.

Step 3 For fine-grained permissions management, create an Identity and Access Management (IAM) user and user group on the IAM console and grant the user specific operation permissions. For details, see **Creating a User and Granting Permissions**.

Step 4 Go to the **Buy DB Instance** page.

Step 5 On that page, click the **Custom Config** tab, select a billing mode, configure parameters about your instance, and click **Buy**.

- Billing mode
  - **Yearly/Monthly**: If you select this mode, skip **Step 6** and go to **Step 7**.
  - **Pay-per-use**: If you select this mode, go to **Step 6**.
- Engine Options

Figure 1-2 Billing mode and basic information

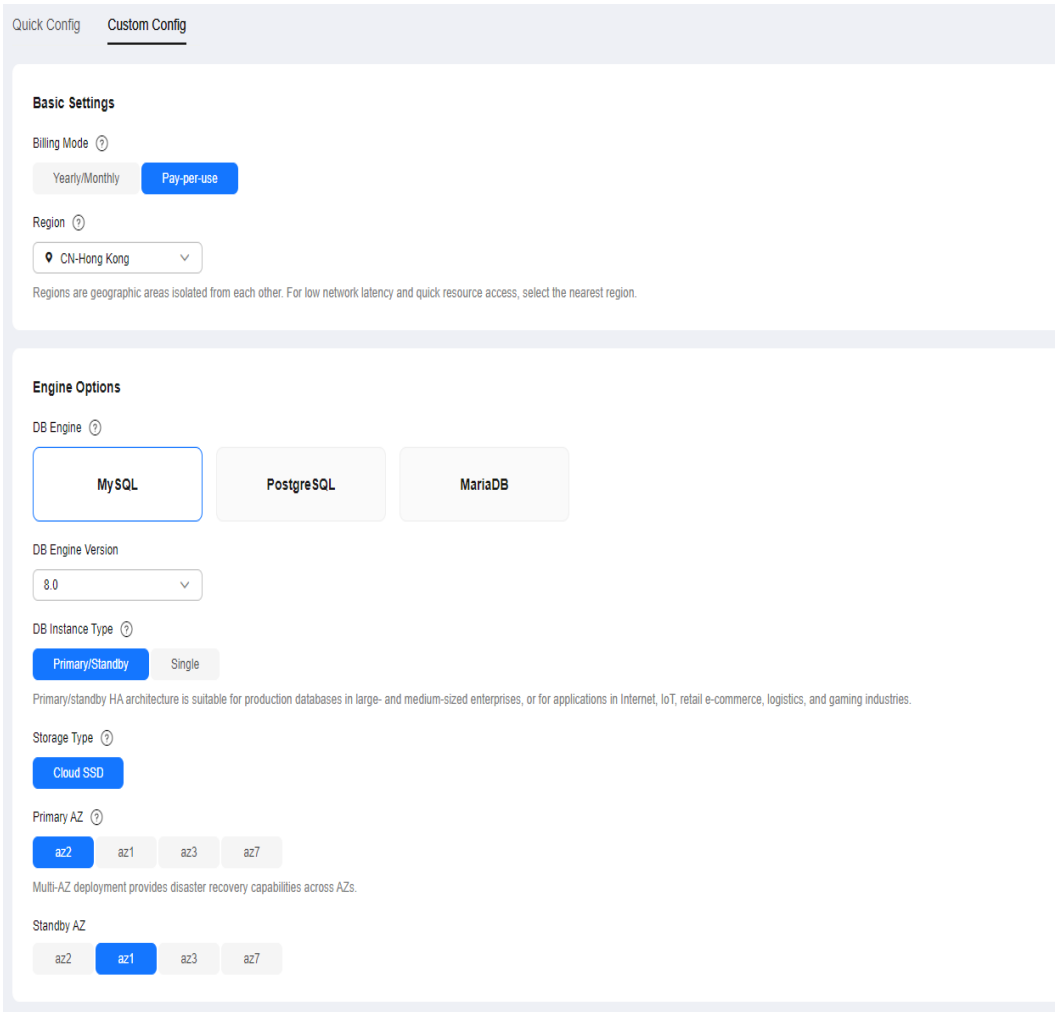


Table 1-1 Basic information

Parameter	Description
Region	Region where your resources are located. **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 Engine	Set to **MySQL**.
DB Engine Version	For details, see **DB Engines and Versions**. Different DB engine versions are supported in different regions. When creating an RDS for MySQL instance, select a proper DB engine version tailored to your workloads. You are advised to select the latest available version because it is more stable, reliable, and secure.

Parameter	Description
DB Instance Type and AZ	– **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. Some regions support both single AZs and multiple AZs and some only support single AZs. To achieve high reliability, RDS will automatically deploy your primary and standby instances in different physical servers even if you deploy them in the same AZ. If you attempt to create primary/standby DB instances in the same AZ in a Dedicated Computing Cluster (DCC) and there is only one physical server available, the creation will fail. 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 less expensive than primary/standby DB instances. It is suitable for development and testing of microsites, and small and medium enterprises, or for learning about RDS.

Parameter	Description
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. – **Cloud SSD**: cloud disks used to decouple storage from compute. The maximum throughput is 350 MB/s. – **Extreme SSD**: uses 25GE network and RDMA technologies to provide you with up to 1,000 MB/s throughput per disk and sub-millisecond latency. – **Ultra-high I/O**: uses the SSD disk type that supports a maximum throughput of 350 MB/s. **NOTE** – If you have purchased the Dedicated Distributed Storage Service (DSS), only the storage type that you have selected when you buy the DSS service is displayed. – The cloud SSD and extreme SSD storage types are supported with general-purpose, dedicated, and Kunpeng general-enhanced DB instances. – After a DB instance is created, you can change its storage type. For details, see **Changing a DB Instance Class**. – The IOPS supported by cloud SSDs depends on the I/O performance of Elastic Volume Service (EVS) disks. For details, see the description about ultra-high I/O in **Disk Types and Performance** of *Elastic Volume Service Service Overview*. – The IOPS supported by extreme SSDs depends on the I/O performance of Elastic Volume Service (EVS) disks. For details, see the description about extreme SSDs in **Disk Types and Performance** of *Elastic Volume Service Service Overview*.

- Instance Configuration

Figure 1-3 Specifications and storage

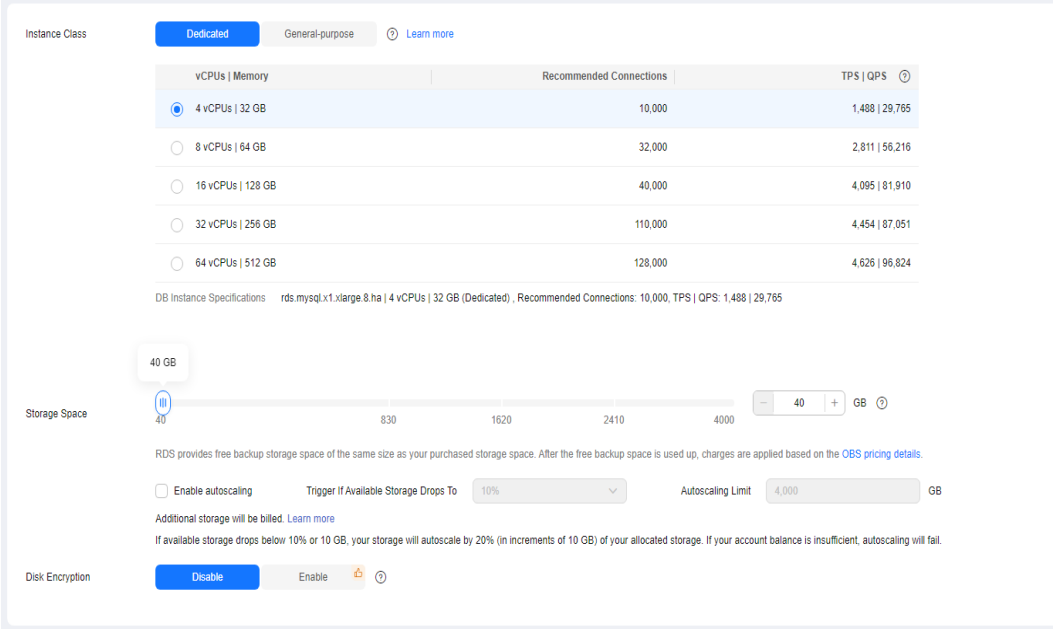


Table 1-2 Specifications and storage

Parameter	Description
Instance Class	Refers to the vCPU and memory of a DB instance. Different instance classes support different numbers of database connections and maximum IOPS. For details about instance classes, see **RDS for MySQL Instance Classes**. After a DB instance is created, you can change its vCPU and memory. For details, see **Changing a DB Instance Class**. **NOTE** Only general-enhanced DB instances are allowed for a DCC.
Resource Type	– **EVS** – **DSS** **NOTE** This option is displayed only when you have purchased **Dedicated Distributed Storage Service (DSS)**.
Storage Pool	Displayed only when you select **DSS** for **Resource Type**. The storage pool is secure because it is physically isolated from other pools.

Parameter	Description
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. If the storage type is cloud SSD or extreme SSD, you can enable storage autoscaling. If the available storage drops to a specified threshold, autoscaling is triggered. – **Enable autoscaling**: If you select this option, autoscaling is enabled. – **Trigger If Available Storage Drops To**: If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered. – **Autoscaling Limit**: The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance. After a DB instance is created, you can scale up its storage space. For details, see **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. – If you specify a read replica when creating a primary DB instance and enable storage autoscaling for the primary DB instance, storage autoscaling is also enabled for the read replica by default.

Parameter	Description
Disk Encryption	– **Disable**: Data stored in the disk is not encrypted. – **Enable**: Enabling disk encryption improves data security, but slightly affects the read and write performance of the database. ▪ **Key Name**: indicates the tenant key. Select one from the drop-down list. ▪ To create a key, click **Create Key** and configure parameters in the displayed dialog box. For more information, see **Creating a Key** in the *Data Encryption Workshop User Guide*. **NOTE** – If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. Disk encryption will not encrypt backup data stored in OBS. To enable backup data encryption, contact customer service. – If disk encryption or backup data encryption is enabled, keep the key properly. Once the key is disabled, deleted, or frozen, the database will be unavailable and data may not be restored. If disk encryption is enabled but backup data encryption is not enabled, you can **restore data to a new instance from backups**. If both disk encryption and backup data encryption are enabled, data cannot be restored. – If a shared KMS key is used, the corresponding CTS events are **createdatakey** and **decrydatakey**. Only the key owner can receive the events.

- Basic Settings and Connectivity

Figure 1-4 Basic settings and connectivity

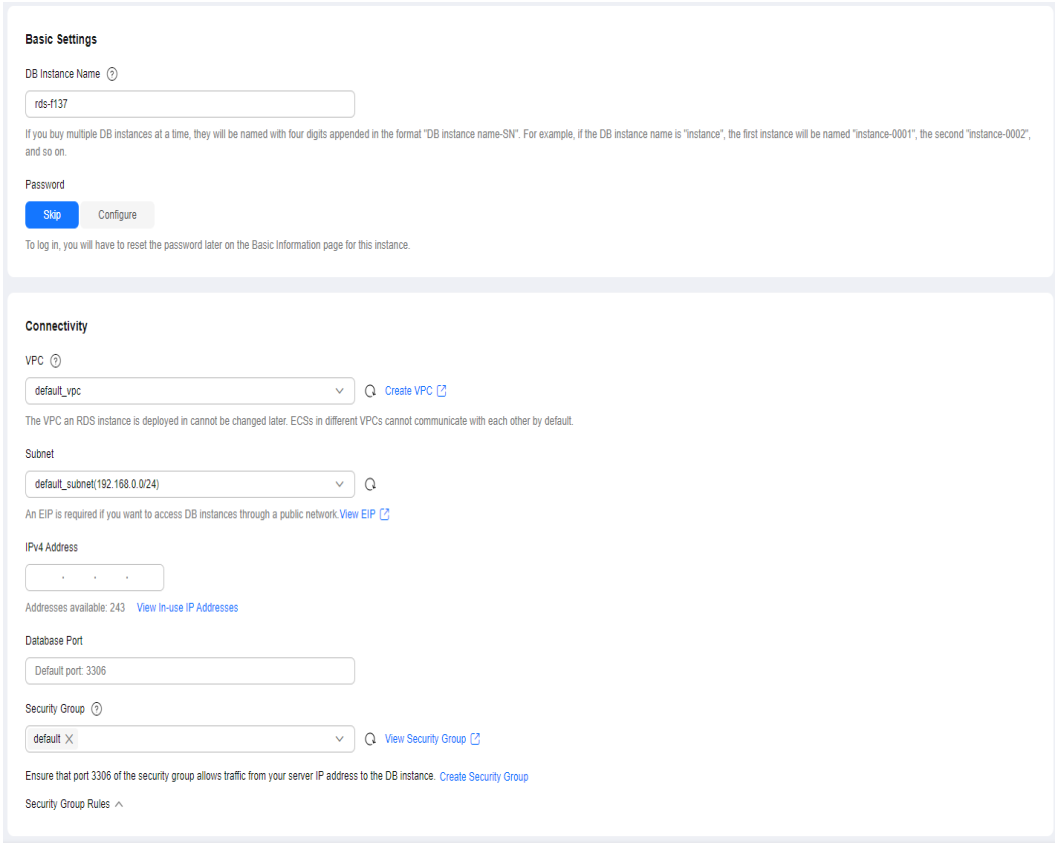
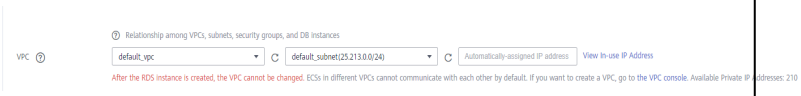


Table 1-3 Network

Parameter	Description
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed. – If you intend to create multiple DB instances and read replicas at a time, the allowed length for each instance name will change. – If you buy multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on.

Parameter	Description
Password	– **Configure** (default setting): Configure a password for your DB instance during the creation process. – **Skip**: Configure a password later after the DB instance is created. **NOTICE** If you select **Skip** for **Password**, you need to reset the password before you can log in to the instance. After a DB instance is created, you can reset the password. For details, see **Resetting the Administrator Password**.
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. If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password. Keep this password secure. The system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see **Resetting the Administrator Password**.
Confirm Password	Must be the same as **Administrator Password**.
VPC	A virtual network in which your RDS DB instances are located. A VPC can isolate networks for different workloads. You can select an existing VPC or create a VPC. For details on how to create a VPC, see "Creating a VPC" in *Virtual Private Cloud User Guide*. If no VPC is available, RDS allocates a VPC to you by default. To use a shared VPC, select a VPC that another account shares with the current account from the drop-down list. VPC owners can share the subnets in a VPC with one or multiple accounts through **Resource Access Manager (RAM)**. Through VPC sharing, you can easily configure, operate, and manage multiple accounts' resources at low costs. For more information about VPC and subnet sharing, see **VPC Sharing**. **NOTICE** After a DB instance is created, the VPC cannot be changed.

Parameter	Description
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 create RDS DB instances and cannot be disabled. – IPv4 address: A floating IPv4 address is automatically assigned when you create a DB instance. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the DB instance is created, you can change the floating IP address. – IPv6 address: A DB instance assigned a floating IPv6 address will be created only when the vCPUs and memory you selected support IPv6 addresses. A floating IPv6 address is automatically assigned during instance creation and cannot be specified. After the DB instance is created, this floating IP address cannot be changed. If no IPv6 subnets are available, **submit a service ticket**. **NOTICE** When creating a single-node DB instance, ensure that there are at least two available private IP addresses. – If you also need to create single-node read replicas, there should be at least four available private IP addresses. – If you need to create HA read replicas, there should be at least five available private IP addresses. When creating a primary/standby DB instance, ensure that there are at least three available private IP addresses. If HA read replicas are about to be created, there should be at least six available private IP addresses. **Figure 1-5** Viewing available private IP addresses 
Database Port	The default database port is **3306**. You can change it after a DB instance is created. RDS for MySQL instances can use database ports 1024 to 65535, excluding 12017, 33071, and 33062, which are reserved for RDS system use.

Parameter	Description
Security Group	Enhances security by controlling access to RDS from other services. In addition, a network **access control list (ACL)** can help control inbound and outbound traffic of subnets in your VPC. Ensure that the security group you select allows the client to access the DB instance. When creating a DB instance, you can select multiple security groups. For better network performance, you are advised to select no more than five security groups. In such a case, the access rules of all the selected security groups apply on the instance. If no security group is available or has been created, RDS allocates a security group to you by default.

- Additional Options

Figure 1-6 Additional options

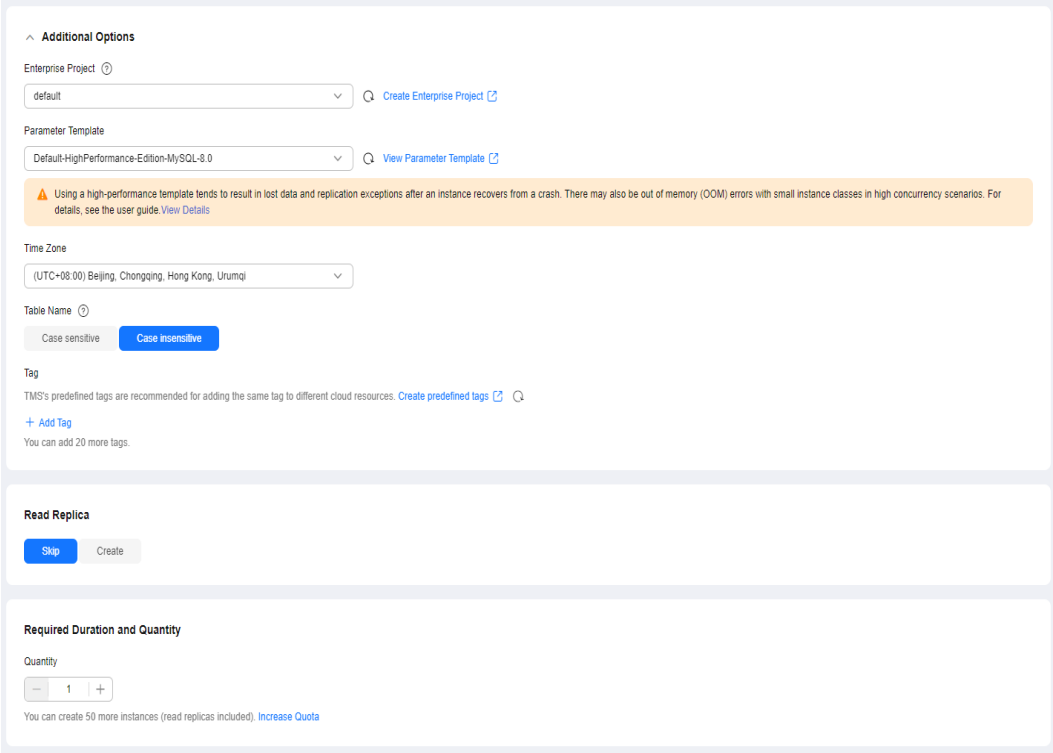


Table 1-4 Additional options

Parameter	Description
Enterprise Project	If your account has been associated with an enterprise project, select the target project from the **Enterprise Project** drop-down list. For more information about enterprise projects, see *Enterprise Management User Guide*.

Parameter	Description
Parameter Template	Contains engine configuration values that can be applied to one or more DB instances. If you intend to create a primary/standby DB pair, they use the same parameter template. You can modify the instance parameters as required after the instance is created. **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 **Modifying Parameters in a Parameter Template**.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. You can select a time zone during instance creation and change it later as needed.
Table Name	Specifies whether table names are case sensitive. The case sensitivity of table names for created RDS for MySQL 8.0 instances cannot be changed.
Tag	Tags an RDS DB instance. This parameter is optional. Adding tags to RDS DB instances helps you better identify and manage the DB instances. A maximum of 20 tags can be added for each DB instance. If your organization has configured tag policies for RDS, add tags to DB instances based on the policies. If a tag does not comply with the policies, DB instance creation may fail. Contact your organization administrator to learn more about tag policies. After a DB instance is created, you can view its tag details on the **Tags** page. For details, see **RDS for MySQL Tags**.
Certificate	(Optional) Specifies the certificate created by Cloud Certificate Manager (CCM). The default certificate is the system certificate that is automatically generated. You can also select another certificate from the drop-down list. **NOTICE** If you want to specify a certificate when creating a DB instance, contact customer service to apply for the permission.

- Read Replica

Table 1-5 Read replica

Parameter	Description
Read Replica	You can determine whether to create read replicas when creating a DB instance. – By default, **Skip** is selected. – If you select **Create**, read replicas are named with "read" and two digits appended to the primary instance name by default. 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 of read replicas are the same as those of the primary instance.
Replica Type	– **HA read replica**: If the physical server where the primary read replica resides fails, the standby read replica automatically takes over workloads from the primary read replica. When you purchase an HA read replica, select the same value for **Table Name** as the DB instance. – **Read replica**: If you want to buy single read replicas, you are advised to buy more than one single read replica and enable database proxy. That way, if one read replica fails, the database proxy can route traffic to other read replicas or the primary DB instance. When you purchase a single read replica, select the same value for **Table Name** as the DB instance.
Storage Type	Determines the instance read/write speed. The higher the maximum throughput is, the higher the read/write speed can be. – **Cloud SSD**: cloud disks used to decouple storage from compute. – **Extreme SSD**: uses the 25GE network and RDMA technology to provide you with up to 1 million random read/write performance per disk and low latency per channel. **NOTE** If you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default.
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. After a DB instance is purchased, the region cannot be changed. Therefore, exercise caution when selecting a region.
Instance Class	Refers to the vCPU and memory of a read replica.

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

- Required Duration and Quantity

Table 1-6 Required duration and quantity

Parameter	Description
Required Duration	This option is available only for yearly/monthly DB instances. The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy. If you want to set this parameter to 5 years, the restrictions are as follows: – You need to obtain the required permissions by **submitting a service ticket**. – This setting is supported only in CN North-Beijing4, CN East-Shanghai1, CN South-Guangzhou, and CN Southwest-Guiyang1. – This setting is supported only with general-purpose instances.
Auto-renew	– This option is available only for yearly/monthly DB instances and is not selected by default. – If you select this option, the auto-renew cycle is determined by the selected required duration.
Quantity	RDS supports batch creation of DB instances. If you intend to create primary/standby DB instances and set **Quantity** to **1**, a primary DB instance and a synchronous standby DB instance will be created.

If you have any questions about the price, click **Pricing details** at the bottom of the page.

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 for pay-per-use DB instances.

- If you need to modify your settings, click **Previous**.

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

Skip **Step 7** and **Step 8** and go to **Step 9**.

Step 7 Confirm the order for yearly/monthly DB instances.

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

Step 8 Select a payment method and complete the payment.

NOTE

This operation applies only to the yearly/monthly billing mode.

Step 9 To view and manage your DB instance, go to the **Instances** page.

- When your DB instance is being created, the status is **Creating**. The status changes to **Available** after the instance is created. To view the detailed progress and result of the creation, go to the **Task Center** page.
- The automated backup policy is enabled by default. You can change it after the DB instance is created. An automated full backup is immediately triggered once your DB instance is created.
- After a DB instance is created, you can enter a description for it.
- The default database port is **3306**. You can change it after a DB instance is created.

  NOTE

  You are advised to change the database port in a timely manner.

  For details, see **Changing a Database Port**.

----End

Related Operations

Creating a DB Instance Using an API

Modifying RDS for MySQL Instance Parameters

1.3 Instance Connection

1.3.1 Overview

Before connecting to a DB instance, you must create one first. For details about how to create a DB instance, see **Buying an RDS for MySQL DB Instance**. You can connect to an RDS for MySQL instance through a command-line interface (CLI), graphical user interface (GUI), Data Admin Service (DAS), or using Java database connectivity (JDBC).

Connecting to a DB Instance over a Private or Public Network Using CLI

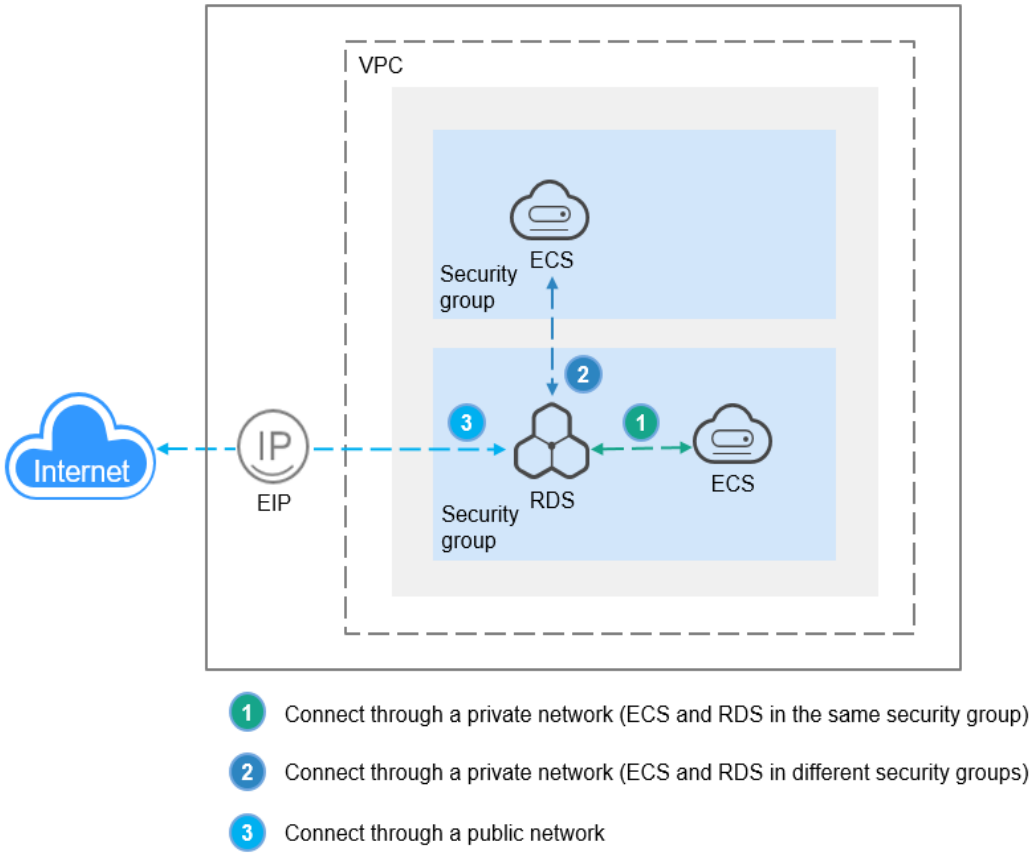
Table 1-7 lists how to use CLI to connect to an RDS for MySQL instance over a private or public network.

Table 1-7 Connecting to a DB instance over a private or public network

Connection Method	IP Address	Security Group Rules	Description
Private network	Private IP address	• If the ECS and RDS DB instance are in the same security group, they can communicate with each other over a private network by default. No security group rules need to be configured. • If they are in different security groups, 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. For details, see **Configuring a Security Group Rule**. – ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security group 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.	• Secure and high-performance • Recommended

Connection Method	IP Address	Security Group Rules	Description
Public network	You need to purchase an EIP. For pricing details, see EIP Billing.	To access a DB instance from resources outside the security group that the DB instance is associated with, you need to configure an **inbound** rule for the security group. For details, see Configuring a Security Group Rule.	• Less secure • 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 private IP address to access the DB instance.

Figure 1-7 Connecting to a DB instance over a private or public network



Connection Methods

Table 1-8 Connection methods

Connection Method	Description
Connecting to an RDS for MySQL Instance Through DAS (Recommended)	DAS enables you to manage databases on a web-based console. It supports SQL execution, advanced database management, and intelligent O&M, simplifying database management and improving both efficiency and data security. The permissions required for connecting to DB instances through DAS are enabled by default.
Connecting to an RDS for MySQL Instance Through the MySQL CLI Client	In Linux, you need to install a MySQL client on the ECS and connect to the instance through the MySQL CLI over a private or public network. • A private IP address is provided by default. When your applications are deployed on an ECS that is in the same region and VPC as the RDS for MySQL instance, you are advised to use a floating IP address to connect to the instance through the ECS. • If you cannot access your RDS instance through a floating IP address, bind an EIP to the instance and connect to the instance through the EIP.
Connecting to an RDS for MySQL Instance Through the GUI	In Windows, you can use any common database client to connect to an RDS for MySQL instance.
Connecting to an RDS for MySQL Instance Through JDBC	If you are connecting to an instance through JDBC, the SSL certificate is optional. For security reasons, you are advised to download the SSL certificate to encrypt the connection. SSL is disabled by default for RDS for MySQL instances. You can enable SSL by referring to **Configuring an SSL Connection**. SSL encrypts connections to databases but it increases the connection response time and CPU usage. Therefore, you are advised not to enable SSL.

1.3.2 Connecting to an RDS for MySQL Instance Through DAS (Recommended)

Scenarios

Data Admin Service (DAS) enables you to connect to and manage DB instances with ease on a web-based console. By default, you have the remote login permission. Using DAS to connect to your DB instance is recommended, which is more secure and convenient.

Procedure

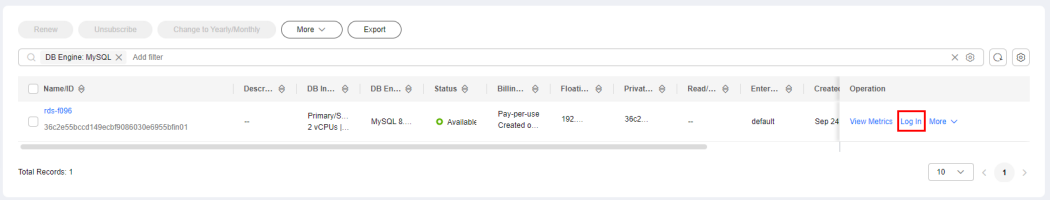
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

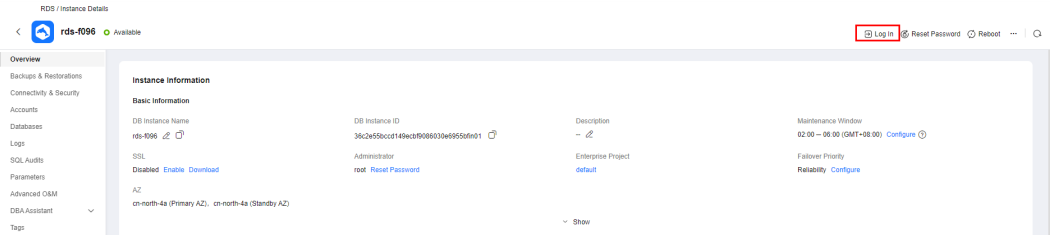
Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Figure 1-8 Logging in to an instance



Alternatively, click the DB instance on the **Instances** page. On the displayed **Overview** page, click **Log In** in the upper right corner.

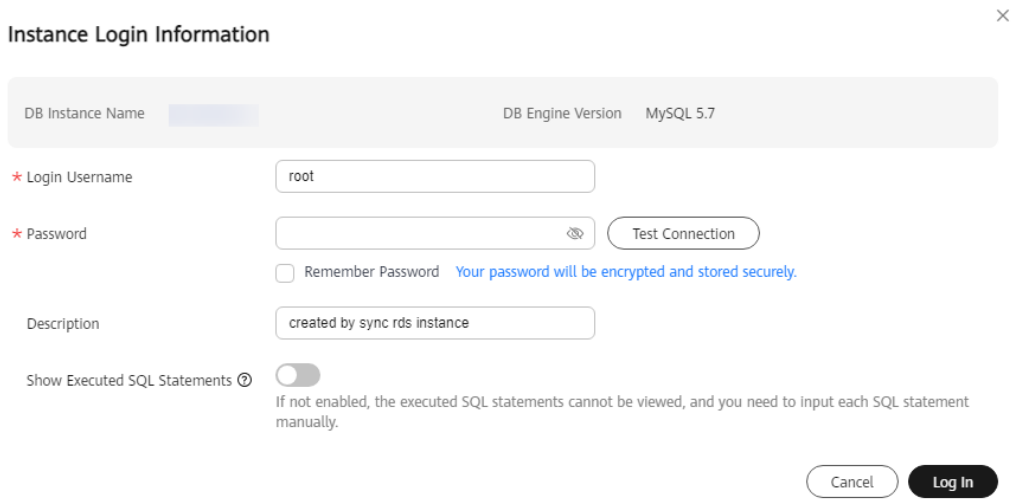
Figure 1-9 Logging in to an instance



Step 5 On the displayed login page, enter the username and password and click **Log In**.

- **Login Username**: Enter **root**.
- **Password**: Enter the **root** password you specified during instance creation. If you forget the password, reset it. For details, see **Resetting the Administrator Password to Restore Root Access**.

Figure 1-10 Login page



----End

FAQs

Q: What can I do if the DAS console is not displayed after I click **Log In** in the **Operation** column of an instance on the **Instances** page?

A: Set your browser to allow pop-ups and try again.

- **What Should I Do If I Can't Connect to My DB Instance Due to Insufficient Permissions?**
- **What Should I Do If I Can't Connect to My RDS for MySQL Instance?**

Follow-up Operations

After logging in to the DB instance, you can create or migrate your databases.

- **Creating a MySQL Database Using the Console**
- **Creating a MySQL Database Using an API**
- **Migration Solution Overview**

1.3.3 Connecting to an RDS for MySQL Instance Through the MySQL CLI Client

1.3.3.1 Using MySQL CLI to Connect to an Instance Through a Private Network

If your applications are deployed on an ECS that is in the same region and VPC as your DB instance, you are advised to use a floating IP address to connect to the DB instance through the ECS.

This section uses a Linux ECS as an example to describe how to connect a Linux ECS to a DB instance with SSL enabled through a floating IP address. SSL encrypts connections to the DB instance, making in-transit data more secure.

If you want to connect to a DB instance with SSL disabled, see **Buying a DB Instance and Connecting to It Using a MySQL Client**.

Step 1: Buy an ECS

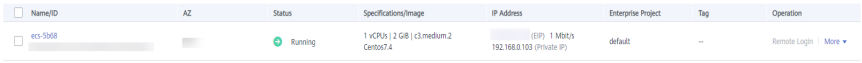
1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Linux ECS, go to **3**.

     NOTICE

     If the ECS image is CentOS, CentOS 7.4 64bit must be used.

   - If there is a Windows ECS, see **Buying a DB Instance and Connecting to It Using MySQL-Front**.
   - If no ECS is available, go to **2**.

   Figure 1-11 ECS

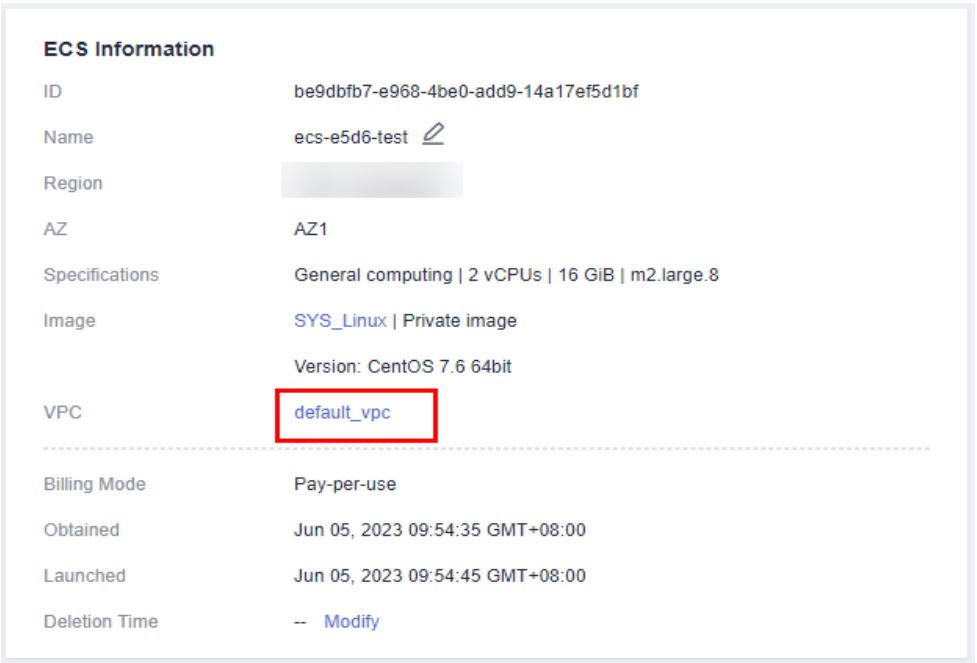
2. Buy an ECS and select Linux (for example, CentOS 7.4 64bit) as its OS.

   To download a MySQL client to the ECS, bind an EIP to the ECS. The ECS must be in the same region, VPC, and security group as the RDS for MySQL DB instance for mutual communications.

   For details about how to purchase a Linux ECS, see **Purchasing a Custom ECS** in *Elastic Cloud Server User Guide*.

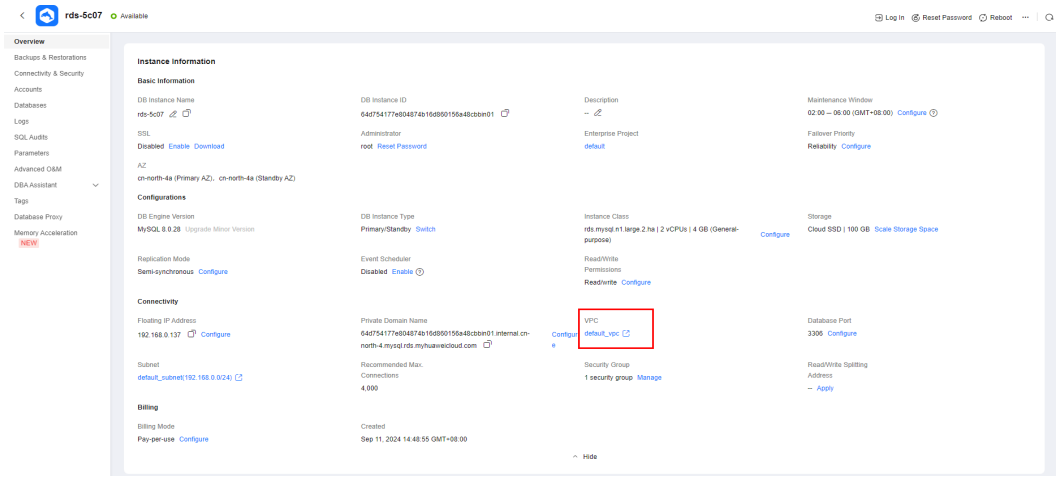
3. On the **ECS Information** page, view the region and VPC of the ECS.

   Figure 1-12 ECS information

4. On the **Overview** page of the RDS for MySQL instance, view the region and VPC of the DB instance.

Figure 1-13 Overview

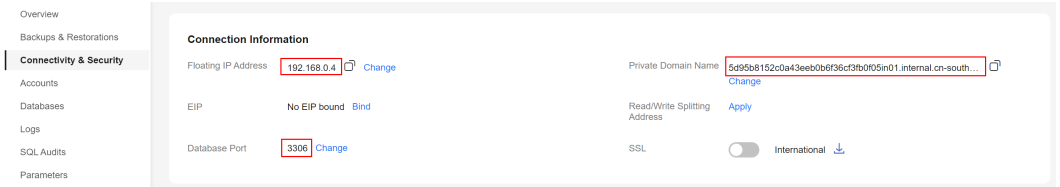


5. Check whether the ECS and RDS for MySQL instance are in the same region and VPC.
   - If yes, go to **Step 2: Test Connectivity and Install a MySQL Client**.
   - If they are not in the same region, purchase another ECS or DB instance. The ECS and DB instance in different regions cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.
   - If the ECS and DB instance are in different VPCs, change the VPC of the ECS to that of the DB instance. For details, see **Changing a VPC**.

Step 2: Test Connectivity and Install a MySQL Client

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name.
3. Choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, obtain the floating IP address (or private domain name) and database port of the DB instance.

Figure 1-14 Connection information



4. After logging in to the ECS, check whether the floating IP address and database port obtained in **3** can be connected.

```
curl -kv Floating_IP_address:Port
```

Example:

```
curl -kv 192.168.0.4:3306
```


- If yes, network connectivity is normal. Go to **5**.

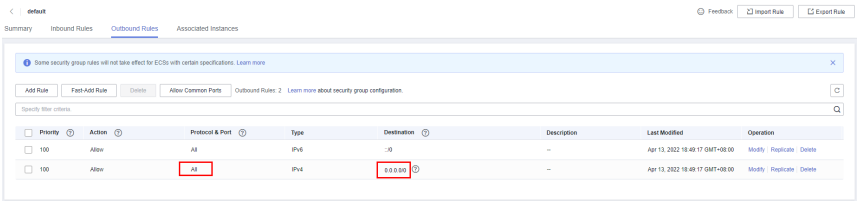
  Figure 1-15 Normal network connectivity

  ```
  [root@ecs-5a57 ~]# curl -kv 192.168.0.4:3306
  * Rebuilt URL to: 192.168.0.4:3306/
  *   Trying 192.168.0.4...
  * TCP_NODELAY set
  * Connected to 192.168.0.4 (192.168.0.4) port 3306 (#0)
  > GET / HTTP/1.1
  > Host: 192.168.0.4:3306
  > User-Agent: curl/7.61.1
  > Accept: */*
  >
  Warning: Binary output can mess up your terminal. Use "--output -" to tell
  Warning: curl to output it to your terminal anyway, or consider "--output
  Warning: <FILE>" to save to a file.
  * Failed writing body (0 != 5)
  * Closing connection 0
  [root@ecs-5a57 ~]#
  ```

- If no, check the security group rules.

  - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the floating IP address and port of the DB instance.

    Figure 1-16 ECS security group

  - If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS. For details, see **Configuring a Security Group Rule**.

5. Import the MySQL client installation package to the ECS using either of the following methods. Then go to **6**.

   Method 1:

   Download a MySQL client installation package for Linux using the ECS. To do so, you need to bind an EIP to the ECS.

   - MySQL 8.0:
     ```
     wget https://dev.mysql.com/get/mysql-community-client-8.0.28-1.el6.x86_64.rpm
     ```
   - MySQL 5.7:
     ```
     wget https://dev.mysql.com/get/mysql-community-client-5.7.38-1.el6.x86_64.rpm
     ```

   NOTE

   A MySQL client running a version later than that of the DB instance is recommended.

   Method 2:

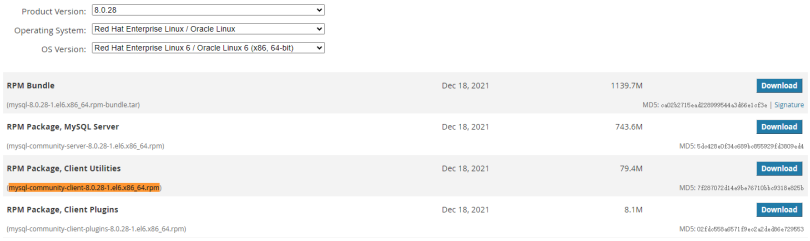
   Download a MySQL client installation package for Linux using a browser and upload the package to the ECS.

You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

- MySQL 8.0:

  Click **here** to download an installation package. **mysql-community-client-8.0.28-1.el6.x86_64.rpm** is used as an example.

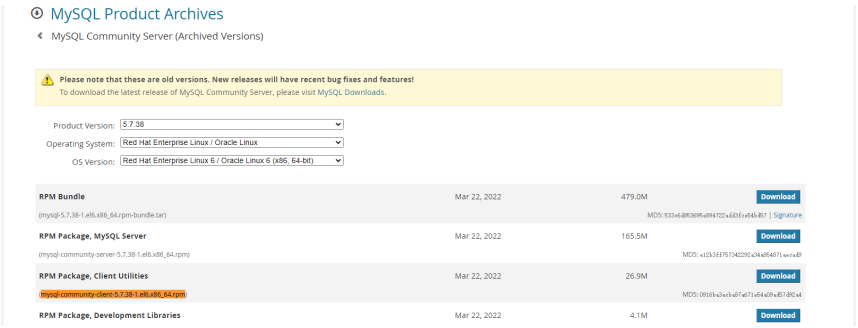
  Figure 1-17 Downloading a MySQL 8.0 installation package

- MySQL 5.7:

  Click **here** to download an installation package. **mysql-community-client-5.7.38-1.el6.x86_64.rpm** is used as an example.

  Figure 1-18 Downloading a MySQL 5.7 installation package

NOTE

A MySQL client running a version later than that of the DB instance is recommended.

6. Run the following command to install the MySQL client:
   - MySQL 8.0:
     ```
     rpm -ivh --nodeps mysql-community-client-8.0.28-1.el6.x86_64.rpm
     ```
   - MySQL 5.7:
     ```
     rpm -ivh --nodeps mysql-community-client-5.7.38-1.el6.x86_64.rpm
     ```

   NOTE

   - If any conflicts occur during the installation, add the **replacefiles** parameter to the command and install the client again.
     ```
     rpm -ivh --replacefiles mysql-community-client-installation_package_version-1.el6.x86_64.rpm
     ```
   - If a message is displayed prompting you to install a dependent package during the installation, add the **nodeps** parameter to the command and install the client again.
     ```
     rpm -ivh --nodeps mysql-community-client-installation_package_version-1.el6.x86_64.rpm
     ```

Step 3: Connect to the DB Instance Using a CLI

On the **Instances** page of the RDS console, click the instance name to go to the **Overview** page. Under **SSL**, check whether SSL is enabled.

- If it is disabled (default), use a **non-SSL connection**.
- If it is enabled, use an **SSL connection**. SSL encrypts connections to the instance, making in-transit data more secure.

Non-SSL Connection

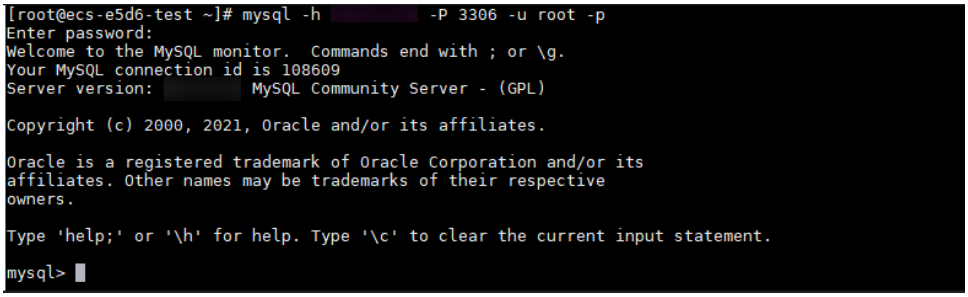
1. Run the following command on the ECS to connect to the DB instance:
   ```
   mysql -h <host> -P <port> -u <userName> -p
   ```
   Example:
   ```
   mysql -h 172.16.0.31 -P 3306 -u root -p
   ```

   Table 1-9 Parameter description

Parameter	Description
<*host*>	The floating IP address or private domain name of the DB instance, which can be obtained from **3**. **NOTE** If your DB instance is connected through a private domain name, changing its floating IP address does not interrupt services.
<*port*>	The database port of the DB instance, which can be obtained from **3**. The default value is **3306**.
<*userName*>	The administrator account **root**.
<*caName*>	The name of the CA certificate. The certificate should be stored in the directory where the command is executed.

2. Enter the password of the database account if the following information is displayed:
   ```
   Enter password:
   ```

   Figure 1-19 Successful connection

SSL Connection

1. On the **Instances** page of the RDS console, click the DB instance name.
2. Click **Download** under **SSL** to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.
3. Upload **ca.pem** to the ECS. It is recommended that the root certificate be stored in the same directory as the MySQL client installation package.

NOTE

- You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the root certificate.
- 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 **How Can I Identify the Validity Period of an SSL Root Certificate?**
- TLS v1.2 or later is recommended. Versions earlier than TLS v1.2 have security risks.
- **ca-bundle.pem** contains both the new certificate provided as of April 2017 and the old certificate.
- Both **ca.pem** and **ca-bundle.pem** can be used for SSL connections because **ca-bundle.pem** contains **ca.pem**.
- RDS for MySQL DB instances do not support X.509-based authentication.

4. Run the following command on the ECS to connect to the DB instance:

```
mysql -h <host> -P <port> -u <userName> -p --ssl-ca=<caName>
```

Example:

```
mysql -h 172.16.0.31 -P 3306 -u root -p --ssl-ca=ca.pem
```

Table 1-10 Parameter description

Parameter	Description
<*host*>	The floating IP address or private domain name of the DB instance, which can be obtained from 3. **NOTE** If your DB instance is connected through a private domain name, changing its floating IP address does not interrupt services.
<*port*>	The database port of the DB instance, which can be obtained from 3. The default value is **3306**.
<*userName*>	The administrator account **root**.
<*caName*>	The name of the CA certificate.

5. Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 1-20 Successful connection

```
[root@ecs-e5d6-test ~]# mysql -h [redacted] -P 3306 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 108609
Server version: [redacted] MySQL Community Server - (GPL)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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>
```

FAQs

What Should I Do If I Can't Connect to My RDS DB Instance?

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Creating a MySQL Database Using the Console**
- **Creating a MySQL Database Using an API**
- **Managing MySQL Databases Using DAS**
- **Migration Solution Overview**

1.3.3.2 Using MySQL CLI to Connect to an Instance Through a Public Network

If you cannot access your DB instance through a floating IP address, bind an EIP to the DB instance and connect to it through the EIP.

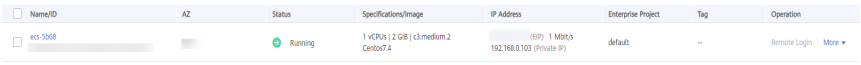
This section uses a Linux ECS as an example to describe how to connect a Linux ECS to a DB instance with SSL enabled through a public network. SSL encrypts connections to the DB instance, making in-transit data more secure.

You can also access your DB instance through Network Address Translation (NAT). If you have configured both NAT and EIP, the EIP is preferentially used.

Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Linux ECS, go to **3**.
   - If no ECS is available, go to **2**.

   Figure 1-21 ECS

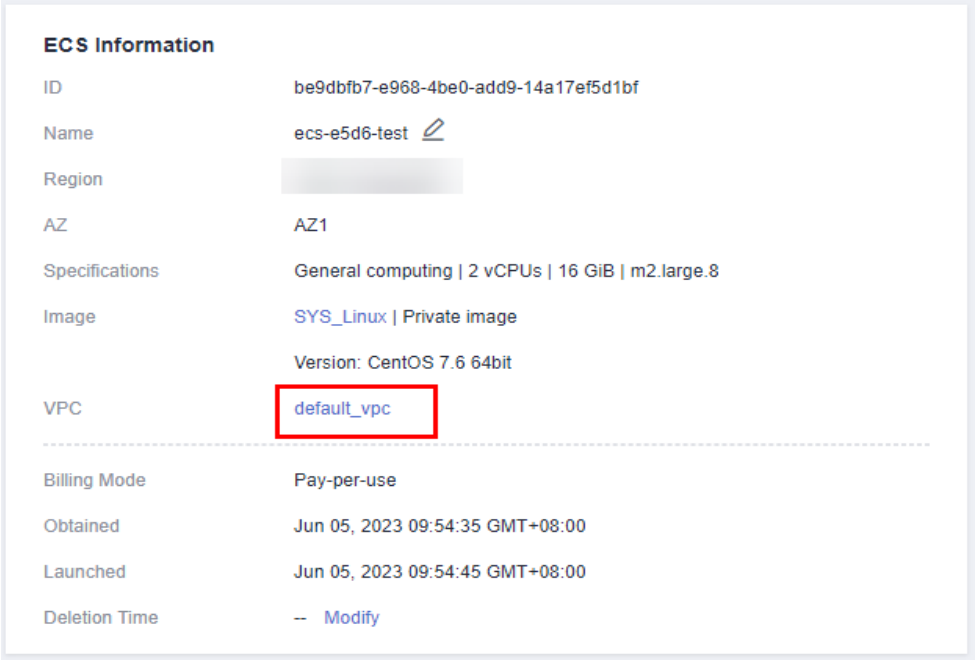
   

2. Buy an ECS and select Linux (for example, CentOS) as its OS.

   To download a MySQL client to the ECS, bind an EIP to the ECS.

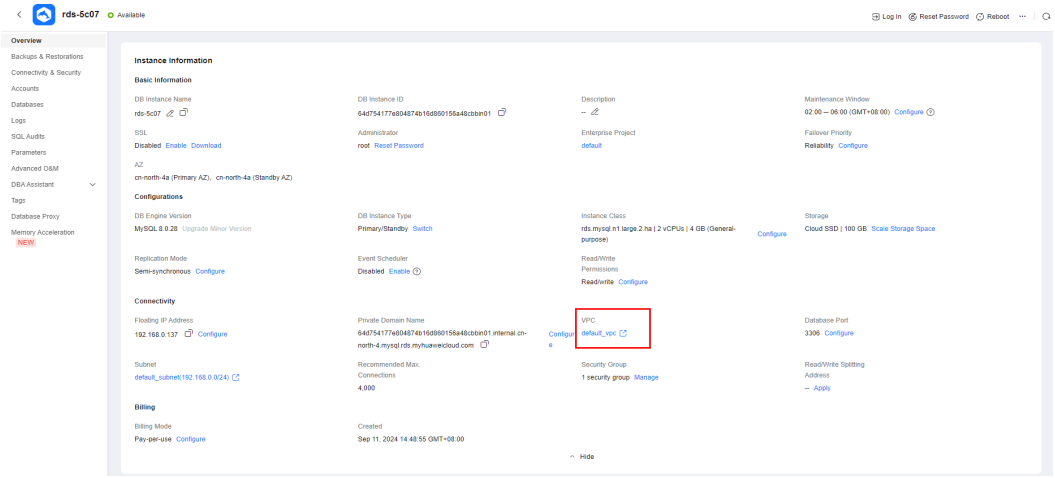
   For details about how to purchase a Linux ECS, see **Purchasing a Custom ECS** in *Elastic Cloud Server User Guide*.
3. On the **ECS Information** page, view the region and VPC of the ECS.

Figure 1-22 ECS information



4. On the **Overview** page of the RDS for MySQL instance, view the region and VPC of the DB instance.

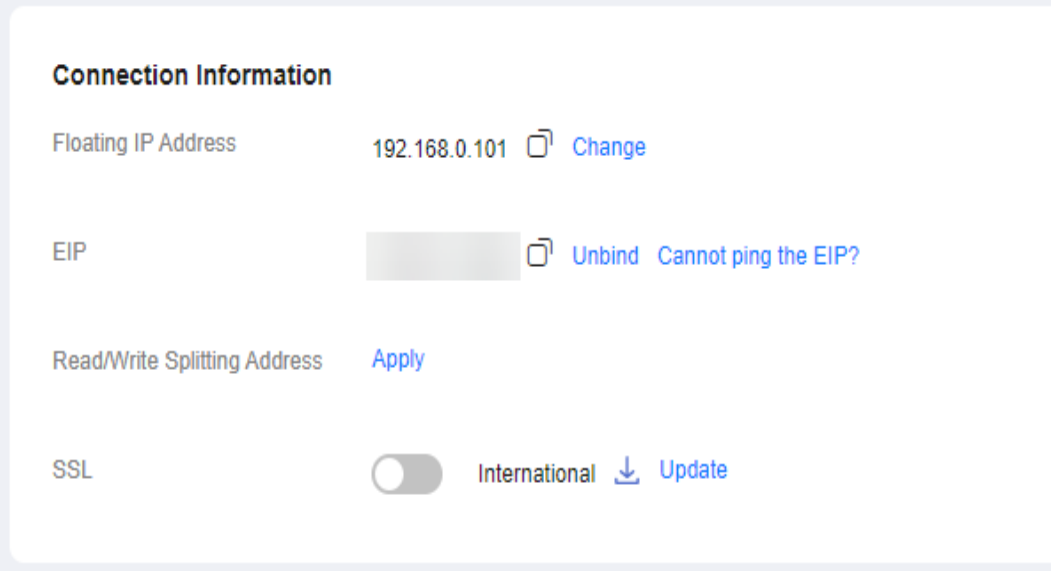
Figure 1-23 Overview



Step 2: Test Connectivity and Install a MySQL Client

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name.
3. Choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, obtain the EIP and database port of the DB instance.

Figure 1-24 Connection information



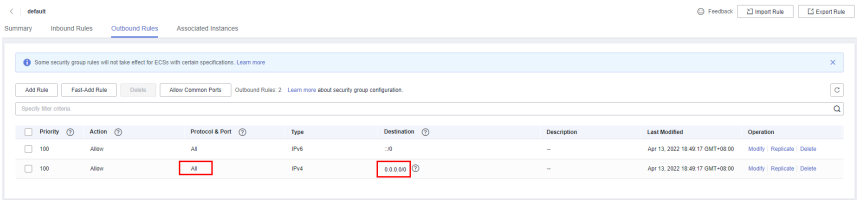
If no EIP has been bound to the DB instance, see **Binding an EIP**.

4. After logging in to the ECS, check whether the EIP and database port obtained in **3** can be connected.

```
curl -kv EIP.Port
```

   - If yes, network connectivity is normal. Go to **5**.
   - If no, check the security group rules.
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the EIP and port of the DB instance.

       Figure 1-25 ECS security group

     - If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS. For details, see **Configuring a Security Group Rule**.

5. Import the MySQL client installation package to the ECS using either of the following methods. Then go to **6**.

   Method 1:

   Download a MySQL client installation package for Linux using the ECS. To do so, you need to bind an EIP to the ECS.

   - MySQL 8.0:

```
wget https://dev.mysql.com/get/mysql-community-client-8.0.28-1.el6.x86_64.rpm
```

   - MySQL 5.7:


```
wget https://dev.mysql.com/get/mysql-community-client-5.7.38-1.el6.x86_64.rpm
```

NOTE

A MySQL client running a version later than that of the DB instance is recommended.

Method 2:

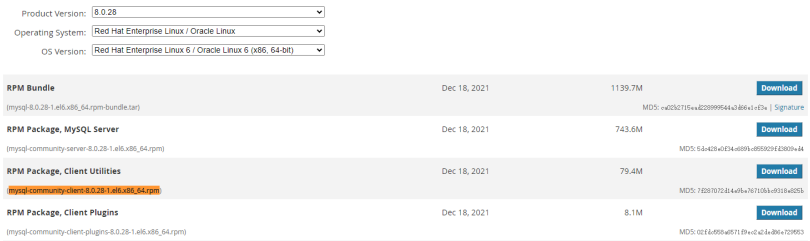
Download a MySQL client installation package for Linux using a browser and upload the package to the ECS.

You can use any terminal connection tool, such as WinSCP and PuTTY, to upload the installation package to the ECS.

- MySQL 8.0:

  Click **here** to download an installation package. **mysql-community-client-8.0.28-1.el6.x86_64.rpm** is used as an example.

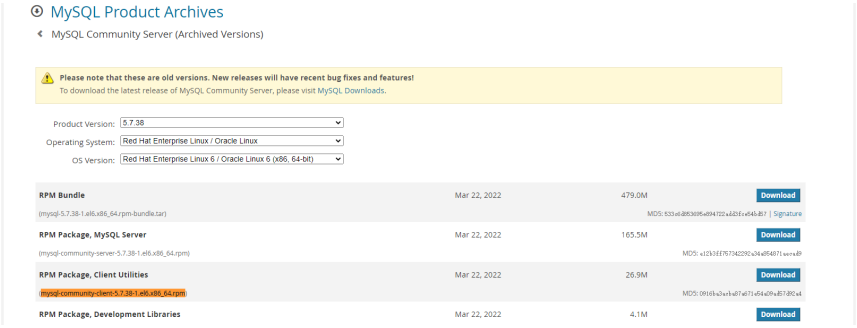
  Figure 1-26 Downloading a MySQL 8.0 installation package

- MySQL 5.7:

  Click **here** to download an installation package. **mysql-community-client-5.7.38-1.el6.x86_64.rpm** is used as an example.

  Figure 1-27 Downloading a MySQL 5.7 installation package

NOTE

A MySQL client running a version later than that of the DB instance is recommended.

6. Run the following command to install the MySQL client:
   - MySQL 8.0:
     ```
     rpm -ivh --nodeps mysql-community-client-8.0.28-1.el6.x86_64.rpm
     ```
   - MySQL 5.7:
     ```
     rpm -ivh --nodeps mysql-community-client-5.7.38-1.el6.x86_64.rpm
     ```


NOTE

- If any conflicts occur during the installation, add the **replacefiles** parameter to the command and install the client again.
  rpm -ivh --replacefiles mysql-community-client-*installation_package_version*-1.el6.x86_64.rpm
- If a message is displayed prompting you to install a dependent package during the installation, add the **nodeps** parameter to the command and install the client again.
  rpm -ivh --nodeps mysql-community-client-*installation_package_version*-1.el6.x86_64.rpm

Step 3: Connect to the DB Instance Using Commands (SSL Connection)

1. On the **Instances** page, click the DB instance name.
2. In the **Basic Information** area, check whether SSL is enabled.
   - If yes, go to **3**.
   - If no, click **Enable**. In the displayed dialog box, click **OK** to enable SSL. Then go to **3**.
3. Click **Download** under **SSL** to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.
4. Upload **ca.pem** to the ECS.

   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 **How Can I Identify the Validity Period of an SSL Root Certificate?**
   - TLS v1.2 or later is recommended. Versions earlier than TLS v1.2 have security risks.
   - **ca-bundle.pem** contains both the new certificate provided as of April 2017 and the old certificate.
   - Both **ca.pem** and **ca-bundle.pem** can be used for SSL connections because **ca-bundle.pem** contains **ca.pem**.
   - RDS for MySQL DB instances do not support X.509-based authentication.
5. Run the following command on the ECS to connect to the DB instance:
   mysql -h <*host*> -P <*port*> -u <*userName*> -p --ssl-ca=<*caName*>

   Example:

   ```
   mysql -h 172.16.0.31 -P 3306 -u root -p --ssl-ca=ca.pem
   ```

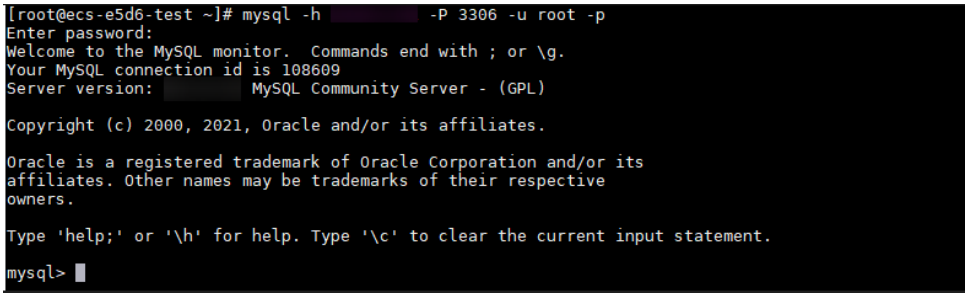

Table 1-11 Parameter description

Parameter	Description
<host>	The EIP or public domain name, which can be obtained from 3. **NOTE** To connect to a DB instance using a public domain name, apply for a public domain name first. For details, see **Applying for and Changing a Public Domain Name**. After a public domain name is generated, changing the EIP will interrupt database connections. Exercise caution when performing this operation.
<port>	The database port of the DB instance, which can be obtained from 3. The default value is **3306**.
<userName>	Administrator account **root**.
<caName>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

6. Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 1-28 Successful connection



FAQs

What Should I Do If I Can't Connect to My RDS DB Instance?

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Creating a MySQL Database Using the Console**
- **Creating a MySQL Database Using an API**
- **Managing MySQL Databases Using DAS**
- **Migration Solution Overview**

1.3.3.3 Installing a MySQL Client

MySQL provides client installation packages for different OSs on its official website. MySQL 5.6 is used here as an example. You can download the **latest**

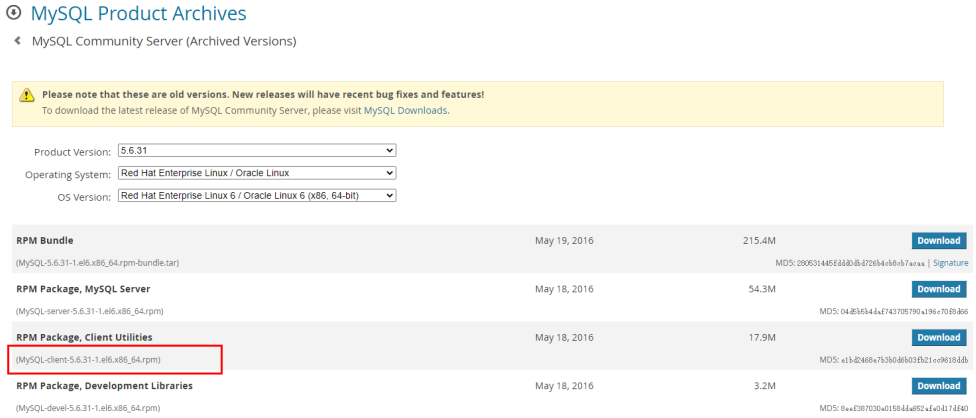
version or **any other version** for your project. The following procedure illustrates how to obtain the required installation package and install the MySQL client into a Red Hat Linux system.

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 1-29 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 dependent package during the installation, add the **nodeps** parameter to the command and install the client again.
  ```
  rpm -ivh --nodeps MySQL-client-5.6.31-1.el6.x86_64.rpm
  ```

----End

1.3.4 Connecting to an RDS for MySQL Instance Through MySQL-Front

MySQL-Front is a Windows front end for MySQL databases. It allows you to interact with MySQL databases through a GUI, including connecting to a database, running SQL commands, and managing tables and records.

This section uses MySQL-Front 5.4 as an example to describe how to use MySQL-Front to connect to an RDS for MySQL instance through an EIP.

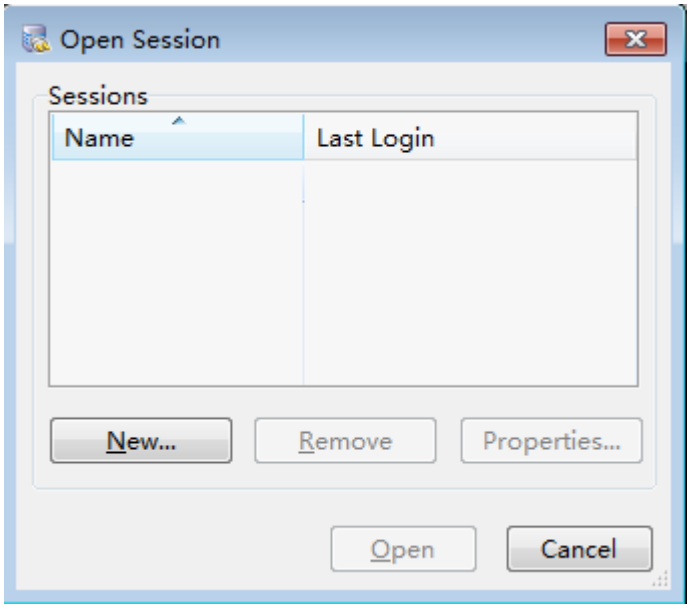
Procedure

Step 1 Download and install MySQL-Front (for example, MySQL-Front 5.4).

Step 2 Start the MySQL-Front.

Step 3 In the displayed dialog box, click **New**.

Figure 1-30 Connection management



Step 4 Enter the information of the DB instance to be connected and click **Ok**, as shown in **Figure 1-31**.

Figure 1-31 Adding an account

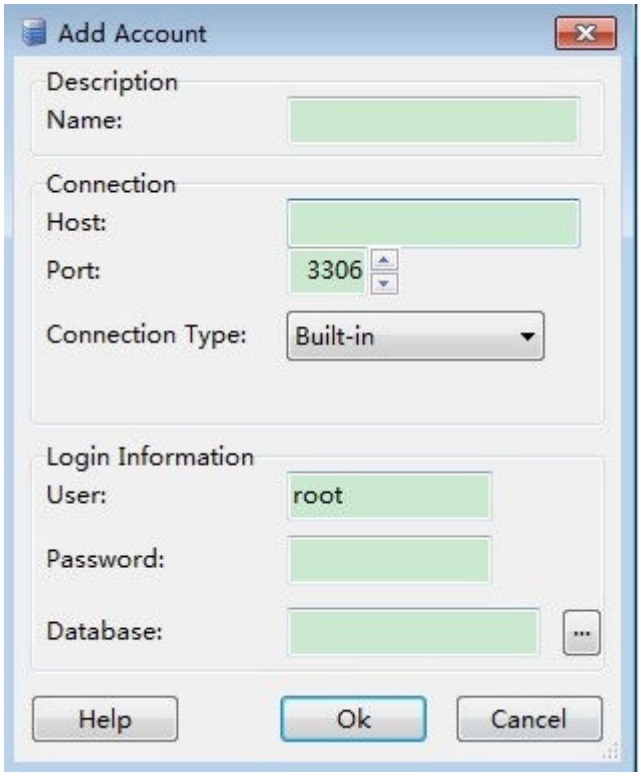
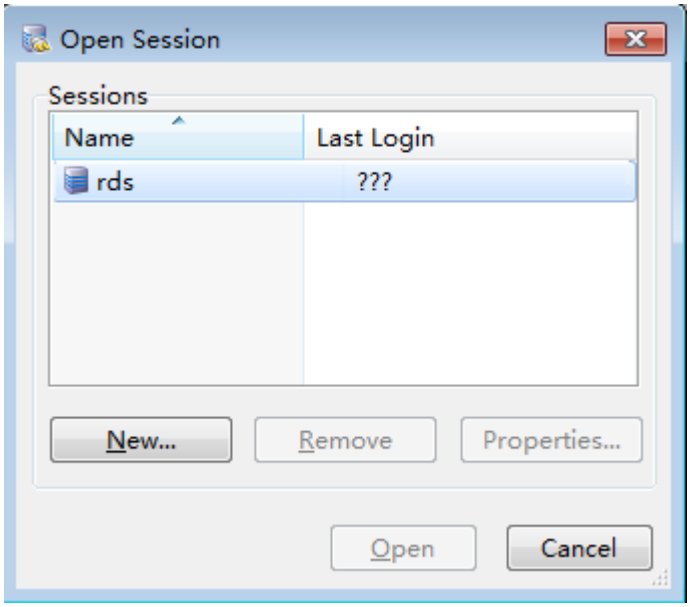


Table 1-12 Parameter description

Parameter	Description
Name	Specify a name for the database connection task. If you do not specify this parameter, it will be the same as that configured for **Host** by default.
Host	Enter the EIP bound to the DB instance. To obtain the EIP: 1. Click the instance name to enter the **Overview** page. 2. In the navigation pane, click **Connectivity & Security** and check the EIP.
Port	Enter the database port of the DB instance. To obtain the port: 1. Click the instance name to enter the **Overview** page. 2. In the navigation pane, click **Connectivity & Security** and check the database port.
User	Enter the username used for accessing the instance. The default user is **root**.
Password	Enter the password of the username.

Step 5 In the displayed window, select the connection that you have created in **Step 4** and click **Open**. If the connection information is correct, the DB instance will be connected.

Figure 1-32 Opening a session



NOTE

If the connection fails, see **What Should I Do If an ECS Cannot Connect to an RDS DB Instance?**

----End

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Creating a MySQL Database Using the Console**
- **Creating a MySQL Database Using an API**
- **Managing MySQL Databases Using DAS**
- **Migrating MySQL Databases Using DRS**
- **Migrating Data to RDS for MySQL Using mysqldump**

1.3.5 Connecting to an RDS for MySQL Instance Through JDBC

If you are connecting to an instance through JDBC, an SSL certificate is optional, but using an SSL certificate can improve the security of your data. SSL is disabled by default for newly created instances. You can enable SSL by referring to **Configuring an SSL Connection**. SSL encrypts connections to your instances but it increases the connection response time and CPU usage. For this reason, enabling SSL is not recommended.

Prerequisites

You are familiar with:

- Computer basics.
- Java.
- JDBC.

Connection with the SSL Certificate

NOTE

Download the SSL certificate and verify it before connecting to your instance. RDS for MySQL DB instances do not support X.509-based authentication.

Step 1 Download the CA certificate or certificate bundle.

1. On the **Instances** page, click the instance name to go to the **Overview** page.
2. Under **SSL**, click **Download**.

Step 2 Use keytool to generate a truststore file using the CA certificate.

```
<keytool installation path> ./keytool.exe -importcert -alias <MySQLCACert> -file <ca.pem> -keystore <truststore_file> -storepass <password>
```

Table 1-13 Parameter description

Parameter	Description
<keytool installation path>	Bin directory in the JDK or JRE installation path, for example, **C:\Program Files (x86)\Java\jdk11.0.7\bin**.
<MySQLCACert>	Name of the truststore file. Set it to a name specific to the service for future identification.
<ca.pem>	Name of the CA certificate downloaded and decompressed in **Step 1**, for example, ca.pem.
<truststore_file>	Path for storing the truststore file.
<password>	Password of the truststore file.

Code example (using keytool in the JDK installation path to generate the truststore file):

```
Owner:  CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Issuer: CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Serial number: 1
Valid from: Thu Feb 16 11:42:43 EST 2017 until: Sun Feb 14 11:42:43 EST 2027
Certificate fingerprints:
    MD5: 18:87:97:37:EA:CB:0B:5A:24:AB:27:76:45:A4:78:C1
    SHA1: 2B:0D:D9:69:2C:99:BF:1E:2A:25:4E:8D:2D:38:B8:70:66:47:FA:ED
    SHA256:C3:29:67:1B:E5:37:06:F7:A9:93:DF:C7:B3:27:5E:09:C7:FD:EE:2D:18:86:F4:9C:40:D8:26:CB:DA:95:
A0:24
    Signature algorithm name: SHA256withRSA Subject Public Key Algorithm: 2048-bit RSA key
    Version: 1
    Trust this certificate? [no]: y
    Certificate was added to keystore
```


Step 3 Connect to your RDS for MySQL instance through JDBC.

jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2

Table 1-14 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** • If you are accessing the DB instance through an ECS, *instance_ip* is the floating IP address of the instance. You can obtain this IP address on the **Connectivity & Security** page. • If you are accessing the DB instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can obtain this IP address on the **Connectivity & Security** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can obtain this port number on the **Connectivity & Security** page.
<database_name>	Database name used for connecting to the DB instance. The default value is **mysql**.
<param1>	**requireSSL**, indicating whether the server supports SSL. Its value can be either of the following: • **true**: The server supports SSL. • **false**: The server does not support SSL. **NOTE** For details about the relationship between **requireSSL** and **sslmode**, see **Table 1-15**.
<param2>	**useSSL**, indicating whether the client uses SSL to connect to the server. Its value can be either of the following: • **true**: The client uses SSL to connect to the server. • **false**: The client does not use SSL to connect to the server. **NOTE** For details about the relationship between **useSSL** and **sslmode**, see **Table 1-15**.
<param3>	**verifyServerCertificate**, indicating whether the client verifies the server certificate. It can be either of the following: • **true**: The client verifies the server certificate. • **false**: The client does not verify the server certificate. **NOTE** For details about the relationship between **verifyServerCertificate** and **sslmode**, see **Table 1-15**.

Parameter	Description
<param4>	**trustCertificateKeyStoreUrl**. Its value is **file:<truststore_file>**. Replace **<truststore_file>** with the path for storing the truststore file set in **Step 2**.
<param5>	**trustCertificateKeyStorePassword**. Its value is the password of the truststore file set in **Step 2**.

Table 1-15 Relationship between connection parameters and sslmode

useSSL	requireSSL	verifyServerCer-tificate	sslMode
false	N/A	N/A	DISABLED
true	false	false	PREFERRED
true	true	false	REQUIRED
true	N/A	true	VERIFY_CA

Code example (Java code for connecting to an RDS for MySQL instance):

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;

// There will be security risks if the username and password used for authentication are directly written into code. Store the username and password in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables. Before running this example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.
public class JDBCTest {


    String USER = System.getenv("EXAMPLE_USERNAME_ENV");
    String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;

       // Set the required parameters in the URL based on the site requirements.
       String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
param1=value1&param2=value2";

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection(url, USER, PASS);

            stmt = conn.createStatement();
            String sql = "show status like 'ssl%'";
            ResultSet rs = stmt.executeQuery(sql);

            int columns = rs.getMetaData().getColumnCount();
            for (int i = 1; i <= columns; i++) {
```



```
            System.out.print(rs.getMetaData().getColumnName(i));
            System.out.print("\t");
        }

        while (rs.next()) {
            System.out.println();
            for (int i = 1; i <= columns; i++) {
                System.out.print(rs.getObject(i));
                System.out.print("\t");
            }
        }
        rs.close();
        stmt.close();
        conn.close();
    } catch (SQLException se) {
        se.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        // release resource ....
    }
  }
}
```

----End

Connection Without the SSL Certificate

NOTE

You do not need to download the SSL certificate because certificate verification on the server is not required.

Step 1 Connect to the RDS for MySQL DB instance through JDBC.

jdbc:mysql://*<instance_ip>*:*<instance_port>*/*<database_name>***?useSSL=**false

Table 1-16 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** • If you are accessing the DB instance through an ECS, *instance_ip* is the floating IP address of the instance. You can obtain this IP address on the **Connectivity & Security** page. • If you are accessing the DB instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can obtain this IP address on the **Connectivity & Security** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can obtain this port number on the **Connectivity & Security** page.
<database_name>	Database name used for connecting to the DB instance. The default value is **mysql**.

Code example (Java code for connecting to an RDS for MySQL instance):


```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
// There will be security risks if the username and password used for authentication are directly written into code. Store the username and password in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables. Before running this example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;

            // Set the required parameters in the URL based on the site requirements.
            String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
param1=value1&param2=value2";
            String USER = System.getenv("EXAMPLE_USERNAME_ENV");
            String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");
        try {
            Class.forName("com.mysql.jdbc.Driver");

                    conn = DriverManager.getConnection(url,USER,PASS);
            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

Related Issues

- Symptom

  When you use JDK 8.0 or a later version to connect to an RDS for MySQL instance with an SSL certificate downloaded, an error similar to the following is reported:

```
javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or
cipher suites are inappropriate)
    at sun.security.ssl.HandshakeContext.<init>(HandshakeContext.java:171) ~[na:1.8.0_292]
    at sun.security.ssl.ClientHandshakeContext.<init>(ClientHandshakeContext.java:98) ~
[na:1.8.0_292]
    at sun.security.ssl.TransportContext.kickstart(TransportContext.java:220) ~
[na:1.8.0_292]
    at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:428) ~
[na:1.8.0_292]
    at
com.mysql.cj.protocol.ExportControlled.performTlsHandshake(ExportControlled.java:316) ~
[mysql-connector-java-8.0.17.jar:8.0.17]
    at
com.mysql.cj.protocol.StandardSocketFactory.performTlsHandshake(StandardSocketFactory.java
:188) ~[mysql-connector-java8.0.17.jar:8.0.17]
    at
com.mysql.cj.protocol.a.NativeSocketConnection.performTlsHandshake(NativeSocketConnection.
java:99) ~[mysql-connector-java8.0.17.jar:8.0.17]
```



```
    at
com.mysql.cj.protocol.a.NativeProtocol.negotiateSSLConnection(NativeProtocol.java:331) ~
[mysql-connector-java8.0.17.jar:8.0.17]
... 68 common frames omitted
```

- Solution

  Specify the corresponding parameter values in the code link of **Step 3** based on the JAR package used by the client. Example:

  - mysql-connector-java-5.1.*xx*.jar (For 8.0.18 and earlier versions, use the **enabledTLSProtocols** parameter. For details, see **Connecting Securely Using SSL**.)
    In the database connection URL jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2, replace **param1=value1** with **enabledTLSProtocols=TLSv1.2**.
  - mysql-connector-java-8.0.*xx*.jar (For versions later than 8.0.18, use the **tlsVersions** parameter.)
    In the database connection URL jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2, replace **param1=value1** with **tlsVersions=TLSv1.2**.

1.3.6 Connection Management

1.3.6.1 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 read/write splitting is enabled, the floating IP addresses of primary DB instances and read replicas cannot be changed.

Changing the floating IP address will interrupt the database connection. You are advised to change a floating IP address during off-peak hours.

Only floating IPv4 addresses can be changed.

Procedure

When you buy a DB instance, select a VPC and subnet on the **Buy DB Instance** page. Then, a floating IP address will be automatically assigned to your instance.

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.

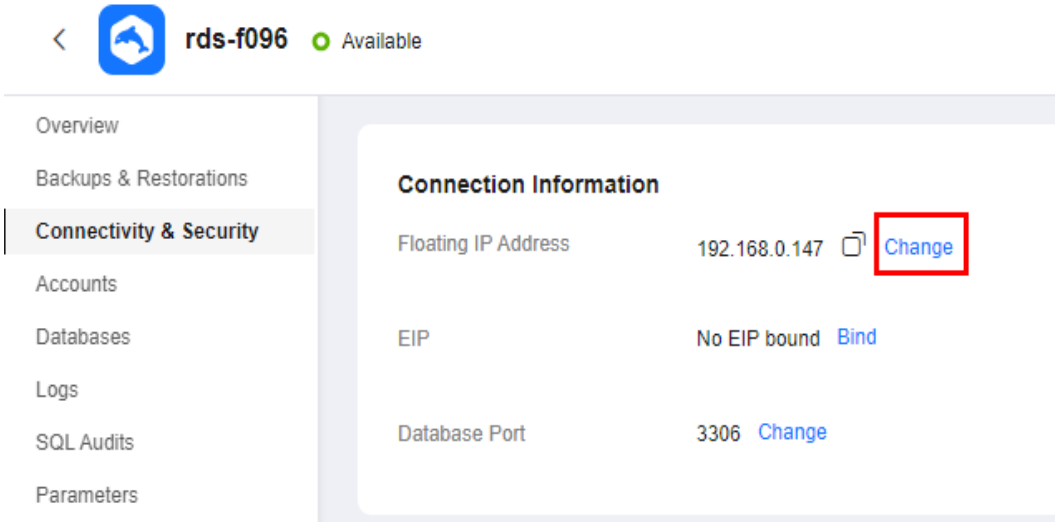
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **Floating IP Address**, click **Configure**.

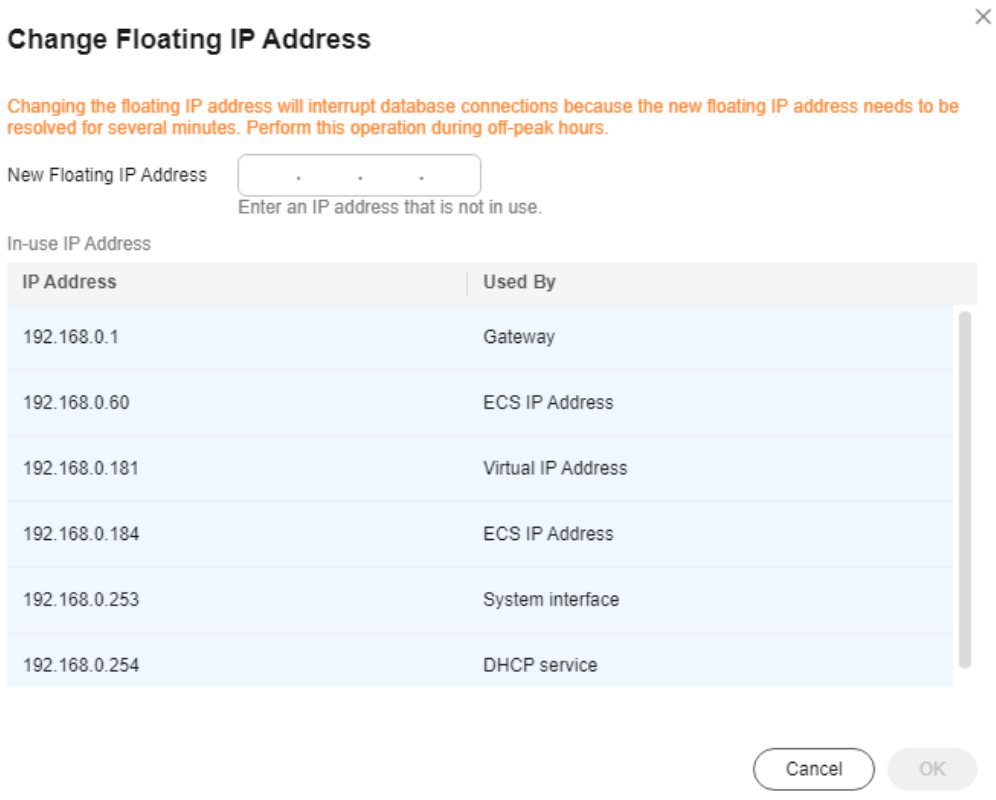
Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Change** next to the **Floating IP Address** field.

Figure 1-33 Floating IP address



Step 6 In the displayed dialog box, check the number of in-use IP addresses. If the in-use IP addresses are less than 254, there are unused floating IP addresses.

Figure 1-34 Changing a floating IP address



Step 7 Enter an available IP address and click **OK**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

Step 8 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.3.6.2 Changing a Private Domain Name

You can connect to RDS DB instances through private domain names.

Constraints

- Changing the private domain name will interrupt your database connection. To reconnect to the instance, change the connection address of your applications. The new private domain name is applied to the instance about 5 minutes after the change.
- If your DB instance is connected through a private domain name, changing its floating IP address does not interrupt services.

Procedure

When you buy a DB instance, the system automatically assigns a private domain name to your instance.

After the DB instance is created, you can change the private domain name as needed.

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **Private Domain Name**, click **Configure**.

Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Change** next to the **Private Domain Name** field.

Step 6 In the displayed dialog box, enter a new domain name and click **Yes**.

NOTE

- Only the prefix of a private domain name can be modified.
- The prefix of a private domain name contains 8 to 63 characters, and can include only letters and digits.
- The new private domain name must be different from existing ones.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.3.6.3 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.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

Constraints

After read/write splitting is enabled, the database ports of primary DB instances and read replicas cannot be changed.

Changing the database port of a DB instance will cause the instance to reboot.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Overview** page, find **Database Port** and click **Configure** under it.

Alternatively, choose **Connectivity & Security** in the navigation pane on the left. On the displayed page, click **Change** next to the **Database Port** field.

NOTE

RDS for MySQL instances can use database ports 1024 to 65535, excluding 12017, 33071, and 33062, which are reserved for RDS system use.

- In the displayed dialog box, enter a new port and click **Yes**.

  If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

  Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

  - 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.
  - If you change the database port of a read replica, the change will not affect other DB instances. Only the read replica will reboot.
  - This process takes about 1–5 minutes.
- In the displayed dialog box, click **No** to cancel the modification.

Step 6 View the result on the **Overview** page.

----End

1.3.6.4 Binding and Unbinding an EIP

Scenarios

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.

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, add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see **Configuring a Security Group Rule**.
- You can buy an EIP on the **network console** and bind it to a DB instance. One EIP can be bound to only one DB instance. For pricing details, see **Elastic IP pricing details**.

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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Bind** in the connection topology.

Step 6 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **Yes**.

Step 7 On the **Connectivity & Security** 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance that has an EIP bound.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** in the connection topology. In the displayed dialog box, click **Yes**.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 On the **Connectivity & Security** page, view the results.

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

1.3.6.5 Applying for and Changing a Public Domain Name

You can apply for a public domain name for your DB instance after binding an EIP to it, and connect to the instance through the public domain name.

Constraints

- To apply for or change a public domain name, you need to contact customer service to apply for required permissions.
- Before applying for a public domain name, you need to bind an EIP to your instance.
- After a public domain name is generated, changing the EIP will interrupt database connections. Exercise caution when performing this operation.

Applying for a Public Domain Name

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Apply** next to the **Public Name** field.

----End

Changing a Public Domain Name

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Change** next to the **Public Name** field.

Step 6 In the displayed dialog box, enter a new public domain name. Click **OK**.

NOTE

- Only the prefix of a public domain name can be modified.
- The prefix of a public domain name can contain 8 to 63 characters, and can include only letters and digits.
- The new public domain name must be different from existing ones.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.3.6.6 Configuring a Certificate

RDS allows you to reset and download a certificate.

Contact customer service to apply for the required permissions.

Resetting a Certificate

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 Under **SSL**, click **Update**.

Alternatively, choose **Connectivity & Security** in the navigation pane on the left. In the **Connection Information** area, click **Update** next to the **SSL** field.

Step 6 In the displayed dialog box, select the target certificate and click **OK**.

NOTE

Updating a certificate will cause the DB instance to reboot.

Step 7 View the update result on the **Overview** page.

----End

Downloading a Certificate

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 Under **SSL**, click **Download** to download the root certificate or certificate bundle.

Alternatively, choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, click next to 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 **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.
- TLS v1.2 or later is recommended. Versions earlier than TLS v1.2 have security risks.

----End

1.3.6.7 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.

For security, you need to create security group rules to allow specific IP addresses and ports to access your RDS DB instance.

- 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.
- When you attempt to connect to an RDS DB instance through a private network, check whether the ECS and 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.

This section describes how to configure an inbound rule for an RDS DB instance.

For details about the requirements of security group rules, see the **Adding a Security Group Rule** section in the *Virtual Private Cloud User Guide*.

Precautions

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are 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 100 security groups in your cloud account.
- By default, you can add up to 50 security group rules to a security group.
- One RDS instance can be associated with multiple security groups, and one security group can be associated with multiple RDS instances.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for a 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

To ensure the security of your data and DB instances, you are advised to use the principle of least privilege for database access. Change the default database port **3306**, and set the IP address to the remote server's address or the remote server's smallest subnet address to control the access from the remote server.

The default value of **Source** is **0.0.0.0/0**, indicating that RDS DB instances in the security group can be accessed from any IP address.

For details about the requirements of security group rules, see **Adding a Security Group Rule** in the *Virtual Private Cloud User Guide*.

Procedure

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

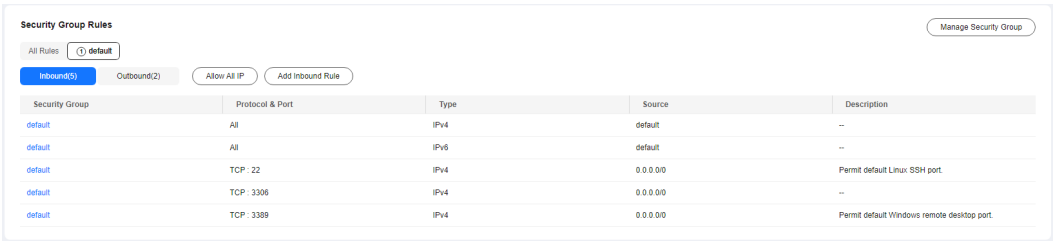
Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Connectivity & Security**. In the **Security Group Rules** area, click the security group name to view the security group rules.

Figure 1-35 Security group rules



Step 6 Click **Add Inbound Rule** or **Allow All IP** to configure security group rules.

To add more inbound rules, click ⊕.

NOTE

Allow All IP allows all IP addresses to access RDS DB instances in the security group, which poses high security risks. Exercise caution when performing this operation.

Figure 1-36 Adding an inbound rule

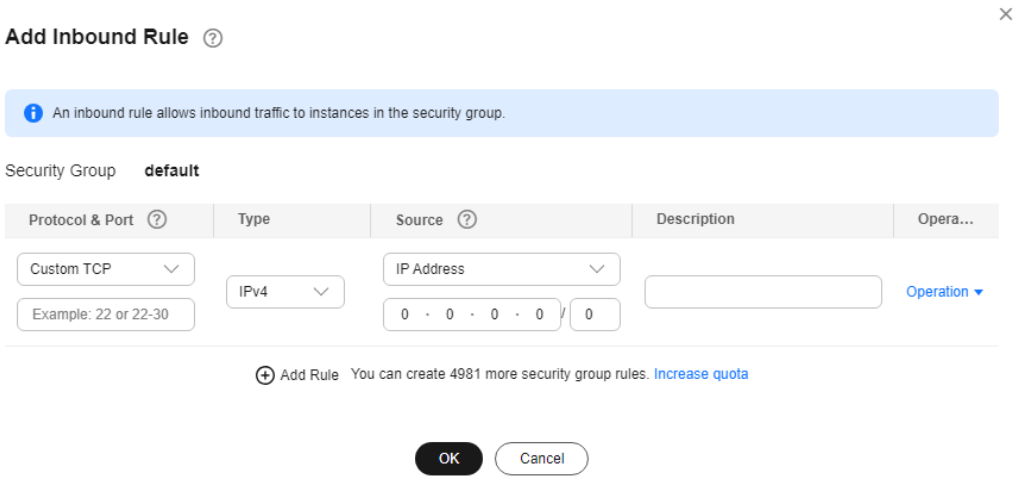


Table 1-17 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	**Protocol**: network protocol. Available options: **All ports**, **Custom TCP**, **Custom UDP**, **ICMP**, and **GRE**.	Custom TCP

Parameter	Description	Example Value
	Port: the port over which the traffic can reach your DB instance. RDS for MySQL instances can use database ports 1024 to 65535, excluding 12017 and 33071, which are reserved for RDS system use.	3306
Type	IP address type. • IPv4 • IPv6	IPv4
Source	Source address. It can be a single IP address, an IP address group, or a security group to allow access from them to your DB instance. Examples: • Single IP address: 192.168.10.10/32 (IPv4); 2002:50::44/128 (IPv6) • All IP addresses: 0.0.0.0/0 (IPv4); ::/0 (IPv6) • IP address range: 192.168.1.0/24 (IPv4); 2407:c080:802:469::/64 (IPv6) • Security group: default_securitygroup	0.0.0.0/0
Description	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

1.4 Database Usage

1.4.1 Suggestions on Using RDS for MySQL

1.4.1.1 Instance Usage Suggestions

DB Instances

DB Instance Types

- Primary/Standby
  - A primary/standby pair provides an HA architecture. It is suitable for production databases of large and medium enterprises in the Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other sectors.
  - When a primary instance is being created, a standby instance is provisioned along with it to provide data redundancy. The standby instance is invisible to you after being created.
  - If a failover occurs due to a primary instance failure, your database client will be disconnected from the instance briefly and then reconnects to the instance.
- Single-node
  - A single-node architecture is more cost-effective than primary/standby pairs.
  - It is recommended for development and testing of microsites, and small- and medium-sized enterprises, or for learning about RDS.
  - If a fault occurs on a single instance, the instance cannot recover in a timely manner.
- Read replica

  Read replicas include single read replicas and HA read replicas.
  - Single read replica

    If you want to use single read replicas, you are advised to buy more than one single read replica and enable database proxy. By doing so, the database proxy can route traffic to other read replicas if a read replica is faulty.
  - HA read replica

    If the physical server where the primary read replica is deployed fails, the standby read replica automatically takes over workloads.

  When you purchase a read replica, select the same value for **Table Name** as that of the DB instance.

  Recommendations for using read replicas:

  a. Configure HA read replicas if you plan to configure no more than two read replicas for a DB instance.

  b. If your DB instance is associated with more than two read replicas, enable database proxy for cost-effectiveness.

> **NOTE**
>
> If the replication between a read replica (single or HA) and the DB instance is abnormal, it can take a long time to rebuild and restore the read replica (depending on the data volume).
>
> After a read replica is created, you can **change its time zone** by adjusting the **time_zone** parameter. Ensure that the read replica uses the same time zone as the primary instance to avoid data synchronization errors.

Instance Classes

- Dedicated

  The instance has dedicated CPU and memory resources to ensure stable performance. The performance of a dedicated instance is never affected by other instances on the same physical machine. This instance class is good when performance stability is important.
- General-purpose

  CPU resources are shared with other general-purpose DB instances on the same physical machine. CPU usage is maximized through resource overcommitment. This instance class is a cost-effective option and suitable for scenarios where performance stability is not critical.

Database Connection

- Configure RDS for MySQL parameters that match your workloads.
- Keep an appropriate number of active connections.
- Periodically release persistent connections because maintaining them may require a large cache and use up memory.

Reliability and Availability

- Select primary/standby DB instances for production databases.
- Deploy primary and standby instances in different AZs.
- Create read replicas and enable read/write splitting for workloads involving frequent read/write operations.
- Change instance classes during off-peak hours.
- Select an instance class and storage space appropriate to your workloads.
- After scaling up your DB instance, scale up its read replicas in a timely manner to prevent service exceptions caused by insufficient storage of read replicas.

Backup and Restoration

- Backups may fail if the instance has a heavy workload during peak hours. To prevent this, create manual backups during off-peak hours and tailor the time window for automated backups to your workload requirements.
- Set the backup cycle to **All** for write-intensive DB instances.
- Configure a backup retention period suited to your workload. The default value is 7 days.
- Set the local retention period of binlogs as required. The default value is **0**, indicating that local binlogs are deleted once they are successfully backed up to OBS.

- Before restoring tables to a specified point in time, check whether any large tables without primary keys were deleted before the selected point in time. If yes, it is difficult to estimate when the restoration can be complete.
- Select the right storage type before creating a DB instance. DB instances using local SSDs cannot be restored to existing instances.
- If a DB instance is deleted, its automated full backups and binlog backups are also deleted. Create a manual backup of all data before deleting a DB instance.
- Configure a custom recycling policy to ensure that any instances that are deleted by mistake can be rebuilt.

SQL Audit

- Enable **Audit Logging** when periodic audits are required.
- Enable SQL Explorer when SQL analysis is required.

Routine O&M

- Periodically check slow query logs and error logs to identify problems in advance.
- Periodically check the resource usage of DB instances. If the resources are insufficient, scale up the resources in a timely manner.
- Monitor instance metrics. If any metric is beyond its expected range, address related issues as soon as possible.
- Before deleting or modifying a record, run SELECT to check that it is the one you desire.

Security

- Prevent your database from being directly accessed from the Internet. If you want to allow access from the Internet, bind an EIP to your DB instance and configure a whitelist.
- Use SSL to connect to your DB instance.

1.4.1.2 Database Usage Suggestions

Database Naming

- The names of database objects like databases, tables, and columns should be in lowercase. Different words in the name are separated with underscores (_).
- Reserved words and keywords cannot be used to name database objects in RDS for MySQL.
  - Reserved words and keywords for MySQL 8.0: **https://dev.mysql.com/doc/refman/8.0/en/keywords.html**
  - Reserved words and keywords for MySQL 5.7: **https://dev.mysql.com/doc/refman/5.7/en/keywords.html**
- In addition to those of MySQL 8.0 Community Edition, some other keywords and reserved words are added to RDS for MySQL. Do not use such words to name objects.

Table 1-18 New reserved words of RDS for MySQL

Reserved Word	Scenario
RECYCLE_BIN	Recycle bin

- Each database object name must be explainable and contain a maximum of 32 characters.
- Each temporary table in databases is prefixed with **tmp** and suffixed with a date.
- Each backup table in databases is prefixed with **bak** and suffixed with a date.
- All columns storing the same data in different databases or tables must have the same name and be of the same type.

Database Design

- All tables use the InnoDB storage engine unless otherwise specified. InnoDB supports transactions and row locks. It delivers excellent performance, making it easy to recover data.
- Databases and tables all use the UTF8 character set to avoid characters getting garbled by character set conversion.
- All tables and fields require comments that can be added using the COMMENT clause to maintain the data dictionary from the beginning of the design.
- The length of a single row in the table cannot exceed 1024 bytes.
- To avoid cross-partition queries, RDS for MySQL partitioned tables are not recommended. Cross-partition queries will decrease the query efficiency. A partitioned table is logically a single table, but the data is actually stored in multiple different files.
- Do not create too many columns in one table. Store cold and warm data separately to reduce the width of a table. In doing so, more rows of data can be stored in each memory page, decreasing disk I/O and making more efficient use of the cache.
- Columns that are frequently used together should be in the same table to avoid JOIN operations.
- Do not create reserved fields in a table. Otherwise, modifying the column type will lock the table, which has a greater impact than adding a field.
- Do not store binary data such as images and files in databases.
- Full-text indexes are not recommended because there are many limitations on full-text indexes for MySQL Community Edition.
- Create no more than 20,000 tables in an instance.
- Do not maintain your client connection to an instance for more than 8 hours.
- To prevent out of memory (OOM) exceptions from occurring when your instance handles a large number of concurrent requests, set **tmp_table_size**, **innodb_buffer_pool_size**, **max_connections**, **sort_buffer_size**, **read_buffer_size**, **read_rnd_buffer_size**, **join_buffer_size**, **thread_stack**, and **binlog_cache_size** to the values not exceeding their default values.

Field Design

- Ensure that each table contains no more than 50 fields.
- Select a small data type for each column as much as possible. Numeric data is preferred, followed by dates or binary data, and the least preferred is characters. The larger the column data type, the more the space required for creating indexes. As a result, there are fewer indexes on a page and more I/O operations required, so database performance deteriorates.
- If the integer type is used as the database field type, select the shortest column type. If the value is a non-negative number, it must be the unsigned type.
- Each field should have the NOT NULL attribute. The default value for the numeric type such as INT is recommended to be **0**, and that for the character type such as VARCHAR is recommended to be an empty string.
- Do not use the ENUM type. Instead, use the TINYINT type.

  Change ENUM values using ALTER. The ORDER BY operations on ENUM values are inefficient and require extra operations.

  If you have specified that ENUM values cannot be numeric, other data types (such as char) can be used.
- If the numeric data type is required, use DECIMAL instead of FLOAT or DOUBLE.

  FLOAT and DOUBLE data cannot be stored precisely, and value comparison results may be incorrect.
- When you want to record a date or specific time, use the DATETIME or TIMESTAMP type instead of the string type.
- Store IP addresses using the INT UNSIGNED type. You can convert IP addresses into numeric data using function inet_aton or inet_ntoa.
- The VARCHAR data should be as short as possible. Although the VARCHAR data varies in length dynamically on disks, it occupies the maximum length in memory.
- Use VARBINARY to store variable-length character strings that are case-sensitive. VARBINARY is case-sensitive by default and quick to process because no character sets are involved.

Index Design

- Create a primary key for each InnoDB table. Neither use a frequently-updated column as the primary key nor a multi-column primary key. Do not use the UUID, MD5, or character string column as the primary key. Use a column whose values can increment continuously as the primary key. So, the auto-increment ID column is recommended.
- Use no more than 5 indexes in a single table. Indexes speed up queries, but too many indexes may slow down writes. Inappropriate indexes sometimes reduce query efficiency.
- Do not create an independent index for each column in a table. A well-designed composite index is much more efficient than a separate index on each column.
- Create an index on the following columns:

  - Columns specified in the WHERE clause of SELECT, UPDATE, or DELETE statements
  - Columns specified in ORDER BY, GROUP BY, or DISTINCT
  - Columns associated for joining multiple tables.
- The index column order is as follows:
  - Put the column with the highest selectivity on the far left when creating a composite index. Selectivity = Different values in a column/Total rows in the column
  - Put the column with the smallest field length on the far left of the composite index. The smaller length a field has, the more data one page stores, and the better the I/O performance is.
  - Put the most frequently used column on the left of the composite index, so you can create fewer indexes.
- Avoid using redundant indexes, such as primary key (id), index (id), and unique index (id).
- Avoid using duplicate indexes, such as index(a,b,c), index(a,b), and index(a). Duplicate and redundant indexes may slow down queries because the RDS for MySQL query optimizer does not know which index it should use.
- When creating an index on the VARCHAR field, specify the index length based on selectivity. Do not index the entire field.

  If an index with the length of 20 bytes is the string type, its selectivity can reach 90% or above. In this case, use **count(distinct left(column name, index length))/count(*)** to check index selectivity.
- Use covering indexes for frequent queries.

  A covering index is a special type of index where all required fields for a query are included in the index. The index itself contains columns specified in WHERE and GROUP BY clauses, but also column combinations queried in SELECT, without having to execute additional queries.
- Constraints on foreign keys are as follows:

  The character sets of the columns for which a foreign key relationship is established must be the same, or the character sets of the parent and child tables for which a foreign key relationship is established must be the same.

SQL Statement Development

- Use prepared statements to perform database operations in programs. Prepared statements can be executed multiple times in a program once they are written, more efficient than SQL statements.
- Avoid implicit conversions because they may cause index to become invalid.

  Do not perform function conversions or math calculations on columns in the WHERE clause. Otherwise, the index becomes invalid.
- Do not use double percent signs (%%) or place % before a query condition, or the index cannot be used.
- Do not use **select *** for queries because using **select ***:
  - Consumes more CPUs, IP addresses, and bandwidth.
  - Causes covering indexes to become unavailable.

  - Increases the impact of table structure changes on code.
- Do not use subqueries. Subqueries generate temporary tables that do not have any indexes. If there is a lot of data, the query efficiency is severely affected. Convert subqueries into associated queries.
- Minimize the use of JOIN operations for more than 5 tables. Use the same data type for the fields that require JOIN operations.

  Each JOIN operation on a table occupies extra memory (controlled by **join_buffer_size**) and requires temporary table operations, affecting query efficiency. Do not use NATURAL JOIN.
- Reduce interactions with the same database as much as possible. The database is more suitable for processing batch operations.
- Replace OR clauses with IN clauses because IN clauses can effectively use indexes. Specify no more than 500 values for an IN clause.
- Do not perform reverse queries, for example, NOT IN and NOT LIKE.
- Do not use ORDER BY RAND() for random sorting.

  This operation loads all data that meets the conditions from the table to the memory for sorting, consuming more CPUs, I/O, and memory resources.

  Obtain a random value from the program and retrieve data from the involved database based on the value.
- If deduplication is not required, use UNION ALL instead of UNION.

  UNION ALL does not sort out result sets.
- Combine multiple operations and perform them in batches. The database is good for batch processing.

  This reduces interactions with the same database.
- If there are more than 1 million rows of write operations, perform them in multiple batches.

  A large number of batch writes may result in excessive primary/standby latency.
- If ORDER BY is used, use the order of indexes.
  - The last field of ORDER BY is a part of a composite index and is placed at the end of the composite index order.
  - Avoid file_sort to speed up queries.

  Correct example: in **where a=? and b=? order by c;**, index: **a_b_c**

  Wrong example: If an index supports range search, the index order cannot be used. For example, **WHERE a>10 ORDER BY b;**, index: **a_b** (sorting is not allowed)
- Use ANSI-standard SQL statements instead of MySQL extended SQL statements for DML operations. Common MySQL extended SQL statements include:
  - REPLACE INTO
  - INSERT ... ON DUPLICATE KEY UPDATE
- Stored procedures are not recommended because they are difficult to debug, extend, and transplant.
- To avoid logical dependency on the database, do not use triggers, event schedulers, or views for service logic.

- Large transactions are not recommended. If possible, a transaction should contain no more than five SQL statements because large transactions have problems such as long data lock time, too many caches, and connection consumption.
- TRUNCATE TABLE is faster than DELETE and uses fewer system and log resources. If the table to be deleted does not have a trigger and the entire table needs to be deleted, TRUNCATE TABLE is recommended.
- Do not run the **flush logs** command frequently to prevent automatic binlog deletion failures.
- Keep the time that a transaction can run no more than 180 seconds. There should be no more than 10 concurrent transactions whose duration is longer than 30 seconds.
- Do not modify more than 1 million rows in a single transaction.
- Do not run large SQL statements that are generated by the system. For example, if you run an **SELECT** statement of 9 MB, the memory consumption increases by about 37 MB during the execution, which is about 4 times the size of the statement.

Session-level Parameters That Are Recommended Not to Be Modified

- **foreign_key_checks**

  Description: The bool type. The default global value is **ON**. If this parameter is set to **ON** when a foreign key is created, the system checks the standardization of the foreign key, for example, the foreign key cannot reference other keys in the same table.

  Reasons for not changing its value:

  - If **foreign_key_checks** is set to **ON**, the foreign keys that cannot be checked using this parameter are not used in a standard manner. You are advised to optimize the SQL statements.
  - If this parameter is set to **OFF** at the thread level and some non-standard foreign keys are created on the primary instance, the DDL statements used for creating these foreign keys will fail to be executed during replication to the standby instance because the value of **foreign_key_checks** is still **ON** on the standby instance, thus causing replication exceptions. For details, see **Instance Reboot Failure or ERROR 1146: Table 'xxx' doesn't exist Reported During Table Operations**.
  - You are advised to set **foreign_key_checks** to **ON** also for single-node instances.

- **innodb_strict_mode**

  Description: The bool type. The default global value is **ON**. If this parameter is set to **ON**, an error instead of a warning is reported for some non-standard InnoDB table operations. For example, an error will be reported when the InnoDB page size is 16 KB but the size of a single row exceeds 8 KB.

  Reasons for not changing its value:

  - If this parameter is set to **OFF** at the thread level for the primary instance, the DDL statement ALTER TABLE can be executed successfully on the primary instance when there are many columns and the length of a single row exceeds 8 KB. However, because this parameter is still set to

ON on the standby instance, an error will be reported when this statement is executed on the standby instance, causing a replication exception.

- If you do need to change the value of this parameter, change it globally, for example, both on the primary and standby instances.

- **default_storage_engine**

  Description: The enumerated type. The value can only be an available storage engine name. The default global value is **InnoDB**. The default storage engine will be used if no storage engine is explicitly specified in the DDL statement CREATE TABLE or ALTER TABLE.

  Reasons for not changing its value:

  - RDS for MySQL primary/standby replication uses binlog-based replication of the open source community. If no storage engine is explicitly specified in the DDL statement CREATE TABLE or ALTER TABLE, the default storage engine will not be recorded in binlogs (consistent with the community). But when this statement is executed on the standby instance during primary/standby replication, the default value (**InnoDB**) of **default_storage_engine** of the replication thread is used.
  - If this parameter is changed to a storage engine other than InnoDB at the session level on the primary instance, running CREATE TABLE or ALTER TABLE will cause inconsistent storage engines used by tables created on the primary instance and standby instance.

- **unique_checks**

  Description: The bool type. The default global value is **ON**, which indicates that a uniqueness check will be performed on the unique keys of secondary indexes in InnoDB tables.

  Reasons for not changing its value:

  If this parameter is set to **OFF** at the session level for the primary instance, the uniqueness of the unique keys of secondary indexes will not be checked, and a DML statement with duplicate unique keys of secondary indexes can be executed successfully. However, on the standby instance, this DML statement will fail to be executed due to the uniqueness check, causing a replication exception.

- **sql_log_bin**

  Description: This parameter controls whether SQL statements of the current session are recorded in binlogs. The default value is **ON**.

  Reasons for not changing its value:

  RDS for MySQL primary/standby replication uses binlog-based replication of the open-source community. If this parameter is set to **OFF** for a single session for the primary instance, updates made to tables are not recorded in binlogs and cannot be synchronized to the standby instance. Data will be inconsistent between the primary and standby instances.

- **old_alter_table**

  Description: The bool type. The default value is **OFF**. If the value is set to **ON**, the algorithm implemented by copying temporary tables used in the ALTER TABLE statement affects database performance.

  Reasons for not changing its value:

- The ALTER IGNORE option may fail to be replayed on the standby instance.
- The performance of ALTER TABLE based on COPY is poor.

1.4.2 Database Management

1.4.2.1 Creating a Database

Scenarios

After a DB instance is created, you can create databases on it.

If your RDS instance is associated with a DDM instance, go to the DDM console to manage databases and accounts.

Constraints

- Databases cannot be created for DB instances that are in the process of being restored.
- You can only manage databases in the primary instance, for example, creating or authorizing users for databases.
- Database names must be unique.
- After a database is created on the RDS console, its name cannot be changed.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.
- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Creating a Database Through RDS

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

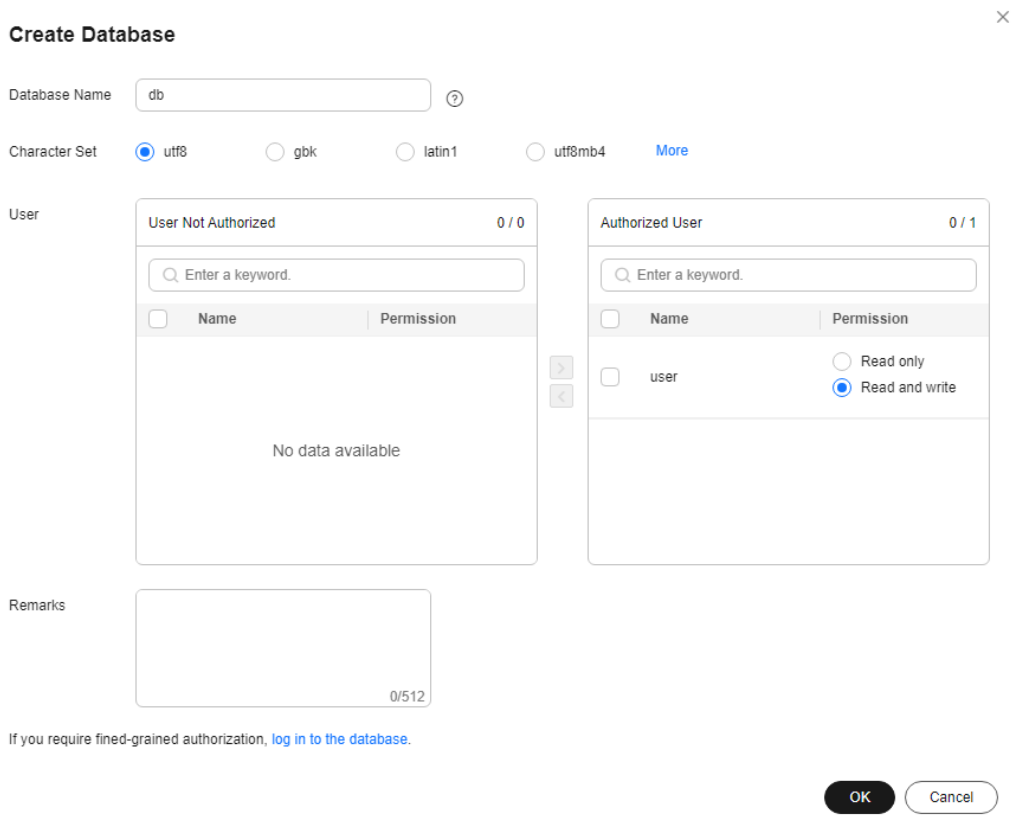
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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**.

Figure 1-37 Creating a database



- The database name consists of 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and dollar signs ($) are allowed. RDS for MySQL 8.0 does not support dollar signs ($). 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.
- The remarks can be empty or contain up to 512 characters. This parameter is available only for specified kernel versions. If your kernel version does not meet the following requirements, upgrade the kernel to the latest version by referring to **Upgrading a Minor Version**.
  - For RDS for MySQL 5.6, the kernel version should be 5.6.51.3 or later.
  - For RDS for MySQL 5.7, the kernel version should be 5.7.33.1 or later.
  - For RDS for MySQL 8.0, the kernel version should be 8.0.21.4 or later.
- Select unauthorized users and click to authorize permissions 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**.
- If you require fine-grained permissions control, log in to the database through the DAS console.

Step 6 After the database is created, 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

Creating a Database Through DAS

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Step 5 On the displayed login page, enter the correct username and password and click **Log In**.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Run the following command to create a database.

create database *database_name*;

----End

1.4.2.2 Modifying Database Remarks

Scenarios

RDS allows you to modify remarks for databases.

The **Remarks** column is displayed for all instances of the latest minor version. If the **Remarks** column is not displayed on the console, **check** and **upgrade the minor version of your instance**.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Databases**.

Step 6 Locate the target database and click in the **Remarks** column.

NOTE

The database remarks can be empty or contain up to 512 characters.

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

----End

1.4.2.3 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.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.
- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

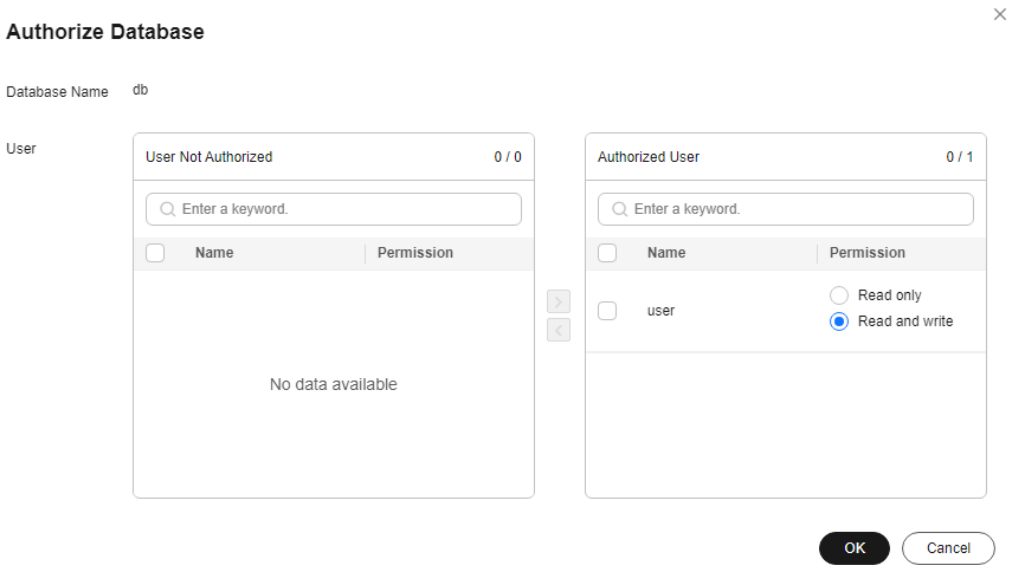
Step 4 On the **Instances** page, click the DB instance name.

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 no users are available, you can create one by referring to **Creating a Database Account**.

Figure 1-38 Granting database permissions



Step 7 Click **OK**.

----End

1.4.2.4 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.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.
- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 On the **Databases** page, locate the target database and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.4.3 Account Management (Non-Administrator)

1.4.3.1 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.

If your RDS instance is associated with a DDM instance, go to the DDM console to manage databases and accounts.

You can create a database account using RDS or DAS:

- **RDS**: RDS is easy to use. There are no special commands to remember.
- **DAS**: DAS is a powerful platform that offers more flexibility, but you need to be familiar with the creation commands. The process requires a bit more expertise.

Account Type

Table 1-19 Account description

Account Type	Description
Administrator account **root**	Only the administrator account **root** is provided on the instance creation page. For details about the supported permissions, see **RDS for MySQL Constraints**. **NOTE** Running **revoke**, **drop user**, or **rename user** on **root** may cause service interruption. Exercise caution when running any of these statements.
System accounts	To provide O&M services, the system automatically creates system accounts when you create RDS for MySQL DB instances. These system accounts are unavailable to you. • **rdsAdmin**: a management account with the highest permission. It is used to query and modify instance information, rectify faults, migrate data, and restore data. • **rdsRepl**: a replication account, used to synchronize data from the primary instance to the standby instance or read replicas. • **rdsBackup**: a backup account, used for backend backup. • **rdsMetric**: a metric monitoring account used by watchdog to collect database status data. • **rdsProxy**: a database proxy account, 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.
Other accounts	Accounts created through the console, APIs, or SQL statements After an account is created, you can assign permissions to it as required. For details, see **Changing Permissions for a Database Account**.

Constraints

- Accounts cannot be created for DB instances that are being restored.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.

- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Creating a Database Account Through RDS

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

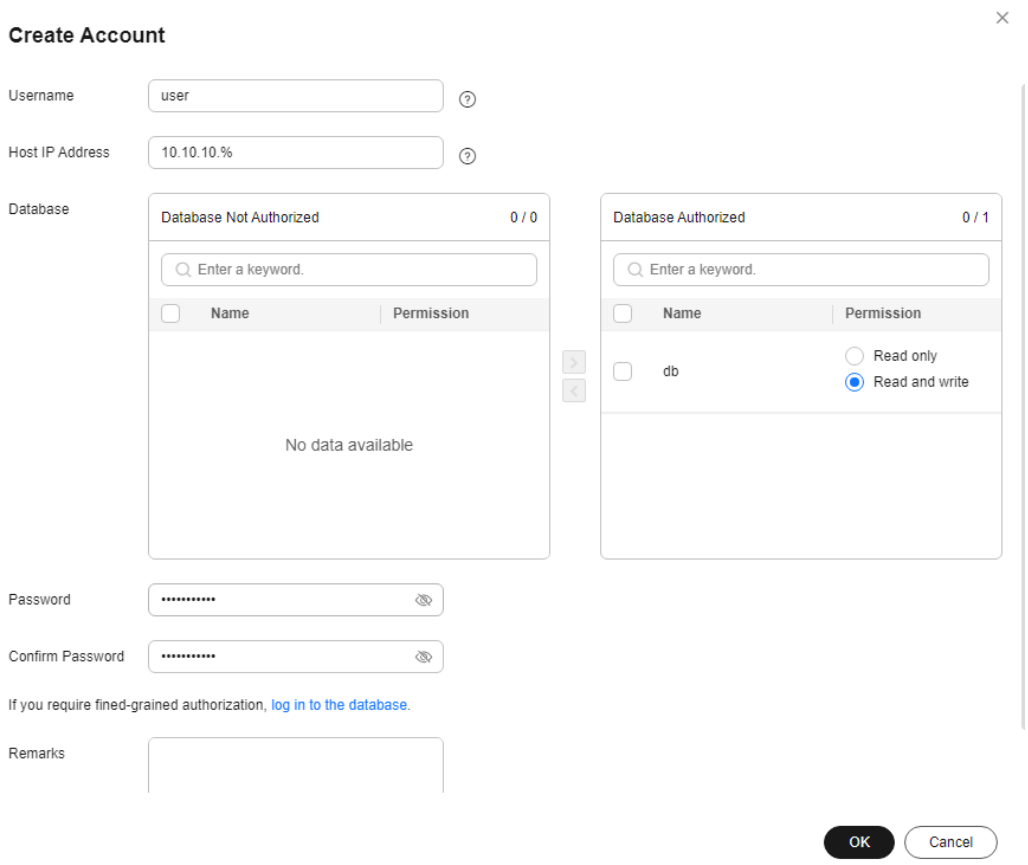
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane, choose **Accounts** and then click **Create Account**. In the displayed dialog box, specify **Username** and **Host IP Address**, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

Figure 1-39 Creating a database account



- If the DB engine version is MySQL 5.6, the username can contain 1 to 16 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

- If the DB engine version is MySQL 5.7 or 8.0, the username can contain 1 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- You can specify IP addresses that are allowed to access your DB instance.
  - To enable all IP addresses to access your instance, enter **%** for **Host IP Address**.
  - To enable all IP addresses in the subnet 10.10.10.*X* to access your instance, enter **10.10.10.%** for **Host IP Address**.
  - To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.1,172.16.213.9** (no spaces before or after the comma).
- 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 **Creating a Database**. You can also modify the permissions after the account creation by referring to **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 (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |).
- The password must be different from the username or username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.
- If you require fine-grained permissions control, log in to the database through the DAS console.

Step 6 After the database account is created, you can add remarks (for 8.0.25 and later versions), reset the password, modify permissions, change the host IP addresses for the account, and delete the account.

----End

Creating a Database Account Through DAS

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Step 5 On the displayed page, enter the correct username and password and click **Log In**.

Step 6 Create an account.

- On the top menu bar, choose **Account Management** > **User Management**. On the displayed page, click **Create User**. Then, configure basic information,

advanced settings, global permissions, and object permissions, and click **Save**. In the displayed dialog box, click **OK**.

For details about how to set permissions, see **Creating a User**.

- You can also choose **SQL Operations** > **SQL Query** from the top menu bar and run the following command to create an account:

  create user *username*;

----End

1.4.3.2 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.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.
- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Procedure

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

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

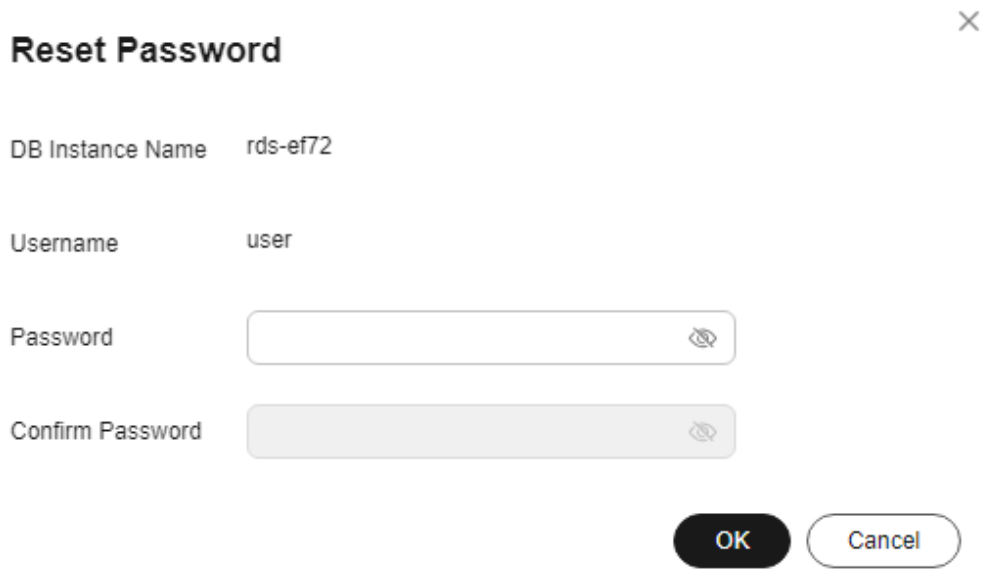
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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**.

Figure 1-40 Resetting a password



- 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 must be different from the username or username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.
- After the password is reset, the database will not be rebooted and permissions will not be changed.
- You can query password reset records on the CTS console. For details, see the *Cloud Trace Service User Guide*.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

1.4.3.3 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

- Take care when changing account permissions. Inappropriately configured account permissions can impact the DB instance or workloads.
- Permissions cannot be changed for DB instances that are in the process of being restored.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.
- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Procedure

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

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

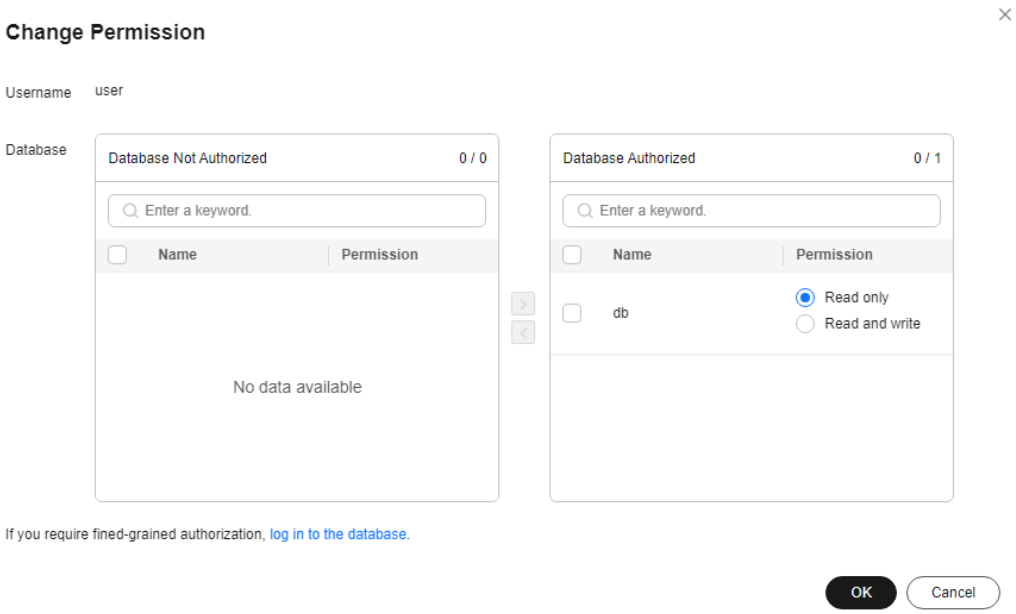
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **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.

Figure 1-41 Changing permissions



- If there are no unauthorized databases, you can create one by referring to **Creating a Database**.
- If you require fine-grained permissions control, log in to the database through the DAS console.

Step 7 Click **OK**.

----End

1.4.3.4 Modifying Host IP Addresses

Scenarios

You can change the host IP addresses that are allowed to access your instance as needed.

Constraints

- This operation cannot be performed for DB instances that are being restored.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.
- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Accounts**. Locate the target account and choose **More** > **Modify Host IP Address** in the **Operation** column.

- To enable all IP addresses to access your instance, enter **%** for **Host IP Address**.
- To enable all IP addresses in the subnet 10.10.10.*X* to access your instance, enter **10.10.10.%** for **Host IP Address**.
- To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.1,172.16.213.9** (no spaces before or after the comma).

Step 6 Click **OK**.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.4.3.5 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.
- Databases and accounts created using other methods than the RDS console and APIs are also displayed on the RDS console. If the names of the created databases or accounts do not meet the **database naming rule** or **account naming rule**, for example, the names containing Chinese characters or

unsupported special characters, the databases or accounts cannot be managed on the RDS console or through APIs.

- If the name of any database or account on the source database does not meet the **database naming rule** or **account naming rule**, the database or account cannot be managed on the RDS console or through APIs after being migrated to the destination RDS for MySQL instance.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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 **OK**.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.5 Database Migration

1.5.1 Migration Solution Overview

You can migrate data from DDM, GaussDB, TaurusDB, self-managed MySQL databases, self-managed Oracle databases, or MySQL databases built on other clouds to RDS for MySQL, or from one RDS for MySQL instance to another RDS for MySQL instance.

Data migration tools include Data Replication Service (DRS), mysqldump, and Data Admin Service (DAS). You are advised to use DRS because it is easy to use and can complete a migration task in minutes. DRS facilitates data transfer between databases, helping you reduce DBA labor costs and hardware costs.

DRS provides real-time migration and synchronization.

- Real-time migration: With DRS, you can migrate data from sources to destinations in real time. You just need to create a replication instance to connect to both the source and destination and configure objects to be migrated. DRS will help you compare metrics and data between the source

and destination, so you can determine the best time to switch to the destination database while minimizing service downtime.

- Real-time synchronization: It refers to the real-time flow of workload data from sources to destinations through a synchronization instance while consistency of data is ensured. It is different from migration. Migration means moving entire data of a database to another. Synchronization refers to the continuous flow of data between different applications.

For more information, see **What Is DRS?**

NOTE

During a DRS full migration, a large number of binlogs are generated in a short period of time and may be temporarily stored on your instance, which may cause the storage space to be used up. You are advised to **enable autoscaling** for your instance. To free up storage space, you can **set an appropriate retention period for binlogs** and **clear the binlogs** in a timely manner.

Migration Solutions

Table 1-20 RDS for MySQL migration solutions

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
RDS for MySQL	Small	One-time	Some time	Use mysqldump to copy data from the source to the destination RDS for MySQL instance.	**Migrating Data to RDS for MySQL Using mysqldump**
	Any	One-time or continuous	Minimal	Use DRS to migrate table, database, or instance data of the source to the destination RDS for MySQL instance. **NOTE** For details about real-time migration billing, see **DRS Migration Billing**.	**From MySQL to RDS for MySQL**
	Medium	One-time	Some time	Use DAS to export data from the source and then import the data to the destination RDS for MySQL instance.	**Migrating Data to RDS for MySQL Using the Export and Import Functions of DAS**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from the source to the destination RDS for MySQL instance.	**From MySQL to RDS for MySQL**
	Any	Continuous	Minimal	Use DRS to synchronize data from the source to the DR RDS for MySQL instance. **NOTE** If there is a regional failure, this solution can ensure service continuity by synchronizing data between databases.	• **Configuring Remote Single-Active DR for an RDS for MySQL Instance Using DRS** • **From MySQL to RDS for MySQL (Dual-Active DR)**
DDM	Any	One-time or continuous	Minimal	Use DRS to synchronize data from DDM to RDS for MySQL.	**From DDM to RDS for MySQL**
GaussDB distributed	Any	One-time or continuous	Minimal	Use DRS to synchronize data from a GaussDB distributed instance to RDS for MySQL.	**From GaussDB Distributed to RDS for MySQL**
GaussDB primary/standby	Any	One-time or continuous	Minimal	Use DRS to synchronize data from a GaussDB primary/standby instance to RDS for MySQL.	**From GaussDB Primary/Standby to RDS for MySQL**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
TaurusDB	Any	One-time or continuous	Minimal	Use DRS to synchronize data from TaurusDB to RDS for MySQL.	**From TaurusDB to RDS for MySQL**
• On-premises MySQL databases • MySQL databases on ECSs	Any	One-time or continuous	Minimal	Use DRS to migrate data from self-managed MySQL databases to RDS for MySQL. DRS supports incremental migration, so you can replicate ongoing data changes to keep sources and destinations in sync while minimizing the impact of service downtime and migration. **NOTE** For details about real-time migration billing, see **DRS Migration Billing**.	**Migrating Data from Self-Managed MySQL Databases to RDS for MySQL**
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from self-managed MySQL databases to RDS for MySQL. Real-time synchronization refers to the real-time flow of workload data from sources to destinations through a synchronization instance while consistency of data is ensured. It is suitable for real-time analysis, and data transfer in report systems and data warehouses.	**From MySQL to RDS for MySQL**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
	Any	Continuous	Minimal	Use DRS to synchronize data from self-managed MySQL databases to the DR RDS for MySQL instance. **NOTE** If there is a regional failure, this solution can ensure service continuity by synchronizing data between databases.	• **From MySQL to RDS for MySQL (Single-Active DR)** • **From MySQL to RDS for MySQL (Dual-Active DR)**
• On-premises Oracle databases • Oracle databases on ECSs	Any	One-time or continuous	Minimal	Use DRS to synchronize data from self-managed Oracle databases to RDS for MySQL.	**From Oracle to RDS for MySQL**
MySQL databases on other clouds	Any	One-time or continuous	Minimal	Use DRS to migrate data from MySQL databases on other clouds to RDS for MySQL. **NOTE** For details about real-time migration billing, see **DRS Migration Billing**.	**Migrating MySQL Databases from Other Clouds to RDS for MySQL**
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from MySQL databases on other clouds to RDS for MySQL.	**From MySQL to RDS for MySQL**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
	Any	Continuous	Minimal	Use DRS to synchronize data from MySQL databases on other clouds to the DR RDS for MySQL instance. **NOTE** If there is a regional failure, this solution can ensure service continuity by synchronizing data between databases.	• From MySQL to RDS for MySQL (Single-Active DR) • From MySQL to RDS for MySQL (Dual-Active DR)

DRS Migration Billing

- Real-time migration supports only the pay-per-use billing mode.

  Real-time migration tasks are free of configuration and traffic fees in the first seven days, lowering your costs for migrating data to the cloud.
- Real-time synchronization and DR support pay-per-use and yearly/monthly billing modes.

  Real-time migration and synchronization will provide long-term discounts, lowering your costs for data transfers.

For more information, see **Data Replication Service Billing**.

1.5.2 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 the DB instance. For details, see **Binding an EIP**.
     ii. Ensure that the local device can access the EIP.
2. Install a MySQL client on the prepared ECS or device.

   For details, see **How Can I Install the MySQL Client?**

NOTE

The MySQL client version must be the same as the DB engine version of your RDS for MySQL instance. A 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 **Changing a Floating IP Address**.

RDS system databases **mysql** and **sys** cannot be imported from one RDS for MySQL instance to another.

Exporting Data

Before migrating a database to RDS, its data needs to be exported.

NOTICE

- The export tool must match the DB engine version.
- Database migration is performed offline. Before the migration, you have to stop all applications using the source database.
- Take care when exporting or importing data. Improper operations can cause instance or workload exceptions.

Step 1 Log in to the source database.

Step 2 Use the mysqldump tool to export the table structure to an SQL file.

NOTICE

The **mysql** database is required for RDS management. When exporting the table structure, do not specify **--all-database**. Otherwise, a database fault will occur.

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 to an SQL file.

NOTICE

The **mysql** database is required for RDS management. When exporting data, do not specify **--all-database**. Otherwise, a database fault will occur.

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 3306 -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 the RDS DB instance.

Step 2 Connect to the RDS DB instance through a client.

Step 3 Import the table structure 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, specify a database in the command. 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 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, specify a database in the command. 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 5 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

1.5.3 Migrating Data to RDS for MySQL Using the Export and Import Functions of DAS

Scenarios

Data Admin Service (DAS) is a one-stop management platform that allows you to manage Huawei Cloud databases on a web console. It offers database development, O&M, and intelligent diagnosis, making it easy to use and maintain databases.

To back up or migrate data, you can use DAS to export data from the source database first and then import the data to from your local PC or OBS bucket to the destination database.

For more information, see **Import and Export**.

Constraints

- Take care when exporting or importing data. Improper operations can cause instance or workload exceptions.
- Only one file that is no larger than 1 GB can be imported at a time.
- Only data files in the CSV or SQL format can be imported.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- Data cannot be exported or imported using cross-region OBS buckets.
- If there are more than 100,000 tables in an RDS for MySQL 8.0 instance (or more than 10,000 tables in an RDS for MySQL 5.7 or 5.6 instance), an error will be reported when you export data using the **Export Database** function of DAS. In this case, use the **Export SQL Result** function instead.

Exporting Data

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

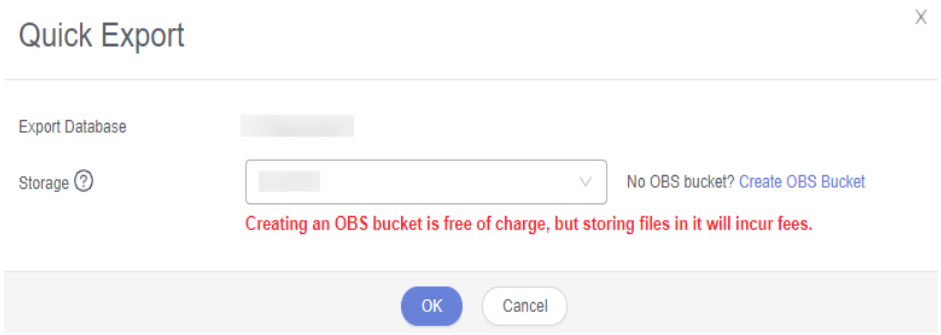
Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.

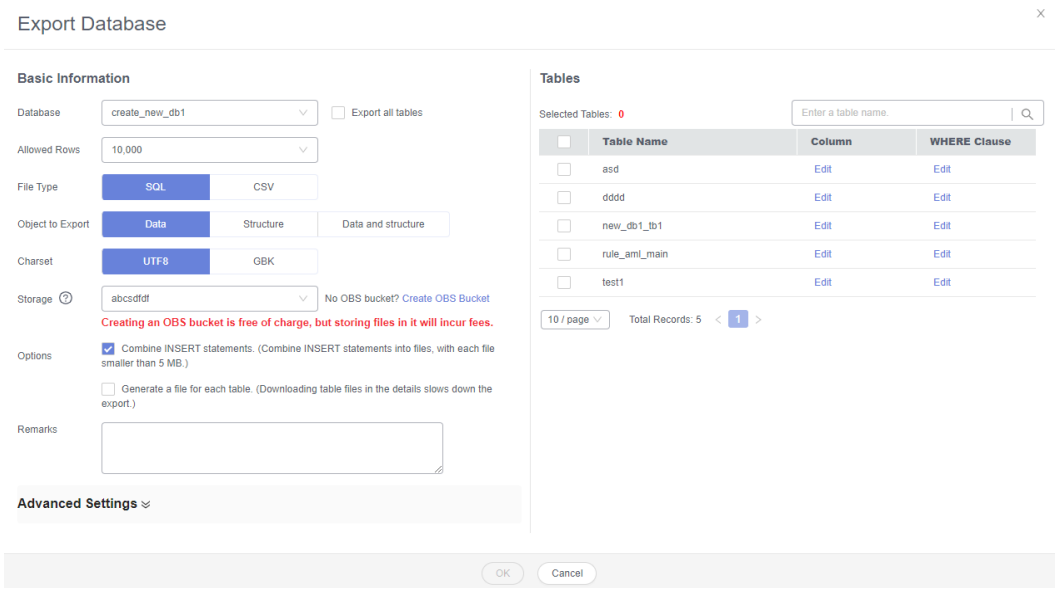
Alternatively, click **Quick Export** and select the target database. On the displayed page, select a storage path and click **OK**.

Figure 1-42 Quick export



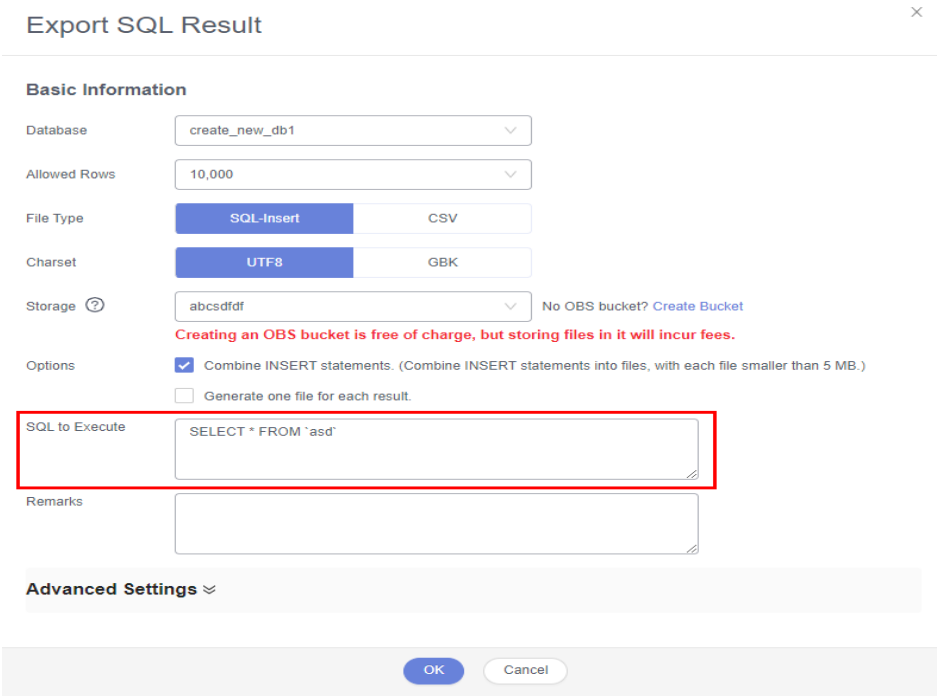
Step 8 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

Figure 1-43 Creating an export task



NOTE

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.



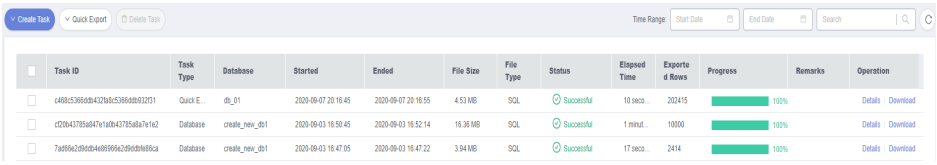
- Databases are classified into user databases and system databases. System databases cannot be exported. If system database data is required, deploy system database services in a created user database, so that you can export the system database data from the user database.
- DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

Figure 1-44 Task list



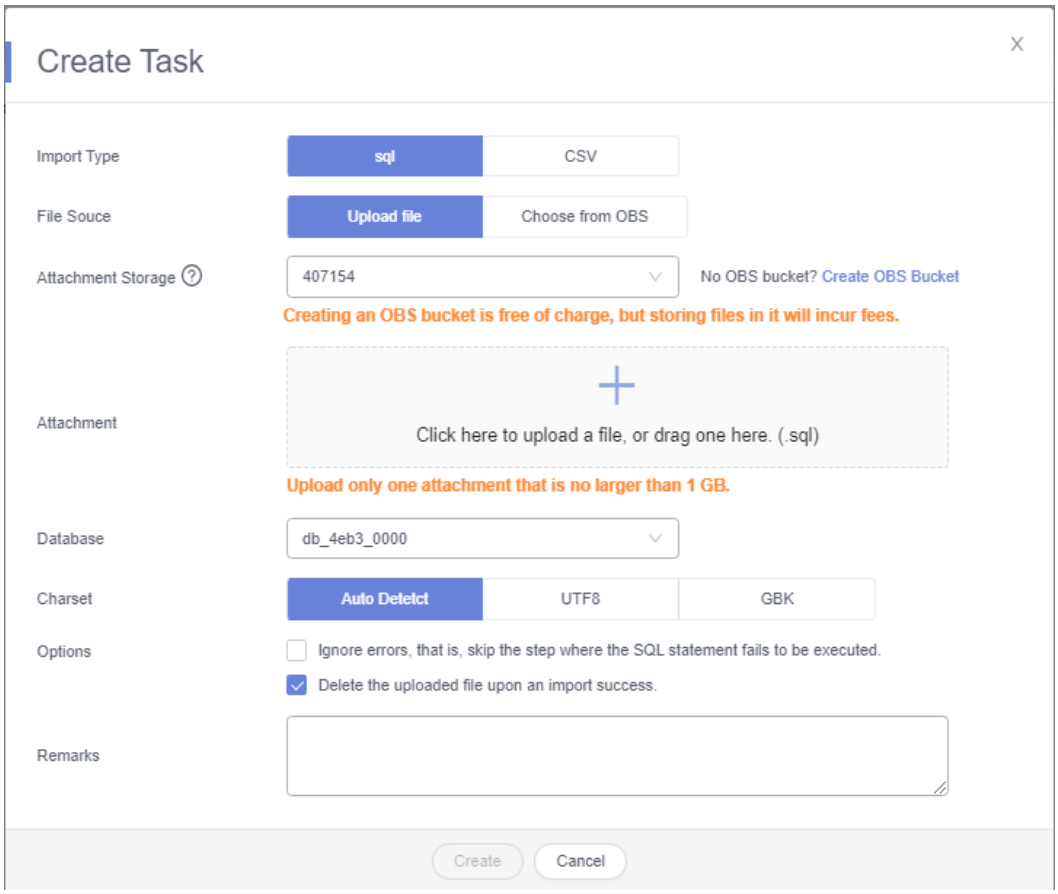
----End

Importing Data

Step 1 On the top menu bar, choose **Import and Export** > **Import**.

Step 2 Import a file from your local PC or an OBS bucket.

Figure 1-45 Creating an import task



- From your local PC

  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.

  For security purposes, imported files are stored in OBS buckets.

  NOTE

  - To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
  - If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database.

- From an OBS bucket

  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

  NOTE

  The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting the import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 4 View the import progress in the task list or check task details.

----End

1.6 Version Upgrade

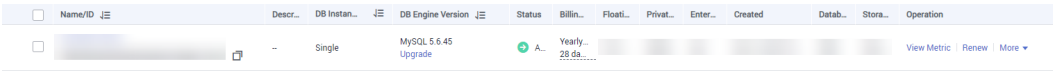
1.6.1 Upgrading a Minor Version

Scenarios

RDS for MySQL supports minor version upgrades to improve performance, add new functions, and fix bugs.

By default, a newly created DB instance uses the latest minor version. When a new minor version is released on Huawei Cloud, the **Upgrade** link is displayed in the **DB Engine Version** column on the **Instances** page. You can click **Upgrade** to go to the minor version upgrade page.

Figure 1-46 Version upgrade



Upgrade Methods

A minor version can be upgraded in either of the following ways:

- Upon submission: The system **upgrades the minor version** upon your manual submission of the upgrade request.
- In maintenance window: The system upgrades the minor version during the maintenance window you specified. For details about how to change the maintenance window, see **Changing the Maintenance Window**.

If the kernel version of your instance has potential risks or major defects, has expired, or has been brought offline, the system will notify you by SMS message or email and deliver an upgrade task during the maintenance window.

Precautions

- When any new minor version is released for addressing issues and vulnerabilities from the open source community, upgrade the minor version of your instance **immediately** or **during the maintenance window**.
- The upgrade will cause the DB instance to reboot and briefly interrupt services. To limit the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.
- A minor version upgrade involves switchovers between primary and standby instances, which cause a brief service interruption. Besides, there can be two waits of up to 10s for a single SQL statement to update or write data because the default replication between primary and standby instances is semi-synchronous. To avoid the waits, change the replication mode to asynchronous before the upgrade.

- If primary and standby DB instances are deployed in the same AZ, a minor version upgrade will trigger a switchover. If they are deployed in different AZs, a minor version upgrade will trigger two switchovers.
- When you upgrade a minor version of a primary DB instance, minor versions of read replicas (if any) will also be upgraded automatically (they cannot be upgraded separately). Perform the upgrade during off-peak hours because the DB instance will be rebooted after the upgrade is complete.
- If your RDS instance is involved in a DRS task, upgrading the minor version may cause the DRS task to fail.

  You are advised to check the retention period of RDS instance binlogs before upgrading the minor version.

  - If the binlogs are within the retention period, the DRS task will automatically restart after the minor version is upgraded.
  - If the binlogs are beyond the retention period, you need to reconfigure or recreate a DRS task.
- A minor version upgrade cannot be rolled back after the upgrade is complete. If the upgrade fails, the DB instance will be automatically rolled back to the source version.
- You are advised to perform a full backup before upgrading a minor version.
- A minor version can be upgraded in minutes.
- DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a minor version upgrade.

  During a minor version upgrade, if you are prompted that there are DDL operations being executed on the primary instance, do as follows:

  - Change the status of the event whose **STATUS** is **SLAVESIDE_DISABLED** to **ENABLED** or **DISABLED**, and then perform the upgrade.
  - Delete the events whose **STATUS** is **SLAVESIDE_DISABLED** and then perform the upgrade.

Constraints

- If the replication delay between primary and standby DB instances is longer than 300 seconds, the minor version cannot be upgraded.
- For primary/standby DB instances, the standby DB instance is upgraded first and then the primary DB instance is upgraded afterwards.
- Minor versions cannot be upgraded for DB instances with abnormal nodes.
- RDS for MySQL DB instances with the event scheduler function enabled do not support minor version upgrades. If you want to perform a minor version upgrade, disable event scheduler first. For operation details, see **Enabling or Disabling Event Scheduler**.
- TLSv1.1 is not supported for RDS for MySQL 8.0.28 or later versions. To modify the TLS version, change the value of the parameter **loose_tls_version**.

Procedure

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

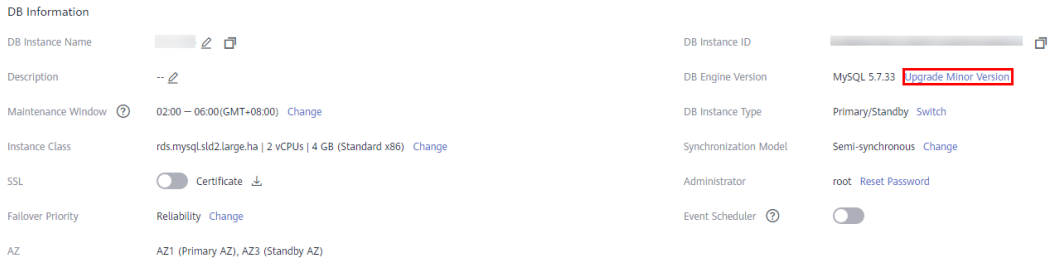
Step 2 Click  in the upper left corner and select a region.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **DB Engine Version**, click **Upgrade Minor Version**.

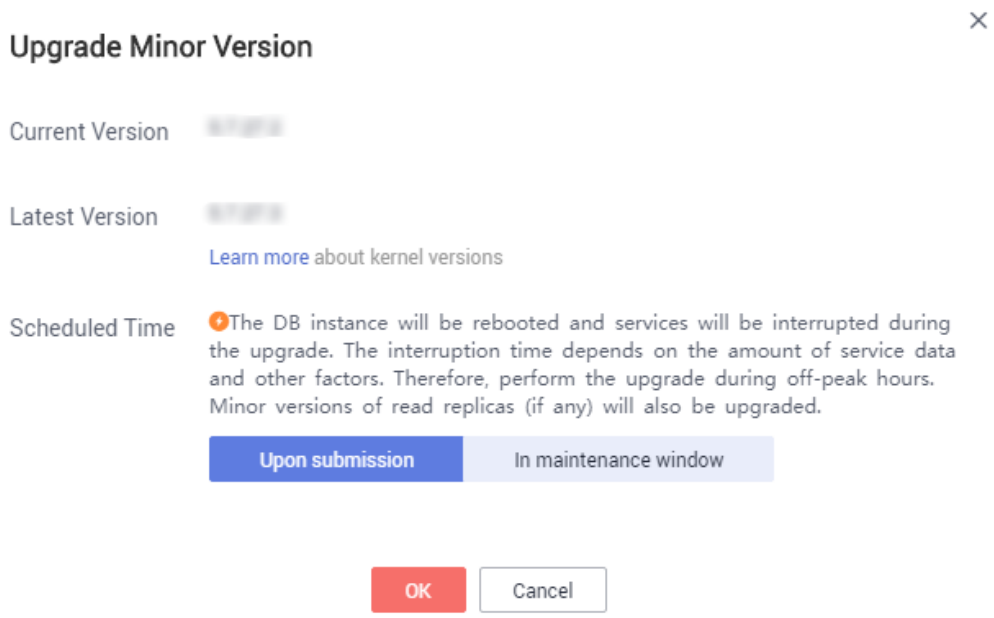
Figure 1-47 Upgrading a minor version



Step 6 In the displayed dialog box, select a scheduled time and click **OK**.

- Upon submission: The system upgrades the minor version immediately after you have submitted your upgrade request.
- In maintenance window: The system will upgrade the minor version during the maintenance window that you have configured.

Figure 1-48 Selecting a scheduled time



----End

Follow-up Operations

Return to the instance list. In the navigation pane on the left, choose **Task Center** and check the progress of the upgrade task.

- If you have selected **Upon submission** for **Scheduled Time**:

  On the **Instant Tasks** page, search for the task **Upgrading a MySQL DB instance engine version** and check the execution progress. Instant tasks cannot be canceled.
- If you have selected **In maintenance window** for **Scheduled Time**:

  On the **Scheduled Tasks** page, search for the instance ID and check the execution status of the upgrade task.

  If the task is in the **To be executed** state, you can click **Cancel** to cancel the task.

For details, see **Viewing a Task**.

1.6.2 Upgrading an RDS for MySQL Instance from 5.7 to 8.0

Scenarios

RDS for MySQL allows you to upgrade the major version of a DB instance in either of the following ways:

- **Upgrading a major version on the RDS console**: For details about version functions, see **RDS for MySQL Kernel Version Description**.

  To use this function, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console to apply for required permissions.
- **Upgrading a major version using DRS**: You can migrate instance data from an earlier version to a later version.

Precautions

- A pre-check is required for DB instances to be upgraded from MySQL 5.7 to 8.0. Note that:
  - The time required for the pre-check depends on how many tables there are in your instance. To prevent service interruptions, perform the upgrade during off-peak hours.
  - The check report is retained for 24 hours. Download it before it is deleted.
  - If the **Parameters** item fails the check, rectify the fault based on the check items in the check details by referring to **Table 1-21**. Check items whose severity is **Error** must be rectified before the upgrade can be performed. Check items whose severity is **Warning** only requires you to learn about the kernel feature changes after the upgrade.
  - After the fault is rectified, you need to click **Retry** to obtain the new check results.
- Only the latest minor version of MySQL 5.7 can be upgraded to that of MySQL 8.0. Ensure that your instance uses the latest minor version of MySQL 5.7.
- You are advised not to perform a major version upgrade during the backup time window.
- You are advised to perform a full backup before upgrading a major version.
- Upgrading a major version will cause a connection interruption for 10 to 120 seconds. Ensure that your applications support automatic reconnection.

Perform this operation during off-peak hours because upgrading a major version during peak hours takes longer time.

- When you upgrade a major version of a primary DB instance, major versions of its read replicas (if any) will also be upgraded. Major versions of read replicas cannot be upgraded separately.
- A major version upgrade cannot be rolled back after the upgrade is complete.
- Before the upgrade, compare the old and new versions carefully. To ensure that the syntax and features of the old version used by your applications are compatible with the new version, create a new RDS for MySQL 5.7 or 8.0 instance to test the syntax before the upgrade.
- You are advised to restore data to a new instance and perform a test first. After confirming that all functions are normal, upgrade the original instance.
- When your instance is being upgraded, storage autoscaling does not take effect, so sufficient storage must be reserved to ensure that data writes can continue during the upgrade.
- Scheduled major version upgrades need to be prepared in advance and cannot be canceled.
- After a major version upgrade is complete, the backups before the upgrade cannot be used for the instance of the new version, and the time points before the upgrade cannot be selected for point-in-time recovery (PITR).
- DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a major version upgrade.
- After a major version upgrade, specification parameters are reset to the default values of the new version, including **threadpool_size**, **innodb_buffer_pool_size**, **innodb_io_capacity**, **innodb_io_capacity_max**, **innodb_buffer_pool_instances**, **back_log**, and **max_connections**.
- The value ranges of a given parameter may be different in RDS for MySQL 5.7 and 8.0. For example, if **max_execution_time** is set to a value less than **60000** in an RDS for MySQL 5.7 instance, it will be reset to the default value **0** after a major version upgrade. This is because the minimum value of **max_execution_time** in RDS for MySQL 8.0 can only be **60000**.

Constraints

- For details about kernel versions, see **Kernel Version Description**.
- If the replication delay between primary and standby instances is longer than 300 seconds, the major version cannot be upgraded.
- The major version cannot be upgraded for DB instances with abnormal nodes.
- RDS for MySQL 5.7 and later versions no longer support Sequence Engine.
- RDS for MySQL DB instances support a maximum of 500,000 tables (including system tables and data tables). If the number of tables is greater than 500,000, the major version upgrade may fail.
- RDS for MySQL DB instances with event scheduler enabled do not support major version upgrades. If you want to perform a major version upgrade, disable event scheduler first. For details, see **Enabling or Disabling Event Scheduler**.
- After your instance is upgraded to MySQL 8.0, read replicas (if any) share the SQL statement concurrency control rules of the primary instance. To prevent

those rules of the primary instance from affecting workloads on the read replicas, review and adjust the rules of the primary instance before the upgrade.

- Before an upgrade, run the **XA recover;** statement to check whether there are still any XA transactions in the prepared state. If there are, commit or roll back the transactions, and then perform the upgrade.
- If an instance meets one of the following conditions, it cannot be upgraded from MySQL 5.7 to 8.0:
  - It has been associated with a DDM instance or database proxy has been enabled for it.
  - It is a single-node instance.
  - Its read replicas have SQL statement concurrency control rules. To upgrade the instance, delete such rules from the read replicas.

Upgrade Check Items and Handling Suggestions for Upgrade Failures

Table 1-21 Check items and rectification

Check Item	Description	Rectification
utf8mb3Check	Check the character set utf8mb3.	In MySQL 5.7, the character set utf8 is equivalent to utf8mb3. In MySQL 8.0, the character set utf8 is equivalent to utf8mb4. After the upgrade, if you use utf8 to create tables, utf8mb4 is actually used. No action is required.
removedSysVars	Check the removed system variables.	Some system variables have been deleted from MySQL 8.0. This check item does not affect the upgrade. No action is required.
sysVarsNewDefaults	Check the default values of system variables.	Some system variables have new default values in MySQL 8.0. If you have changed the values of such variables before the upgrade, the new values are retained after the upgrade. If you have not changed the values, the new default values in MySQL 8.0 are used. This check item does not affect the upgrade. No action is required.

Check Item	Description	Rectification
zeroDatesCheck	Check for zero dates, DATETIME, and TIMESTAMP values.	In MySQL 8.0.16 or later versions, when a zero-value date (for example, 2024-00-00) is used as a query condition, MySQL converts the character string to DATE. If the conversion fails, an error is reported. This check item does not affect the upgrade but affects the query logic after the upgrade. You do not need to handle the error reported for the **global.sql_mode** setting. You are advised to check for such values. If there is any, change it to a non-zero value.
enumSetElementLenghtCheck	Check the length of elements in ENUM and SET columns.	If any column contains more than 255 characters, modify it and ensure that its length does not exceed this limit.
reservedKeywordsCheck	Check reserved words.	If there are any objects with names conflicting with the reserved words of MySQL 8.0, change the names.
mysqlDollarSignNameCheck	Check the dollar sign ($).	If any database object name contains $, delete it from the name.
mysqlInvalid57NamesCheck	Check for invalid database names, table names, and column names.	Modify invalid names.
groupByAscCheck	Check the GROUP BY ASC and DESC syntax.	The GROUP BY ASC and DESC syntax has been removed from MySQL 8.0. Modify the database objects that contain the GROUP BY ASC or DESC syntax. You can remove the ASC and DESC keywords from the GROUP BY clause and place them in the ORDER BY clause.
checkTableOutput	Check tables using the **check table x for upgrade** command.	Rectify the fault based on the description in the check results.
engineMixupCheck	Check whether InnoDB recognizes tables that belong to other engines.	If there are tables recognized by the InnoDB engine, but the SQL layer considers that they belong to other engines, **submit a service ticket**.

Check Item	Description	Rectification
foreignKeyLengthCheck	Check the length of the foreign key constraint name.	If the foreign key constraint name contains more than 64 characters, modify it.
nonNativePartitioningCheck	Check for partitioned tables of non-native partitioning engines.	MySQL 8.0 supports only InnoDB and NDB as partitioning engines. Convert the engine to InnoDB or delete the partitions.
routinesSyntaxCheck	Check for syntax incompatibility.	If the definitions of database objects such as stored procedures and functions contain incompatible syntax, for example, the syntax conflicts with the reserved words in MySQL 8.0, modify the syntax based on the description in the check results.
maxdbFlagCheck	Check for obsolete MAXDB sql_mode flags.	In MySQL 8.0, the **MAXDB** option has been deleted from **sql_mode**. Modify the **sql_mode** parameter to exclude MAXDB.
sqlModeFlagCheck	Check for obsolete sql_mode flags.	Some sql_mode flags have been deleted from MySQL 8.0. Modify the **sql_mode** parameter based on the description in the check results.
removedSysLogVars	Check the removed system variables for system logs.	Some system variables for system logs have been removed from MySQL 8.0. No action is required.
mysqlIndexTooLargeCheck	Check the index length.	Both MySQL 5.7 and MySQL 8.0 allow the maximum index length of 767 bytes. In MySQL 8.0, if utf8mb4 is used, an index cannot be longer than 191 characters. Change the index length to no more than 191 characters.
circularDirectoryCheck	Check whether any tablespace data file path uses a circular directory.	The target version does not allow tablespace data file paths to contain circular directory references (for example, /../). **Submit a service ticket** for handling the problem.
columnsWhichCannotHaveDefaultsCheck	Check default values of columns.	Default values are not allowed for columns of the BLOB, TEXT, GEOMETRY, or JSON type. Run the ALTER TABLE statement to delete the default values.

Check Item	Description	Rectification
removedFunctionsCheck	Check the removed functions.	Some functions have been deleted from the target version. Modify the corresponding database objects based on the description in the check results.
mysqlOrphanedRoutinesCheck	Check for orphaned stored procedures or functions.	Orphaned stored procedures or functions cannot run because the database objects referenced by them do not exist. Delete such stored procedures or functions.
mysqlEmptyDotTableSyntaxCheck	Check for obsolete identifiers.	Change such identifiers in database objects based on the description in the check results.
mysqlSchemaCheck	Check for table name conflicts.	There are some tables added to MySQL 8.0. Run the RENAME TABLE statement to change the conflicting table names.
mysqlInvalidEngineForeignKeyCheck	Check for foreign keys pointing to tables from other engines.	Run the ALTER TABLE statement to change the table's engine or delete the foreign key reference.
lowerCaseNameCheck	Check whether any table name contains uppercase letters when **lower_case_table_names** is set to **1**.	If any table name contains uppercase letters when **lower_case_table_names** is set to **1**, the upgrade will fail. Change the value of **lower_case_table_names** to **0** first, run the RENAME TABLE statement to change the table name to lowercase, and then change the **lower_case_table_names** value back to **1**.
specVarInConfigFileCheck	Check whether the values of **sql_mode** and **loose_tls_version** in the configuration file are outdated.	Some sql_mode flags have been removed from the target version. In MySQL 8.0.28 and later versions, the **loose_tls_version** parameter does not support TLSv1 or TLSv1.1. Change the values of **sql_mode** and **loose_tls_version** based on the check results.
reversedUserCheck	Check whether the **mysql.infoschema@localhost** account has been created.	MySQL 8.0 has a built-in **mysql.infoschema@localhost** account. If this account exists in MySQL 5.7, the upgrade will fail. Delete this account before the upgrade.

Check Item	Description	Rectification
schemaInconsistencyCheck	Check for database structure inconsistency due to table file removal or corruption.	If the **.frm** table file is missing, **submit a service ticket**.
geometryIndexCheck	Check whether any spatial coordinate is a spatial index when it is used as an index.	In the target version, the spatial coordinate index must be a spatial index. Recreate or delete the index.
danglingIndexCheck	Check for dangling indexes.	If any table has dangling FTS_DOC_ID due to deletion of the full-text index column, run the OPTIMIZE TABLE statement.
viewColumnCheck	Check the length of the view column name.	In the target version, a view column name can contain a maximum of 64 characters. If this limit is exceeded, run the ALTER VIEW statement to modify the column name.
partitionedTablesInSharedTablespaceCheck	Check whether there are any partitioned tables in shared tablespaces.	The target version does not support shared tablespaces. Run the ALTER TABLE statement to move partitioned tables to independent tablespaces.
partitionsReferencedCheck	Check whether any partitioned table is referenced by an ordinary table using a foreign key.	Partitioned tables cannot be referenced by ordinary tables. Delete the foreign key reference.
partitionsRangeDateCheck	Check for tables partitioned by time.	In MySQL 8.0, if tables are partitioned by time, the content in the time column must be in the standard format (for example, YYYY-MM-DD hh:mm:ss or YY-MM-DD hh:mm:ss). Change the time values in the table to the standard format.

Table 1-22 Handling Suggestions for Upgrade Failures

Symptom	Impact	Handling Suggestion
After RDS for MySQL 5.7 is upgraded to 8.0, indexes become invalid when some SQL statements are executed because the default character set is changed. By default, MySQL 5.7 uses utf8mb3, while MySQL 8.0 uses utf8mb4.	If you create a table using the default character set in MySQL 5.7, the upgrade will not change the character set to that of MySQL 8.0. After the upgrade, if you create another table and join the two tables with different character sets by running **JOIN**, the SQL statement execution will be prolonged due to index selection.	Modify the character sets of databases, tables, and fields and the default character set in MySQL 8.0 to be the same as those in MySQL 5.7.
The full-text index is deleted.	The upgrade may fail.	Run the **OPTIMIZE TABLE** statement to recreate the table and check for dangling FTS_DOC_ID. For details, see **Table 1-21**.
There are foreign key constraints (specified by the **foreign_key_check** parameter) on partitioned tables.	The upgrade fails.	Remove the foreign key constraint.
RDS for MySQL 5.7 contains the **mysql.events** table whose DEFINER column is blank or null.	The upgrade fails.	Set the DEFINER column to a non-null value.
The case formats of column names at the server and InnoDB layers do not match.	The upgrade fails.	Run the **OPTIMIZE TABLE** statement to recreate the table.
The case formats of the fields contained in the indexes at the server and InnoDB layers do not match.	The upgrade fails.	Run the **OPTIMIZE TABLE** statement to recreate the table.
The BTREE SPATIAL INDEX index, which is not supported by RDS for MySQL 8.0, is used.	The upgrade fails.	Delete the BTREE SPATIAL INDEX index.

Upgrading a Major Version on the RDS Console

Step 1 Log in to the management console.

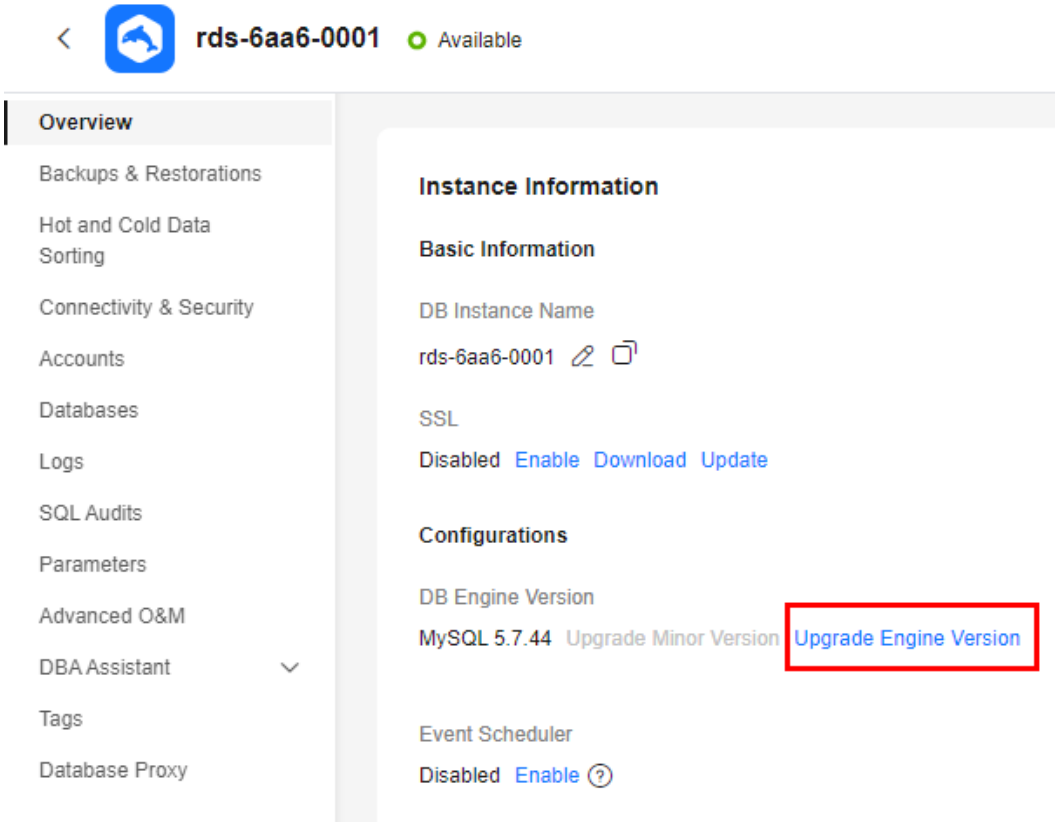
Step 2 Click in the upper left corner and select a region and a project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

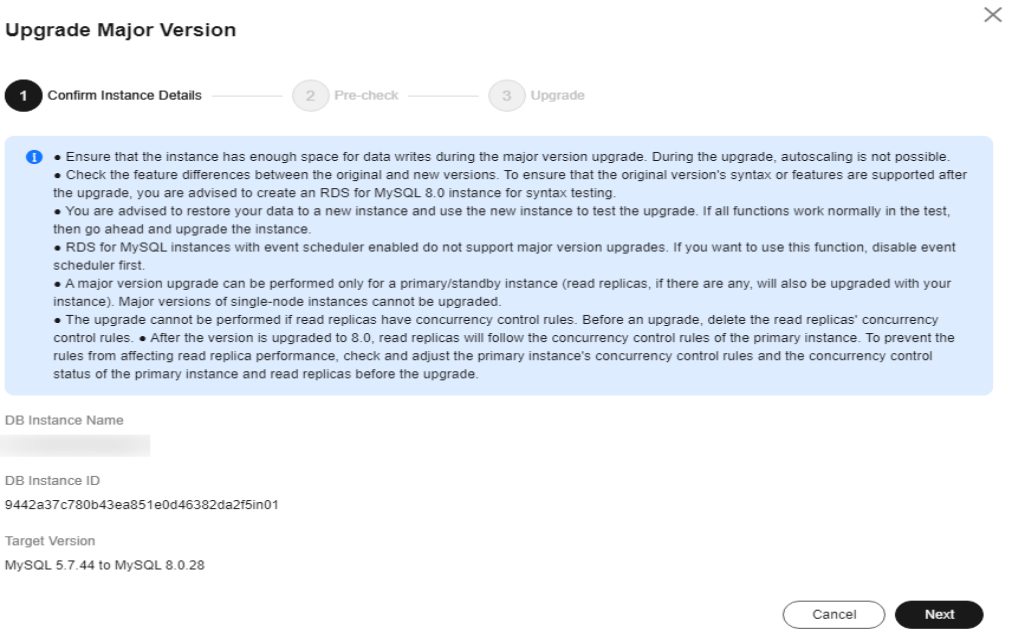
Step 5 Under **DB Engine Version**, click **Upgrade Engine Version**.

Figure 1-49 Upgrading a major version



Step 6 In the displayed dialog box, confirm instance details and click **Next**.

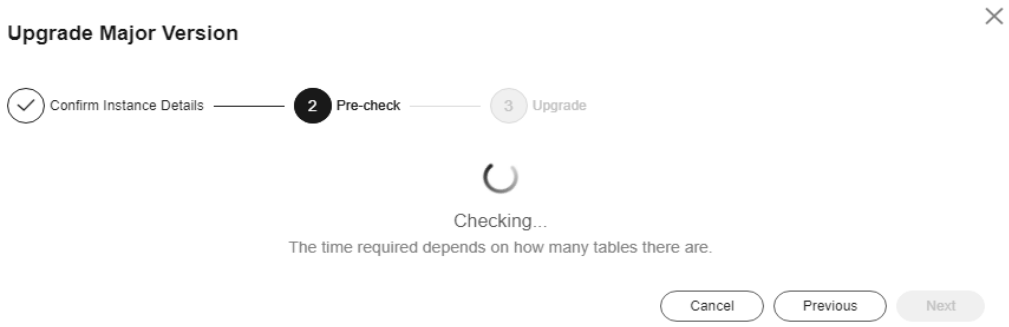
Figure 1-50 Confirming instance details



- If the instance is to be upgraded from MySQL 5.7 to 8.0, go to **Step 7**.
- In other scenarios, go to **Step 8**.

Step 7 Perform a pre-check.

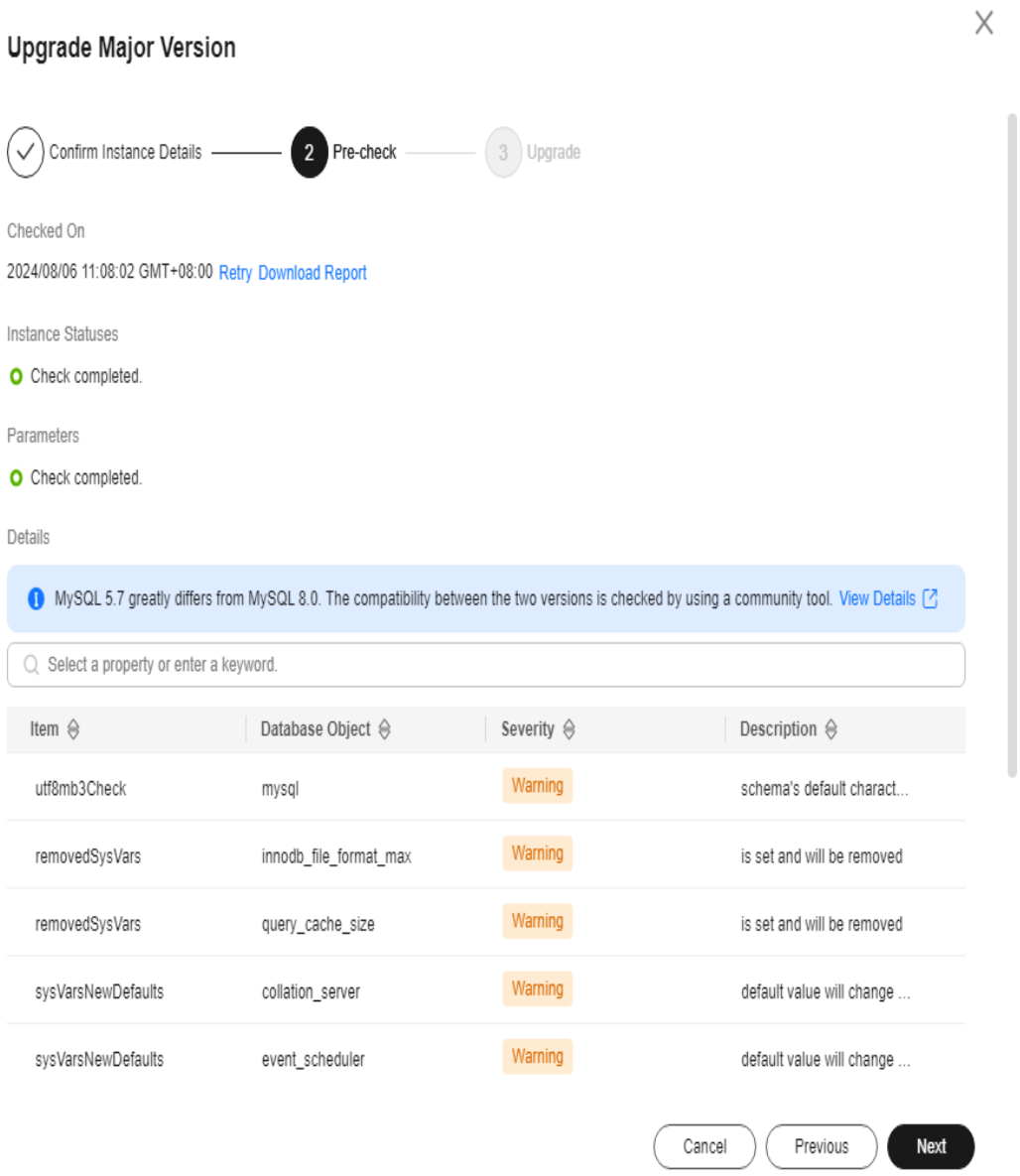
Figure 1-51 Performing a pre-check



After the pre-check is complete, rectify the fault (if any) based on the pre-check results by referring to **Table 1-21**.

After the fault is rectified, click **Retry** to perform a check again until the values of both **Instance Statuses** and **Parameters** are **Check completed**. Then, click **Next**.

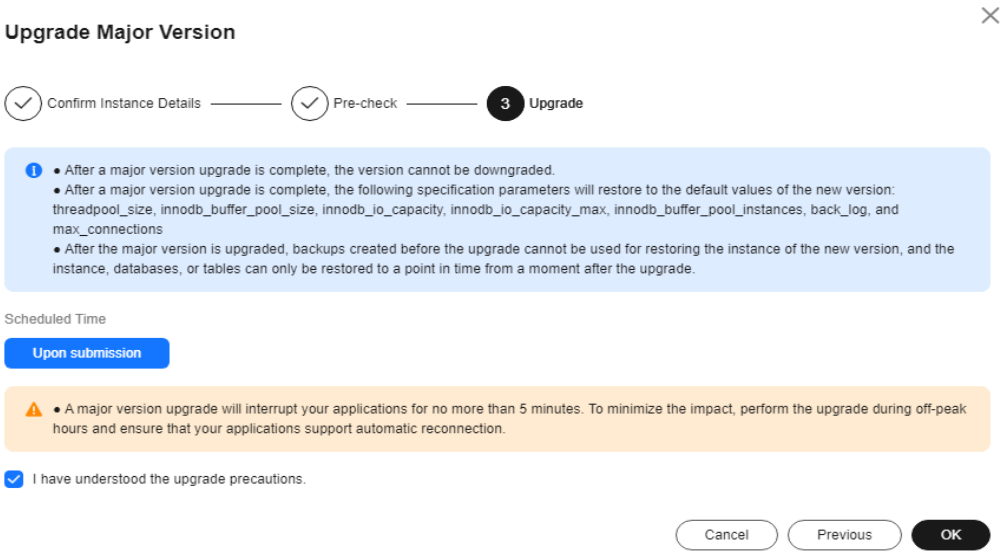
Figure 1-52 Check completed



Step 8 Select a scheduled time and click **OK**.

- **Upon submission**: The system upgrades your instance to the latest version of 5.7 or 8.0 immediately after you submit the upgrade request.
- **In maintenance window**: The system will upgrade your instance to the latest version of 5.7 or 8.0 during the maintenance window you specified. For details about how to configure a maintenance window, see **Changing the Maintenance Window**.

Figure 1-53 Selecting a scheduled time



If the upgrade fails, rectify the fault by referring to **Table 1-22**.

----End

Upgrading a Major Version Using DRS

You can migrate data from an RDS for MySQL 5.6 instance to an RDS for MySQL 5.7 instance using Data Replication Service (DRS). Before the migration, create a DB instance of the target version.

On the **Instances** page, click the instance you want to migrate. On the displayed **Overview** page, click **Migrate Database** in the upper right corner.

For more information, see **Creating a Migration Task** in the *Data Replication Service User Guide*.

Table 1-23 MySQL database version information

Source Database Version	Destination Database Version	Migration Type
RDS for MySQL/Self-managed 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 supports migration only from an earlier version to a later version.

1.6.3 Upgrading an RDS for MySQL Instance from 5.6 to 5.7

Scenarios

You can upgrade your RDS for MySQL instance from 5.6 to 5.7 using either of the following methods:

- **Upgrading a major version on the RDS console**: For details about version functions, see **RDS for MySQL Kernel Version Description**.

  To use this function, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console to apply for required permissions.
- **Upgrading a major version using DRS**: You can migrate instance data from an earlier version to a later version.

Precautions

- You are advised to perform a full backup before upgrading a major version.

  Upgrading a major version will cause a connection interruption for 10 to 120 seconds. Ensure that your applications support automatic reconnection. Perform this operation during off-peak hours because upgrading a major version during peak hours takes much more time.
- When you upgrade the major version of a primary DB instance, the major versions of its read replicas (if any) will also be upgraded. You cannot upgrade the major version of a read replica without upgrading that of the primary instance.
- A major version upgrade cannot be rolled back after the upgrade is complete.
- Before the upgrade, compare the old and new versions carefully. To ensure that the syntax and features of the old version used by your applications are compatible with the new version, create a new RDS for MySQL 5.7 instance to test the syntax before the upgrade.
- You are advised to clone the original instance and use the cloned instance to perform an upgrade check. After confirming that all functions are normal, upgrade the original instance.
- Sufficient storage needs to be reserved to ensure data writes during the upgrade.
- Scheduled major version upgrades need to be prepared in advance and cannot be canceled.
- After a major version upgrade is complete, the backups before the upgrade cannot be used for the instance of the new version, and the time points before the upgrade cannot be selected for point-in-time recovery (PITR).
- DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a major version upgrade.
- After a major version upgrade, specification parameters are reset to the default values of the new version, including **threadpool_size**,

innodb_buffer_pool_size, **innodb_io_capacity**, **innodb_io_capacity_max**, **innodb_buffer_pool_instances**, **back_log**, and **max_connections**.

Constraints

- If the replication delay between primary and standby instances is longer than 300 seconds, the major version cannot be upgraded.
- The major version cannot be upgraded for DB instances with abnormal nodes.
- RDS for MySQL 5.7 and later versions no longer support Sequence Engine.
- RDS for MySQL DB instances support a maximum of 500,000 tables. If the number of tables is greater than 500,000, the major version upgrade may fail.
- RDS for MySQL DB instances with event scheduler enabled do not support major version upgrades. If you want to perform a major version upgrade, disable event scheduler first. For details, see **Enabling or Disabling Event Scheduler**.

Upgrading a Major Version on the RDS Console

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

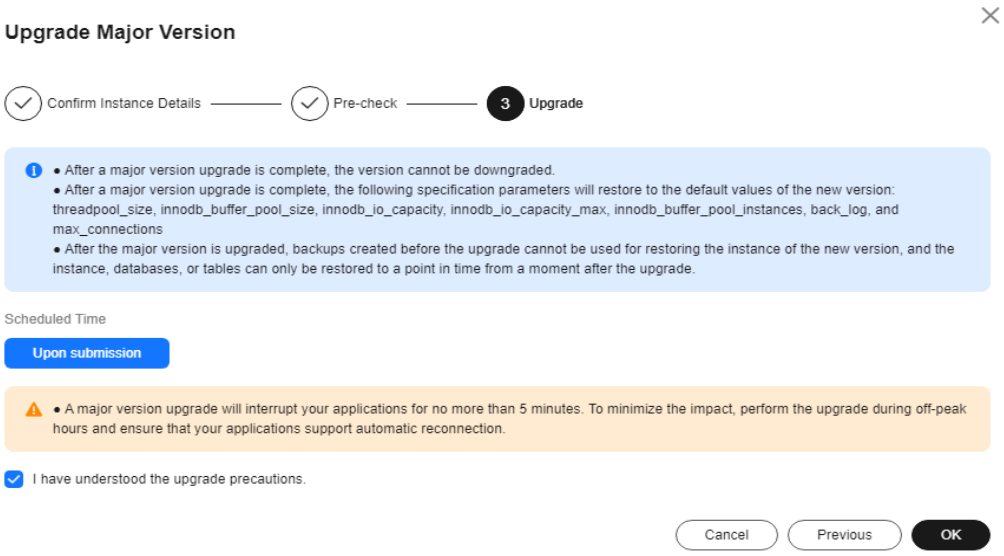
Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **DB Engine Version**, click **Upgrade Engine Version**.

Step 6 In the displayed dialog box, select a scheduled time and click **OK**.

- **Upon submission**: The system upgrades your instance to the latest version of 5.7 immediately after you submit the upgrade request.
- **In maintenance window**: The system will upgrade your instance to the latest version of 5.7 during the maintenance window you specified. For details about how to configure a maintenance window, see **Changing the Maintenance Window**.

Figure 1-54 Selecting a scheduled time



----End

Upgrading a Major Version Using DRS

You can migrate data from an RDS for MySQL 5.6 instance to an RDS for MySQL 5.7 instance using Data Replication Service (DRS). Before the migration, create a DB instance of the target version.

On the **Instances** page, click the instance you want to migrate. On the displayed **Overview** page, click **Migrate Database** in the upper right corner.

For more information, see **Creating a Migration Task** in the *Data Replication Service User Guide*.

Table 1-24 MySQL database version information

Source Database Version	Destination Database Version	Migration Type
RDS for MySQL/Self-managed 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 supports migration only from an earlier version to a later version.

1.7 Instance Management

1.7.1 Instance Overview

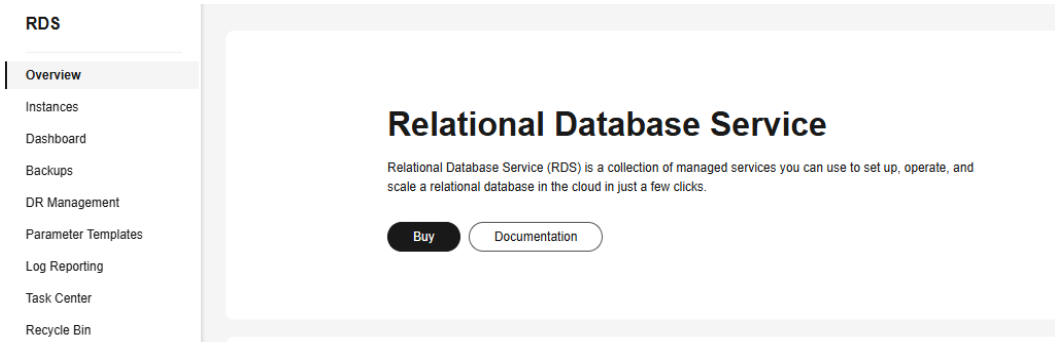
The **Overview** page gives you an overview of your DB instances, including instances by status, alarms, and key performance metrics. It also displays detected exceptions and provides handling suggestions with RDS intelligent diagnosis.

Learning About RDS

If you are new to RDS, you can quickly learn about RDS concepts, common functions, APIs, and how to buy and get started with a DB instance by referring to **RDS Progressive Knowledge**.

1. To purchase an RDS for MySQL instance, click **Buy** on the **Overview** page and select your desired version and specifications. For details, see **Buying an RDS for MySQL DB Instance**.
2. After the purchase is complete, you can view instances by status, alarms, and intelligent diagnosis statistics on the **Overview** page.

Figure 1-55 Overview



Viewing Instances by Status

The statuses of all RDS for MySQL instances under your account are displayed after you select **MySQL** for **Relational Database Service** in the upper part of the **Overview** page.

Figure 1-56 Instances by Status

Table 1-25 Status description

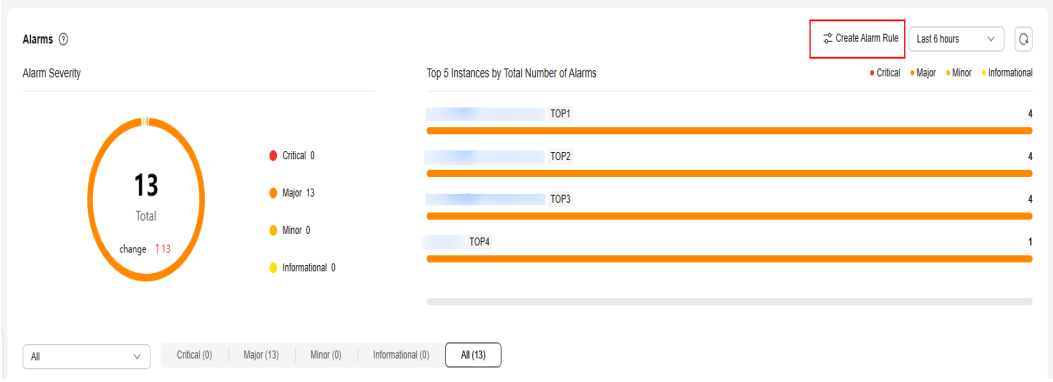
Item	Description	Operation	Solution
Total instances	Total number of DB instances and read replicas in all statuses.	Click **Total instances** to go to the instance list and view all instances.	-
Abnormal	Total number of instances in the **Abnormal** status.	Click **Abnormal** to go to the instance list and view abnormal instances.	**Submit a service ticket**.
Out of storage	Total number of instances in the **Storage full** status.	Click **Out of storage** to go to the instance list and view instances that are out of storage.	For details, see **Full Storage of RDS for MySQL Instances**.
Frozen	Total number of instances in the **Frozen** status.	Click **Frozen** to go to the instance list and view frozen instances.	For details, see **Resource Freezing, Release, Stopping, Deletion, and Unsubscription**.
Pending reboot	Total number of instances in the **Parameter change. Pending reboot** status. **NOTE** Modifications to some parameters require an instance reboot before they can be applied.	Click **Pending reboot** to go to the instance list and view instances waiting to be rebooted.	Reboot the instances.

Viewing Alarms

Based on the configuration of alarm rules, you can see active alarms of all RDS for MySQL instances under your current account, including alarms in the **Alarm** (metric) and **Triggered** (event) states.

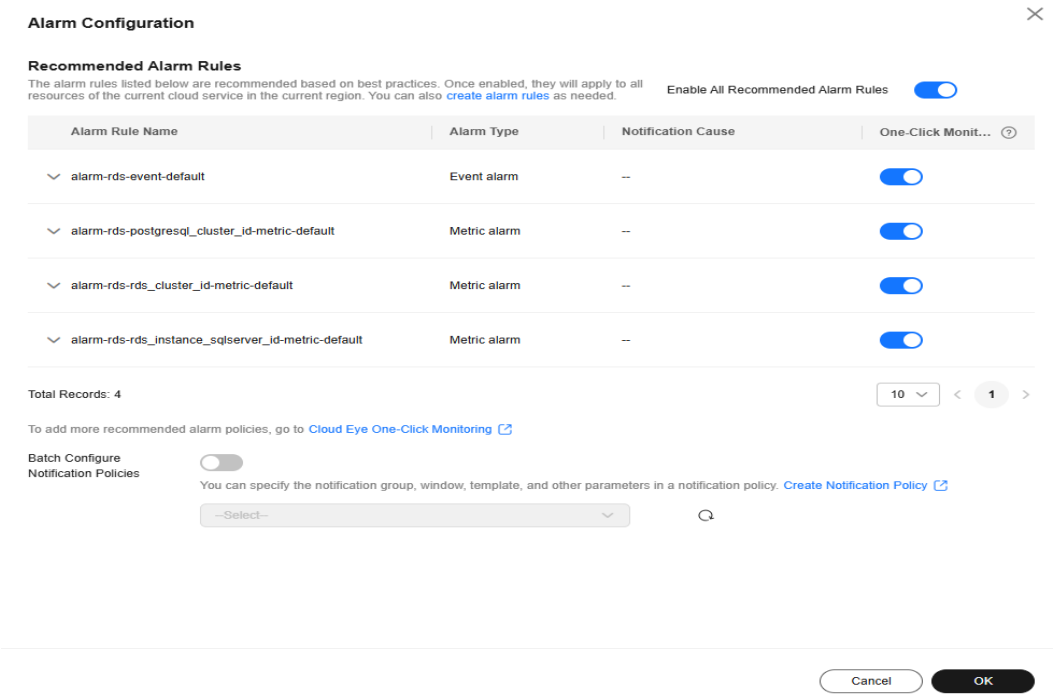
1. In the **Alarms** area, click **Create Alarm Rule**.

Figure 1-57 Create Alarm Rule



2. In the dialog box displayed on the right, toggle on the **Enable All Recommended Alarm Rules** switch. Once this function is enabled, it applies to all resources in the current region.

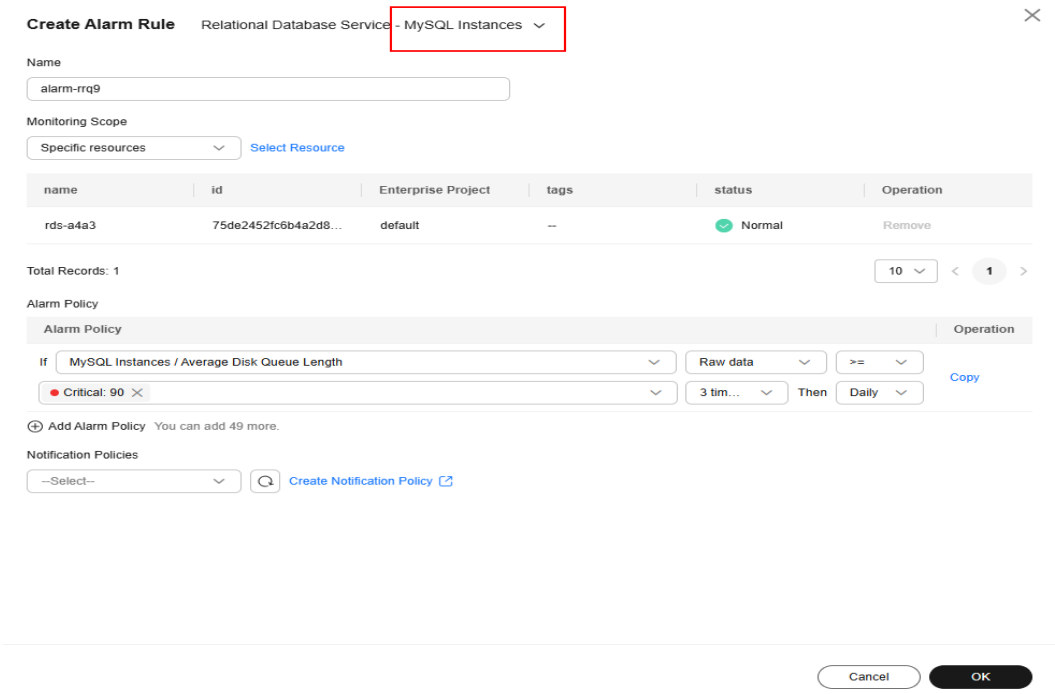
Figure 1-58 Enable All Recommended Alarm Rules



After the function is enabled, you can modify alarm policies and disable alarm rules.

3. If the recommended alarm rules do not meet your requirements, you can create custom ones. Click **create alarm rules** to create alarm rules to monitor metrics or events for your DB instance. For details, see **Setting Alarm Rules**.

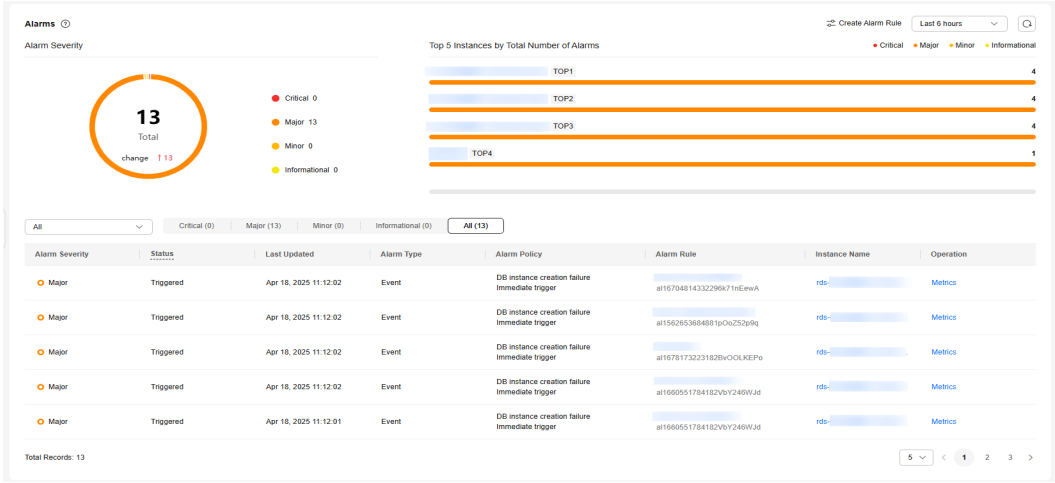
Figure 1-59 Creating an alarm rule



4. To view alarm details, select a time window in the upper part of the **Alarms** area.

   The time window can be **Last 1 hour**, **Last 6 hours**, **Last 12 hours**, **Last day**, **Last week**, or **Last month**.

Figure 1-60 Alarms



Instance alarm details are displayed from multiple dimensions. For details, see **Table 1-26** and **Table 1-27**.

Table 1-26 Alarms

Item	Description
Alarm Severity	Displays the number of alarms at each severity. The alarm severity can be critical, major, minor, or informational.
Top 5 Instances by Total Number of Alarms	Displays alarm statistics of the top 5 instances with the largest number of alarms. Hover over an instance to see the number of its alarms of each severity. You can view alarm details of a specific instance and alarm severity.

Table 1-27 Alarm list

Item	Description
Alarm Severity	There are four severities: critical, major, minor, and informational. Above the alarm list, you can choose to view alarms of a specific severity or all severities.
Status	• **Alarm**: A metric value has reached the preset alarm threshold, triggering an alarm for the resource, but the alarm has not been cleared. • **Triggered**: An event configured in the alarm policy triggered an alarm.
Last Updated	The latest time when the alarm was triggered.
Alarm Type	The alarm type that the alarm rule applies to. • **Metric** • **Event**

Item	Description
Alarm Policy	The policy for triggering an alarm. • If you set **Alarm Type** to **Metric**, the system triggers an alarm if the metric value hits the preset threshold in consecutive periods. For example, an alarm is triggered if the average CPU usage is 80% or higher for three consecutive 5-minute periods. – For details about how to troubleshoot high CPU usage, see **High CPU Usage of RDS for MySQL Instances**. – For details about how to troubleshoot high memory usage, see **High Memory Usage of RDS for MySQL Instances**. – For details about how to troubleshoot full storage, see **Full Storage of RDS for MySQL Instances**. • If you set **Alarm Type** to **Event**, the event that triggers an alarm is an instant operation. For example, if an instance fails to be created, an alarm is triggered. For details about the supported events and handling suggestions, see **Events Supported by Event Monitoring**.
Alarm Rule	The name or ID of the alarm rule.
Instance Name	The name of the instance that triggers the alarm. You can select **All** or a specific instance from the drop-down list to view the alarm details. Refresh the page to display details of the latest triggered alarms in real time.
Operation	Click **Metrics**. In the dialog box displayed on the right, you can see the metric changes in the selected time window.

Viewing Intelligent Diagnosis

Intelligent Diagnosis checks instance health using instance operation data and intelligent algorithms and provides diagnosis results and suggestions.

1. To view problematic instances and related metrics, click the name of a diagnosis item.
2. To view detailed diagnosis results and optimization suggestions, click **Diagnosis Details**.

Figure 1-61 Intelligent Diagnosis

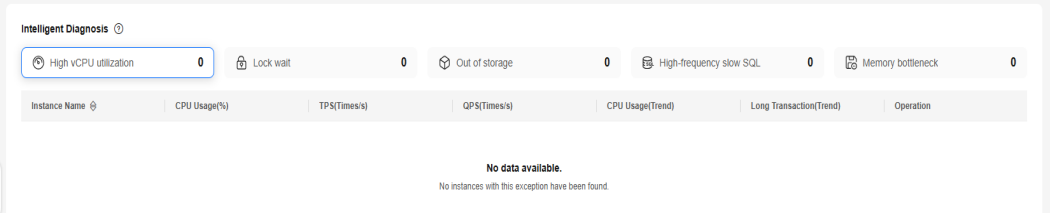


Table 1-28 provides the supported diagnosis items and handling suggestions.

Table 1-28 Intelligent diagnosis details

Diagnosis Item	Metric	Metric Description	Solution	Practices
High vCPU utilization	CPU Usage(%)	CPU usage of the monitored object	• Evaluate the SQL execution plan and add indexes to avoid full table scanning. • Upgrade vCPU specifications for compute-intensive workloads.	**High CPU Usage of RDS for MySQL Instances**
	TPS(Times/s)	Execution times of submitted and rollback transactions per second		
	QPS(Times/s)	Query times of SQL statements (including stored procedures) per second		
	CPU Usage(Trend)	CPU usage of the monitored object		
	Long Transaction(Trend)	Maximum duration for starting a transaction A complete long transaction is counted only when the **BEGIN** and **COMMIT** commands exist before and after the related operation commands, respectively.		

Diagnosis Item	Metric	Metric Description	Solution	Practices
Lock wait	Row Locks Waits Transactions(Counts)	Number of InnoDB row lock waits This metric indicates the total number of historical transactions waiting for row locks. Lock waits will be cleared after the instance is rebooted.	• Metadata locks: Terminate sessions with metadata locks to resume blocked operations. • InnoDB locks: Check whether there are lock waits. Terminate the source sessions with locks and resume blocked operations. For details, see **Managing Locks & Transactions**.	• **RDS for MySQL Metadata Locks** • **Slow Response Due to Deadlocks**
	Average Row Lock Wait Time(ms)	Average wait time of historical InnoDB row locks		
	Current Row Lock Waits(Counts)	Number of current InnoDB row lock waits This metric indicates the number of transactions that are currently waiting for row locks.		
	MDL Locks(Count)	Number of metadata locks		
	Long Transaction(Trend)	Maximum duration for starting a transaction A complete long transaction is counted only when the **BEGIN** and **COMMIT** commands exist before and after the related operation commands, respectively.		

Diagnosis Item	Metric	Metric Description	Solution	Practices
Out of storage	Storage Space Usage(%)	Storage space usage of the monitored object	• Scaling up storage space: You can configure storage autoscaling. When the storage usage reaches the threshold, autoscaling is triggered. • Reducing disk data: Delete useless historical data. • If temporary files generated by sorting queries occupy too much storage space, optimize your SQL query statements.	**Full Storage of RDS for MySQL Instances**
	Used Storage Space(GB)	Used storage space of the monitored object		
	Total Storage Space(GB)	Total storage space of the monitored object		
	Storage Space Usage(Trend)	Storage space usage of the monitored object		

Diagnosis Item	Metric	Metric Description	Solution	Practices
High-frequency slow SQL	Slow Query Logs(Count/min)	Number of MySQL slow query logs generated per minute	• Optimize slow SQL statements based on the execution plan. • Upgrade vCPU specifications for compute-intensive workloads.	**Slow SQL Statements Due to Improper Composite Index Settings**
	Slow Query Logs(Trend)	Number of MySQL slow query logs generated per minute		
Memory bottleneck	Memory Usage(%)	Memory usage of the monitored object	• Upgrade instance specifications. • Optimize SQL statements to reduce the use of temporary tables. • Reconnect sessions at a specific interval to release memory of the sessions.	• **Out of Memory (OOM) Errors** • **High Memory Usage of RDS for MySQL Instances**
	Total Connections(Count)	Total number of connections that attempt to connect to the MySQL server		
	Current Active Connections(Count)	Number of connections that are not in the sleep state		
	Memory Usage(Trend)	Memory usage of the monitored object		

1.7.2 Monitoring Dashboard

Cloud Eye monitors operating statuses of DB instances. RDS allows you to view real-time performance metrics and historical metrics of all RDS for MySQL instances under your account. This helps you identify abnormal instances and take actions in a timely manner.

Viewing Real-Time Metrics

1. On the **Dashboard** page, select **MySQL** for **Monitoring Dashboards** to view the real-time performance metrics of RDS for MySQL instances under your account.
2. You can click  in the metric columns to sort metrics by size.

Table 1-29 List description

Item	Description
Instance Name/ID	Only monitoring data of created DB instances is displayed. You can click an instance name to go to the **Overview** page of the instance.
Instance Type	The following types are available: • **Single-node** • **Primary/Standby** • **Read replica**
DB Engine Version	All RDS for MySQL versions can be displayed.
Status	The following statuses are available: • **Normal**: Real-time monitoring data is displayed. **NOTE** The monitoring data and graphics are available for a new instance after the instance runs for about 10 minutes. • **Abnormal**: There is no monitoring data. The default values for all metrics are 0. The monitoring data is available only after the instance becomes normal. • **Stopped**: There is no monitoring data. The default values for all metrics are 0. The monitoring data is available only after the instance is started.

Item	Description
Monitoring Metrics	For details about metric description and handling suggestions for abnormal metrics, see **Table 1-30**. The following metrics are available: • CPU Usage (%) • Memory Usage (%) • Storage Usage (%) • TPS • QPS • IOPS • Active Connections • Slow SQL Statements

Table 1-30 Monitoring metrics

Metric	Description	Solution	Practices
CPU Usage (%)	CPU usage of the monitored object	• Evaluate the SQL execution plan and add indexes to avoid full table scanning. • Upgrade vCPU specifications for compute-intensive workloads.	**High CPU Usage of RDS for MySQL Instances**

Metric	Description	Solution	Practices
Memory Usage (%)	Memory usage of the monitored object	• Upgrade instance specifications. • Optimize SQL statements to reduce the use of temporary tables. • Reconnect sessions at a specific interval to release memory of the sessions.	• **Out of Memory (OOM) Errors** • **High Memory Usage of RDS for MySQL Instances**

Metric	Description	Solution	Practices
Storage Usage (%)	Storage space usage of the monitored object	• Scaling up storage space: You can configure storage autoscaling. When the storage usage reaches the threshold, autoscaling is triggered. • Reducing disk data: Delete useless historical data. • If temporary files generated by sorting queries occupy too much storage space, optimize your SQL query statements.	**Full Storage of RDS for MySQL Instances**
TPS	Execution times of submitted and rollback transactions per second	• Evaluate the SQL execution plan and add indexes to avoid full table scanning. • Upgrade vCPU specifications for compute-intensive workloads.	**High CPU Usage of RDS for MySQL Instances**
QPS	Query times of SQL statements (including stored procedures) per second		

Metric	Description	Solution	Practices
IOPS	Average number of I/O requests processed by the system in a specified period	• Upgrade instance specifications. • To reduce data reads from the disk, optimize the application to make it read data from the buffer first.	**Insufficient Disk Bandwidth**
Active Connections	Number of connections that are not in the sleep state	• Check whether applications are connected to the instance, optimize the connections, and release unnecessary connections. • Upgrade the specifications if needed.	• **What Do I Do If the Number of RDS Database Connections Reaches the Upper Limit?** • **What Is the Maximum Number of Connections to an RDS DB Instance?**
Slow SQL Statements	Number of MySQL slow query logs generated per minute	• Optimize slow SQL statements based on the execution plan. • Upgrade vCPU specifications for compute-intensive workloads.	**Slow SQL Statements Due to Improper Composite Index Settings**

Viewing Historical Metrics

You can select multiple instances from the monitoring list and click **View history monitor** to view metric trend charts of the instances in the **Historical Metrics** area.

- You can view metric changes of up to 10 instances at a time.
- The following monitoring time windows are supported: last 1 hour, last 3 hours, last 12 hours, last 24 hours, last 7 days, and a custom time period.
- For details about monitoring metrics and handling suggestions for abnormal metrics, see **Table 1-30**.
- If the real-time replication delay metric is abnormal, see **Primary/Standby Replication Delay Scenarios and Solutions**.

1.7.3 Instance Lifecycle

1.7.3.1 Buying a Same DB Instance as an Existing DB Instance

Scenarios

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

NOTE

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

Procedure

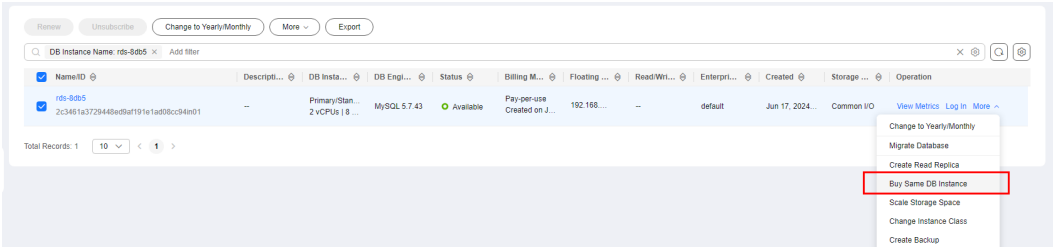
Step 1 **Log in to the management console**.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Figure 1-62 Buying a same instance



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 RDS for MySQL DB instance configurations, see **Buying an RDS for MySQL DB Instance**.

Step 6 Confirm the instance specifications.

- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**.

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 **Instances** page.

----End

1.7.3.2 Stopping an Instance

Scenarios

If you use DB instances only for routine development, you can temporarily stop pay-per-use instances to save money.

Billing

After a DB instance is stopped, the VM where the DB instance is located is no longer billed. Other resources, including EIPs, storage resources, database proxies, and backups, are still billed.

Constraints

- Only pay-per-use instances using cloud SSDs or extreme SSDs can be stopped. RDS DB instances created in a **Dedicated Computing Cluster (DCC)** cannot be stopped.
- A stopped instance will not be moved to the recycle bin after being deleted.
- Stopping a DB instance will also stop its automated backups. After the DB instance is started, a full backup is automatically triggered.
- If you stop a primary instance, read replicas (if there are any) will also be stopped. Both the primary instance and read replicas can be stopped for up to 15 days. You cannot stop a read replica without stopping the primary instance.
- If you do not manually start your stopped DB instance after 15 days, your DB instance is automatically started during the next maintenance window. For details about the maintenance window, see **Changing the Maintenance Window**. To start an instance, see **Starting an Instance**.
- A stopped pay-per-use instance may fail to start due to insufficient ECS resources. In this case, try again later or **restore data to a new DB instance** using the latest backup. If you need assistance, **submit a service ticket**.
- A DB instance can fail to be stopped during peak hours. You are advised to stop instances during off-peak hours.

Procedure

Step 1 Log in to the management console.

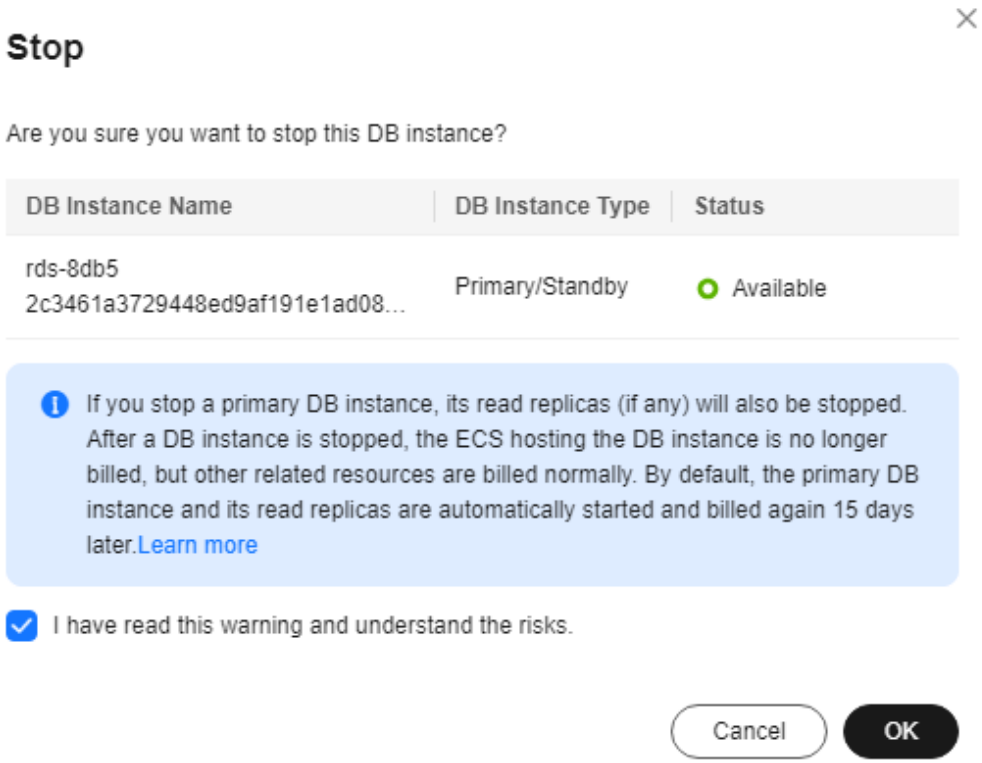
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary instance that you want to stop and choose **More** > **Stop** in the **Operation** column.

Step 5 In the displayed dialog box, select **I have read this warning and understand the risks.** and click **OK**.

Figure 1-63 Stopping an instance



Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the instance list and view the status of the instance. If the status is **Stopped**, the instance is stopped successfully.

----End

1.7.3.3 Starting an Instance

Scenarios

You can stop your instance temporarily to save money. After stopping your instance, you can restart it to begin using it again.

Billing

After a DB instance is started, the VM where the DB instance is located is billed again.

Constraints

- If you start a primary instance, read replicas (if there are any) will also be started.
- Only DB instances in **Stopped** state can be started.
- When a stopped DB instance is started, a full backup is automatically triggered.
- A stopped pay-per-use instance may fail to start due to insufficient ECS resources. In this case, try again later or **restore data to a new DB instance** using the latest backup. If you need assistance, **submit a service ticket**.

Procedure

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

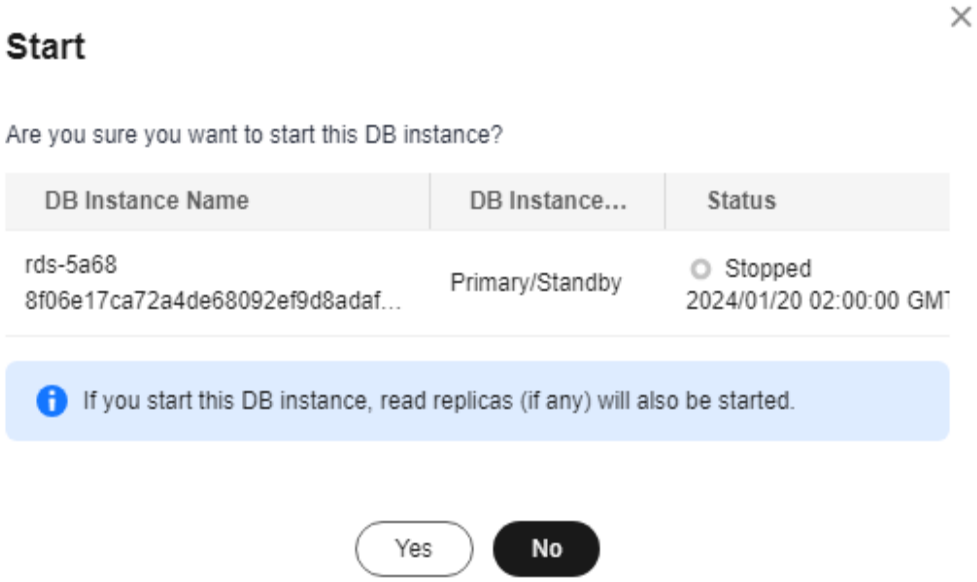
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary instance that you want to start and choose **More** > **Start** in the **Operation** column.

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

Figure 1-64 Starting an instance



Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the instance list and view the status of the instance. If the status is **Available**, the instance is started successfully.

----End

1.7.3.4 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.

- After a primary/standby DB instance is rebooted, it takes about one minute to establish the replication relationship. During this period, some operations, such as changing the instance class, cannot be performed.
- A reboot task configured during the current maintenance window will not be executed until the next maintenance window.

Rebooting a DB Instance or Read Replica

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **Instances** page to go to the **Overview** page. In the upper right corner, click **Reboot**.

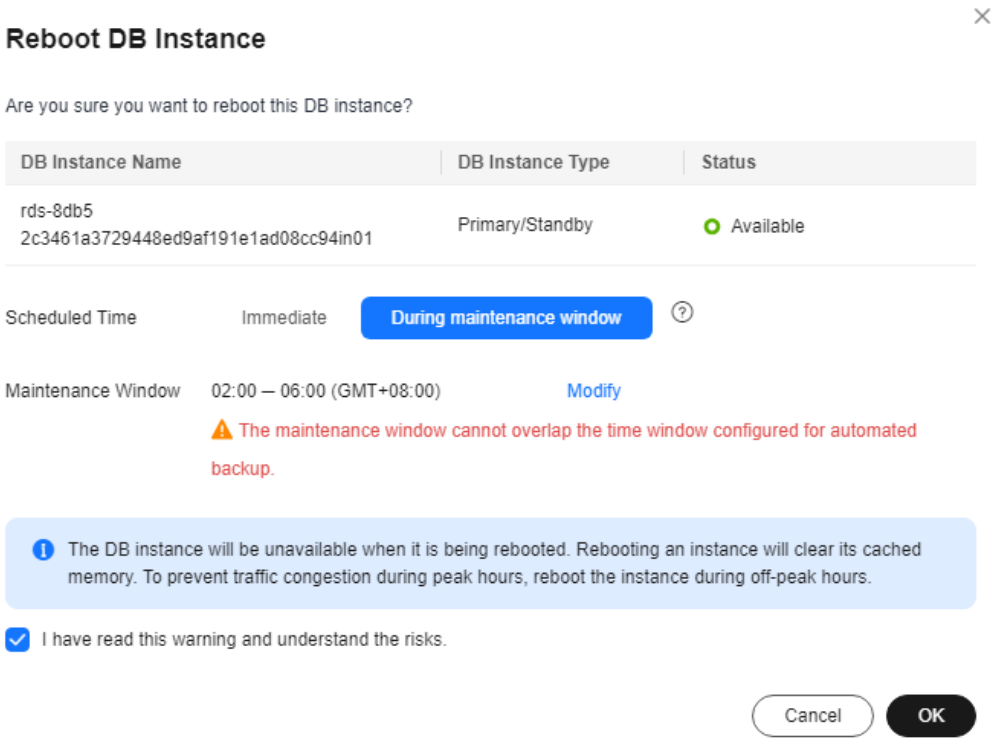
For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 5 In the displayed dialog box, select a scheduled time, select the check box, and click **OK**.

- **Immediate**: RDS reboots the instance immediately.
- **During maintenance window**: RDS will reboot the instance during the maintenance window you configured.

  If you select **During maintenance window**, you can further click **Modify** under the option to change the maintenance window to a preferred time.

Figure 1-65 Rebooting an instance as scheduled



Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

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

----End

Rebooting DB Instances or Read Replicas in Batches

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select one or more DB instances or read replicas (maximum: 50) to be rebooted and choose **More** > **Reboot** above the DB instance list.

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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list and view the statuses of the DB instances. If their statuses are **Available**, they have rebooted successfully.

----End

Follow-up Operations

Return to the instance list. In the navigation pane on the left, choose **Task Center** and check the progress of the reboot task.

- If you have selected **Immediate** for **Scheduled Time**:

  On the **Instant Tasks** page, search for **Rebooting a MySQL DB instance** and check the execution progress. Instant tasks cannot be canceled.
- If you have selected **During maintenance window** for **Scheduled Time**:

  On the **Scheduled Tasks** page, search for the instance ID and check the execution status of the reboot task.

  If the task is in the **To be executed** state, you can click **Cancel** to cancel the task.

For details, see **Viewing a Task**.

1.7.3.5 Selecting Displayed Items

Scenarios

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

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

- **Table Text Wrapping**: If you enable this function, excess text will move down to the next line.
- **Operation Column**: If you enable this function, the **Operation** column is always fixed at the rightmost position of the table.
- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Disk Encryption** (**submit a service ticket** to apply for required permissions), **Billing Mode**, **Floating IP Address**,

Private Domain Name, **IPv6 Address**, **Read/Write Splitting Address**, **Proxy ID**, **Enterprise Project**, **Created**, **Database Port**, **Storage Type**, **Tags**, and **Operation**.

----End

1.7.3.6 Exporting DB Instance Information

Scenarios

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

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click **Export** above the DB instance list. By default, information about all DB instances is exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, DB instance tag, enterprise project, or floating IP address, or select the DB instances to be exported, and click **Export** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

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

----End

1.7.3.7 Deleting Pay-per-Use DB Instances or Read Replicas

Scenarios

To release resources, you can delete DB instances or read replicas billed on the pay-per-use basis as required on the **Instances** page.

Billing

- You will not be billed for the instances that were not successfully created.
- If you delete a pay-per-use DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be deleted.
- A maximum of 50 pay-per-use DB instances can be deleted at a time.
- 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 **rebuild a DB instance** that was deleted up to 7 days ago from the recycle bin.
- You can use a manual backup to restore a DB instance. For details, see **Restoring a DB Instance from Backups**.

Deleting a Pay-per-Use DB Instance

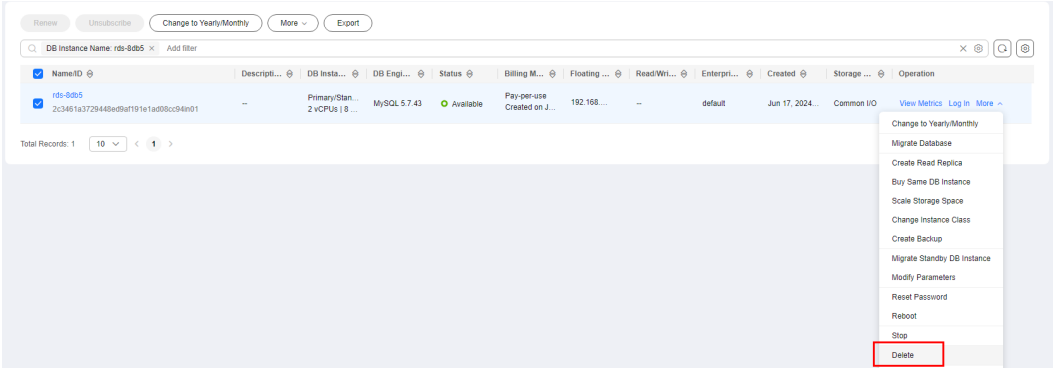
Step 1 **Log in to the management console**.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary DB instance to be deleted and choose **More** > **Delete** in the **Operation** column.

Figure 1-66 Deleting a DB instance



Step 5 In the displayed dialog box, enter **delete**, select **Yes** for **Confirm**, select **I have read this warning and understand the risks.**, and click **Yes** to deliver the request.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Pay-per-Use Read Replica

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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, enter **delete**, select **Yes** for **Confirm**, select **I have read this warning and understand the risks.**, and click **Yes** to deliver the request.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 8 Refresh the DB instance list later to check that the deletion is successful.

----End

Deleting Pay-per-Use DB Instances in Batches

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the RDS for MySQL pay-per-use DB instances to be deleted and choose **More** > **Delete** above the DB instance list.

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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list later to check that the deletion is successful.

----End

1.7.3.8 Recycling a DB Instance

Scenarios

RDS allows you to move unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances to the recycle bin. You can rebuild a DB instance that was deleted up to 7 days ago from the recycle bin.

If resources are not renewed after expiration, you can rebuild DB instances from the recycle bin to restore data.

Constraints

- The recycle bin is free for use.
- Read replicas cannot be moved to the recycle bin.
- A stopped instance will not be moved to the recycle bin after being deleted.
- The recycle bin is enabled by default and cannot be disabled.
- After you submit a deletion request for your DB instance, a full backup will be performed. You can rebuild the DB instance only after the full backup is complete.

Modifying Recycling Policy

NOTICE

Instances in the recycle bin are retained for 7 days by default. 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.

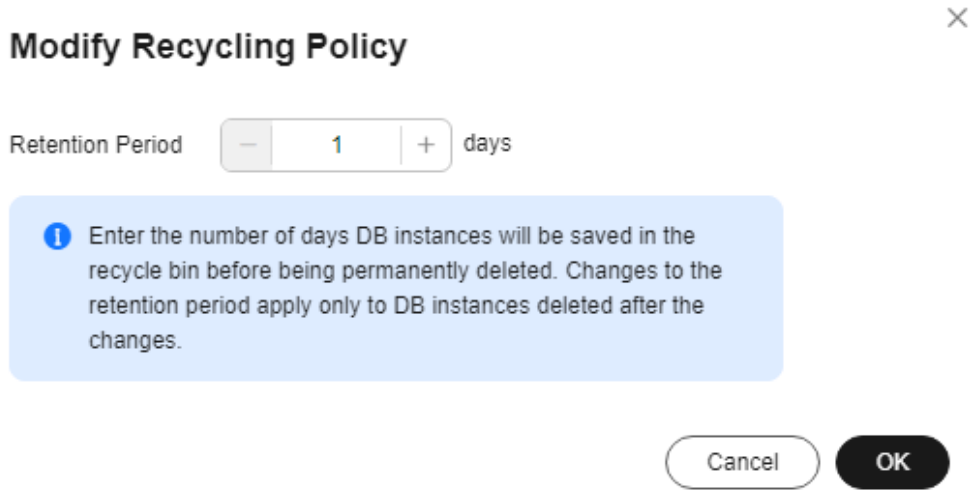
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances to 1 to 7 days.

Step 6 Then, click **OK**.

Figure 1-67 Modifying the recycling policy



----End

Rebuilding a DB Instance

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

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 On the **Recycle Bin** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.

Step 6 On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see **Restoring a DB Instance from Backups**.

----End

1.8 Instance Modifications

1.8.1 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **DB Instance Name**, click to edit the instance name.

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

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

Step 5 View the results on the **Overview** page.

----End

1.8.2 Changing a DB Instance Description

Scenarios

After a DB instance is created, you can add a description.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance you wish to edit the description for and click in the **Description** column to make your modification. When you are finished, click **OK**.

Alternatively, click the instance name to go to the **Overview** page. Under **Description**, click to edit the instance description.

NOTE

The DB instance description can include up to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.).

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

Step 5 View the results on the **Overview** page.

----End

1.8.3 Changing the Replication Mode

Scenarios

You can change the replication mode for primary/standby DB instances to **Asynchronous** or **Semi-synchronous**.

- **Asynchronous**:
  - When applications update data, the primary DB instance responds to the applications immediately after data is updated. This mode provides better performance than the semi-synchronous mode.
- **Semi-synchronous** (default value):
  - When applications update data, the primary DB instance responds to the applications only after the standby DB instance receives logs, which affects database performance.
  - If the standby DB instance is abnormal, the primary DB instance waits for the response of the standby DB instance for several seconds and does not respond to write operations during this period.

- If the standby DB instance is recovered during the waiting period, the primary DB instance starts to respond to write operations normally.
- If the standby DB instance is not recovered during the waiting period, the replication mode is automatically switched to asynchronous. After the switchover is complete, the primary DB instance starts to respond to write operations.

Procedure

Step 1 Log in to the management console.

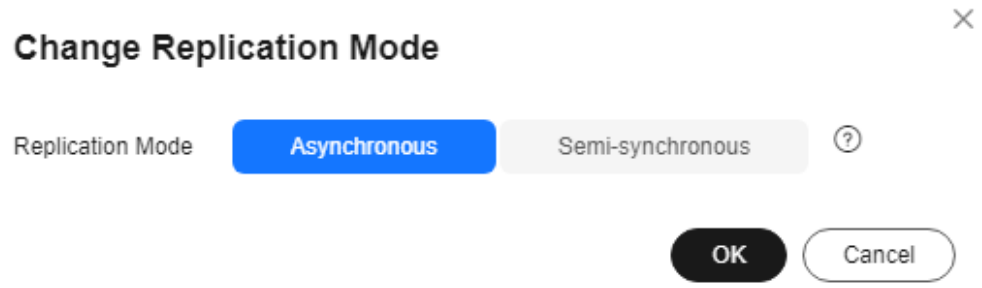
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 On the displayed **Overview** page, find **Replication Mode** and click **Configure** under it. In the displayed dialog box, select a mode and click **OK**.

Figure 1-68 Changing the replication mode



Step 6 On the **Overview** page, check for the new replication mode.

----End

1.8.4 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose highest priority is data consistency. In extreme scenarios, there may be a small amount of data lost.
- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Constraints

The failover priority cannot be changed when the DB instance is stopped or its instance class is being changed.

Procedure

Step 1 Log in to the management console.

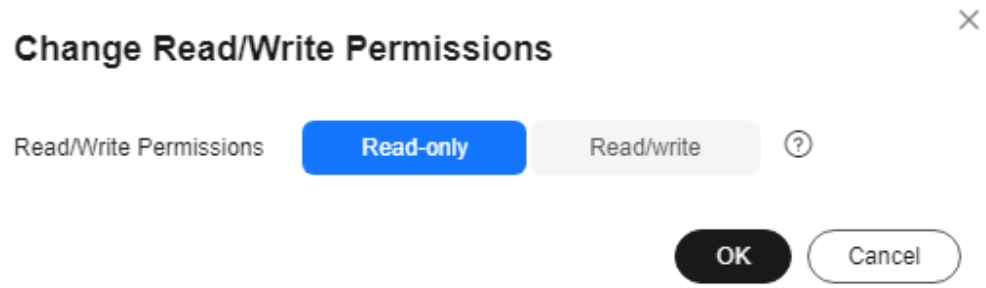
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 On the displayed **Overview** page, find **Failover Priority** and click **Configure** under it. In the displayed dialog box, select a priority and click **OK**.

Figure 1-69 Changing the failover priority



Step 6 View the results on the **Overview** page.

----End

1.8.5 Changing Read/Write Permissions

Scenarios

RDS for MySQL allows you to change the read/write permissions of your instance to meet different workload requirements. You can select **Read-only** or **Read/write** for **Read/Write Permissions**.

- **Read-only**

  If **Read-only** is selected, data in the DB instance cannot be modified anymore. You can set **Read/Write Permissions** to **Read-only** even for a DB instance that is already read-only due to full storage. In this case, after the instance becomes available, it is still read-only.

- **Read/write**

  If **Read/write** is selected, the DB instance becomes readable and writable. You can set **Read/Write Permissions** to **Read/write** even for a DB instance

that is read-only due to full storage. In this case, only after the instance becomes available, it is readable and writable.

Constraints

- This function is only available for single and primary/standby DB instances.
- Read/write permissions cannot be changed when the instance is in any of the following statuses: creating, changing instance class, frozen, or abnormal.
- This function is available for open beta testing (OBT) in some regions. If this function is not available in your region, **submit a service ticket** to request permissions.

Precautions

- Before setting **Read/Write Permissions** to **Read-only**, to ensure data consistency, ensure that no data is being written to the database.
- If the DB instance is abnormal (except when the storage is full), it cannot be set to read-only.
- If your instance is set to read-only, you can still perform operations on the **Accounts** and **Databases** pages on the console.
- If your instance is set to read-only, non-administrator users cannot write data to it anymore. When your instance is processing large transactions or DDL requests, changing it to read-only may fail due to timeout.
- If your RDS instance is associated with a DDM instance, changing it to read-only will affect DDM instance functions.
- If you have selected **Read/write** but the DB instance is still read-only, check whether the DB instance is involved in an ongoing DRS migration task or if the instance storage is full.
- If your instance becomes read-only for other reasons (such as full storage and DRS migration), it cannot be changed to readable and writable by setting **Read/Write Permissions** to **Read/write**.
- This function configures read/write permissions only for primary DB instances.

Procedure

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

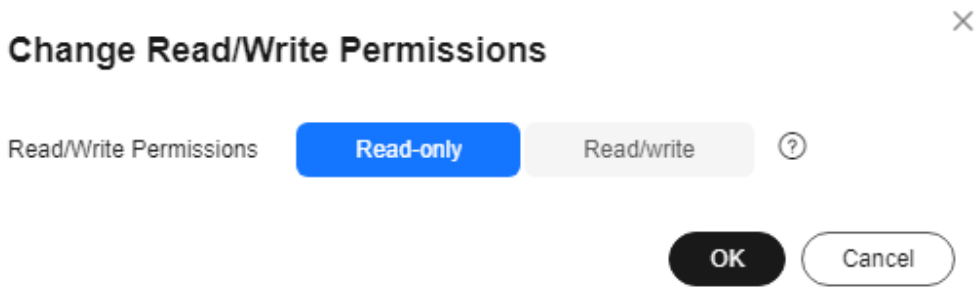
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Overview** page, find **Read/Write Permissions** and click **Configure** under it. In the displayed dialog box, select **Read-only** or **Read/write** as required and click **OK**.

Figure 1-70 Changing read/write permissions



----End

1.8.6 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 events 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 is performed, the event scheduler setting 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 scheduler setting is the same as that of the original DB instance.
- After a single DB instance is changed to a primary/standby DB instance, the event scheduler setting is the same as that of the primary DB instance.

Constraints

- To use this function, your RDS for MySQL kernel version must be at least 5.6.43.2, 8.0.17.4, or 5.7.25.2. If your database version does not meet this requirement, you can **upgrade the minor version**.
- 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Overview** page, find **Event Scheduler** and click **Enable** under it.

NOTICE

After the event scheduler is enabled, reactivate the previously created events (that is, re-execute the event scripts) to ensure that the event statuses on the primary and standby instances are the same.

----End

Disabling Event Scheduler

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Overview** page, find **Event Scheduler** and click **Disable** under it.

----End

FAQ

What Should I Do If I Do Not Have the Permission to Change the event_scheduler Settings?

Answer: You can only enable or disable the event scheduler on the console. For details, see this section.

1.8.7 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPUs and memory) of a DB instance as required.

RDS for MySQL supports both an increase and a decrease in the instance class.

Constraints

Table 1-31 Constraints

Phase	Constraints
Before the change	• You can change the DB instance class only when your account balance is greater than or equal to $0 USD. • Changing the instance class will temporarily occupy IP addresses, and the IP addresses will not be released until 12 hours later by default. Before changing an instance class, ensure that there are available **floating IP addresses** (two for a primary/standby instance and one for a single instance or read replica) in the subnet. Otherwise, the change will fail. • An instance class can fail to be changed when: – The instance status is **Abnormal**, **Creation failed**, **Storage full**, or **Creating**. – The replication delay between the primary instance and standby instance is greater than 300 seconds, or the replication relationship is abnormal. – The standby instance is being rebuilt. – The database proxy is abnormal or being changed. – The instance has a scheduled instance class change task.
During the change	• If there are any large transactions being processed during an instance class change, the change may fail. • Any session that holds a global read lock can cause a change failure. • An instance cannot be deleted while its instance class is being changed. • The following operations cannot be performed on an instance whose instance class is being changed: rebooting the instance, scaling up storage space, modifying the parameter template, creating a manual backup, creating a database account, and creating a database. • If you change the instance class of a read replica, binlog clearing will be disabled for the primary DB instance. Pay attention to the storage space usage.
After the change	• After an instance class is changed, some parameters are automatically changed to the default values defined in the new instance class. For details, see **Parameter Changes**.

Change Duration and Impact on Workloads

- Changing an instance class will interrupt services for about 10 to 120 seconds. Ensure that your applications support automatic reconnection. Perform this

operation during off-peak hours because changing an instance class during peak hours takes much more time.

- The time required for changing an instance class (during off-peak hours) is as follows:
  - For an instance using a cloud disk, the process takes 5 to 15 minutes.
  - If the change takes an extended period of time, contact customer service.
- If you choose to change an instance class during the maintenance window, after the request is submitted, you can locate the scheduled task and click **Execute Now** in the **Operation** column. (If this option is not displayed on the console, **submit a service ticket**.) The time required for changing an instance class can be a little bit long during peak hours due to high replication delay.

Instance Class Type Change

You can change a general-purpose instance to a dedicated instance, and vice versa. However, changing a dedicated instance to a general-purpose instance may compromise database performance. Exercise caution when performing this operation.

Billing

Table 1-32 Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Instance class upgrade	After an instance class is upgraded, the new instance class takes effect in the original usage period. You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month RDS for MySQL 5.7 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The instance price was $59.56 USD per month. On June 15, 2023, you changed the instance class to 4 vCPUs | 8 GB. The instance price became $121.56 USD per month. **Price difference of upgrade = Price for the new instance class × Remaining period - Price for the original instance class × Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Price difference of the upgrade = $121.56 USD x 0.5 - $59.56 USD x 0.5 = $31 USD

Billing Mode	Operation	Impact on Fees
	Instance class downgrade	After an instance class is downgraded, the new instance class takes effect in the original usage period. RDS refunds the difference in price based on the remaining period. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month RDS for MySQL 5.7 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The instance price was $59.56 USD per month. On June 15, 2023, you changed the instance class to 2 vCPUs | 4 GB. The instance price became $50.56 USD per month. **Refunded fees = Price for the original instance class × Remaining period - Price for the new instance class × Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and refunded fees are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Refunded fees = $59.56 USD x 0.5 - $50.56 USD x 0.5 = $4.5 USD
Pay-per-use	Instance class upgrade	After an instance class is changed, the new instance class is billed by hour. For details, see **Product Pricing Details**.
	Instance class downgrade	

Parameter Changes

vCPU-related parameters, such as **threadpool_size** and **slave_parallel_workers**, will be reset according to the following rules during an instance class change.

Table 1-33 Parameter value changes with vCPU changes

Scenario	Rule for a vCPU Parameter with No Changes Made to Its Value	Rule for a vCPU Parameter Changed to a Custom Value
vCPU increase	The parameter will be reset to the default value of the new instance class.	The larger one between the custom value and the default value of the new instance class will be used.
vCPU decrease	The parameter will be reset to the default value of the new instance class.	The smaller one between the custom value and the default value of the new instance class will be used.

Memory-related parameters, such as **innodb_buffer_pool_size**, **innodb_log_buffer_size**, **innodb_log_files_in_group**, **max_connections**, **innodb_page_cleaners**, **innodb_buffer_pool_instances**, and **back_log**, will be reset according to the following rules during an instance class change.

Table 1-34 Parameter value changes with memory changes

Scenario	Rule for a Memory Parameter with No Changes Made to Its Value	Rule for a Memory Parameter Changed to a Custom Value
Memory increase	The parameter will be reset to the default value of the new instance class.	The larger one between the custom value and the default value of the new instance class will be used.

Scenario	Rule for a Memory Parameter with No Changes Made to Its Value	Rule for a Memory Parameter Changed to a Custom Value
Memory decrease	The parameter will be reset to the default value of the new instance class.	The smaller one between the custom value and the default value of the new instance class will be used.

However, values of **innodb_io_capacity** and **innodb_io_capacity_max** will be reset to the default values of the new instance class if no custom values have been specified for them or they will remain unchanged if you have specified custom values for them.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

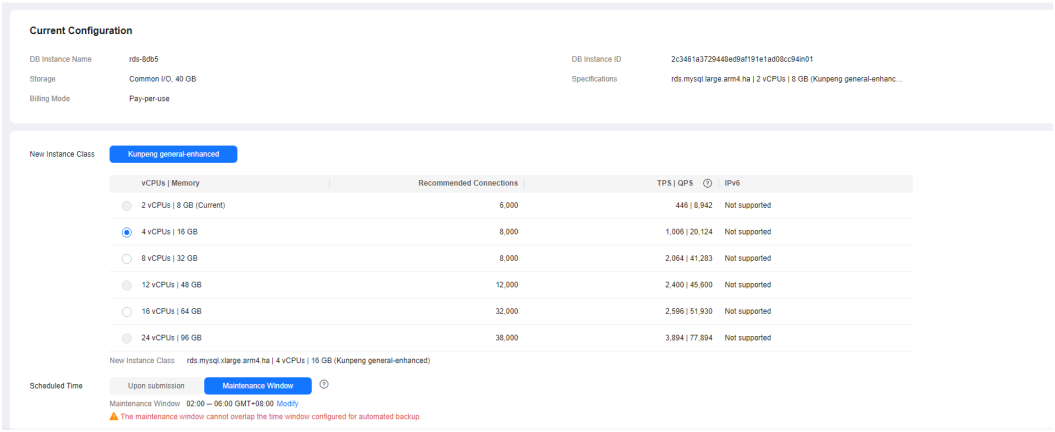
Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Change Instance Class** in the **Operation** column.

Alternatively, click the instance name to go to the **Overview** page. Under **Instance Class**, click **Configure**.

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

For RDS for MySQL DB instances billed on the pay-per-use basis, choose a new instance class and a scheduled time, and click **Next**.

Figure 1-71 Changing a DB instance class



DB instances in a DCC only support the general-enhanced instance class.

- **Upon submission**: An instance class change is applied immediately after the request is submitted.
- **Maintenance Window**: If you select **Maintenance Window** for **Scheduled Time**, after the request is submitted, you can locate the scheduled task and click **Execute Now** in the **Operation** column as required. (If this option is not displayed on the console, **submit a service ticket** to obtain required permissions.) For details about how to view a task, see **Viewing a Task**. Services are temporarily interrupted during an instance class change. You are advised to set the maintenance window to off-peak hours.

Step 6 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.

  To view the cost incurred by the DB instance class change, choose **Billing & Costs** > **Bills** in the upper right corner.
- For yearly/monthly DB instances:
  - If you intend to scale down the DB instance class, click **Submit**.

    The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 7 Check the change result.

Return to the **Instances** page and view the instance status. During the change period, the instance status is **Changing instance class**. You can view the execution progress of **Changing a MySQL DB instance class** on the **Task Center** page. After a few minutes, view the DB instance class on the **Overview** page to check that the change is successful.

For DB instances using cloud disks, if you have selected **Maintenance Window** for **Scheduled Time**, the status of the DB instance on the **Instances** page is **Changing instance class** in the maintenance window.

----End

Follow-up Operations

Return to the instance list. In the navigation pane on the left, choose **Task Center** and check the progress of the change task.

- If you have selected **Upon submission** for **Scheduled Time**:

  On the **Instant Tasks** page, search for "Changing a MySQL DB instance class" and check the execution progress. Instant tasks cannot be canceled.

- If you have selected **Maintenance Window** for **Scheduled Time**:

  On the **Scheduled Tasks** page, search for the instance ID and check the execution status of the change task.

  If the task is in the **To be executed** state, you can click **Cancel** to cancel the task.

For details, see **Viewing a Task**.

1.8.8 Changing a Storage Type

Scenarios

If the storage type of your RDS DB instance does not match your workloads, you can change it as needed.

For details about storage types, see **DB Instance Storage Types**.

Constraints

Table 1-35 Constraints

Phase	Constraints
Before the change	• To change a storage type, **submit a service ticket** to request required permissions. • The storage type of an instance can be changed only when the instance is in the **Available** state. • Changing the storage type may affect storage performance, so the storage type should be changed during off-peak hours. • If the storage type of a read replica is different from that of its associated DB instance, the data synchronization may be affected. To change the storage type of a DB instance, change that of its read replica (if any) first to ensure that the storage type of the read replica is the same as that of the DB instance.

Phase	Constraints
During the change	Changing a storage type takes several minutes or even hours, depending on the throughput, storage space, original storage type, and new storage type.
After the change	In rare cases, a storage type may fail to be changed due to a backend issue. If this happens, try again later.

Table 1-36 Storage types

Instance Type	Original Storage Type	Target Storage Type	Remarks
• Primary/ Standby • HA read replica	Cloud SSD	Extreme SSD	Extreme SSD can also be changed to cloud SSD.

Billing

Table 1-37 Billing

Billing Mode	Operation	Impact on Price
Yearly/ Monthly	Storage type change	You will be billed for the new storage type based on the time remaining in the requested period of your instance. • Price difference payment: If the new storage type is more expensive than the old one, you need to pay for the difference based on how much time is left in the subscription. • Price difference refund: If the new storage type is cheaper than the old one, Huawei Cloud will refund the difference to you based on how much time is left in the subscription. The following prices are for reference only. The actual prices are displayed on the console. Suppose you purchased a one-month RDS for MySQL 8.0 primary/standby instance (instance class: general-purpose, 2 vCPUs | 4 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on August 1, 2024. The instance price was $88.69 USD per month. On August 15, 2024, you changed the storage type to extreme SSD. The instance price became $103.67 USD per month. **Price difference = Price for the new storage type x Remaining period – Price for the original storage type x Remaining period** Remaining period = Remaining days of each calendar month/Maximum number of days in each calendar month In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in August)/30 (Maximum number of days in August) = 0.5. Price difference = $103.67 USD x 0.5 - $88.69 USD x 0.5 = $7.49 USD
Pay-per-use	Storage type change	The new storage type is billed by the hour. For details, move your pointer over the price on the **Price Calculator** page to view the storage price.

Procedure

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

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

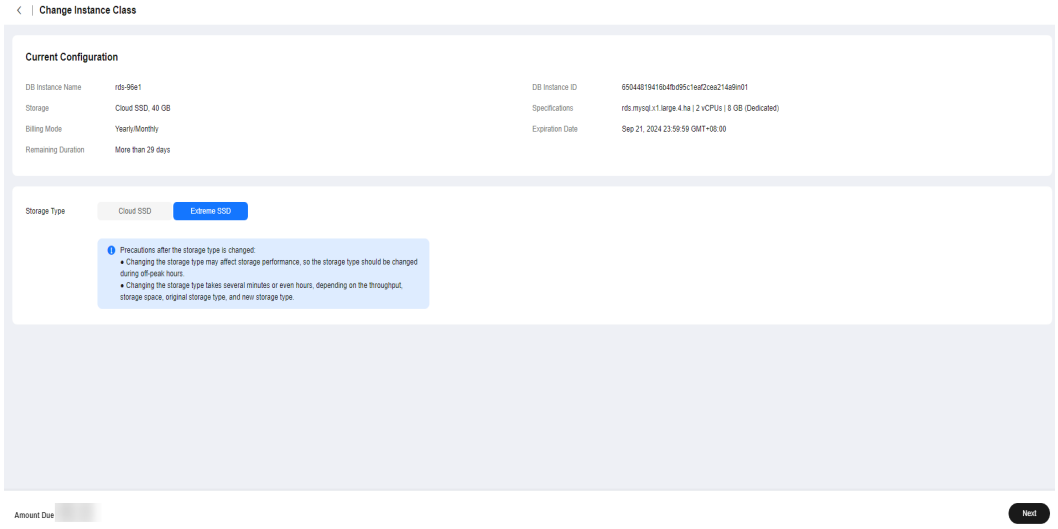
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **Instance Class**, click **Configure**.

Step 5 On the displayed page, select a new storage type and click **Next**.

Figure 1-72 Changing a storage type



Step 6 Confirm the new storage type.

- To modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**. The new storage type is applied only after the payment is successful. If the storage type is downgraded, Huawei Cloud will refund the price difference to you.

Step 7 Check the result.

Return to the **Instances** page. After a few minutes, click the instance name and check the new storage type on the displayed **Overview** page.

----End

1.8.9 Configuring Auto Scaling of vCPUs and Memory

Scenarios

RDS for MySQL allows you to set limits on CPU usage. When the average CPU usage of your DB instance reaches the configured limits, the vCPU and memory specifications will be scaled up or down.

Constraints

- Only primary/standby instances using cloud disks support auto scaling.
- Configuring auto scaling requires the iam:agencies:listAgencies permission. If you do not have this permission, **create a custom policy**.
- To configure auto scaling, you need to submit a service ticket by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console to request required permissions.
- Auto scaling of instance specifications will briefly interrupt services.
- Auto scaling cannot be triggered if the instance is abnormal, stopped, or frozen.
- Auto scaling cannot be enabled if the instance has other ongoing tasks or scheduled specification changes. If it has been enabled, it still cannot be triggered.
- Auto scaling cannot be triggered for any yearly/monthly instance that has unfinished orders or is in arrears.
- If your instance has read replicas, the read replica specifications should not be too small, or they may have excessive latency and a heavy load.
- There are temporary IP addresses used when auto scaling is performed, and they are not released immediately after the scaling is complete. There is a 12-hour delay by default. If there are not enough private IP addresses available, the scaling will fail.
- During a scale-up, if the next larger specification is sold out or unavailable, the system continues to check the following larger specifications until one is matched. During a scale-down, to avoid out of memory (OOM) problems, if the next smaller specification is sold out or unavailable, the system will stop matching downwards.

Billing

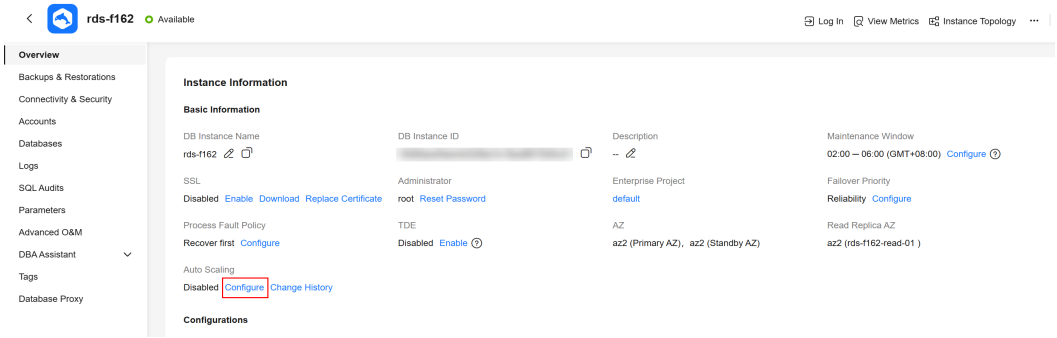
The billing for auto scaling is the same as that for changing an instance class. For details, see **Billing**.

Configuring Auto Scaling

Step 1 On the **Instances** page, click the name of the target primary/standby instance to go to the **Overview** page.

Step 2 Click **Configure** under **Auto Scaling**.

Figure 1-73 Configuring Auto Scaling



Step 3 In the displayed dialog box, configure the required parameters.

Figure 1-74 Setting auto scaling parameters

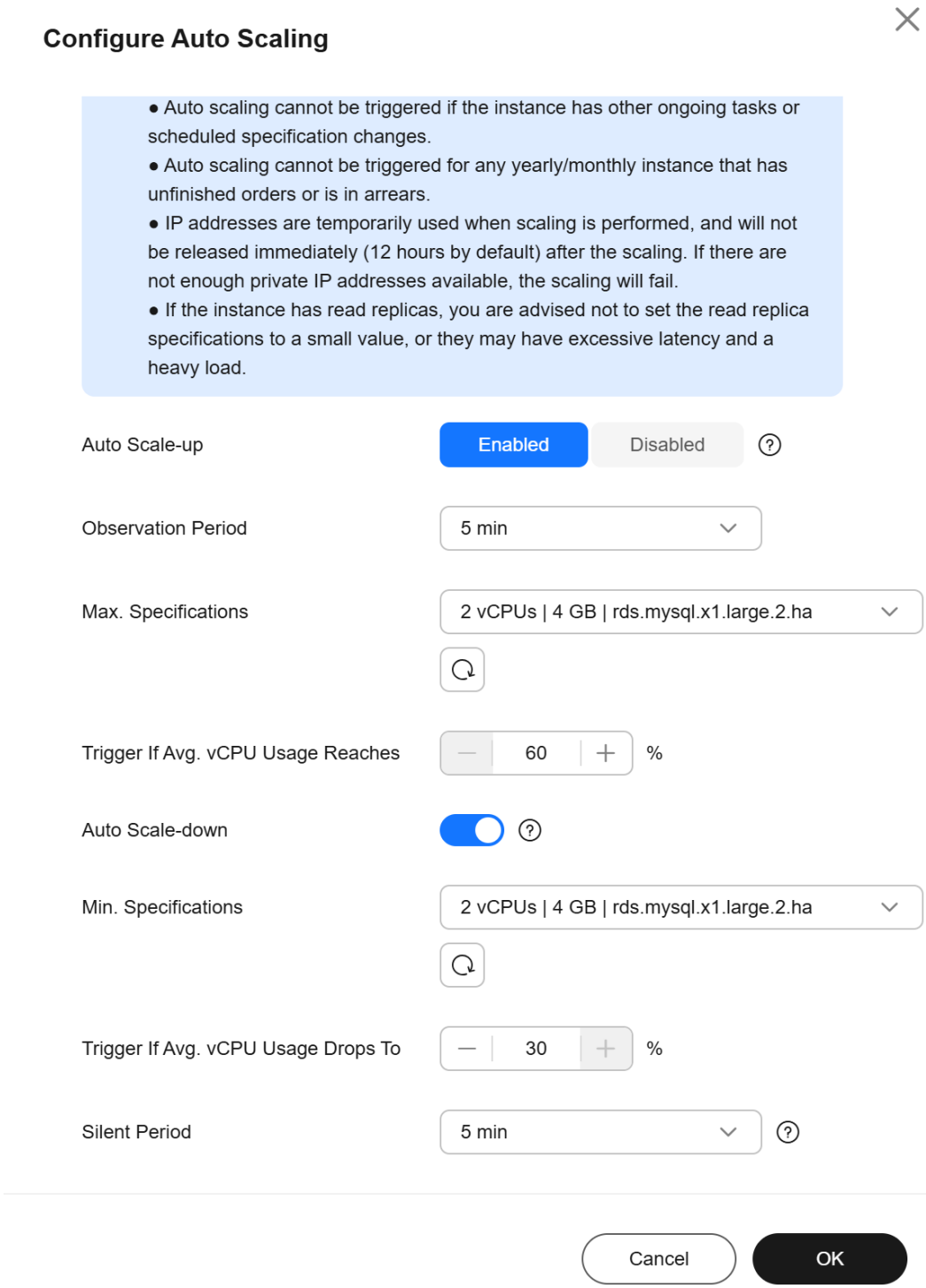


Table 1-38 Parameter description

Parameter	Description
Auto Scale-up	You can enable or disable it as needed.
Observation Period	• When auto scale-up is enabled, the system periodically checks the vCPU usage. If the average vCPU usage exceeds the preset limit within the observation period, the system adapts the specifications to the read and write traffic. The system then enters a silent period. • The minimum observation period is 5 minutes.
Max. Specifications	The maximum specifications after the final auto scale-up. The system scales up specifications level by level. Each time the specifications are upgraded to a higher level, the system then enters a silent period.
Trigger If Avg. vCPU Usage Reaches	The vCPU usage limit for triggering an auto scale-up. If the average vCPU usage exceeds this limit during the observation period, an auto scale-up is triggered.
Auto Scale-down	You can enable or disable it as needed.
Min. Specifications	The minimum specifications after the final auto scale-down. The system scales down specifications level by level. Each time the specifications are downgraded to a lower level, the system then enters a silent period.
Trigger If Avg. vCPU Usage Drops To	The vCPU usage limit for triggering an auto scale-down. When the average vCPU usage in the observation period is less than this limit, an auto scale-down is triggered.
Silent Period	The silent period is the minimum interval between two specification changes (triggered automatically or manually). During the silent period, RDS for MySQL will not trigger auto scaling.

Step 4 Click **OK**.

----End

Checking the Change History

Step 1 On the **Instances** page, click the name of the target primary/standby instance to go to the **Overview** page.

Step 2 Click **Change History** under **Auto Scaling**.

Step 3 In the displayed dialog box, check the change time, change type, status, original specifications, and new specifications of the historical tasks.

----End

1.8.10 Scaling Up Storage Space

Scenarios

If the storage space is not enough for your workloads, you can scale up storage space of your DB instance. **The backup space will also increase with the storage.**

You can configure an alarm rule for the storage space usage, so you are alerted if the threshold is reached. For details, see **Setting Alarm Rules**.

For details about the causes and solutions of insufficient storage space, see **What Should I Do If My Data Exceeds the Database Storage Space of an RDS DB Instance?**

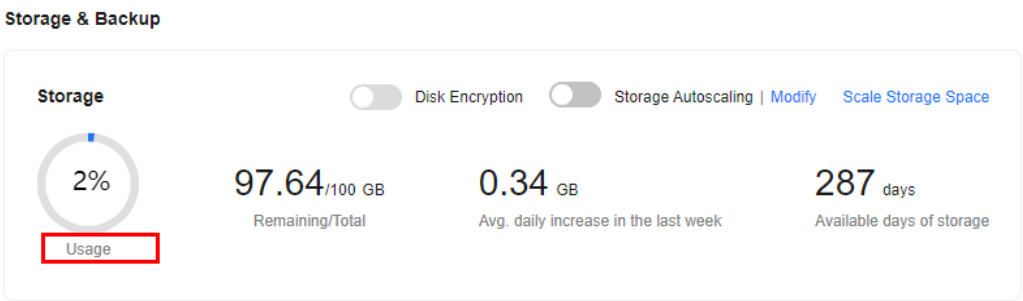
During scale-up, services are not interrupted. If you need to change the storage type, refer to **Instance Class Type Change**.

Storage Full Situations Causing Instances to Be Read-Only

Table 1-39 Conditions for causing instances to be read-only

Storage Size	Condition
Storage size is less than 1 TB.	• If the storage usage is displayed as 100%, the instance becomes read-only. • If the storage usage decreases to 87%, the instance exits the read-only state.
Storage size is greater than or equal to 1 TB.	• If the available storage space is less than 30 GB, the instance becomes read-only. • If the available storage space is greater than or equal to 150 GB, the instance exits the read-only state.

Figure 1-75 Checking storage usage



Constraints

- You can scale up storage space only when your account balance is greater than or equal to $0 USD.

- You can scale up storage space only when your instance status is **Available** or **Storage full**.
- The maximum allowed storage is 4,000 GB. If you want to increase the storage upper limit to 10 TB, **submit a service ticket**.
- When storage space is being scaled up, the DB instance is in **Scaling up** state and the backup tasks of the instance 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.
- Reboot is not required during scale-up.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up, not down.
- If you scale up a DB instance with disk encryption enabled, the expanded storage space will also use the original key for encryption.

Billing

Table 1-40 Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Storage scale-up	You need to pay for the difference in price based on the remaining period. The following prices are for reference only. Final prices on the console may differ. Suppose you purchased a one-month RDS for MySQL 5.7 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The unit price of storage space is $0.214 USD/GB per month. On June 15, 2023, you scaled up the storage by 60 GB. The total storage after scale-up is 100 GB. **Price difference = Scale-up volume x Unit price x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Price difference = 60 GB x $0.214 USD x 0.5 = $6.42 USD
Pay-per-use	Storage scale-up	The new storage space is billed by hour. For details, see **Product Pricing Details**.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 instance name to enter the **Overview** page. In the **Storage & Backup** area, click **Scale Storage Space**.
- If the storage space is full, locate the target DB instance on the **Instances** page and click **Scale** in the **Status** column.

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 the settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time period, the status of the DB instance on the **Instances** page will be **Scaling up**. After a while, click the DB instance and view the new storage space on the displayed **Overview** page to verify that the scale-up is successful.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click in front of it. Locate the 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 read replica name to enter the **Overview** page. In the **Storage & Backup** area, click **Scale Storage Space**.
- If the storage space is full, locate the read replica on the **Instances** page and click **Scale** in the **Status** column.

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 and the read replica uses pay-per-use billing, click **Submit**.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time period, the status of the read replica on the **Instances** page will be **Scaling up**. After a while, click the read replica name and view the new storage space on the displayed **Overview** page to verify that the scale-up is successful.

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

----End

1.8.11 Configuring Storage Autoscaling

Scenarios

With storage autoscaling enabled, when RDS detects that you are running out of database space, it automatically scales up your storage.

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

You can enable storage autoscaling in either of the following ways:

- Enable this function when you create a DB instance. For details, see **Buying an RDS for MySQL DB Instance**.
- Enable this function after you create a DB instance. See the operations provided in this section.

Constraints

- You can enable storage autoscaling only when your account balance is greater than or equal to $0 USD.

- The storage space can be scaled up automatically only when your instance status is **Available** or **Storage full**.
- To apply for the storage autoscaling permission, submit a service ticket by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- Storage autoscaling for RDS for MySQL DB instances is supported only for cloud SSD and extreme SSD storage types. For details about storage types, see **DB Instance Storage Types**.
- The maximum allowed storage is 4,000 GB.
- For a primary/standby DB instance, autoscaling the storage for the primary node will also autoscale the storage for the standby node.
- Storage autoscaling is unavailable when the DB instance is in any of the following statuses: changing instance class, upgrading a minor version, migrating the standby DB instance, and rebooting.
- If a yearly/monthly DB instance has pending orders, it will not be autoscaled.

Procedure

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

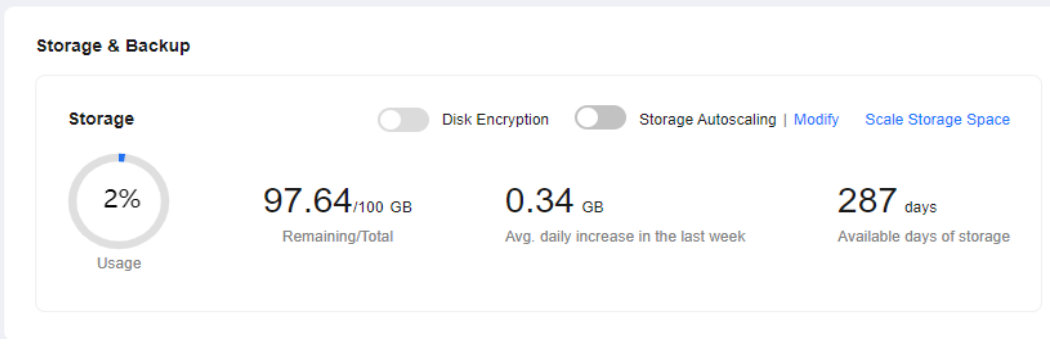
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance or read replica name (click in front of a DB instance to locate its read replica).

Step 5 In the **Storage & Backup** area, toggle on the **Storage Autoscaling** switch.

Figure 1-76 Storage space



Step 6 In the displayed dialog box, set the following parameters:

Figure 1-77 Configuring autoscaling

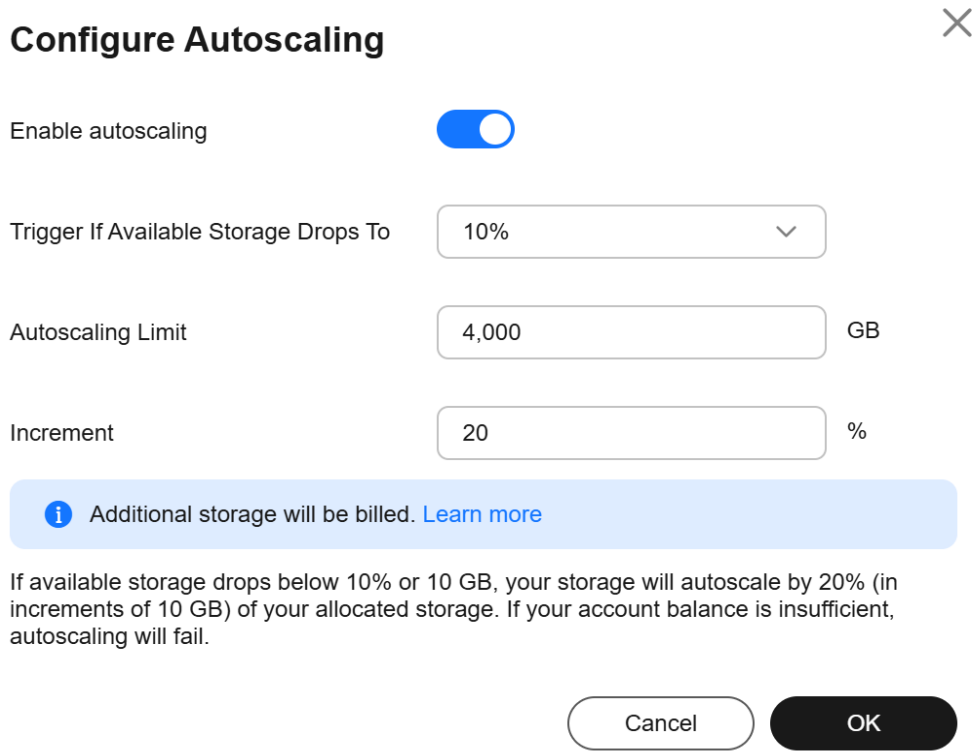


Table 1-41 Parameter description

Parameter	Description
Enable autoscaling	If you select this option, autoscaling is enabled.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered.
Autoscaling Limit	The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance.

Step 7 Click **OK**.

----End

1.8.12 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 once or twice. Ensure that your applications support automatic reconnection.

Procedure

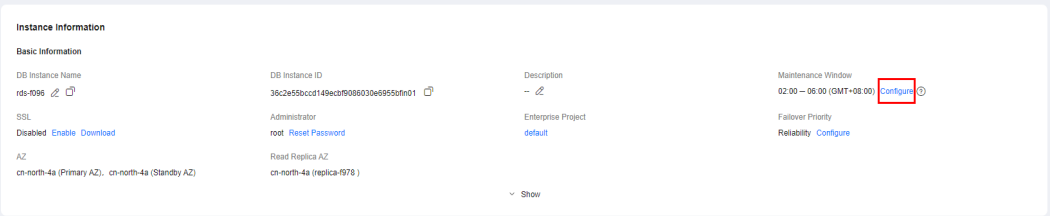
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

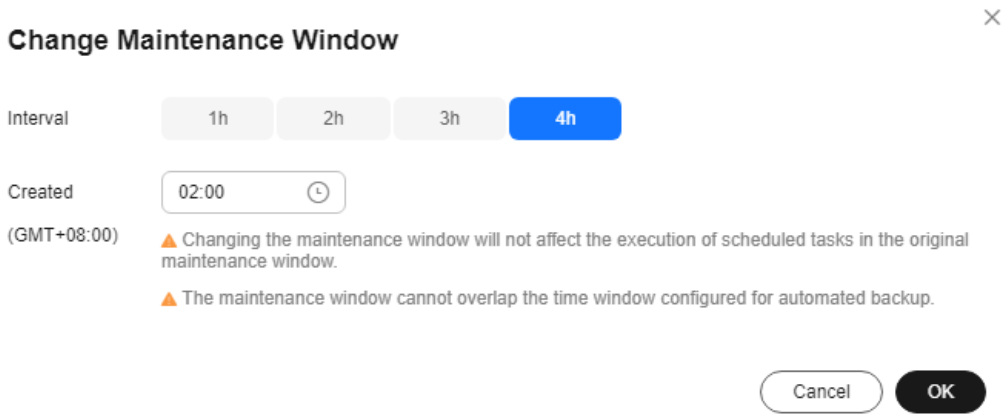
Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page. Under **Maintenance Window**, click **Configure**.

Figure 1-78 Changing the maintenance window



Step 5 In the displayed dialog box, select an interval and a start time, and click **OK**.

Figure 1-79 Changing the maintenance window



NOTE

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

----End

1.8.13 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.
- The time required for changing an instance type from single to primary/standby depends on the amount of data to be backed up. Changing a single instance to a primary/standby instance does not affect workloads on the instance.

Precautions

RDS single DB instances can be changed to primary/standby DB instances, but not the other way around. You can use Data Replication Service (DRS) or the export and import tool of the client to migrate data from primary/standby DB instances to single DB instances.

Changing a single-node instance to primary/standby does not change its networking information, including the VPC, subnet, security group, private IP address, private domain name, and database port.

Billing

Table 1-42 Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Changing a single DB instance to primary/ standby	You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are subject to the price displayed on the console. Suppose you purchased a one-month RDS for MySQL 5.7 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The instance price was $59.56 USD per month. On June 15, 2023, you changed the instance type from single to primary/standby. The instance price became $155.69 USD per month. **Price difference = Price for the primary/ standby instance × Remaining period - Price for the single instance × Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as fellows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Price difference = $155.69 USD x 0.5 - $59.56 USD x 0.5 = $48.06 USD
Pay-per-use	Changing a single DB instance to primary/ standby	The primary/standby DB instance is billed by hour. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

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**.

For yearly/monthly DB instances, click **Pay Now**.

Step 6 Check the instance status on the **Instances** page.

- The DB instance is in the **Changing type to primary/standby** status. You can view the task progress (not the time 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

1.8.14 Promoting a Read Replica to Primary

Scenarios

RDS enables you to promote a read replica to a single-node 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.

If your DB instance fails and you want to quickly obtain a readable and writable instance, you can promote one of the instance's read replicas to primary.

Constraints

- This function is available only to RDS for MySQL 5.7 and 8.0.
- Only pay-per-use read replicas can be changed to single DB instances.
- This function is unavailable for DB instances with proxy enabled.
- Only general-purpose and Kunpeng general-enhanced read replicas using cloud SSDs can be promoted to single DB instances.

Procedure

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

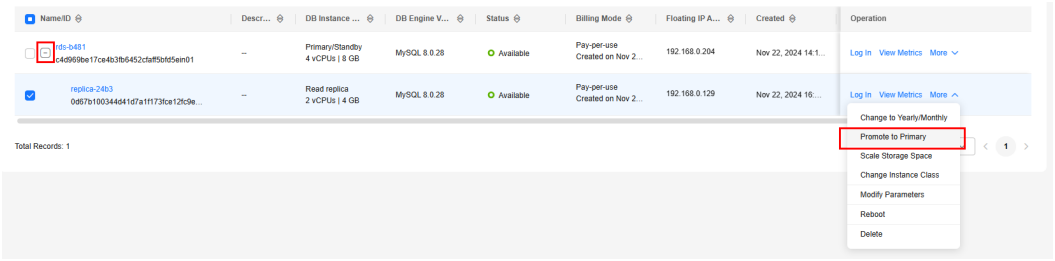
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target RDS for MySQL instance and click  in front of it to expand the read replica list.

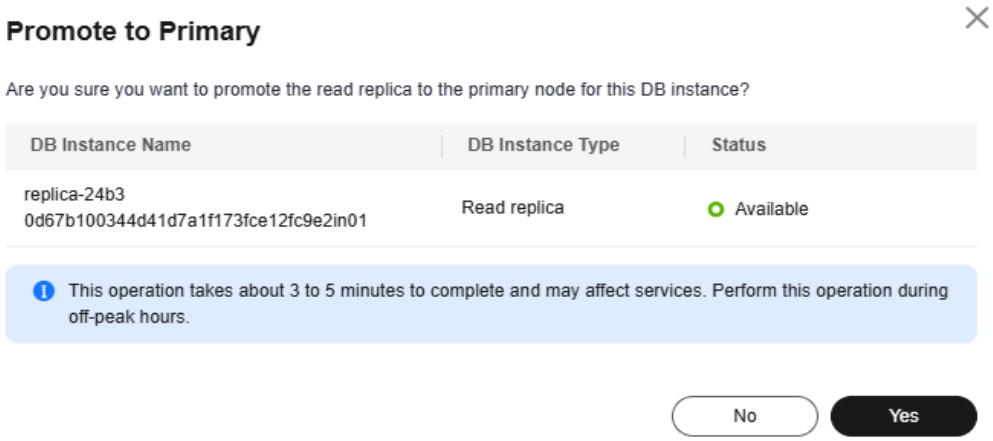
Step 5 Find the desired read replica and choose **More** > **Promote to Primary** in the **Operation** column.

Figure 1-80 Locating a read replica



Step 6 In the displayed dialog box, click **Yes**.

Figure 1-81 Promoting a read replica to primary



Step 7 After the promotion is complete, check and manage the new instance on the **Instances** page.

- During the promotion, the read replica status is **Promoting to primary**. Upon the completion of the promotion, a full backup is automatically performed.
- After the read replica is promoted to primary, it is disassociated from the original DB instance. Its **Status** becomes **Available** and **DB Instance Type** becomes **Single**.
- The billing mode on the new instance remains unchanged.

----End

1.8.15 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 manual switchover does not change the connection information of the DB instance, including its VPC, subnet, security group, floating IP address, private domain name, and database port.

You can switch the primary and standby instances only when the following conditions are met:

- The primary/standby instance is running properly.
- The primary/standby replication is normal.
- The replication delay is less than 5 minutes, and the data on the primary and standby instances is consistent.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target primary/standby instance name to go to the **Overview** page.

Step 5 Under **DB Instance Type**, click **Switch**.

NOTICE

A primary/standby switchover may cause a brief interruption of several seconds or minutes (depending on the replication delay). If transaction logs are generated at a speed higher than 30 MB/s, services will probably be interrupted for several minutes. To prevent traffic congestion, perform a switchover during off-peak hours.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the ***Identity and Access Management User Guide***.

Step 7 In the displayed dialog box, click **OK**.

Step 8 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, 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

1.8.16 Changing the AZ of a Standby DB Instance

Scenarios

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

Constraints

- Primary/standby instances running MySQL 5.6, 5.7, or 8.0 support standby instance migration to another AZ.
- Batch write operations during peak hours may cause migration failures. To ensure successful migration, perform the migration during off-peak hours.
- DDL operations and scheduled events will be suspended during migration. To prevent service interruptions, perform the migration during off-peak hours.
- For details about regions and AZs, see **Regions and AZs**.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 Check the DB instance status on the **Instances** 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 **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**.
- On the **Overview** page, find **AZ** and check the AZ hosting the standby DB instance.

----End

1.8.17 Updating the OS of a DB Instance

To improve database performance and security, the OS of an RDS for MySQL instance needs to be updated timely.

Every time you upgrade the kernel version of your instance, RDS for MySQL determines whether to update the OS and selects the right cold patch to upgrade the OS if necessary.

Updating the OS does not change the DB instance version or other information.

In addition, RDS for MySQL installs hot patches as required to fix major OS vulnerabilities within the maintenance window you specified.

1.9 Data Backups

1.9.1 Introduction to Backups

What Are Database Backups?

RDS for MySQL automatically creates backups for DB instances during the backup time window. The backups are stored based on a preset retention period (1 to 732 days).

A backup file is generated each time a backup is complete. If the instance fails or data is damaged, you can use the backup file to restore the instance, ensuring data reliability.

Function Description

Backup Types

RDS for MySQL supports multiple backup types. For details, see **Backup Types**.

Where Data Is Backed Up

- Single-node instance

  A single-node architecture, which is more cost-effective than mainstream primary/standby DB instances. After a backup is triggered, data is backed up from the primary instance and stored as a package on OBS. The backup does not take up storage space of the instance.

- Primary/standby instance

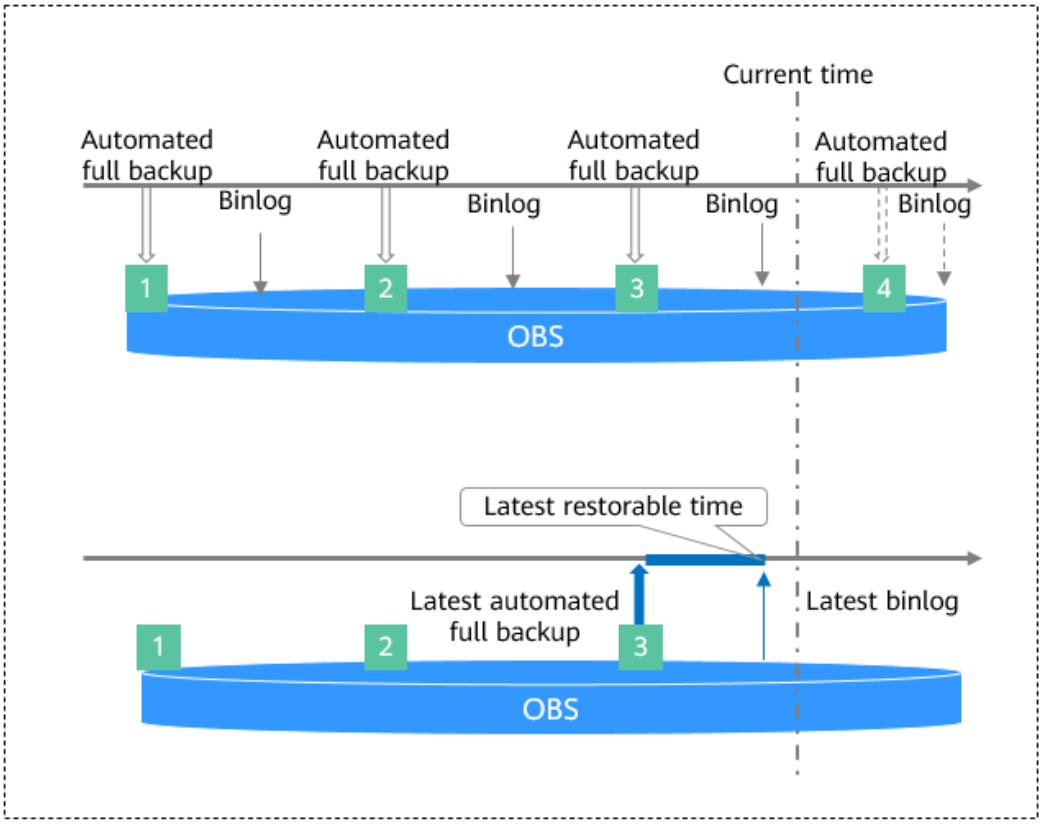
  An HA architecture. In a primary/standby pair, each instance has the same instance class. After a backup task is triggered, data is backed up from the standby instance (or from the primary instance only when there is a long primary/standby replication delay) and stored as a package on OBS. The backup does not take up storage space of the instance.

  If a database or table in the primary instance is maliciously or mistakenly deleted, the database or table in the standby instance will also be deleted. In this case, you can only use backups to restore the deleted data.

How Data Is Backed Up

RDS for MySQL automated backup is enabled by default and cannot be disabled. RDS for MySQL performs automated full backups based on the time window and interval specified in the backup policy. It also backs up data modifications made after the most recent automated full or binlog backup every 5 minutes or when a certain amount of incremental data is generated. When you restore an instance to a point in time, the most recent full backup will be downloaded from OBS for restoration. After the restoration is complete, binlog backups will be replayed to the specified point in time.

Figure 1-82 How data is backed up



Backup Compression Ratio

RDS uses **sysbench** to import data models and a certain amount of data. After data is backed up, the compression ratio is about 80%. The more duplicate data there is, the higher the compression ratio is.

Compression ratio = Space occupied by backup files/Space occupied by data files x 100%

Backup Storage and Billing

Backups are saved as packages in OBS buckets. Backups occupy backup space in OBS. If the free space RDS provides is used up, the additional space required will be billed. For details, see **How Is RDS Backup Data Billed?**

After CBR snapshot-based backup is enabled, the free backup space is unavailable. You are billed for database server backup vaults on a pay-per-use basis. For details, see **How Is CBR Billed?**

Deleting Backups

- Manual backups and automated backups can be deleted in different ways:
  - Manual backups can only be manually deleted.
  - Automated backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your **automated backup policy**. When the retention period expires, automated backups will be deleted.
- Local binlogs

  If the retention period is set to **0**, the binlogs of your DB instance will be deleted once they are synchronized to the standby instance and read replicas and successfully backed up to OBS.

  If the retention period is set to a value greater than 0, for example, 1 day, the binlogs will be retained for one day after they are synchronized to the standby instance and read replicas from the primary instance and successfully backed up to OBS. After the retention period expires, the binlogs will be automatically deleted.

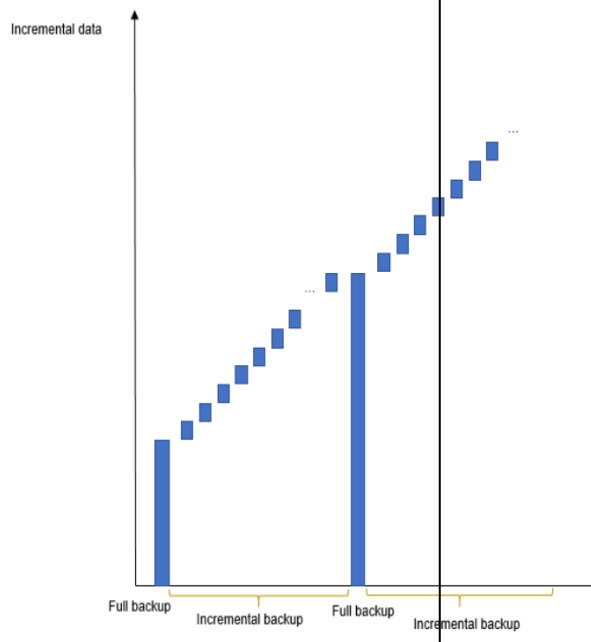
1.9.2 Backup Types

RDS for MySQL supports multiple backup types. Based on different dimensions, there are the following backup types:

Full Backups and Incremental Backups Based on Data Volume

Table 1-43 Comparison between full backups and incremental backups

Backup Type	Full backups	Incremental backups
Description	All data in an instance is backed up.	Only data changes within a certain period of time are backed up.
Enabled by Default	Yes	Yes
Retention Period	• You can specify how many days automated backups can be retained for. If you shorten the retention period, the new backup policy takes effect for existing backups. • Manual backups will not be deleted until you delete them manually.	Incremental backups will be deleted along with automated full backups.

Characteristics	• A full backup is to back up all data of your DB instance in the current point of time. • You can use a full backup to restore the complete data generated when the backup was created. • Full backups can be created automatically or manually.	• The system automatically backs up data modifications made after the most recent automated or incremental backup every 5 minutes or when a certain amount of incremental data is generated. • Incremental backups can only be created automatically. • Restoring data from incremental backups relies on the most recent full backup, as shown in **Figure 1-83**, so the most recent full backup that exceeds the retention period will not be deleted to ensure successful restoration. **Figure 1-83** Incremental data restoration 
How to Check Backup Size	Click the target instance name. On the **Backups & Restorations** page, click the **Full Backups** tab and check the backup size.	Click the target instance name. On the **Backups & Restorations** page, click the **Binlog Backups** tab and check the backup size.

Automated Backups and Manual Backups Based on Backup Methods

Table 1-44 Comparison between automated backups and manual backups

Backup Type	Automated backups	Manual backups
Description	• You can set an automated backup policy on the console, and the system will back up your instance data based on the time window and backup cycle you set in the backup policy and will store the backups for the retention period you specified. • Automated backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your automated backup policy. Retained backups (including full and incremental backups) will be automatically deleted at the end of the retention period.	• Manual backups are user-initiated full backups of your DB instance. They are retained until you delete them manually. • Regularly backing up your DB instance is recommended, so if your DB instance fails or data is corrupted, you can restore it using backups.
Enabled by Default	Yes	Yes
Retention Period	Automated backups are retained for the number of days you specified. The retention period ranges from 1 to 732 days.	Manual backups are always retained until you delete them manually.
How to Configure	See **Configuring an Intra-Region Backup Policy**.	See **Creating a Manual Backup**.

Intra-Region Backups and Cross-Region Backups Based on Backup Regions

NOTE

To use cross-region backup, **submit a service ticket** to request required permissions.

Table 1-45 Comparison between intra-region backups and cross-region backups

Backup Type	Intra-region backups	Cross-region backups
Description	Backups are stored in the same region as your DB instance.	Backups are stored in a different region from that of your DB instance.
Enabled by Default	Yes	No
Retention Period	Backups are retained for the number of days you specified. The retention period ranges from 1 to 732 days.	Backups are retained for the number of days you specified. The retention period ranges from 1 to 1,825 days.
Characteristics	Backups are stored in the same region as your DB instance. Intra-region backup (automated backup) is enabled by default and cannot be disabled.	Backups are stored in a region different from the one where your DB instance is located. After you enable cross-region backup, the backups are automatically stored in the region you specify.
How to Configure	**Configuring an Intra-Region Backup Policy**	**Configuring a Cross-Region Backup Policy**
How to Check Backup Size	Click **Backups** in the navigation pane. On the **Same-Region Backups** tab, check the backup size.	Click **Backups** in the navigation pane. On the **Cross-Region Backups** tab, click **View Cross-Region Backup** and check the backup size.

1.9.3 Performing Backups

1.9.3.1 Configuring an Intra-Region 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.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

You do not need to set an interval for incremental backup because RDS automatically backs up incremental data every 5 minutes or when a certain amount of incremental data is generated. Incremental backups can be used to restore data to a specific point in time.

Differences Between Regular Backup and Sparse Backup

Table 1-46 Backup functions

Item	Regular Backup	Sparse Backup
CBR backup	Supported	Supported
Function description	By default, regular backup is used.	To enable sparse backup, **submit a service ticket** to request required permissions. After sparse backup is enabled, a regular backup policy is displayed by default. You can add sparse backup policies as needed.
Number of backup policies	One policy	Up to 10 backup policies can be configured for an instance.
Backup cycle	Only weekly backup is supported. At least one day in a week must be selected.	Backups can be created by week, month, or year. You can configure these three types of backup policies for your instance flexibly. To reduce storage costs, you can choose to back up data once a year at least.
Backup time	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00
Retention period	1 to 732 days	1 to 732 days
Deletion of backup policies	Not supported	The default backup policy cannot be deleted, but the manually added backup policies can be deleted.

Constraints

Constraints (CBR Disabled)

- For primary/standby instances, if the standby instance is faulty or the replication delay of the standby instance exceeds one day, backups will be performed on the primary instance.
- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- Rebooting the instance is not allowed during full backup. Exercise caution when selecting a backup time window.
- 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.
- Performing a full backup may decrease instance throughput and increase replication delay because it occupies node resources, especially disk bandwidth.

Constraints (CBR Enabled)

- The backup time is proportional to how much data your instance has. Too much data can decrease the backup efficiency. If you have large amounts of data and want to speed up the backup process, **submit a service ticket** to enable **Cloud Backup and Recovery (CBR)**.
- After CBR is enabled, snapshot backup is used. Existing automated and manual backups can still be used to restore data.
- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- DDL operations cannot be performed when a CBR snapshot is being created. If a DDL operation is being performed, a snapshot will be created after the DDL operation is complete.
- After CBR is enabled, the next full backup is a snapshot backup. You can use the snapshot backup to restore data.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

After CBR is enabled, the free backup space is unavailable. You are billed for database server backup vaults on a pay-per-use basis. For details, see **How Is CBR Billed?**

Procedure

Configuring a Regular Backup Policy (Sparse Backup Disabled)

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

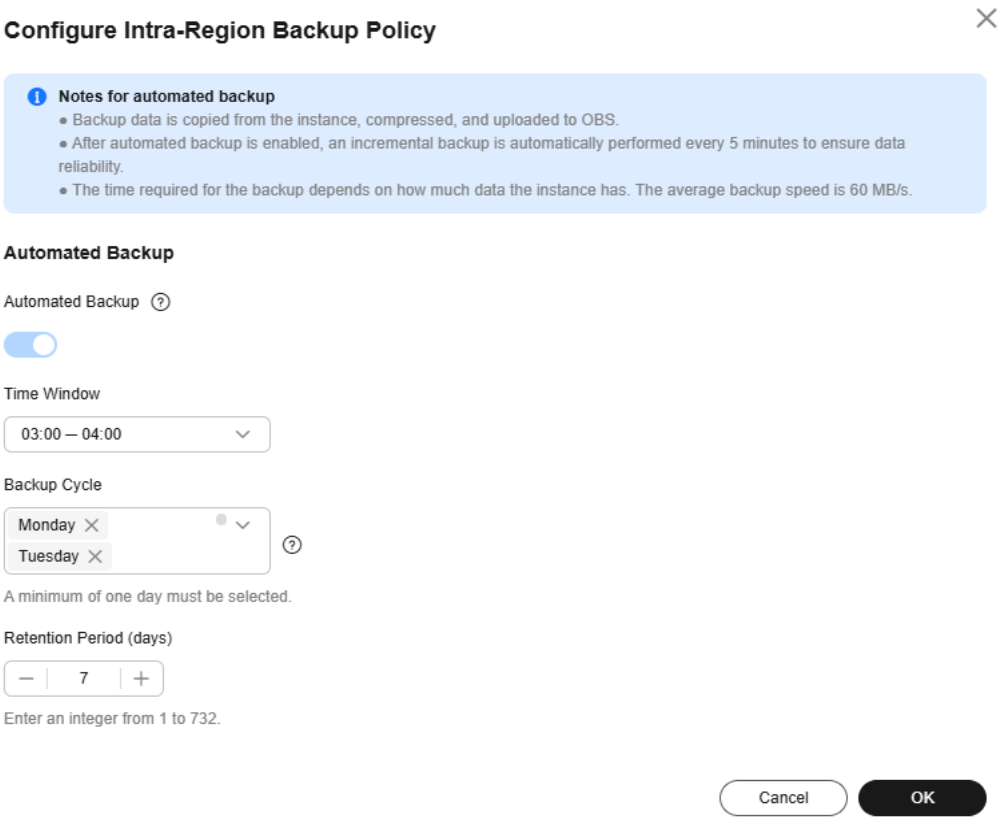
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Backups & Restorations** page, click **Configure Intra-Region Backup Policy**. On the displayed page, you can view the existing backup policy. If you want to modify the policy, adjust the values of the following parameters:

Figure 1-84 Modifying a backup policy



- **Retention Period**: How many days your automated full backups and binlog backups can be retained. The retention period is from 1 to 732 days and the default value is **7**.
  - Extending the retention period improves data reliability.
  - Reducing the retention period takes effect for existing backups. Any backups (except manual backups) that have expired will be automatically deleted. Exercise caution when performing this operation.

  Policy for automatically deleting automated full backups:

  To ensure data integrity, even after the retention period expires, the most recent backup will be retained, for example, if **Backup Cycle** was set to **Monday** and **Tuesday** and **Retention Period** was set to **2**:

  - The full backup generated on Monday will be automatically deleted on Thursday because:

    The backup generated on Monday expires on Wednesday, but it is the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday.
  - The full backup generated on Tuesday will be automatically deleted on the following Wednesday because:

    The backup generated on Tuesday will expire on Thursday, but as it is the last backup, it will be retained until a new backup expires. The next backup will be generated on the following Monday and will expire on the following Wednesday, so the full backup generated on Tuesday will not be automatically deleted until the following Wednesday.

- **Time Window**: Set it to a one-hour period the backup will be scheduled for each day, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.

  NOTE

  To minimize the potential impact on services, set the time window to off-peak hours. The backup time is in UTC format. The backup time window changes with the time zone during the switch between the DST and standard time.

- **Backup Cycle**: Daily backups are selected by default, but you can change it. At least one day must be selected.

Step 6 Click **OK**.

----End

Configuring a Sparse Backup Policy

To use this function, **submit a service ticket** to request required permissions.

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

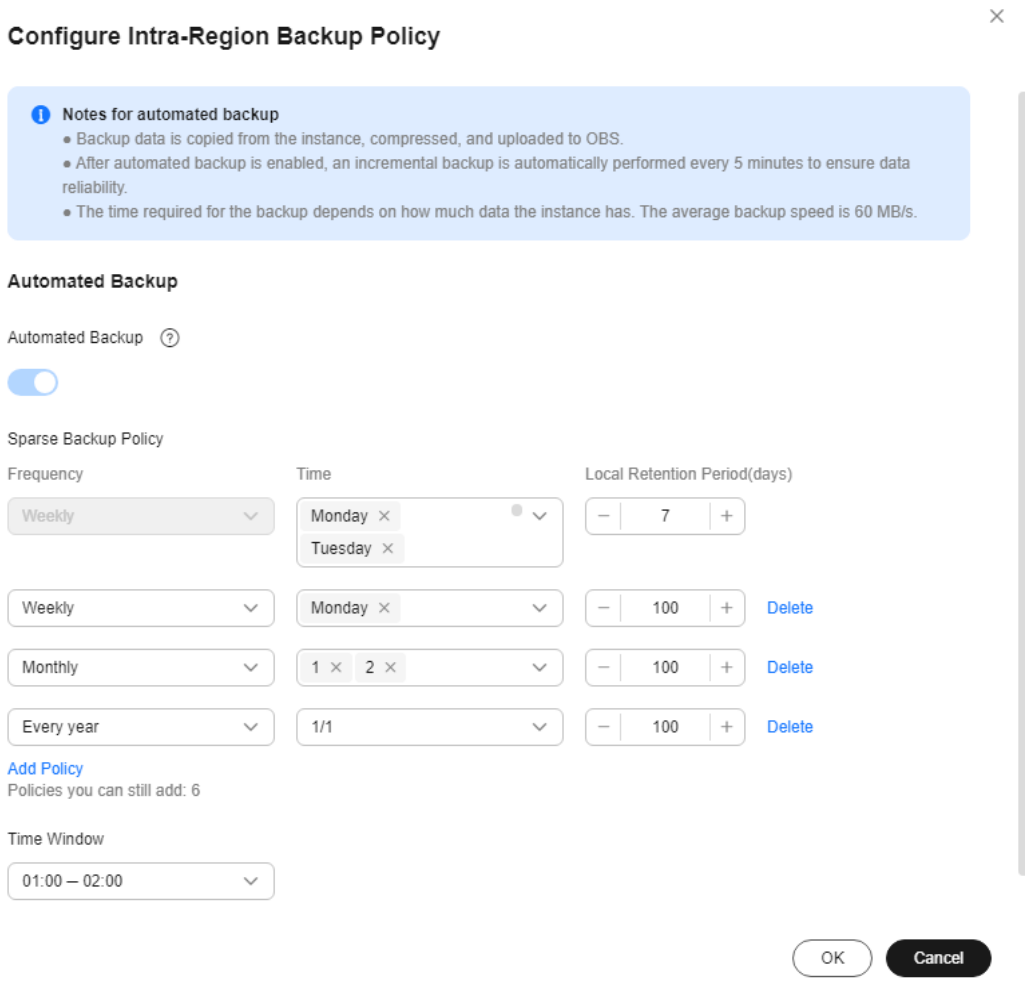
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Configure Intra-Region Backup Policy**. You can see the configured backup policy. To modify the backup policy, adjust the parameter values as needed.

Figure 1-85 Configuring a sparse backup policy



- **Time Window**: Set it to a one-hour period the backup will be scheduled for each day, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.

  NOTE

  To minimize the potential impact on services, set the time window to off-peak hours. The backup time is in UTC format. The backup time window changes with the time zone during the switch between the DST and standard time.

- **Sparse Backup Policy**

  The default backup policy is weekly backup. Select at least one day in a week. The backups can be retained for 1 to 732 days.

  You can add more backup policies to back up data by week, month, or year. A maximum of 10 sparse backup policies can be added for a DB instance.

Table 1-47 Configuring sparse backup policies

Frequency	Time	Retention Period
Every week	Select at least one day in a week.	The retention period is the number of days for storing full automated backups and binlog backups. It can be set to 1 to 732 days. – Extending the retention period improves data reliability. – Reducing the retention period takes effect for existing backups. Any backups (except manual backups) that have expired will be automatically deleted. Exercise caution when performing this operation.
Every month	Select at least one day in a month.	
Every year	Select a specific day in a year.	

- If multiple sparse backup policies need to create backups on the same day, the system generates only one backup on that day and retains the backup based on the longest retention period.
- After sparse backup policies are configured, if no backup is generated in the specified backup window of a day (the backup conflicts with other operations or the backup fails), the backup will be skipped.
- If sparse backup policies are deleted and the backups that have been generated cannot match any existing policy, they will still be kept for the original retention period.

Step 6 Click **OK**.

----End

1.9.3.2 Configuring a Cross-Region Backup Policy

Scenarios

RDS can store backups in a different region from the DB instance for disaster recovery. If a DB instance in one region fails, you can use backups from another region to restore the data to a new DB instance.

If you enable cross-region backup, backups are automatically stored in the region you specify. On the **Backups** page of the RDS console, you can click **View Backup** in the **Operation** column and manage cross-region backups. If cross-region backup is not enabled, backups are stored in the region where your instance is located by default.

You can set cross-region backup policies for up to 150 instances under an account by default.

Precautions

To use cross-region backup, **submit a service ticket** to request required permissions. This function cannot ensure data timeliness. Therefore, the SLA is not

guaranteed. If timeliness is a concern, instead of cross-region backup, you are advised to use Data Replication Service (DRS).

Billing

Table 1-48 Billing method

Specification Code	Pay-per-Use (USD/GB/Hour)
rds.mysql.crossreg.backup.space	0.0002

Supported Regions

To use cross-region backup, **submit a service ticket** to apply for required permissions.

Cross-region backup is only supported in the regions listed in the following table.

Table 1-49 Supported regions

Source Region	Destination Region
CN North-Beijing4	CN East-Shanghai1 and CN South-Guangzhou
CN East-Shanghai1	CN North-Beijing4 and CN South-Guangzhou
CN Southwest-Guiyang1	CN North-Ulanqab1
CN North-Ulanqab1	CN Southwest-Guiyang1

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.

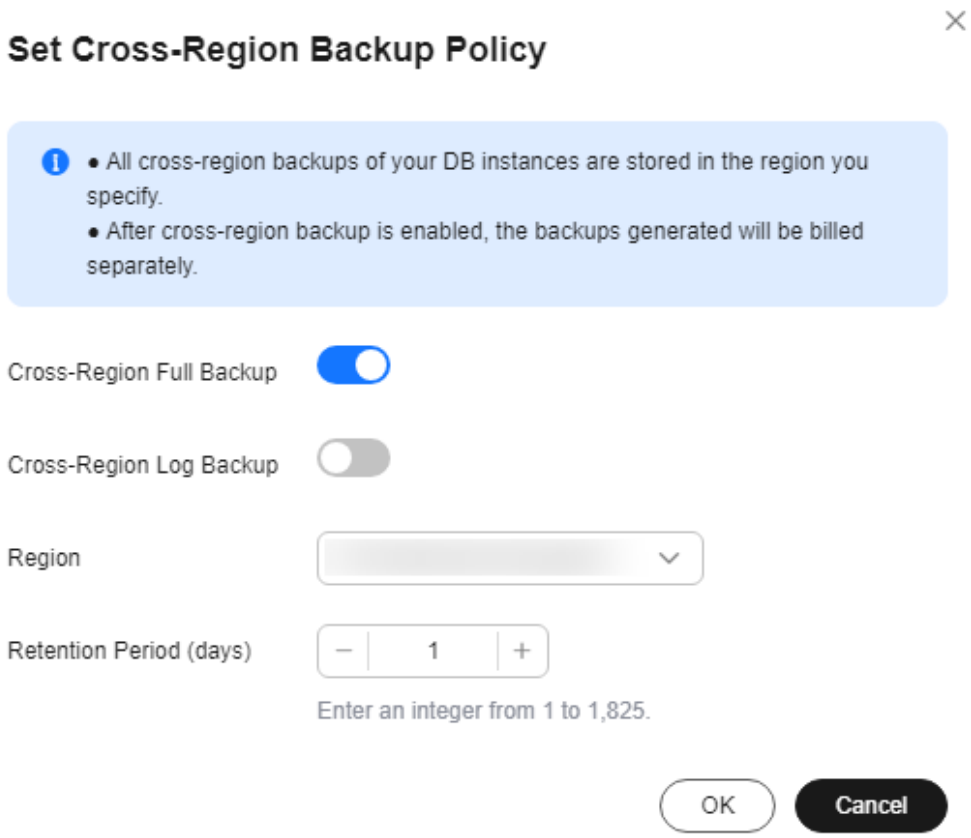
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance you want to configure a backup policy for.

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

Step 6 In the displayed dialog box, enable only **Cross-Region Full Backup** or enable both **Cross-Region Full Backup** and **Cross-Region Log Backup**, and specify the region and retention period.

Figure 1-86 Setting a cross-region backup policy



- Encrypted backups cannot be stored across regions.
- The instance name is synchronized only when cross-region backups are synchronized for the first time. The instance ID is the unique identifier for the cross-region backups.
- 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.
- Only new backup files generated after you set a cross-region backup policy will be stored in OBS in the region you specify.
- After cross-region log backup is enabled, you can restore a DB instance to a specified point in time 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.
- The cross-region backups of all your instances must be stored in the same destination region.
- After you enable cross-region backup, the completed backup files in the region where your instance is located will be asynchronously replicated to the region you specify.

Step 7 Click **OK**.

Step 8 To manage cross-region backups, return to the instance list, choose **Backups** in the navigation pane, and click the **Cross-Region Backups** tab.

- By default, all instances with cross-region backups are displayed.
  - 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.

Figure 1-87 Full backups

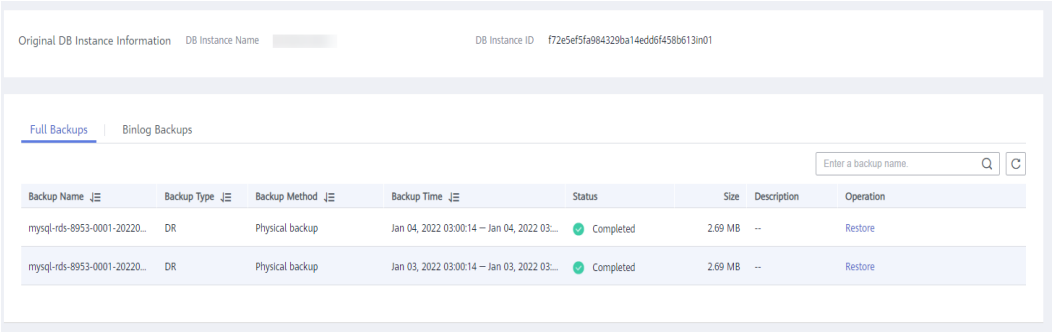
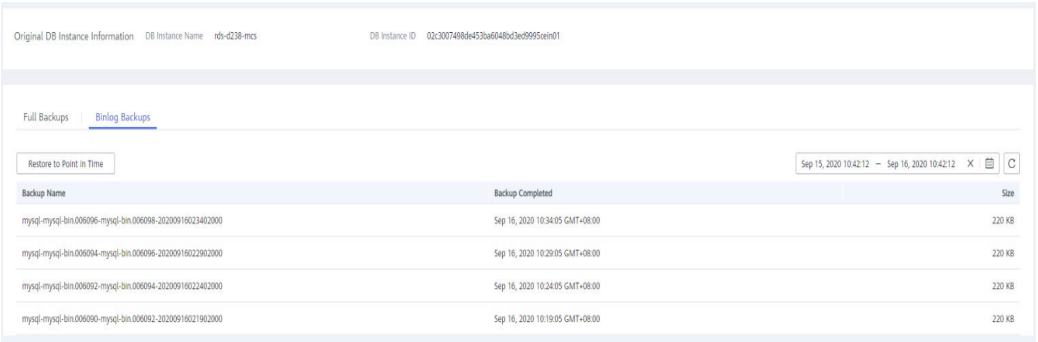


Figure 1-88 Binlog backups



----End

Disabling 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

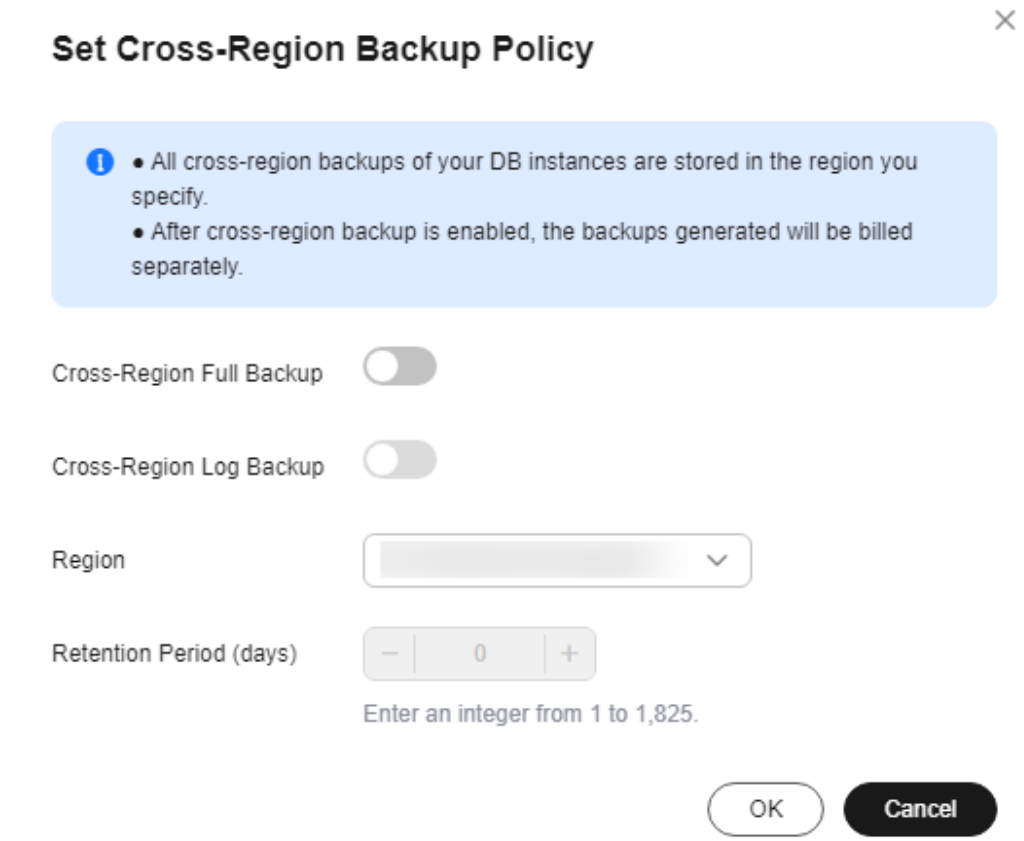
Step 4 On the **Backups** page, click the **Cross-Region Backups** tab.

Step 5 Locate a DB instance and click **Set Cross-Region Backup** in the **Operation** column. On the displayed page, disable the cross-region backup policy.

NOTE

Disabling the cross-region backup policy deletes the backups stored in the destination region.

Figure 1-89 Disabling a cross-region backup policy



Step 6 Click **OK**.

----End

1.9.3.3 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.

Constraints (CBR Disabled)

- You can create manual backups only when your account balance is no less than $0 USD.
- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- 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.
- Performing backups consumes memory resources. If there are a large number of tables in your instance but the available memory is insufficient, a backup task may fail. In this case, you need to reduce the number of tables or upgrade the instance specifications.

Constraints (CBR Enabled)

- The backup time is proportional to how much data your instance has. Too much data can decrease the backup efficiency. If you have large amounts of data and want to speed up the backup process, **submit a service ticket** to enable **Cloud Backup and Recovery (CBR)**.
- After CBR is enabled, snapshot backup is used. Existing automated and manual backups can still be used to restore data.
- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- DDL operations cannot be performed when a CBR snapshot is being created. If a DDL operation is being performed, a snapshot will be created after the DDL operation is complete.
- After CBR is enabled, the next full backup is a snapshot backup. You can use the snapshot backup to restore data.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required. For details, see **Product Pricing Details**.

After CBR is enabled, the free backup space is unavailable. You are billed for database server backup vaults on a pay-per-use basis. For details, see **How Is CBR Billed?**

Method 1

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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**.

- 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.

To check whether the backup has been created, you can click  in the upper right corner of the page to check the DB instance status. If the DB instance status becomes **Available** from **Backing up**, the backup has been created.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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**.

- 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 **Backups** page.

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

----End

1.9.3.4 Replicating a Backup

Scenarios

RDS supports replication of automated and manual backups.

Constraints

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

Snapshot-based backups, including CBR snapshot-based backups, cannot be replicated.

Backup Retention Policy

- If a DB instance is deleted, the automated backups created for it are also deleted.
- If an **automated backup policy** is enabled, the automated backups will be deleted after the backup retention period expires.
- 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.
- Replicating a backup does not interrupt your services.

Billing

Backups are saved as packages in OBS buckets. For details, see **How Is RDS Backup Data Billed?**

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name. On the **Backups & Restorations** page, locate the backup to be replicated and click **Replicate** in the **Operation** column.

Alternatively, choose **Backups** in the navigation pane on the left. On the displayed page, locate the 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 carriage return characters or the following special characters: >!<"&'=

Step 6 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

1.9.4 Managing Backups

1.9.4.1 Downloading a Full Backup File

Scenarios

This section describes how to download a manual or an automated backup for local storage.

RDS for MySQL allows you to download full backup files in .qp format.

Constraints

- Full backup files of frozen DB instances cannot be downloaded.
- When you use OBS Browser+ to download backup data, there is no charge for the outbound traffic from OBS.
- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.

Method 1: Using OBS Browser+

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

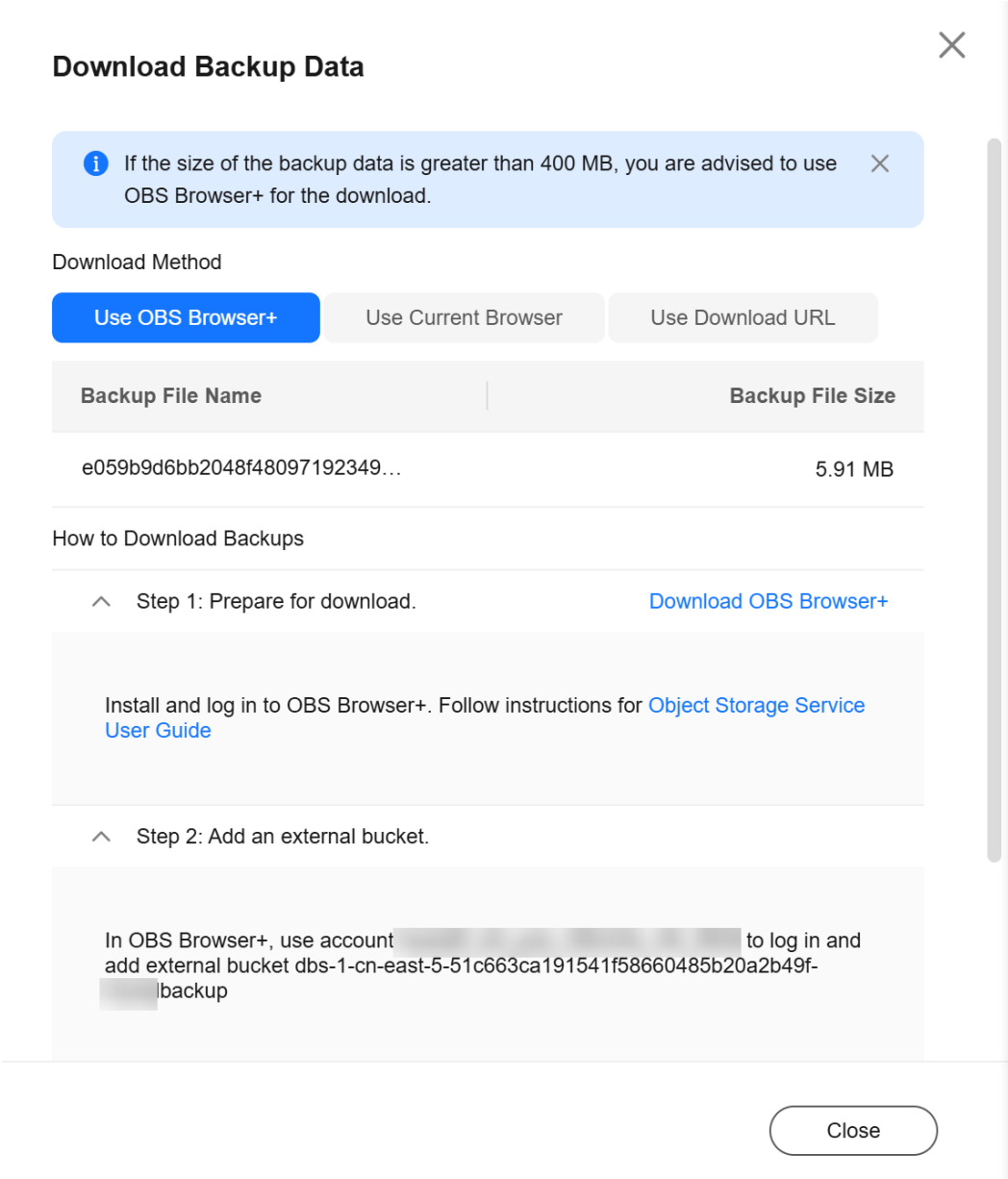
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

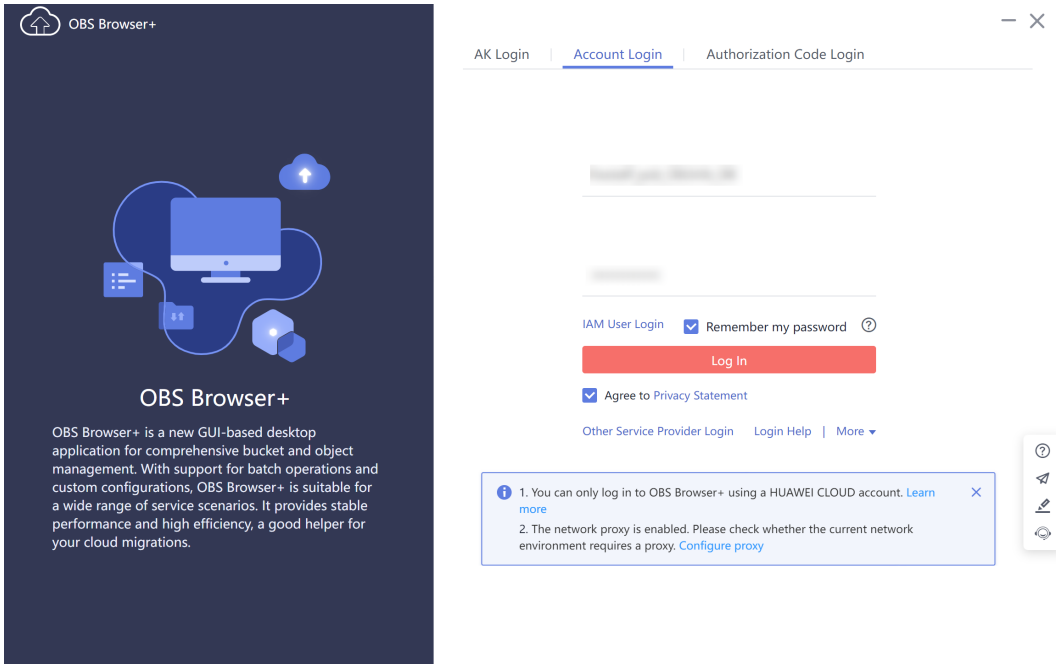
Step 6 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and download the backup as prompted.

Figure 1-90 Using OBS Browser+



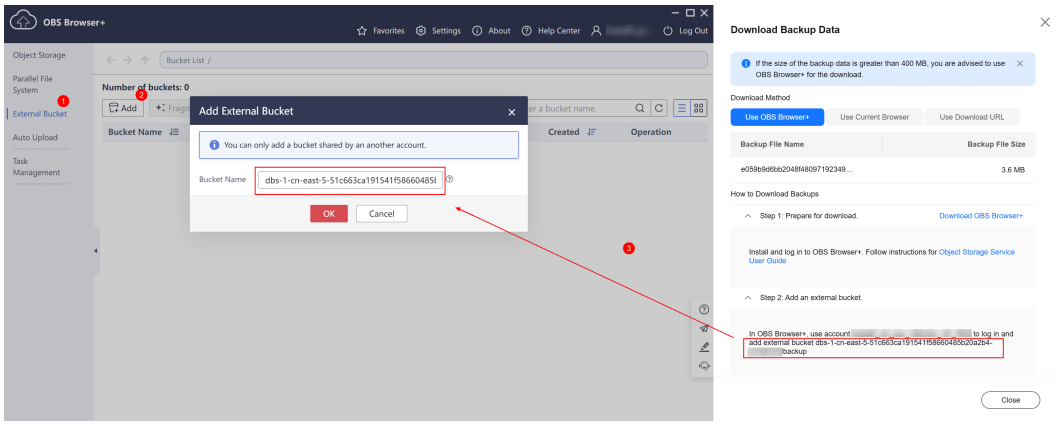
1. Download OBS Browser+ by clicking **Download OBS Browser+** in Step 1 on the download guide page.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

Figure 1-91 Logging in to OBS Browser+



4. Add an external bucket using the bucket name provided in step 2 on the download guide page.

Figure 1-92 Adding an external bucket



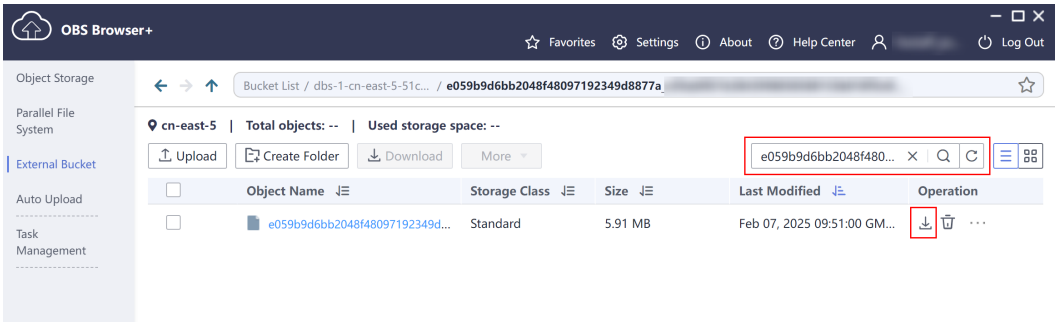
NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name and start a search. In the search result, locate the target backup and click in the **Operation** column.

Figure 1-93 Downloading a backup



----End

Method 2: Using Current Browser

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

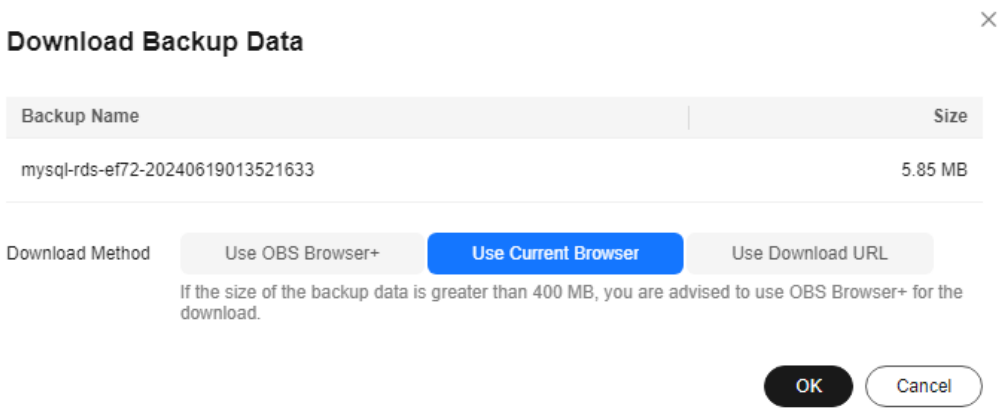
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 In the displayed dialog box, select **Use Current Browser** for **Download Method** and click **OK**.

Figure 1-94 Using the current browser



----End

Method 3: Using Download URL

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

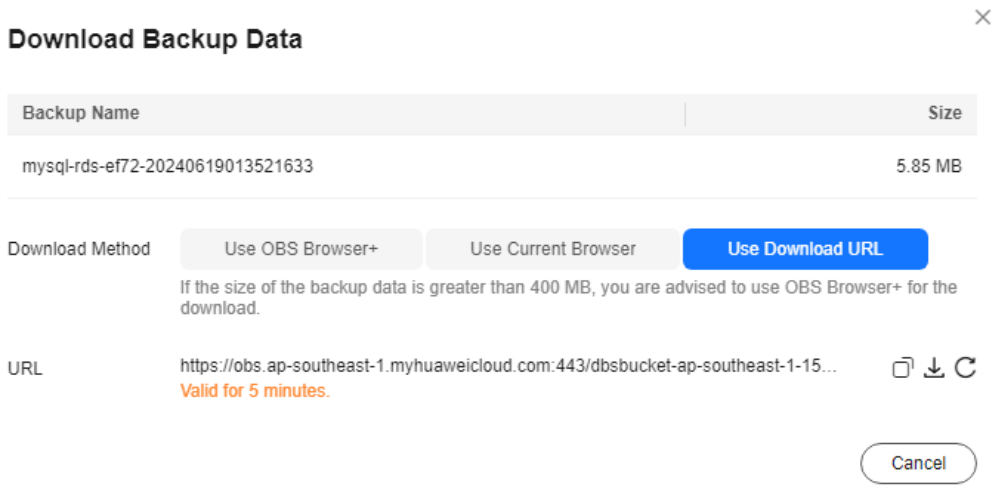
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 In the displayed dialog box, select **Use Download URL** for **Download Method**, click to copy the URL, and enter the URL in your browser.

Figure 1-95 Using the download URL



A valid URL for downloading the backup data is displayed.

- You can use various download tools, such as your browser 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

1.9.4.2 Downloading a Binlog Backup File

Scenarios

RDS for MySQL allows you to download binlog backup files for local storage. For details, see **Downloading a Binlog Backup File** or **Downloading a Merged Binlog**.

NOTE

The completion time displayed in the binlog backup file list indicates the time when the last transaction was submitted.

Binlog backups on the management console are named in the format of "binlog name +timestamp" and use the row-based logging.

Binlog backup files of frozen DB instances cannot be downloaded.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

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 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see *Identity and Access Management User Guide*.

Step 7 After the download is complete, you can view the binlog backups on your computer.

----End

Downloading a Merged Binlog

NOTICE

If the total size of binlogs within the selected period is greater than 500 MB, the binlogs cannot be merged.

When binlogs of a single-node instance are being merged, the CPU usage of the instance increases. Consider the impact on the instance performance before merging binlogs.

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the **Merged Binlogs** page, select a binlog time range and click **Merge**.

NOTE

- The maximum time range can be merged is 24 hours.
- The available time range is consistent with the retention period you have set for the automated backups. For details about how to set the retention period, see **Configuring an Intra-Region Backup Policy**.

Step 6 During the merging process, the merged file status is **Merging**. Wait until the status becomes **Merged successfully** and click **Download** in the **Operation** column.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 8 In the displayed dialog box, select a method to download the merged binlog.

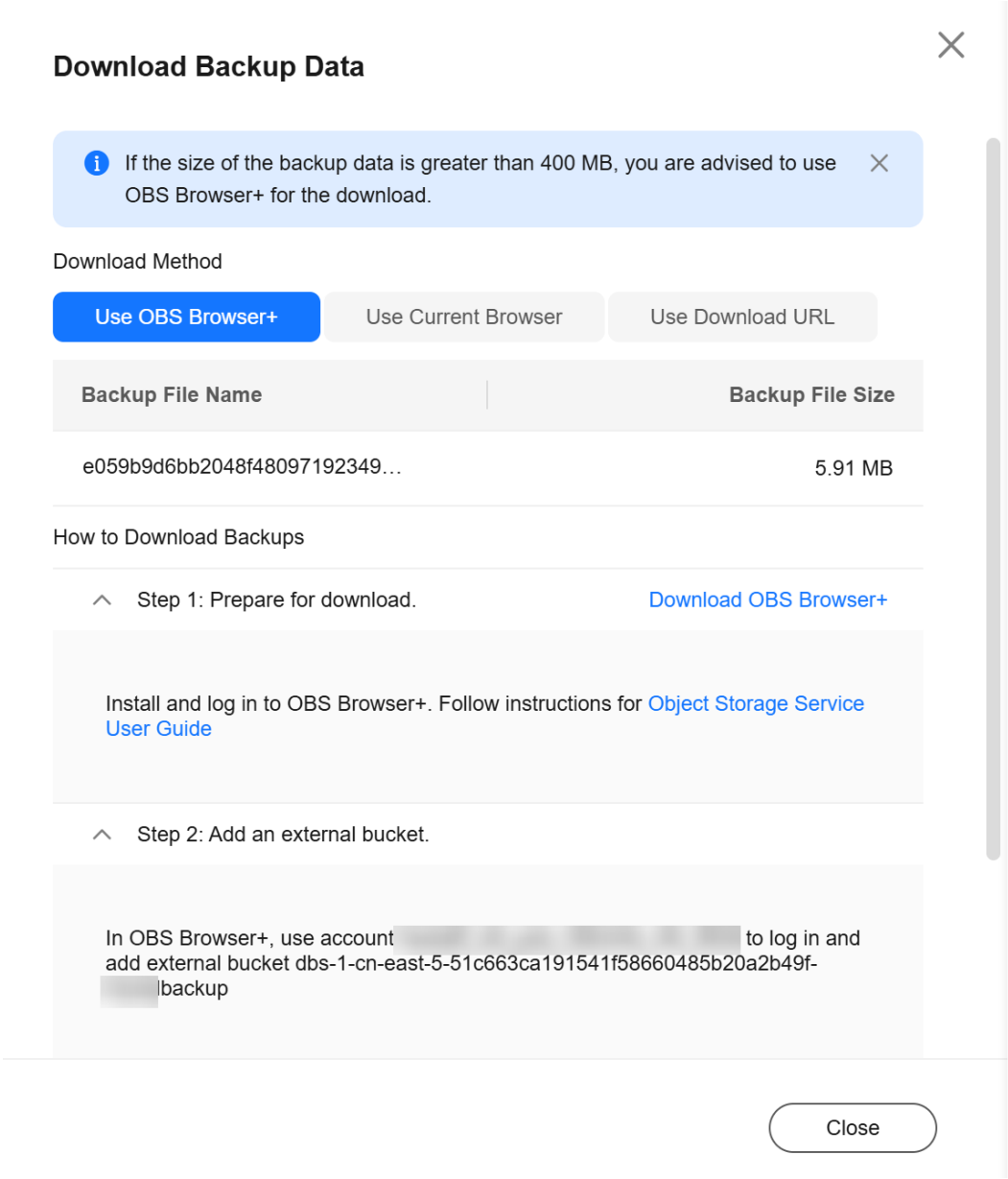
NOTE

- To reduce backup storage usage, delete the merged binlog after the download is complete. On the **Merged Binlogs** page, you can locate the target merged binlog to be deleted and click **Delete** in the **Operation** column.
- If you do not manually delete the merged binlogs, they will be deleted 30 days later.

- **Use OBS Browser+**

  When you use OBS Browser+ to download backup data, there is no charge for the generated outbound traffic.

Figure 1-96 Using OBS Browser+ to download a merged binlog



a. Download OBS Browser+ following step 1 provided on the download guide page.

b. Decompress and install OBS Browser+.

c. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

   For details about how to log in to OBS Browser+, see **Logging In to OBS Browser+** in the *Object Storage Service Tools Guide*.

d. Add an external bucket using the bucket name provided in step 2 on the download guide page.

   In the **Add External Bucket** dialog box of OBS Browser+, enter the bucket name provided in step 2 on the download guide page, and click **OK**.

NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

e. Download a merged binlog.

On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name and start a search. In the search result, locate the target backup and click in the **Operation** column.

- **Use Current Browser**

  Download the merged binlog directly from the current browser.

- **Use Download URL**

  Click to copy the URL within the validity period to download the merged binlog.

  – You can use other download tools to download the merged binlog.

  – You can also run the following command to download the merged binlog:

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

  Variables in the command are described as follows:

  FILE_NAME: indicates the new name of the merged binlog file. 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 merged binlog to be downloaded. If the location contains special characters, escape is required.

----End

1.9.4.3 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

For details about how to export backup data, see **Downloading a Full Backup File** and **Downloading a Binlog Backup File**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane, choose **Backups**. On the displayed page, select the backups you want to export and click **Export** to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.
- The backup information is exported to an Excel file for your further analysis.

Figure 1-97 RDS for MySQL backup information

	A	B	C	D	E	F	G	H	I	J	K	L
1	Backup ID	Backup Name	DB Instance Name	DB Instance ID	Key ID	DB Engine	Backup Type	Backup Method	Backup Time	Status	Size	Description
2	b9c5685880d74842adb023df78894606br01	mysql-rds-19e8-20231126181148953	rds-19e8	b8ba9799b1a24b59bbaa10a74eba422ain01	--	MySQL 5.7.41	Automated	Physical backup	Nov 27, 2023	Completed	3.54 MB	--
3	184d1ca4b967499faa9dbd385d5d97e0br01	mysql-rds-19e8-20231125181145616	rds-19e8	b8ba9799b1a24b59bbaa10a74eba422ain01	--	MySQL 5.7.41	Automated	Physical backup	Nov 26, 2023	Completed	3.53 MB	--
4	09e8bffa5575432baabff6a909cc5d7ebr01	mysql-rds-19e8-20231124181146092	rds-19e8	b8ba9799b1a24b59bbaa10a74eba422ain01	--	MySQL 5.7.41	Automated	Physical backup	Nov 25, 2023	Completed	3.53 MB	--
5	0bcac64afd2b4bcaa88fa0176f97e378br01	mysql-rds-19e8-20231124021028785	rds-19e8	b8ba9799b1a24b59bbaa10a74eba422ain01	--	MySQL 5.7.41	Automated	Physical backup	Nov 24, 2023	Completed	3.51 MB	--
6												

Step 5 View the exported backup information.

----End

1.9.4.4 Using mysqlbinlog to View Binlogs

Scenarios

The mysqlbinlog tool is used to parse binlogs and is contained in the MySQL software package. You can download a MySQL software package of your desired version from the MySQL official website, decompress the package, and obtain the mysqlbinlog tool from the decompressed package (mysqlbinlog 3.4 is for MySQL 5.6 and 5.7). If your mysqlbinlog version is too old to correctly parse binlogs, perform the operations described in this section.

You can also use a third-party tool to parse binlogs for RDS for MySQL.

Procedure

1. **Download a MySQL software package.**

NOTICE

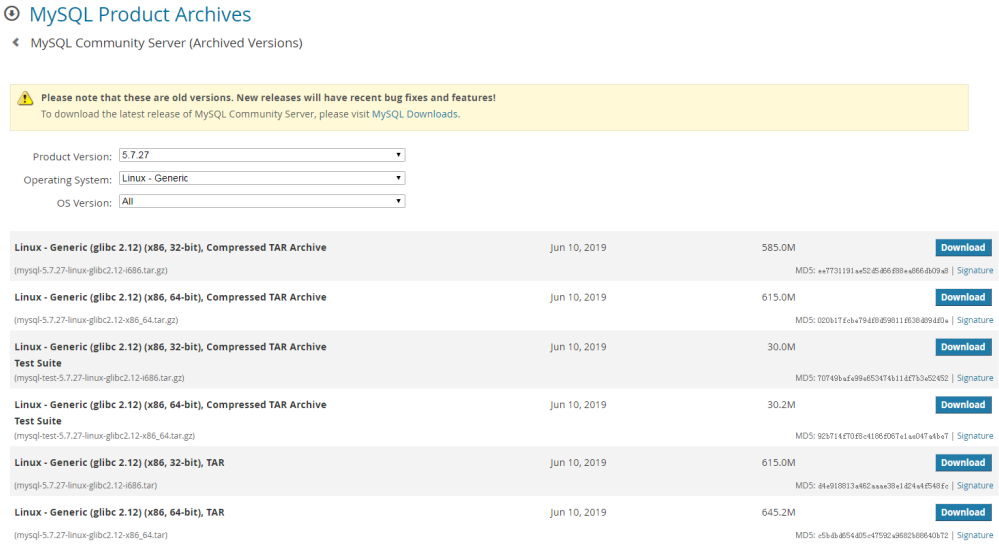
It is recommended that the software package version be the same as your current MySQL major version.

If your MySQL version is 5.7.27, download the following software packages:

- Product Version: 5.7.27
- Operating System: Linux-Generic

The downloaded MySQL software package is **mysql-5.7.27-linux-glibc2.12-x86_64.tar.gz**.

Figure 1-98 Download



2. Decompress the software package and find the mysqlbinlog tool.
3. Find the mysqlbinlog tool version.

```
[root@ecs]# tar -zxf mysql-5.7.27-linux-glibc2.12-x86_64.tar.gz
[root@ecs]# cd mysql-5.7.27-linux-glibc2.12-x86_64/bin
[root@ecs]# ll mysqlbinlog
-rwxr-xr-x 1 7161 31415 11310886 Jun 10 2019 mysqlbinlog
[root@ecs]# ./mysqlbinlog -V
./mysqlbinlog Ver 3.4 for linux-glibc2.12 at x86_64
```

4. Use mysqlbinlog to parse binlogs.

   The following uses mysql-bin.000001 as an example:

```
[root@ecs]# ./mysqlbinlog --no-defaults -vv /root/mysql-bin.000001
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=1*/;
/*!50003 SET @OLD_COMPLETION_TYPE=@@COMPLETION_TYPE,COMPLETION_TYPE=0*/;
DELIMITER /*!*/;
# at 4
#200316 17:54:14 server id 1 end_log_pos 126 CRC32 0x92b3f2ca Start: binlog v
4, server v 5.7.27-5-debug-log created 200316 17:54:14 at startup
ROLLBACK/*!*/;
BINLOG '
xkxvXg8BAAAAegAAAH4AAAAAAAQANS43LjI3LTUtZGVidWctbG9nAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAADGTG9eEzgNAAgAEgAEBAQEEgAAYgAEGggAAAAICAgCAAAACgoKKioAEjQA
Xz0AAcrys5I=
'/*!*/;
# at 126
#200316 17:54:14 server id 1 end_log_pos 157 CRC32 0xfcc47ad6 Previous-GTIDs
# [empty]
# at 157
#200316 17:54:27 server id 1 end_log_pos 204 CRC32 0xa7febd1f Rotate to mysqlbin.
000002 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*/;
```

1.9.4.5 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

1.9.5 Clearing Binlogs

1.9.5.1 Setting a Local Retention Period for RDS 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

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

On the RDS console, you can set the binlog retention period only for the primary instance. The binlog retention period for read replicas is the same as that of the primary instance.

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

If the retention period is set to **0**, the binlogs of your DB instance will be deleted once they are synchronized to the standby instance and read replicas and

successfully backed up to OBS. If the retention period is set to a value greater than 0, for example, 1 day, the binlogs will be retained for one day after they are synchronized to the standby instance and read replicas from the primary instance and successfully backed up to OBS. After the retention period expires, the binlogs will be automatically deleted. For details about how to view binlogs, see **Downloading a Binlog Backup File**.

Precautions

The binlog retention period is measured in hours on the console. However, the values of **expire_logs_days** (MySQL 5.7) and **binlog_expire_logs_seconds** (MySQL 8.0) are measured in days when you query the binlog retention period by running a command, which cannot be used as a reference. To check how long the binlogs can be retained, view the binlog retention period on the console.

Procedure

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

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

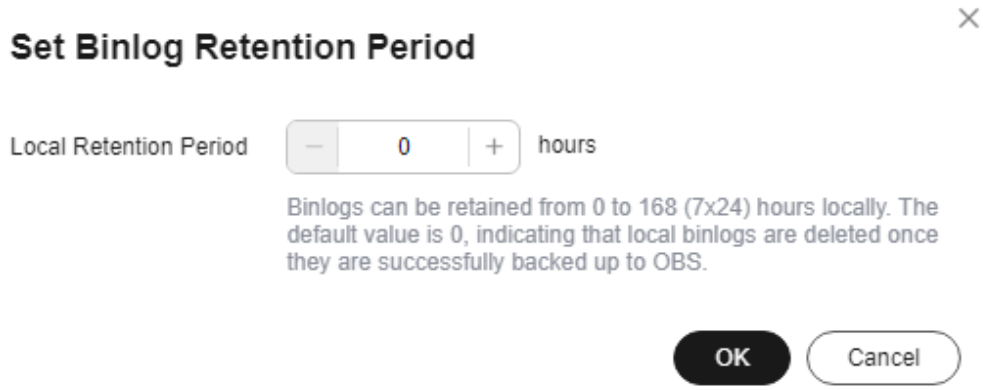
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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 1-99 Setting the binlog retention period



NOTE

When binlogs are deleted depends on the local retention period you configure on the console.

----End

1.9.5.2 Clearing Binlogs from DB Instances

Scenarios

You can clear local binlogs with a few clicks for RDS for MySQL instances to free up storage space.

Binlog Clearing Based on Specified Retention Period

RDS for MySQL allows you to clear the binlogs exceeding the **specified retention period** with just a few clicks.

Precautions

- There is a delay in clearing binlogs. Every time after clearing binlogs, check whether the **storage** occupied by these binlogs was released. Do not submit the request multiple times.
- The following binlogs cannot be cleared using this function:
  - Binlogs that are not backed up to OBS yet
  - Binlogs that have not been synchronized to the standby instance
  - Binlogs that have not been received by the incremental backup parsing tool

Procedure

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

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

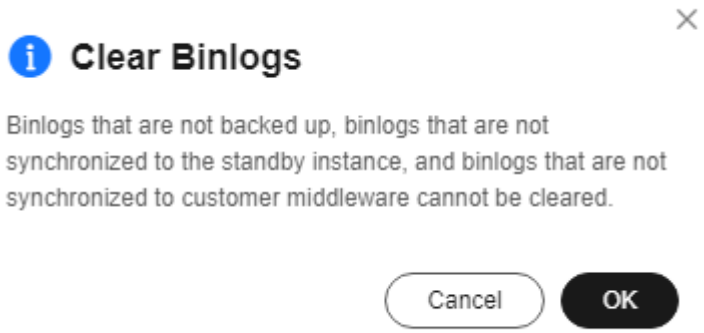
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane, choose **Backups & Restorations**. On the **Binlog Backups** tab page, click **Clear Binlogs**.

Step 6 In the displayed dialog box, click **OK**.

Figure 1-100 Clearing binlogs



----End

1.10 Data Restorations

1.10.1 Restoration Solutions

If your database is damaged or mistakenly deleted, you can restore it from backups.

Restoring a Mistakenly Deleted Instance

- RDS moves unsubscribed yearly/monthly instances and deleted pay-per-use instances to the recycle bin. You can **rebuild an instance** that was deleted up to 7 days ago from the recycle bin.
- You can restore a deleted instance from its retained manual backups. For details, see **Restoring a DB Instance from Backups**.

Restoring Mistakenly Deleted or Modified Data of a DB Instance

Table 1-50 Restoration solutions

Solution	Category	Storage Type	Restoration Time Point	Scope	Restore To	Time Required

		Cloud SSD	Extreme SSD	Point in Time When a Backup Was Generated	All Databases and Tables	Certain Databases and Tables	New Instance	Original Instance	Existing Instance Other than the Original Instance	
Restoring an entire instance	Restoration from backups	√	√	x	√	x	√	√	√	Depending on how much data there is in the instance
	Point-in-time recovery (PITR)	√	√	√	√	x	√	√	√	Depending on how much data there is in the instance

Restoring databases and tables	Standard restoration	√	√	√	x	√	x	√	x	Depending on how much data there is in the instance and in the databases and tables
	Fast restoration	√	√	√	x	√	x	√	x	Depending on how much data there is in the databases and tables

Restoring or Migrating Data to an RDS for MySQL Instance

- You can restore data to an RDS for MySQL instance from backups. For details, see **Restoring Data to RDS for MySQL**.
- You can migrate data to an RDS for MySQL instance using Data Replication Service (DRS), mysqldump, or Data Admin Service (DAS). For details, see **Migration Solution Overview**.

Restoring or Migrating Data to a Self-Managed MySQL Database

- You can restore data to a self-managed MySQL database from backups. For details, see **Restoring Data to an On-Premises MySQL Database**.
- You can migrate data to a self-managed MySQL database using DRS. For details, see **From MySQL to MySQL**.

1.10.2 Restoring Data to RDS for MySQL

1.10.2.1 Restoring a DB Instance from Backups

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, the backup file is downloaded from OBS and then restored to the DB instance at an average speed of 100 MB/s.

Function Description

Table 1-51 Function description

Item	Description
Restoration scope	The entire instance
Instance data after restoration	The instance data after restoration is consistent with that in the full backup used for the restoration. • Restoring data to a new instance creates an instance with the same data as that in the backup. • Restoring data to the original or an existing instance will overwrite the instance data.
Restoration type	• Restoration to a new instance • Restoration to the original instance • Restoration to an existing instance other than the original one
Configurations for restoring to a new instance	• The DB engine and engine version of the new instance are the same as those of the original instance. • The storage space of the new instance is the same as that of the original instance by default and the new instance must be at least as large as the original instance. • Other parameters need to be reconfigured.
Time required	The time required depends on how much data there is in the instance. The average restoration speed is 100 MB/s.

Constraints

- You can restore to new DB instances from backups only when your account balance is greater than or equal to $0 USD. You will pay for the new instance specifications.
- If transparent page compression is enabled by specifying attributes in the CREATE TABLE statement for the original DB instance, the restoration may fail due to insufficient storage space.
- Constraints on restoring data to the original DB instance:
  - 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 all existing data and the DB instance will be unavailable during the restoration process.
- Constraints on restoring data to an existing DB instance:
  - 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 use the same DB engine and the same or a later version than the original DB instance.
  - Ensure that the storage space of the selected existing DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.
- If SQL statement concurrency control is enabled for the DB instance, different constraints apply to different restoration scenarios:
  - Restoration to a new instance: In RDS for MySQL 5.7, the original concurrency control rules become invalid. In RDS for MySQL 5.6 and 8.0, the concurrency control rules of the original instance are retained.
  - Restoration to the original instance: The concurrency control rules of the original instance are restored to the state when the backup was created.
  - Restoration to an existing instance: In RDS for MySQL 5.7, all concurrency control rules of the target instance become invalid. In RDS for MySQL 5.6 and 8.0, the rules of the original instance will overwrite those of the target instance.

Supported Storage Types

If you restore data to your original instance, the storage type of the instance does not change. If you restore data to a new instance or an existing instance other than the original one, the storage types supported are listed in **Table 1-52**.

Table 1-52 Supported storage types

Original Storage Type	Restore To	New Storage Type
Cloud SSD	New instance and existing instance other than the original instance	Cloud SSD
Extreme SSD	New instance and existing instance other than the original instance	• Cloud SSD • Extreme SSD

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

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

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

NOTE

If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

- Create New Instance

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

  - The DB engine and engine version of the new instance are the same as those of the original instance.
  - 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 **Buying an RDS for MySQL DB Instance**.

Figure 1-101 Restoring to a new DB instance

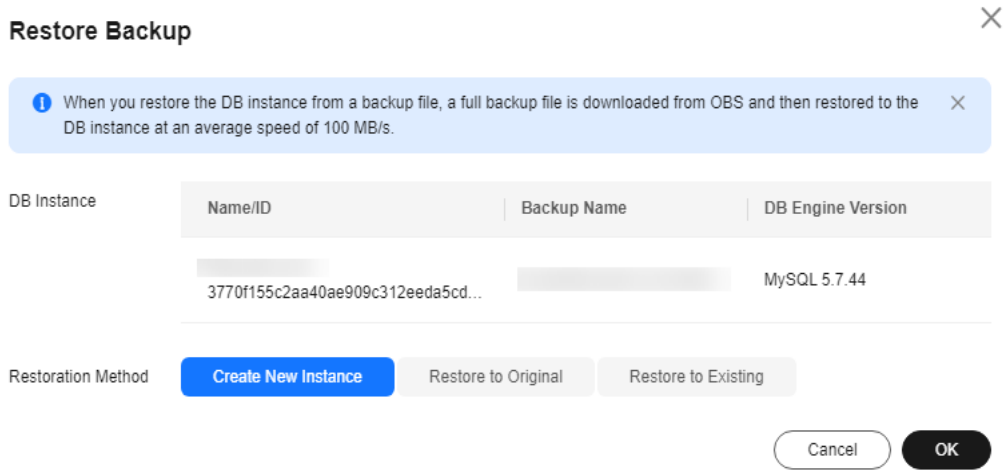
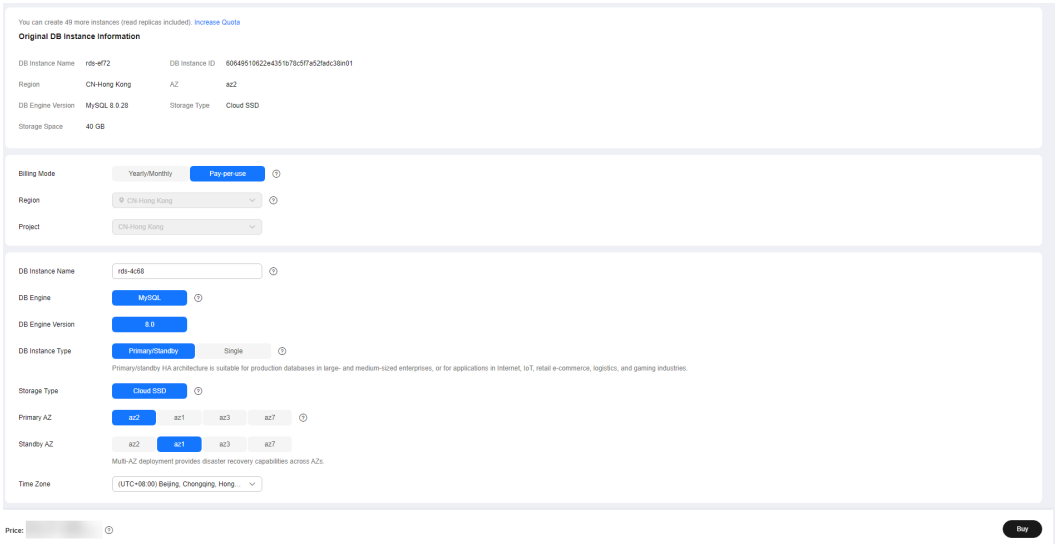


Figure 1-102 Creating a new instance



- Restore to Original
  a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.
  b. Confirm the information and click **OK**.
- Restore to Existing
  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored.
  b. Select an existing instance and click **Next**.
  c. Confirm the information and click **OK**.

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 Original

  On the **Instances** page, the status of the original DB instance changes from **Restoring** to **Available**. If the original DB instance contains read replicas, the read replica status is the same as the original DB instance status.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** 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.

  You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Viewing a Task**.

----End

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

FAQs

How Can I Restore Data If No Backup Is Available?

1.10.2.2 Restoring a DB Instance to a Point in Time

Scenarios

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

You can restore one or multiple DB instances at a 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 80 MB/s.

Function Description

Table 1-53 Function description

Item	Description
Restoration scope	The entire instance
Instance data after restoration	The instance data after restoration is consistent with that in the full backup plus the incremental backup used for the restoration. • Restoring data to a new instance creates an instance with the same data as that generated by that time point. • Restoring data to the original or an existing instance will overwrite the instance data.
Restorable time point	Any time point within the retention period after the earliest full backup is generated
Scenario	• Restoration to a new instance • Restoration to the original instance • Restoration to an existing instance other than the original one
Configurations for restoring to a new instance	• The DB engine and engine version of the new instance are the same as those of the original instance. • Other parameters need to be reconfigured.
Time required	The time required depends on how much data there is in the instance. The average restoration speed is 80 MB/s.

Constraints

- You can restore to new DB instances from backups only when your account balance is greater than or equal to $0 USD. You will pay for the new instance specifications.
- Do not run the **reset master** command on RDS for MySQL DB instances within their lifecycle. Otherwise, an exception may occur when restoring an RDS for MySQL DB instance to a specified point in time.
- When you restore data to a new DB instance, large transactions in the original DB instance backup may cause a restoration failure. If the restoration fails, **submit a service ticket**.
- Constraints on restoring data to the original DB instance:
  - Restoring to the original DB instance will overwrite data on it and cause the DB instance to be unavailable during the restoration.
- Constraints on restoring data to an existing DB instance:

  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable during the restoration.
  - To restore backup data to an existing DB instance, the selected DB instance must use the same DB engine and the same or a 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.
- If SQL statement concurrency control is enabled for the DB instance, different constraints apply to different restoration scenarios:
  - Restoration to a new instance: In RDS for MySQL 5.7, the original concurrency control rules become invalid. In RDS for MySQL 5.6 and 8.0, the concurrency control rules of the original instance are retained.
  - Restoration to the original instance: The concurrency control rules of the original instance are restored to the state when the backup was created.
  - Restoration to an existing instance: In RDS for MySQL 5.7, all concurrency control rules of the target instance become invalid. In RDS for MySQL 5.6 and 8.0, the rules of the original instance will overwrite those of the target instance.

Restoring a DB Instance

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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**.

NOTE

If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

- 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 **Buying an RDS for MySQL DB Instance**.
- Restore to Original

a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.

b. Confirm the information and click **OK**.

- Restore to Existing

a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored.

b. Select an existing instance and click **Next**.

c. Confirm the information and click **OK**.

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.

- Restore to Original

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  A new restoration time range is available. There will be a difference between the new and original time ranges. This difference reflects the duration of the restoration.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Viewing a Task**.

  After the restoration is complete, a full backup will be automatically triggered.

----End

Restoring Multiple DB Instances at a Time

NOTE

Only users with the batch restoration permission can restore data to a specified time point in batches. You can contact customer service to apply for the required permission.

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

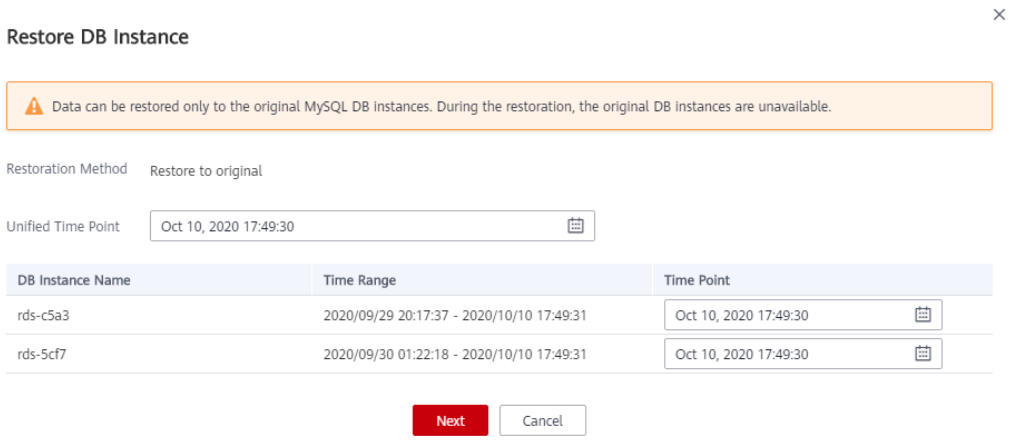
Step 4 On the **Instances** page, select target DB instances and choose **More** > **Restore** above the DB instance list.

NOTE

Only RDS for MySQL DB instances support batch restoration.

Step 5 In the displayed dialog box, set a unified restoration time point, or select different time points for different DB instances.

Figure 1-103 Batch restoration



Step 6 Click **Next** to confirm the information.

Step 7 Click **OK** to submit the batch restoration task.

Step 8 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

FAQs

How Can I Restore Data If No Backup Is Available?

1.10.2.3 Restoring Databases or Tables to a Point in Time

Scenarios

RDS allows you to restore databases or tables using point-in-time recovery (PITR). This ensures your data integrity and minimizes impact on the original instance performance. You can select databases or tables and restore them to a specified point in time. During database or table PITR, RDS downloads the most recent full backup from OBS and restores it to a temporary DB instance, and then replays binlogs to the specified point in time on the temporary instance. After that, data on the temporary instance is written to the target databases or tables of the original instance at an average speed of 35 MB/s.

The time required depends on the amount of data to be restored on the DB instance. Restoring databases or tables will not overwrite data in the DB instance. You can select the databases or tables to be restored.

RDS for MySQL allows you to restore databases or tables of one instance or multiple instances at a time.

Constraints

- Take care when restoring tables. Improper operations can cause instance or service exceptions.
- During a standard or fast table restoration, the foreign keys of tables containing foreign keys will be deleted and the table structures will be changed.
- During table PITR, a maximum of 2,000 tables can be restored for one instance at a time.
- During database PITR, a maximum of 1,000 databases and 20,000 tables can be restored for a single instance at a time.
- If you want to restore databases or tables of multiple instances at a time, the instances must be of the same engine version and in the **Available** state.
- You can select a maximum of 20 instances at a time for PITR.
- During the PITR, DB instances and read replicas cannot be rebooted or deleted, and their instance specifications cannot be modified.
- During the PITR, the database or table information to be restored is read from the latest full backup before the selected time point. You can select any time point within the restoration time range. Therefore, a database or table can be restored to the earliest full backup time point when its information exists.
- If a table you selected does not exist at the specified point in time, the table will not be restored.
- Table-level PITR does not support view restoration. To restore a view, restore the tables involved in the view and create the view again.
- Database-level PITR restores only table data in the databases. The new databases generated after restoration do not contain views.
- If a DB instance has more than 20,000 tables, RDS does not collect the database and table metadata at a historical time point for performance purposes. Instead, RDS searches for the database and table information from the current instance for restoration. If the target database and table are not

displayed but they do exist at the specified time point, you can create an empty database and table with the same names and restore them.

- During database-level PITR, the table names in the original database cannot contain periods (.). Otherwise, the restoration task may fail.

Prerequisites

After the restoration, a new database or table will be generated in the DB instance. Ensure that the DB instance has sufficient storage space for the generated database or table.

Restoring Databases or Tables of One Instance

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

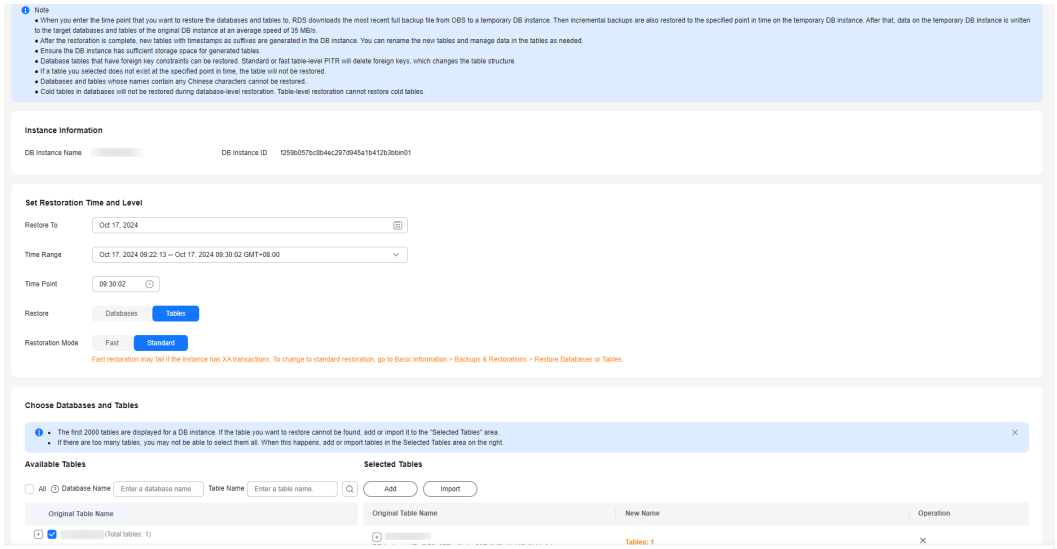
Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Backups & Restorations**. On the displayed page, click **Restore Databases or Tables**.

Step 6 Specify restoration information and click **Next: Confirm**.

- To facilitate your operations, you can search for the databases or tables to be restored.
- After the restoration is complete, new databases or tables with timestamps appended as suffixes to original database or table names are generated in the DB instance. You can rename the new databases or tables.
- The new table name must be unique and consist of 1 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.
- Databases whose names contain periods (.) cannot be restored.
- Database-level PITR and table-level PITR are not supported for databases and tables that contain JSON virtual columns.
- Cold tables cannot be restored when you restore databases or tables using PITR.
- During database PITR, if you create new databases with the same names as the restored databases, the data in the new databases may be overwritten, causing data loss.
- During table PITR, if you create new tables with the same names as the restored tables, the data in the new tables may be overwritten, causing data loss.
- If the DB instance and the selected time point support fast restoration, you can select **Fast** for **Restoration Mode**.
- If your DB instance has XA transactions, select **Standard** for **Restoration Mode** because fast restoration can cause data loss.
- During a fast restoration, if the target table is renamed in the replayed binlog, the target table may fail to be restored.

Figure 1-104 Restoring databases or tables to a point in time



Step 7 On the displayed page, confirm the information and click **Submit**.

Step 8 On the **Instances** page, check that the DB instance status is **Restoring**. During the restoration, services are not interrupted.

You can also view the progress and result of restoring databases or tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the databases or tables as required.

----End

NOTE

- Data is restored at an average speed of 35 MB/s.
- Restoring databases or tables to a specified point in time does not affect new data. The restored database or table is a temporary database or table with a timestamp suffix. You can manage the data in the temporary database or table as required.

Restoring Databases or Tables of Multiple Instances in Batches

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

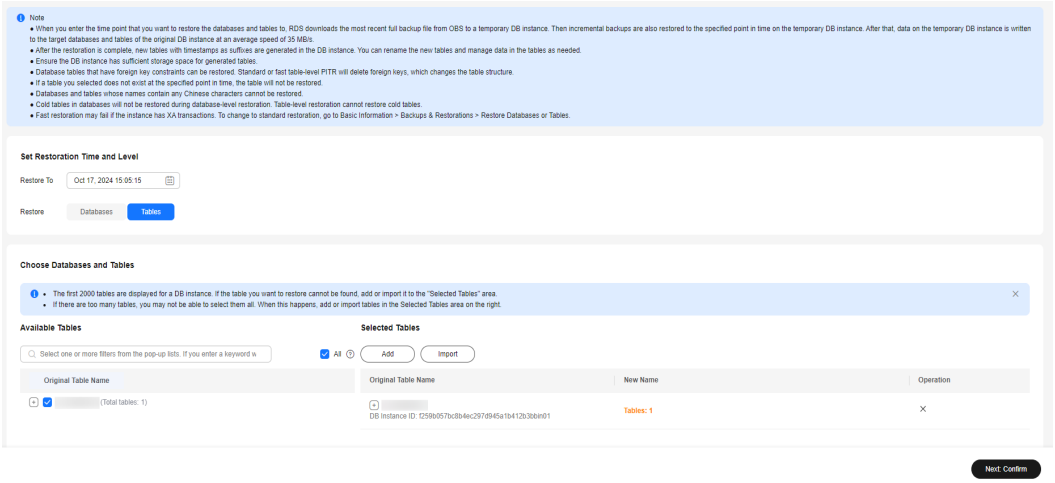
Step 4 On the **Instances** page, select multiple instances and choose **More** > **Restore Databases or Tables** above the instance list.

Step 5 Specify restoration information and click **Next: Confirm**.

- To facilitate your operations, you can search for the instances, databases, or tables to be restored by name.

- After the restoration is complete, new databases or tables with timestamps appended as suffixes to original database or table names are generated in the DB instance. You can rename the new databases or tables.
- The new table name must be unique and consist of 1 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.
- Databases whose names contain periods (.) cannot be restored.
- Cold tables cannot be restored when you restore databases or tables using PITR.
- During database PITR, if you create new databases with the same names as the restored databases, the data in the new databases may be overwritten, causing data loss.
- During table PITR, if you create new tables with the same names as the restored tables, the data in the new tables may be overwritten, causing data loss.
- During database PITR, a maximum of 2,000 databases and 20,000 tables can be restored for one instance at a time.

Figure 1-105 Restoring databases or tables of multiple instances in batches



Step 6 On the displayed page, confirm the information and click **Submit**.

Step 7 On the **Instances** page, check that the instances are **Restoring**. During the restoration process, services are not interrupted.

You can also view the progress and result of restoring databases or tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the databases or tables as required.

----End

NOTE

- Data is restored at an average speed of 35 MB/s.
- Restoring databases or tables to a specified point in time does not affect new data. The restored database or table is a temporary database or table with a timestamp suffix. You can manage the data in the temporary database or table as required.

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

FAQs

How Can I Restore Data If No Backup Is Available?

1.10.2.4 Restoring Data Across Regions

Scenarios

RDS for MySQL supports cross-region backups. If a DB instance in one region fails, you can use backups stored in another region to restore data to a new instance in that region.

Prerequisites

- Cross-region backups have been created. For details, see **Configuring a Cross-Region Backup Policy**.
- Encrypted backups cannot be used to restore data to existing instances.
- Instances with Transparent Data Encryption (TDE) enabled do not support cross-region restoration.
- The kernel version of the target instance cannot be earlier than that of the original instance.

Restoring a Full Backup Across Regions

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

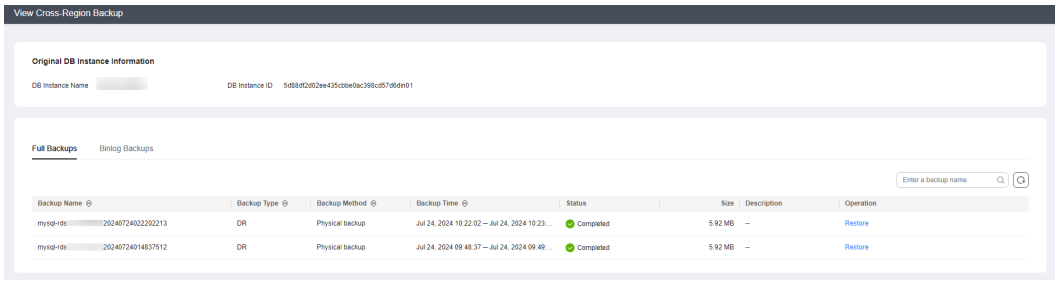
Step 4 In the upper part of the page, select the region where the backup is stored.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click the **Cross-Region Backups** tab.

Step 6 Locate the instance you want to restore and click **View Cross-Region Backup** in the **Operation** column.

Step 7 On the **Full Backups** page, locate the backup you want to restore and click **Restore** in the **Operation** column.

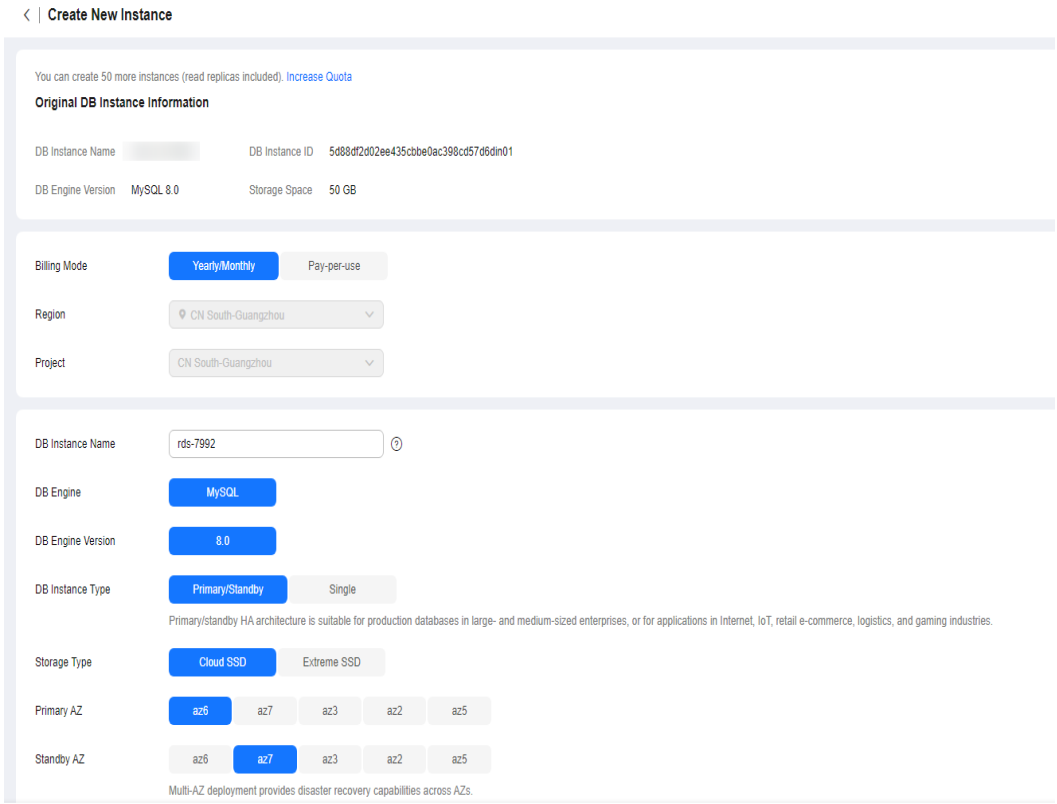
Figure 1-106 Full backups



Step 8 In the displayed dialog box, select a restoration method and click **OK**.

Create New Instance: Set parameters for the new instance and click **Buy**.

Figure 1-107 Creating a new instance



- The DB engine and version of the new instance are the same as those of the original 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 **Buying an RDS for MySQL DB Instance**.

----End

Restoring an Incremental Backup Across Regions

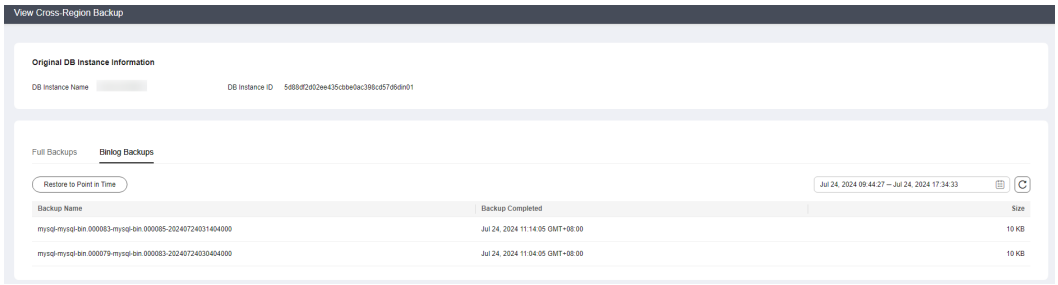
Step 1 In the upper part of the page, select the region where the backup is stored.

Step 2 In the navigation pane on the left, choose **Backups**. On the displayed page, click the **Cross-Region Backups** tab.

Step 3 Locate the instance you want to restore and click **View Cross-Region Backup** in the **Operation** column.

Step 4 On the **Binlog Backups** page, click **Restore to Point in Time**.

Figure 1-108 Restoring an incremental backup



Step 5 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: Set parameters for the new instance and click **Buy**.

- The DB engine and version of the new instance are the same as those of the original 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 **Buying an RDS for MySQL DB Instance**.

----End

1.10.3 Restoring Data to an On-Premises MySQL Database

This section describes how to restore a downloaded RDS for MySQL full backup to an on-premises MySQL database.

If you want to migrate all data from your RDS for MySQL DB instance to an on-premises MySQL database, you can download a full backup in **.qp** format and restore it to the on-premises database following the steps described in this section.

- **Step 1: Download a Full Backup of Your RDS for MySQL DB Instance**
- **Step 2: Install qpress and XtraBackup on the On-Premises MySQL Database**
- **Step 3: Restore the Backup to the On-Premises MySQL Database**

Operation Process

1. Download the target full backup of your RDS for MySQL DB instance.
2. Upload the full backup to the on-premises MySQL database.
3. Use the qpress tool to decompress the full backup.
4. Use the XtraBackup tool to restore the full backup to the **data** directory of the on-premises database and save the original data of the database to the **data_back** file.
5. Restart the on-premises database.

Constraints

- This section only covers restoring a full backup of an RDS for MySQL 5.6, 5.7 or 8.0 DB instance to an on-premises database of the corresponding version. Incremental backup restoration is not included.
- **The minor version of the on-premises MySQL database must be the same as that of your RDS for MySQL DB instance.**

  To view the MySQL kernel version, run **mysql -V** or **mysqld --version**.
- The backup can be restored only to an on-premises database running Linux.
- Since the open-source backup tool supports only the x86 package, you cannot restore a backup to a database built on an Arm-based ECS by following the operations described in this section. To restore data to a database built on an Arm-based ECS, you can use Data Replication Service (DRS) or data export and import. For details, see **Migration Solution Overview**.
- The following software is required to restore a full backup to an on-premises database:
  - MySQL database
  - qpress
  - Percona XtraBackup
  - Use the tools of the corresponding version. Otherwise, the restoration will fail.

  Table 1-54 Version mapping

Database	qpress	Percona XtraBackup
MySQL 8.0	qpress 7	XtraBackup 8.0.0 or later
MySQL 5.7 and MySQL 5.6	qpress 7	XtraBackup 2.4.9 or later

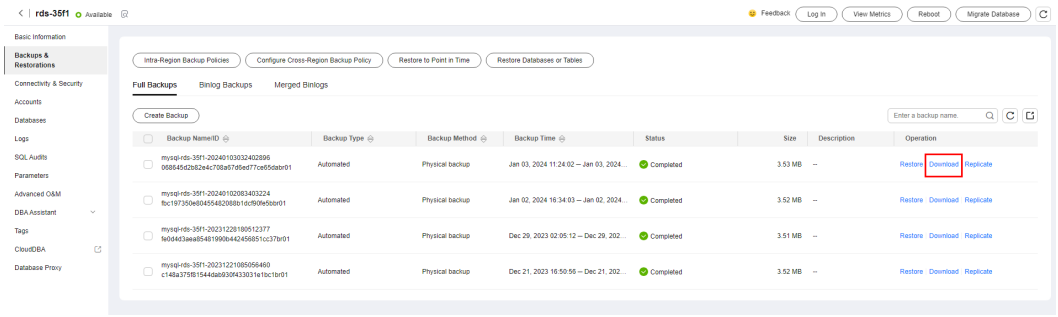
- During the restoration, do not run other workloads on the on-premises database.

Step 1: Download a Full Backup of Your RDS for MySQL DB Instance

RDS for MySQL DB instances automatically perform full backups at the time you specified. You can also create manual backups for your DB instance. The generated **.qp** files can be downloaded and restored to an on-premises database.

1. Click the DB instance name on the RDS console, choose **Backups & Restorations** > **Full Backups**, locate the target full backup, and click **Download** in the **Operation** column to download the backup.

Figure 1-109 Downloading a full backup



2. Use a file transfer tool (such as WinSCP) to upload the full backup file to the Linux device where the on-premises MySQL database is located.

Step 2: Install qpress and XtraBackup on the On-Premises MySQL Database

Method 1: Manual Installation

1. Download qpress and XtraBackup of a correct version. You can also download them according to **Table 1-55**. After the download is complete, upload the installation package to the Linux device where the on-premises MySQL database is located.
   - qpress: **https://repo.percona.com/yum/release/**
   - Percona XtraBackup:
     - For MySQL 5.6 and 5.7, download **XtraBackup 2.4.9** or later.
     - For MySQL 8.0, download **XtraBackup 8.0** or later.

Table 1-55 Download example

Tool	Example
MySQL 5.6	mysql-5.6.51-linux-glibc2.12-x86_64.tar.gz
MySQL 5.7	mysql-5.7.38-linux-glibc2.12-x86_64.tar.gz
MySQL 8.0	mysql-8.0.26-linux-glibc2.12-x86_64.tar
qpress	**qpress-11-1.el7.x86_64.rpm**
Percona XtraBackup	**XtraBackup 2.4.9** (MySQL 5.6 and 5.7)
	XtraBackup 8.0 (MySQL 8.0)

2. Install the qpress rpm package. Enterprise Linux 7 (including CentOS 7, RHEL 7, Rocky Linux 7, and AlmaLinux 7) is used as an example.

   rpm -ivh qpress-11-1.el7.x86_64.rpm

3. Decompress the XtraBackup package and change the name to **xtrabackup**.

   tar -zxvf percona-xtrabackup-2.4.9-Linux-x86_64.tar.gz

   mv percona-xtrabackup-2.4.9-Linux-x86_64 xtrabackup

4. Add **xtrabackup** to environment variables.

```
echo  "export  PATH=$PATH:/usr/local/xtrabackup/bin"  >> /etc/profile
mv xtrabackup/ /usr/local/
source  /etc/profile
```

Method 2: Installation Using wget

1. Install the qpress rpm package.

   wget https://repo.percona.com/yum/release/7/RPMS/x86_64/qpress-11-1.el7.x86_64.rpm

   rpm -ivh qpress-11-1.el7.x86_64.rpm

2. Install Percona XtraBackup.
   - For MySQL 5.6 and 5.7, Percona XtraBackup 2.4.9 is used as an example.

     wget https://downloads.percona.com/downloads/Percona-XtraBackup-2.4/Percona-XtraBackup-2.4.9/binary/redhat/7/x86_64/percona-xtrabackup-24-2.4.9-1.el7.x86_64.rpm

     rpm -ivh percona-xtrabackup-24-2.4.9-1.el7.x86_64.rpm --nodeps --force
   - For MySQL 8.0, Percona XtraBackup 8.0 is used as an example.

     wget https://downloads.percona.com/downloads/Percona-XtraBackup-8.0/Percona-XtraBackup-8.0.32-26/binary/redhat/7/x86_64/percona-xtrabackup-80-8.0.32-26.1.el7.x86_64.rpm

     rpm -ivh percona-xtrabackup-80-8.0.32-26.1.el7.x86_64.rpm --nodeps --force

Step 3: Restore the Backup to the On-Premises MySQL Database

1. Create a temporary directory **backupdir**.

   mkdir backupdir

2. Decompress the full backup.

   NOTE

   Before decompressing the full backup to the temporary directory **backupdir**, ensure that the temporary directory is empty to prevent restoration exceptions.
   - MySQL 5.6 and 5.7:

     xbstream -x -p 4 < ./*full_backup*.qp -C ./backupdir/

     innobackupex --parallel 4 --decompress ./backupdir
   - MySQL 8.0:

     xbstream -x -p 4 < ./*full_backup*.qp -C ./backupdir/

     xtrabackup --parallel 4 --decompress --target-dir=./backupdir

3. Delete the **.qp** file.

   find ./backupdir/ -name '*.qp' | xargs rm -f

4. Apply redo logs.
   - MySQL 5.6 and 5.7:

     innobackupex --apply-log ./backupdir
   - MySQL 8.0:

     xtrabackup --prepare --target-dir=./backupdir

5. Back up data.
   a. Stop the MySQL database service.
      service mysql stop
      For MySQL 5.7, run the following command:
      /bin/systemctl stop mysqld.service
   b. Back up the original database directory.
      mv /usr/local/mysql/data /usr/local/mysql/data_bak
      mkdir /usr/local/mysql/data
   c. Create a new database directory and change the permissions.
      chown mysql:mysql /usr/local/mysql/data
6. Restore data to the on-premises database and change the directory permissions.
   Before performing this step, clear the **data** directory of the on-premises database. For details, see **5.b**.
   – MySQL 5.6 and 5.7:
      innobackupex --defaults-file=/etc/my.cnf --copy-back ./backupdir
      chown -R mysql:mysql /usr/local/mysql/data
   – MySQL 8.0:
      xtrabackup --defaults-file=/etc/my.cnf --copy-back --target-dir=./backupdir
      chown -R mysql:mysql /usr/local/mysql/data

   NOTE
   - The relative path (**./backupdir**) in the commands can be replaced with an absolute path.
   - **--defaults-file** is followed by the location (for example, **/etc/my.cnf**) of the MySQL configuration file. You can specify the location based on the site requirements.
7. Start the MySQL database.
   service mysql start
   For MySQL 5.7, run the following command:
   /bin/systemctl start mysqld.service
8. Log in to the database and view the restoration result.
   show databases

1.11 Read Replicas

1.11.1 Introduction to Read Replicas

Introduction

Currently, RDS for MySQL supports read replicas and read/write splitting.

In read-intensive scenarios, a single DB instance may be unable to handle the read pressure and service performance may be affected. To offload read pressure on the

primary DB instance, you can create one or more read replicas in the same region as the primary instance. These read replicas can process a large number of read requests and increase application throughput.

Once read replicas are created, you can **enable read/write splitting**. Write requests are automatically routed to the primary DB instance and read requests are routed to read replicas by user-defined weights.

If read/write splitting is disabled, separately configure connection addresses for the primary DB instance and each read replica on your applications so that read requests are sent to read replicas and write requests are sent to the primary DB instance.

A read replica uses a single-node architecture (without a standby node). Changes to the primary DB instance are also automatically synchronized to all associated read replicas through the native MySQL replication function. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Single Read Replicas and HA Read Replicas

In read-intensive scenarios, read replicas help offload read pressure from the primary instance.

Data is replicated from the primary instance to read replicas asynchronously. Although there is a replication delay, the data on read replicas will eventually be consistent with that on the primary instance. You can use read replicas if you do not mind such a replication delay.

Read replicas include single read replicas and HA read replicas.

- Single read replicas: If you choose single read replicas, you are advised to buy more than one single read replica and enable database proxy. That way, if one read replica fails, the database proxy can route traffic to other read replicas or the primary DB instance. When you purchase a single read replica, select the same value for **Table Name** as the DB instance.
- HA read replicas: If the physical server where the primary read replica resides is faulty, the standby read replica automatically takes over workloads from the primary read replica. When you purchase an HA read replica, select the same value for **Table Name** as the DB instance.

Recommendations for using read replicas:

- Configure no more than two HA read replicas for a DB instance.
- If your DB instance is associated with more than two read replicas, enable database proxy for cost-effectiveness.

NOTE

If the replication between a read replica (single or HA) and the DB instance is abnormal, it takes a long time to rebuild and restore the read replica (depending on the data volume).

Billing Standards

Read replicas are billed on a yearly/monthly or pay-per-use basis.

Functions

- Read replica specifications can be different from primary DB instance specifications.

  NOTICE

  To prevent a read replica creation failure, long delay, and high load of the read replica, it is recommended that the specifications of the read replica be at least equal to those of the primary instance.

- Read replicas are billed on a yearly/monthly or pay-per-use basis. Yearly/monthly billing provides a larger discount than pay-per-use billing and is recommended for long-term users.
- 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.

Constraints

- A maximum of 10 read replicas can be created for a DB instance.
- You can purchase read replicas only for your created primary DB instance.
- You cannot stop a read replica without stopping the primary instance. If you stop a primary instance, read replicas (if there are any) will also be stopped.
- All databases and tables in the primary instance are synchronized to read replicas. Data of the primary instance, standby instance, and read replicas is consistent.
- Read replicas do not support automated backups or manual backups. Read replicas do not provide binlogs.
- 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 or deletion.
- Read replicas do not support database account creation. Create database accounts on the primary DB instance. For details, see **Creating a Database Account**.
- Read replicas cannot be recycled after they are deleted.

Creating and Managing a Read Replica

- **Creating a Single Read Replica**
- **Creating Read Replicas in Batches**
- **Managing a Read Replica**

1.11.2 Creating an HA Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

After an RDS instance is created, you can create HA read replicas for it as required.

Constraints

- To create HA read replicas, you must have the required permissions. You can **submit a service ticket** to apply for the permissions.
- If you are creating an HA read replica, you can select **Cloud SSD** for **Storage Type**.
  - Cloud SSD: General-purpose, dedicated, and Kunpeng general-enhanced instance classes are supported.
- If you want to change a single read replica to an HA read replica, the single read replica must use **Cloud SSD** as its storage type.
  - Cloud SSD: General-purpose, dedicated, and Kunpeng general-enhanced instance classes are supported.
- If you are a DeC user, you can also create HA read replicas or change read replicas to HA read replicas. When you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default.
- When you create an HA read replica, the SSL certificate, disk encryption, port, and subnet of the HA read replica are unconfigurable. They are kept the same as an existing HA read replica, or your primary instance if no HA read replica has been created.
- To change a non-HA read replica to an HA read replica, ensure that the SSL certificate, disk encryption, port, and subnet of the non-HA read replica are the same as an existing HA read replica, or your primary instance if no HA read replica has been created.
- Ensure that the subnet where your instance resides has at least three IP addresses to be assigned to HA read replicas.
- To ensure reliable operation of your workloads, do not modify parameters of HA read replicas.
- For an HA read replica, do not change its port or change it to a non-HA read replica.
- The instance class of a read replica must be no smaller than that of the primary DB instance. Otherwise, the read replica creation may fail or the replication delay may increase.
- Do not perform DDL operations when creating read replicas. Otherwise, creating read replicas may fail.

Creating an HA Read Replica

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Create Read Replica** in the **Operation** column.

Step 5 On the displayed page, configure required parameters and click **Next**.

Figure 1-110 Basic information

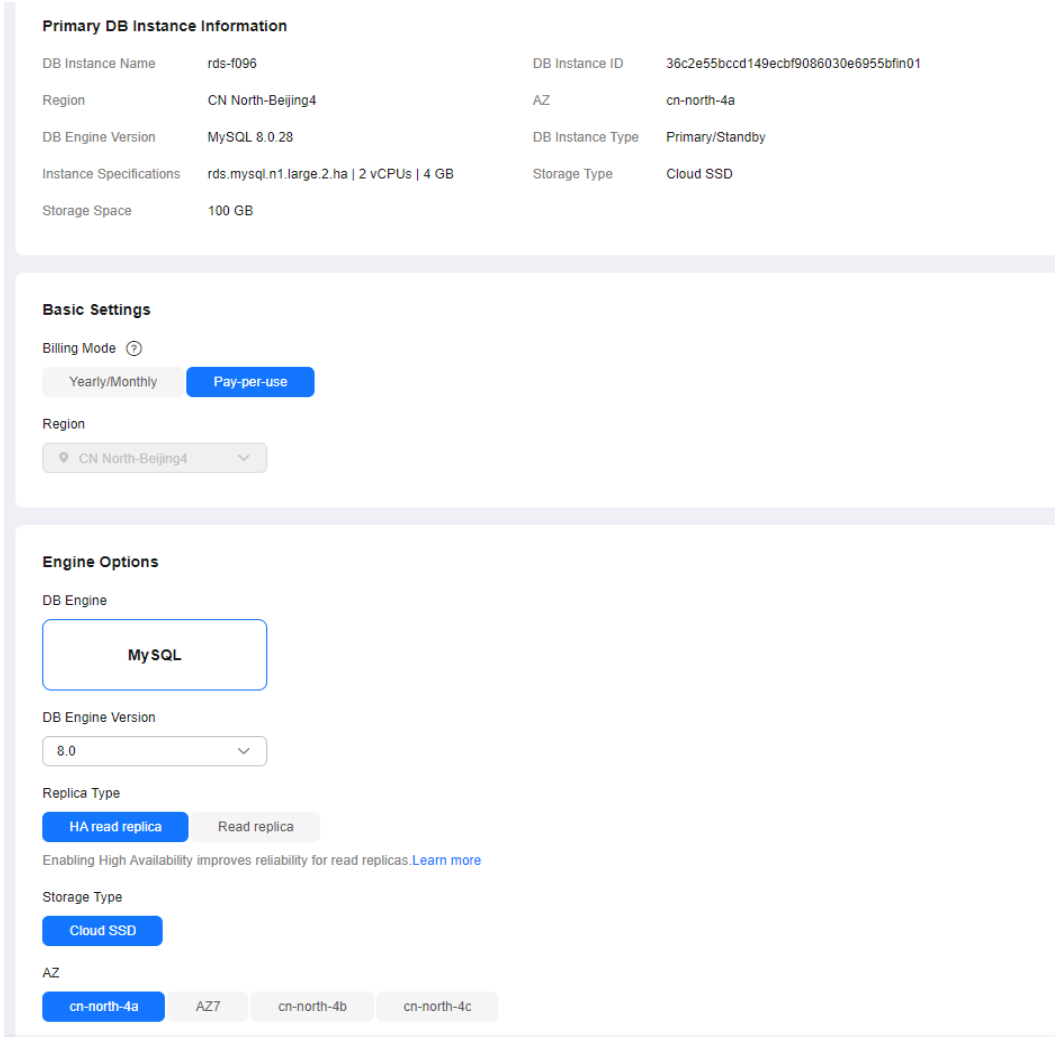


Table 1-56 Basic information

Parameter	Description
Billing Mode	Yearly/monthly billing and pay-per-use billing are supported.
Region	By default, read replicas are in the same region as your DB instance.

Parameter	Description
DB Engine	Same as the DB engine of your DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of your DB instance by default and cannot be changed.
Replica Type	Select **HA read replica**.
Storage Type	The storage type determines the read/write speed of an instance. The higher the maximum throughput is, the higher the read/write speed can be. • **Cloud SSD**: stores data in cloud SSDs to achieve decoupled compute and storage. • **Extreme SSD**: uses the 25GE network and RDMA technology to provide you with up to 1 million random read/write performance per disk and low latency per channel. **NOTE** If you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default.
AZ	RDS allows you to deploy your DB instance and read replicas in a single AZ or across AZs to improve reliability.

Figure 1-111 Specifications

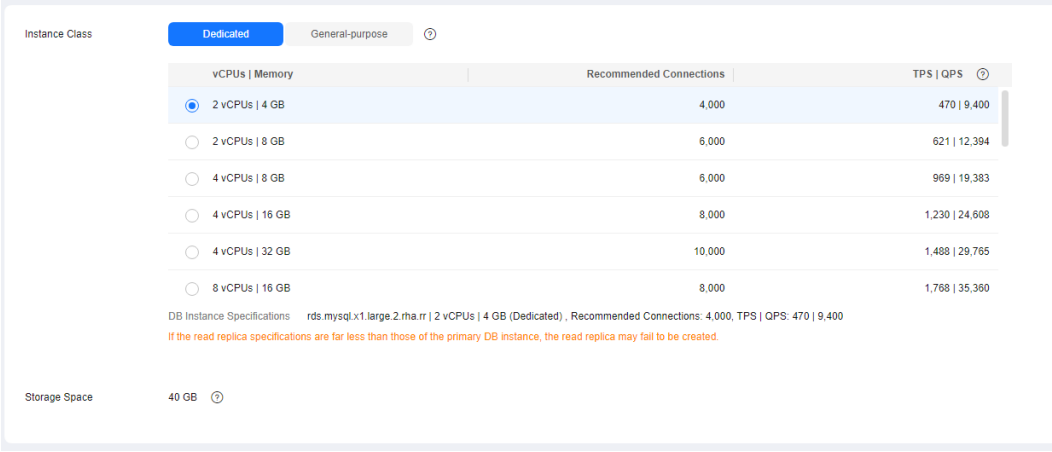


Table 1-57 Specifications

Parameter	Description
Instance Class	vCPUs and memory of an instance. Different instance classes support different numbers of database connections and maximum IOPS. After a DB instance is created, you can change its instance class. For details, see **Changing a DB Instance Class**. DB instances in a DCC only support the general-enhanced instance class.
Storage Space	Storage space contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary instance.

Figure 1-112 Connectivity and additional options

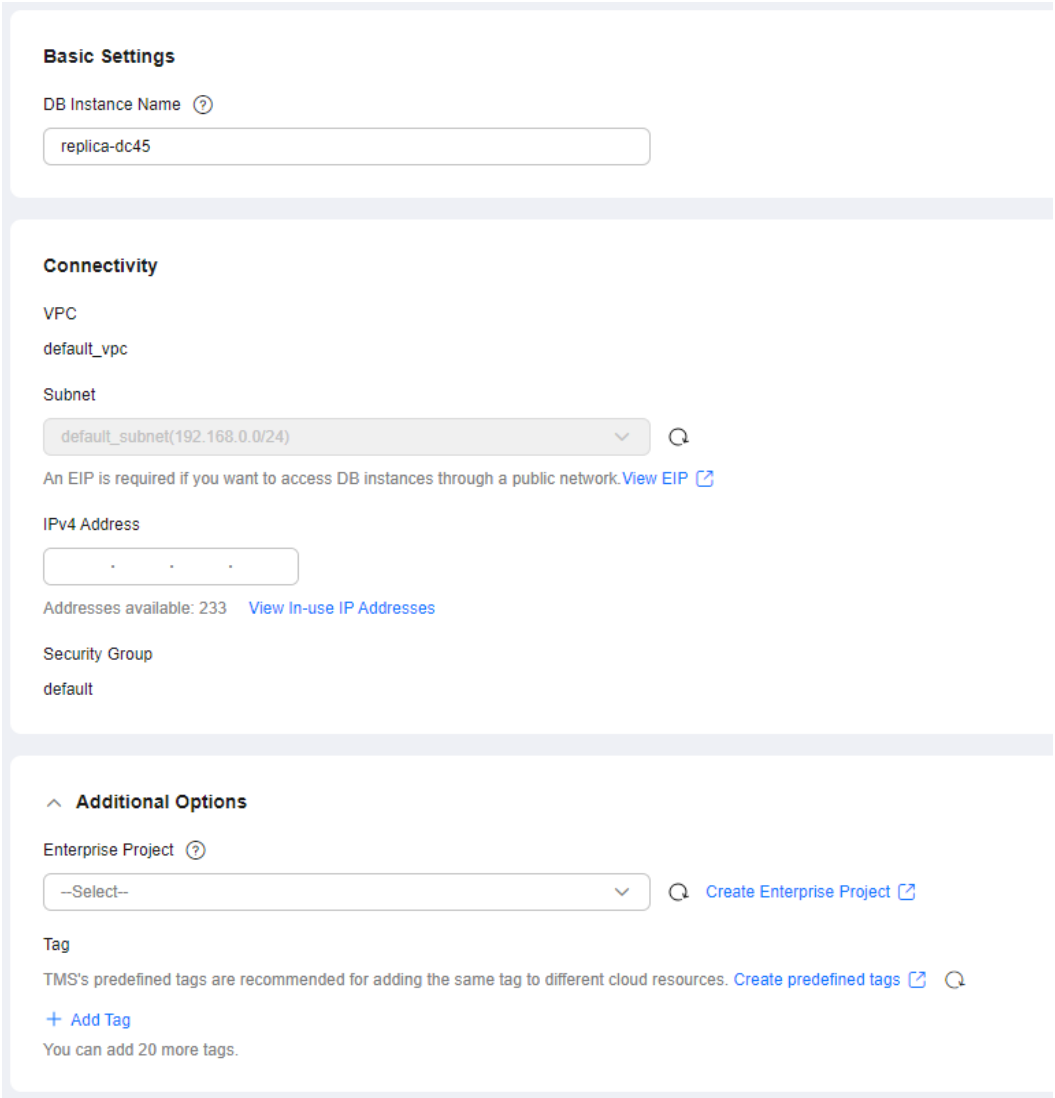


Table 1-58 Connectivity

Parameter	Description
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
VPC	Same as the primary instance's VPC.
Subnet	Same as the primary instance's subnet. • IPv4 address: A floating IPv4 address is automatically assigned when you create a read replica. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the read replica is created, you can change the floating IP address. • IPv6 address: A read replica assigned a floating IPv6 address will be created only when the vCPUs and memory you selected support IPv6 addresses. A floating IPv6 address is automatically assigned during read replica creation and cannot be specified. After the read replica is created, this floating IP address cannot be changed.
Security Group	Same as the primary instance's security group.

Table 1-59 Additional options

Parameter	Description
Enterprise Project	If your account has been associated with an enterprise project, select the target project from the **Enterprise Project** drop-down list. For more information about enterprise projects, see ***Enterprise Management User Guide***.
Tag	Optional. Using tags can help you easily identify and categorize your read replicas. A maximum of 20 tags can be added for each read replica. After a read replica is created, you can view its tag details on the **Tags** page. For details, see **Managing Tags**.

Table 1-60 Yearly/monthly read replicas

Parameter	Description
Required Duration	The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.

Parameter	Description
Auto-renew	• By default, this option is not selected. • If you select this option, the renew cycle is the same as the selected duration.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for pay-per-use read replicas.
- For yearly/monthly read replicas, click **Pay Now**.

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**.

To view the detailed progress and result of the creation, go to the **Task Center** page. For details, see **Task Center**.

----End

FAQ

Q: Does creating read replicas during peak hours increase the load on my primary instance when my primary instance's CPU usage is high?

A: Yes. When a read replica is created, it synchronizes data from the primary instance, which consumes I/O and CPU resources of the primary instance. To avoid this impact, you can create read replicas during off-peak hours.

Q: Does an HA read replica can be deployed across AZs?

A: Yes. Although you can select only one AZ when creating an HA read replica, it is deployed across AZs in fact.

1.11.3 Creating a Single Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

After an RDS instance is created, you can create single read replicas for it as required.

Constraints

- A maximum of 10 read replicas can be created for a DB instance.
- Ensure that the subnet where your instance resides has at least two IP addresses to be assigned to single read replicas.
- The instance class of a read replica must be no smaller than that of the primary DB instance. Otherwise, the read replica creation may fail or the replication delay may increase.

- For details about how to create read replicas in batches, see **Creating Read Replicas in Batches**.
- Do not perform DDL operations when creating read replicas. Otherwise, creating read replicas may fail.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Create Read Replica** in the **Operation** column.

Step 5 On the displayed page, configure required parameters and click **Next**.

Figure 1-113 Basic information

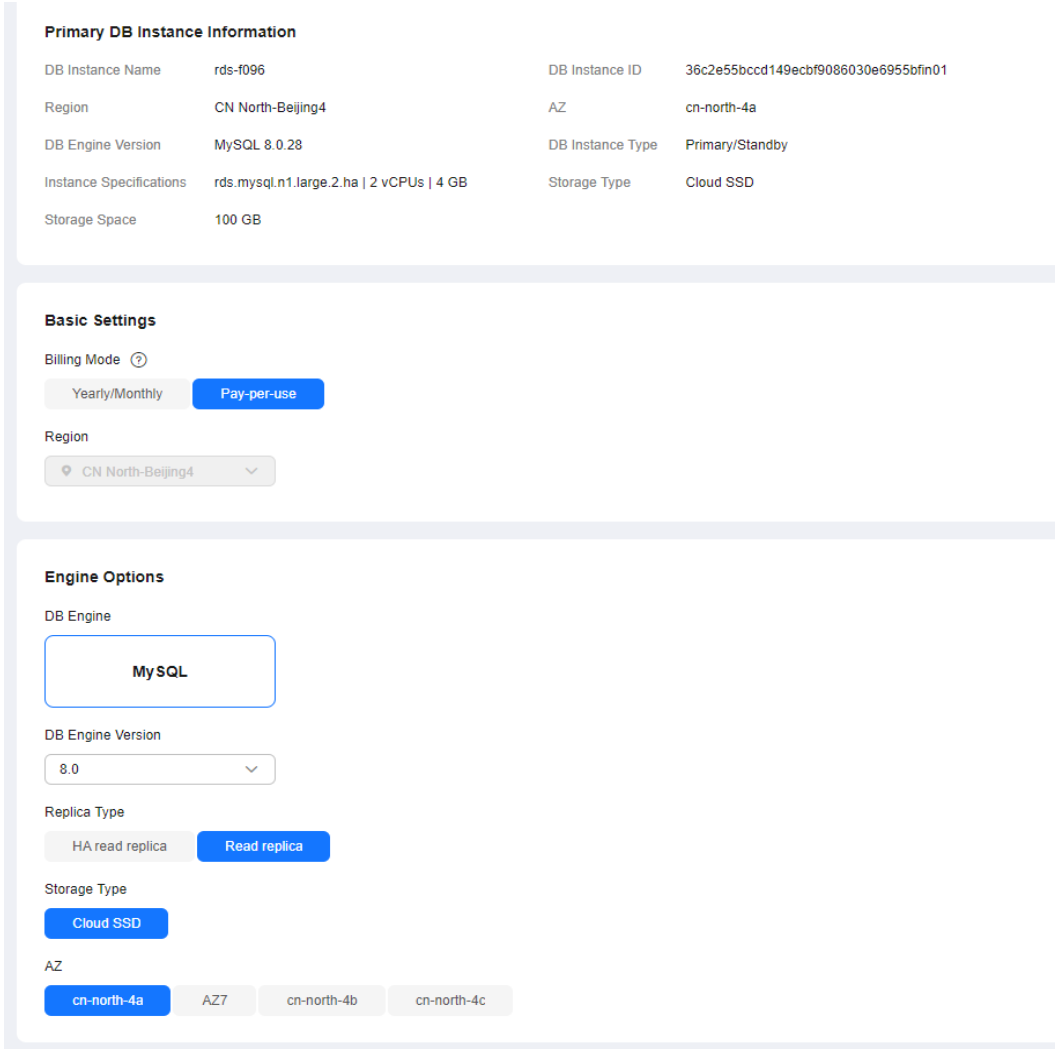


Table 1-61 Basic information

Parameter	Description
Billing Mode	Yearly/monthly billing and pay-per-use billing are supported.
Region	By default, read replicas are in the same region as your DB instance.
DB Engine	Same as the DB engine of your DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of your DB instance by default and cannot be changed.
Replica Type	Select **Read replica**.
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. • **Cloud SSD**: cloud disks used to decouple storage from compute. • **Extreme SSD**: uses the 25GE network and RDMA technology to provide you with up to 1 million random read/write performance per disk and low latency per channel. **NOTE** If you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default.
AZ	RDS allows you to deploy your DB instance and read replicas in a single AZ or across AZs to improve reliability.

Figure 1-114 Instance configuration

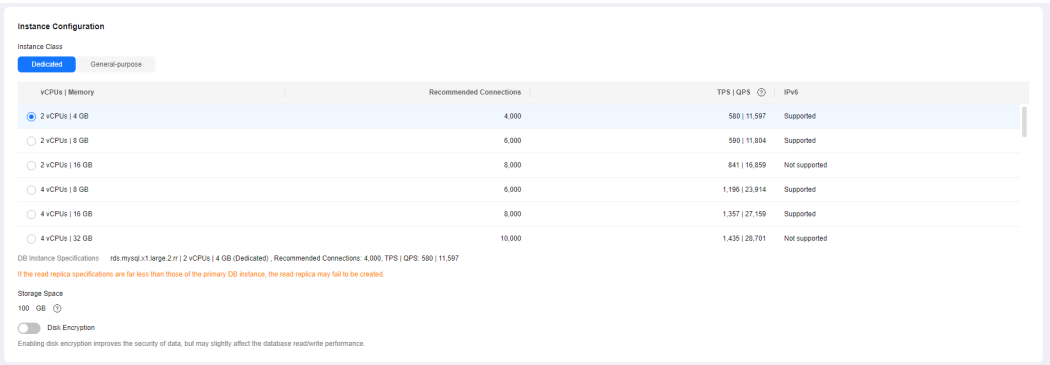


Table 1-62 Instance specifications

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. After a DB instance is created, you can change its instance class (CPU and memory). For details, see section **Changing a DB Instance Class**. DB instances in a DCC only support the general-enhanced instance class.
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.
Disk Encryption	• **Disable**: indicates the encryption function is disabled. • **Enable**: indicates the encryption function is enabled. Enabling disk encryption improves security but affects system performance. **Key Name**: indicates the tenant key. You can select an existing key or create a new one. **NOTE** – If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. – After an RDS DB instance is created, do not disable or delete the key that is currently in use. Otherwise, RDS will be unavailable and data cannot be restored. – For details about how to create a key, see **"Creating a CMK"** in *Data Encryption Workshop User Guide*.

Figure 1-115 Connectivity and additional options

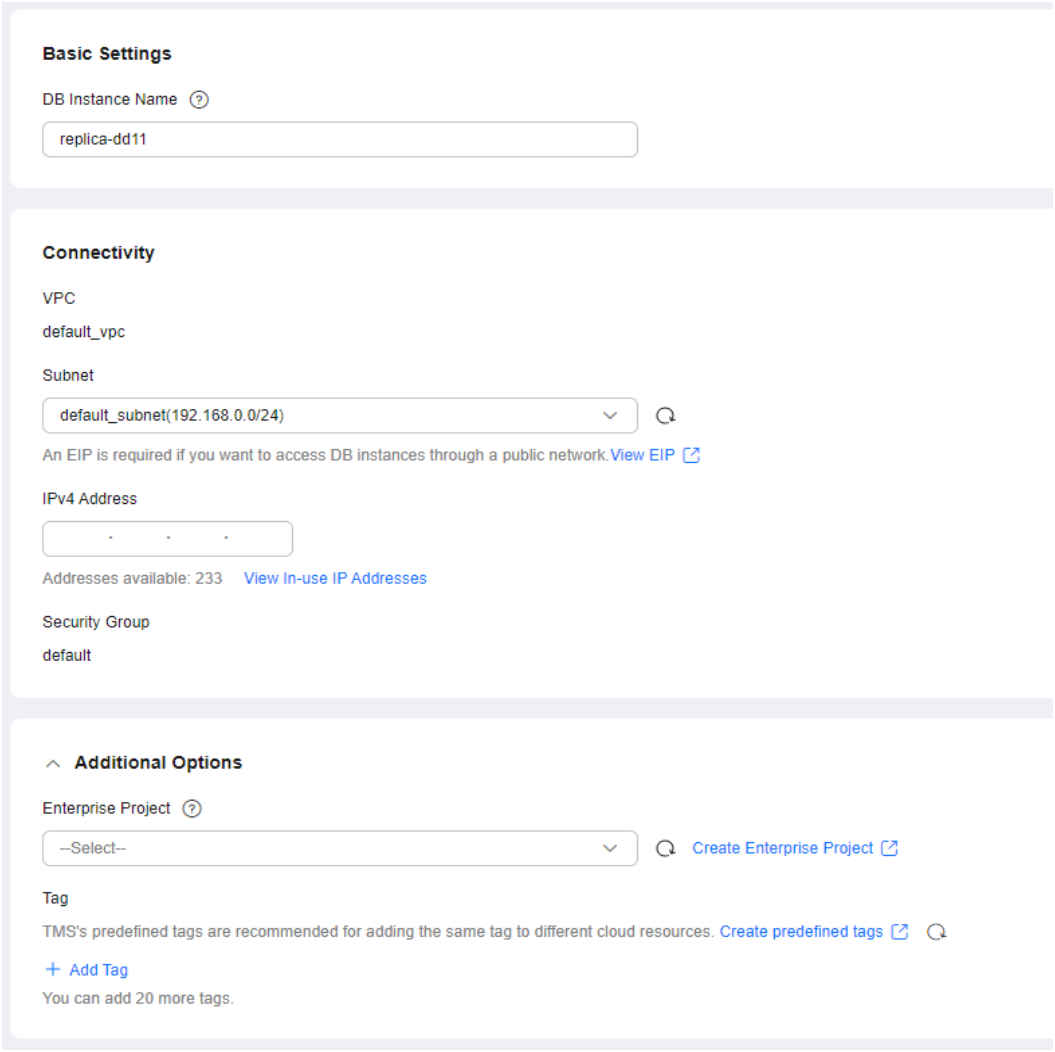


Table 1-63 Network

Parameter	Description
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
VPC	Same as the primary DB instance's VPC.

Parameter	Description
Subnet	Select the same subnet as the primary DB instance or another subnet in the same VPC. For details about how to create a subnet, see **Creating a Subnet for the VPC**. • IPv4 address: A floating IPv4 address is automatically assigned when you create a read replica. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the read replica is created, you can change the floating IP address. • IPv6 address: You can create a read replica using a floating IPv6 address only when the vCPU and memory you selected support IPv6 addresses. A floating IPv6 address is automatically assigned during read replica creation and cannot be specified. After the read replica is created, this floating IP address cannot be changed.
Security Group	Same as the primary DB instance's security group.

Table 1-64 Additional options

Parameter	Description
Enterprise Project	If your account has been associated with an enterprise project, select the target project from the **Enterprise Project** drop-down list. For more information about enterprise projects, see ***Enterprise Management User Guide***.
Tag	Optional. Tags help you easily identify and manage your read replicas. A maximum of 20 tags can be added for each read replica. After a read replica is created, you can view its tag details on the **Tags** page. For details, see **Managing Tags**.

Table 1-65 Yearly/monthly read replicas

Parameter	Description
Required Duration	The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	• By default, this option is not selected. • If you select this option, the auto-renew cycle is determined by the selected required duration.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for pay-per-use read replicas.
- For yearly/monthly read replicas, click **Pay Now**.

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. For details, see **Task Center**.

----End

FAQ

Q: Does creating read replicas during peak hours increase the load on my primary instance when my primary instance's CPU usage is high?

A: Yes. When a read replica is created, it synchronizes data from the primary instance, which consumes I/O and CPU resources of the primary instance. To avoid this impact, you can create read replicas during off-peak hours.

Follow-up Operations

Managing a Read Replica

1.11.4 Changing a Single-Node Read Replica to an HA Read Replica

Scenarios

You can change a single-node read replica to an HA read replica as needed. This operation does not affect your workloads.

A single-node read replica has only one physical node. If it fails, it cannot recover in a timely manner. However, an HA read replica has one primary node and one standby node. If the physical server where the primary read replica is located fails, the standby read replica automatically takes over workloads from the primary read replica. For more information, see **Introduction to Read Replicas**.

Constraints

- If you want to change a single read replica to an HA read replica, the single read replica must use **Cloud SSD** as its storage type.
  - Cloud SSD: General-purpose, dedicated, and Kunpeng general-enhanced instance classes are supported.
- If you are a DeC user, you can create HA read replicas or change single read replicas to HA read replicas. When you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default.

- To change a single read replica to an HA read replica, ensure that the SSL certificate, disk encryption, port, and subnet of the single read replica are the same as an existing HA read replica, or your primary instance if no HA read replica has been created.
- Ensure that the subnet where your instance is located has sufficient IP addresses (≥ 3) to be assigned to HA read replicas.
- To ensure reliable operation of your workloads, do not modify parameters of HA read replicas.
- For an HA read replica, do not change its port or change it to a single read replica.

Changing to an HA Read Replica

A single read replica can be changed to an HA read replica, but an HA read replica cannot be changed to a single read replica.

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

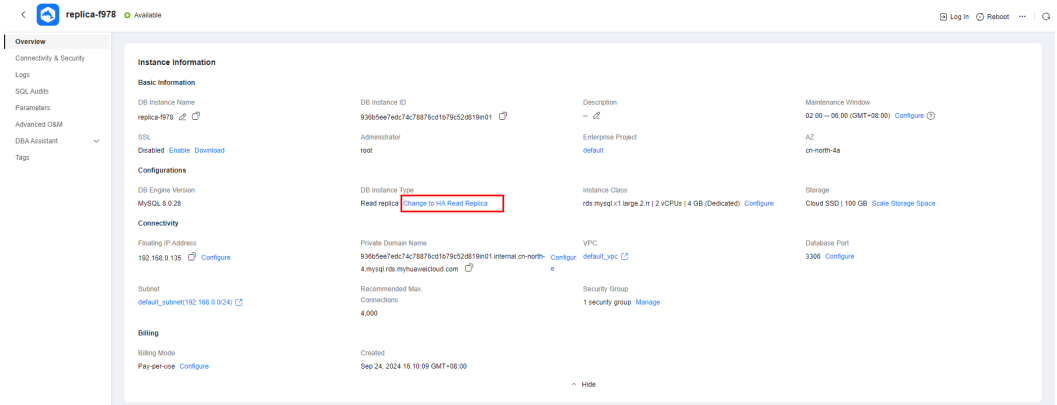
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the DB instance list, click in front of the DB instance and click the target read replica name.

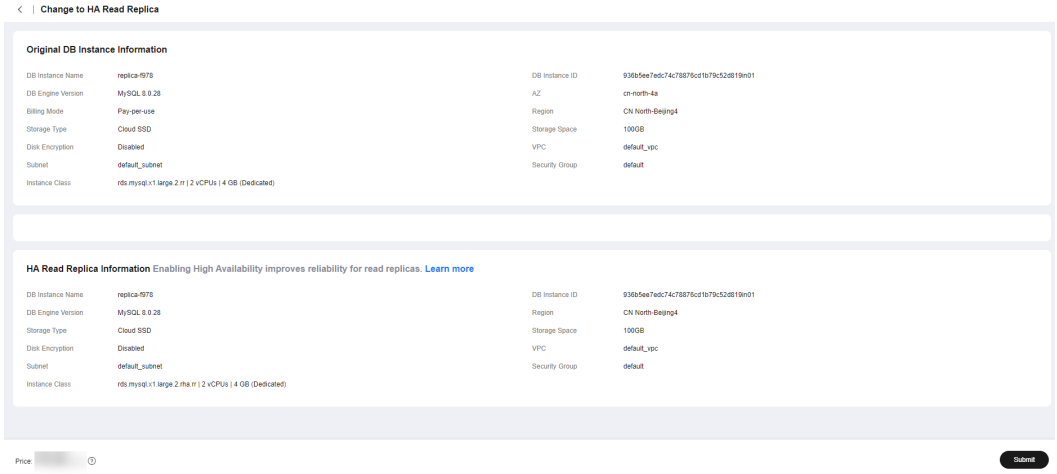
Step 5 On the **Overview** page, find **DB Instance Type** and click **Change to HA Read Replica**.

Figure 1-116 DB information



Step 6 On the displayed page, confirm the information and click **Submit**.

Figure 1-117 Changing to an HA read replica



----End

1.11.5 Creating Read Replicas in Batches

Scenarios

Read replicas are used to enhance read capabilities and reduce the load on primary DB instances. On the **Instances** page, you can select one or more MySQL DB instances and create read replicas for them in batches.

Constraints

- To create read replicas in batches, **submit a service ticket** to apply for required permissions.
- A maximum of 10 read replicas can be created for a DB instance.
- You can create read replicas for a maximum of 50 MySQL DB instances.
- You can create read replicas in batches only for DB instances of the same DB engine version and CPU architecture. For details about the supported instance classes and CPU architectures, see **RDS for MySQL Instance Classes**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select one or more DB instances and choose **More** > **Create Read Replica** above the instance list.

Step 5 On the displayed page, configure required information and click **Next**.

- 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 VPC and storage configurations are the same as those of the primary DB instance.
- Each account can create no more than 5 read replicas total for any given DB instance. In a batch creation, the number of read replicas you can create is limited by whichever DB instance already has the most replicas.

  For example, in a batch creation where most of the DB instances only have a single read replica, if any DB instance in the batch has more than one, for example, 3, you would only be able to add 2 more replicas for each DB instance in that particular batch operation.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for pay-per-use read replicas.
- For yearly/monthly read replicas, click **Pay Now**.

Step 7 After read replicas are created, you can view and manage them.

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. For details, see **Task Center**.

----End

Follow-up Operations

Managing a Read Replica

1.11.6 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

----End

Entering the Management Interface Through DAS

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click ⊞ in front of it. In the expanded panel, locate the read replica you want to manage and click **Log In** in the **Operation** column.

Step 5 On the displayed login page, enter the correct username and password and click **Log In**.

----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.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

1.12 Database Proxy (Read/Write Splitting)

1.12.1 Introduction to RDS for MySQL Database Proxy

Database Proxy is a network proxy service that sits between RDS for MySQL and applications. It is used to handle all requests from the applications to access RDS for MySQL instances.

Read/write splitting enables read and write requests to be automatically routed through a database proxy address. After creating an RDS for MySQL instance, you can **enable database proxy**. Through the proxy address, write requests are routed to the primary instance and read requests to read replicas based on the routing policy of the proxy, reducing the read pressure of the primary instance.

Function Description

Basic Concepts

- Proxy address

  After purchasing a database proxy, you can view the proxy address on the **Database Proxy** page. The database proxy sends write requests to the primary instance and read requests to read replicas through this address.
- Transaction splitting

Database proxies support transaction splitting. With this feature enabled, the read requests prior to write operations in a transaction are routed to read replicas, offloading read pressure from the primary instance.

For more information about transaction splitting, see **Configuring Transaction Splitting**.

- Connection pool

  Database proxies provide session-level connection pooling. It helps reduce the database load caused by frequently setting up short connections.

  For more information about connection pools, see **Configuring Connection Pools**.

- Routing policy

  RDS for MySQL database proxies support weighted and load balancing routing policies.

  – Weighted: Read requests are routed based on the read weights you specify.
  – Load balancing: Read requests are routed to database nodes with fewer active connections. With this policy enabled, you do not need to configure the weights of nodes.

  For more information about routing policies, see **Configuring the Delay Threshold and Routing Policy**.

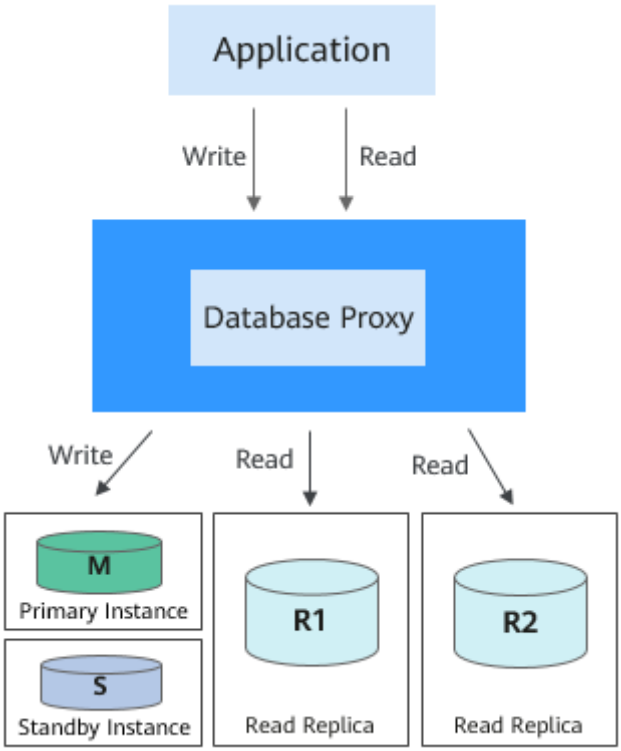
How Read/Write Splitting Works

Read/write splitting uses database proxies to split read and write requests. You can create one or more database proxies for your DB instance.

- Single database proxy

  If your RDS for MySQL instance has only one database proxy, applications connect to the database proxy through the proxy address. Write requests are forwarded to the primary instance and read requests to the primary instance or read replicas based on the **routing policy** you specify.

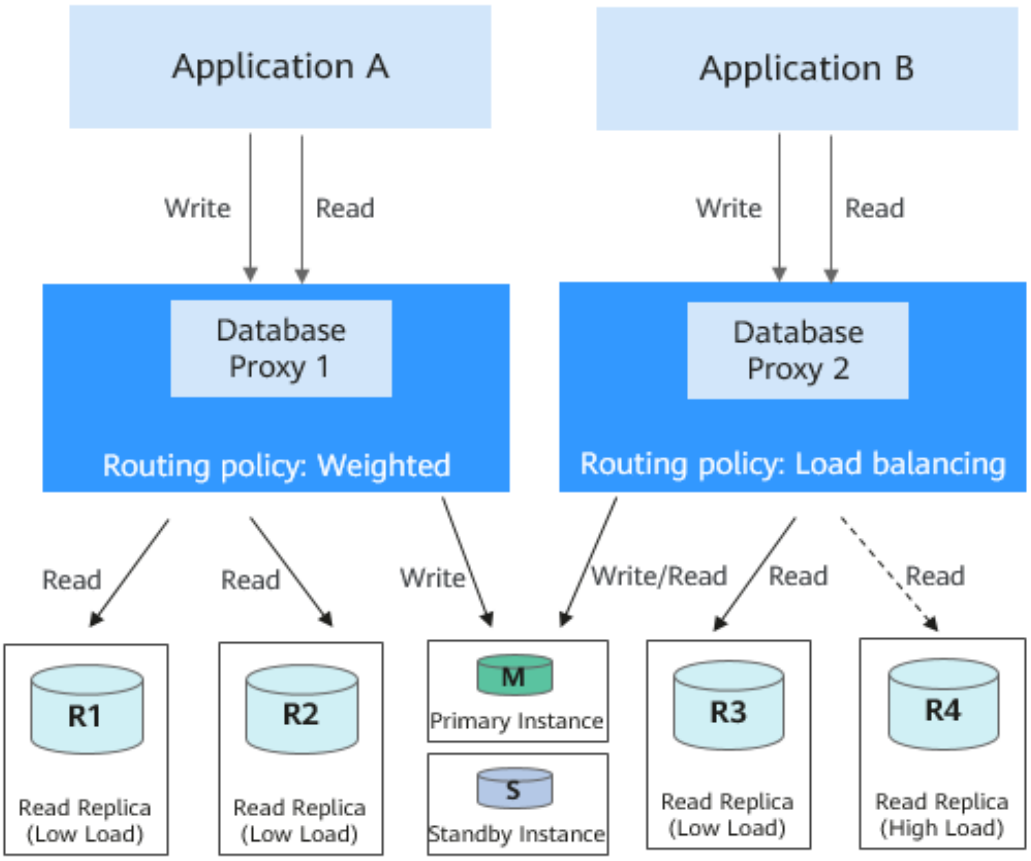
Figure 1-118 Read/write splitting with only one database proxy



- Multiple database proxies

  To isolate workloads from one another, you can create up to four database proxies for an RDS for MySQL instance. Different applications can connect to different database proxies as required. The database proxies connect to specified read replicas and forward read requests from different applications to different read replicas for workload isolation.

Figure 1-119 Read/write splitting with multiple database proxies



Scenario

- Read/write splitting enables read and write requests to be automatically routed. You can easily scale out a database proxy as required at low maintenance costs.
- Read requests are distributed to your read replicas based on weights to balance your database traffic and improve resource utilization.
- A proxy routes read requests of your application only to the read replicas you specify for the proxy.

Advantages of Read/Write Splitting

- Read/write splitting enables read and write requests to be automatically routed. You can easily scale out a database proxy as required at low maintenance costs.
- Read requests are distributed to your read replicas based on weights to balance your database traffic and improve resource utilization.
- A proxy routes read requests of your application only to the read replicas you specify for the proxy.
- By default, database proxies provide overload protection to prevent operations with large result sets from causing out of memory (OOM) exceptions of the server. If the database kernel pressure is high, database traffic throttling is required.

Request Routing Rules

- **The following requests will be routed only to the primary instance.**
  - INSERT, UPDATE, DELETE, and SELECT FOR UPDATE
  - All DDL operations (such as table/database creation, table/database deletion, table structure change, and permission change)
  - All requests in transactions (But if transaction splitting is enabled, some read requests in transactions may be sent to read replicas. For details, see **Configuring Transaction Splitting**.)
  - User-defined functions
  - Stored procedures
  - Multi-statement requests
  - Requests that use temporary tables
  - SELECT last_insert_id()
  - All queries of and changes to user variables
- **The following requests will be routed either to the primary instance or a read replica.**
  - SELECT not in a transaction
  - COM_STMT_EXECUTE
- **The following requests will always be routed to all database nodes.**
  - Changes to all system variables
  - The USE command

Read/Write Role Processing Logic for Database Proxy

Role	Routing Policy	Weight of Primary Instance	Normal Case	All Read Replicas Are Faulty
Read only	Weighted Load balancing	Not configurable	Primary instance: does not process read-only requests. Proxy address: readable but not writable	Primary instance: does not process read-only requests. Proxy address: connection error
Read and write	Load balancing	Assigned by system	Primary instance: readable and writable Proxy address: readable and writable	Primary instance: readable and writable Proxy address: readable and writable

Role	Routing Policy	Weight of Primary Instance	Normal Case	All Read Replicas Are Faulty
	Weighted	> 0	Primary instance: readable and writable Proxy address: readable and writable	Primary instance: readable and writable Proxy address: readable and writable
		= 0	Primary instance: not readable but writable Proxy address: readable and writable	Primary instance: readable and writable Proxy address: readable and writable

Billing

Database proxy can be enabled only for purchased DB instances. After it is enabled, it is separately billed on a pay-per-use basis.

The database proxy service is available for commercial use. It is billed by node. When you purchase a database proxy instance on the console, two nodes are created by default. The total fee is calculated as follows: Total fee = Number of nodes x Unit price. For details about the unit price, see the price of database proxy in **RDS Pricing Details**.

1.12.2 Constraints on Database Proxy

Supported Regions

Database proxy is available in the CN-Hong Kong, AP-Bangkok, and AP-Singapore regions. If your instance is deployed in any other region, **submit a service ticket** to request the required permissions.

To purchase multiple database proxies, you need to **submit a service ticket** to request the required permissions.

Supported Versions

- MySQL 5.6: 5.6.51.7 or later
- MySQL 5.7: 5.7.37.2 or later
- MySQL 8.0: all minor versions

If your kernel version is not supported, **upgrade the minor version**.

Function Constraints

- One RDS for MySQL read replica can connect to different proxies. However, you are advised to configure only one proxy that can allocate requests to the read replica.
- rdsProxy is an internal database proxy account for RDS. To ensure proper read/write splitting, you are advised not to create an account with the same name as rdsProxy.
- If read/write splitting is enabled and you delete a primary RDS for MySQL instance, its read replicas are also deleted and read/write splitting is disabled.
- Read/write splitting does not support the caching_sha2_password identity authentication plugin for RDS for MySQL 8.0.
- After read/write splitting is enabled, the database ports and floating IP addresses of both the primary instance and read replicas cannot be changed.
- Read/write splitting does not support compression protocols.
- Read/write splitting does not support the isolation level READ UNCOMMITTED.
- If multi-statements are executed, all subsequent requests will be routed to the primary instance by default. To restore read/write splitting, disconnect the connection between your application and the read/write splitting address and establish a connection again. Multiple multi-statement processing modes are supported. For details, see **Configuring Multi-Statement Processing Modes**.
- If operations related to temporary tables are performed, all subsequent requests of the current connection will be routed to the primary instance by default. To restore read/write splitting, disconnect the connection and reestablish a connection.
- If **the HANDLER statement** is executed, all subsequent requests will be routed to the primary instance by default. To restore read/write splitting, disconnect the connection and reestablish a connection.
- When the read/write splitting address is used, all transaction requests are routed to the primary instance (you can use the transaction splitting feature to route read requests prior to write operations in a transaction to read replicas). The non-transaction read consistency is not ensured. To ensure read consistency, encapsulate the read requests into a transaction.
- When the read/write splitting address is used, the LAST_INSERT_ID() function can be used only in transactions.
- When user-defined variables are used, statements containing user-defined variables are routed to the primary instance.
- When a database proxy is used, the size of a concatenate SQL statement cannot exceed 100 MB to prevent statement parsing from consuming too many resources.
- When a .NET client is used to connect to database proxies, the MySQL.Data driver version of the client must be 8.0.19 or later because earlier driver versions may be incompatible with database proxies.
- To use transaction splitting, you need to upgrade the database proxy to the latest version.
- Database proxies do not support the SQL mode parameter **PAD_CHAR_TO_FULL_LENGTH**.

Syntax Constraints

Read/write splitting routes frontend requests to backend instance nodes by the configured weights.

Therefore, some SQL statements may have different results when being executed multiple times.

- If you connect to a DB instance through a proxy and run **show processlist**, the result returned displays only the running threads on the proxy node where **show processlist** is executed, so it is different from that returned when you directly connect to the DB instance.
- If a proxy node is abnormal, and you connect to the proxy through the read/write splitting address and run **show processlist** or **kill**, the command execution may be prolonged or freezes, but your services are not affected.
- If you run **show processlist** on a proxy and this proxy has a node deleted, the running threads on this node may be returned.
- If you run **kill** on a proxy, errors such as timeout may be reported. In this case, you can run **show processlist** again to check whether the thread is killed successfully.
- Requests that are routed by a database proxy can be killed only by running **kill** on the proxy.
- When the read/write splitting address is used, the **show errors** and **show warnings** commands are not supported.
- When the read/write splitting address is used, if stored procedures and functions depend on user variables (@variable), the execution result may be incorrect.

1.12.3 Using RDS for MySQL Database Proxies for Read/Write Splitting

You can enable database proxy for your RDS for MySQL instance to automatically forward read and write requests through a proxy address. To reduce read pressure of the primary instance, write requests are forwarded to the primary instance and read requests to read replicas based on the routing policy of the database proxy.

This section describes how to use a database proxy to implement read/write splitting. The process is as follows:

- **Step 1: Enable Database Proxy**
- **Step 2: Grant Access Permissions**
- **Step 3: Check Security Group Rules**
- **Step 4: Use a Proxy Address to Connect to an RDS for MySQL Instance**
- **Step 5: Verify Read/Write Splitting**

Precautions

Keep in mind the following notes on database proxies:

- Before enabling database proxy, ensure that you have purchased an RDS for MySQL instance.

- Both the primary instance and read replicas must be available.
- You have learned the regions and versions that support database proxies. For details, see **Constraints on Database Proxy**.

Step 1: Enable Database Proxy

Buying a Single Database Proxy

In the CN-Hong Kong, AP-Bangkok, and AP-Singapore regions, single database proxy is enabled by default. That means only one database proxy can be purchased for an RDS for MySQL instance. With low flexibility, single-proxy is suitable for a single workload environment where workload isolation is not required. A single proxy provides only one entry for accessing the database.

- For more information, see **Introduction to RDS for MySQL Database Proxy**.
- For details about region restrictions, see **Constraints on Database Proxy**.

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Alternatively, on the **Overview** page, click **Apply** under **Read/Write Splitting Address**.

Step 6 On the displayed page, click **Create Database Proxy**.

Step 7 Configure parameters and click **Next**.

Figure 1-120 Selecting the weighted routing policy

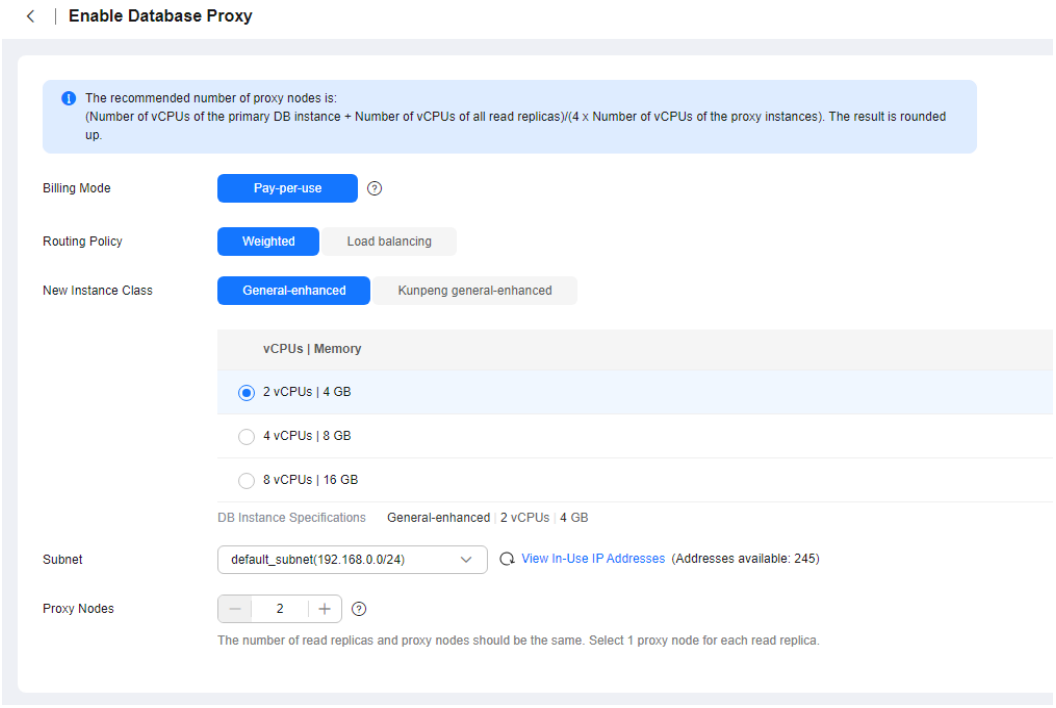


Figure 1-121 Selecting the load balancing policy

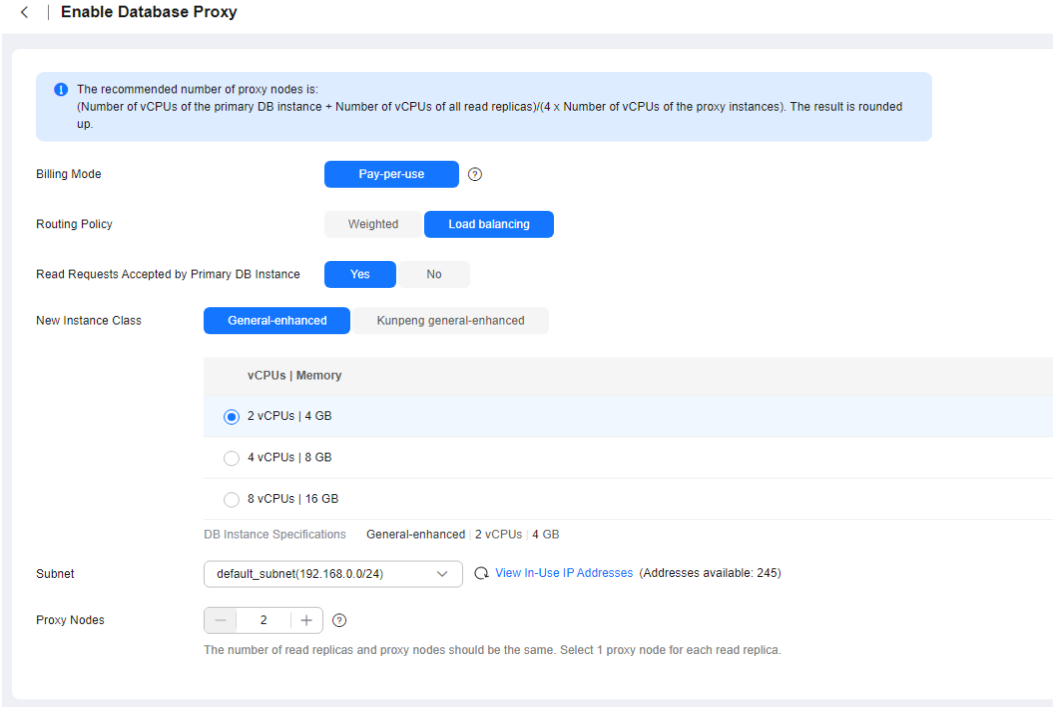


Table 1-66 Parameter description

Parameter	Description
Billing Mode	• Only **Pay-per-use** can be selected for pay-per-use DB instances. • Either **Pay-per-use** or **Yearly/monthly** can be selected for yearly/monthly DB instances. A pay-per-use proxy can be **changed to a yearly/monthly proxy** later. To create a yearly/monthly proxy, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
Routing Policy	• **Weighted**: You can change the weights of your DB instance and read replicas after read/write splitting is enabled. • **Load balancing**: This policy is only available if proxy load balancing is enabled. After **Load balancing** is selected, read requests are automatically distributed to multiple read replicas based on the number of active connections to balance the load among these read replicas. You can change the routing policy after the database proxy is created. For details, see **Configuring the Delay Threshold and Routing Policy**.
Read Requests Accepted by Primary DB Instance	This parameter is only available if **Load balancing** is selected. • **Yes**: Read requests can be routed to both the primary instance and read replicas. • **No**: Read requests are routed only to read replicas to offload read pressure from the primary instance.
New Instance Class	Select specifications for the proxy instance based on service demands. You can change the specifications after the proxy instance is created. For details, see **Changing the Instance Class of a DB Proxy Instance**. For details about performance metrics, see **Table 1-88**.
Subnet	Select the subnet where the proxy is located.
Proxy Nodes	Enter an integer from 2 to 8. You can change the nodes after the proxy instance is created. For details, see **Changing the Number of Proxy Nodes**. You are advised to set proxy nodes to the quantity of read replicas, with one proxy node for one read replica.

- Read/write splitting maintains database connectivity but splits read and write requests. If read/write splitting is enabled, an additional address called a read/write splitting address is provided. To use read/write splitting, switch your applications to this address.
- Read/write splitting address: You can connect to databases through the read/write splitting address, with read and write requests distributed to different databases automatically.

The read/write splitting address and the floating IP address of the DB instance are in the same VPC and subnet and are independent from each other.

- Delay threshold: You can set the delay threshold after read/write splitting is enabled. For details, see **Configuring Delay Threshold**.
- DB instances for load balancing: You can select DB instances for load balancing after read/write splitting is enabled.

Step 8 Confirm the database proxy configuration.

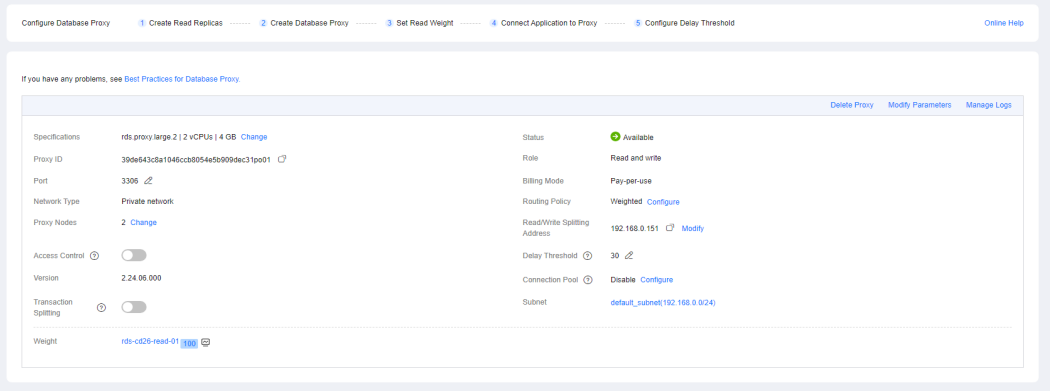
- To modify the configuration, click **Previous**.
- If there is no need to change the settings, click **Submit**.

Step 9 View and manage the proxy on the **Database Proxy** page.

You can view the read/write splitting address on the **Overview** page. Read and write requests can be split through the read/write splitting address.

The read/write splitting address and the floating IP address of the DB instance are in the same VPC and subnet and are independent from each other.

Figure 1-122 Viewing the read/write splitting address



----End

Buying Multiple Database Proxies

In the CN North-Beijing4, CN East-Shanghai1, CN South-Guangzhou, and CN Southwest-Guiyang1 regions, multi-proxy is enabled by default. That means multiple database proxies (up to 4) can be purchased for an RDS for MySQL instance. Multi-proxy is suitable for a complex multi-workload environment. Multiple database proxies provide different entries for accessing the database. This ensures workload isolation and improves resource utilization.

- For more information, see **Introduction to RDS for MySQL Database Proxy**.
- For details about region restrictions, see **Constraints on Database Proxy**.

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

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

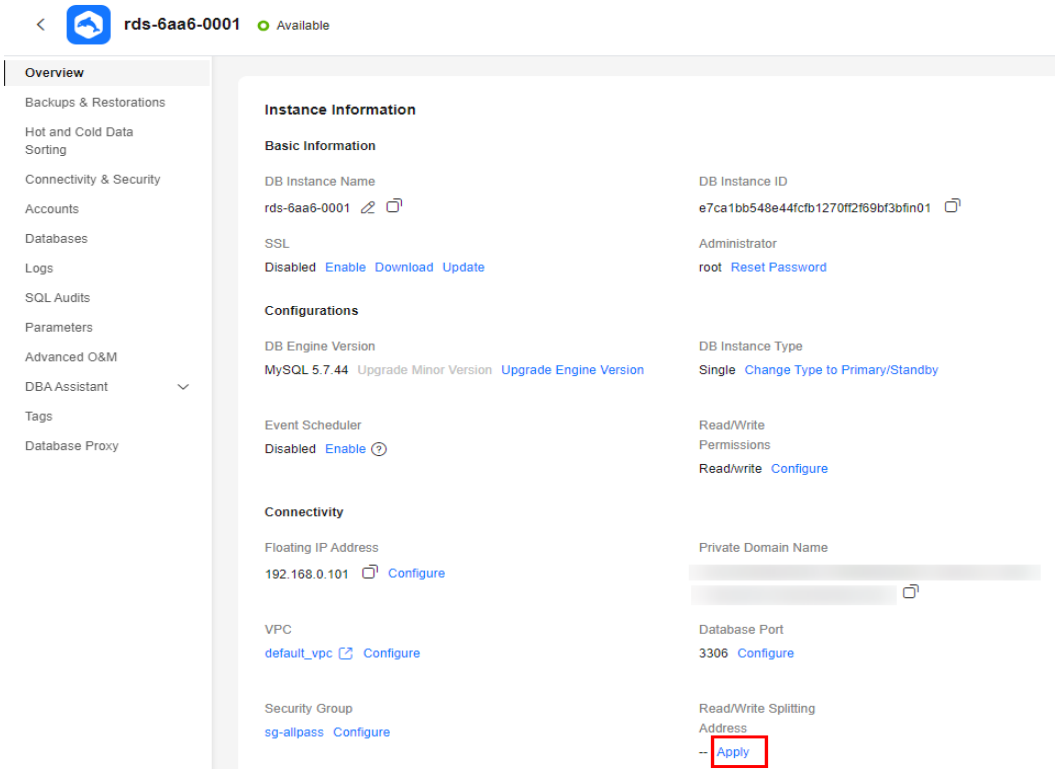
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

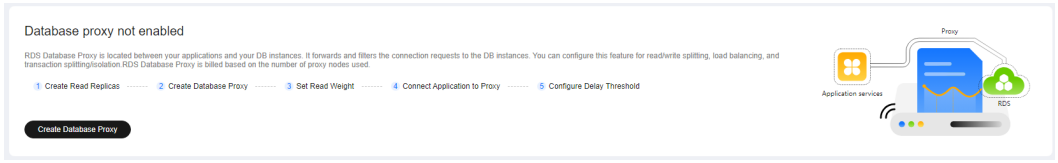
Alternatively, on the **Overview** page, click **Apply** under **Read/Write Splitting Address**.

Figure 1-123 Applying for a read/write splitting address



Step 6 On the displayed page, click **Create Database Proxy**.

Figure 1-124 Creating database proxy



Step 7 On the displayed page, set the required parameters and click **Next**.

Figure 1-125 Setting Routing Policy to Weighted

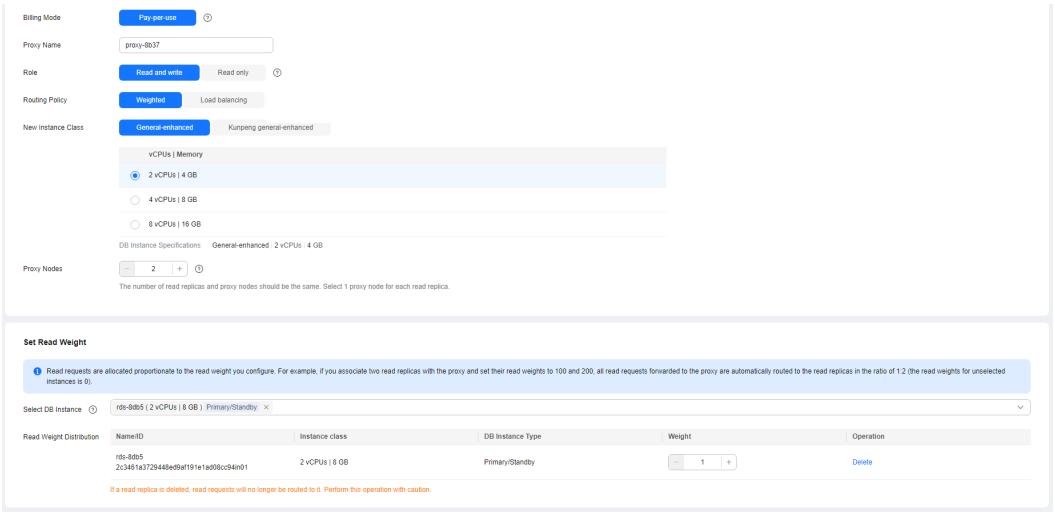


Figure 1-126 Setting Routing Policy to Load balancing

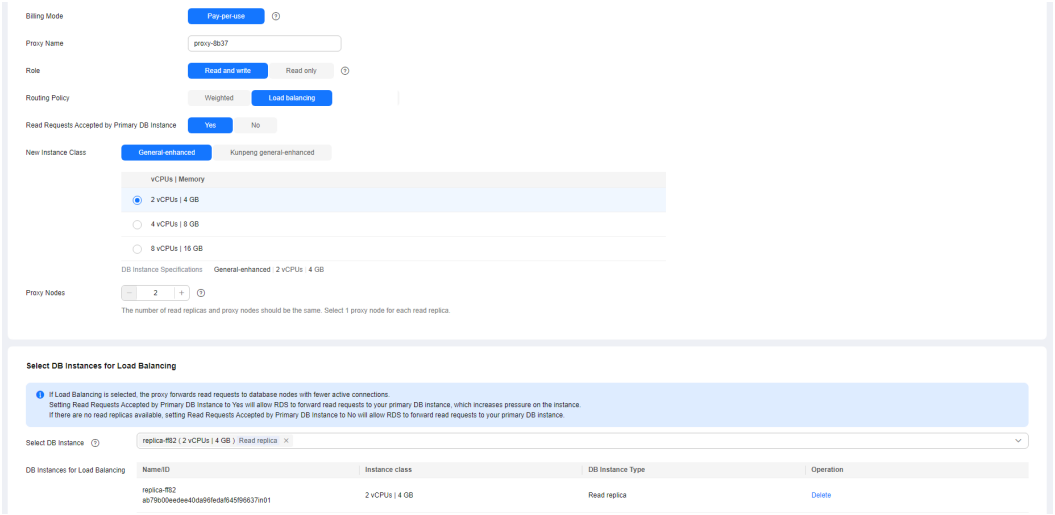


Table 1-67 Parameter description

Parameter	Description
Billing Mode	• Only **Pay-per-use** can be selected for pay-per-use DB instances. • Either **Pay-per-use** or **Yearly/monthly** can be selected for yearly/monthly DB instances. A pay-per-use proxy can be **changed to a yearly/monthly proxy** later. To create a yearly/monthly proxy, submit a request by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
Proxy Name	The proxy name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Role	• **Read and write**: Read and write requests are split. • **Read only**: The proxy is not connected to your primary instance and cannot receive write requests.
Routing Policy	• **Weighted**: You can change the weights of your DB instance and read replicas after read/write splitting is enabled. • **Load balancing**: If selected, to balance the load among read replicas, read requests are automatically distributed to multiple read replicas based on the number of active connections. You can change the routing policy after the database proxy is created. For details, see **Configuring the Delay Threshold and Routing Policy**.
Read Requests Accepted by Primary DB Instance	This parameter is available only when **Load balancing** is selected. • **Yes**: Read requests can be routed to both the primary instance and read replicas, which increases the load of the primary instance. Configure this parameter as required. • **No**: Read requests are routed only to read replicas to offload read pressure from the primary instance.
New Instance Class	Select specifications for the proxy instance based on service requirements. You can change the specifications after the proxy instance is created. For details, see **Changing the Instance Class of a DB Proxy Instance**. For details about performance metrics, see **Table 1-88**.
Proxy Nodes	Enter an integer from 2 to 8. You can change the nodes after the proxy instance is created. For details, see **Changing the Number of Proxy Nodes**. You are advised to set proxy nodes to the quantity of read replicas, with one proxy node for one read replica.

Parameter	Description
Set Read Weight	This parameter is only available if **Weighted** is selected. Select the primary instance and read replicas to which you want to assign weights. Rules for configuring read weights • Read requests are allocated proportionate to the read weight you configure. For example, if you associate two read replicas with the proxy and set their read weights to 100 and 200, all read requests forwarded to the proxy are automatically routed to the read replicas in the ratio of 1:2 (the read weights for unselected instances is 0). • A read replica can be associated with more than one proxy. To balance traffic among the read replicas of your primary instance, set read weights for them based on the existing proxies' weights and on the amount of traffic routed to the read replicas. • You can change read weights of the primary instance and read replicas after read/write splitting is enabled. For details, see **Configuring the Delay Threshold and Routing Policy**.
Select DB Instances for Load Balancing	This parameter is available only when **Load balancing** is selected. Select the DB instances for load balancing. After **Load balancing** is selected, the proxy forwards read requests to database nodes with fewer active connections. You can change the DB instances for load balancing after read/write splitting is enabled. For details, see **Configuring the Delay Threshold and Routing Policy**.

Step 8 Confirm the database proxy configuration.

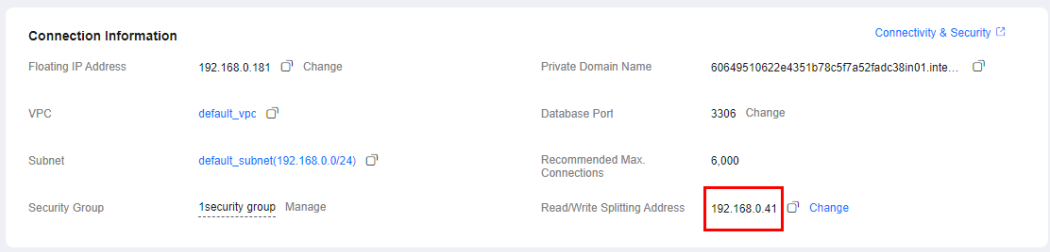
- To modify the configuration, click **Previous**.
- To submit the request, click **Submit**.

Step 9 View and manage the proxy on the **Database Proxy** page.

You can view the read/write splitting address on the **Overview** page. Read and write requests can be split through the read/write splitting address.

The read/write splitting address and the floating IP address of the DB instance are in the same VPC and subnet and are independent from each other.

Figure 1-127 Viewing the read/write splitting address



----End

Step 2: Grant Access Permissions

Before using a database proxy to connect to an RDS for MySQL instance, ensure that the current database account has the permission to access the proxy address.

You can perform the following steps to check and grant an account the permission to access a proxy address.

Step 1 Connect to your RDS for MySQL instance by referring to **Instance Connection**.

Step 2 Check whether **host** of your account contains a database proxy address.

```
SELECT user,host FROM mysql.user;
```

```
mysql> select user,host from mysql.user;
+---------------+-----------+
| user          | host      |
+---------------+-----------+
| app           | %         |
| rdsProxy      | %         |
| repl          | %         |
| root          | %         |
| test          | %         |
| testGTPUser   | %         |
| mysql.session | localhost |
| mysql.sys     | localhost |
| root          | localhost |
+---------------+-----------+
```

Step 3 If the host does not contain the CIDR block where the database proxy is located, grant the access permission to the account.

For example, if you want to connect to an RDS for MySQL instance from the IP address range starting with 192.168.0 as the **root** user, you can set **host** of the account to **192.168.%** on the user management page of Data Admin Service (DAS). For details, **Editing a User**.

Figure 1-128 Configuring the host IP address

----End

Step 3: Check Security Group Rules

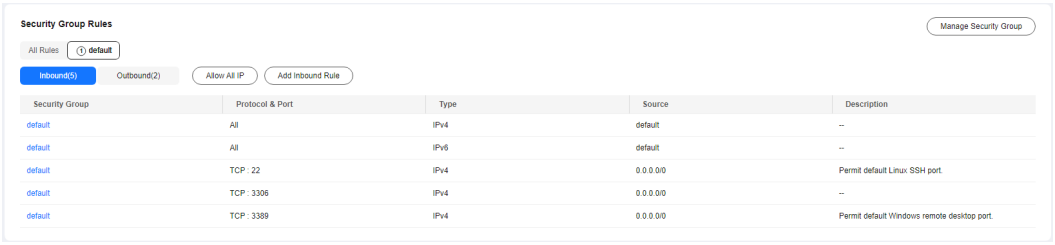
Ensure that there is an inbound rule that allows access from the proxy address. The default port is **3306**.

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **Connectivity & Security**. In the **Security Group Rules** area, click the security group name to view the security group rules.

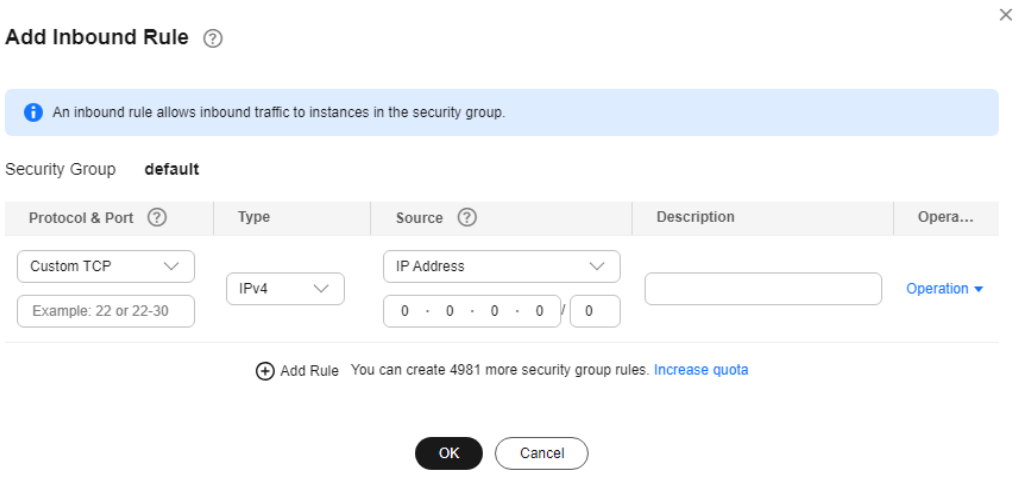
Step 3 On the **Inbound** tab, check whether access through port **3306** is allowed by default.

Figure 1-129 Allowing access through port 3306



If there is no such a rule, click **Add Inbound Rule** or **Allow All IP**.

Figure 1-130 Adding an inbound rule

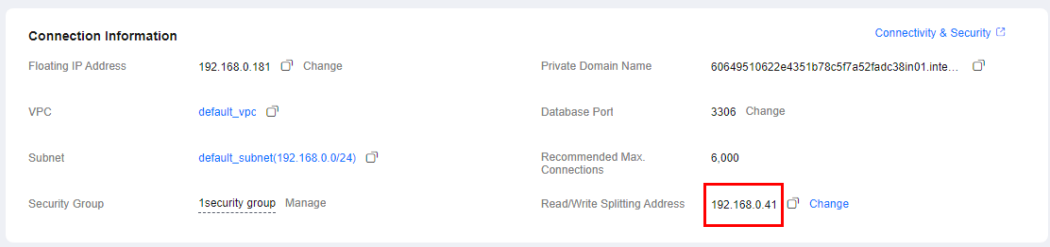


----End

Step 4: Use a Proxy Address to Connect to an RDS for MySQL Instance

Step 1 Check for the proxy address and port on the RDS for MySQL console.

Figure 1-131 Checking for the proxy address and port



Step 2 Log in to an ECS.

For details, see **Logging In to a Linux ECS**.

Step 3 Run the following command to connect to the RDS for MySQL instance through the proxy address:

mysql -h *<hostIP>* **-P** *<port>* **-u** *<userName>* **-p** *<password>*

Table 1-68 Parameter description

Parameter	Description
<hostIP>	The proxy address obtained in **Step 1**.
<port>	The database port obtained in **Step 1**.
<userName>	The username of the database administrator account. The default username is **root**.
<password>	The password of the database administrator account.

NOTE

When you use a MySQL 8.0 client to access a database proxy, the error message "auth user failed" may be displayed.

Add **--default-auth=mysql_native_password** when connecting to the database.

----End

Step 5: Verify Read/Write Splitting

You can run the **show last route** command to check the routing result after you perform a read operation.

The following uses a read operation as an example to describe how to check the routing result of read requests.

Step 1 After connecting to your RDS for MySQL instance, perform a read operation.

Example: **select 1;**


```
mysql> select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.08 sec)
```

Step 2 Run the following command to check the routing result of the read operation in **Step 1**:

show last route

Figure 1-132 Query result

```
mysql> select 1;
+---+
| 1 |
+---+
| 1 |
+---+
1 row in set (0.08 sec)

mysql> show last route;
+-----------------+
| LAST ROUTE      |
+-----------------+
| [redacted]      |
+-----------------+
1 row in set (0.05 sec)
```

NOTE

Do not include **show last route** in service code or multi-statement requests.

----End

Related APIs

- Enabling Database Proxy
- Querying Database Proxies
- Querying Database Proxy Specifications
- Configuring the Routing Policy for a Database Proxy
- Disabling Database Proxy

1.12.4 Database Proxy Configurations

1.12.4.1 Configuring Transaction Splitting

Scenarios

In most cases, an RDS for MySQL proxy instance sends all requests in transactions to the primary DB instance to ensure transaction correctness. However, in some frameworks, all requests are encapsulated into transactions that are not automatically committed using **set autocommit=0**. This causes heavy loads on the primary DB instance.

Function

Database proxies support transaction splitting. With this feature enabled, RDS can route the read requests prior to write operations in a transaction to read replicas, reducing the pressure of the primary DB instance.

Transaction splitting is disabled by default. If it is enabled under the default READ COMMITTED transaction isolation, RDS only starts a transaction for write operations when automatic commit is disabled. Before the transaction starts, read requests are routed to read replicas through load balancers.

Precautions

- Enabling transaction splitting affects global consistency of certain workloads. Before enabling this feature, evaluate its impact on your workloads.
- All proxies of your DB instance must be in the **Available** state.
- Before enabling transaction splitting, you need to upgrade the database proxy to the latest version because the transaction processing logic has been optimized in the latest version.
- After transaction splitting is enabled, read requests of the transactions committed using **BEGIN** cannot be routed to read replicas.
- After transaction splitting is enabled, read requests of transactions started using **SET AUTOCOMMIT = 0** cannot be routed to read replicas.

Configuring Transaction Splitting

NOTE

Transaction splitting takes effect only for connections established after this feature is enabled or disabled.

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 In the navigation pane on the left, click **Database Proxy**.

Step 6 On the displayed page, click next to **Transaction Splitting**.

Step 7 In the displayed dialog box, click **OK**.

----End

1.12.4.2 Configuring Connection Pools

Scenarios

A session-level connection pool is suitable for short connections. A session-level connection pool helps reduce the database load caused by frequent establishment of short connections.

Connection Pool is disabled by default. You can enable a session-level connection pool.

How a Session-Level Connection Pool Works

When your client disconnects from your database, RDS checks whether the connection is idle. If it is, RDS places the connection in the connection pool and retains the connection for a short period of time.

When your client re-initiates a connection, any available connection in the connection pool is used, reducing the overhead of establishing a new connection to the database. If no connections are available in the connection pool, a new connection will be established.

Constraints

- Only RDS for MySQL 8.0 and 5.7 support the connection pool function.
- All proxies of your DB instance must be in the **Available** state.
- This function is incompatible with Application Lossless and Transparent (ALT). If ALT is enabled, the connection pool becomes invalid.
- When any of the following operations is performed, the connection is locked until the connection ends. That is, the connection will not be placed in the connection pool for other users to use.
  - Running the **PREPARE** statement
  - Creating a temporary table
  - Modifying user variables
  - Inserting or querying big data (for example, more than 16 MB)
  - Running the LOCK TABLE statement
  - Executing a multi-statement query (concatenated SQL statements with semicolons, for example, SELECT 1;SELECT 2)
  - Calling a stored procedure

Procedure

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

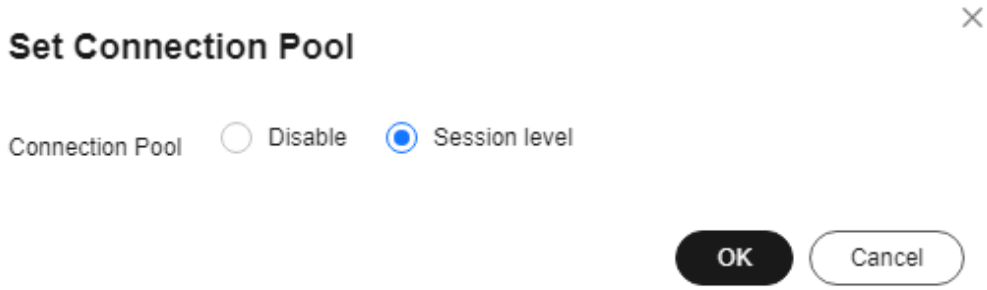
Step 4 On the **Instances** page, click the primary instance name.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Step 6 On the displayed page, click **Configure** next to **Connection Pool**.

Step 7 Set **Connection Pool** to **Session level** and click **OK**.

Figure 1-133 Configuring connection pool



----End

1.12.4.3 Modifying Read/Write Splitting Parameters

Scenarios

After read/write splitting is enabled, you can modify proxy parameters, for example, **multiStatementType**.

Constraints

- To modify read/write splitting parameters, you need to **submit a service ticket** to apply for required permissions.
- To change the value of **multiStatementType**, the proxy version must be 2.22.11.000 or later.
- All proxies of your DB instance must be in the **Available** state.

Procedure

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Step 6 On the displayed page, click **Modify Parameters** in the proxy information area.

Step 7 Change the value of **multiStatementType** and click **OK**.

multiStatementType is not available for read-only proxies.

- **Strict** (default): After a multi-statement request is sent to the primary instance, the read/write splitting of the current connection becomes unavailable and all subsequent requests are routed to the primary instance.
- **Loose**: After a multi-statement request is sent to the primary instance, the read/write splitting of the current connection still works.
- **Parse**: After a multi-statement request is sent to the primary instance, the request is parsed to determine whether to split subsequent read and write requests.

----End

1.12.4.4 Configuring the Delay Threshold and Routing Policy

After read/write splitting is enabled and read replicas are created, you can configure the delay threshold and routing policy as required.

Table 1-69 Read/write splitting parameters

Parameter	Description
Delay Threshold	The maximum delay for data to be synchronized from primary DB instances to read replicas. This parameter is only applied when there are read replicas. To prevent data inconsistencies between primary DB instances and read replicas from lasting too long, if the delay of a read replica exceeds the configured threshold, read requests are not forwarded to the read replica regardless of the read weight distributed to it. When read/write splitting is enabled, the default delay threshold is 30s and the default value range is 0–7,200s. It is recommended that the threshold be greater than or equal to 30s. Traffic is not allocated to read replicas whose delay exceeds the configured threshold.
Read Weight Distribution	After read/write splitting is enabled, you can configure read weights for the primary DB instance and read replicas. If no read replicas are selected for the database proxy, read/write splitting cannot be used. The read weight ranges from 1 to 1,000. Read replicas with higher read weight distributions process more read requests. For example, if the read weights distributed to one primary DB instance and four read replicas are 0, 100, 200, 500, and 300, respectively, the primary DB instance does not process any read requests (write requests are still automatically routed to the primary DB instance) while the four read replicas process read requests with a ratio of 1:2:5:3.

Constraints

All proxies of your DB instance must be in the **Available** state.

Configuring Delay Threshold

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 In the navigation pane on the left, click **Database Proxy**.

Step 6 In the proxy information area, click next to the **Delay Threshold** field.

----End

Configuring Routing Policy in Single-Proxy Mode

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

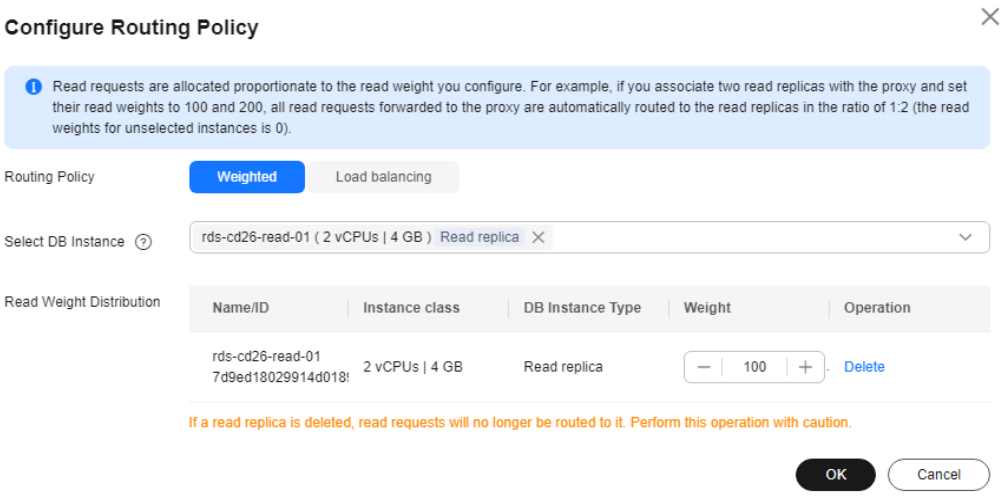
Step 5 In the navigation pane on the left, click **Database Proxy**.

Step 6 If proxy load balancing is not enabled, click **Configure** next to the **Routing Policy** field in the proxy information area. In the displayed dialog box, configure read weights for the primary instance and read replicas.

NOTE

- The system automatically distributes weights to read replicas, including read replicas created afterwards, according to the default distribution rules. If a read replica breaks down or is deleted, the weight is automatically removed. After the read replica recovers, the weight is automatically restored.
- If the weight of a node is set to **0**, read requests will not be routed to the node. If the weights of all nodes are set to **0**, read requests will be randomly routed to these nodes.
- Forcible routing is also supported. For details, see **Adding a Hint to Specify the Direction that a SQL Statement Will Be Routed**.

Figure 1-134 Setting read weight

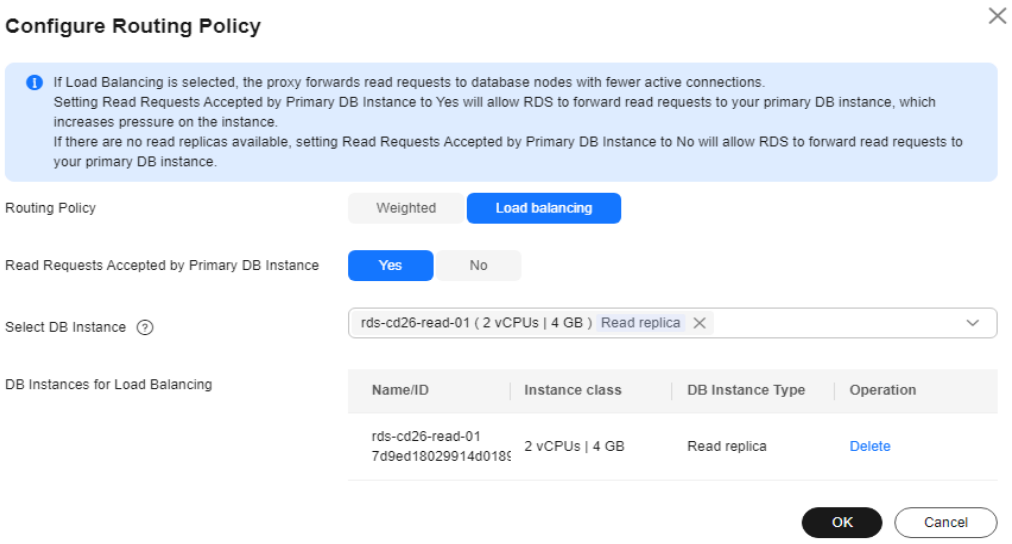


Step 7 Click **OK** and view the weights on the **Database Proxy** page.

Step 8 If proxy load balancing is enabled, click **Configure** next to the **Routing Policy** field in the proxy information area. In the displayed dialog box, set required parameters.

Select **Load balancing** as the routing policy. Read requests will be automatically distributed to read replicas based on the number of active connections to balance the load among these read replicas.

Figure 1-135 Routing policy - Load balancing



Read Requests Accepted by Primary DB Instance:

- **Yes**: Read requests can be routed to both the primary instance and read replicas, which increases the load of the primary instance. Configure this parameter as required.
- **No**: To offload read pressure from the primary instance, read requests are only routed to read replicas.

Step 9 Click **OK**. On the **Database Proxy** page, view the result. You can select the instances for load balancing as required.

----End

Configuring Routing Policy in Multi-Proxy Mode

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 In the navigation pane on the left, click **Database Proxy**.

Step 6 In the proxy information area, click **Configure** next to the **Routing Policy** field. In the displayed dialog box, set required parameters.

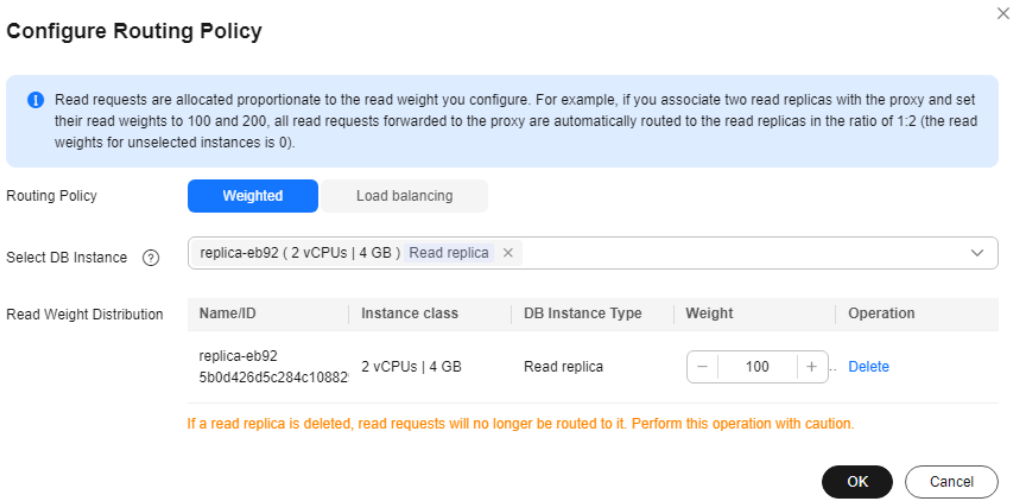
- **Weighted**: You can distribute read weights for the DB instance and read replicas. For details, see **Table 1-69**.

  NOTE

  The system automatically distributes weights to read replicas, including read replicas created afterwards, according to the default distribution rules. If a read replica breaks down or is deleted, the weight is automatically removed. After the read replica recovers, the weight is automatically restored.

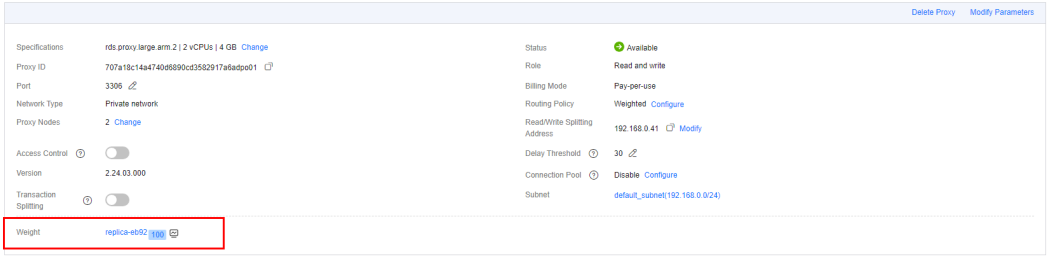
  Forcible routing is also supported. For details, see **Adding a Hint to Specify the Direction that a SQL Statement Will Be Routed**.

Figure 1-136 Weighted



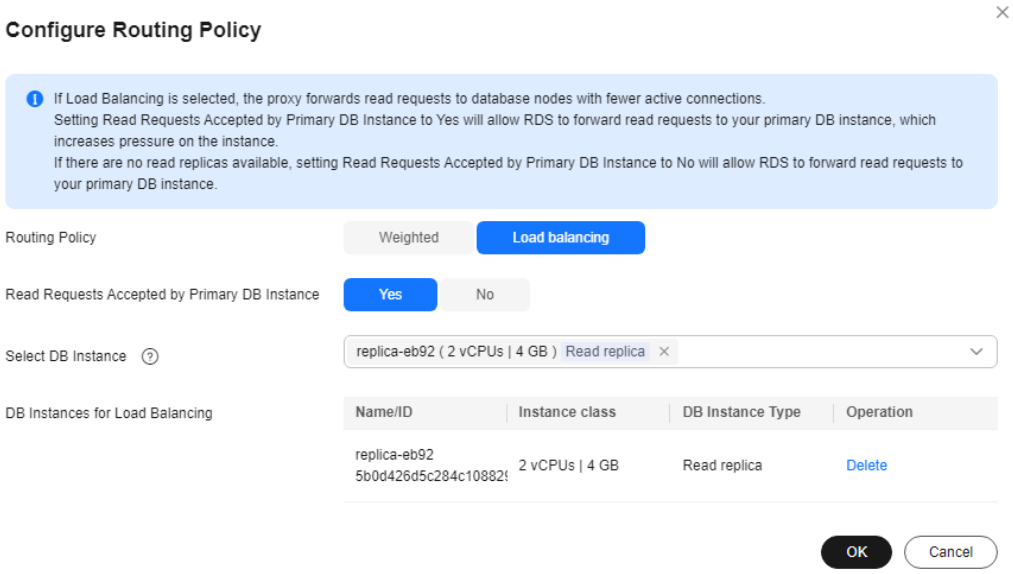
Click **OK** and view the weights in the proxy information area.

Figure 1-137 Viewing weights



- **Load balancing**: If selected, to balance the load among read replicas, read requests are automatically distributed to multiple read replicas based on the number of active connections.

Figure 1-138 Load balancing



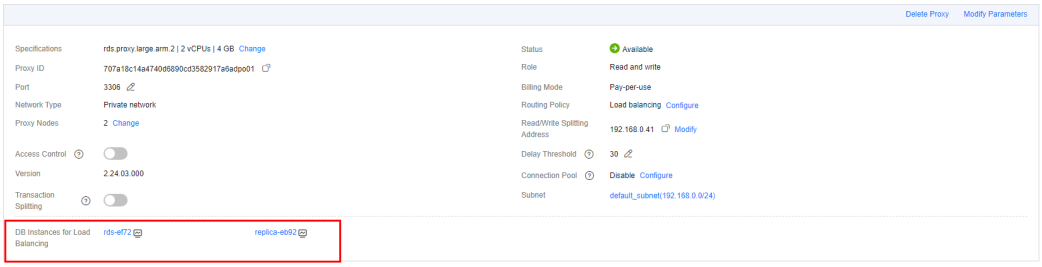
In the **Select DB Instance** drop-down list, select the instances for load balancing.

NOTE

To add new read replicas for load balancing, select the read replicas from the **Select DB Instance** drop-down list and click **OK**.

Click **OK** and view DB instances for load balancing in the proxy information area.

Figure 1-139 Viewing DB instances for load balancing



----End

Adding a Hint to Specify the Direction that a SQL Statement Will Be Routed

Hints supported by read/write splitting are as follows:

- /*FORCE_MASTER*/: A SQL statement is routed to the primary DB instance.
- /*FORCE_SLAVE*/: A SQL statement is routed to a read replica.

NOTE

- In addition to the weight distribution system of read/write splitting, hints are a useful type of SQL syntax that allows you to specify whether a SQL statement is executed on the primary DB instance or on a read replica.
- Hints are only used as routing suggestions. In non-read-only SQL and non-transaction scenarios, SQL statements cannot be routed to read replicas.

1.12.4.5 Enabling or Disabling Access Control

If load balancing is enabled for a database proxy instance, the security group associated with the proxy instance does not apply. You need to use access control to grant access from specific IP addresses.

NOTE

If access control is not displayed on the RDS console, the security group associated with the proxy instance is used.

Enabling Access Control

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

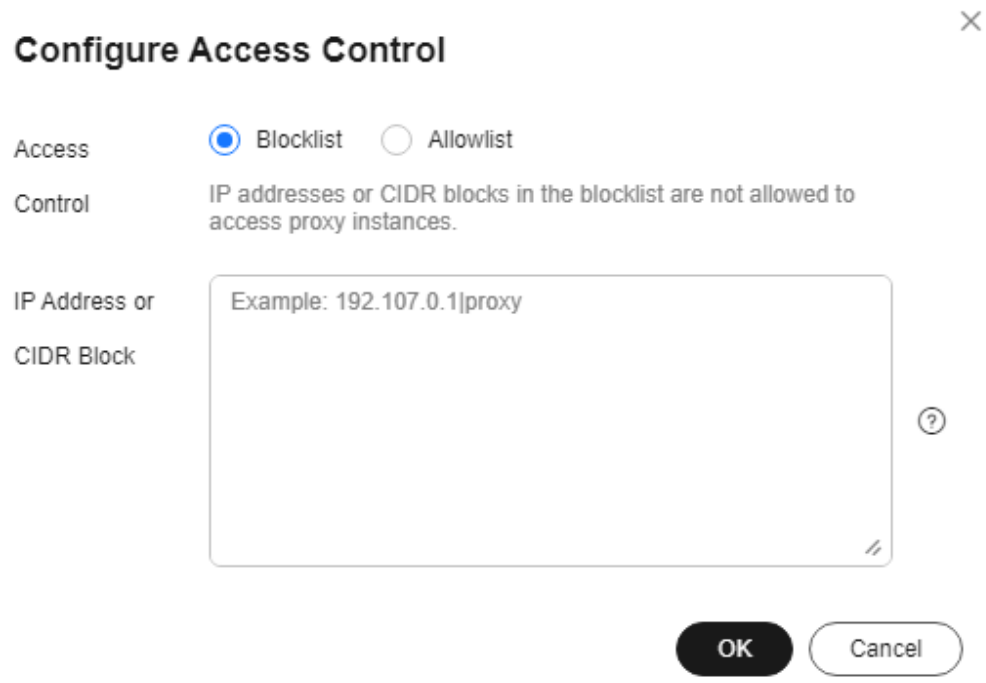
Step 4 On the **Instances** page, click the target DB instance.

Step 5 On the **Database Proxy** page, in the proxy information area, click next to the **Access Control** field.

Step 6 Click **Configure**. In the displayed dialog box, set the access control mode and IP addresses or CIDR blocks.

- **Access Control**: The blocklist and allowlist cannot be configured at the same time. If you switch between lists, your previously entered settings will be lost. IP addresses or CIDR blocks in the blocklist are not allowed to access proxy instances.
- **IP Address or CIDR Block**: Enter valid IP addresses or CIDR blocks that meet the following requirements:
  - Each line contains an IP address or a CIDR block and ends with a line break.
  - Each IP address or CIDR block can include a description separated by a vertical bar symbol (|), for example, 192.168.10.10|RDS01. The description can include up to 50 characters but cannot contain angle brackets (<>).
  - Up to 300 IP addresses or CIDR blocks can be added.

Figure 1-140 Configuring access control



----End

Disabling Access Control

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Database Proxy** page, in the proxy information area, click next to the **Access Control** field.

----End

1.12.4.6 Changing the Read/Write Splitting Address

Scenarios

After read/write splitting is enabled, you can change the read/write splitting address.

Precautions

Changing the read/write splitting address will interrupt database connections and services. Therefore, change the read/write splitting address during off-peak hours or when services are stopped.

Constraints

The new IP address is not in use and must be in the same subnet as the RDS for MySQL instance.

Procedure

You can change the read/write splitting address for DB instances with read/write splitting enabled.

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 Click **Database Proxy** in the navigation pane on the left. On the displayed page, click **Modify** next to the **Read/Write Splitting Address** field.

Figure 1-141 Read/write splitting area (single-proxy)

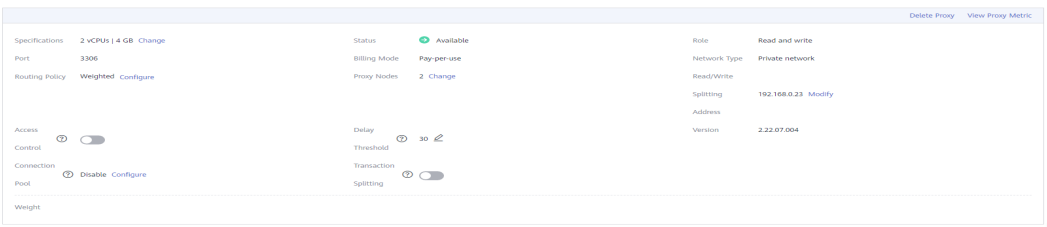
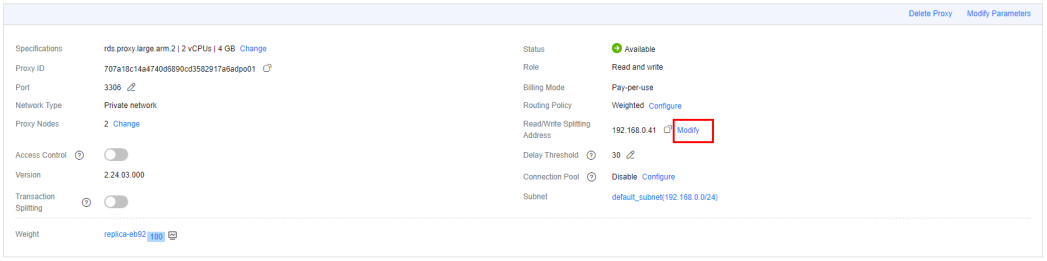


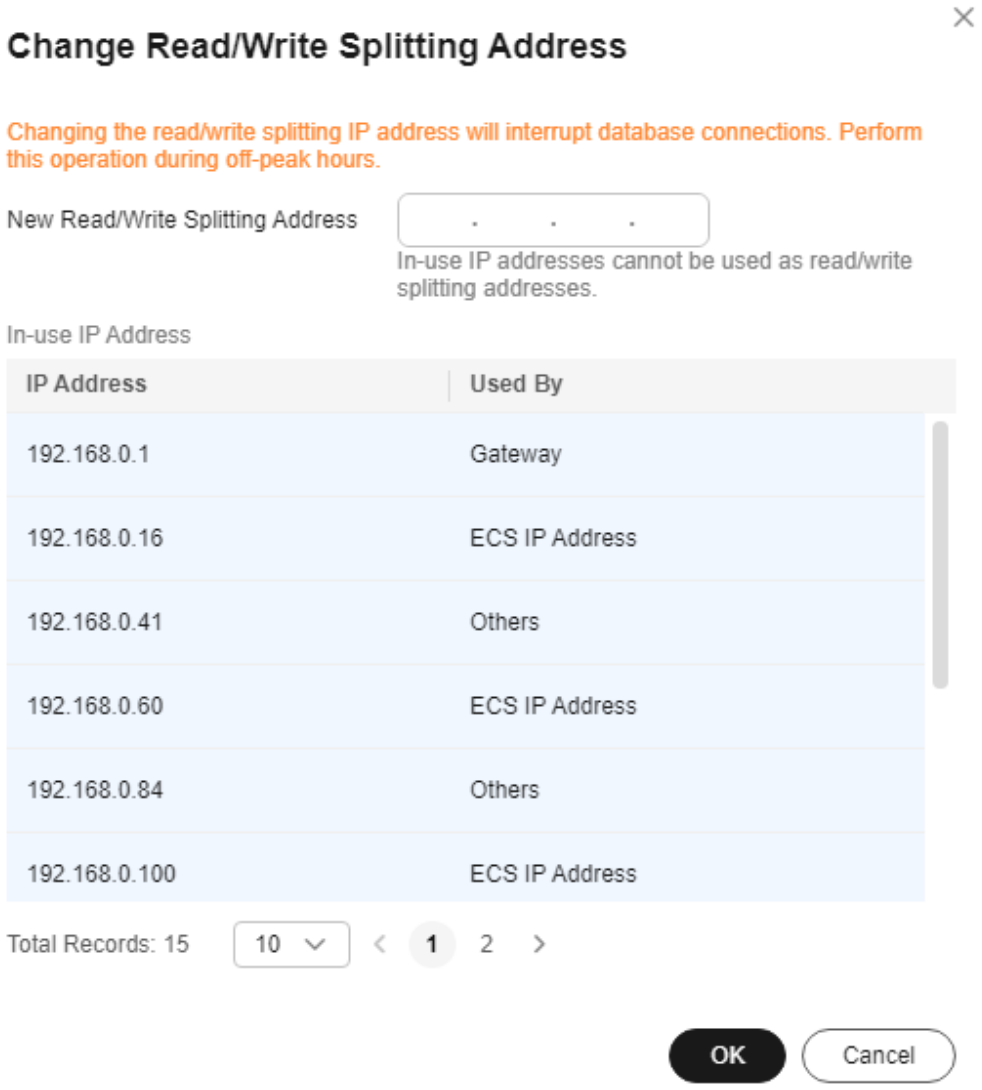
Figure 1-142 Read/write splitting area (multi-proxy)



Step 6 In the displayed dialog box, enter a new address. Click **OK**.

In-use IP addresses cannot be used as read/write splitting addresses.

Figure 1-143 Changing the read/write splitting address



----End

1.12.4.7 Applying for and Changing a Private Domain Name for a Database Proxy

Scenarios

After a database proxy is created, you can apply for a private domain name for it and change the domain name later as needed.

Precautions

- To use a private domain name for a database proxy, you need to **submit a service ticket** to request required permissions.
- The private domain name must be unique in a given region.
- Changing the private domain name for a database proxy will interrupt your database connection. To reconnect to the proxy, change the connection address of your applications. The new private domain name is applied to the proxy about 5 minutes after the change.

Applying for a private domain name

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

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

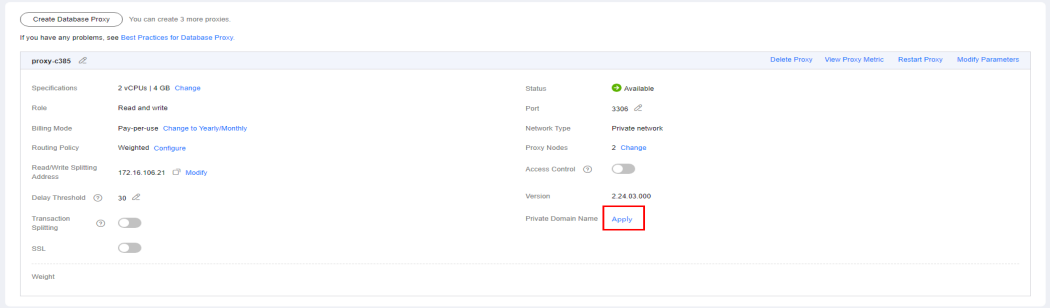
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Step 6 In the proxy information area, click **Apply** next to the **Private Domain Name** field.

Figure 1-144 Proxy information



Step 7 In the displayed dialog box, click **OK**.

----End

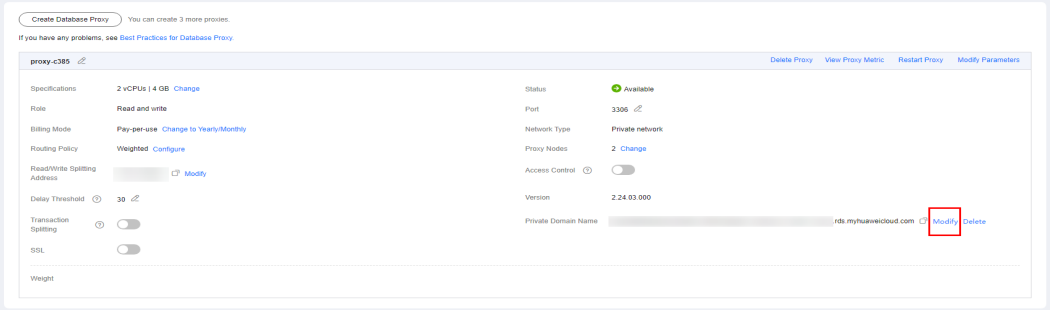
Changing a Private Domain Name

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane on the left, choose **Database Proxy**.

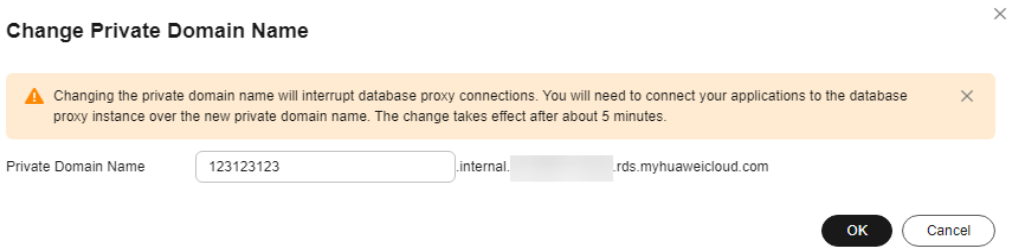
Step 3 In the proxy information area, click **Change** next to the **Private Domain Name** field.

Figure 1-145 Proxy information



Step 4 In the displayed dialog box, enter a new domain name and click **OK**.

Figure 1-146 Changing a private domain name



NOTE

- Only the prefix of a private domain name can be modified.
- The prefix of a private domain name contains 8 to 63 characters, and can include only lowercase letters and digits.
- The new private domain name must be different from existing ones.

Step 5 If the private domain name is no longer needed, click **Delete** next to the **Private Domain Name** field.

----End

1.12.4.8 Changing the Read/Write Splitting Port

Scenarios

After read/write splitting is enabled, you can change the read/write splitting port as needed.

Constraints

To change the read/write splitting port, you need to **submit a service ticket** to apply for required permissions.

The read/write splitting port can be changed only for ELB proxies.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Step 6 On the displayed page, click next to the **Port** field.

A database proxy can use a port ranging from 1024 to 65535, excluding 12017, 33071, and 1033, which are used by RDS.

- To submit the change, click .
  - In the displayed dialog box, click **OK**.

    Changing the proxy port number interrupts the database connection. You are advised to change the port number during off-peak hours.
  - To cancel the change, click **Cancel**.
- To cancel the change, click .

----End

1.12.4.9 Changing the Number of Proxy Nodes

Scenarios

After read/write splitting is enabled, you can change the number of proxy nodes as required.

Prerequisites

- Read/write splitting has been enabled.
- The primary instance, read replicas, and database proxies must be all available if you want to change nodes of ELB proxies.

Constraints

- The number of proxy nodes ranges from 2 to 8.
- Only one order can be created at a time for changing the instance classes or nodes of multiple yearly/monthly proxies of the same DB instance.

Procedure

Step 1 Log in to the management console.

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

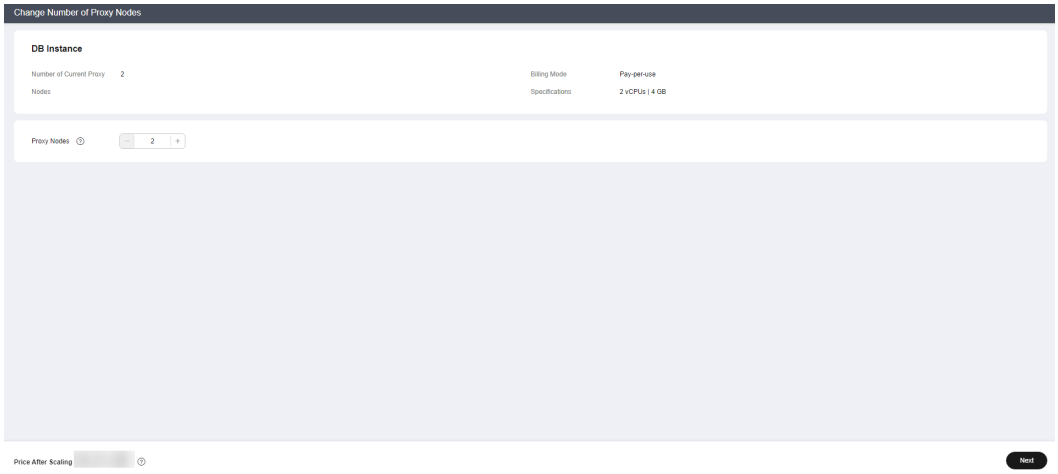
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the proxy information area on the **Database Proxy** page, click **Change** next to the **Proxy Nodes** field.

Step 6 Set the number of proxy nodes and click **Next**.

Figure 1-147 Changing the number of proxy nodes



Step 7 Confirm the settings and click **Submit**.

----End

1.12.4.10 Changing the Instance Class of a DB Proxy Instance

Scenarios

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

Only the instance classes of pay-per-use DB proxy instances can be changed.

Constraints

- You can change the instance class of a DB proxy instance only when the statuses of your primary DB instance, read replicas, and DB proxy instance are **Available**.
- A DB proxy instance cannot be deleted when its instance class is being changed.
- Changing the instance class of a DB proxy instance will cause the instance to reboot. Therefore, perform the operation during off-peak hours.
- Only one order can be created at a time for changing the instance classes or nodes of multiple yearly/monthly proxies of the same DB instance.

Procedure

Step 1 Log in to the management console.

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

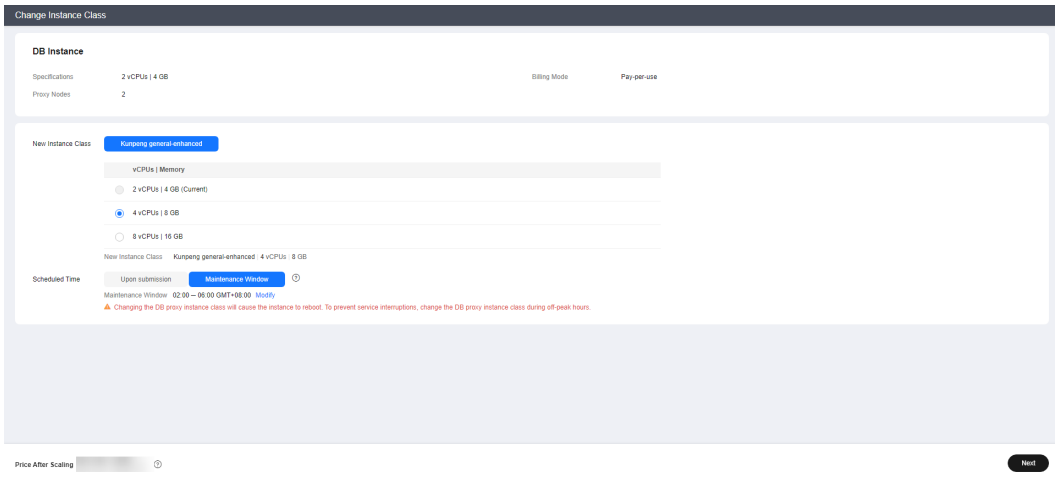
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 Choose **Database Proxy** from the navigation pane on the left. In the proxy information area, click **Change** next to the **Specifications** field.

For pay-per-use DB proxy instances, configure **New Instance Class** and **Scheduled Time** on the **Change Instance Class** page, and then click **Next**.

Figure 1-148 Changing a DB proxy instance class



- You can change the DB proxy instance class if required.
- Changing the DB proxy instance class will cause the instance to reboot. To prevent service interruptions, change the DB proxy instance class during off-peak hours.
- If you have selected **Maintenance Window** for **Scheduled Time**, the DB proxy instance will be rebooted during the instance class change time and services will be interrupted. To prevent service interruptions, you are advised to set the maintenance window to off-peak hours. For details, see **Changing the Maintenance Window**.

Step 6 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB proxy instances, click **Submit**.

  To view the cost incurred by the DB instance class change, choose **Billing & Costs** > **Bills** in the upper right corner.

Step 7 View the instance class change result.

Changing the DB proxy instance class takes 13–15 minutes. During this period, the status of the primary DB instance on the **Instances** page is **Changing proxy**

instance class. After a few minutes, view the proxy instance class on the **Database Proxy** page to check that the change is successful.

----End

1.12.4.11 Configuring Multi-Statement Processing Modes

Scenarios

You can configure the way how database proxies process **multiple statements** as needed.

Multi-Statement Processing Modes

- **Strict** (default): If a request containing multiple statements is routed to the primary instance, the subsequent read and write requests sent over the same connection are all routed to the primary instance. Read/write splitting can be restored only after you disconnect your connection to the DB instance and re-establish it. Your database proxy will not parse these statements, so the **Strict** mode is suitable when short connections are used or there is no connection reuse.
- **Loose**: If a request containing multiple statements is routed to the primary instance, the subsequent requests sent over the current connection can still be routed to the primary instance or read replicas. Your database proxy will not parse these statements, so **Loose** is recommended when multiple statements contain only DML SQL statements and do not contain operations like setting session variables, creating temporary tables, creating stored procedures, or executing uncommitted transactions.
- **Parse**: A read-only request containing multiple statements is routed based on weights. A read/write request containing multiple statements is routed to the primary instance, and the database proxy parses these statements and determines whether to split subsequent read and write requests received over the current connection based on the operations in the SQL statements (**Parse Mode**). Parsing a multi-statement request consumes more resources. The impact on proxy performance depends on the length and complexity of the statements, so it is recommended that the statements be less than 100 MB.

Parse Mode

If a multi-statement request contains any of the following operations, all subsequent requests are routed to the primary instance. To restore read/write splitting, you need to disconnect the connection and then re-establish it.

- Creating temporary tables
- Creating stored procedures
- Executing uncommitted transactions (For example, **begin** is executed but **commit** or **rollback** is not executed.)
- Executing complex or special syntax (In this case, parsing these statements will fail.)

Changing the multi-statement processing mode applies to your proxy immediately. You do not need to reboot the proxy. If read/write splitting is invalid on the

connection over which the proxy has processed a multi-statement request, changing the multi-statement processing mode will not restore read/write splitting on this connection. You need to re-establish it.

1.12.4.12 Changing a Proxy from Pay-per-Use to Yearly/Monthly

Scenarios

If you want to use a pay-per-use proxy created for a yearly/monthly DB instance for a long time, you can change the proxy from pay-per-use to yearly/monthly to reduce costs.

Constraints

- Enabling database proxy
  - Only pay-per-use proxies can be created for pay-per-use DB instances.
  - Either pay-per-use or yearly/monthly proxies can be created for yearly/monthly DB instances.
    - To create yearly/monthly proxies, **submit a service ticket** to apply for required permissions.
    - The expiration time of yearly/monthly proxies is the same as that of the yearly/monthly DB instance by default.
    - If auto-renewal is enabled for the yearly/monthly DB instance, it is also enabled for the proxies by default.
- Changing the billing mode
  - To change a proxy from pay-per-use to yearly/monthly, you need to **submit a service ticket** to apply for required permissions.
  - Pay-per-use proxies in HA mode cannot be changed to yearly/monthly.

Procedure

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

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

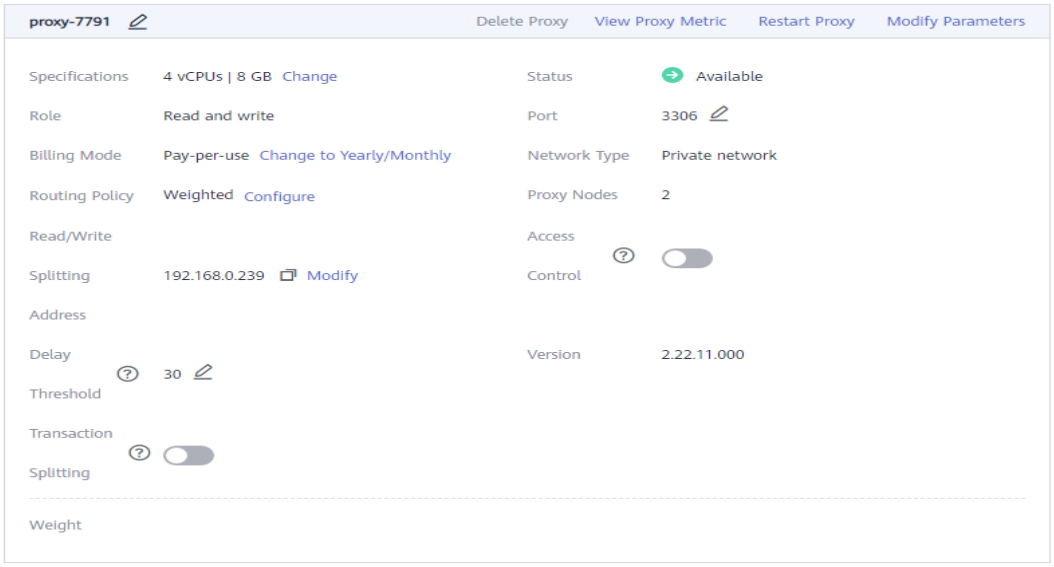
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the name of the target yearly/monthly instance.

Step 5 In the navigation pane on the left, click **Database Proxy**.

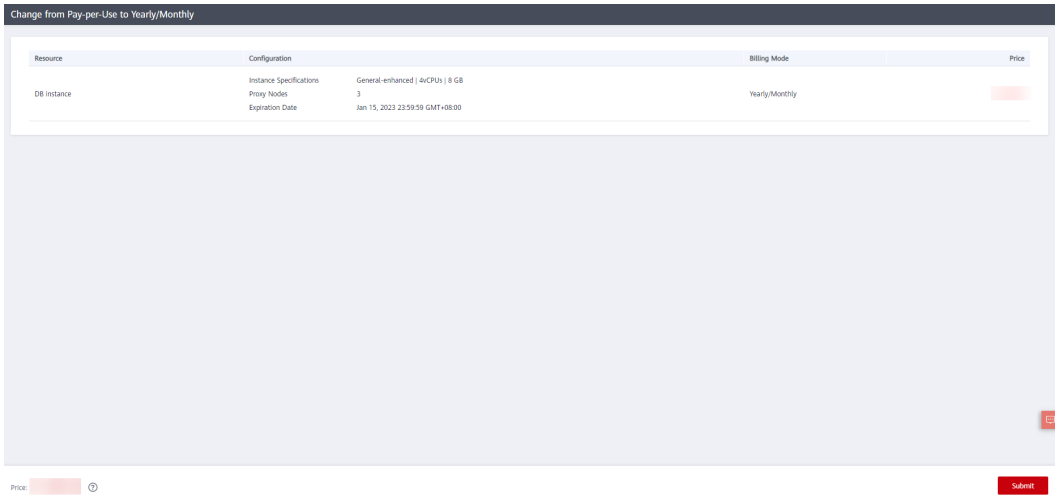
Step 6 In the proxy information area, click **Change to Yearly/Monthly** next to the **Billing Mode** field.

Figure 1-149 Proxy information area



Step 7 On the displayed page, confirm the information and click **Submit**.

Figure 1-150 Changing from pay-per-use to yearly/monthly



----End

1.12.5 Database Proxy Lifecycle

1.12.5.1 Restarting a Database Proxy

Scenarios

After read/write splitting is enabled, you can restart the database proxy when necessary.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Step 6 On the displayed page, click **Restart Proxy** in the proxy information area.

Step 7 In the displayed dialog box, click **OK**.

NOTE

Restarting the proxy interrupts the database connection. You are advised to restart it during off-peak hours.

----End

1.12.5.2 Disabling Read/Write Splitting

You can disable read/write splitting as required. If the multi-proxy function is enabled, you can delete the proxy.

Billing

After a database proxy is deleted, it is no longer billed.

Procedure

Step 1 Log in to the management console.

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

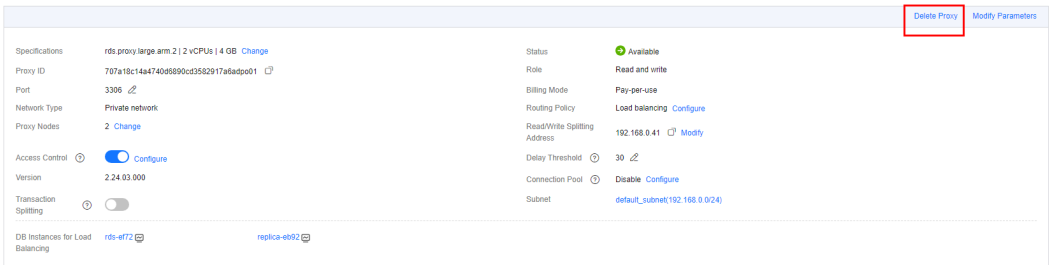
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Database Proxy**.

Step 6 In the proxy instance area, click **Delete Proxy**.

Figure 1-151 Proxy instance area



Step 7 In the displayed dialog box, click **OK**.

NOTE

- If the database proxy is disabled, read/write splitting is also disabled and services using the read/write splitting address are interrupted. You need to switch your applications to the instance address.
- After read/write splitting is disabled, read replicas are still be billed. You can release them if they are not required for your workloads anymore.

----End

1.12.6 Database Proxy Kernel Versions

1.12.6.1 Kernel Versions

The following table describes the updates in each kernel version of RDS for MySQL database proxy.

Version	Description
2.24.09.020	New features • Support for IPv6 Resolved issues • Resolved the issue that proxy resource reclamation is slow after SSL is enabled. • Resolved the issue that read requests cannot be split from write requests after transaction splitting is enabled with **set autocommit** set to **0** and **commit** used to commit transactions.
2.24.06.000	Resolved issues • Resolved the issue that SELECT statements starting with **(** are sent to the primary instance.
2.23.12.000	New features • Reduced proxy authentication synchronization latency, so that new accounts and databases can be synchronized more quickly. • Full-link error tracking • Display of slow SQL statements recorded by database proxy
2.23.09.002	Resolved issues • Optimized the logic for the proxy to retry service SQL statements after the instance breaks down.
2.23.09.001	Resolved issues • Fixed the issue that an error is occasionally reported during execution of the prepared SELECT FOR UPDATE statement.

Version	Description
2.23.09.000	New features • **Change User** protocol • Parsing of multiple hints • **show processlist** and **kill** commands • Resolved the issue that the **set autocommit** setting is synchronized to read replicas after transaction splitting is enabled.
2.23.06.001	Resolved issues • Resolved the increased backend database connections caused by enabling session connection pool.
2.23.06.000	• New features Binlog pulling through the proxy kernel • Resolved issues Optimized the performance of the **prepare stmt** protocol again.
2.23.02.007	Resolved issues • Optimized the performance of the **prepare stmt** protocol. • Resolved unexpected traffic allocation of the **/* FORCE_SLAVE*/ Hint** statement.
2.23.02.000	• Resolved issues Optimized the database proxy performance.
2.22.11.000	• New features Multi-statement processing modes • Resolved issues Optimized the error messages reported during SQL statement execution in some scenarios.
2.22.07.000	• New features – Session-level connection pooling – Dynamic load balancing • Resolved issues – Optimized the logic for setting session-level transaction isolation levels of database proxies. By default, the transaction isolation levels are synchronized with those of the database.
2.7.5.0	Application Lossless and Transparent (ALT)
2.7.4.0	• New features – A query for more than 16 MB of data • Resolved issues – Optimized the way how metrics of read-only proxies are collected by Cloud Eye.

Version	Description
2.3.9.8	New features • Batch execution of prepared statements
2.3.9.7	• New features – Support for MySQL 8.0 – Transaction splitting – Read-only mode • Resolved issues Optimized the execution logic of prepared statements to improve performance.
2.3.9.0	• New features Added database proxy metrics **Front-End Connections Created per Second**, **Transaction Queries per Second**, and **Multi-Statement Queries per Second**. For details, see **Table 1-88**. • Resolved issues – Optimized the database proxy performance. – Resolved traffic congestion occurring when your applications connect to database proxy over short connections.
2.3.8.0	• New features Real client IP addresses can be obtained through database proxies. • Resolved issues – Fixed the issue that monitoring data of database proxy is inaccurate. – Shortened the downtime of database proxies during a primary/standby switchover.
2.3.6.0	• Fixed the issue of connection failures caused by database overload. • Improved proxies' compatibility with MySQL protocols.
2.3.1.0	• Maintaining connectivity between clients and database proxies. • Monitoring metrics of database proxies. For details, see **Table 1-88**.

1.12.6.2 Upgrading the Kernel Version of Database Proxy

Scenarios

You can manually upgrade the RDS for MySQL database proxy to the latest kernel version to improve performance, add new functions, and fix problems.

For details about kernel versions, see **Kernel Versions**.

Upgrade Methods

A kernel version can be upgraded in either of the following ways:

- Upon submission: The system **upgrades the proxy instance version** immediately after you submit the upgrade request.
- In maintenance window: When you submit an upgrade request, the system waits for a maintenance window to perform the upgrade. For details on how to change the maintenance window, see **Changing the Maintenance Window**.

If the kernel version of your instance has potential risks or major defects, has expired, or has been brought offline, the system will notify you by SMS message or email and deliver an upgrade task during the maintenance window.

Precautions

- The upgrade duration depends on how many nodes your database proxy has. Perform the upgrade during off-peak hours.
- During the upgrade, short connections are not affected. Persistent connections lasting for more than 24 hours will be interrupted intermittently.
- Only HA or cluster proxies can be upgraded. To confirm the proxy type, **submit a service ticket**.

Constraints

- Only proxy instances with kernel version 2.3.0.1 or later can be upgraded manually on the console.
- A version upgrade cannot be rolled back after the upgrade is complete.

Procedure

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

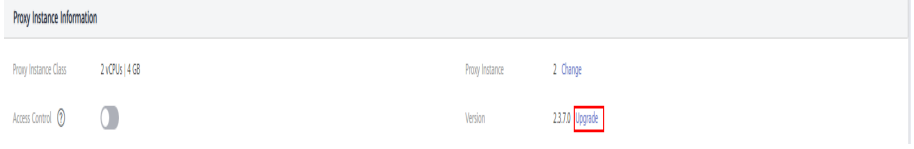
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

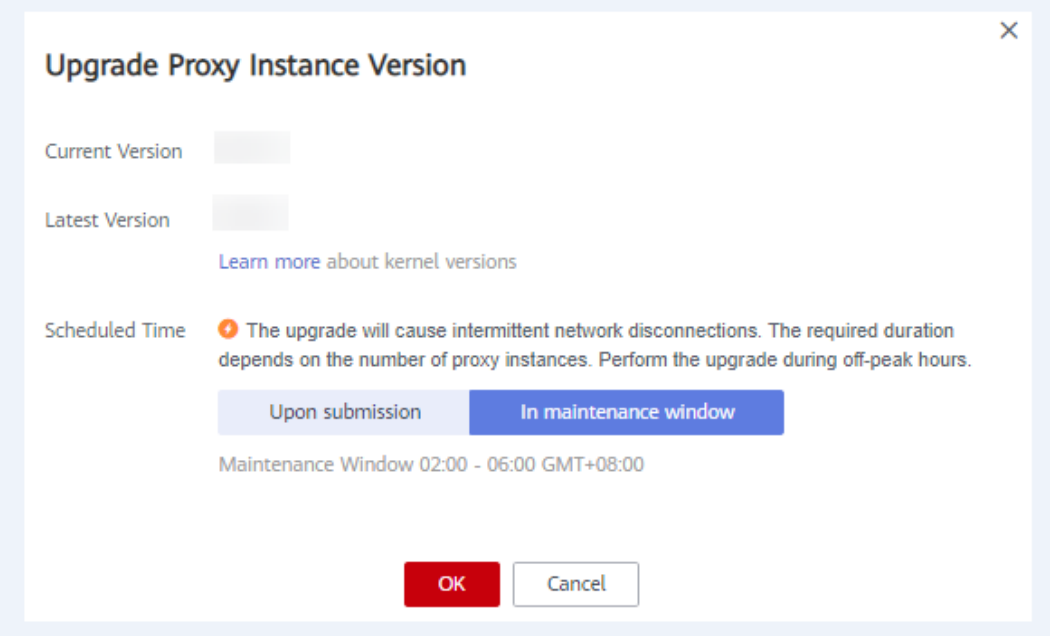
Step 5 On the **Database Proxy** page, in the proxy information area, click **Upgrade** next to the **Version** field.

Figure 1-152 Upgrading the proxy instance version



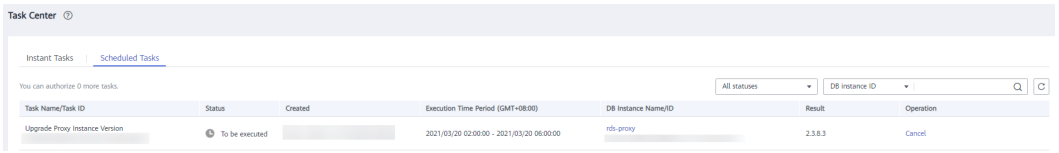
Step 6 In the displayed dialog box, select a scheduled time and click **OK**.

Figure 1-153 Selecting a scheduled time



- Upon submission: The system upgrades the proxy instance to the latest version immediately after you submit the request. You can view the task progress in **Task Center** > **Instant Tasks**.
- In maintenance window: The system upgrades the proxy instance to the latest version during a maintenance window. You can view the task progress in **Task Center** > **Scheduled Tasks**.

Figure 1-154 Viewing a scheduled task



----End

1.12.7 Best Practices for Database Proxy

Using Hints for Read/Write Splitting

In addition to the weight distribution system of read/write splitting, hints are a useful type of SQL syntax that allows you to specify whether a SQL statement is executed on the primary DB instance or on a read replica.

- Hints are only used as routing suggestions. In non-read-only SQL and non-transaction scenarios, SQL statements cannot be routed to read replicas.
- If you connect to an instance using a MySQL CLI and want to run HINT in the CLI, add the **-c** option in the statement.

Hints supported by read/write splitting are as follows:

/*FORCE_MASTER*/: A SQL statement is routed to the primary DB instance.

/*FORCE_SLAVE*/: A SQL statement is routed to a read replica.

For example, **select * from table1** will be routed to a read replica by default. If you change it to **/*FORCE_MASTER*/ select * from table1**, it will be forcibly routed to the primary DB instance.

CAUTION

/*FORCE_MASTER*/ only works for read/write addresses. If your primary DB instance is read-only, adding **/*FORCE_MASTER*/** will not help route the SQL statement to the primary instance.

Connection Pool Configuration

To ensure that your application obtains an available connection from a connection pool, you need to configure how the connection pool will check connection availability. For example, set **testOnBorrow** to **true** for a JDBC or Druid connection pool or set **connectionTestQuery** to **SELECT 1** for a HikariCP connection pool.

```
<bean id="hikariConfig" class="com.zaxxer.hikari.HikariConfig">
    <property name="poolName" value="springHikariCP" />
    <property name="connectionTestQuery" value="SELECT 1" />
    <property name="dataSourceClassName"       value="com.mysql.jdbc.jdbc2.optional.MysqlDataSource" />
    <property name="dataSourceProperties">
        <props>
            <prop key="url">${jdbc.url}</prop>
            <prop key="user">${jdbc.username}</prop>
            <prop key="password">${jdbc.password}</prop>
        </props>
    </property>
</bean>

<bean id="dataSource" class="com.zaxxer.hikari.HikariDataSource" destroy-method="close">
    <constructor-arg ref="hikariConfig" />
</bean>
```

Read Requests Routed to the Primary DB Instance

1. If a query statement is placed in a transaction, all transaction requests will be routed to the primary DB instance. If **set autocommit=0** is configured before a query statement, the query statement will be treated as a transaction and routed to the primary DB instance.
2. If no read replica exists, all read replicas are abnormal, or the read weights allocated to the read replicas are 0, queries will be routed to the primary DB instance. You can set read weights allocated to read replicas and the primary DB instance after read/write splitting is enabled. For details, see **Configuring the Delay Threshold and Routing Policy**.
3. If multiple statements (for example, **insert ***;select ***) are executed, all subsequent requests will be routed to the primary DB instance. To restore read/write splitting, disconnect the connection from your applications and then reconnect.
4. Read operations with locks (for example, **SELECT for UPDATE**) will be routed to the primary DB instance.
5. When the **/*FORCE_MASTER*/** hint is used, requests will be routed to the primary DB instance.

1.13 Problem Diagnosis and SQL Analysis

1.13.1 Function Overview

DBA Assistant provides visualized database O&M and intelligent diagnosis for developers and database administrators (DBAs), making database O&M easy and efficient. By analyzing alarms, resources, health data, performance metrics, and storage usage, it helps users quickly locate faults and keep track of instance status.

NOTE

To use DBA Assistant on the RDS console, IAM users must have the **RDS FullAccess**, **DAS FullAccess**, **DAS Administrator**, and **CES FullAccess** permissions. For details, see **Creating a User and Granting Permissions**.

Function Description

Scenarios

- Setting a slow session threshold can help you quickly identify abnormal sessions and kill the sessions when an exception occurs in your instance, so that your instance can recover quickly and ensure database availability.
- If your DB instance is unstable due to a large number of concurrent SQL requests from new services, you can set concurrency control rules for SQL statements to limit concurrent SQL statements and ensure instance stability.
- If your instance storage is full, you can learn about the storage usage and disk space distribution on the **Storage Analysis** page. You can enable storage autoscaling. When the available storage of your instance drops to the threshold, autoscaling is triggered. For details, see **Configuring Storage Autoscaling**.
- You can configure auto flow control to limit active connections in high burst traffic or abnormal read/write scenarios to ensure the availability of core workloads.

Functions

Table 1-70 lists the functions supported by DBA Assistant.

Table 1-70 Function description

Function	Description	Reference
Dashboard	Shows the status of your instance, including alarms, resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.	**Viewing the Overall Status of a DB Instance**
Sessions	The **Sessions** page displays slow sessions, active sessions, and total sessions. You can quickly filter slow sessions or active sessions by user, host IP address, or database name. Kill Session and Concurrency Control can be used for urgent instance recovery to ensure database availability.	**Managing Real-Time Sessions**
Performance	The **Performance** page displays key metrics of your instance and provides metric comparison between different days. You can keep track of metric changes and detect exceptions in a timely manner. Monitoring by Seconds helps accurately locate faults.	**Viewing Performance Metrics of a DB Instance**
Storage Analysis	Storage occupied by data and logs and historical changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DBA Assistant can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner. **Autoscaling**, **Abnormal Tables**, **Top 50 Databases**, and **Top 50 Tables** are also available on this page.	**Managing Disk Capacity**
Locks & Transactions	The **Locks & Transactions** page displays metadata locks and InnoDB locks. You can manage blocked transactions, optimize your workloads, and reduce lock conflicts based on the lock information.	**Managing Locks & Transactions**
Historical Transactions	This function is used to analyze and discover large transactions and the transactions that have been there for a long time without being committed in the database.	**Managing Historical Transactions**
Slow Query Log	Displays slow queries within a specified time period. You can view top 5 slow query logs by user or IP address, sort statistics, and identify sources of slow SQL statements.	**Viewing Slow Query Logs of a DB Instance**

Function	Description	Reference
SQL Explorer	After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.	• **Viewing Top SQL Statements of a DB Instance** • **Creating a SQL Insights Task**
Concurrency Control	Concurrency Control restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.	**Creating a Concurrency Control Rule**
Auto Flow Control	Automatically detects database exceptions such as high vCPU usage and excessive active sessions, and limits traffic based on specified priorities. You can control traffic by database or user as required. Limiting traffic of non-core databases or from non-core users can ensure that core workloads remain stable.	**Configuring Auto Flow Control**
Daily Reports	The **Daily Reports** page provides overall information about your instance status of the previous day, including slow SQL analysis, all SQL analysis, and performance & storage analysis. You can download and subscribe to analysis reports. A daily diagnosis is recommended.	**Daily Reports**
Anomaly Snapshots	This function intelligently detects instance anomalies and records information about session, lock, and transaction snapshots to facilitate subsequent fault locating.	**Managing Anomaly Snapshots**

1.13.2 Performance Monitoring

1.13.2.1 Viewing the Overall Status of a DB Instance

On the **Overview** page, you can get knowledge of the overall status of your RDS for MySQL instance, including alarms, intelligent anomaly diagnosis, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.

Functions

Table 1-71 lists the functions provided on the **Overview** page.

Table 1-71 Function description

Function	Description
Alarms	To view alarm details, click the number next to an alarm severity.
Intelligent Anomaly Diagnosis	Shows the health status of your instance based on operational data analytics and intelligent algorithms.
Performance Monitoring	Shows key performance metrics of the instance, including the CPU usage, memory usage, slow SQL queries, and connections.

Alarms

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

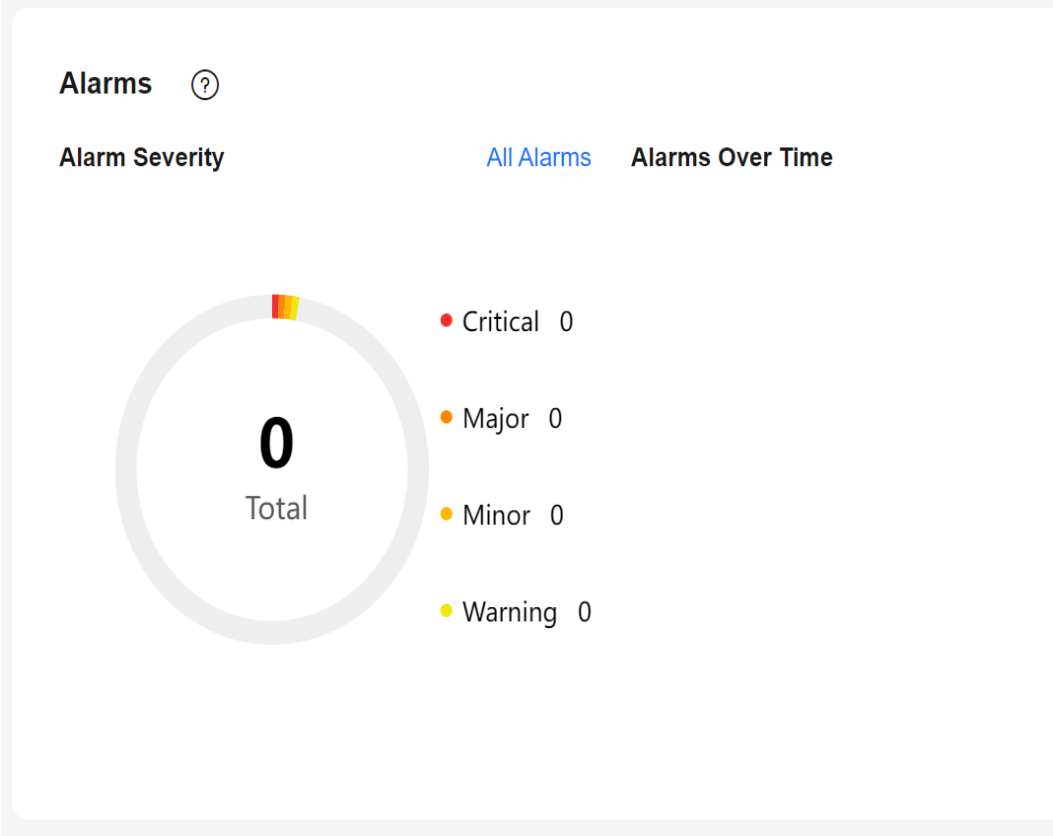
Step 4 On the **Instances** page, click the DB instance name.

Step 5 On the **Overview** page, view the status of your instance.

- In the **Alarms** area, view alarm information of your instance.

  To view the list of all alarms, click **All Alarms**. To view alarm details, click the number next to an alarm severity.

Figure 1-155 Alarms

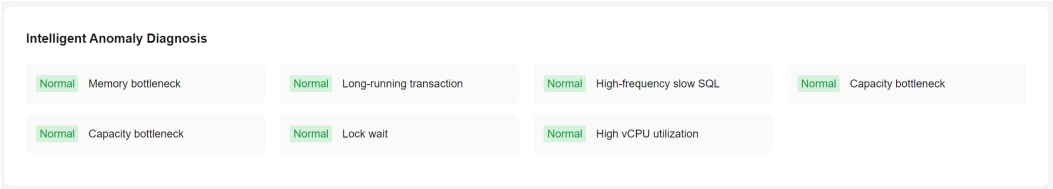


- In the **Intelligent Anomaly Diagnosis** area, view the health diagnosis results of your instance.

  NOTE

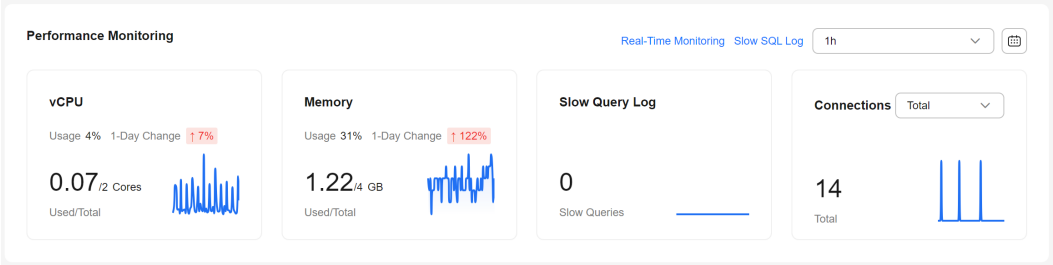
  Intelligent Anomaly Diagnosis provides diagnosis results for the check items in the past 5 minutes. If any diagnosis result is abnormal, the check item is abnormal in the past 5 minutes.

Figure 1-156 Intelligent Anomaly Diagnosis



- In the **Performance Monitoring** area, view key performance metrics of your instance.

Figure 1-157 Performance Monitoring



----End

1.13.2.2 Viewing Performance Metrics of a DB Instance

DBA Assistant allows you to view the performance metrics of your DB instance. Historical trends of performance metrics within a specified time period help you learn about the status and resource usage of your DB instance. If any alarm is reported, you can take actions timely.

Procedure

Step 1 Log in to the management console.

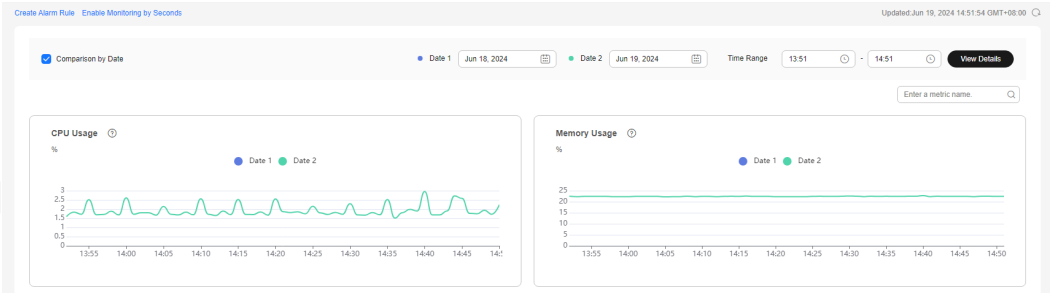
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Performance** tab to view historical performance for each metric of your instance within the same time range on different days.



----End

1.13.3 Problem Diagnosis

1.13.3.1 Managing Real-Time Sessions

Scenarios

You can query session snapshots of your instance while sorting, filtering, and displaying the snapshots as needed. You can filter and identify the desired slow SQL sessions and active sessions by user, host, and database. Sessions can be killed for urgent instance recovery to ensure database availability.

Precautions

Killing a session may cause service disconnection. Your applications should be able to reconnect to the instance.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

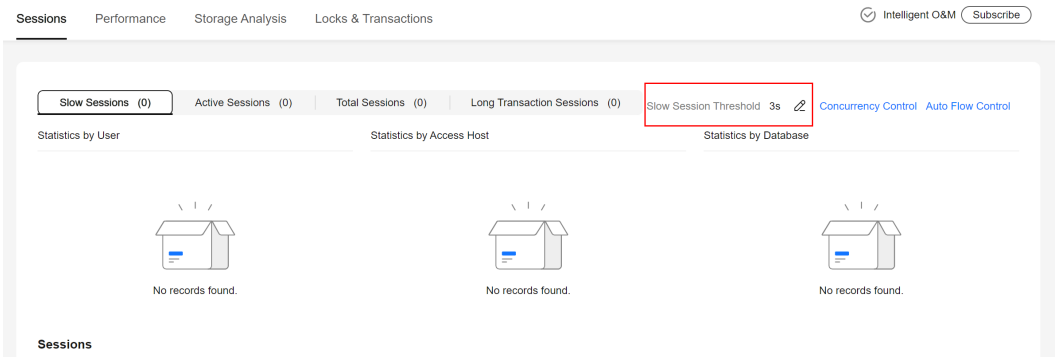
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Sessions** tab. You can perform the following operations on this tab page:

- Viewing session statistics

  In the sessions statistics, you can view statistics on the slow sessions, active sessions, total sessions, long transaction sessions by user, access host, and database.

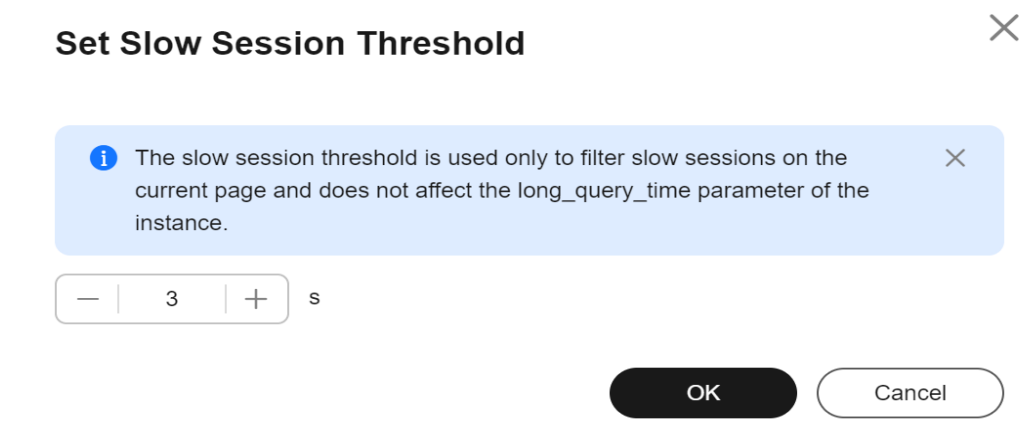
  Figure 1-158 Sessions



- Setting a slow session threshold

  Click next to the **Slow Session Threshold** field. In the displayed dialog box, set a slow session threshold and click **OK**. Sessions whose execution durations are greater than this threshold are automatically displayed.

Figure 1-159 Setting a slow session threshold



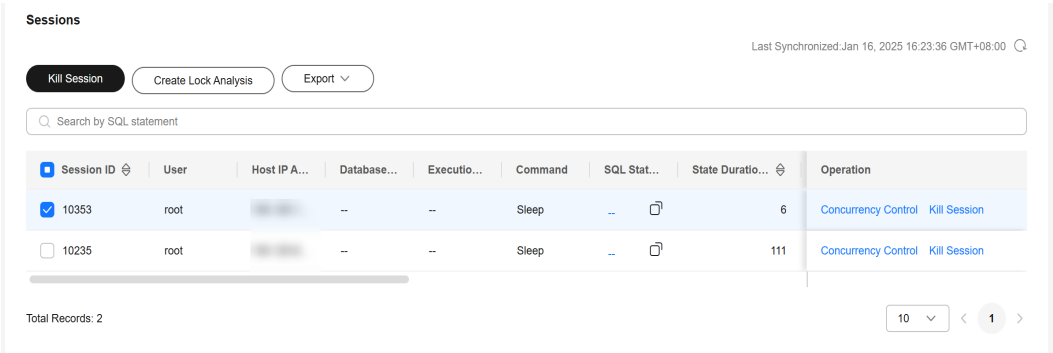
- Killing abnormal sessions

  In the session list, you can view session details. You can also select the abnormal session you want to end and click **Kill Session** to recover the database.

  NOTE

  A maximum of 100 sessions can be killed at a time.

Figure 1-160 Killing a session



- Configuring SQL statement concurrency control

  In the session list, click **Concurrency Control** and set the SQL type and keyword to match SQL statements. When the number of matched SQL statements exceeds the configured upper limit, the DB instance will refuse to execute the SQL statements, thus ensuring the instance stability.

  For details, see **Creating a Concurrency Control Rule**.

- Creating lock analysis

  To create a lock analysis task, log in to the DB instance on the **Locks & Transactions** page first. For details, see **Managing Locks & Transactions**. Then, click **Create Lock Analysis**. A lock analysis record is generated, which is used to check whether there are any sessions that hold locks.

- Exporting the session list

  Click **Export** to export all or specified sessions.

----End

1.13.3.2 Managing Disk Capacity

DBA Assistant allows you to view the storage usage of your DB instance in real time to prevent insufficient storage space.

Overview

Step 1 Log in to the management console.

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

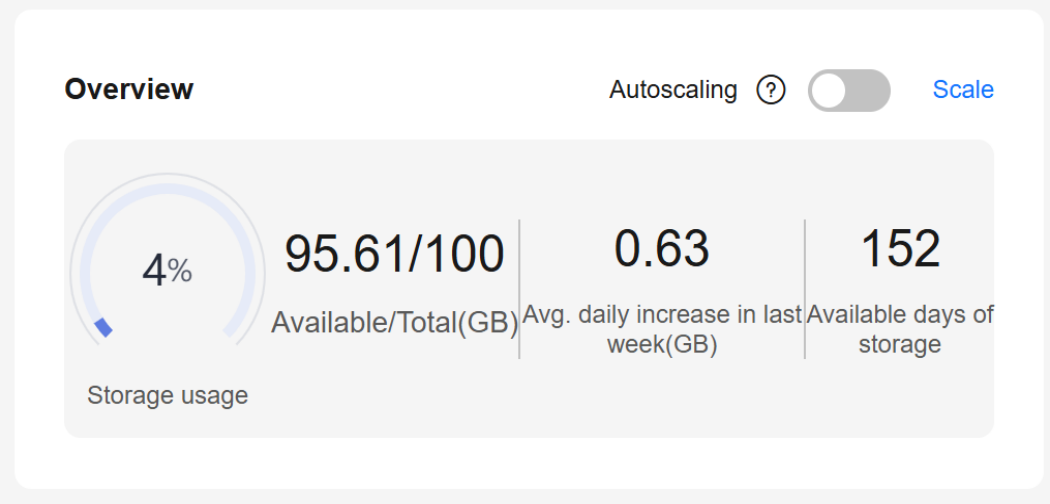
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab to view storage usage. If your storage is insufficient, scale it up. Or you can enable storage autoscaling.

Figure 1-161 Overview



NOTE

If the average daily increase in last week is 0 GB, the estimated available days of storage are unlimited and are not displayed.

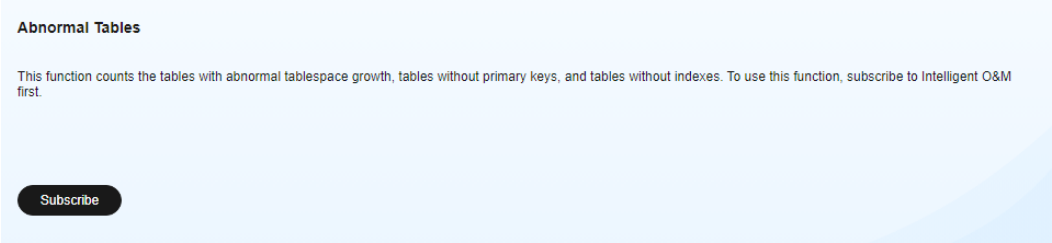
----End

Abnormal Tables

This function counts tables with abnormal tablespace growth, tables without primary keys, and tables without indexes. To use this function, subscribe to Intelligent O&M first.

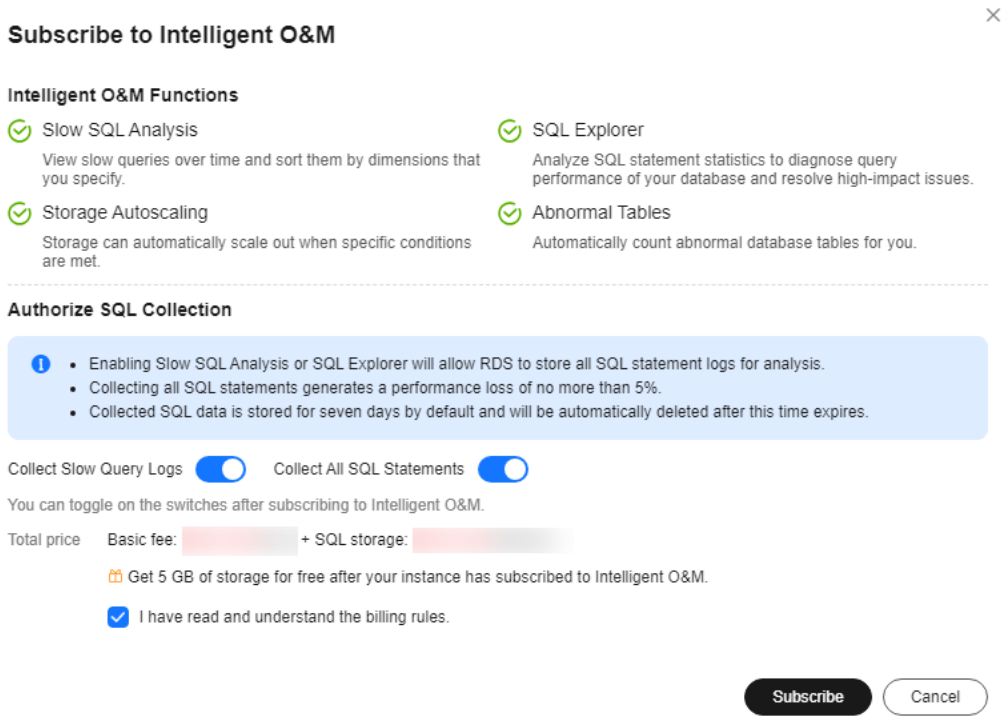
Step 1 Click the **Storage Analysis** tab to view abnormal tables.

Figure 1-162 Abnormal Tables



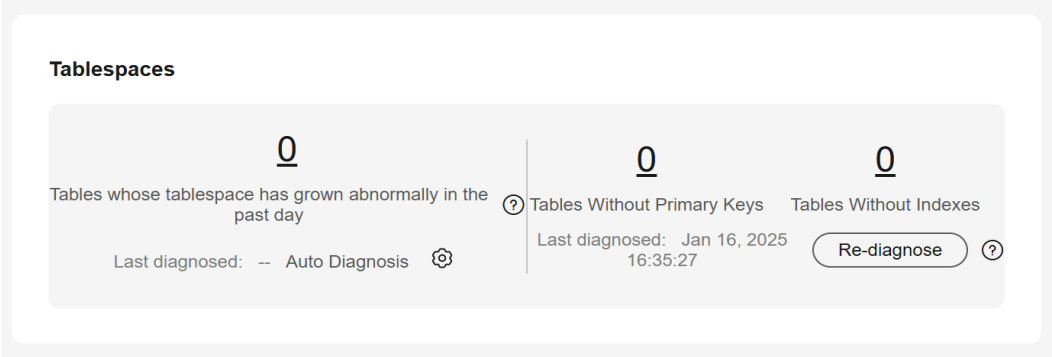
Step 2 Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.

Figure 1-163 Subscribing to Intelligent O&M



Step 3 After subscribing to Intelligent O&M, view the table diagnosis results of your instance.

Figure 1-164 Table diagnosis results



Step 4 Click next to **Auto Diagnosis**. In the displayed dialog box, configure the daily tablespace increase limit and click **OK**.

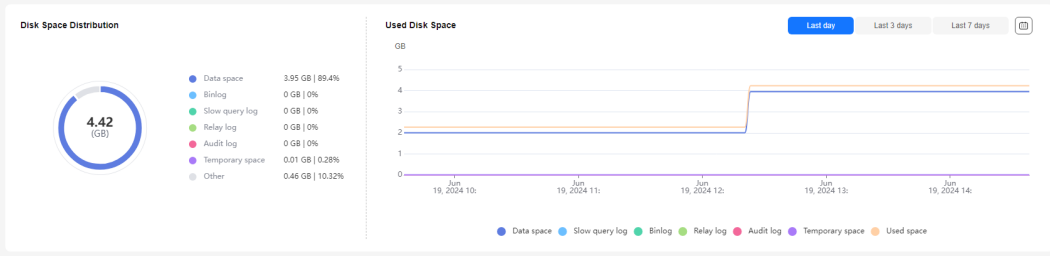
Figure 1-165 Configuring a daily tablespace increase limit

----End

Disk Space Distribution

You can view storage space distribution of your instance.

Figure 1-166 Disk space distribution



NOTE

If the total number of files in your disk space (including data space, binlog space, slow query log space, relay log space, audit log space, temporary space, and other space) exceeds 10,000, RDS will not collect information about the files or display disk space distribution and usages over time on the console. This prevents performance slowdowns caused by collecting statistics on too many files. If this happens, **submit a service ticket**.

- **Data space**: Disk space occupied by user data (including temporary table files and ib_logfile files generated by the database)
- **Binlog**: Disk space occupied by binlogs
- **Slow query log**: Disk space occupied by slow logs
- **Relay log**: Disk space occupied by relay logs
- **Audit log**: Disk space occupied by audit logs
- **Temporary space**: Disk space occupied by temporary files
- **Other**: Disk space occupied by files such as ibdata1, ib_buffer_pool, ib_doublewrite, and error.log, plus the reserved space (about 5% of the disk space) for OS

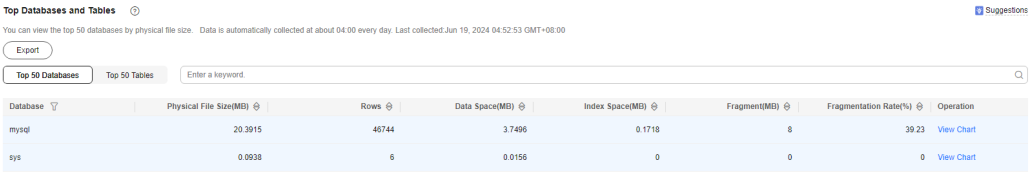
Top Databases and Tables by Physical File Size

You can view the top 50 databases and tables by physical file size and identify the databases and tables with high usage based on storage space distribution.

NOTE

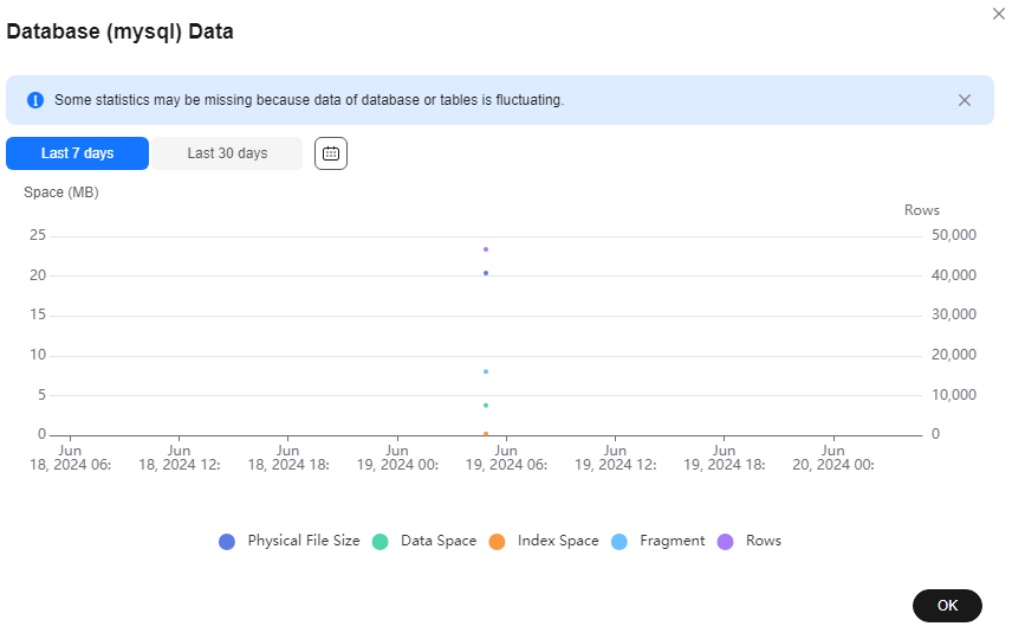
- Physical file sizes are precisely recorded, but other fields' values are estimated. If there is a large gap between a file size and another field, run ANALYZE TABLE on the table.
- A database or table whose name contains special characters, including slashes (/) and #p#p, is not counted.
- Top databases and tables are available only in RDS for MySQL 5.7 and 8.0.
- If the instance memory usage is greater than 85% or there are more than 50,000 tables in your instance, to prevent data collection from affecting the instance performance, top databases and tables will not be counted.

Figure 1-167 Top 50 databases



Click **View Chart** to view data volume changes in the last 7 days, last 30 days, or a custom time period (spanning no more than 30 days).

Figure 1-168 Viewing data volume changes



FAQ

Q: What can I do if the storage space of my DB instance is full?

A: Reduce the storage usage by referring to **Storage Full Situations Causing Instances to Be Read-Only**, so that the DB instance becomes available and data can be written to the instance. You can use either of the following methods to reduce the storage usage:

- Scale up the storage space: **Services are not interrupted during storage scale-up**. You can also enable autoscaling. When the available storage of a DB instance drops to the threshold, autoscaling is triggered.
- Reducing disk data: Delete useless historical data.
  a. If your instance becomes read-only, you need to **submit a service ticket** to cancel the read-only status first. If your instance is not in the read-only state, you can delete data directly.
  b. Check the top 50 databases and tables with large physical files and identify the historical table data that can be deleted. For details, see **Top Databases and Tables by Physical File Size**.
  c. To clear up space, you can optimize tables with a high fragmentation rate during off-peak hours.

     To delete data of an entire table, run **DROP** or **TRUNCATE**. To delete part of table data, run **DELETE** and **OPTIMIZE TABLE**.
- If temporary files generated by sorting queries occupy too much storage space, optimize your SQL query statements.

  You can query **slow query logs**, and analyze and optimize the problematic SQL statements.

1.13.3.3 Managing Locks & Transactions

Scenarios

DBA Assistant allows you to check whether your DB instance has metadata locks and InnoDB locks. You can also check the recent deadlock analysis and full deadlock analysis.

For details, see **Lock Analysis**.

Introduction

Metadata Locks

- Metadata locks are used for tables to prevent conflicting DDL and DML operations from being executed concurrently on these tables. Executing DDL statements on a table generates metadata write locks. If there is a metadata lock, all subsequent SELECT, DML, and DDL operations on the table will be blocked, causing a connection backlog.
- Metadata locks are displayed in real time. You can quickly identify problems and terminate the sessions with metadata locks to restore blocked operations.
- DML locks are not included. You can view and analyze them on the **InnoDB Locks** page.
- A maximum of 1,000 records can be displayed.

InnoDB Locks

- InnoDB lock waits generated before DML operations are displayed in real time. You can quickly locate the session waits and any blocks that happened when multiple sessions update the same piece of data at the same time, and can terminate the source session that holds locks to restore blocked operations.
- DDL locks, also called metadata locks, are not included. You can view and analyze them on the **Metadata Locks** page.
- To view lock information of RDS for MySQL 8.0 instances, set **performance_schema** to **ON**. You can run the **SHOW GLOBAL VARIABLES LIKE "performance_schema"** command or refer to **Modifying Parameters of an RDS for MySQL Instance** to check the **performance_schema** settings.

Deadlock Analysis

- DBA Assistant analyzes the latest deadlock log returned by **SHOW ENGINE INNODB STATUS**. If there are multiple deadlocks, only the latest one is analyzed.
- The **innodb_deadlock_detect** parameter must be set to **ON** (only for RDS for MySQL 5.7).

Full Deadlock Analysis

- DBA Assistant analyzes error logs at regular intervals, parses deadlock information, and performs comprehensive deadlock analysis.
- Dependent parameters:
  - The **innodb_deadlock_detect** parameter must be set to **ON** (only for RDS for MySQL 5.7).
  - The **innodb_print_all_deadlocks** parameter must be set to **ON** and the **log_error_verbosity** parameter must be set to **3** (only for other versions than 5.7).
- Up to 10,000 records can be displayed.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Locks & Transactions** tab and enter the administrator password to log in to the instance.

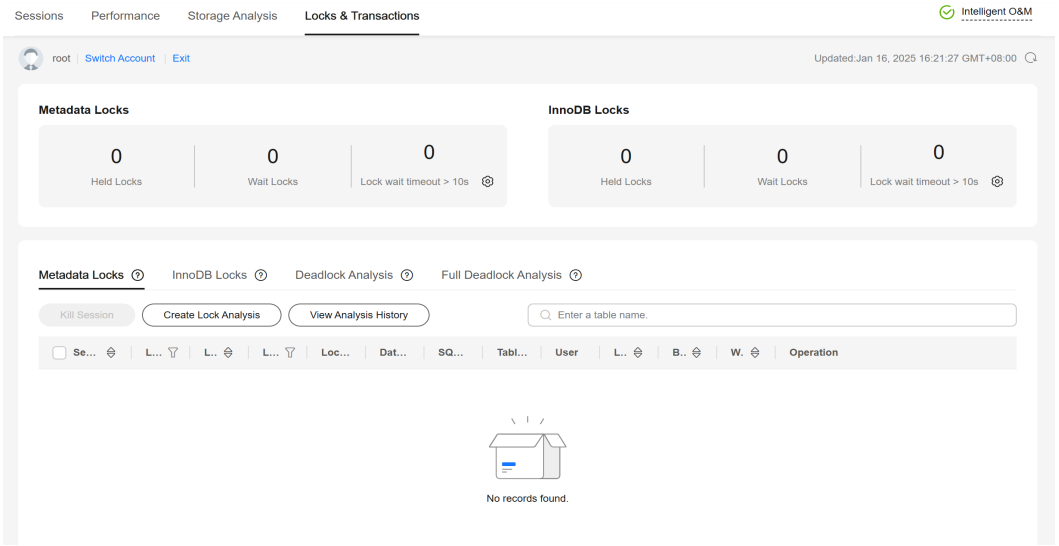
Step 7 On the **Locks & Transactions** tab page, perform the following operations:

- Click the **Metadata Locks** tab, create a lock analysis task and check whether the instance has metadata locks.

NOTE

By default, locks whose wait time is longer than 10s are displayed, but you can change this time if needed.

Figure 1-169 Metadata Locks

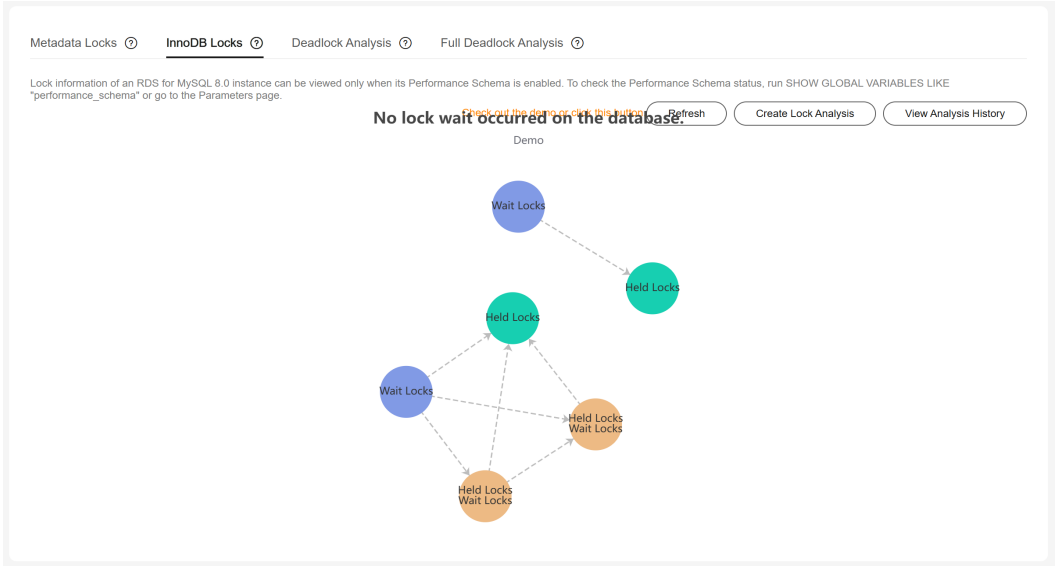


- Click the **InnoDB Locks** tab, create a lock analysis task and check whether the instance has InnoDB locks.

  NOTE

  By default, locks whose wait time is longer than 10s are displayed, but you can change this time if needed.

Figure 1-170 InnoDB Locks

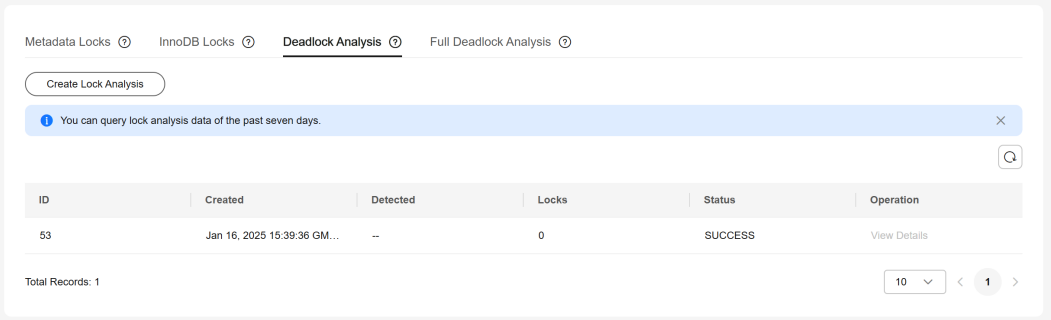


- Click the **Deadlock Analysis** tab and create a lock analysis task. DBA Assistant analyzes the latest deadlock log returned by **SHOW ENGINE INNODB STATUS**. If there are multiple deadlocks, only the latest one is analyzed.

NOTE

You can only query lock analysis data of the past seven days.

Figure 1-171 Deadlock Analysis

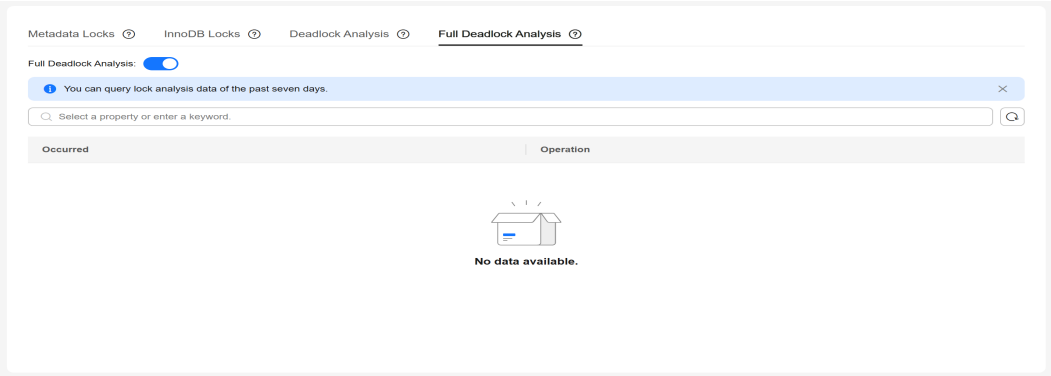


- Click the **Full Deadlock Analysis** tab and enable **Full Deadlock Analysis**. DBA Assistant regularly examines error logs, extracts deadlock details from them, and conducts a full deadlock analysis.

NOTE

You can only query lock analysis data of the past seven days.

Figure 1-172 Full Deadlock Analysis



----End

1.13.3.4 Managing Historical Transactions

Introduction

This function is used to analyze and discover large transactions and the transactions that have been there for a long time without being committed in the database.

Data on historical transactions comes from the output of the **SHOW ENGINE INNODB STATUS** command. It shows the snapshot of historical transactions and is collected every 5 minutes.

Constraints

- The snapshot of historical transactions only lists the transactions that are currently being executed. It does not include those that started and committed during a collection interval.
- To collect historical transactions, enable **Collect Slow Query Logs** or **Collect All SQL Statements** first.
- A maximum of 10,000 records generated in the past 7 days can be displayed.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

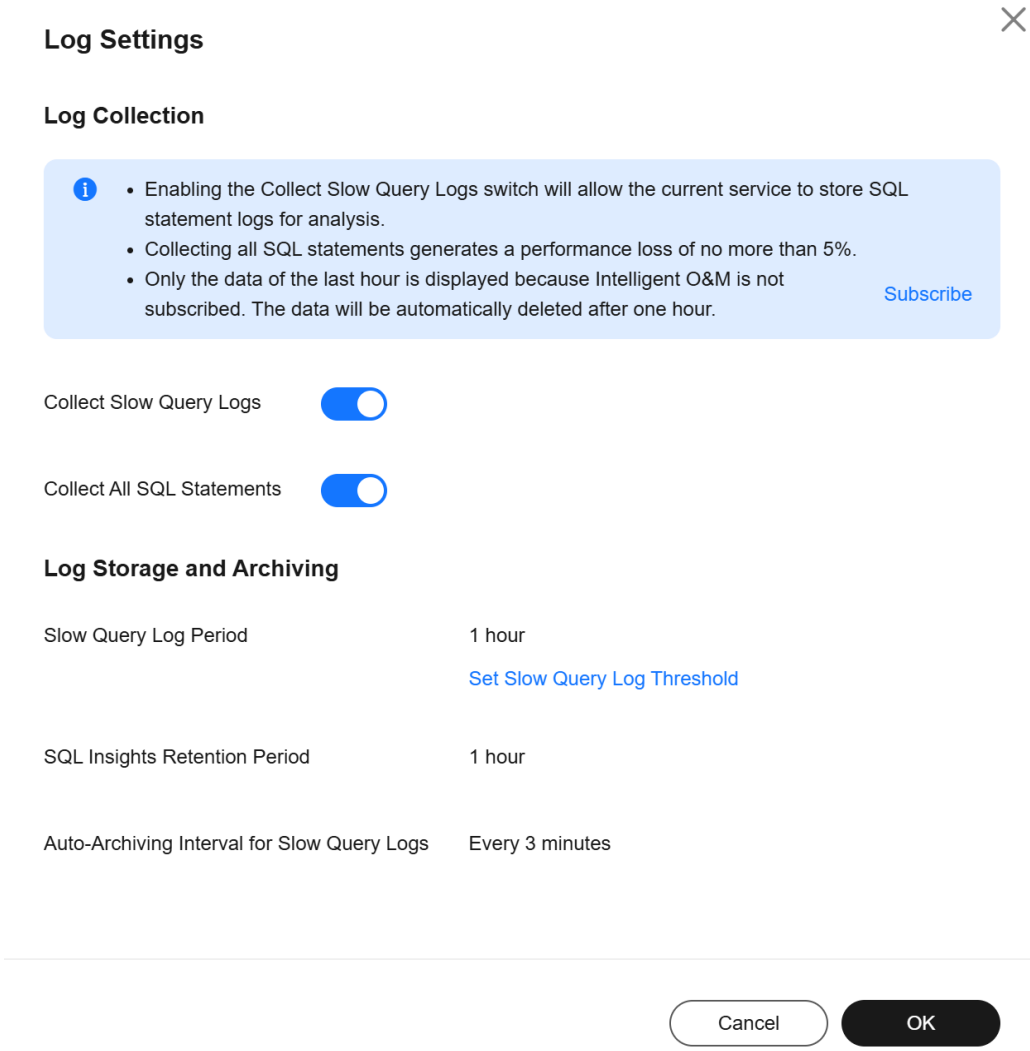
Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Historical Transactions** tab.

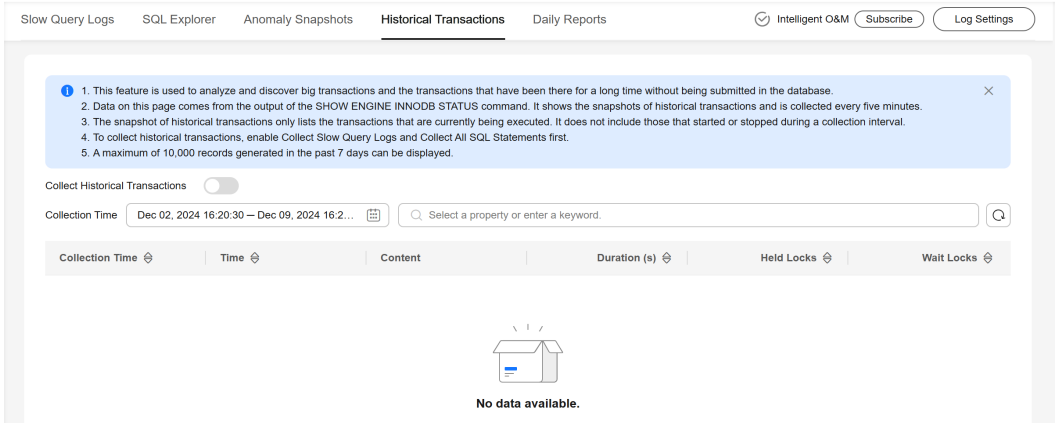
Step 7 In the upper part of the page, click **Log Settings**, enable **Collect Slow Query Logs** or **Collect All SQL Statements**, and click **OK**.

Figure 1-173 Log Settings



Step 8 Enable **Collect Historical Transactions** and view historical transactions.

Figure 1-174 Collect Historical Transactions



----End

1.13.3.5 Daily Reports

Scenarios

You can start a diagnosis for your DB instance and subscribe to diagnosis reports.

- **Starting a Diagnosis**: You can perform an overall health diagnosis on your instance and view details of the current and historical diagnosis reports.
- **Subscribing to Diagnosis Reports**: Simple Message Notification (SMN) can send diagnosis exception reports to the preset email address so that you can learn about the overall health status of your instance in real time.

Billing

When you use SMN, only pay for what you use. There are no minimum fees. For details, see **Billing**.

Starting a Diagnosis

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

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

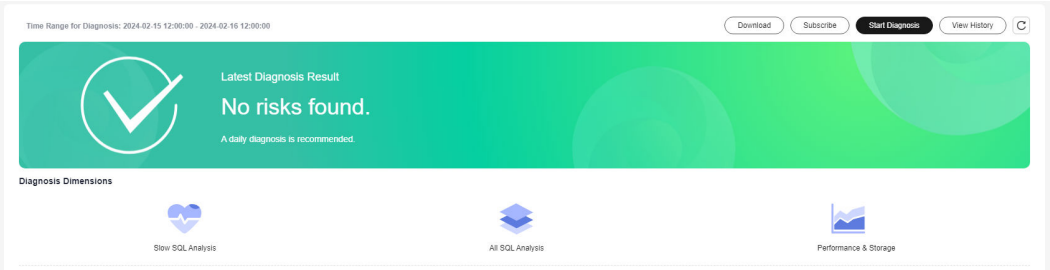
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

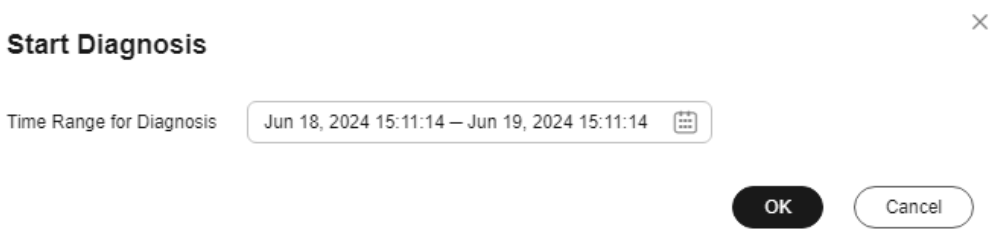
Step 6 Click the **Daily Reports** tab.

Figure 1-175 Daily Reports



Step 7 Click **Start Diagnosis**. Select a time range for the diagnosis. The time span is within one day.

Figure 1-176 Start Diagnosis



NOTE

If you want to be notified of risks through email, see **Subscribing to Diagnosis Reports**.

Step 8 In the **Diagnosis Dimensions** area, click **Slow SQL Analysis**, **All SQL Analysis**, or **Performance & Storage** to view diagnosis report details.

Figure 1-177 Diagnosis Dimensions

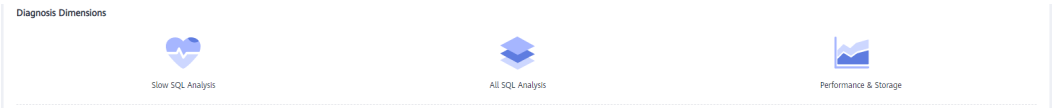
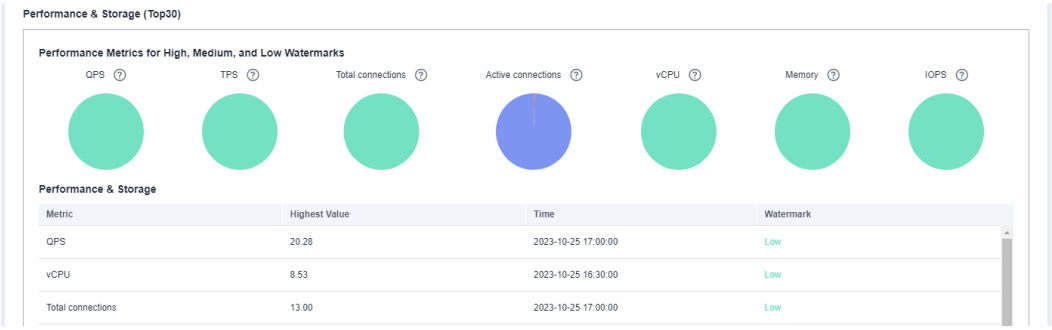


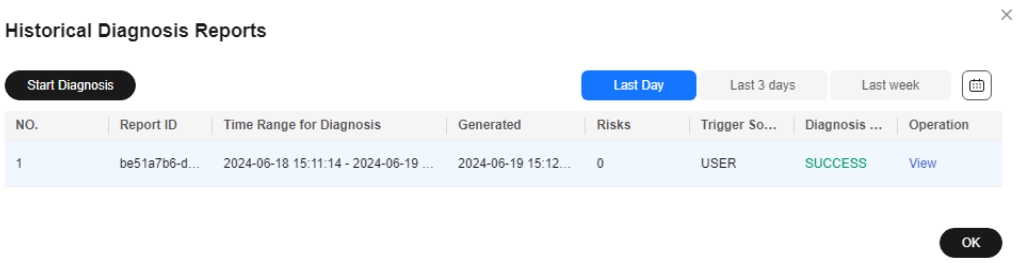
Figure 1-178 Performance & Storage



Step 9 You can also view historical diagnosis reports or download a report to your local PC.

- To view historical diagnosis reports, click **View History** in the upper right corner of the page.
- To download a report to your local PC, click **Download** in the upper right corner of the page.

Figure 1-179 Historical Diagnosis Reports



----End

Subscribing to Diagnosis Reports

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 3 Click the **Daily Reports** tab.

Step 4 In the upper right corner of the page, click **Subscribe** and set subscription parameters. For details about the parameters, see **Table 1-72**.

Figure 1-180 Subscribe to Daily Report

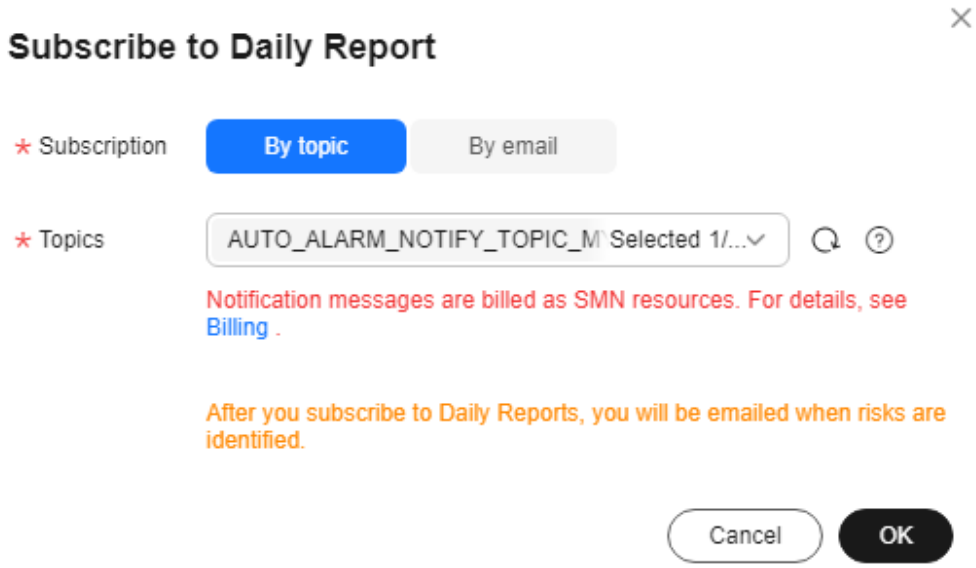


Table 1-72 Subscription parameters

Parameter	Description
Subscription	Select **By topic** or **By email**.
Topics	A topic is used to publish messages and subscribe to notifications. It serves as a message transmission channel between publishers and subscribers. If there are no topics you want to select, **create one**. After a topic is created, click **Add Subscription** in the **Operation** column of the topic. In the displayed dialog box, specify a protocol (only **Email** is supported) and an endpoint.
Email Addresses	If you select **By email** for **Subscription**, you need to specify **Email Addresses**. An email will be sent to the specified email address only when risks are identified after a diagnosis is performed. You can enter up to 15 email addresses and separate each email address with a semicolon (;).

Step 5 Click **OK**.

Step 6 If you want to unsubscribe from diagnosis reports, click **Unsubscribe** in the upper right corner of the page. In the displayed dialog box, confirm the information and click **OK**.

----End

1.13.3.6 Managing Anomaly Snapshots

Scenarios

This function intelligently detects instance anomalies and records information about session, lock, and transaction snapshots to facilitate subsequent fault locating.

Constraints

- Enabling anomaly collection will cause about 5% of instance performance loss.
- For an HA read replica with anomaly snapshots enabled, if the primary read replica node fails and workloads are automatically switched over to the standby read replica node, anomaly snapshots will not be enabled for the new primary read replica node.
- Each anomaly snapshot can be retained for a maximum of seven days. A maximum of 10 anomaly snapshots can be retained for each node at the same time.
- Anomaly snapshots record long-running transactions.

Enabling Anomaly Collection

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

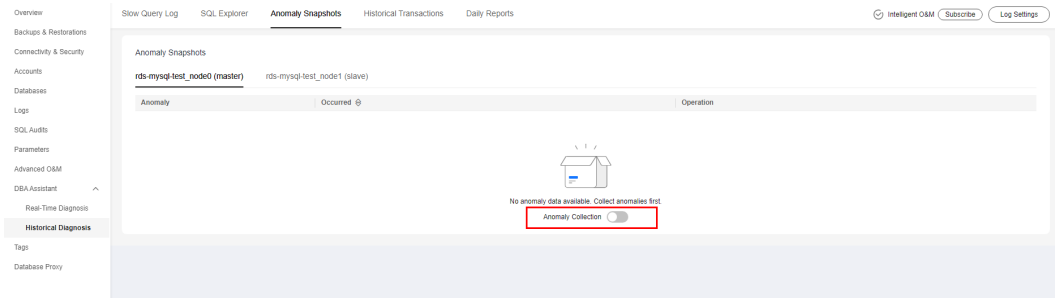
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Anomaly Snapshots** tab.

Step 7 On the displayed page, toggle on the **Anomaly Collection** switch.

Figure 1-181 Anomaly Snapshots



----End

Viewing Anomaly Snapshots

Step 1 Click the **Anomaly Snapshots** tab.

Step 2 On the displayed page, view session snapshots, metadata lock snapshots, InnoDB lock snapshots, and transaction snapshots of the DB instance.

- To view anomaly causes, click **Diagnosis Details** in the **Operation** column.
- To view details about slow SQL statements, click **Slow SQL** in the **Operation** column. For details, see **Viewing Slow Query Logs of a DB Instance**.

----End

1.13.4 SQL Analysis

1.13.4.1 Viewing Slow Query Logs of a DB Instance

Scenarios

Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, host, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

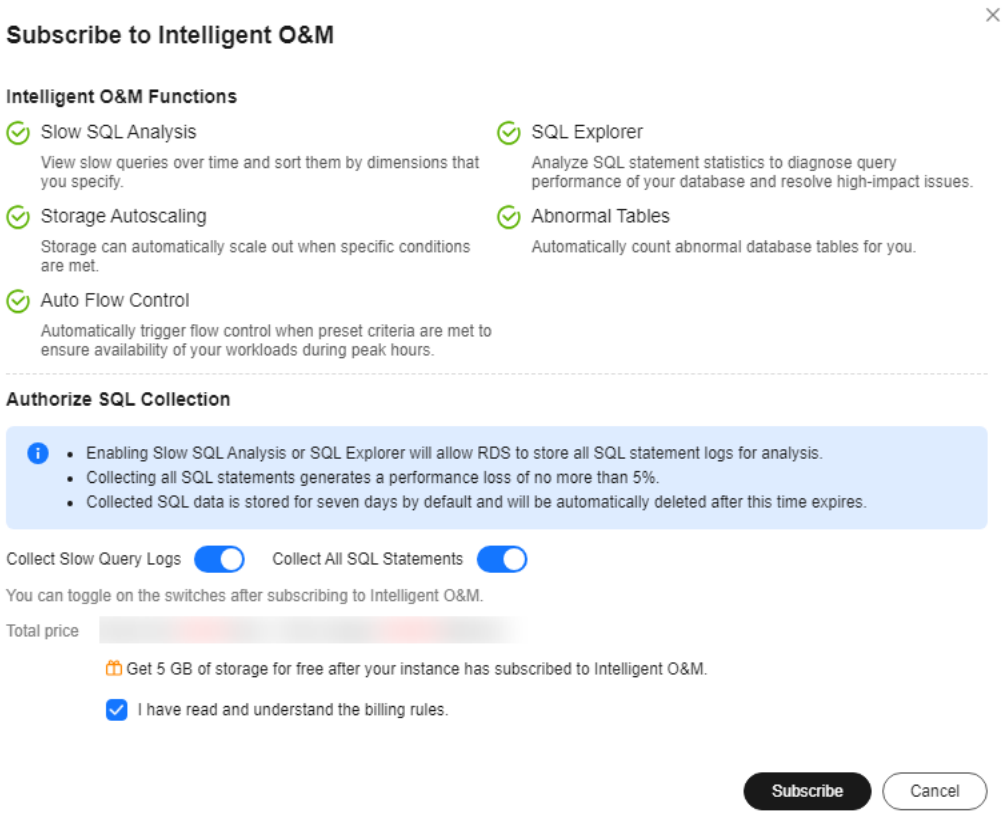
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Slow Query Log** tab.

NOTE

Slow SQL Analysis needs to be purchased separately. To use this function, subscribe to Intelligent O&M first.

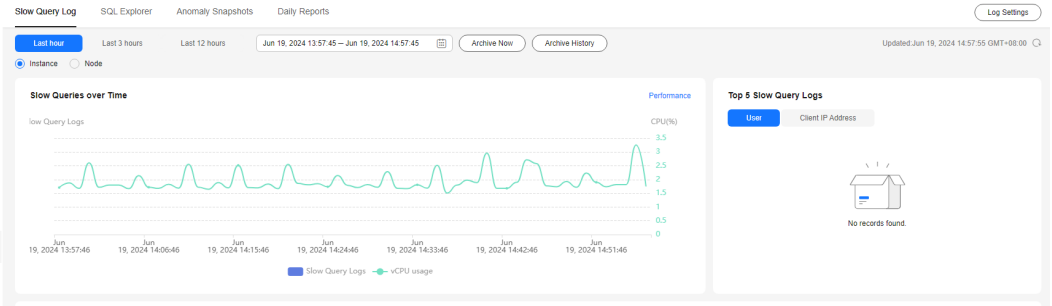
Step 7 Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.



Step 8 After subscribing to Intelligent O&M, view slow queries over time of your instance.

Step 9 You can view slow queries over time and you can see the slow log archive history for the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day).

Figure 1-182 Viewing slow queries over time



Step 10 View slow query log details and template statistics.

- To export slow query log information, click **Export**.
- To view log export history, click **View Export List**.

----End

1.13.4.2 Viewing Top SQL Statements of a DB Instance

Scenarios

Top SQL shows the SQL queries that have been contributing the most to DB load. You can sort them by multiple dimensions.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

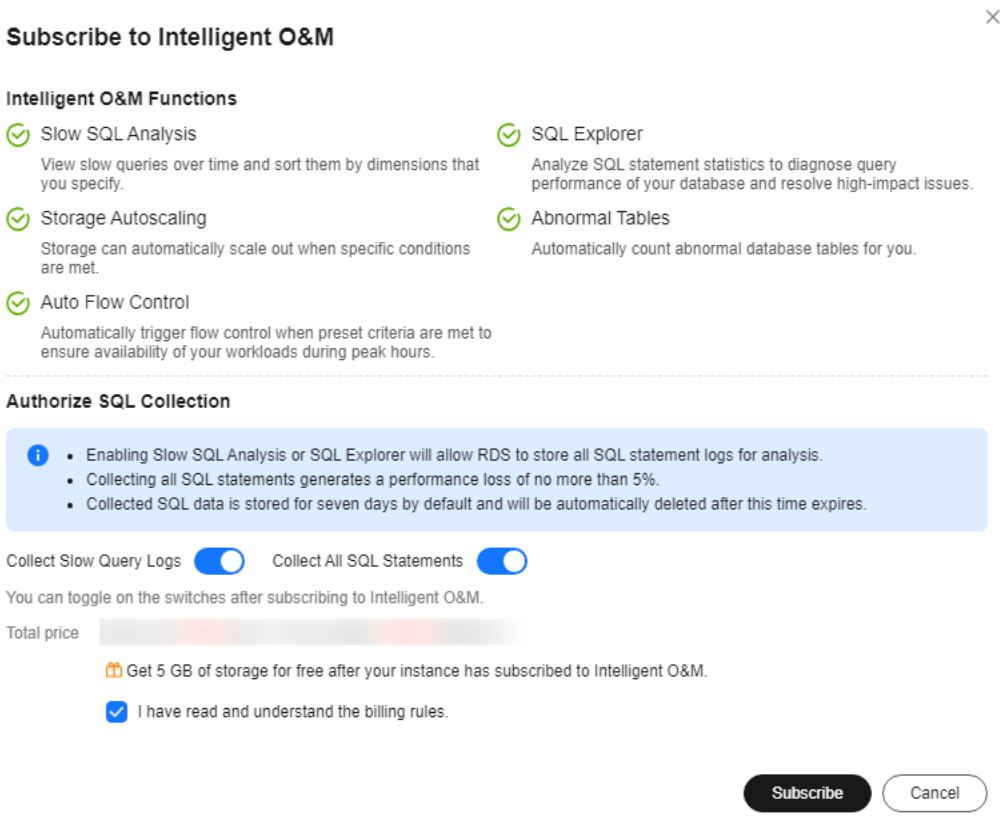
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Top SQL**.

NOTE

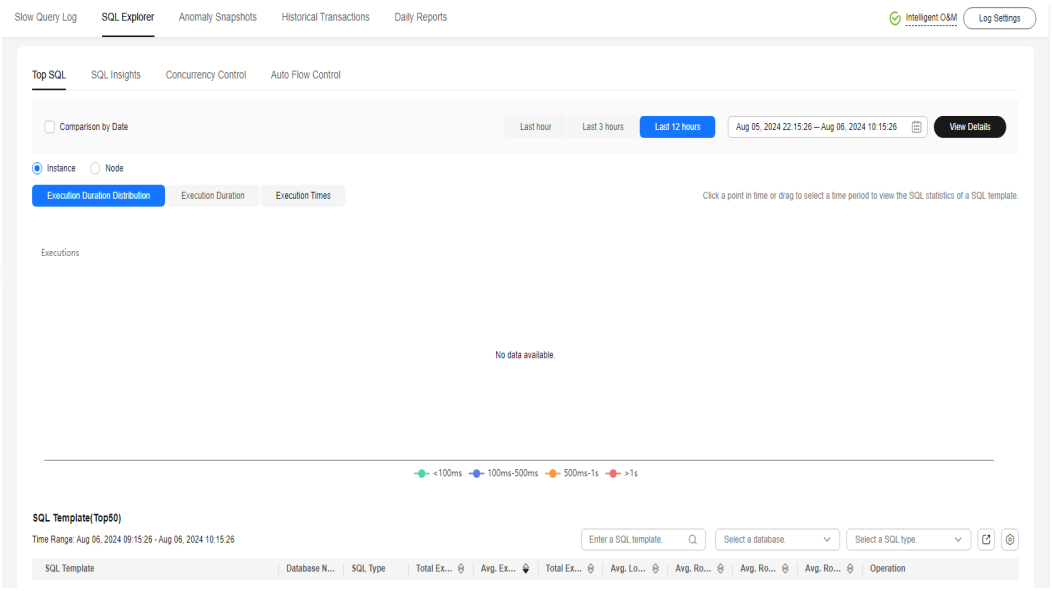
Top SQL needs to be purchased separately. To use this function, subscribe to Intelligent O&M first.

Step 7 Click **Subscribe**. In the displayed dialog box, learn about Intelligent O&M functions and pricing.



Step 8 After subscribing to Intelligent O&M, view top SQL statements of your instance.

Step 9 You can view execution durations of the top SQL statements in the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day).



----End

1.13.4.3 Creating a SQL Insights Task

Scenarios

SQL Insights allows you to not only query all executed SQL statements, but also analyze and search for the tables that are accessed and updated most frequently, and the SQL statements that have the longest lock wait, helping you quickly identify exceptions.

Constraints

- You need to enable **Collect All SQL Statements** before using SQL Insights.
- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected anymore and the collected SQL data will be deleted.
- If there is a buffer overrun, some data cannot be recorded.
- Any SQL statement that exceeds 4,096 bytes is discarded by default.

  In RDS for MySQL 5.7.33.3 and later minor versions, you can set the **rds_sql_tracer_reserve_big_records** parameter to **ON** (which indicates that SQL statements containing more than 4,096 bytes are stilled recorded) on the **Parameters** page to remove this constraint. For details, see **Modifying Parameters of an RDS for MySQL Instance**. RDS for MySQL 5.6 and 8.0 do not support this parameter.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click **SQL Explorer** and then **SQL Insights**.

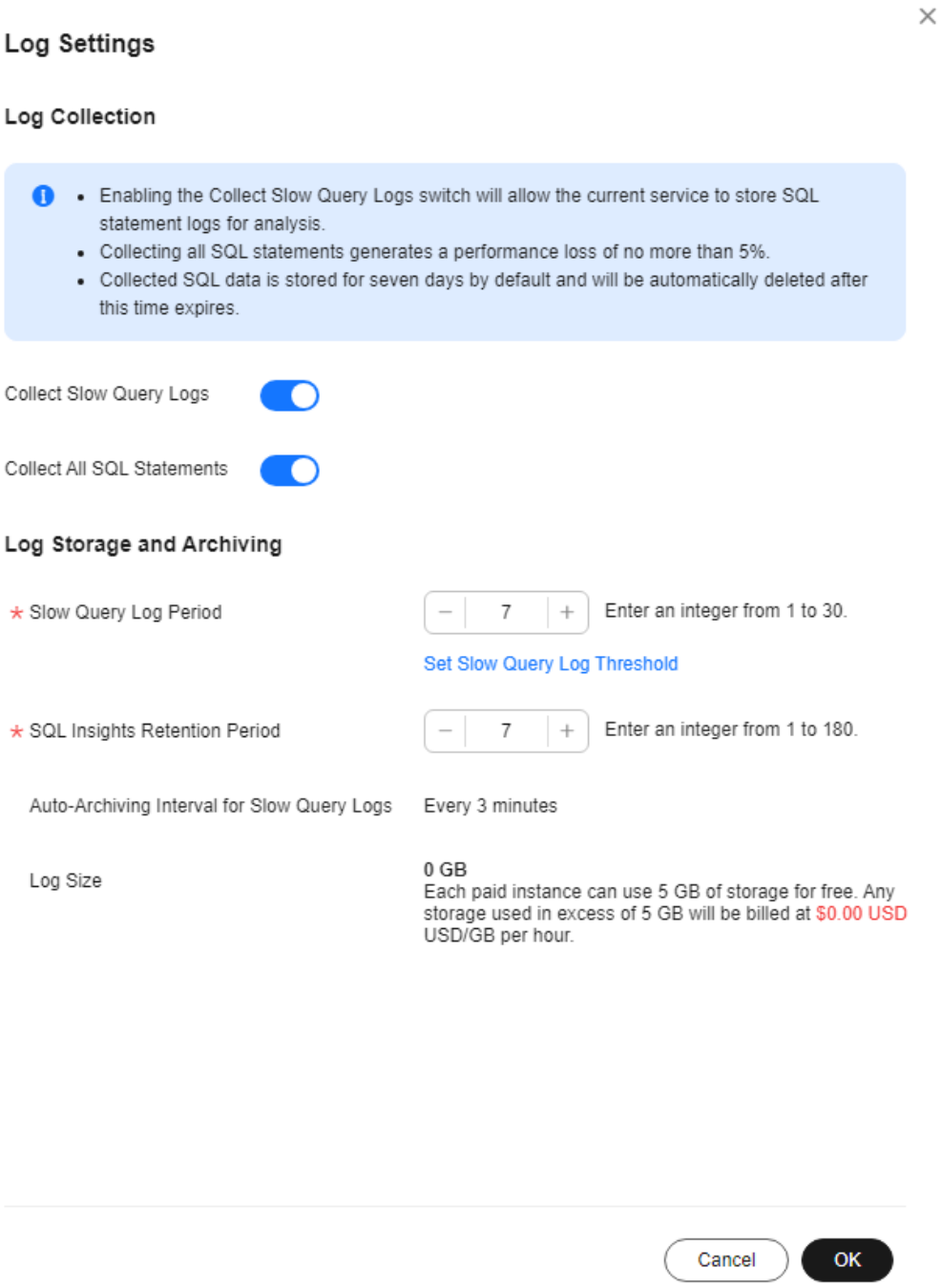
Step 7 Click next to **Collect All SQL Statements**.

To disable this function, click **Log Settings** in the upper right corner, toggle off the **Collect All SQL Statements** switch, and click **OK**.

NOTE

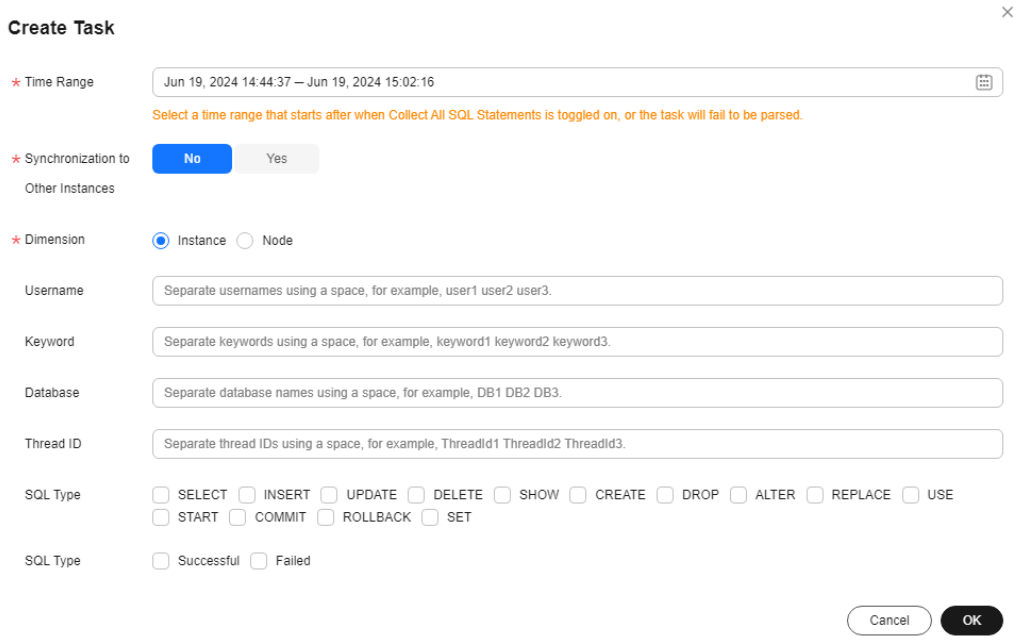
Collecting all SQL statements generates a performance loss of no more than 5%.

Figure 1-183 Log settings



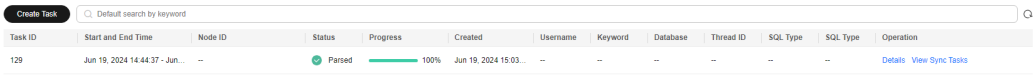
Step 8 Click **Create Task**. In the displayed dialog box, specify **Time Range**, **Dimension**, and other configuration items, and click **OK**.

Figure 1-184 Creating a SQL Insights task



Step 9 In the task list, click **Details** in the **Operation** column to view task details.

Figure 1-185 Viewing task details



----End

1.13.4.4 Creating a Concurrency Control Rule

Scenarios

You can create rules to control concurrent execution of SQL statements by specifying SQL type, keywords, and maximum concurrency. To maintain better performance at high concurrency, SQL statements that meet the specified SQL type and keyword and exceed the maximum concurrency will not be executed.

High SQL concurrency can be caused by the following factors:

- A sharp increase in requests: Concurrent SQL statements of a certain type surge due to cache penetration and abnormal calls.
- Stacked slow queries: If a large number of SQL statements without indexes are called, many slow SQL statements will be generated, affecting services.

Supported Versions

Concurrency Control is available to the RDS for MySQL versions listed in **Table 1-73**.

Table 1-73 Supported versions

Major Version	Minor Version (Primary Instance)	Minor Version (Read Replica)	Setting Rules for Read Replicas Separately
5.6	≥ 5.6.50.3	≥ 5.6.51.6	Not supported
5.7	≥ 5.7.31.4	≥ 5.7.37.1	≥ 5.7.38.221000
8.0	≥ 8.0.25.1	≥ 8.0.25.1	Not supported

In some versions, concurrency control rules are not applied to requests sent by user **root**. For details, see **Table 1-74**.

NOTE

If your instance running 5.7.43.231000 or later or 8.0.28.231000 or later, concurrency control rules will be applied to user **root**. To disable concurrency control for user **root**, **submit a service ticket**.

Table 1-74 Versions in which requests from **root** are not limited by concurrency control rules

Major Version	Minor Version (Primary Instance)
5.6	≥ 5.6.51.4
5.7	5.7.33.1 ≤ Version < 5.7.43.231000
8.0	8.0.25.1 ≤ Version < 8.0.28.231000

To achieve better performance at high concurrency, you are advised to upgrade your DB instance to the latest minor version. For details about how to upgrade a minor version, see **Upgrading a Minor Version**.

Constraints

- A maximum of 100 concurrency control rules can be configured.
- Only SELECT, UPDATE, DELETE, and INSERT statements are supported for concurrency control.
- INSERT statements are only supported for RDS for MySQL 5.7 (5.7.44.240100 or later) and 8.0 (8.0.32.240100 or later) for concurrency control. To use this function, contact customer service to apply for required permissions.
- If a SQL statement matches multiple concurrency control rules, only the most recently added rule is applied.
- SQL statements that have been executed before a concurrency control rule is added are not counted.
- If the replication delay is too long, adding or deleting a concurrency control rule for a read replica does not take effect immediately.

- Concurrency control rules are not applied to system tables.
- Concurrency control rules are not applied to SQL statements not used for data query, such as **select sleep(*****);**.
- Concurrency control rules are not applied to stored procedures, triggers, or functions.
- You can run the following SQL statement through DAS to view the execution of concurrency control rules: **select * from information_schema.rds_sql_filter_info;**
- Too many concurrency control rules affect the database performance. Delete unnecessary rules after using them.
- Concurrency control rules are not applied to system databases.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Concurrency Control**.

Step 7 Toggle on the concurrency control switch .

NOTE

Concurrency control rules take effect only after concurrency control is enabled.

Step 8 Click **Add Rule**. Configure the parameters listed in **Table 1-75**.

Figure 1-186 Adding a rule (entering keywords)

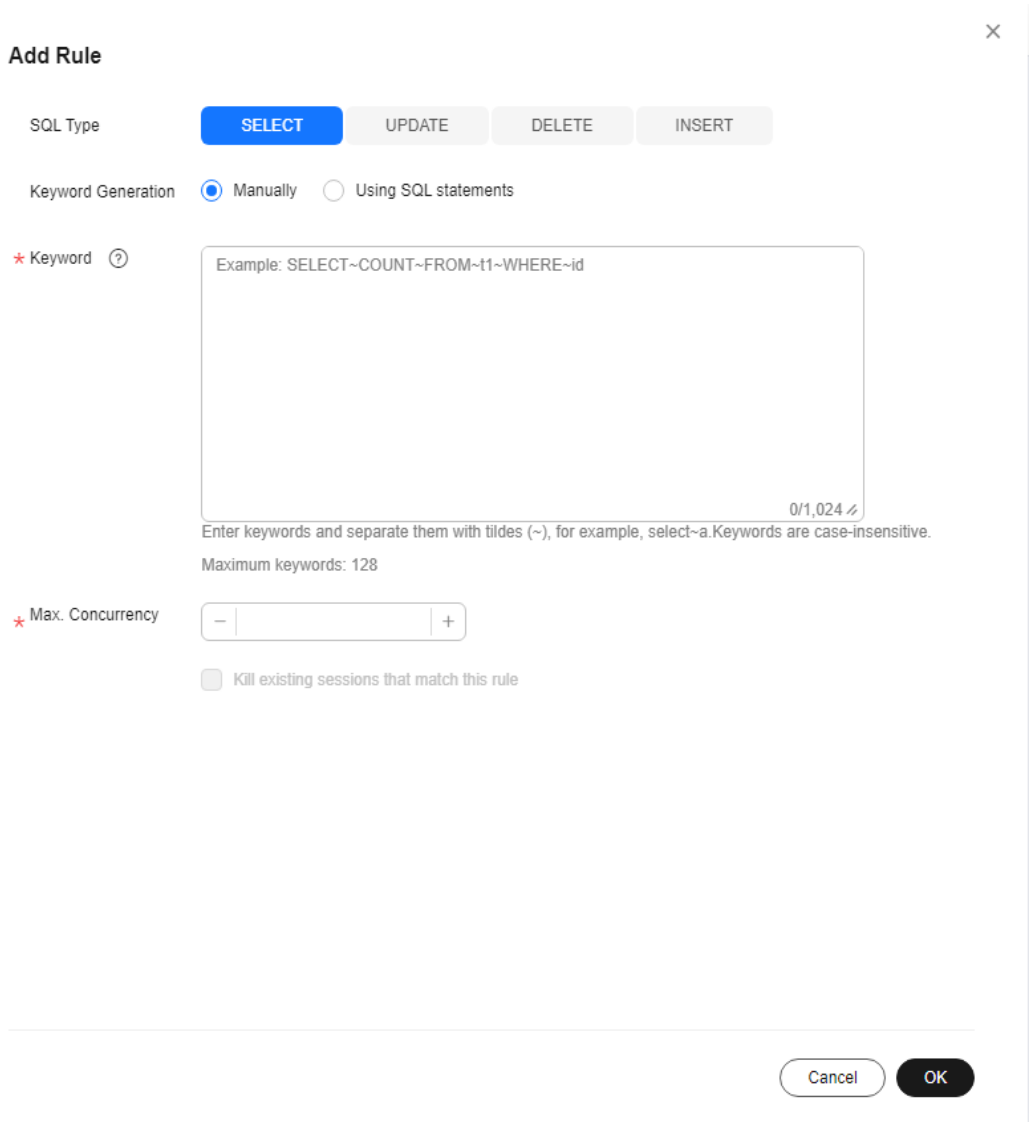


Figure 1-187 Adding a rule (generating keywords from a SQL statement)

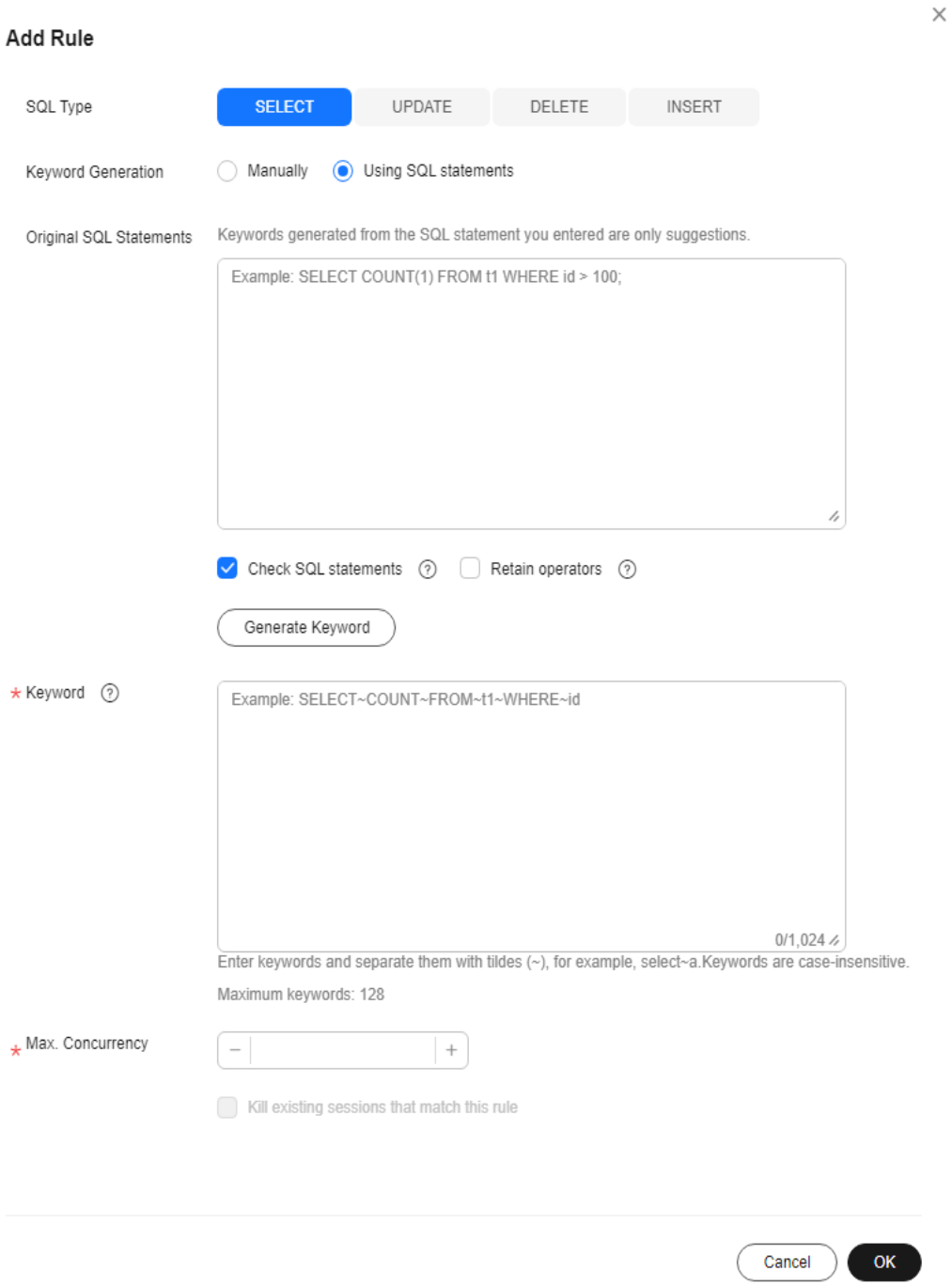


Table 1-75 Parameter description

Parameter	Description
SQL Type	There are four options: **SELECT**, **UPDATE**, **DELETE**, and **INSERT**.

Parameter	Description
Keyword	A maximum of 128 keywords (case-insensitive) are supported. You can specify keywords in either of the following ways: • **Manually**: Take **select~a** as an example. **select** and **a** are two keywords contained in a concurrency control rule. The keywords are separated by a tilde (~). In this example, the rule restricts the execution of only the SQL statements containing keywords **select** and **a**. • **Using SQL statements**: You can enter a SQL statement and then click **Generate Keyword**. The generated keywords are for reference only. Exercise caution when using them. SQL statements match the keywords from first to last. For example, if one rule contains the keyword **a~and~b**, the statement ***** a>1 and b>2** can match the keyword, but ***** b>2 and a>1** cannot. Empty characters before and after each keyword will be ignored, for example, spaces, '\n', '\r', and '\t'.
Max. Concurrency	If the number of concurrent SQL statements matching the keyword exceeds this limit, the SQL statements will not be executed. The value ranges from 0 to 1,000,000,000.
Kill existing sessions that match this rule	If this option is selected, all sessions generated by users subject to this concurrency control rule will be killed. For details about the versions where user **root** is not subject to concurrency control rules, see **Table 1-74**.

Step 9 Confirm the settings and click **OK**.

----End

Follow-up Operations

To delete a concurrency control rule, locate it in the rule list and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

Figure 1-188 Deleting a rule

1.13.4.5 Configuring Auto Flow Control

Scenarios

Auto Flow Control allows you to set criteria, such as the vCPU threshold and maximum number of active connections. When the criteria are met, the system

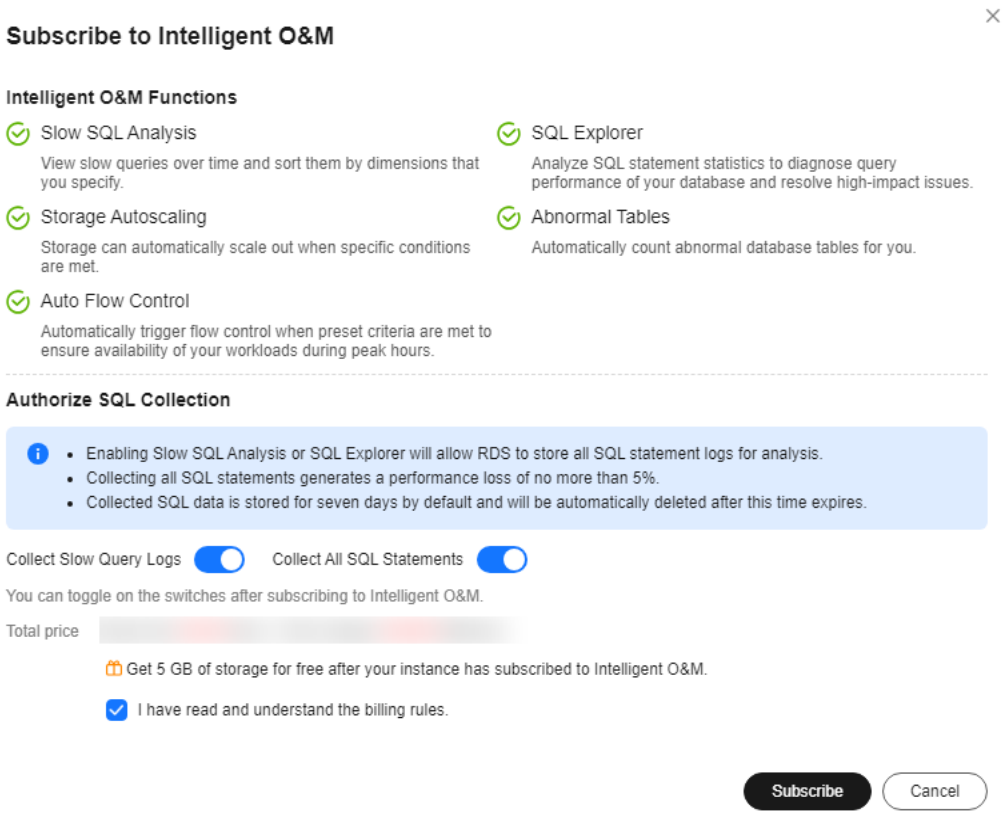
automatically performs flow control on sessions to ensure the availability of core services in scenarios such as high burst traffic and abnormal reads/writes.

Prerequisites

If you use auto flow control for the first time, subscribe to Intelligent O&M first.

1. In the **Auto Flow Control** area, click **Subscribe**. In the displayed dialog box, you can learn about the value-added functions and billing rules.
2. (Optional) If you enable **Collect Slow Query Logs** and **Collect All SQL Statements**, slow query logs and all SQL statements are collected and analyzed from multiple angles. For details, see **Viewing Slow Query Logs of a DB Instance** and **Viewing Top SQL Statements of a DB Instance**.

Figure 1-189 Subscribing to Intelligent O&M



Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

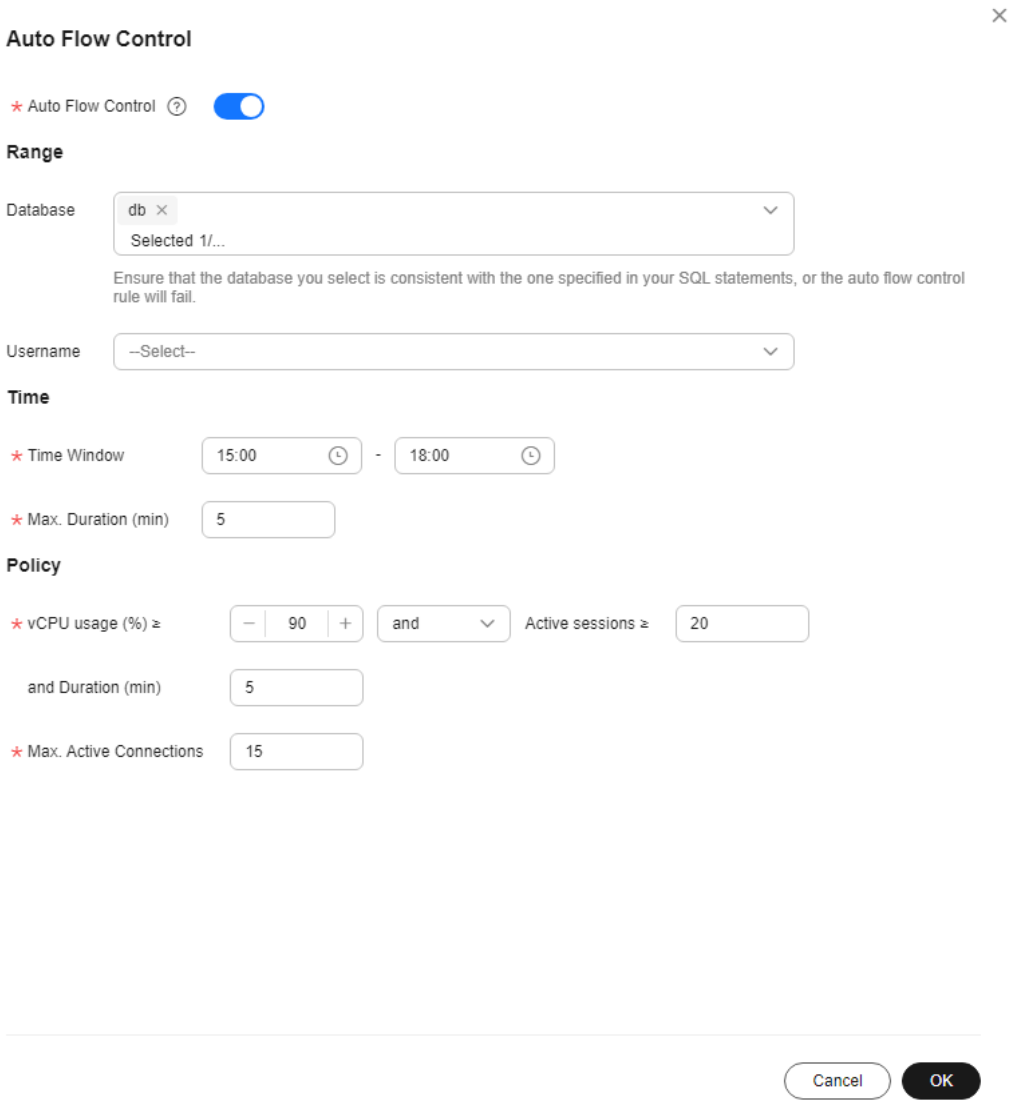
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click **SQL Explorer** and then **Auto Flow Control**.

Step 7 Click **Auto Flow Control**.

Step 8 Toggle on  and configure required parameters. For details about the parameters, see **Table 1-76**.

Figure 1-190 Example



Example for setting Auto Flow Control parameters

Set **Time Window** to **15:00-18:00**, **Max. Duration** to **Last 5 minutes**, **vCPU usage ≥** to **90%**, **Active sessions ≥** to **20**, and **and Duration (min)** to **5**. When all the criteria are met, Auto Flow Control is triggered. If your vCPU usage or number of active sessions falls below the threshold during the time window, flow control ends.

Table 1-76 Parameter description

Parameter	Description
Database	The name of the database for which auto flow control needs to be enabled. Ensure that the database you select is consistent with the one specified in your SQL statements, or the auto flow control policy will fail. If the database name is not specified, auto flow control is applied to all databases by default.
Username	The name of the user that auto flow control is applied.
Time Window	The time when flow control is applied. Auto flow control can be triggered only once within the time window.
Max. Duration	Maximum length of time that SQL statements matching the auto flow control policy can be throttled within the time window.
vCPU usage	vCPU usage threshold for the instance (≥ 70%). You also need to specify the relationship between the vCPU usage and active sessions. Their relationship can be **and** or **or**.
Active sessions	Threshold for active sessions. Value range: 1 to 5000
Duration (min)	How long vCPU usage and active sessions exceed the specified values. The duration can range from 2 minutes to 60 minutes. For example, if you set **vCPU usage ≥** to **90%**, **Active sessions ≥** to **1000**, and **Duration (min)** to **30**, auto flow control will be triggered only when the vCPU usage and active sessions exceed 90% and 1,000 for 30 minutes.
Max. Active Connections	Maximum number of active connections allowed. The value can range from 1 to 5,000. Auto flow control stops when the number of active connections drops to this limit.

Step 9 Click **OK**.

Step 10 You will see a record generated on the page every time Auto Flow Control is triggered. You can see historical details, too.

----End

Follow-up Operations

After auto flow control is enabled, you can determine whether to kill sessions based on service requirements. To learn how to kill sessions, see **Managing Real-Time Sessions**.

1.13.5 Common Performance Problems

1.13.5.1 High CPU Usage of RDS for MySQL Instances

If the CPU usage of your RDS for MySQL instance is high or close to 100%, database performance deteriorates. For example, data read/write becomes slow, connecting to the instance takes a longer time, or errors are reported when you are trying to delete data.

NOTICE

The following functions of Data Admin Service (DAS) are only available to the accounts that are using them, from November 25, 2021, 00:00 GMT+08:00: SQL tuning, table structure comparison and synchronization, data tracking and rollback, data generator, and DBA intelligent O&M in Development Tool, as well as space, intelligent parameter recommendation, historical transaction, and binlog parsing functions in Intelligent O&M.

- For DB instances created after November 25, 2021, 00:00 GMT+08:00, **Solution 1** is recommended.
- For DB instances created before November 25, 2021, 00:00 GMT+08:00, **Solution 2** is recommended.

You are advised to **enable SQL audit** in advance so that you can view SQL execution records in audit logs to locate the fault when the CPU usage is high.

Solution 1

Analyze slow SQL logs and CPU usage to locate slow queries and then optimize them.

1. View the slow SQL logs to check for slowly executed SQL queries and view their performance characteristics (if any) to locate the cause.

   For details on viewing RDS for MySQL logs, see **Slow Query Log**.
2. View the CPU usage metric of your DB instance.

   For details, see **Performance**.
3. Create read replicas to reduce read pressure from primary DB instances.
4. Add indexes for associated fields in multi-table association queries.
5. Do not use the SELECT statement to scan all tables. You can specify fields or add the WHERE condition.

Solution 2

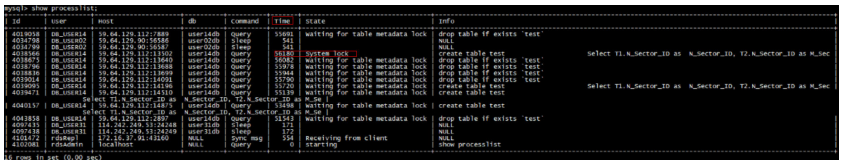
You can identify slow query statements and optimize them according to the suggestions provided by DAS to reduce the CPU usage.

Step 1 Connect to the RDS for MySQL DB instance.

You can connect to an instance through a private or public network. For details, see the *Relational Database Service Getting Started.*

Step 2 Run the following command to show the running threads and locate the queries that are slowly executed:

show full processlist

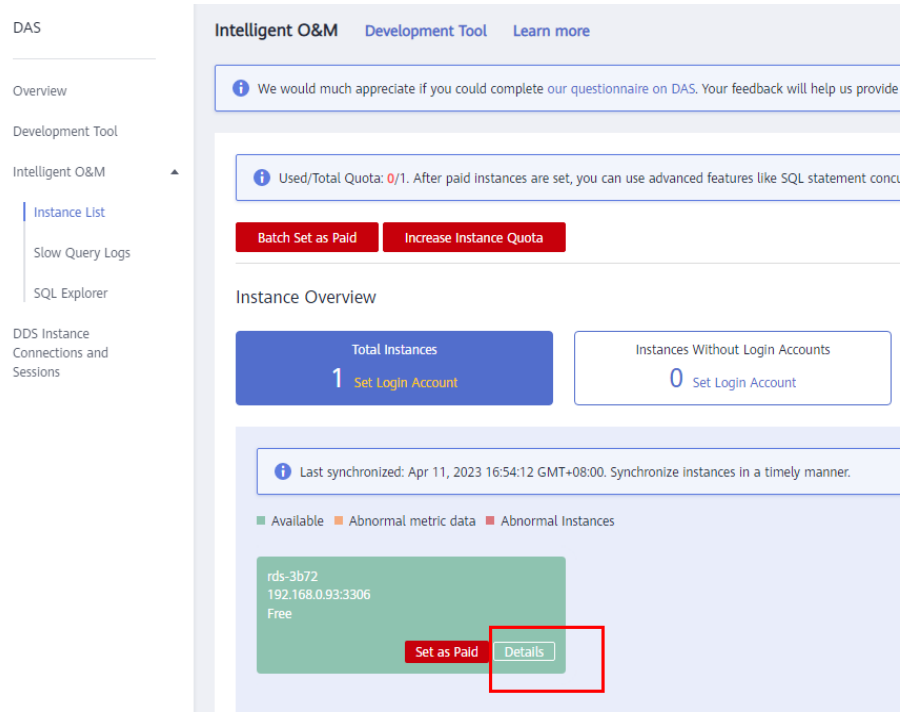


Check the values in the **Time** and **State** columns. As shown in the preceding figure, the long-running transaction ID is **4038566**.

Step 3 Use SQL diagnosis of DAS to identify SQL statements that are executed frequently, consume a large amount of resources, or take a long time to execute. You can optimize the statements according to the diagnosis suggestions to ensure the stability of the database performance.

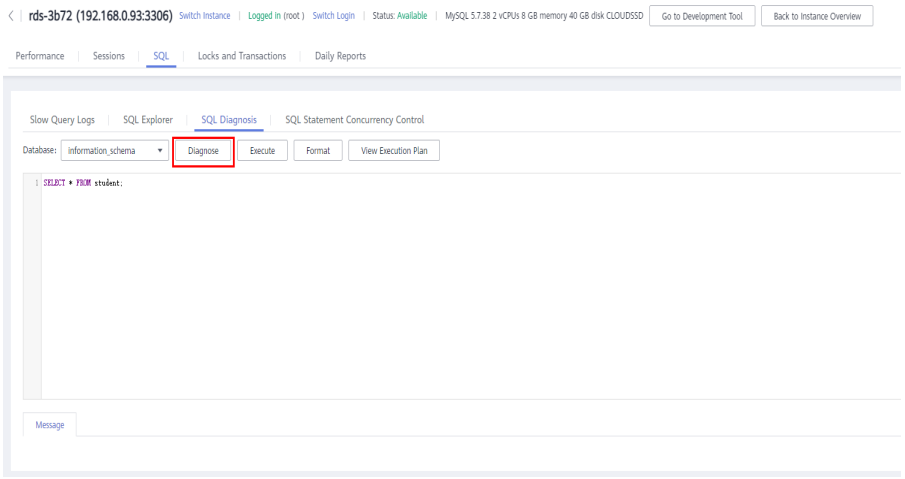
1. Log in to the DAS console.
2. In the navigation pane, choose **Intelligent O&M** > **Instance List**.
3. Click **Details** on the instance.

Figure 1-191 Instance list



4. Choose **SQL** > **SQL Diagnosis**.
5. Select a database, enter an SQL statement, and click **Diagnose**.

Figure 1-192 SQL diagnosis



6. View diagnosis details and obtain statement optimization suggestions.

Figure 1-193 Tuning details

NOTE

- Only the diagnosis of SELECT, INSERT, UPDATE, and DELETE statements is supported. An INSERT statement must contain a SELECT clause.
- SQL statements for querying system databases like **information_schema**, **performance_schema**, and **mysql** are not supported.
- SQL statements that use views are not supported.
- Using SQL diagnosis can obtain table structure and data distribution information (non-original). The obtained information is only used for logic diagnosis, but not stored on the DAS server.
- Obtaining table structure and data distribution information may cause additional load on the DB instance, but has little impact on its performance.
- Only the SQL diagnosis history is stored on the DAS server. You can delete it from the server permanently.

----End

1.13.5.2 High Memory Usage of RDS for MySQL Instances

For a DB instance storing mission-critical application data

Scale up your instance by referring to **Changing a DB Instance Class**.

For a DB instance not storing mission-critical application data

Check the memory usage of the local computer. If the memory usage curve is stable, no action is required.

For a DB instance storing mission-critical application data and configured with a large instance class

1. During off-peak hours, change the value of **performance_schema** to **OFF**. For RDS for MySQL 5.6 and earlier versions, you should reboot the instance for the change to take effect.
2. View the memory usage of your instance using DBA Assistant. For details, see **Viewing Performance Metrics of a DB Instance**.

   If the memory usage remains high, perform either of the following operations:

   - Scale up the instance class.
   - Change the value of **innodb_buffer_pool_size**. **Table 1-77** lists the recommended values for different memory specifications. The actual value ranges are displayed on the RDS console.

Table 1-77 Recommended values for different memory specifications

Memory (GB)	Recommended Value in Version 5.6	Recommended Value in Version 5.7	Recommended Value in Version 8.0
2	536,870,912 bytes (512 MB)	536,870,912 bytes (512 MB)	536,870,912 bytes (512 MB)
4	1,073,741,824 bytes (1 GB)	1,073,741,824 bytes (1 GB)	1,073,741,824 bytes (1 GB)
8	4,294,967,296 bytes (4 GB)	4,294,967,296 bytes (4 GB)	5,368,709,120 bytes (5 GB)
16	8,589,934,592 bytes (8 GB)	8,589,934,592 bytes (8 GB)	9,663,676,416 bytes (9 GB)
32	22,548,578,304 bytes (21 GB)	22,548,578,304 bytes (21 GB)	21,474,836,480 bytes (20 GB)
64	47,244,640,256 bytes (44 GB)	47,244,640,256 bytes (44 GB)	47,244,640,256 bytes (44 GB)
128	96,636,764,160 bytes (90 GB)	94,489,280,512 bytes (88 GB)	94,489,280,512 bytes (88 GB)

Memory (GB)	Recommended Value in Version 5.6	Recommended Value in Version 5.7	Recommended Value in Version 8.0
192	146,028,888,064 bytes (136 GB)	146,028,888,064 bytes (136 GB)	146,028,888,064 bytes (136 GB)
256	193,273,528,320 bytes (180 GB)	193,273,528,320 bytes (180 GB)	193,273,528,320 bytes (180 GB)
384	298,500,227,072 bytes (278 GB)	300,647,710,720 bytes (280 GB)	300,647,710,720 bytes (280 GB)
512	412,316,860,416 bytes (384 GB)	412,316,860,416 bytes (384 GB)	412,316,860,416 bytes (384 GB)
768	618,475,290,624 bytes (576 GB)	618,475,290,624 bytes (576 GB)	618,475,290,624 bytes (576 GB)
1024	824,633,720,832 bytes (768 GB)	824,633,720,832 bytes (768 GB)	824,633,720,832 bytes (768 GB)

NOTICE

- Change the value of **innodb_buffer_pool_size** as needed.
- MySQL has a dynamic memory balancing mechanism. If the memory usage is less than 90%, no action is required. You are advised to set the alarm threshold for memory usage to 90% or above.
- The memory used by the buffer pool will gradually increase to the value of **innodb_buffer_pool_size** as the database runs. You can check the memory usage of the buffer pool based on the metric **Buffer Pool Usage**.
- RDS for MySQL memory is allocated to the engine layer and server layer.
  - The memory allocated to the engine layer includes the InnoDB buffer pool, log buffer, and full text index cache. The InnoDB buffer pool is resident memory and accounts for a large proportion.

    The InnoDB buffer pool is a memory area that holds cached InnoDB data for tables, indexes, and other auxiliary buffers. You can use the **innodb_buffer_pool_size** parameter to define the buffer pool size.
  - The memory allocated to the server layer is occupied by the thread cache, binlog cache, sort buffer, read buffer, and join buffer. These caches and buffers are usually released when connections are closed.

  Such memory allocation keeps memory usage of a running RDS for MySQL instance at about 80%.

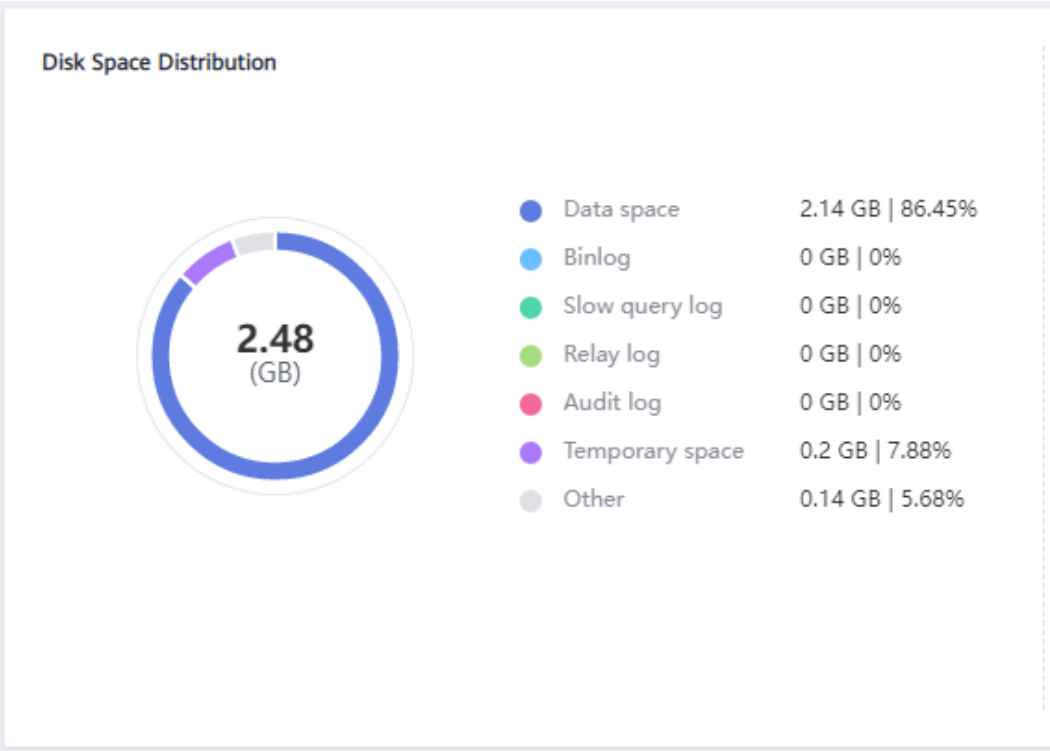
1.13.5.3 Full Storage of RDS for MySQL Instances

Symptom

There is not enough storage available for an RDS instance and the instance becomes read-only, so applications cannot write any data to the instance.

You can check which data or files occupy too much storage in the **Disk Space Distribution** area on the **Storage Analysis** page. For details, see **Storage Analysis**.

Figure 1-194 Disk space distribution



Causes

1. Increased workload data
2. Too much data being stored
3. Too many RDS for MySQL binlogs generated due to a large number of transactions and write operations
4. Too many temporary files generated due to a large number of sorting queries executed by applications

Solution

1. For insufficient storage caused by increased workload data, **scale up storage space**.

   If the original storage has reached the maximum, **upgrade the specifications** first.

For instances using cloud disks, you can configure **autoscaling** so that RDS can autoscale your storage when the storage usage reaches the specified threshold.

2. If too much data is stored, delete unnecessary historical data.
   a. If the instance becomes read-only, you need to contact customer service by **submitting a service ticket** to cancel the read-only status first.
   b. Check the top 50 databases and tables with large physical files and identify the historical table data that can be deleted. For details, see **Storage Analysis**.
   c. To clear up space, you can optimize tables with a high fragmentation rate during off-peak hours.

      To delete data of an entire table, run **DROP** or **TRUNCATE**. To delete part of table data, run **DELETE** and **OPTIMIZE TABLE**.
3. If binlog files occupy too much space, **clear local binlogs**.
4. If temporary files generated by sorting queries occupy too much storage space, optimize your SQL statements.

   You can query **slow query logs** and **top SQL statements**, and analyze and optimize the problematic SQL statements.
5. Subscribe to daily health reports to obtain SQL and performance analysis results, including slow SQL analysis, all SQL analysis, performance & storage analysis, and performance metric trend charts. You can receive a diagnosis report if there are any risks detected.

   For details, see **Daily Reports**.

1.13.5.4 RDS for MySQL Metadata Locks

RDS for MySQL uses metadata locking to manage concurrent access to database objects and to ensure data consistency. Metadata locks have been introduced since MySQL 5.5. A metadata lock on a table prevents any data from being read or written, resulting in SQL statements being blocked. You can use Data Admin Service (DAS) to resolve this issue.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Alternatively, click the instance name on the **Instances** page. On the displayed **Overview** page, click **Log In** in the upper right corner.

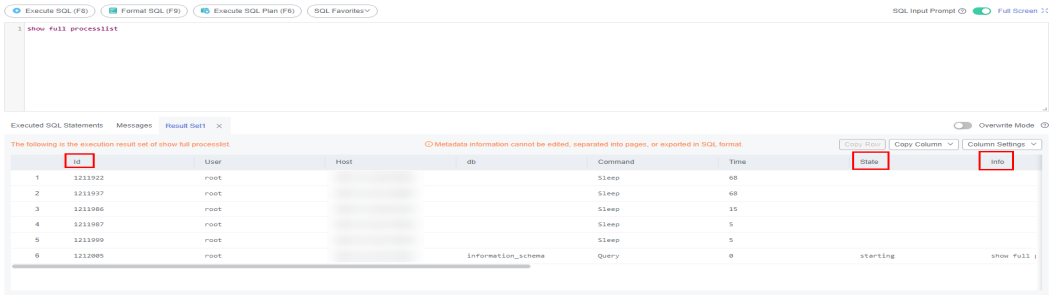
Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Run the following SQL statement in the SQL window to view the states of all database threads:

show full processlist

Figure 1-195 Execution result



Step 8 Check whether a large number of **Waiting for table metadata lock** are displayed in the **State** column, which would indicate that SQL statements are being blocked. Locate the sessions in the table operations in the **Info** column and record the values in the **Id** column.

Step 9 Run the following command in the SQL window to unlock the metadata lock:

kill *Id*

Figure 1-196 Execution result

----End

1.13.5.5 Troubleshooting Slow SQL Issues for RDS for MySQL DB Instances

This section describes how to troubleshoot slow SQL statements on RDS for MySQL DB instances. For any given service scenario, query efficiency depends on the architecture and on the database table and index design. Poorly designed architecture and indexes will cause many slow SQL statements.

Slow SQL Statements Caused by SQL Exceptions

- Causes and symptoms

  There are many causes for SQL exceptions, for example, unsuitable database table structure design, missing indexes, or too many rows that need to be scanned.

  On the slow query logs page of the management console, you can download slow query logs to identify the slow SQL statements and see how long they took to execute. For details, see **Viewing and Downloading Slow Query Logs**.

- Solution

  Optimize the SQL statements that you need to execute.

Slow SQL Statements Caused by DB Instance Limits

- Causes and symptoms

  DB instance performance can be limited because:

  - Your workloads have been increasing but the storage has not been scaled up accordingly.
  - The performance of your DB instance has been deteriorating as the physical server of the instance ages.
  - The amount of data has been increasing, and the data structure has been changing.

  You can view the resource usage of the DB instance on the console. If the values of all resource usage metrics are close to 100%, your DB instance may reach its maximum performance. For details, see **Viewing Performance Metrics of a DB Instance**.

- Solution

  Upgrade the instance class. For details, see **Changing a DB Instance Class**.

Slow SQL Statements Caused by Version Upgrades

- Causes and symptoms

  Upgrading your DB instance may change the SQL execution plan. The join types determined in the execution plan are, in descending order of efficiency:

  system > const > eq_ref > ref > fulltext > ref_or_null > index_merge > unique_subquery > index_subquery > range > index > all

  For more information, see **official MySQL documentation**.

  If your application frequently resends query requests that specify range and index joins but RDS processes these query requests slowly, a number of SQL statements are parallelized. In this case, your application is slow to release threads. As a result, the connections in the connection pool get depleted, affecting all the workloads on your DB instance.

You can log in to the console to see how many current connections your DB instance has established. For details, see **Viewing Performance Metrics of a DB Instance**.

- Solution

  Analyze the index usage and the number of rows to scan, estimate the query efficiency, reconstruct SQL statements, and adjust indexes. For details, see **Executing SQL Plan**.

  NOTE

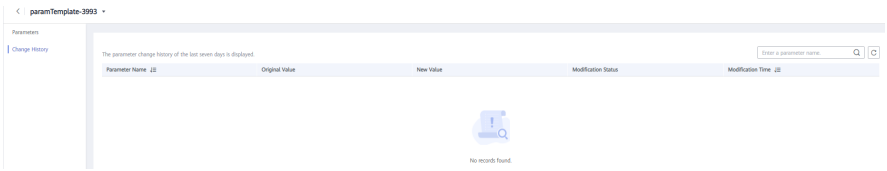
  Incorrectly optimizing slow queries may cause service exceptions. Exercise caution when performing this operation.

Slow SQL Statements Caused by Inappropriate Parameter Settings

- Causes and symptoms

  Inappropriate settings of some parameters (such as **innodb_spin_wait_delay**) can impact performance.

  You can view parameter modifications on the console. For details, see **Viewing Parameter Change History of a DB Instance**.

- Solution

  Modify related parameters based on your specific service scenario. For details, see **Suggestions on RDS for MySQL Parameter Tuning**.

Slow SQL Statements Caused by Batch Operations

- Causes and symptoms

  A large number of operations are performed to import, delete, and query data.

  You can view **Total Storage Space**, **Storage Space Usage**, and **IOPS** on the console. For details, see **Viewing Performance Metrics of a DB Instance**.

- Solution

  Perform batch operations during off-peak hours, or split them.

Slow SQL Statements Caused by Scheduled Tasks

- Causes and symptoms

  If the load of your DB instance changes regularly over time, there may be scheduled tasks causing this.

  You can view **DELETE Statements per Second**, **INSERT Statements per Second**, **INSERT_SELECT Statements per Second**, **REPLACE Statements per Second**, **REPLACE_SELECTION Statements per Second**, **SELECT Statements per Second**, and **UPDATE Statements per Second** on the console to determine whether the load has been changing regularly. For details, see **Viewing Monitoring Metrics**.

- Solution

  Adjust the time when scheduled tasks are run. You are advised to run scheduled tasks during off-peak hours and change the maintenance window to off-peak hours. For details, see **Changing the Maintenance Window**.

1.14 Security and Encryption

1.14.1 Database Account Security

Setting the Account Password Complexity

For details about the database password strength requirements on the RDS console, see the database configuration table in **Buying an RDS for MySQL DB Instance**.

RDS for MySQL 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 password.

Account Description

To provide O&M services, the system automatically creates system accounts when you create RDS for MySQL DB instances. These system accounts are unavailable to you.

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: 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: the replication account, which is used to synchronize data from primary DB instances to standby DB instances or read replicas.
- rdsBackup: the backup account, which is used for backend backup.
- rdsMetric: the metric monitoring account, which is used by watchdog to collect database status data.
- rdsProxy: 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.

Setting Password Complexity

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance to navigate to the **Overview** page.

Passwords must:

- Consist of at least eight characters.
- Contain at least one uppercase letter, one lowercase letter, one digit, and one special character.
- Must be different from the user name.

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

The following parameters can be modified only for RDS for MySQL 5.6 and 5.7.

- **validate_password_length**: Set this parameter to **8**.
- **validate_password_mixed_case_count**: Set this parameter to **1**.
- **validate_password_number_count**: Set this parameter to **1**.
- **validate_password_special_char_count**: Set this parameter to **1**.
- **validate_password_policy**: Set this parameter to **MEDIUM**.

NOTICE

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

- If the value is **Yes** and the DB instance status on the **Instances** 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

1.14.2 Resetting the Administrator Password to Restore Root Access

Scenarios

If you forget the password of the administrator account **root**, you can reset the password. The new password is applied immediately without rebooting the instance. You can only reset the administrator password from the primary instance.

Precautions

- If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password.
- 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.
- 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.
- If you have logged in to your instance as the **root** user, resetting the password may interrupt services. Exercise caution when performing this operation.

Method 1

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

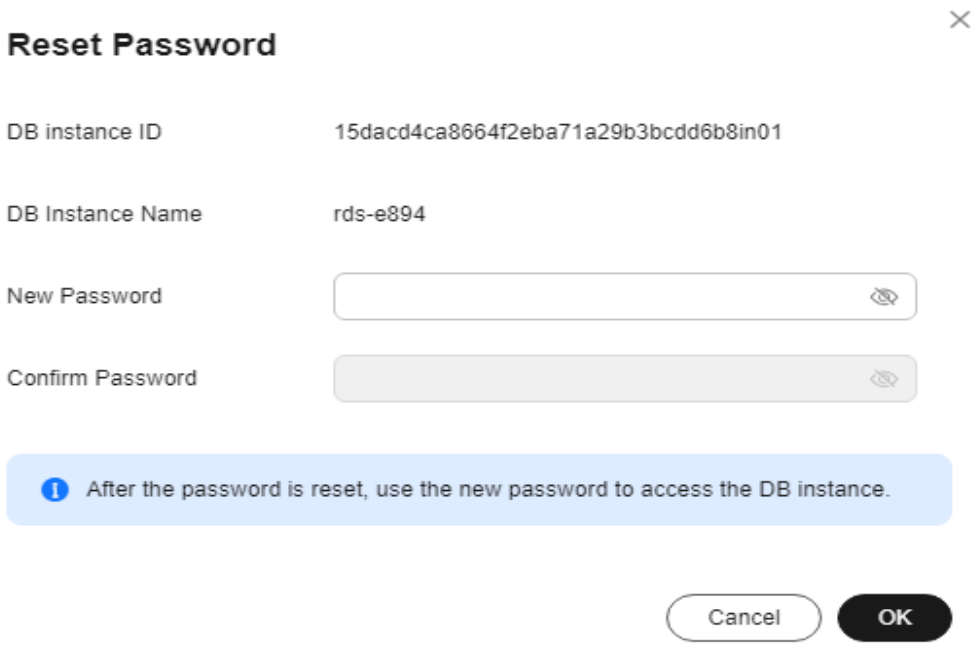
Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 Enter and confirm the new password.

Figure 1-197 Resetting the administrator password



NOTICE

Keep this password secure. The system cannot retrieve it.

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 (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |). 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

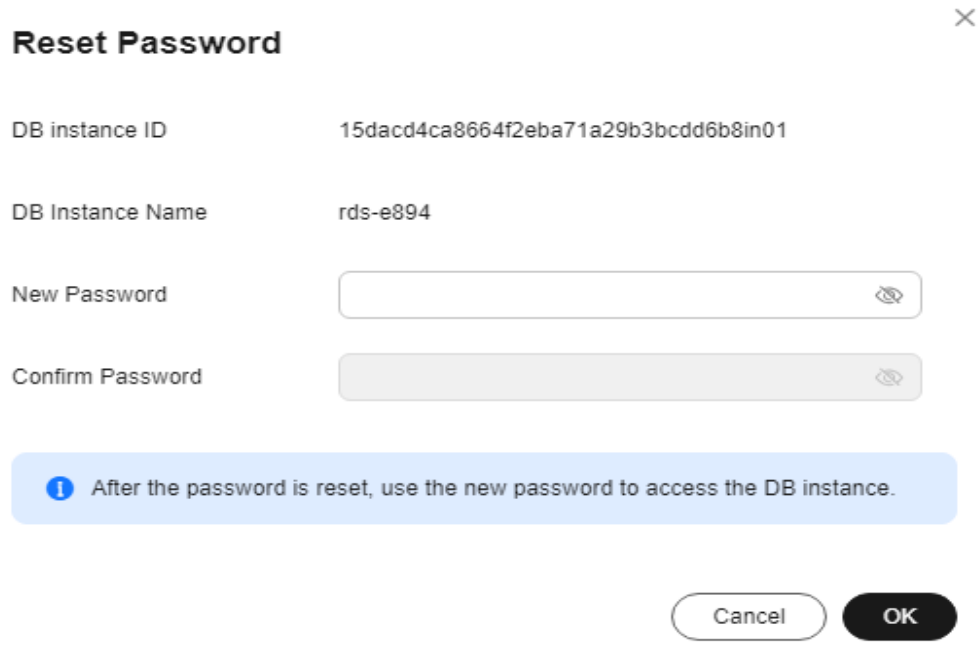
Step 5 On the **Overview** page, find **Administrator** and click **Reset Password** under it.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Enter and confirm the new password.

Figure 1-198 Resetting the administrator password



NOTICE

Keep this password secure. The system cannot retrieve it.

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 (~ ! @ # $ % ^ * - _ = + ? , ( ) & . |). 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

1.14.3 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 be changed as well.

Precautions

- If you need to change the security group of a DB instance with read/write splitting enabled, **submit a service ticket** to apply for required permissions.
- Changing the security group of a DB instance with read/write splitting enabled will also change the security group of its read replicas. The security group of the read replicas cannot be changed independently. Check whether the new security group meets the expectation to prevent impact on existing workloads.
- You can add or modify rules for the security group associated with your DB instance, or delete the security group.

Managing Security Groups

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance or read replica.

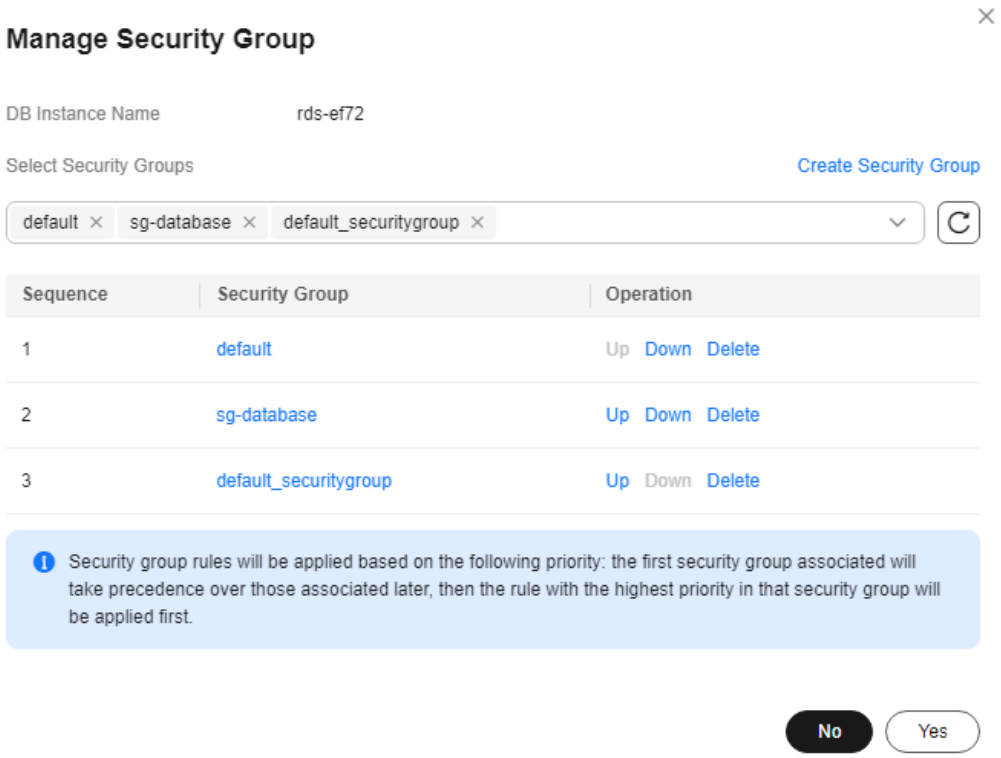
Step 5 On the **Overview** page, click **Manage** under **Security Group**.

- You can select multiple security groups at a time. The security group rules will be applied based on the following sequence: the first security group associated will take precedence over those associated later, then the rule with the highest priority in that security group will be applied first.
- To create a new security group, click **Create Security Group**.

NOTE

Using multiple security groups may impact the network performance. Selecting more than five security groups is not recommended.

Figure 1-199 Managing security groups



Step 6 Click **Yes** to submit the modification.

----End

1.14.4 Performing a Server-Side Encryption

Introduction

The RDS console provides server-side encryption with Data Encryption Workshop (DEW)-managed keys.

DEW uses a third-party hardware security module (HSM) to protect keys, enabling you to easily create and control encryption keys. For security reasons, keys are not displayed in plaintext outside of HSMs. With DEW, all operations on keys are controlled and logged, and usage records of all keys can be provided to meet regulatory compliance requirements.

If server-side encryption is enabled, disk data will be encrypted and stored on the server when you create a DB instance or expand disk capacity. When downloading encrypted objects, the encrypted data will be decrypted on the server and displayed to you in plaintext.

Encrypting Disks Using Server-Side Encryption

For server-side encryption, you need to first create a key using Data Encryption Workshop (DEW) or use the default key that DEW comes with. When creating a DB instance, select **Enable** for disk encryption and select or create a key. The key is the end tenant key and is used for server-side encryption.

- You will need the KMS administrator permission for the region where RDS is deployed. This permission can be granted using Identity and Access Management (IAM). On the IAM console, add permission policies to user groups. For details, see **Creating a User Group and Assigning Permissions**.
- If you want to use a user-defined key to encrypt objects to be uploaded, create a key using DEW. RDS supports only symmetric keys. For details, see **Creating a CMK**.
- If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. Disk encryption will not encrypt backup data stored in OBS. To enable backup data encryption, **submit a service ticket** to request required permissions.
- If disk encryption or backup data encryption is enabled, keep the key properly. Once the key is disabled, deleted, or frozen, the database will be unavailable and data may not be restored.
  - If disk encryption is enabled but backup data encryption is not enabled, you can **restore data to a new instance from backups**.
  - If both disk encryption and backup data encryption are enabled, data cannot be restored.
- If you scale up a DB instance with disks encrypted, the expanded storage space will also be encrypted using the original encryption key.

1.14.5 Configuring an SSL Connection

Secure Socket Layer (SSL) is an encryption-based Internet security protocol for establishing an encrypted link between a server and a client. It provides authenticated Internet connections to ensure the privacy and integrity of online communications. SSL:

- Authenticates users and servers, ensuring that data is sent to the correct clients and servers.
- Encrypts data, preventing it from being intercepted during transmission.
- Ensures data integrity during transmission.

Clients using versions earlier than 5.1 have SSL compatibility issues. By default, SSL is disabled for new RDS for MySQL instances. If your client has no SSL compatibility issues, you can enable SSL by referring to **Enabling SSL**. Enabling SSL will increase the network connection response time and CPU resource consumption. Before enabling it, evaluate any potential impacts on service performance.

You can connect to a DB instance through a client using an SSL or non-SSL connection.

- If SSL is disabled (default), use a non-SSL connection.
- If SSL is enabled, use an SSL connection. SSL encrypts connections to the instance, making in-transit data more secure.

NOTICE

Enabling or disabling SSL will cause DB instances to reboot and interrupt connections. Exercise caution when performing this operation.

To enhance security, the cipher suite ECDHE-RSA-AES128-GCM-SHA256, ECDHE-RSA-AES256-GCM-SHA384, DHE-RSA-AES128-GCM-SHA256, or DHE-RSA-AES256-GCM-SHA384 is recommended for SSL connection. To use these cipher suites, **submit a service ticket** to configure the **ssl_cipher** parameter.

Enabling SSL

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 On the **Overview** page, find **SSL** and click **Enable**.

Step 6 In the displayed dialog box, click **OK**.

Step 7 Wait for some seconds and check that SSL has been enabled on the **Overview** page.

----End

Disabling SSL

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 On the **Overview** page, find **SSL** and click **Disable**.

Step 6 In the displayed dialog box, click **OK**.

Step 7 Wait for some seconds and check that SSL has been disabled on the **Overview** page.

----End

1.14.6 Configuring the TDE Function

Scenarios

Transparent Data Encryption (TDE) performs real-time I/O encryption and decryption on data files. Data is encrypted before being written to disks and is

decrypted when being read from disks to memory. This effectively protects the security of databases and data files.

TDE ensures data security in the following scenarios:

- Hard disks are stolen, causing data leakage.
- Hackers intrude the system and copy the files, causing data leakage. If TDE is not enabled for a database, hackers can browse all data in it as long as they obtain the database file. If TDE is enabled, all data in the database is encrypted. No one can access the data without a key.

Constraints

- You need to enable Key Management Service (KMS) for your RDS for MySQL instance first. The Customer Master Key (CMK) used for encryption is generated and managed by KMS. RDS does not provide any keys or certificates required for encryption.
- TDE is only available to DB instances whose engine version is MySQL 5.7 (5.7.38.221000 or later) and storage type is cloud SSD.
- To enable TDE, submit a service ticket by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.
- Once enabled, TDE cannot be disabled and the default CMK cannot be changed.
- TDE encrypts instance data, including full backups but excluding incremental backups.
- TDE cannot be enabled for instances:
  - With remote disaster recovery backup enabled.
- The following operations cannot be performed for DB instances with TDE enabled:
  - Enabling remote disaster recovery backup.
  - Restoring a TDE-encrypted backup to another existing DB instance.
  - Restoring data to another existing DB instance during point-in-time recovery (PITR).
  - Restoring manual backups created using custom keys.
  - Downloading manual and automated backups created using default keys.
- Enabling TDE will not cause your instance to reboot, but will increase CPU usage significantly. You are advised to enable TDE during off-peak hours.

Enabling Instance-Level TDE

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

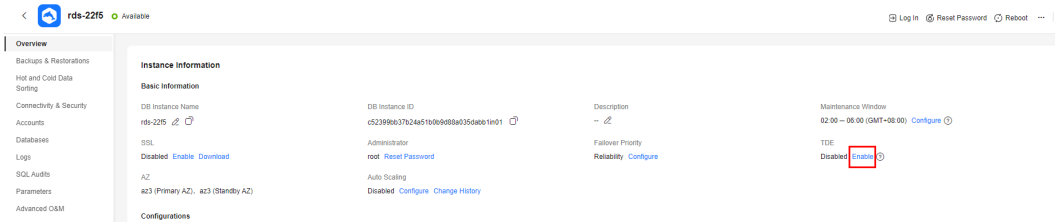
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 Under **TDE**, click **Enable**.

Figure 1-200 Enabling instance-level TDE

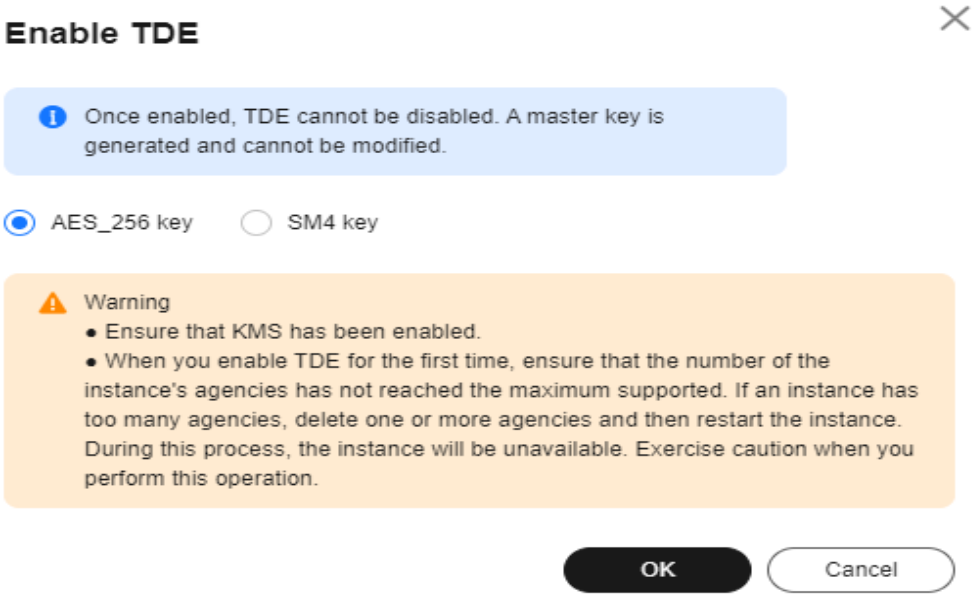


Step 6 In the displayed dialog box, click **OK**.

NOTE

Once enabled, TDE cannot be disabled. Exercise caution when deciding to enable TDE.

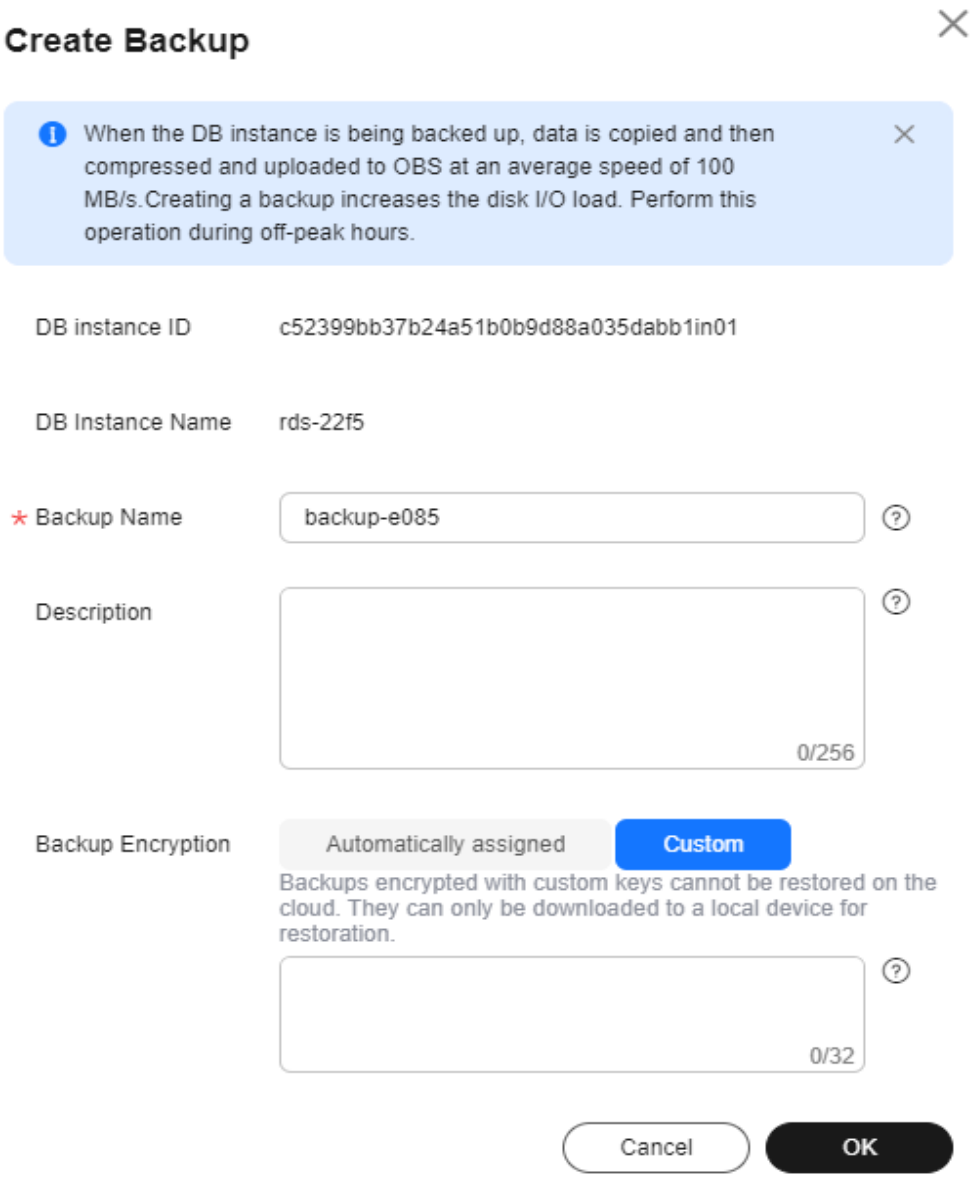
Figure 1-201 Enabling TDE



Step 7 After TDE is enabled, to restore data to an on-premises database, use either of the following methods.

- Method 1: Decrypt data.
  a. Decrypt data by referring to **Decryption**.
  b. Create a manual backup for the instance to be restored.
  c. Restore data from the manual backup.
- Method 2: Use the transition key **--transition-key**.
  a. When creating a manual backup on the console, enter a custom key string as prompted to re-encrypt the data. For details, see **Creating a Manual Backup**.

Figure 1-202 Custom encryption



b. Download a full backup and use the third-party full backup tool Percona XtraBackup to restore the backup locally.

   i. **prepare** phase: --transition-key = {*custom_key*}

   ii. **copy-back** phase: --transition-key={*custom_key*} --generate-new-master-key

----End

Encrypting or Decrypting a Table

NOTE

- Ensure that instance-level TDE has been enabled.
- After TDE is enabled, common database tools can still be used.
- When table data is queried, the data is decrypted and read to the memory, so the query result is displayed in plaintext. After TDE is enabled, backup files are encrypted, preventing data leakage caused by backup leakage.

Step 1 **Connect to the target DB instance**.

Step 2 Run the following commands to encrypt or decrypt a table. In the commands, *tablename* indicates the name of the table to be encrypted or decrypted.

- Encryption
  ```
  alter table tablename encryption='Y';
  ```
- Decryption
  ```
  alter table tablename encryption='N';
  ```

----End

1.14.7 Configuring a Password Expiration Policy

Scenarios

Using the same password too long makes it easier for hackers to crack or guess your password. Requiring password changes after a certain amount of time can improve security.

You can configure a password expiration policy for your instance in either of the following ways:

- **Modify the database parameter**: RDS for MySQL 5.7 and 8.0 allow you to set the global variable **default_password_lifetime** to define the number of days before your password expires and must be changed.
- **Configure the password expiration policy through DAS**: Different password expiration policies can be configured for different users.

Precautions

- Once your password expires, you cannot log in to the database.
- After the password expiration policy is configured, you need to periodically check whether your password is about to expire.

Modifying the Database Parameter

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Parameters**.

Step 6 On the displayed page, change the value of **default_password_lifetime**.

The value of this parameter indicates how many days until a password expires. The default value is **0**, indicating that the created user password will never expire.

Step 7 Click **Save**. In the displayed dialog box, click **Yes**.

----End

Configuring the Password Expiration Policy Through DAS

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Alternatively, click the DB instance on the **Instances** page. On the displayed **Overview** page, click **Log In** in the upper right corner.

Step 5 Enter the username and password and click **Log In**.

Step 6 Choose **SQL Operations** > **SQL Query**.

Step 7 In the editing area, compile the statement shown below. The unit of **password_life_time** is day. You are advised to set it to **180**.

ALTER USER *username* PASSWORD EXPIRE INTERVAL *password_life_time* DAY;

Step 8 Click **Execute SQL**. Then, view SQL execution status on the **Executed SQL Statements**, **Messages**, and **Result** tab pages.

----End

1.14.8 Unbinding an EIP

The Elastic IP (EIP) service enables your RDS instances to communicate with the Internet using static public IP addresses and scalable bandwidths. But this increases the risk of network-wide attacks on your instances. Using an EIP leaves you open to DoS or DDoS attacks.

As an internal component, the database can be accessed using an internal IP address. Therefore, you are advised to unbind the EIP from the database.

Unbinding an EIP

Step 1 Log in to the management console.

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance that has an EIP bound.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** in the connection topology. In the displayed dialog box, click **Yes**.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 On the **Connectivity & Security** page, view the results.

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

1.14.9 Using the Database of the Latest Version

When the MySQL community releases new CVE vulnerabilities, we analyze the impacts of the vulnerabilities in a timely manner and determine patch release plans based on the analysis results. You are advised to upgrade your database and fix the vulnerabilities in a timely manner to prevent the vulnerabilities from affecting data security.

For more information about minor versions, see **RDS for MySQL Kernel Version Description**.

Procedure

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

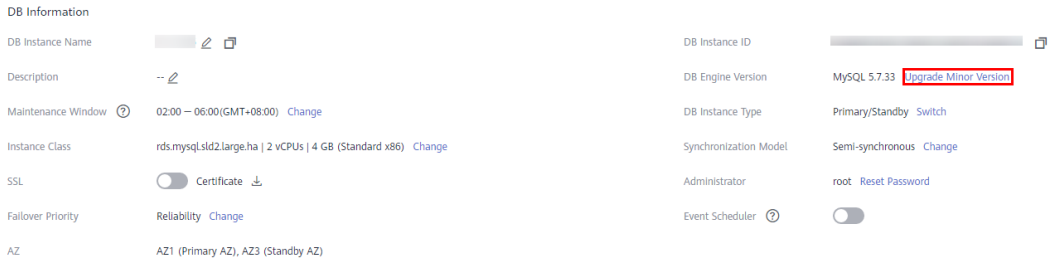
Step 2 Click 📍 in the upper left corner and select a region.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **DB Engine Version**, click **Upgrade Minor Version**.

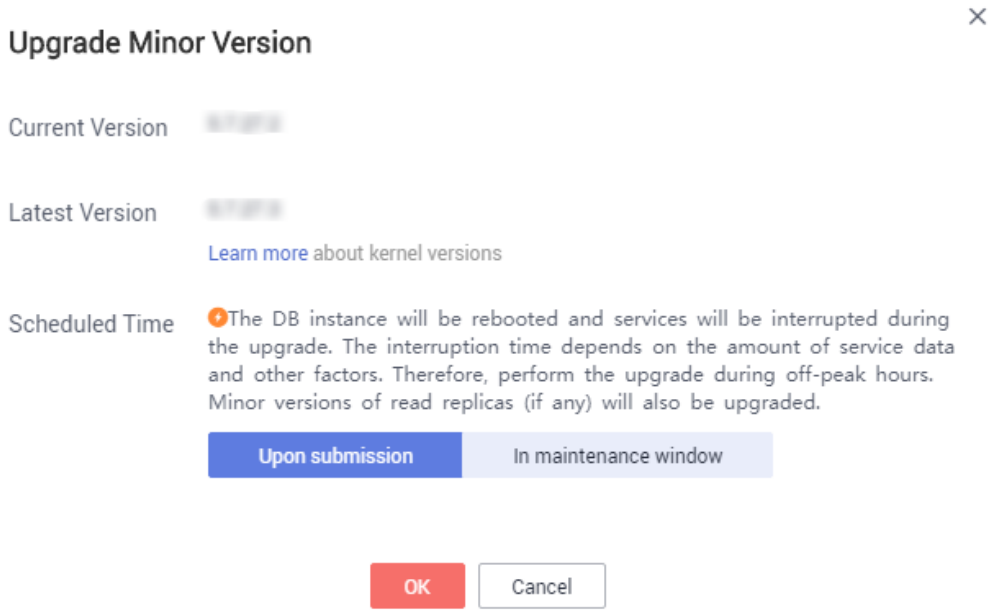
Figure 1-203 Upgrading a minor version



Step 6 In the displayed dialog box, select a scheduled time and click **OK**.

- Upon submission: The system upgrades the minor version immediately after you have submitted your upgrade request.
- In maintenance window: The system will upgrade the minor version during the maintenance window that you have configured.

Figure 1-204 Selecting a scheduled time



----End

1.14.10 Using DBSS (Recommended)

Database Security Service (DBSS) is an intelligent database security service. Based on the machine learning mechanism and big data analytics technologies, it can audit your databases, detect SQL injection attacks, and identify high-risk operations.

You are advised to use DBSS to provide extended data security capabilities. For details, see **Database Security Service**.

Advantages

- DBSS can help you meet security compliance requirements.
  - DBSS can help you comply with DJCP (graded protection) standards for database audit.
  - DBSS can help you comply with security laws and regulations, and provide compliance reports that meet data security standards (such as Sarbanes-Oxley).
- DBSS can back up and restore database audit logs and meet the audit data retention requirements.
- DBSS can monitor risks, sessions, session distribution, and SQL distribution in real time.
- DBSS can report alarms for risky behavior and attacks and respond to database attacks in real time.
- DBSS can locate internal violations and improper operations and keep data assets secure.

Deployed in bypass pattern, database audit can perform flexible audits on the database without affecting user services.

- Database audit monitors database logins, operation types (data definition, operation, and control), and operation objects based on risky operations to effectively audit the database.
- Database audit analyzes risks and sessions, and detects SQL injection attempts so you can stay apprised of your database status.
- Database audit provides a report template library to generate daily, weekly, or monthly audit reports according to your configurations. It sends real-time alarm notifications to help you obtain audit reports in a timely manner.

1.15 Parameters

1.15.1 Modifying Parameters of an RDS for MySQL Instance

You can change parameter values in a custom parameter template and apply it to optimize RDS database performance.

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

Keep in mind the following points before modifying the parameters:

- Modifying instance parameters: If you modify dynamic parameters on the **Parameters** page of a DB instance and save the modifications, the modifications take effect immediately regardless of the **Effective upon Reboot** setting. However, if you modify static parameters on the **Parameters** page of a DB instance and save the modifications, the modifications do not take effect until you manually reboot the DB instance.
- Modifying parameter template parameters: If you modify parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you apply the template to your DB instances. If you modify static parameters in a custom

parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you apply the template to your DB instances and manually reboot those DB instances. For details, see **Applying a Parameter Template**.

Global parameters can only be modified on the console. Session-level parameters can be modified using SQL statements. When you modify a parameter, the time when modifications take effect varies with the parameter type.

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**. You need to manually reboot the DB instance for the latest modifications to take effect for that DB instance.

NOTE

For better parameter modification experience, you are advised to **upgrade the minor version of your DB instance** to the latest.

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 with incorrect settings is applied to a DB instance, this instance may fail to start. If this happens, you can re-configure the custom parameter template based on the settings of the default parameter template.

Before modifying parameters, make sure you understand their meanings and fully verify the changes in a test environment to avoid instance or service exceptions caused by inappropriate parameter settings.

Modifying a Custom Parameter Template and Applying It to a DB Instance

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

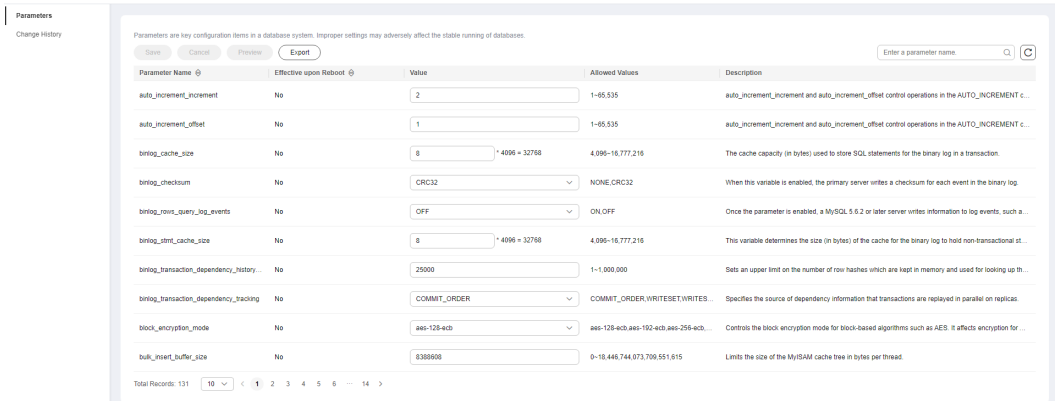
Step 4 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

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

For parameter details, see **Suggestions on RDS for MySQL Parameter Tuning**.

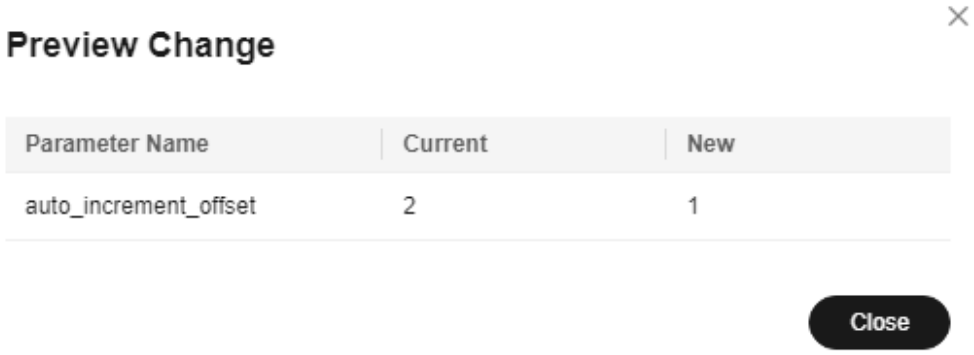
Available operations are as follows:

Figure 1-205 Modifying parameters in a parameter template



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

Figure 1-206 Previewing changes



Step 6 Click **Change History** to view the changes.

Step 7 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

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

If the DB instance status is **Parameter change. Pending reboot**, you need to reboot the DB instance 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 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

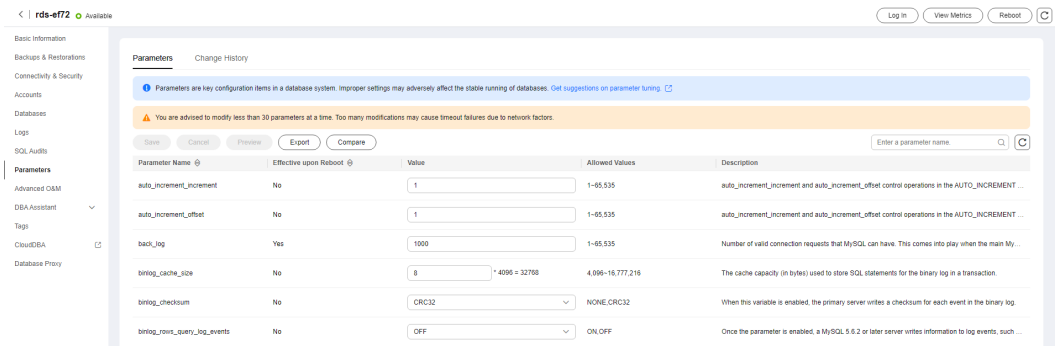
Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, modify parameters as required.

NOTICE

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

- If the value is **Yes** and the DB instance status on the **Instances** 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.

Figure 1-207 Parameters



- 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

Modifying Parameters of Multiple DB Instances at a Time

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the DB instances with the same DB engine of the same version and choose **More** > **Modify Parameters** above the DB instance list.

Step 5 On the **Modify Parameters** page, select the parameters you want to modify, change the parameter values, and click **Apply**.

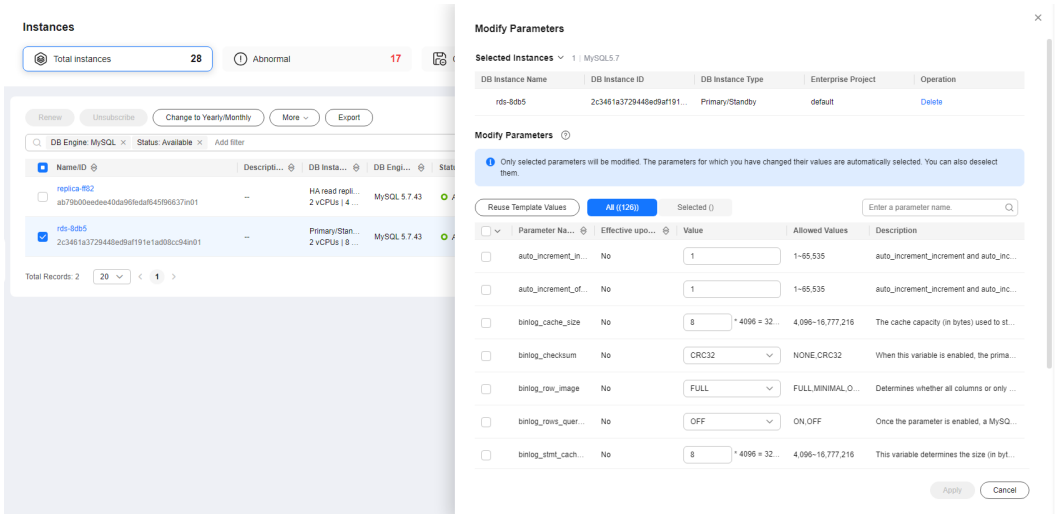
A maximum of 30 parameters can be modified at a time. Only selected parameters will be applied to your DB instances. The modified parameters are automatically selected. You can also deselect them.

NOTICE

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

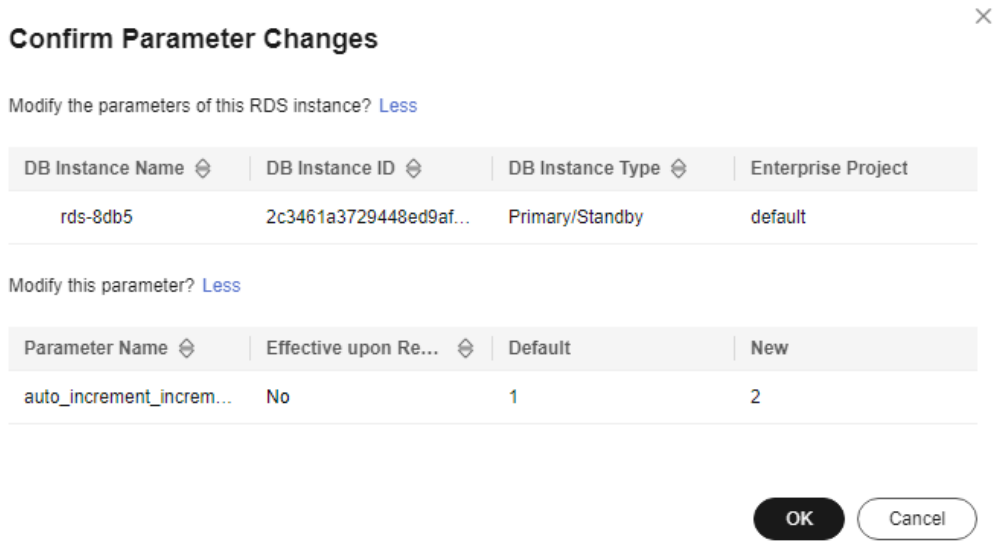
- If the value is **Yes** and the DB instance status on the **Instances** page is **Parameter change. Pending reboot**, a reboot is required for the modifications to take effect.
  - If you have modified parameters of a primary DB instance, you need to reboot the primary DB instance for the modifications 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.

Figure 1-208 Modifying parameters



Step 6 In the displayed dialog box, click **OK**.

Figure 1-209 Confirming parameter changes



Step 7 After the parameters are modified, click the **Parameter Changes** tab on the **Parameter Templates** page to view details about the modified parameters.

----End

Viewing Parameter 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.

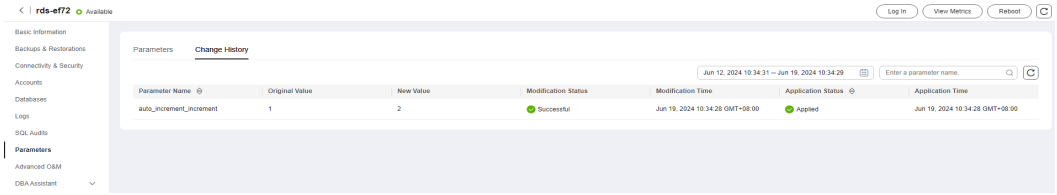
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

You can view the parameter change history within a specified period (no more than two years). By default, the parameter change history of the last seven days is queried.

Figure 1-210 Viewing parameter change history



You can view the parameter names, original parameter values, new parameter values, modification statuses, modification time, application statuses, and application time.

----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.

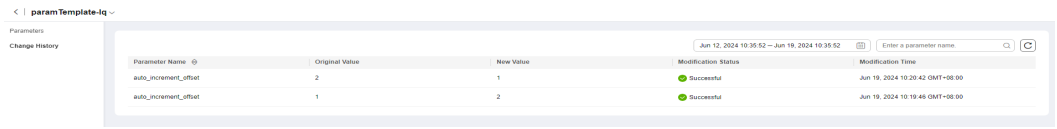
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane, choose **Parameter Templates**. On the **Custom Templates** page, click the target parameter template.

Step 5 On the displayed page, choose **Change History** in the navigation pane.

You can view the parameter change history within a specified period (no more than two years). By default, the parameter change history of the last seven days is queried.

Figure 1-211 Viewing parameter change history



You can view the parameter names, original parameter values, new parameter values, modification statuses, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

Viewing Parameter Changes

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click the **Parameter Changes** tab.

Step 5 Click **View Details** in the **Operation** column.

You can view detailed information about the modified parameters.

----End

Common Parameters

Table 1-78 Common parameters

Parameter	Description	Reference
time_zone	The time zone. Keep the value of this parameter for read replicas the same as that for the primary instance.	How Can I Change the Time Zone?
default_password_lifetime	The global automatic password expiration policy, in days.	How Do I Configure a Password Expiration Policy for RDS for MySQL DB Instances?
tx_isolation	The default transaction isolation level.	How Do I Change the RDS Transaction Isolation Level?
character_set_server	The server character set.	How Do I Use the utf8mb4 Character Set to Store Emojis in an RDS for MySQL DB Instance?
lower_case_table_names	The case sensitivity of table names. If this parameter is set to **0**, table names are stored as specified and are case sensitive. If it is set to **1**, table names are stored in lowercase and are case insensitive.	How Do I Set Case Sensitivity for RDS for MySQL Table Names?
group_concat_max_len	The maximum permitted result length in bytes for the GROUP_CONCAT() function.	Incorrect GROUP_CONCAT Results

Parameter	Description	Reference
max_connections	The maximum number of concurrent client connections. If this parameter is set to **default**, the parameter value depends on how much memory there is.	**What Is the Maximum Number of Connections to an RDS DB Instance?**

1.15.2 Managing Parameter Templates

1.15.2.1 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.

If you create a DB instance without specifying a custom DB parameter template, a default parameter template is used. 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 **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 **Replicating a Parameter Template**.

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

- When you change a parameter value in a parameter template that has been applied to a DB instance and save the change, the change takes effect only to the DB instance and does not affect other DB instances.
- The changes to parameter values in a custom parameter template take effect only after you apply the template to DB instances. For details, see **Applying a Parameter Template**.
- When you change dynamic parameter values in parameter templates in batches and save the changes, the changes will take effect only after you

apply the parameter templates to DB instances. When you change static parameter values in parameter templates in batches and save the changes, the changes will take effect for DB instances only after you apply the parameter templates to DB instances and manually reboot the DB instances.

- 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.

Differences Between Standard and High-Performance Parameter Templates

Only RDS for MySQL 5.7 and 8.0 support high-performance parameter templates. Compared with standard parameter templates, high-performance templates provide higher read/write speed but lower data security. The parameter comparisons are as follows:

Table 1-79 Differences between standard and high-performance parameter templates for MySQL 5.7

Parameter	Description	Recommended Value in a High-Performance Template	Recommended Value in a Standard Template
sync_binlog	This parameter is used to control how often the MySQL server synchronizes the binlogs to disk. If the default value is used, the binlogs are synchronized to disk each time a transaction is committed. If it is set to **0**, the MySQL server relies on the operating system to flush the binlogs to disk from time to time as it does for any other file. This setting provides the best performance but lowest security. If it is set to *N* (*N* > 1), the binlogs are synchronized to disk after *N* transactions are committed. This parameter has been adjusted in high-performance parameter templates. If you use a high-performance parameter template for your DB instance, data may be lost after your instance is recovered from a crash and replication exceptions may occur.	1000	1
binlog_cache_size	This parameter specifies the size of the binlog cache. If write operations are frequent, increasing the value of this parameter can improve performance. After this parameter is adjusted, out-of-memory risks increase in high-concurrency scenarios, especially for instances with small specifications.	2097152	32768
innodb_flush_log_at_trx_commit	Setting this parameter to **0** improves write performance in low-concurrency scenarios. Adjusting this parameter may cause 1 second of data to be lost in extreme scenarios.	2	1

Table 1-80 Differences between standard and high-performance parameter templates for MySQL 8.0

Parameter	Description	Recommended Value in a High-Performance Template	Recommended Value in a Standard Template
transaction_isolation	This parameter specifies the default transaction isolation level. A high-performance template uses the READ COMMITTED level. Compared with REPEATABLE READ, this level reduces row lock conflicts and does not have gap locks. This isolation level can prevent dirty reads, but phantom reads and non-repeatable reads may still occur.	READ-COMMITTED	REPEATABLE-READ
sync_binlog	This parameter is used to control how often the MySQL server synchronizes the binlogs to disk. If the default value is used, the binlogs are synchronized to disk each time a transaction is committed. If it is set to **0**, the MySQL server relies on the operating system to flush the binlogs to disk from time to time as it does for any other file. This setting provides the best performance but lowest security. If it is set to *N* (*N* > 1), the binlogs are synchronized to disk after *N* transactions are committed. This parameter has been adjusted in high-performance parameter templates. If you use a high-performance parameter template for your DB instance, data may be lost after your instance is recovered from a crash and replication exceptions may occur.	1000	1

Parameter	Description	Recommended Value in a High-Performance Template	Recommended Value in a Standard Template
binlog_cache_size	This parameter specifies the size of the binlog cache. If write operations are frequent, increasing the value of this parameter can improve performance. After this parameter is adjusted, out-of-memory risks increase in high-concurrency scenarios, especially for instances with small specifications.	2097152	32768
innodb_flush_log_at_trx_commit	Setting this parameter to **0** improves write performance in low-concurrency scenarios. Adjusting this parameter may cause 1 second of data to be lost in extreme scenarios.	2	1

Procedure

Step 1 Log in to the management console.

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

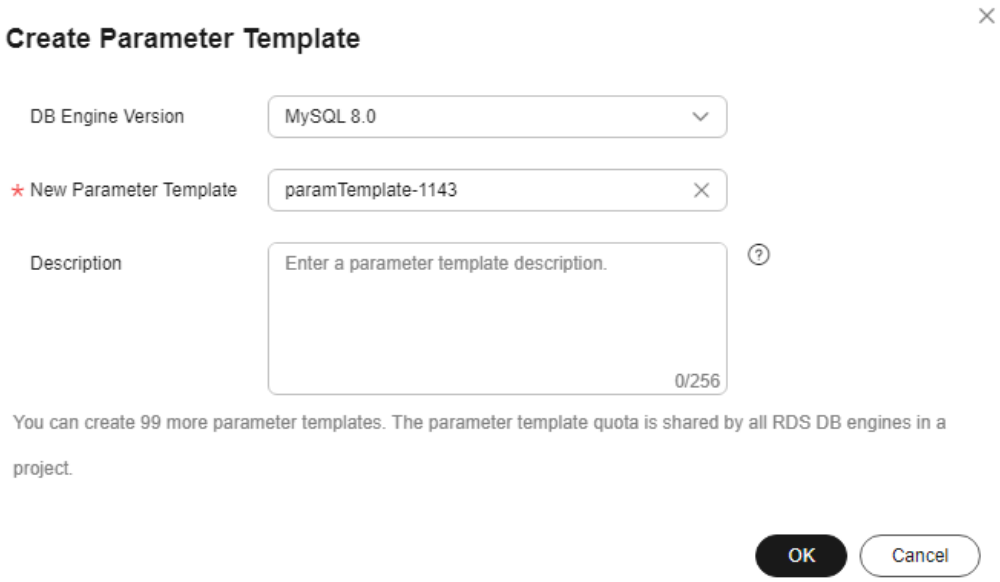
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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: >!<"&'=

Figure 1-212 Creating a parameter template



----End

1.15.2.2 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.
- After a default parameter template is applied to an instance, all parameters in the instance are changed to be the same as those in the default parameter template, and all specification parameters are reset based on the instance specifications.

Applying a Parameter Template

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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 **Viewing Application Records of a Parameter Template**.

----End

Viewing Application Records of a Parameter Template

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

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

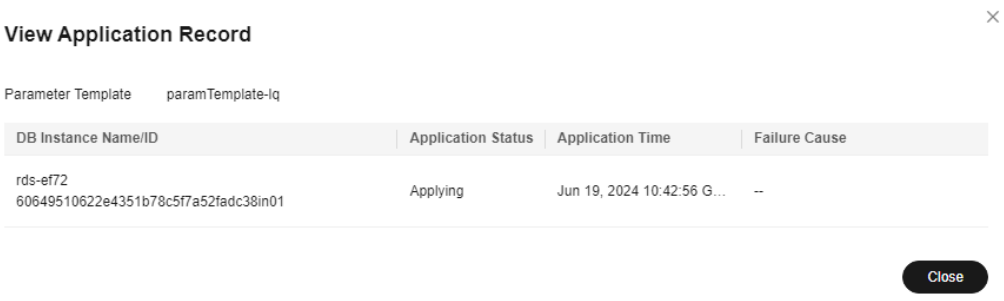
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and 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).

Figure 1-213 Viewing application records of a parameter template



----End

1.15.2.3 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. You can also export the parameter template to generate a new parameter template for future use.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

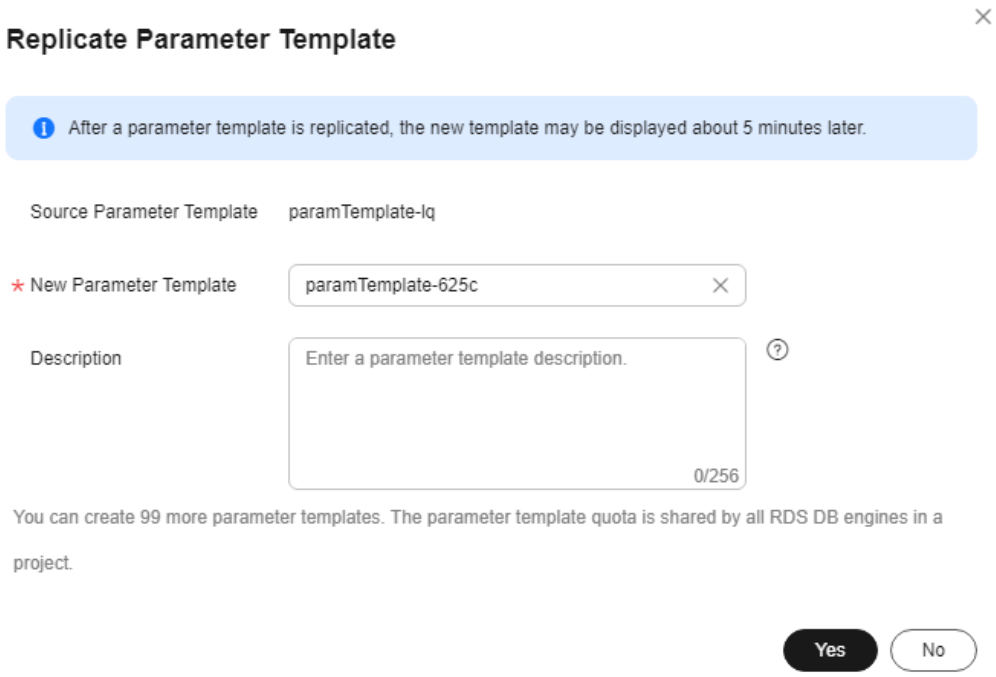
Alternatively, click the target DB instance on the **Instances** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

NOTE

To ensure that your parameter templates are applicable to all types of DB instances and databases can be started normally, the values of **innodb_flush_log_at_trx_commit** and **sync_binlog** exported from primary DB instances or read replicas are **1** by default.

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

Figure 1-214 Replicating a 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: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Templates** page.

----End

1.15.2.4 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.

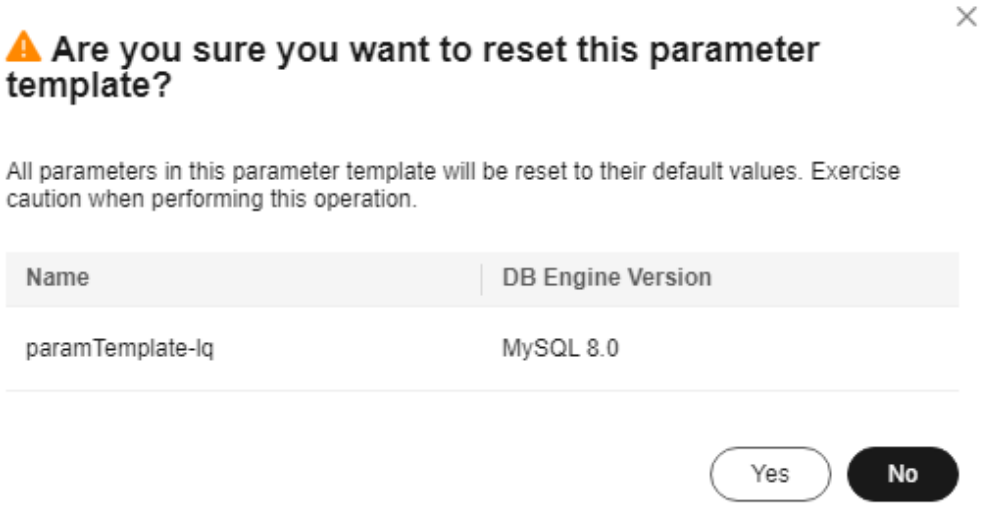
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

Figure 1-215 Confirming the reset



Step 6 Apply the parameter template to your DB instance. 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

1.15.2.5 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.

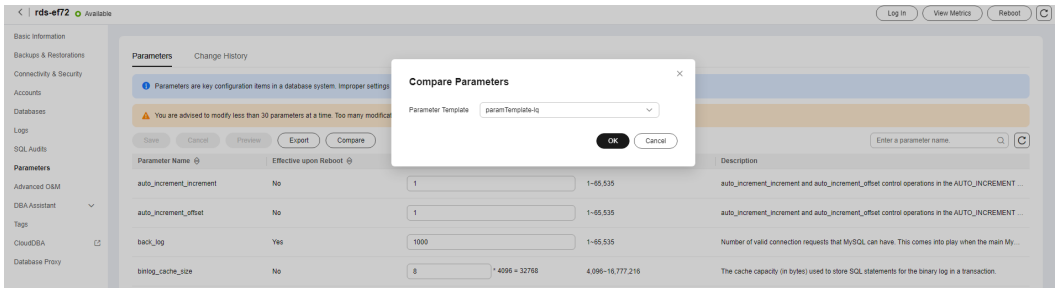
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

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

Figure 1-216 Comparing instance parameters with those in a specified parameter template



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.

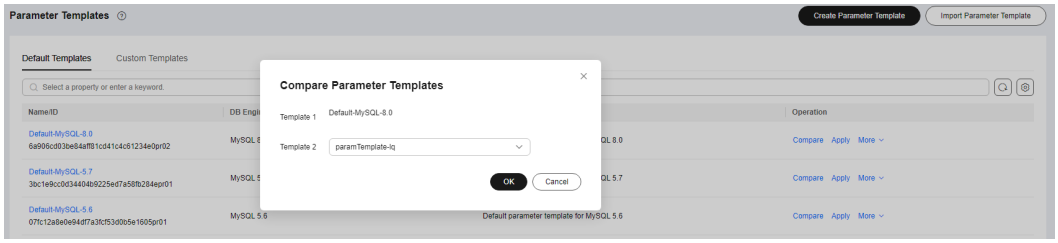
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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**.

Figure 1-217 Selecting a parameter template to be compared



- 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

1.15.2.6 Exporting a Parameter Template

Scenarios

Exporting instance parameters:

- You can export parameters of a DB instance as a new parameter template for future use. To apply the exported parameter template to new DB instances, see **Applying a Parameter Template**. If the specification parameters of the instance have been modified, the exported parameter template also contains the specification parameters. For details about the specification parameters, see **Parameter Changes**.
- You can also export the parameter information (including parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analyzing details.

Exporting a parameter template:

- You can export an RDS for MySQL parameter template (including parameter names, values, and descriptions) to a CSV file for viewing and analyzing details.

Exporting Instance Parameters

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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 in the **Custom Templates** tab on the **Parameter Templates** page.

- Exporting to a file

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

  NOTE

  The file name can contain 4 to 81 characters.

----End

Exporting a Parameter Template

Step 1 Log in to the management console.

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

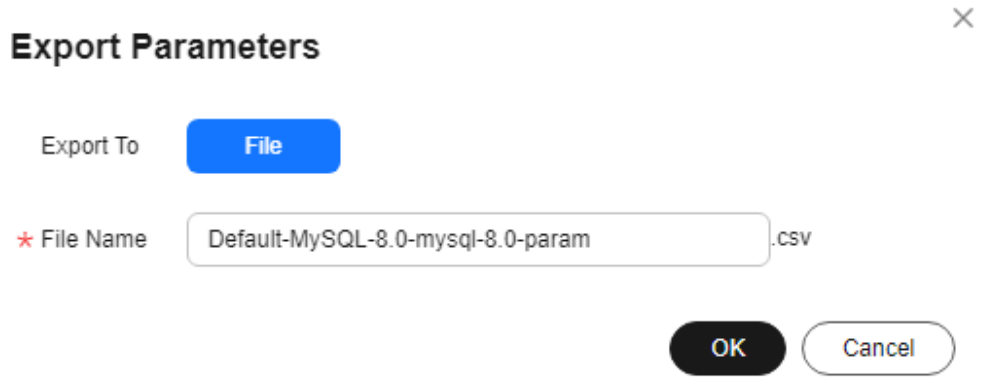
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Default Templates** or **Custom Templates** as needed. On the displayed page, locate the target template and choose **More** > **Export** in the **Operation** column.

Step 5 In the displayed dialog box, enter a file name and click **OK**.

The file name can contain 4 to 81 characters.

Figure 1-218 Exporting a parameter template



----End

1.15.2.7 Importing a Parameter Template

Scenarios

RDS allows you to import new parameter templates for future use. To apply an imported parameter template to new DB instances, see **Applying a Parameter Template**.

Constraints

- Only parameter templates that were exported from the **Parameter Templates** page on the RDS console can be imported.
- If any modification to an exported parameter template causes a change in the file format, the template may not be able to be imported.
- The parameter template to be imported cannot contain parameters related to specifications. For details about such parameters, see **Constraints**.

Procedure

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

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

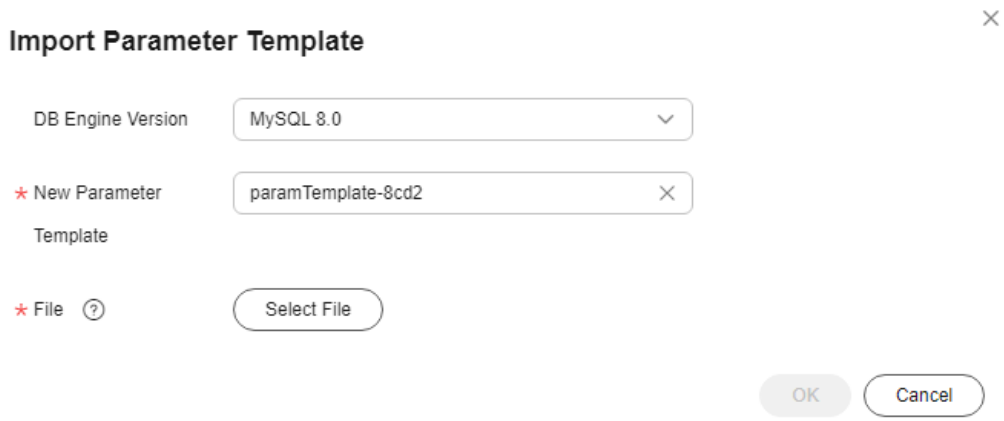
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Import Parameter Template**.

Step 5 In the displayed dialog box, select a DB engine version, enter a new parameter template name, click **Select File**, select the target parameter list (containing parameter names, values, and description), and click **OK**.

Only one file (CSV format) can be imported at a time. The file size cannot exceed 50 KB.

Figure 1-219 Importing a parameter template



----End

1.15.2.8 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or 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

1.15.2.9 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.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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

1.15.3 Suggestions on RDS for MySQL Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect database performance. This section describes some important parameters for your reference. For details, visit the **MySQL official website**.

For details on how to modify RDS for MySQL parameters on the console, see **Modifying Parameters of an RDS for MySQL Instance**.

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 table names stored on disks are case-sensitive when databases and tables are created. The value **1** indicates that table names are case-insensitive and are lowercase by default.

  NOTE

  RDS for MySQL 8.0 does not support modifications to 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, reboot and run **SELECT @@GLOBAL.GTID_EXECUTED** on the primary instance first. 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.

Recommended value for POC: **2**

- **sync_binlog**

  Default value: **1**

  Function: Controls how often the RDS for 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. This is the safest setting. If the value is set to **0**, synchronization of the binary log to the disk is not controlled by the RDS for MySQL server but relies on the OS to flush the binary log to the disk. This setting provides the best performance. However, if a power failure occurs or the OS crashes, all binary log information in **binlog_cache** will be lost.

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

  Recommended value for POC: **1000**

- **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 RDS 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.

- **innodb_buffer_pool_size**

  Default value: Varies depending on the DB instance classes.

  Function: Specifies the size of the InnoDB buffer pool. The InnoDB buffer pool is used to cache table and index data. Increasing the value of this parameter reduces disk I/O.

  Impact: Setting this parameter to a large value may cause system breakdown. Exercise caution when changing this parameter value.

  Recommended value for POC: 70% to 75% of the memory for your DB instances with 32 GB memory or above

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.
- The **max_connections** parameter sets the maximum number of clients allowed to connect to a database concurrently. The default value of this parameter varies depending on the system architecture. System built-in connections occupy some connections. To prevent concurrent connection conflicts, you are advised not to set this parameter to a value less than 30. This parameter cannot be set to a value smaller than the number of current connections.
- 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.
- The **innodb_print_all_deadlocks** parameter is set to **OFF** by default to avoid the performance overhead caused by frequent log writes. If it is set to **ON** and there are a large number of deadlocks in the system, frequent log writes will increase I/O overhead. If all deadlock information is recorded, there will be more and more error logs. For example, if deadlocks frequently occur in high-concurrency scenarios, the storage space is quickly consumed. You can set this parameter to **ON** temporarily when you want to troubleshoot and analyze deadlocks.

Associated Parameters

- **character_set_server**: If you change the value of this parameter, the system changes the values of **collation_server**, **character_set_database**, and **collation_database** accordingly.

  The parameters **character_set_server** and **collation_server** are correlated with each other. The value of **collation_server** starts with the value of **character_set_server**. For example, if **character_set_server** is set to **latin1**, the value of **collation_server** starts with **latin1**.
- **innodb_io_capacity**: The value of this parameter must be less than or equal to the value of **innodb_io_capacity_max**. For example, if **innodb_io_capacity_max** is set to **2000**, the maximum value of **innodb_io_capacity** is **2000**.
- **innodb_buffer_pool_size**: The value of this parameter must be a multiple of the product of the value for **innodb_buffer_pool_chunk_size** and **innodb_buffer_pool_instances**. For example, if **innodb_buffer_pool_chunk_size** is **134217728** and **innodb_buffer_pool_instances** is **1**, the value of **innodb_buffer_pool_size** must be a multiple of **134217728**.

Constraints on Parameter Modification

- When the **innodb_adaptive_hash_index** and **innodb_buffer_pool_size** parameters are modified at the same time, the value of **innodb_adaptive_hash_index** will fail to be changed from **OFF** to **ON**.

- The value of **innodb_buffer_pool_size** must be an integer multiple of the product of **innodb_buffer_pool_instances** and **innodb_buffer_pool_chunk_size**.
- If **innodb_buffer_pool_instances** is set to **2**, the value of **innodb_buffer_pool_size** must be greater than or equal to 1 (unit: GB).
- For MySQL 8.0, if the kernel version is earlier than 8.0.18, the value of **max_prepared_stmt_count** cannot exceed 1048576.

Other Parameters

- **max_prepared_stmt_count**: limits the upper limit of prepared statements. Too many prepared statements consume server memory resources. If this parameter is set to a small value, your DB instance may be vulnerable to the denial of service (DoS) attacks. You are advised to change this parameter value based on service requirements.
- The values of the following parameters will be adjusted based on kernel rules:
  - **key_cache_age_threshold**: automatically adjusted to a multiple of 100.
  - **join_buffer_size** and **key_cache_block_size**: automatically adjusted to multiples of 128.
  - **query_cache_size**, **query_prealloc_size**, **innodb_log_buffer_size**, **max_allowed_packet**, and **thread_stack**: automatically adjusted to multiples of 1024.
  - **read_buffer_size**, **read_rnd_buffer_size**, **binlog_cache_size**, and **binlog_stmt_cache_size**: automatically adjusted to multiples of 4096.
  - **data_buffer_size**, **log_buffer_size**, **shared_pool_size**, and **temp_buffer_size**: automatically adjusted to multiples of 1048576.
- **binlog_format**: set to **row** by default, indicating that binary logs are recorded as the modified data by row, including the data before and after modification. You are advised not to change the value of this parameter.
- **log_timestamps**: controls the time zone of timestamps in various logs, such as error logs and slow query logs. The default value is the system time zone and cannot be changed.
- **skip_name_resolve**: set to **ON** by default, indicating that the system skips domain name resolution and determines whether a connection can be set up based on the IP address in the whitelist.
- **innodb_strict_mode**: restricts the InnoDB check policy. The default value is **OFF**.
- **binlog_rows_query_log_events**: controls whether to write original SQL statements into binlogs. If this parameter is set to **ON**, database performance may deteriorate when a large amount of data is updated. Before you change the parameter value, consider the compatibility with tools such as Otter.

1.16 Log Management

1.16.1 Log Reporting

Scenarios

If you enable log reporting for your DB instance, new logs generated for the instance will be uploaded to Log Tank Service (LTS).

Prerequisites

- Ensure that there are available LTS log groups and log streams in the same region as your instance.

  For details about how to create a log group and log stream, see **Overview**.

Billing

You will be billed for enabling this function under LTS. For details, see **LTS Pricing Details**.

Constraints

- Error logs and slow query logs cannot share the same log stream.
- If a structuring template (MySQL slow log template or MySQL error log template) has been bound to a log stream, ensure that the template type is the same as the log type when you select the log stream. For example, if a MySQL error log template has been bound to a log stream, the log stream cannot be selected for slow query logs.

  For details about how to bind a system template to a log stream, see **Log Structuring**.

Enabling Log Reporting in Batches

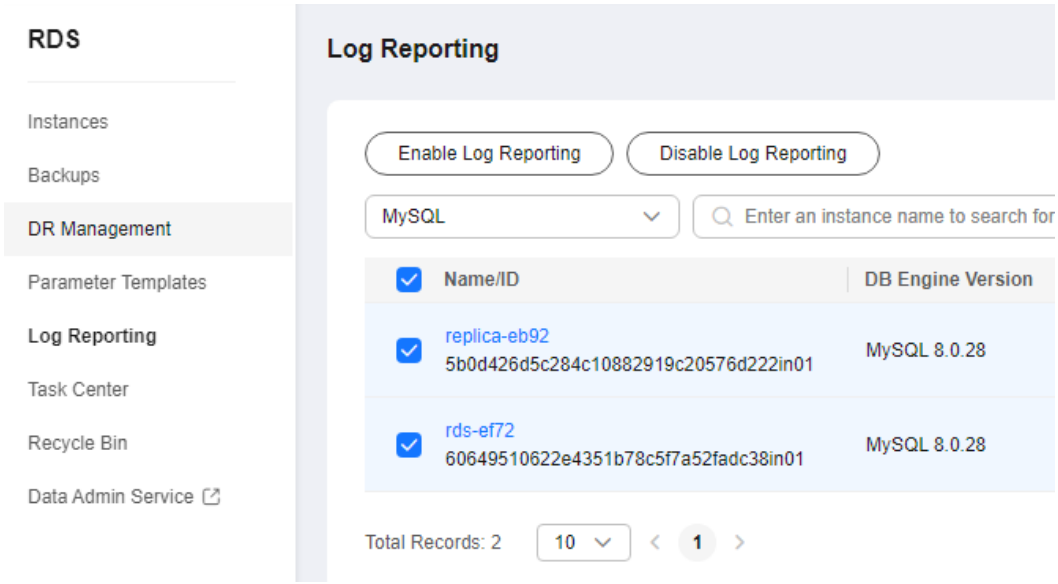
Step 1 **Log in to the management console**.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane, choose **Log Reporting**.

Figure 1-220 Log reporting



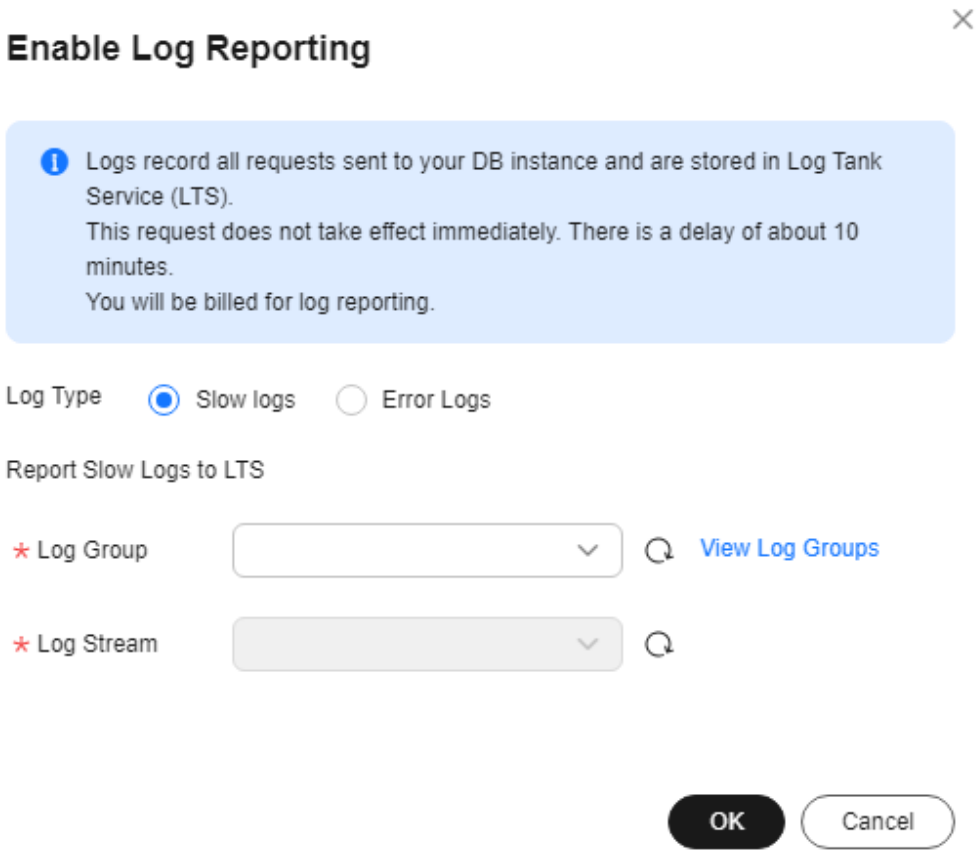
Step 5 Select one or more instances and click **Enable Log Reporting**.

Step 6 Select an LTS log group and log stream and click **OK**.

NOTE

- Error logs and slow query logs cannot share the same log stream.
- This request does not take effect immediately. There is a delay of about 10 minutes.

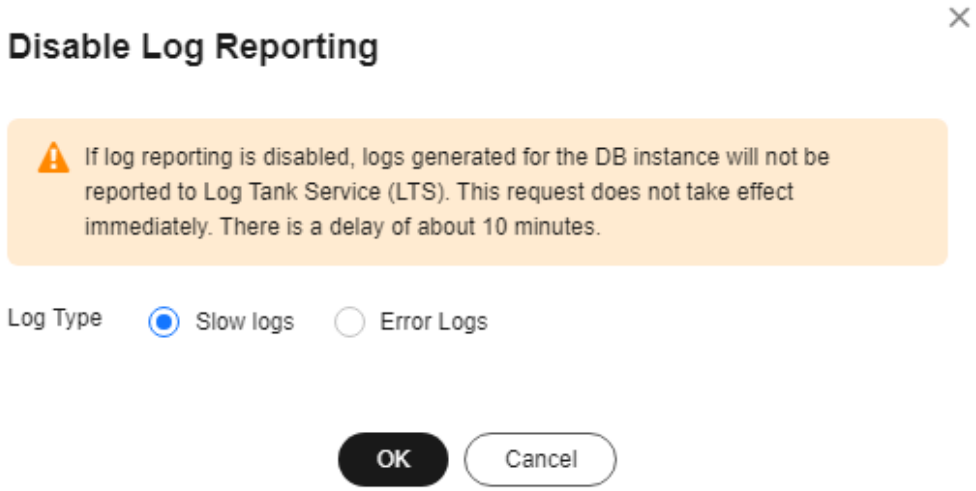
Figure 1-221 Enabling log reporting



Step 7 To disable log reporting, select one or more instances and click **Disable Log Reporting**.

Step 8 In the displayed dialog box, click **OK**.

Figure 1-222 Disabling log reporting



----End

1.16.2 Viewing and Downloading Error Logs

RDS log management allows you to view database-level logs, including error logs and slow SQL query logs.

Error logs help you analyze problems with databases. You can download error logs for further analysis.

You can view error logs generated within the last month.

Viewing Log Details

Step 1 Log in to the management console.

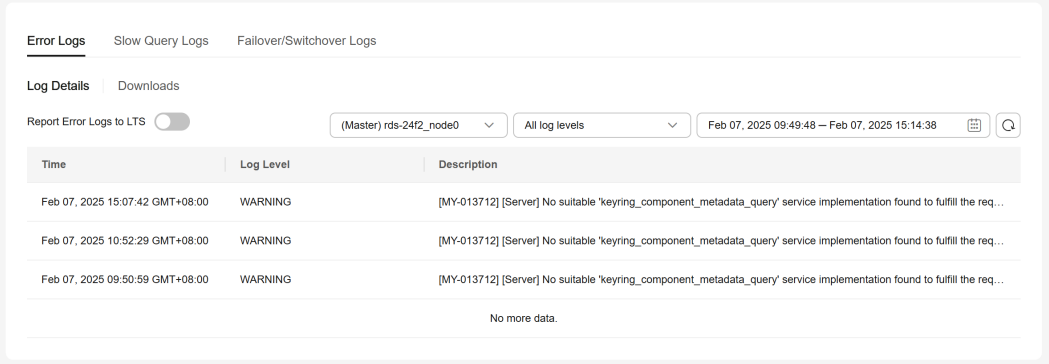
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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.

Figure 1-223 Error log details



- You can select a log level in the upper right corner to view logs of the selected level.

  NOTE

  For RDS for MySQL DB instances, the following levels of logs are displayed:
  - All log levels
  - ERROR
  - WARNING
  - NOTE
- A maximum of 2,000 error log records can be displayed. To view more error log records, **submit a service ticket**.
- You can click in the upper right corner to view 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**.

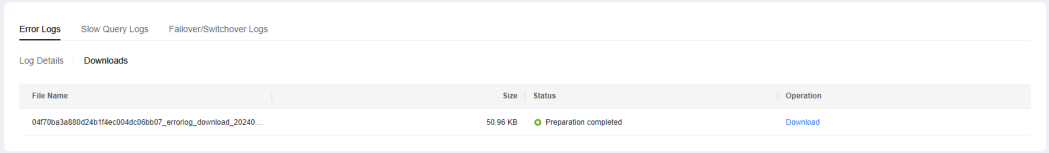
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 1-224 Downloading an error log



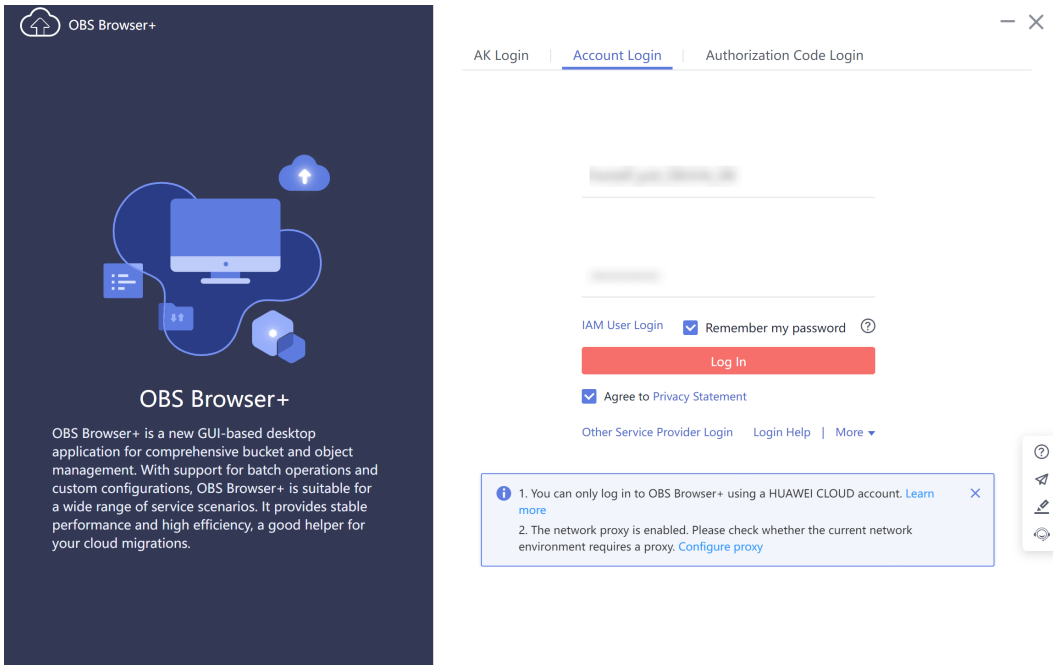
- The system automatically loads the downloading preparation tasks. The loading duration is determined by the log file size and network environment.
  - When the log is being prepared for download, the log status is **Preparing**.
  - When the log is ready for download, the log status is **Preparation 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 redownload the log, click **OK**.
- The downloaded logs contain only the logs of the primary node.
- If the size of a log to be downloaded is greater than 40 MB, you need to use OBS Browser+ to download it.

  a. **Download OBS Browser+**.

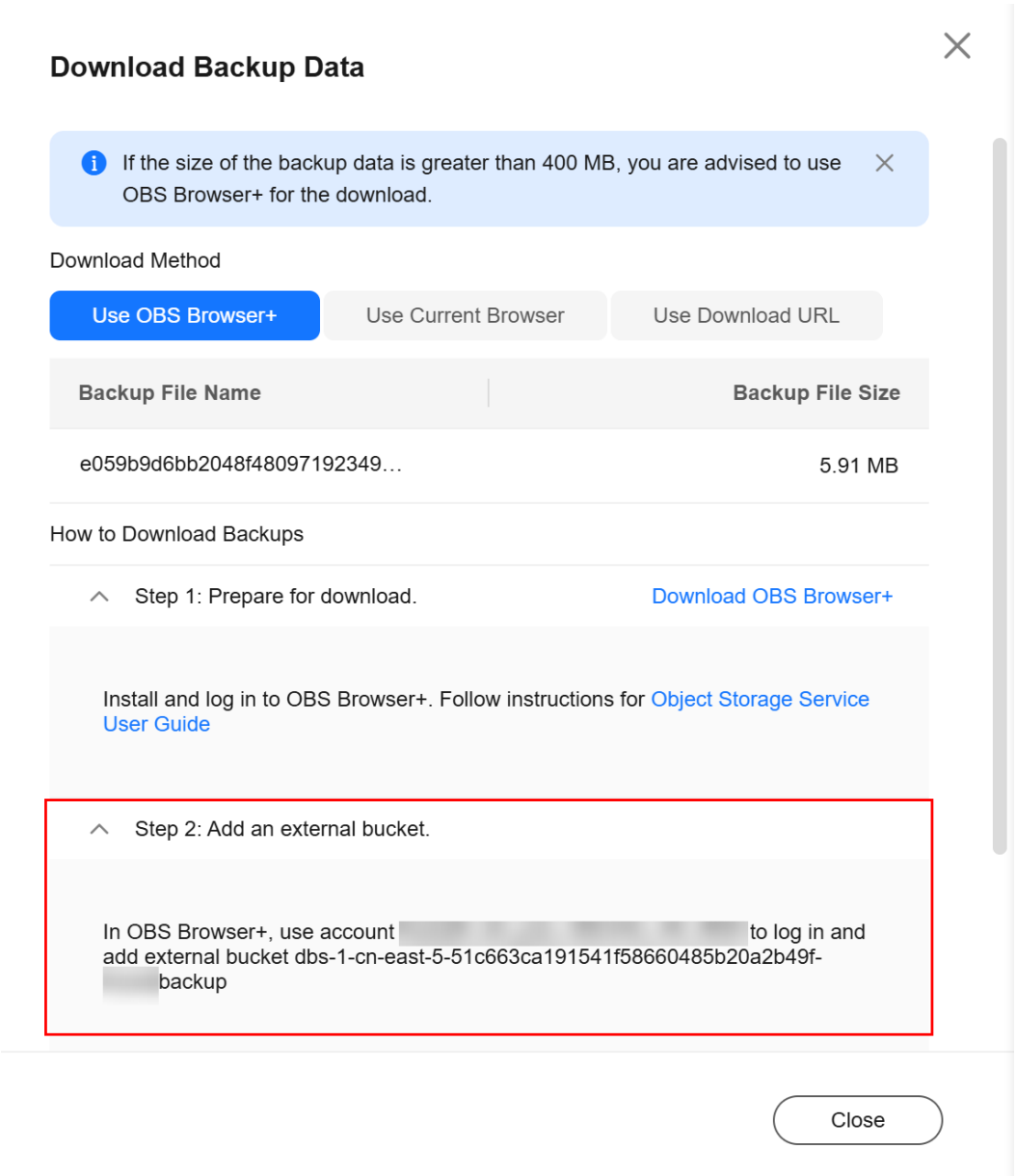
  b. Decompress and install OBS Browser+.

  Figure 1-225 Logging in to OBS Browser+



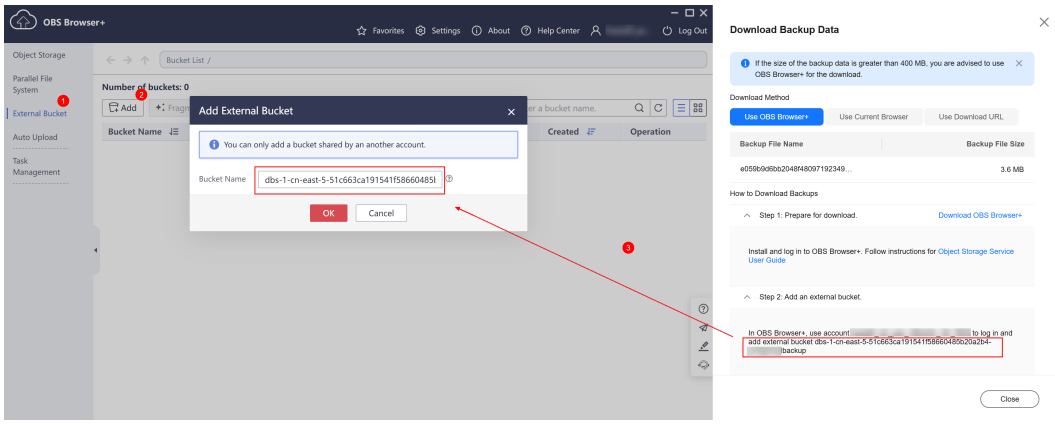
  c. On the OBS Browser+ download page of any backup, obtain the name of an external bucket. For details, see **Method 1: Using OBS Browser+**.

Figure 1-226 Obtaining an external bucket name



d. Add an external bucket.

Figure 1-227 Adding an external bucket

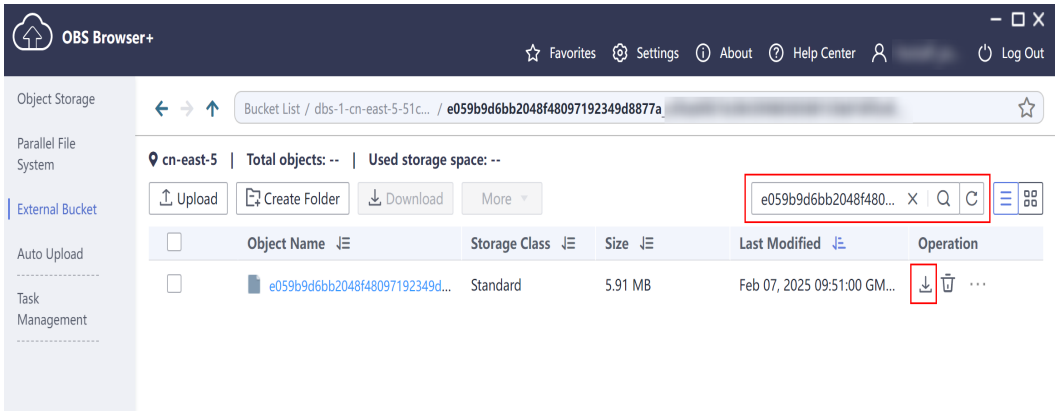


NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

e. Click the name of the external bucket to go to the object list page. In the search box on the right, enter the log file name and start a search. In the search result, click to download the log file.

Figure 1-228 Downloading a log file



----End

Enabling Error Log Reporting to LTS

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

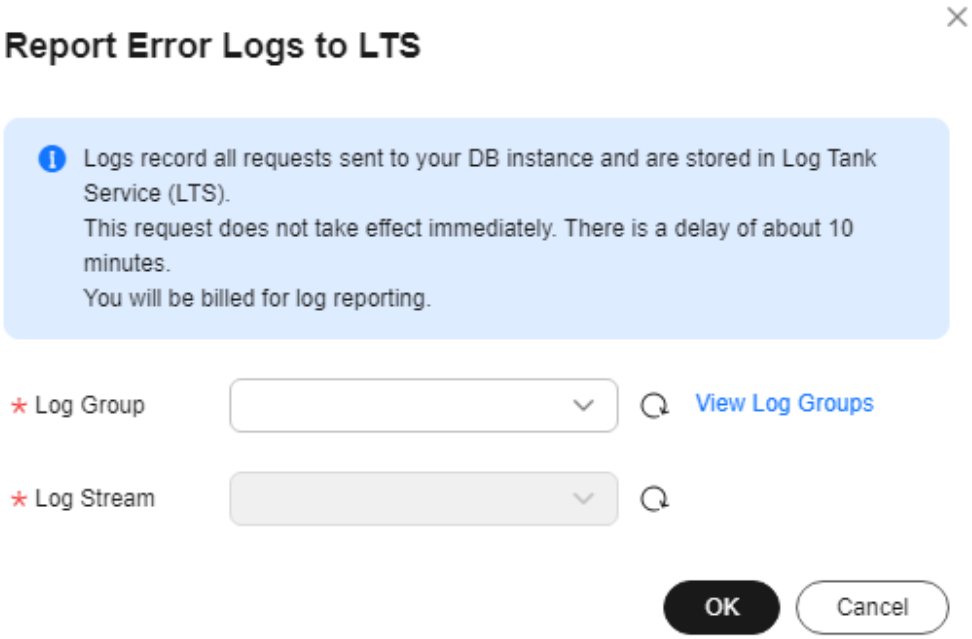
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, click **Logs**. On the **Error Logs** page, click **Log Details**.

Step 6 Click  next to **Report Error Logs to LTS**.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 1-229 Enabling error log reporting to LTS



----End

1.16.3 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed the **long_query_time** value (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.

Slow query logs generated within the last month can be viewed.

RDS supports the following statement types:

- All statement types
- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE

Parameter Description

Table 1-81 Parameters related to slow queries

Parameter	Description
long_query_time	Specifies how many microseconds a SQL query has to take to be defined as a slow query log. The default value is 1s. When the execution time of an SQL statement exceeds the value of this parameter, the SQL statement is recorded in slow query logs. The recommended value is **1s**. Note: The lock wait time is not calculated into the query time.
log_queries_not_using_indexes	Specifies whether to record the slow query without indexes. The default value is **OFF**.
log_throttle_queries_not_using_indexes	Limits the number of SQL statements without indexes per minute that can be written to the slow query log. The default value is **0**.

Showing Original Logs

NOTE

Original logs will be automatically deleted 30 days later. If the instance is deleted, its logs are also deleted.

Step 1 Log in to the management console.

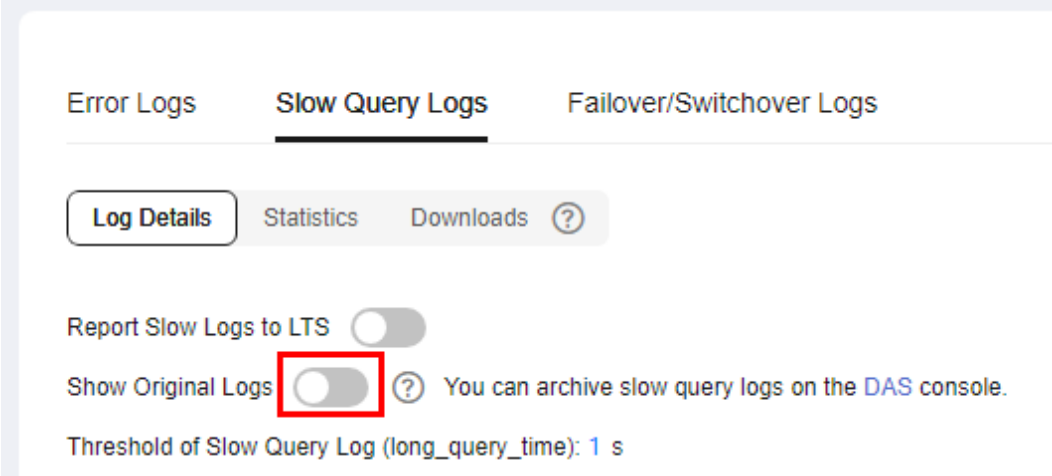
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Log Details** and then click on the right of **Show Original Logs**.

Figure 1-230 Enabling Show Original Logs



Step 6 In the displayed dialog box, click **Yes** to enable the display of original slow query logs.

----End

Viewing Log Details

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

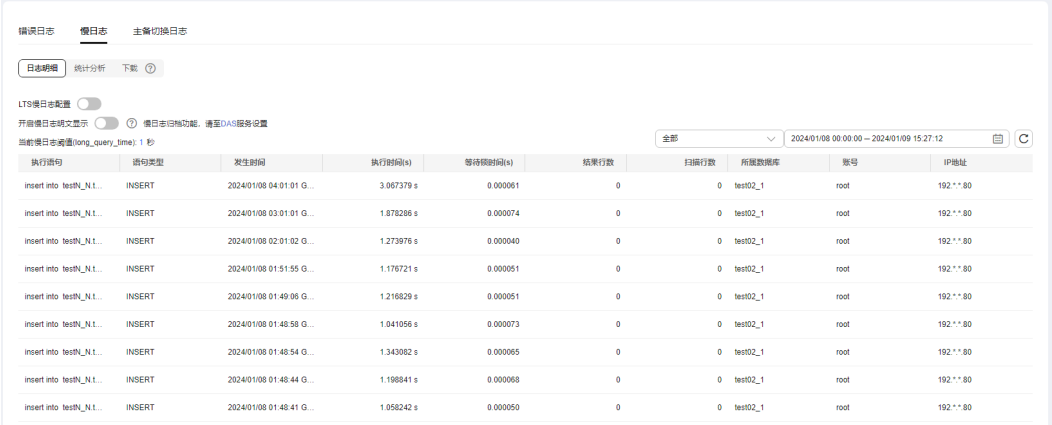
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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.

Figure 1-231 Slow query log details



NOTE

- You can view the slow query log records of a specified execution statement type or a specific time period.
- Only SELECT statements return the number of result rows. The number of result rows for the INSERT, UPDATE, DELETE, and CREATE statements is 0 by default.
- You can view slow query logs of a specified database name (which cannot contain any special characters). The database name supports only exact search.
- Slow query logs only record executed statements whose execution duration exceeds the threshold.
- 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, RDS starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. 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.
- A maximum of 2,000 slow log records can be displayed. To view more slow log records, contact customer service.
- If the length of a single line of an SQL statement exceeds 10 KB or the total number of lines exceeds 200, the SQL statement will be truncated. When you view slow query log details, the SQL statement may be incomplete after special processing and is for reference only.

----End

Viewing Statistics

Step 1 Log in to the management console.

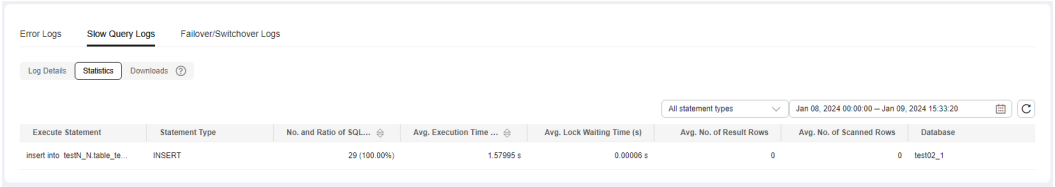
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Statistics** to view details.

Figure 1-232 Statistics



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.
- However, if **Show Original Logs** is enabled, all of the slow SQL statements are displayed. For example, if **select sleep(1)** and **select sleep(2)** are executed in sequence, both of them will be displayed.
- **No. and Ratio of SQL Executions** indicates the ratio of the slow executions to the total executions of the SQL statement.
- On the **Statistics** page, only the latest 5,000 slow SQL statements within a specified period are analyzed.
- You can filter slow log statistics by database name (which cannot contain any special characters), statement type, or time period. The database name supports only exact search.
- If any database name in the slow log statistics contains special characters such as < > ', the special characters will be escaped.

----End

Downloading a Slow Query Log

You can download a log file that is not larger than 40 MB directly **from the console**. The time range is calculated from the time you download the file back to the time when the accumulated file size reaches 40 MB.

To download a log file larger than 40 MB, use **OBS Browser+**.

Downloading a Slow Query Log Directly from the Console

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

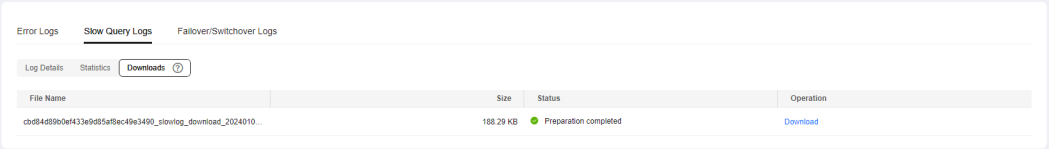
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 1-233 Downloading a slow query log



- 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 redownload the log, click **OK**.
- The downloaded log file contains only the logs of the primary node.

----End

Downloading a Slow Query Log Using OBS Browser+

Step 1 Log in to the management console.

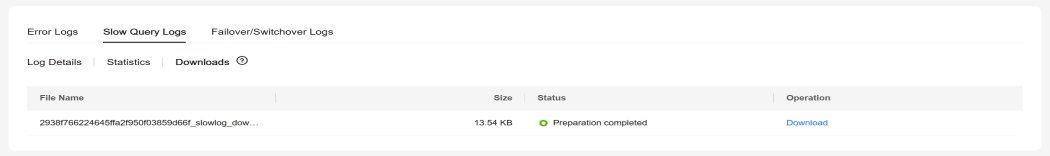
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name to go to the **Overview** page.

Step 5 In the navigation pane, click **Logs**. On the **Slow Query Logs** page, obtain the name of the slow query log file.

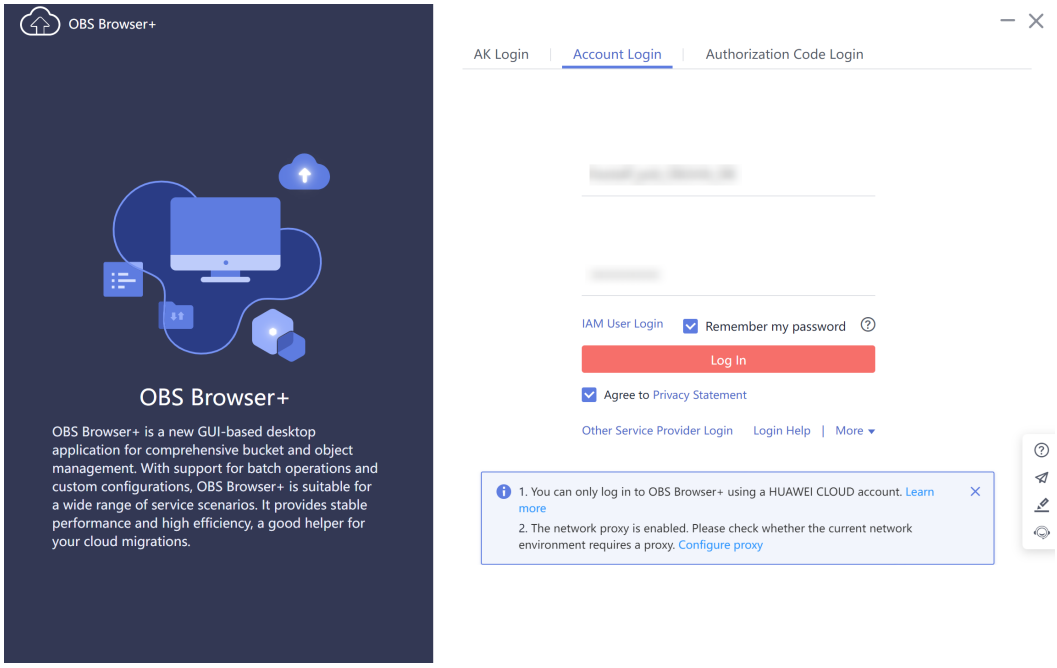
Figure 1-234 Downloading a slow query log



Step 6 Download OBS Browser+.

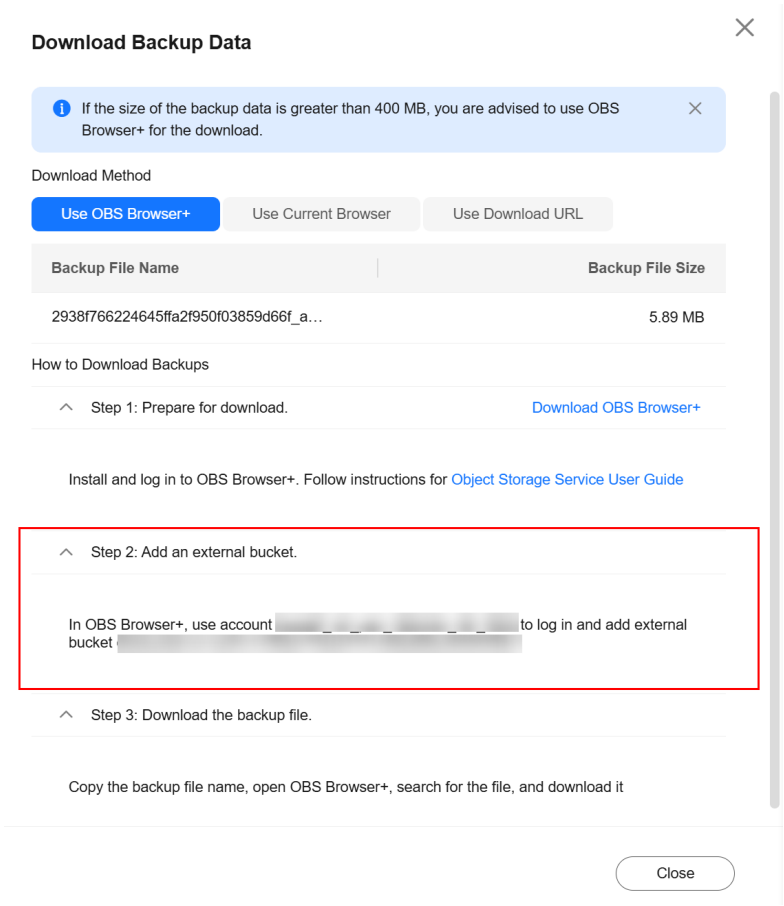
Step 7 Decompress and install OBS Browser+.

Figure 1-235 Logging in to OBS Browser+



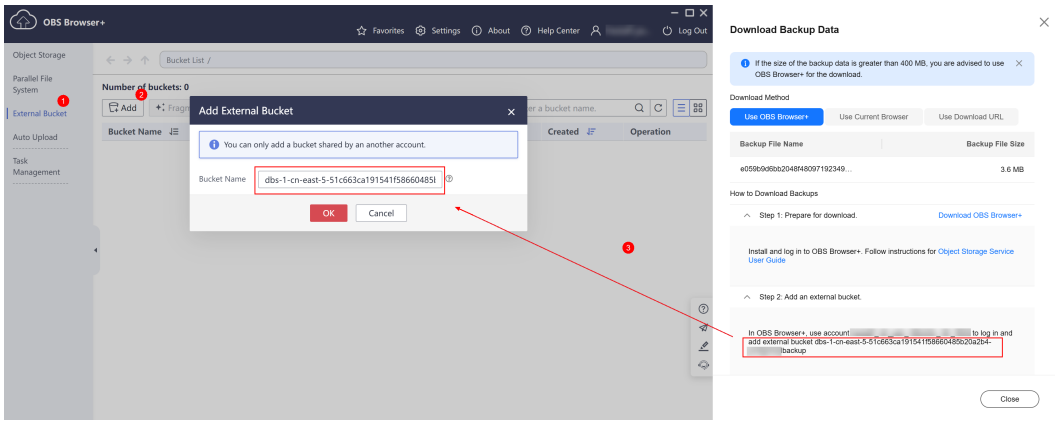
Step 8 On the OBS Browser+ download page of any backup of the current instance, **obtain the name of an external bucket**.

Figure 1-236 Obtaining an external bucket name



Step 9 Add an external bucket.

Figure 1-237 Adding an external bucket

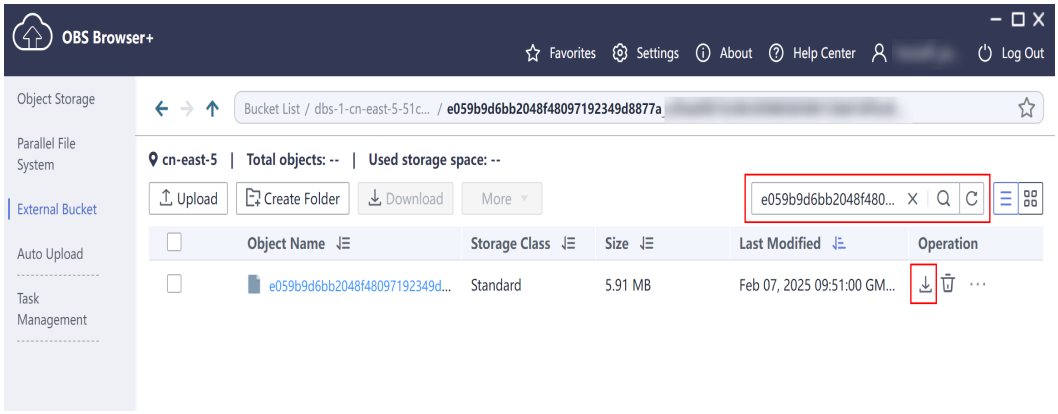


NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

Step 10 Click the name of the external bucket to go to the object list page. In the search box on the right, search for the log file name obtained in **Step 5** and click to download the log file.

Figure 1-238 Downloading a log file



NOTE

The downloaded log file contains only the logs of the primary node.

----End

Enabling Slow Log Reporting to LTS

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

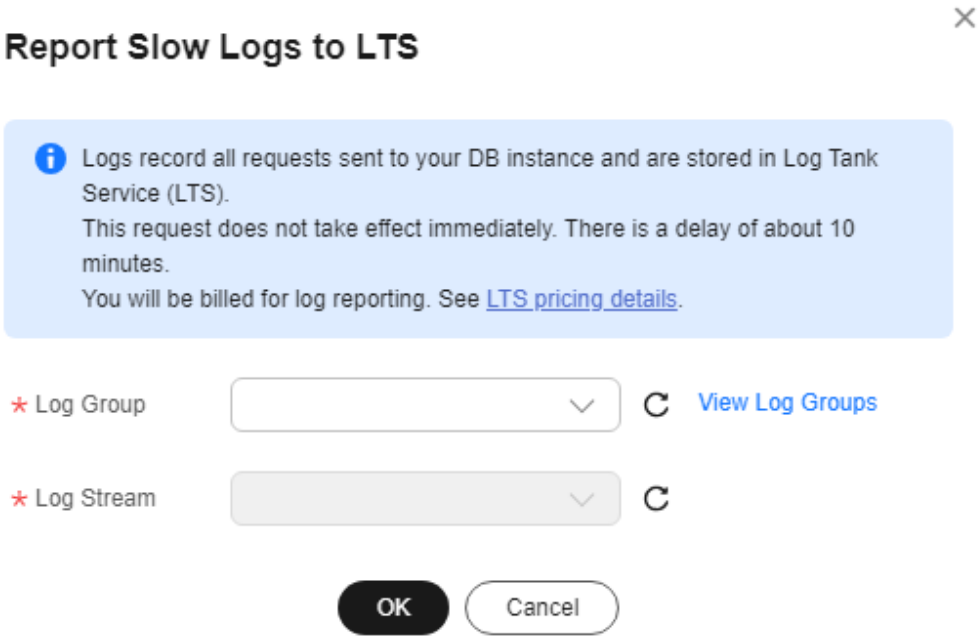
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, click **Logs**. On the **Slow Query Logs** page, click **Log Details**.

Step 6 Click  next to **Report Slow Logs to LTS**.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 1-239 Enabling slow log reporting to LTS



----End

1.16.4 Viewing Failover/Switchover Logs

You can view failover or switchover logs of RDS for MySQL DB instances to evaluate the impact on services.

Precautions

You can query only failover and switchover logs generated within recent 30 days. The logs cannot be dumped to OBS buckets.

Procedure

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

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

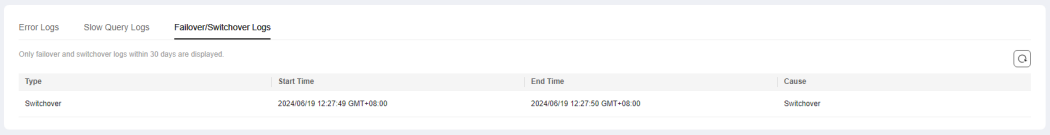
Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the displayed page, click **Failover/Switchover Logs** to view log details.

These logs record the failovers caused by database exceptions and manual switchovers.

Figure 1-240 Failover/Switchover logs



----End

1.16.5 Enabling SQL Audit

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

- Both primary DB instances and read replicas support SQL audit logging.
- Times in audit logs use the Coordinated Universal Time (UTC) format, regardless of the time zone configured for the DB instance.
- After SQL auditing is enabled, RDS records SQL operations in audit logs. The generated audit log files are temporarily stored in the instance and then uploaded to OBS and stored in the backup space. If there is not enough free backup space available for generated audit logs, the additional space required is billed.
- Audit logs are cleared every hour. After you change the retention period of audit logs, expired audit logs will be deleted 1 hour later.
- After SQL auditing is enabled, a large number of audit logs may be generated during peak hours. As a result, there are many audit log files temporarily stored in the instance, and the storage may be full. You are advised to enable **storage autoscaling**.

Supported Database Versions

Only the versions listed below support SQL audit. If your DB engine version is too old, upgrade it to the latest version by referring to **Upgrading the Minor Version**.

- RDS for MySQL 5.6 instances using cloud disks: 5.6.43 and later versions
- RDS for MySQL 5.7 instances using cloud disks: 5.7.23 and later versions
- RDS for MySQL 8.0

Constraints

While a standby read replica is acting as a primary read replica due to an exception in the primary, the audit logs generated are invisible.

If your instance is deployed in LA-Mexico City2 and you want to enable SQL audit for it, you need to **submit a service ticket**.

Enabling SQL Audit

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

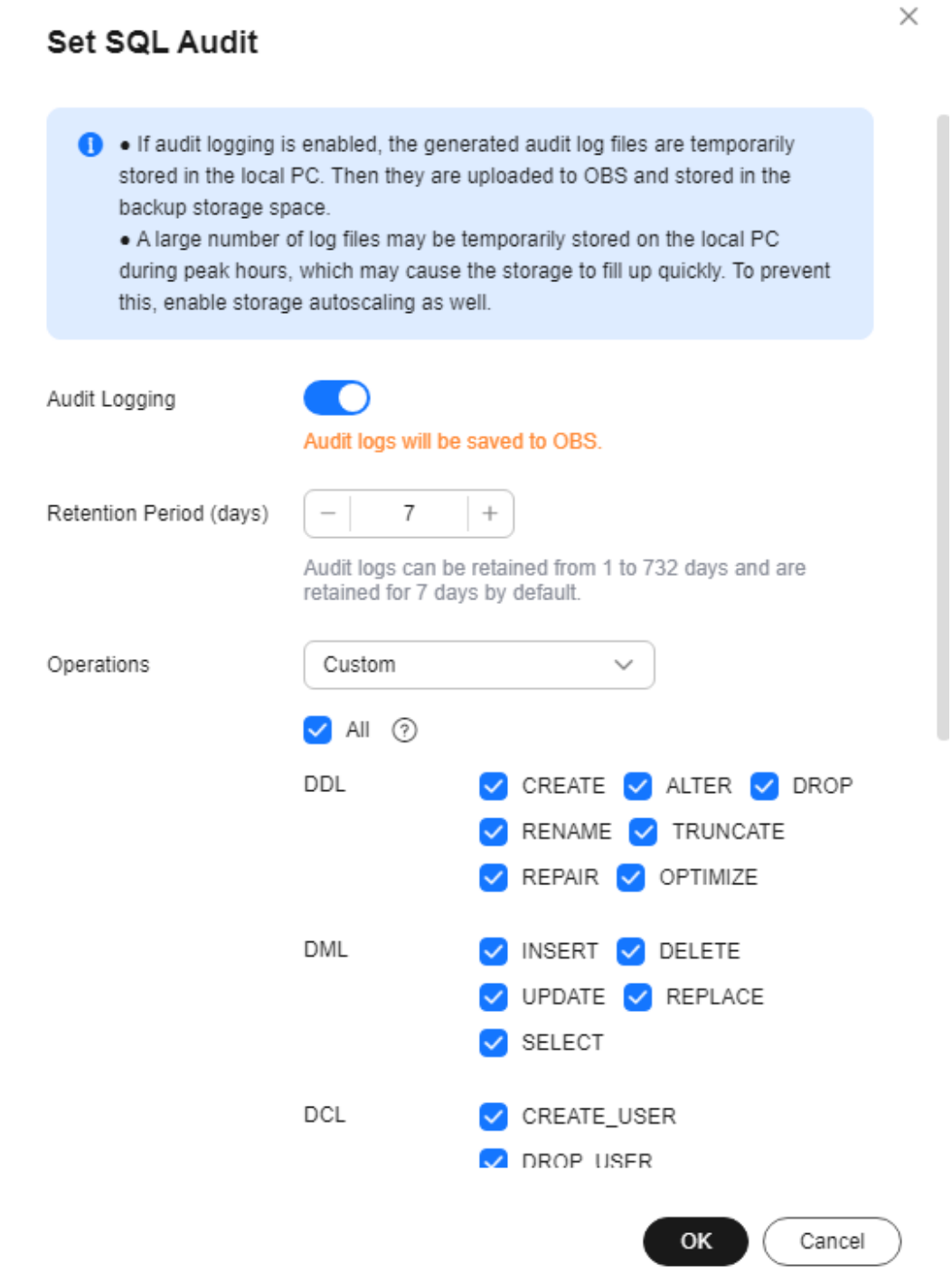
Step 4 On the **Instances** page, click the target instance name.

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 enable SQL audit, toggle  (disabled) to  (enabled).
- Audit logs can be retained from 1 to 732 days and are retained for 7 days by default.

Figure 1-241 Setting SQL audit



NOTE

The **Operations** option is available only in the CN North-Beijing4, CN South-Guangzhou, and CN-Hong Kong regions. To use this option in other regions, **submit a service ticket**.

The SQL statements executed by PreparedStatement and scheduled tasks through a MySQL client will be treated as **PREPARED_STATEMENT** and **CREATE**,

respectively. However, the SQL statements executed by PreparedStatement through JDBC will not be recorded.

After SQL auditing is enabled, Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and other operation types are supported. The details are as follows:

Table 1-82 DDL types and operations

Type	Operation	Remarks
CREATE	create_db, create_event, create_function, create_index, create_procedure, create_table, create_trigger, create_udf, create_view	-
ALTER	alter_db, alter_db_upgrade, alter_event, alter_function, alter_instance, alter_procedure, alter_table, alter_tablespace	The alter_user operation is no longer supported from January 26, 2024.
DROP	drop_db, drop_event, drop_function, drop_index, drop_procedure, drop_table, drop_trigger, drop_view	-
RENAME	rename_table	-
TRUNCATE	truncate	-
REPAIR	repair	Added on January 26, 2024
OPTIMIZE	optimize	Added on January 26, 2024

Table 1-83 DML types and operations

Type	Operation	Remarks
INSERT	insert, insert_select	-
DELETE	delete, delete_multi	-
UPDATE	update, update_multi	The update_multi operation was added on January 26, 2024.
REPLACE	replace, replace_select	-
SELECT	select	-

Table 1-84 DCL types and operations

Type	Operation	Remarks
CREATE_USER	create_user	-
DROP_USER	drop_user	-
RENAME_USER	rename_user	-
GRANT	grant_roles, grant	The grant_roles operation was added on January 26, 2024.
REVOKE	revoke, revoke_all, revoke_roles	The revoke_roles operation was added on January 26, 2024.
ALTER_USER	alter_user	Added on January 26, 2024
ALTER_USER_DEFAULT_ROLE	alter_user_default_role	Added on January 26, 2024

Table 1-85 Other types and operations

Type	Operation	Remarks
BEGIN/COMMIT/ROLLBACK	begin, commit, release_savepoint, rollback, rollback_to_savepoint, savepoint	-
PREPARED_STATEMENT	execute_sql,prepare_sql, dealloc_sql	The dealloc_sql operation was added on January 26, 2024.
CALL_PROCEDURE	call_procedure	Added on January 26, 2024
KILL	kill	Added on January 26, 2024
SET_OPTION	set_option	Added on January 26, 2024
CHANGE_DB	change_db	Added on January 26, 2024
UNINSTALL_PLUGIN	uninstall_plugin	Added on January 26, 2024
INSTALL_PLUGIN	install_plugin	Added on January 26, 2024
SHUTDOWN	shutdown	Added on January 26, 2024

Type	Operation	Remarks
SLAVE_START	slave_start	Added on January 26, 2024
SLAVE_STOP	slave_stop	Added on January 26, 2024
LOCK_TABLES	lock_tables	Added on January 26, 2024
UNLOCK_TABLES	unlock_tables	Added on January 26, 2024
FLUSH	flush	Added on January 26, 2024
XA	xa_commit,xa_end,xa_prepare,xa_recover,xa_rollback,xa_start	Added on January 26, 2024

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." and click **OK**, all audit logs will be deleted.

NOTICE

Deleted audit logs cannot be recovered. Exercise caution when performing this operation.

----End

Enabling Audit Log Reporting to LTS

NOTICE

- To enable audit log reporting to LTS, **submit a service ticket** to apply for the required permissions.
- After this function is enabled, audit logs record all requests sent to your DB instance and are stored in LTS.
- This function does not take effect immediately. There is a delay of about 10 minutes.
- You will be billed for enabling this function. For details, see LTS **pricing details**.
- After this function is enabled, all audit policies are reported by default.
- Keep the following points in mind before you enable audit logging or audit log reporting to LTS:
  - Enabling audit logging or audit log reporting to LTS generates audit logs and the sensitive information in the audit logs is not anonymized.
  - If you enable audit logging first and then enable audit log reporting to LTS, LTS reuses the audit policy set for your instance and you will also be billed for reporting audit logs to LTS. Only after you disable audit logging, billing for audit logging will be terminated.
  - If you enable audit logging first and then enable audit log reporting to LTS, you are not advised to disable audit logging before audit log reporting to LTS is running properly.
- Audit logs uploaded to LTS may be lost in the scenarios described below. If audit logging is enabled, you can download all audit log files from OBS.
  - There is a low probability that some logs are lost when the service traffic is heavy, audit logs are generated too fast, or the LTS service fails.
  - The maximum size of a single audit log record that can be uploaded to LTS is 512 KB. If the size of an audit log record exceeds this limit, the audit log record will be truncated.

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

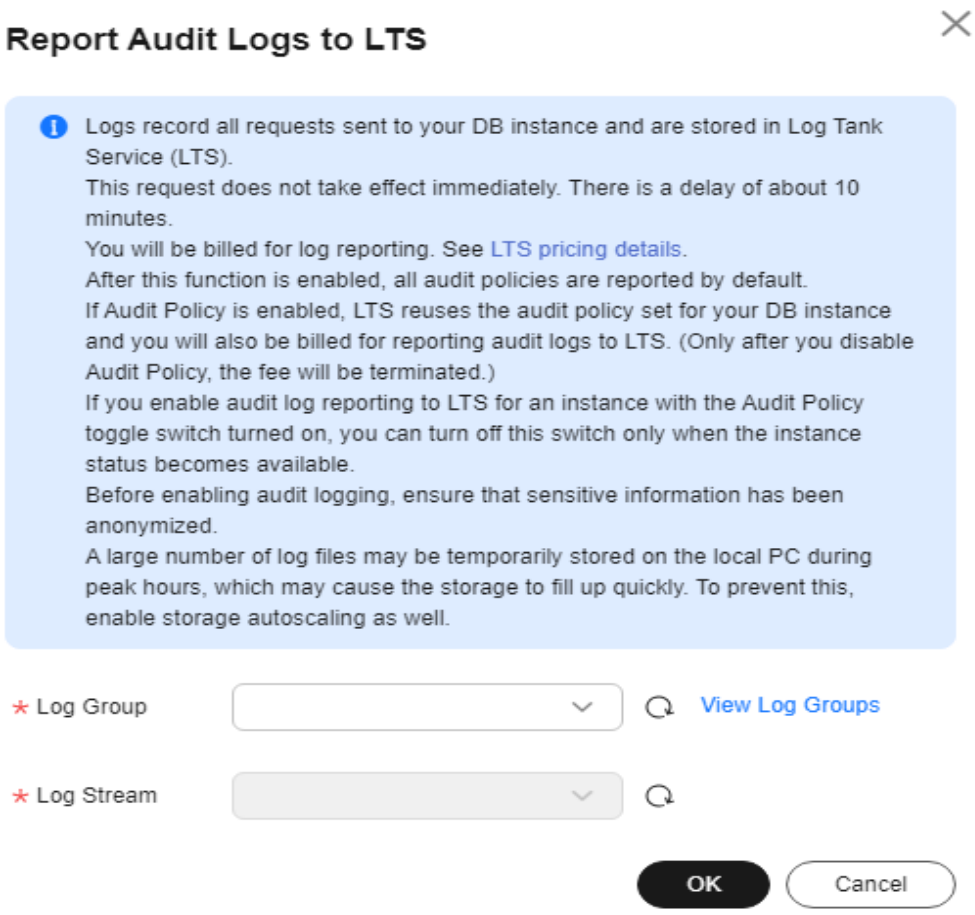
Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane, choose **SQL Audits**.

Step 6 Click next to the **Report Audit Logs to LTS** field.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 1-242 Enabling audit log reporting to LTS



----End

1.16.6 Downloading SQL Audit Logs

If you **enable SQL audit**, the system records all SQL operations and uploads logs every half an hour or when the size is accumulated to 100 MB. You can download audit logs to view details. The minimum time unit of audit logs is second. By default, SQL audit is disabled. Enabling this function may affect database performance.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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.

NOTE

You are advised to download no more than six audit log files at a time. Too many files can fail to be downloaded completely due to the limit on the number of concurrent requests of the browser.

Step 7 The following figure shows the SQL audit log content. For field descriptions, see **Table 1-86**.

Figure 1-243 RDS for MySQL audit logs

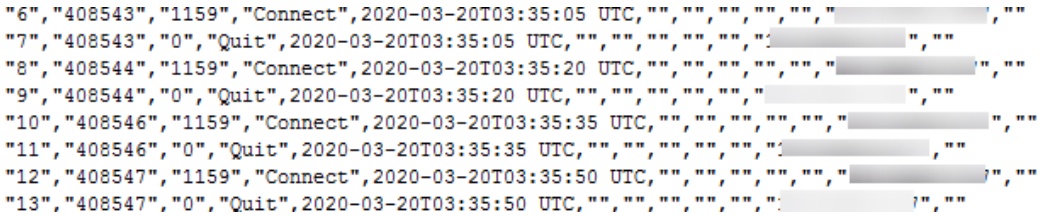


Table 1-86 Audit log field description

Parameter	Description
record_id	ID of a single record, which is the unique global ID of each SQL statement recorded in the audit log.
connection_id	ID of the session executed for 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	UTC time for the record.
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 connection is disconnected.)
user	Login account.

Parameter	Description
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. For local connection, the field is empty.
default_db	Default database on which SQL statements are executed.

----End

1.17 Metrics and Alarms

1.17.1 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.

The monitoring interval can be 1 minute, 1 second, or 5 seconds. The default monitoring interval is 1 minute. To enable Monitoring by Seconds, see **Configuring Monitoring by Seconds**.

Namespace

- Namespace of RDS for MySQL single or primary/standby instance metrics: SYS. RDS
- Namespace of database proxy metrics: SYS.DBPROXY

DB Instance Monitoring Metrics

Table 1-87 lists the performance metrics of RDS for MySQL instances.

Table 1-87 Performance metrics

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100	%	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0–100	%	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds006_conn_count	Total Connections	Total number of connections that attempt to connect to the MySQL server	≥ 0	counts	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds007_conn_active_count	Current Active Connections	Number of connections not in the sleep state	≥ 0	counts	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds008_qps	QPS	Query times of SQL statements (including stored procedures) per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds009_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0	transactions/s	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
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	ratio	N/A	RDS for MySQL instance	1 minute
rds011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0–1	ratio	N/A	RDS for MySQL instance	1 minute
rds012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to used pages in the InnoDB buffer	0–1	ratio	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds014_innodb_writes	InnoDB Write Throughput	Number of write bytes per second in the InnoDB buffer	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds015_innodb_read_count	InnoDB File Read Frequency	Number of times that InnoDB reads data from files per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds016_innodb_write_count	InnoDB File Write Frequency	Number of times that InnoDB writes data to files per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds017_innodb_log_write_req_count	InnoDB Log Write Requests per Second	Number of InnoDB log write requests per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds018_innodb_log_write_count	InnoDB Log Physical Write Frequency	Number of InnoDB physical write times to log files per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds019_innodb_log_fsync_count	InnoDB Log fsync() Write Frequency	Number of completed fsync() write times to InnoDB log files per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds020_temp_tbl_rate	Temporary Tables Created per Second	Number of temporary tables created on hard disks per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds021_myisam_buf_usage	Key Buffer Usage	MyISAM key buffer usage	0–1	ratio	N/A	RDS for MySQL instance	1 minute
rds022_myisam_buf_write_hit	Key Buffer Write Hit Ratio	MyISAM key buffer write hit ratio	0–1	ratio	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds023_myisam_buf_read_hit	Key Buffer Read Hit Ratio	MyISAM key buffer read hit ratio	0–1	ratio	N/A	RDS for MySQL instance	1 minute
rds024_myisam_disk_write_count	MyISAM Disk Write Frequency	Number of times that indexes are written to disks per second	≥ 0	counts/s	N/A	RDS for 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	N/A	RDS for 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	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
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	N/A	RDS for MySQL instance	1 minute
rds028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute
rds032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute
rds033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0	queries/s	N/A	RDS for MySQL instance	1 minute 5 seconds 1 second
rds035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥ 0	rows/s	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥ 0	rows/s	N/A	RDS for MySQL instance	1 minute
rds037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥ 0	rows/s	N/A	RDS for MySQL instance	1 minute
rds038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥ 0	rows/s	N/A	RDS for MySQL instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0–100	%	N/A	RDS for MySQL instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4000	GB	1024	RDS for MySQL instance	1 minute
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4000	GB	1024	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds072_conn_usage	Connection Usage	Percent of used MySQL connections to the total number of connections	0–100	%	N/A	RDS for MySQL instance	1 minute
rds173_replication_delay_avg	Average Replication Delay	Average replication delay within 60s between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master.	≥ 0	s	N/A	RDS for MySQL instance	10 seconds

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds073_replication_delay	Real-Time Replication Delay	Real-time replication delay between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master.	≥ 0	s	N/A	RDS for MySQL instance	1 minute 5 seconds
rds074_slow_queries	Slow Query Logs	Number of slow query logs generated per minute by MySQL	≥ 0	counts/min	N/A	RDS for MySQL instance	1 minute
rds075_avg_disk_ms_per_read	Disk Read Time	Average time required for each disk read in a specified period	≥ 0	ms	N/A	RDS for MySQL instance	1 minute
rds076_avg_disk_ms_per_write	Disk Write Time	Average time required for each disk write in a specified period	≥ 0	ms	N/A	RDS for MySQL instance	1 minute
rds077_vma	VMA	Virtual memory area size of an RDS process	≥ 0	counts	N/A	RDS for MySQL instance	1 minute
rds078_threads	Threads	Number of threads in a process	≥ 0	counts	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds079_vm_hwm	Peak Resident Set Size	Peak physical memory usage of an RDS process	≥ 0	KB	1024	RDS for MySQL instance	1 minute
rds080_vm_peak	Peak Virtual Memory Size	Peak virtual memory usage of an RDS process	≥ 0	KB	1024	RDS for MySQL instance	1 minute
rds081_vm_ioutils	The time percentage of disk I/O in a non-idle state	Percentage of time when the disk is not idle (there is I/O activity). This parameter indicates how busy the disk is. The disk can process I/O requests in parallel. If the value of this parameter reaches 100%, the disk may not reach its maximum processing capability.	0–100	%	N/A	RDS for MySQL instance	1 minute
rds082_semi_sync_tx_avg_wait_time	Transaction Wait Time	Average wait time of transactions in semi-synchronous mode	≥ 0	μs	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
sys_swap_usage	SWAP Usage	SWAP usage of the monitored object	0–100	%	N/A	RDS for MySQL instance	1 minute
rds_innodb_lock_waits	Row Locks Waits Transactions	Number of InnoDB transactions waiting for row lock	≥ 0	counts	N/A	RDS for MySQL instance	1 minute
rds_bytes_recv_rate	Received Bytes per Second	Number of bytes received by the database per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds_bytes_sent_rate	Sent Bytes per Second	Number of bytes sent from the database per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds_innodb_pages_read_rate	Data Volume Read By InnoDB per Second	Data volume read by InnoDB per second	≥ 0	pages/s	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_innodb_pages_written_rate	Data Volume Written by InnoDB per Second	Data volume written by InnoDB per second	≥ 0	pages/s	N/A	RDS for MySQL instance	1 minute
rds_innodb_os_log_written_rate	Redo Log Size Written per Second	Size of redo logs written per second	≥ 0	bytes/s	N/A	RDS for MySQL instance	1 minute
rds_innodb_buffer_pool_read_requests_rate	Innodb_buffer_pool Read Requests per Second	Number of innodb_buffer_pool read requests per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds_innodb_buffer_pool_write_requests_rate	Innodb_buffer_pool Write Requests per Second	Number of innodb_buffer_pool write requests per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_innodb_buffer_pool_pages_flushed_rate	Innodb_buffer_pool Page Flushes per Second	Number of innodb_buffer_pool page flushes per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds_innodb_log_waits_rate	Flush Times to Disks Due to Insufficient Log Buffer	Times of transaction logs flushed to disks due to insufficient log buffer	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds_created_tmp_tables_rate	Temporary Tables Created per Second	Number of temporary tables created per second	≥ 0	counts/s	N/A	RDS for MySQL instance	1 minute
rds_wait_thread_count	Waiting Threads	Total number of waiting threads in an instance.	≥ 0	counts	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_threadpool_waiting_threads	Number of Waiting Threads in the Thread Pool	Number of waiting threads in the thread pool	≥ 0	counts	N/A	RDS for MySQL instance	1 minute
rds_innodb_row_lock_time_avg	Average Row Lock Wait Time	Average wait time of historical InnoDB row locks	> 0	ms	N/A	RDS for MySQL instance	1 minute
rds_innodb_row_lock_current_waits	Current Row Lock Waits	Number of current InnoDB row lock waits This metric indicates the number of transactions that are currently waiting for row locks.	≥ 0	counts	N/A	RDS for MySQL instance	1 minute
rds_mdl_lock_count	MDL Locks	Number of MDL locks	≥ 0	counts	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_buffer_pool_wait_free	Dirty Pages to Be Flushed to Disks	When InnoDB needs to read or create a page and no clean pages are available, InnoDB flushes some dirty pages first and waits for that operation	≥ 0	counts	N/A	RDS for MySQL instance	1 minute
rds_conn_active_usage	Active Connection Usage	Usage of active connections	0–100	%	N/A	RDS for MySQL instance	1 minute
rds_innodb_log_waits_count	Log Waits	Number of times that the log buffer was too small and a wait was required for it to be flushed before continuing The value is an accumulated value and increases by 1 each time a wait occurs.	≥ 0	counts	N/A	RDS for MySQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_long_transaction	Long Transaction	Maximum duration for starting a transaction A complete long transaction is counted only when the **BEGIN** and **COMMIT** commands exist before and after the related operation commands, respectively.	≥ 0	seconds	N/A	RDS for MySQL instance	1 minute

Table 1-88 lists the metrics of RDS for MySQL database proxy.

Table 1-88 RDS for MySQL database proxy metrics

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100	%	N/A	RDS for MySQL proxy instance	1 minute 5 seconds 1 second
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0–100	%	N/A	RDS for MySQL proxy instance	1 minute 5 seconds 1 second

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0	bytes/s	1024 (IEC)	RDS for MySQL proxy instance	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0	bytes/s	1024 (IEC)	RDS for MySQL proxy instance	1 minute
rds_proxy_frontend_connections	Frontend Connections	Number of connections between applications and the proxy	≥ 0	counts	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_backend_connections	Backend Connections	Number of connections between the proxy and RDS database	≥ 0	counts	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_average_response_time	Average Response Time	Average response time	≥ 0	ms	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_query_per_seconds	QPS	Query times of SQL statements	≥ 0	counts	N/A	RDS for MySQL proxy instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds_proxy_read_query_proportions	Read Proportion	Proportion of read requests to total requests	0–100	%	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_write_query_proportions	Write Proportion	Proportion of write requests to total requests	0–100	%	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_frontend_connection_creation	Front-End Connections Created per Second	Number of connections created per second between the database proxy and applications	≥ 0	counts/s	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_transaction_query	Transaction Queries per Second	Number of SELECT statements executed in transactions per second	≥ 0	counts/s	N/A	RDS for MySQL proxy instance	1 minute
rds_proxy_multi_statement_query	Multi-Statement Queries per Second	Number of multi-statements executed in transactions per second	≥ 0	counts/s	N/A	RDS for MySQL proxy instance	1 minute

Dimension

Key	Value
rds_cluster_id	RDS for MySQL DB instance ID
dbproxy_instance_id	RDS for MySQL proxy instance ID
dbproxy_node_id	RDS for MySQL proxy node ID

1.17.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS metrics on the management console. For details, see **Viewing Metrics of DB Instances**.

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.

Viewing Metrics of DB Instances

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** in the upper right corner of the page.

Step 5 On the displayed page, view the instance monitoring metrics.

- On the Cloud Eye console, click **Select Metric** in the upper right corner. In the displayed dialog box, you can select the metrics to be displayed and sort them by dragging them to desired locations.
- You can sort graphs by dragging them based on service requirements.
- You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, 7 days, and 6 months.

----End

Real-Time Monitoring

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab to view real-time monitoring data such as CPU usage, memory usage, and storage space usage.

You can also click **View details** to view more metrics on the Cloud Eye console.

----End

1.17.3 Setting Alarm Rules

Scenarios

You can set 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 in the upper left corner and select a region and a project.

Step 3 Click **Service List**. Under **Management & Governance**, click **Cloud Eye**.

Step 4 In the navigation pane, choose **Alarm Management** > **Alarm Rules**.

Step 5 Click **Create Alarm Rule** in the upper right corner.

Step 6 On the displayed page, configure required parameters.

Figure 1-244 Configuring alarm rule information

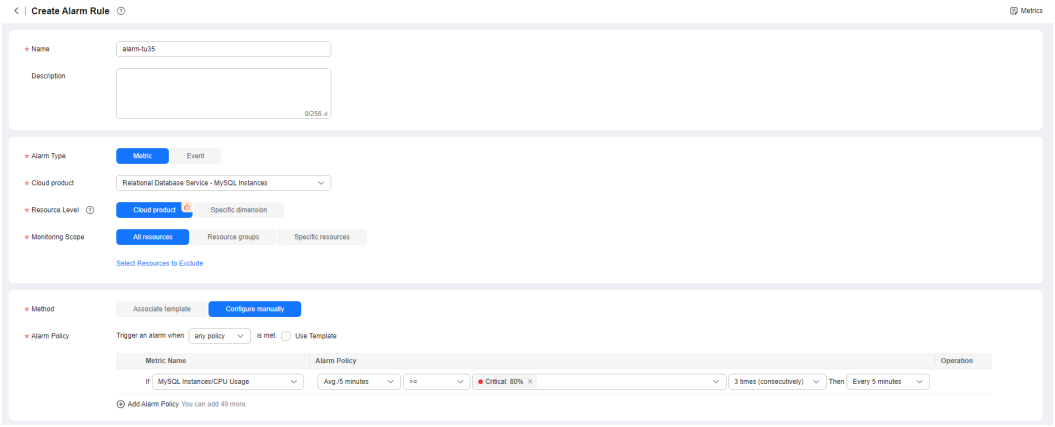


Table 1-89 Alarm rule information

Parameter	Description
Name	Alarm rule name. The system generates a random name, which you can modify.
Description	Description about the rule.
Alarm Type	Select **Metric**.
Cloud Product	Select **Relational Database Service - MySQL Instances**.
Resource Level	**Cloud product** is recommended.
Monitoring Scope	• **All resources**: An alarm will be triggered if any resource of the current cloud product meets the alarm policy. To exclude resources that do not require monitoring, click **Select Resources to Exclude**. • **Resource groups**: An alarm will be triggered if any resource in the selected resource group meets the alarm policy. • **Specific resources**: Click **Select Specific Resources** to select resources.
Method	• **Associate template**: After an associated template is modified, the policies contained in this alarm rule to be created will be modified accordingly. You are advised to select **Use existing template**. The existing templates already contain three common alarm metrics: CPU usage, memory usage, and storage space usage. • **Configure manually**: Configure alarm policies manually.
Template	If you select **Associate template** for **Method**, you need to select a template. You can select a default alarm template or create a custom template.

Parameter	Description
Alarm Policy	If you select **Configure manually** for **Method**, you need to configure alarm policies. Whether to trigger an alarm depends on whether the metric data in consecutive periods reaches the threshold. For example, Cloud Eye triggers an alarm every 5 minutes if the average CPU usage of the monitored object is 80% or more for three consecutive 5-minute periods. **NOTE** A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm is triggered.
Alarm Severity	The alarm severity can be **Critical**, **Major**, **Minor**, or **Informational**.

Figure 1-245 Configuring alarm notification

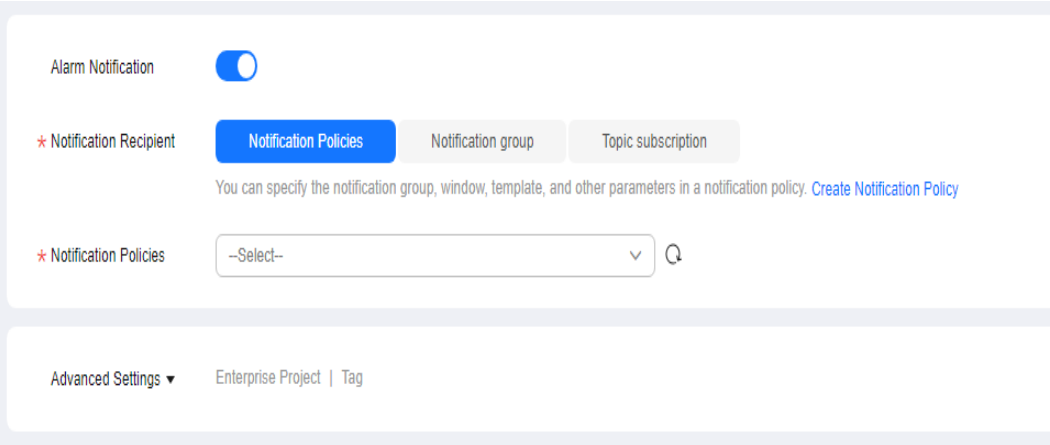


Table 1-90 Alarm notification

Parameter	Description
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Recipient	You can select a notification group or topic subscription as required.
Notification Group	Notification group the alarm notification is to be sent to.
Notification Object	Object the alarm notification is to be sent to. You can select the account contact or a topic. • The account contact is the mobile phone number and email address of the registered account. • A topic is used to publish messages and subscribe to notifications.

Parameter	Description
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.
Tag	A tag is a key-value pair. Tags identify cloud resources so that you can easily categorize and search for your resources.

Step 7 Click **Create**. The alarm rule is created.

For details about how to create alarm rules, see **Creating an Alarm Rule** in the *Cloud Eye User Guide*.

----End

1.17.4 Configuring Alarm Reporting

Scenarios

If you want to be notified when CPU and disk usage thresholds are reached, enable alarm reporting for RDS for MySQL DB instances in an enterprise project. Once alarm reporting is enabled, the DB instances you create after this function is enabled are automatically added to Cloud Eye. If a configured threshold is reached, you will see an alarm reported on Cloud Eye and be notified by a text message or email sent from Simple Message Notification (SMN).

To view or modify the DB instances with alarm reporting enabled, go to the Cloud Eye console. For details, see **Modifying an Alarm Rule**.

To view or modify the phone number or email address, open the details page of the alarm rule on Cloud Eye and click the topic under **Alarm Notifications**. For details, see **Adding a Subscription**. You will be billed for the notifications sent using SMN. For details, see **Product Pricing Details**.

Supported Regions

Alarm reporting is supported in the following regions: CN North-Beijing2, CN North-Beijing4, CN East-Shanghai1, CN South-Guangzhou, CN Southwest-Guiyang1, CN-Hong Kong, CN North-Ulanqab201, and CN North-Ulanqab202.

Alarm Policy

- By default, the following metrics are supported for alarm reporting: CPU Usage (rds001_cpu_util), Storage Space Usage (rds039_disk_util), and

Connection Usage (rds072_conn_usage). For more metrics, see **Configuring Displayed Metrics**.

- When you enable alarm reporting, the phone number and email of your Huawei Cloud account are bound by default.
- After alarm reporting is enabled, new instances are automatically added to the alarm monitoring resource list. If a configured threshold is reached, you will see an alarm reported on Cloud Eye and be notified by a text message or email sent from SMN.
- To add a DB instance to or remove it from the alarm monitoring resource list, go to the Cloud Eye console. For details, see **Modifying an Alarm Rule**.

Configuring Alarm Reporting

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

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

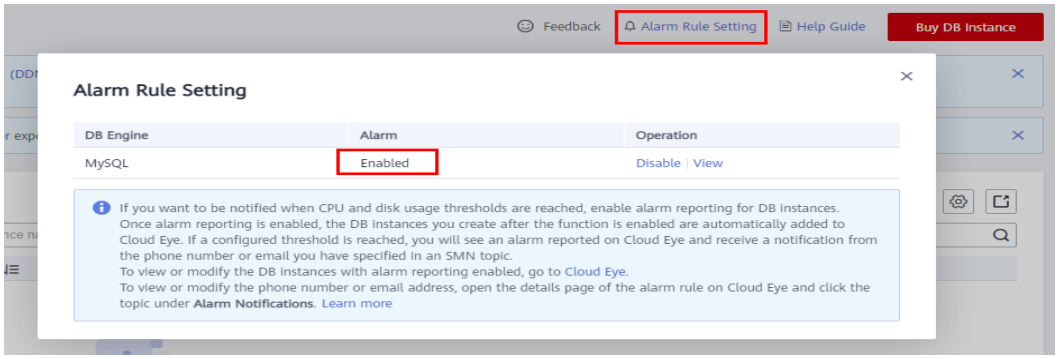
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Click **Alarm Rule Setting** in the upper right corner.

Step 5 In the displayed dialog box, configure alarm reporting.

- To enable or disable alarm reporting, click **Enable** or **Disable** in the **Operation** column.
- To view alarm details, click **View** in the **Operation** column.

Figure 1-246 Configuring alarm reporting



----End

1.17.5 Configuring Monitoring by Seconds

RDS for MySQL supports monitoring by seconds. You can set the monitoring interval to 1 second or 5 seconds to view the metric values.

Constraints

DB instances with fewer than four vCPUs do not support monitoring by seconds.

Billing

Table 1-91 Pay-per-use price

Specification Code	Price (USD/Hour)
rds.mysql.second.monitor	0.012

Enabling Monitoring by Seconds

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

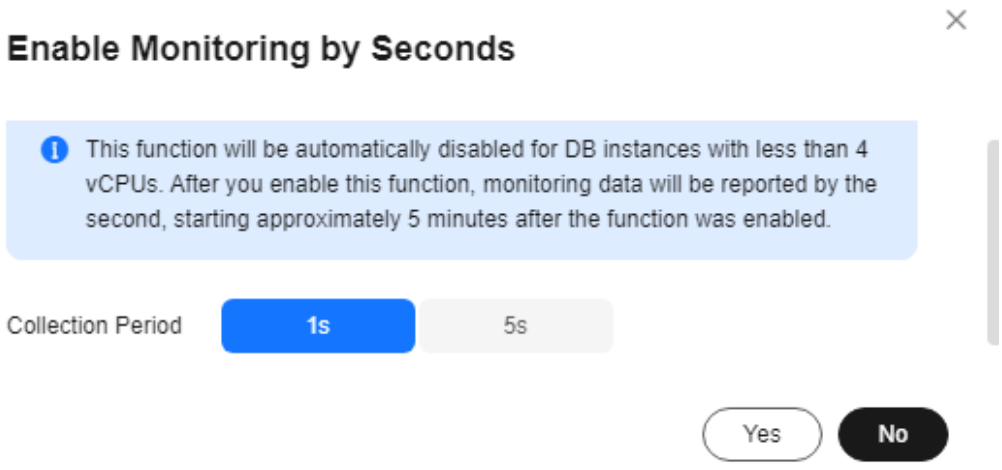
Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab and click next to **Monitoring by Seconds**.

Step 7 In the displayed dialog box, select a collection period and click **Yes**.

After you enable this function, monitoring data will be reported again and will be displayed by the second, starting approximately 5 minutes after the function was enabled.

Figure 1-247 Enabling Monitoring by Seconds



----End

Disabling Monitoring by Seconds

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

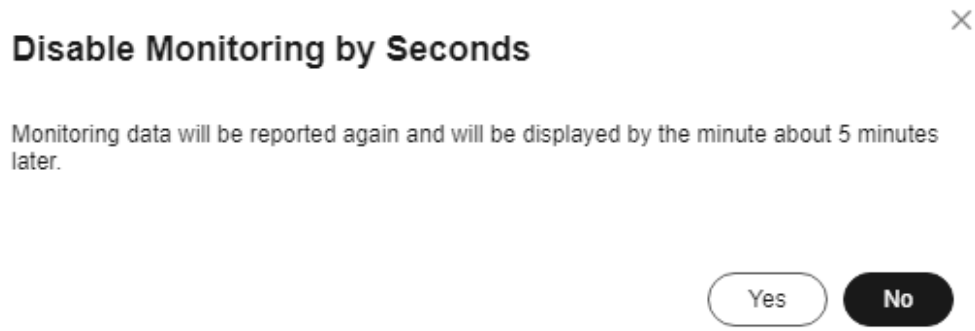
Step 5 In the navigation pane on the left, choose **Advanced O&M**.

Step 6 On the displayed page, click the **Real-Time Monitoring** tab and click next to **Monitoring by Seconds**.

Step 7 In the displayed dialog box, click **Yes**.

After you disable this function, monitoring data will be reported again and will be displayed by the minute, starting approximately 5 minutes after the function was disabled.

Figure 1-248 Disabling Monitoring by Seconds



----End

1.17.6 Event Monitoring

1.17.6.1 Introduction to Event Monitoring

Event monitoring provides event data reporting, query, and alarm reporting. You can create alarm rules for both system and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on RDS resources that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, for example, resetting the administrator password or modifying the backup policy.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see **Events Supported by Event Monitoring**.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by services to Cloud Eye.

For details about how to report custom events, see **Reporting Events**.

1.17.6.2 Viewing Event Monitoring Data

Scenarios

This section describes how to view the event monitoring data.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using the following method:

On the **Instances** page, click the DB instance name. On the displayed **Overview** page, click **View Metrics** in the upper right corner.

Step 5 Click  to return to the main page of Cloud Eye.

Step 6 In the navigation pane on the left, choose **Event Monitoring**.

On the displayed **Event Monitoring** page, all system events generated in the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view the events generated in different periods.

Step 7 Click **View Graph**. On the details page, click **View Event** in the **Operation** column of a specific event to view details.

----End

1.17.6.3 Creating an Alarm Rule to Monitor an Event

Scenarios

This section describes how to create an alarm rule to monitor an event.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the **Create Alarm Rule** page, configure the parameters.

Table 1-92 Parameter description

Parameter	Description
Name	Specifies the alarm rule name. The system generates a random name, which you can modify.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create one.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Event Type	Specifies the event type of the metric corresponding to the alarm rule.
Event Source	Specifies the service the event is generated for. Select **Relational Database Service** or **Database Proxy Service**.
Monitoring Scope	Specifies the monitoring scope for event monitoring.
Method	Specifies the means you use to create the alarm rule.
Alarm Policy	**Event Name** indicates the instantaneous operations users performed on system resources, such as login and logout. For events supported by event monitoring, see **Events Supported by Event Monitoring**. You can select a trigger mode and alarm severity as needed.

Click  to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 1-93 Alarm notification

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.

Parameter	Description
Notification Object	Specifies the object that receives alarm notifications. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification.

Step 6 Click **Create**.

----End

1.17.6.4 Events Supported by Event Monitoring

Table 1-94 Resource exception events

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
RDS	DB instance creation failure	createInstanceFailed	Major	A DB instance fails to create because the number of disks is insufficient, the quota is insufficient, or underlying resources are exhausted.	Check the number of disks and quota size. Release resources and create DB instances again.	DB instances cannot be created.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Cross-region backup synchronization failure	crossRegionBackupSyncFailed	Minor	Generally, this problem is caused by insufficient underlying network and replication resources.	If this event is continuously reported, submit a service ticket to adjust the underlying resource allocation.	Backups cannot be used for restoration in the destination region.
	Full backup failure	fullBackupFailed	Major	A single full backup failure does not affect the files that have been successfully backed up, but prolongs the incremental backup restoration time during the point-in-time recovery (PITR).	Create a manual backup again.	Backup failed.
	Primary/standby switchover failure	activeStandBySwitchFailed	Major	The standby DB instance does not take over workloads from the primary DB instance due to network or server failures. The original primary DB instance continues to provide workloads within a short time.	Check whether the connection between your application and the database is re-established.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Replication status abnormal	abnormalReplicationStatus	Major	The possible causes are as follows: The replication delay between the primary and standby instances is too long, which usually occurs when a large amount of data is written to databases or a large transaction is processed. During peak hours, data may be blocked. The network between the primary and standby instances is disconnected.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write.
	Replication status recovered	replicationStatusRecovered	Major	The replication delay between the primary and standby instances is within the normal range, or the network connection between them has restored.	No action is required.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	DB instance faulty	faultyDBInstance	Major	A single or primary DB instance was faulty due to a disaster or a server failure.	Check whether an automated backup policy has been configured for the DB instance and submit a service ticket.	The database service may be unavailable.
	DB instance recovered	DBInstanceRecovered	Major	RDS rebuilds the standby DB instance with its high availability. After the instance is rebuilt, this event will be reported.	No action is required.	None
	Failure of changing single DB instance to primary/ standby	singleToHaFailed	Major	A fault occurs when RDS is creating the standby DB instance or configuring replication between the primary and standby DB instances. The fault may occur because resources are insufficient in the data center where the standby DB instance is located.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write of the DB instance.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Database process restarted	DatabaseProcessRestarted	Major	The database process is stopped due to insufficient memory or high load.	Log in to the Cloud Eye console. Check whether the memory usage increases sharply, the CPU usage is too high for a long time, or the storage space is insufficient. You can increase the CPU and memory specifications or optimize the service logic.	Downtime occurs. In this case, RDS automatically restarts the database process and attempts to recover the workloads.
	Instance storage full	instanceDiskFull	Major	Generally, the cause is that the data space usage is too high.	Scale up the instance.	The DB instance becomes read-only because the storage space is full, and data cannot be written to the database.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Instance storage full recovered	instanceDiskFullRecovered	Major	The instance disk is recovered.	No action is required.	The instance is restored and supports both read and write operations.
	MySQL instance connection limit reached	mysqlConnectionsFull	Major	The maximum number of connections supported has been reached as the workload was increasing.	• Release unnecessary connections. • Reduce the load by controlling concurrency. • Upgrade the instance class to allow more connections.	New connections cannot be established.
	MySQL instance connections full recovered	mysqlConnectionsFullRecovered	Major	The number of instance connections has been reduced to below the maximum number of connections.	Check whether the workload on your instance is running properly.	The number of instance connections is below the maximum allowed.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	New connection errors caused by a MySQL overload	highLoadInstanceConnectionsAbnormal	Major	New connections cannot be set up or are abnormal because resources such as CPUs, the memory, storage, or network bandwidth are insufficient.	• Scale up system resources like CPUs, the memory, and storage. • Adjust MySQL parameters, for example, increasing the connection pool size and adjusting the cache size. • Select the abnormal session you want to end and kill it for the databases to recover.	New connections cannot be set up or are abnormal.
	New connection failure caused by MySQL overload resolved	highLoadInstanceConnectionsAbnormalRevocered	Major	The new connection failure caused by MySQL overload has been resolved.	Check whether the workload on your instance is running properly.	The new connection failure has been resolved.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Kafka connection failed	kafkaConnectionFailed	Major	The network is unstable or the Kafka server does not work properly.	Check your network connection and the Kafka server status.	Audit logs cannot be sent to the Kafka server.
Database proxy	Proxy instance access to DB instance failure	proxy_connection_failure_cause_security_group	Major	No rules in the security group of the DB instance allow the proxy instance to access the DB instance.	Add the proxy instance address to the rules of the security group.	Service requests routed through the proxy instance are interrupted.
	Connection failure between proxy instance and DB instance	proxy_connection_failure_to_db	Major	The proxy instance failed to establish a new connection with the primary DB instance, and it may fail to establish a new connection with a read replica. The DB instance or proxy instance is overloaded, or the network between the them is abnormal.	Change values of related parameters based on metrics (connections, active connections, and CPU usage) of the DB instance and proxy instance. If the metrics are normal, submit a service ticket.	Service requests routed through the proxy instance are interrupted.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Connection failure between database proxy and read replica	proxy_connection_failure_to_replica	Minor	The proxy instance failed to establish a new connection with a read replica. The read replica is overloaded, or the network between the proxy instance and read replica is abnormal.	Change values of related parameters based on metrics (connections, active connections, and CPU usage) of the read replica. If the metrics are normal, submit a service ticket.	Read requests routed through the proxy instance are partially interrupted.

Table 1-95 Operation events

Event Source	Event Name	Event ID	Event Severity	Description
RDS	Reset administrator password	resetPassword	Major	The password of the database administrator is reset.
	Operate DB instance	instanceAction	Major	The storage space is scaled or the instance class is changed.
	Delete DB instance	deleteInstance	Minor	The DB instance is deleted.
	Modify backup policy	setBackupPolicy	Minor	The backup policy is modified.

Event Source	Event Name	Event ID	Event Severity	Description
	Modify parameter group	updateParameterGroup	Minor	The parameter group is modified.
	Delete parameter group	deleteParameterGroup	Minor	The parameter group is deleted.
	Reset parameter group	resetParameterGroup	Minor	The parameter group is reset.
	Change database port	changeInstancePort	Major	The database port is changed.
	Primary/standby switchover or failover	PrimaryStandbySwitched	Major	Only automatic failovers are monitored. Manual primary/standby switchovers are not supported.

1.18 Billing Management

1.18.1 Renewing DB Instances

Scenarios

You can renew one or multiple yearly/monthly DB instances at a time.

NOTE

Pay-per-use DB instances cannot be renewed.

The statuses of yearly/monthly DB instances to be renewed must be **Available** or **Abnormal**.

Renewing One DB Instance

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the DB instance and click **Renew** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Overview** page. In the **Billing** area, click **Renew** under **Billing Mode**.

Step 5 Renew the DB instance.

----End

Renewing DB Instances in Batches

Step 1 Log in to the management console.

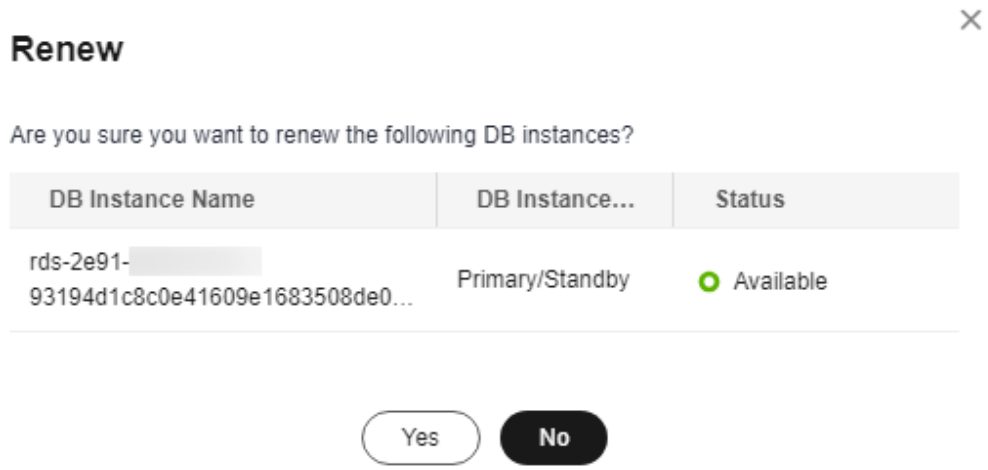
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the DB instances and click **Renew** above the DB instance list.

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

Figure 1-249 Renewing DB instances



----End

1.18.2 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly

Scenarios

If you use RDS for MySQL for a long time, you can change the billing mode of one or multiple DB instances from pay-per-use to yearly/monthly at a time to save money.

NOTE

- Pay-per-use DB instances cannot be changed to yearly/monthly if their status is: frozen, creation failed, changing instance class, or scaling up.
- Currently, RDS DB instances in a Dedicated Computing Cluster (DCC) only supports pay-per-use billing.
- Changing the billing mode from pay-per-use to yearly/monthly does not affect services.

Changing the Billing Mode of One DB Instance

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and choose **More** > **Change to Yearly/Monthly** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Overview** page. In the **Billing** area, click **Configure** under **Billing Mode**.

Step 5 Select the renewal duration, in months. The minimum duration is one month.

- If you do not need to modify your settings, click **Pay** to go to the payment page.
- If you are not sure about the settings, the system will reserve your order. You can choose **Billing & Costs** > **Unpaid Orders** in the upper right corner and pay or cancel the order. The instance status is **Changing to Yearly/Monthly. Payment incomplete. Pay Now.**

Step 6 Select a payment method and click **Confirm**.

Step 7 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the instance list, click to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Yearly/Monthly**.

----End

Changing the Billing Mode of DB Instances in Batches

NOTE

Only pay-per-use DB instances can be changed to yearly/monthly DB instances.

The statuses of pay-per-use DB instances must be **Available** or **Abnormal**.

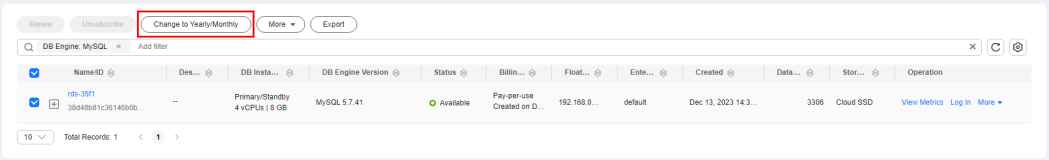
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select multiple DB instances and click **Change to Yearly/Monthly** above the DB instance list.

Figure 1-250 Changing pay-per-use DB instances to yearly/monthly in batches



Step 5 Select the renewal duration in the unit of month. The minimum duration is one month.

- If you do not need to modify your settings, click **Pay** to go to the payment page.
- If you are not sure about the settings, the system will reserve your order. You can choose **Billing & Costs** > **Unpaid Orders** in the upper right corner and pay or cancel the order. The instance status is **Changing to Yearly/Monthly. Payment incomplete. Pay Now.**

Step 6 Select a payment method and click **Confirm**.

Step 7 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the instance list, click to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Yearly/Monthly**.

----End

1.18.3 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Scenarios

You can change the billing mode of a DB instance from yearly/monthly to pay-per-use.

NOTICE

The pay-per-use billing mode is not applied until a yearly/monthly subscription expires, and only if auto-renew is not in effect.

Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the yearly/monthly DB instance and choose **More** > **Change to Pay-per-use** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Overview** page. In the **Billing** area, click **Change to Pay-per-use** under **Billing Mode**.

Step 5 On the displayed page, change the billing mode of the DB instance.

Step 6 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the DB instance list, click to refresh the list. After the DB instance billing mode is changed to pay-per-use, the instance status will change to **Available** and the billing mode will change to **Pay-per-use**.

----End

1.18.4 Unsubscribing from a Yearly/Monthly Instance

Scenarios

To delete a DB instance billed on the yearly/monthly basis, you need to unsubscribe the order. You can unsubscribe a single instance order by referring to **Unsubscribing a Single DB Instance (Method 1)** and **Unsubscribing a Single DB Instance (Method 2)** or unsubscribe multiple instance orders at a time by referring to **Unsubscribing DB Instances in Batches**. For unsubscription fees, see **Unsubscription Rules**.

If you unsubscribe from a DB instance, its read replicas (if any) will also be unsubscribed.

To release DB instances or read replicas billed on a pay-per-use basis, you need to locate the target DB instances or read replicas and click **Delete** on the **Instances** page. For details, see **Deleting Pay-per-Use DB Instances or Read Replicas**.

Constraints

- A DB instance cannot be unsubscribed when any operations are being performed on it. It can be unsubscribed only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be unsubscribed.

Unsubscribing a Single DB Instance (Method 1)

Unsubscribe a yearly/monthly DB instance or read replica on the **Instances** page.

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance or read replica and choose **More** > **Unsubscribe** in the **Operation** column.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription results. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing a Single DB Instance (Method 2)

Unsubscribe a yearly/monthly DB instance or read replica on the **Billing Center** page.

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

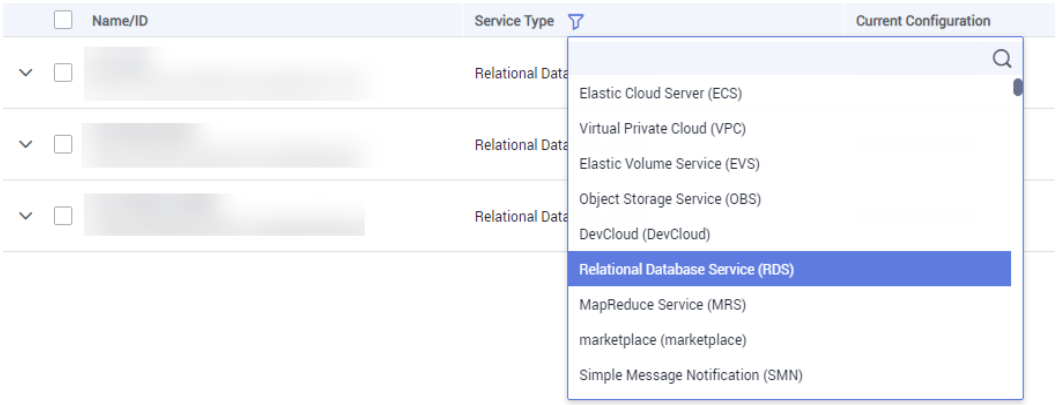
Step 4 In the upper right corner, click **Billing & Costs**.

Step 5 In the navigation pane, choose **Orders** > **Unsubscriptions**.

Step 6 On the displayed page, select the order to be unsubscribed and click **Unsubscribe** in the **Operation** column.

- You can select **Relational Database Service (RDS)** in the **Service Type** column to filter all RDS orders.

  Figure 1-251 Filtering all RDS orders



- Alternatively, search for target orders by name, order No., or ID in the search box.
- A maximum of 20 resources can be unsubscribed at a time.

Step 7 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 8 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 9 View the unsubscription result. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing DB Instances in Batches

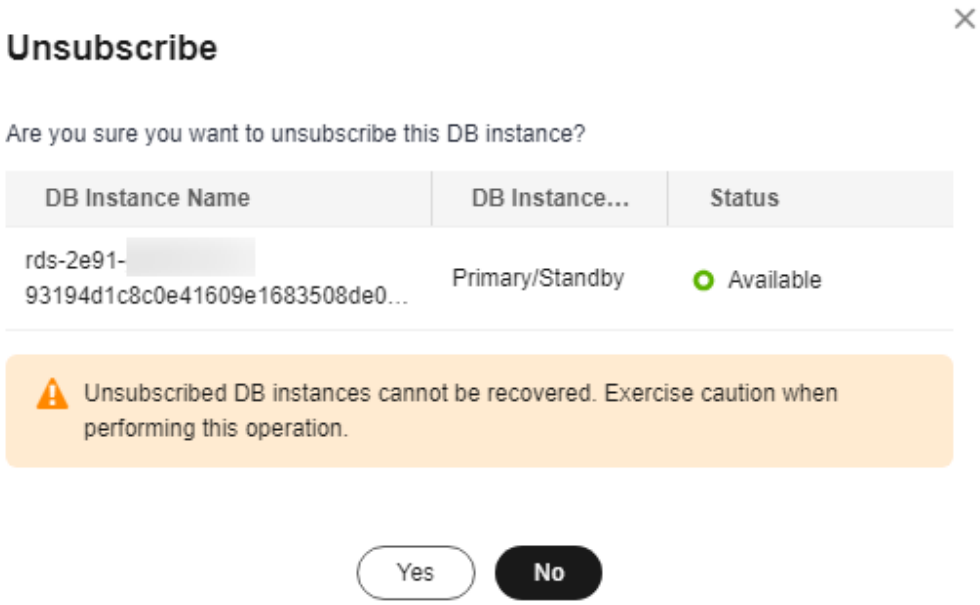
Step 1 **Log in to the management console**.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the target DB instances to be unsubscribed and click **Unsubscribe** above the DB instance list. In the displayed dialog box, click **Yes**.

Figure 1-252 Unsubscribing yearly/monthly orders in batches



Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription results. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

1.19 Interconnection with CTS

1.19.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 1-96 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	setInstanceTag
Deleting a tag	instance	setInstanceTag
Editing a tag	instance	setInstanceTag
Deleting a DB instance	instance	deleteInstance
Performing a primary/standby switchover	instance	instanceFailOver
Changing the replication mode	instance	instanceFailOver-Mode

Operation	Resource Type	Trace Name
Changing a failover priority	instance	instanceFailOver-Strategy
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 merged binlogs	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
Changing the billing mode of a DB instance from pay-per-use to yearly/monthly or renewing a DB instance	all	bssUpdateMetadata
Creating a database account	instance	createDBUser
Resetting a password	instance	resetDBUserPassword

Operation	Resource Type	Trace Name
Changing account permissions	instance	grantDBUser
Modifying the host IP addresses of an account	instance	UpdateHostPrivilege
Deleting a database account	instance	deleteDBUser
Creating a database	instance	createDatabase
Authorizing a database	instance	grantDBUser
Deleting a database	instance	deleteDatabase

1.19.2 Viewing Tracing Events

For details about how to view audit logs, see **Querying Real-Time Traces**.

1.20 Task Center

1.20.1 Viewing a Task

You can view the progress and results of scheduled and instant tasks on the **Task Center** page.

Supported Tasks

Table 1-97 Supported tasks

Task Type	Category	Task Name
Instant tasks	Instance creation	Creating a MySQL DB instance, creating a MySQL read replica
	Instance lifecycle	Rebooting a MySQL DB instance
	Instance modifications	Scaling a MySQL DB instance, changing the MySQL instance type from single to primary/standby, switching MySQL primary/standby DB instances, applying for a MySQL private domain name, migrating a standby MySQL DB instance, changing a MySQL DB instance class, binding an EIP to a MySQL DB instance, unbinding an EIP from a MySQL DB instance
	Version upgrade	Upgrading a MySQL DB instance engine version

Task Type	Category	Task Name
	Backup and restoration	Restoring to a new MySQL DB instance, restoring to an existing MySQL DB instance, restoring to the current MySQL DB instance, restoring tables to a point in time, restoring databases to a point in time
Scheduled tasks	Instance lifecycle	Starting a MySQL DB instance, rebooting a MySQL DB instance
	Instance modifications	Changing a MySQL DB instance class
	Version upgrade	Upgrading a MySQL DB instance engine version

Viewing an Instant Task

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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, order ID, 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 can only show up to 30 days of past tasks.
- 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.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

FAQ

Q: Why does the task progress percentage remain unchanged?

A: The task progress percentage does not change linearly. If the percentage remains unchanged for a long time, some time-consuming steps are being performed. Please wait.

1.20.2 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.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed **Instant Tasks** page, locate the 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:

- 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the 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 **OK** to cancel the task. Then, click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

You can delete the records of scheduled tasks in any of the following statuses:

- Completed
- Failed
- Canceled
- To be executed
- To be authorized

----End

1.21 RDS for MySQL Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

- Log in to the management console. Click **Service List** and choose **Management & Governance** > **Tag Management Service**. Set predefined tags on the TMS console.

- A tag consists of a key and value. You can add only one value for each key.
- Up to 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 Log in to the management console.

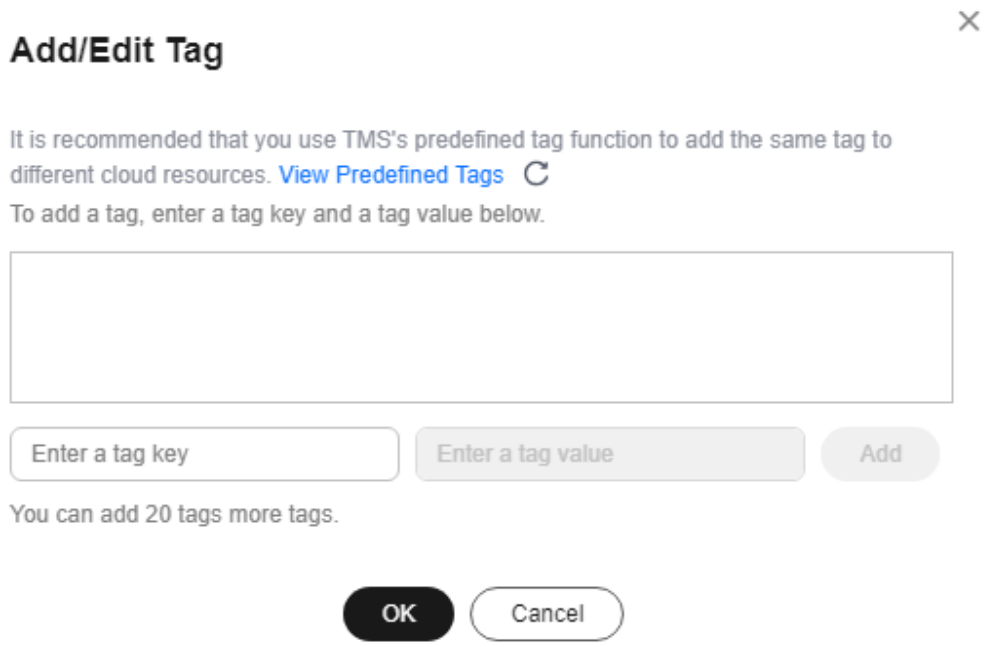
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Tags**. On the displayed page, click **Add/Edit Tag**. In the displayed dialog box, enter a tag key and value, click **Add**, and click **OK**.

Figure 1-253 Adding a tag



- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with your DB instances (except the current instance).
- The tag key must be unique. It must consist of 1 to 128 characters and can include letters, digits, spaces, and the following characters: _ . : = + - @. It cannot start or end with a space, or start with **_sys_**.
- The tag value (optional) can consist of up to 255 characters and can include letters, digits, spaces, and the following characters: _ . : / = + - @.

Step 6 After a tag has been added, view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane on the left, choose **Tags**. Select the tag to be deleted and click **Delete**. In the displayed dialog box, click **OK**.

After a tag has been deleted, it will no longer be displayed on the **Tags** page.

----End

1.22 RDS for MySQL Quotas

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of RDS quotas include the maximum number of DB instances that you can create. Quotas are put in place to prevent excessive resource usage.

If a quota cannot meet your needs, apply for a higher quota.

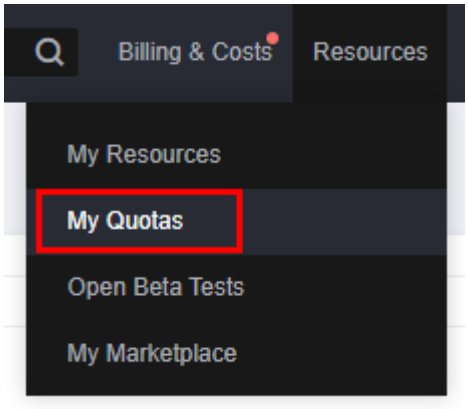
Viewing Quotas

Step 1 Log in to the management console.

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

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Figure 1-254 My quotas



Step 4 On the **Quotas** page, view the used and total quotas of each type of resources.

----End

Increasing Quotas

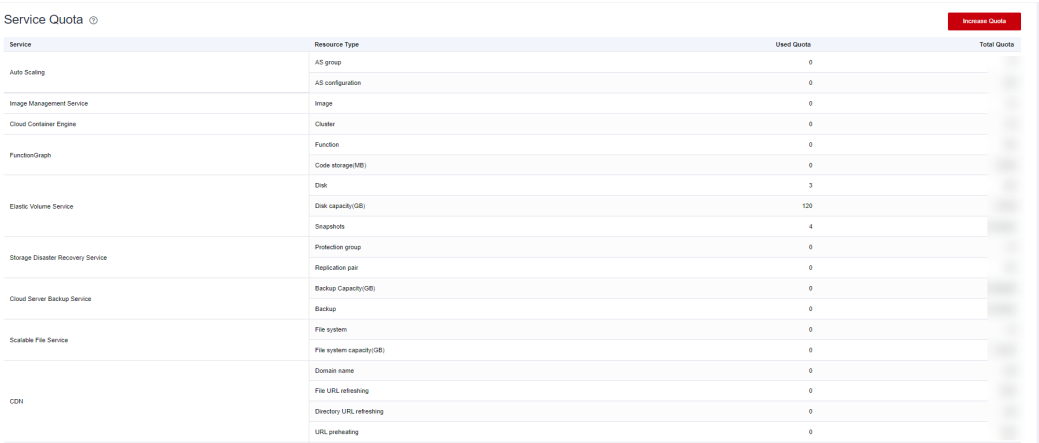
Step 1 **Log in to the management console**.

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

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 1-255 Increasing quotas



Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for adjustment.

Step 6 After all required parameters are configured, select the agreement and click **Submit**.

----End

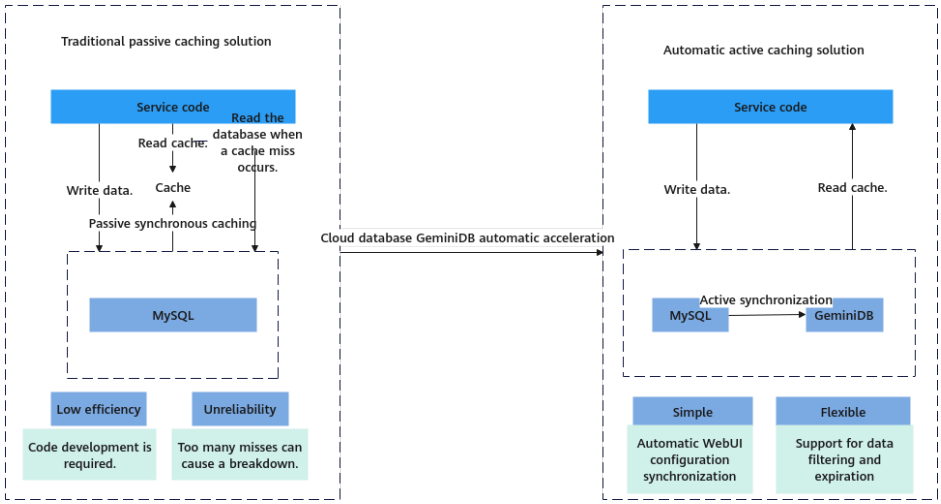
1.23 RDS Memory Acceleration

1.23.1 Memory Acceleration Overview

GeminiDB Redis API offers memory acceleration to enhance the conventional cache solution. With this feature, users can set up rules on the GUI to cache MySQL data automatically, thereby speeding up MySQL access.

The conventional cache solution is inefficient and unreliable as it necessitates users to create code for writing MySQL data to the cache. The active cache solution with cloud data memory acceleration (DB Cache) supports visualized configuration on the GUI, making it easier to set up. Once the configuration is done, data can be synchronized automatically. DB Cache also supports data filtering and expiration time setting, which enhances development efficiency and data reliability.

Figure 1-256 Diagram



1.23.2 Enabling and Using Memory Acceleration

Enable memory acceleration.

Step 1: Create a GeminiDB instance.

Step 2: Create a mapping rule.

Step 3: Use the memory acceleration module.

Precautions

- After memory acceleration is enabled, commands such as RESET MASTER and FLUSH LOGS used to delete binlogs on MySQL instances are not allowed.
- Currently, only hash data from MySQL can be converted to GeminiDB Redis API.
- A Redis key prefix and a delimiter in a new rule can neither include those nor be included in those specified for an existing rule. For example, if the key prefix in a new rule is **pre1:** and is separated by a comma (,) and the key prefix in an existing rule is **pre1** and is separated by a colon (:), the new rule cannot be created.
- Currently, the ENUM, SET, and JSON data cannot be synchronized.
- Currently, only single-table queries are supported during lightweight incremental synchronization. Joint queries are not supported.
- Only GeminiDB Redis instances are charged. There are no other fees for this function.
- If you delete an RDS instance, the GeminiDB Redis instance with DB Cache enabled will not be deleted. If you do not need the GeminiDB Redis instance, delete it in a timely manner to avoid extra fees.
- When you purchase an RDS instance, if you select **Buy Now** for memory acceleration, a GeminiDB instance is automatically purchased to enable DB Cache. You can skip creating a GeminiDB instance and start from **Creating a Mapping Rule**. This function is now in OBT. To use it, choose **Service Tickets > Create Service Ticket** in the upper right corner of the console and contact customer service personnel.

Procedure

Creating a GeminiDB Instance

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane, choose **Memory Acceleration**.

- Click **Create GeminiDB Instance** and perform **Step 6**.
- Click **Use Existing GeminiDB Instance** and select an existing GeminiDB Redis instance.

NOTICE

When you select **Use Existing GeminiDB Instance**, only primary/standby instances are supported. The region, VPC, subnet, and security group of the GeminiDB and RDS instances must be the same.

Step 6 Set parameters listed in **Table 1-98** and click **Submit**.

Table 1-98 Basic information

Parameter	Description
Instance Class	CPU and memory of the instance. For details, see **Table 1-99**.
Database Port	Port number for accessing the instance. You can specify a port number based on your requirements. The port number ranges from 1024 to 65535 except 2180, 2887, 3887, 6377, 6378, 6380, 8018, 8079, 8091, 8479, 8484, 8999, 12017, 12333, and 50069. If you do not specify a port number, port 6379 is used by default. **NOTE** You cannot change the database port after an instance is created.
DB Instance Name	The instance name: • Can be the same as an existing instance name. • Can include 4 to 64 bytes and must start with a letter. It is case-sensitive and allows only letters, digits, hyphens (-), and underscores (_).

Parameter	Description
Database Password	Database password set by the user. • Must be 8 to 32 characters long. • Can include two of the following: uppercase letters, lowercase letters, digits, and special characters: ~!@#%^*-_=+? • For security reasons, set a strong password. The system will verify the password strength. Keep your password secure. The system cannot retrieve it if it is lost.
Confirm Password	Enter the administrator password again.

NOTE

By default, the region, AZ, VPC, and subnet of the GeminiDB and RDS instances are the same.

Table 1-99 GeminiDB Redis instance specifications

Storage (GB)	Nodes	vCPUs	QPS	Maximum Connections per Single-node Instance	Databases
16	2	1	10,000	10,000	1,000
24	2	2	20,000	10,000	1,000
32	2	2	20,000	10,000	1,000
48	2	4	40,000	20,000	1,000
64	2	4	40,000	20,000	1,000
96	2	8	80,000	20,000	1,000
128	2	16	160,000	20,000	1,000

----End

Creating a Mapping Rule

Step 1 **Log in to the Huawei Cloud console**.

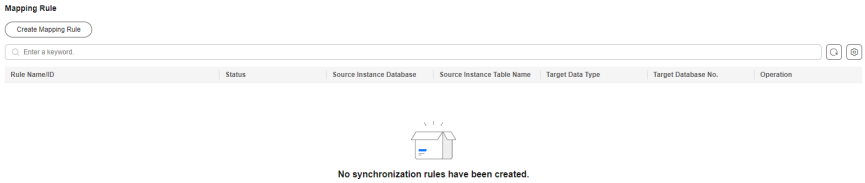
Step 2 Click in the upper left corner and select a region and project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane, choose **Memory Acceleration**. In the **Mapping Rule** area, click **Create Mapping Rule**.

Figure 1-257 Mapping rule

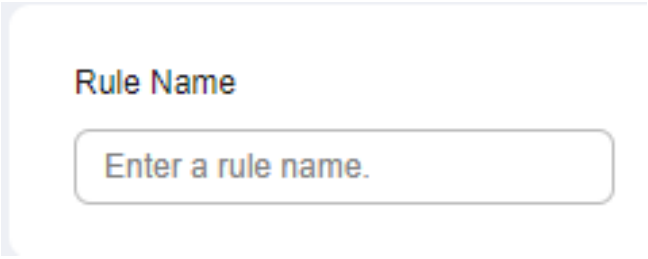


Step 6 On the displayed page, configure parameters.

1. Enter a rule name.

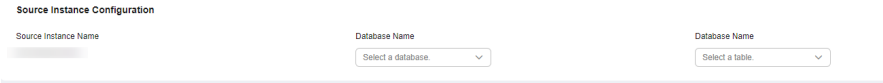
   Rule Name: Enter a mapping rule name. The rule name must be unique within a GeminiDB instance and cannot exceed 256 characters or include number signs (#).

   Figure 1-258 Rule name

2. Configure source instance information.
   - **Database Name**: Select a database of the acceleration instance.
   - **Table Name**: Select a table of the acceleration instance.

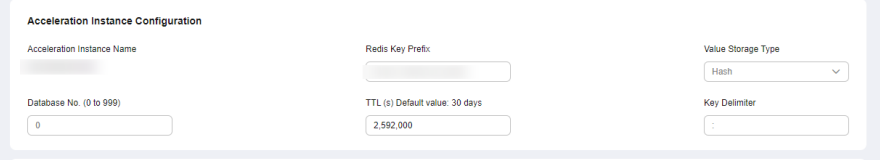
   Figure 1-259 Configuring source instance information

3. Configure acceleration instance information.
   - **Redis Key Prefix**: This parameter is optional. The default value is in the format of *Database name*:*Table name*:*Field name 1*:*Field name 2...* and can contain a maximum of 1,024 characters. If you have created a custom prefix, it will be used instead of the default one.
   - **Value Storage Type**: Data type of the cache. Currently, only hash data is supported.
   - **Database No. (0-999)**: ID of a database that stores cached data in the acceleration instance. The default value is **0**.
   - **TTL (s) Default value: 30 days**: Validity period of cached data in the acceleration instance. The default value is 30 days (2,592,000 seconds). If you enter **-1**, the cached data will never expire.

- **Key Delimiter**: Separator among the Redis key prefix, key, and key fields. It is a single character in length.

Figure 1-260 Configuring acceleration instance information

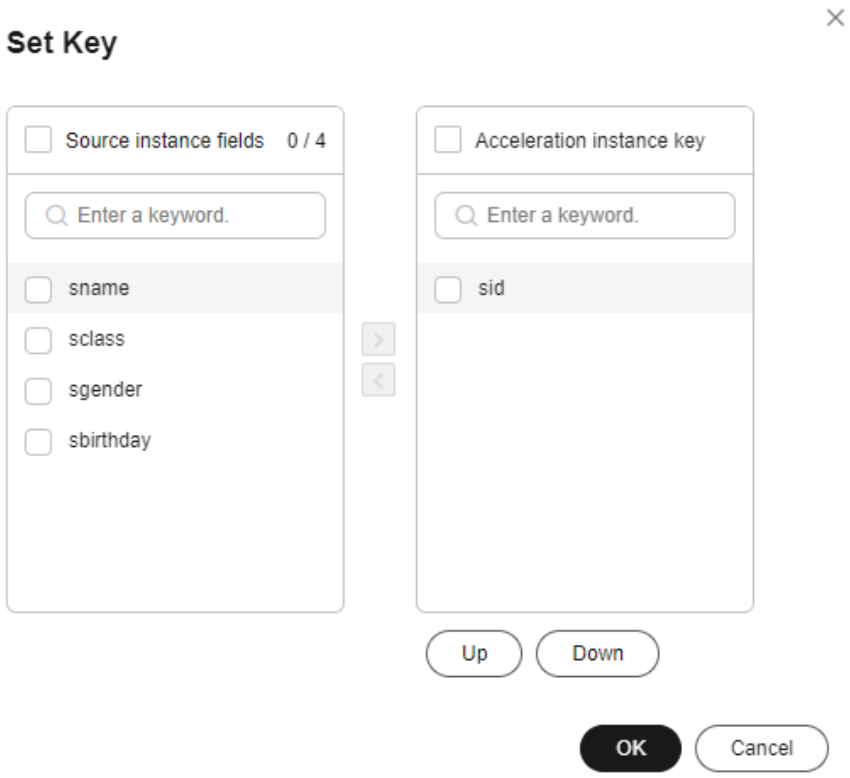


4. Click **Set Key**, select a key field of the acceleration instance, and click **OK**.

   NOTE

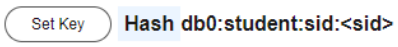
   If an acceleration instance key consists of multiple source instance fields, the key must be unique in a MySQL instance. You can click **Up** or **Down** to adjust the sequence of each field.

Figure 1-261 Key settings



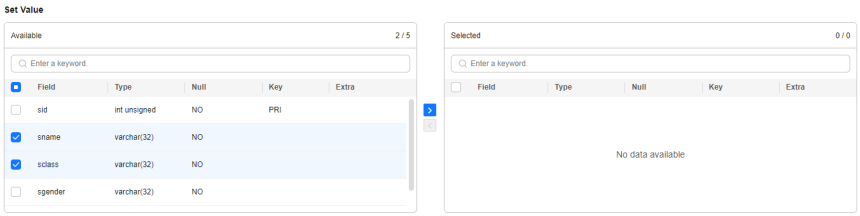
After the parameters are set, the key is displayed.

Figure 1-262 Key



5. Configure the acceleration instance fields.
   Move the required fields in the source instance to the acceleration instance.

Figure 1-263 Configuring acceleration instance fields



6. After setting the parameters, click **Submit**.

----End

Using the Memory Acceleration Module

1. Create database **db1** in the source MySQL instance and create table **students** in **db1**.

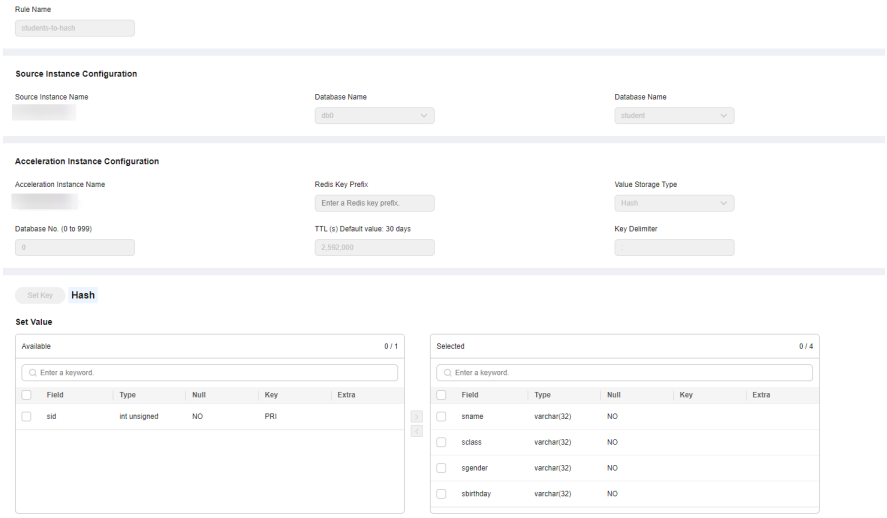
```
mysql> CREATE DATABASE db1;
Query OK, 1 row affected (0.00 sec)

mysql> CREATE TABLE db1.students(
       sid INT UNSIGNED PRIMARY KEY AUTO_INCREMENT NOT NULL,
       sname VARCHAR(20),
       sclass INT,
       sgender VARCHAR(10),
       sbirthday DATE
       );
Query OK, 0 rows affected (0.00 sec)

mysql> DESC db1.students;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| sid       | int unsigned | NO   | PRI | NULL    | auto_increment |
| sname     | varchar(20)  | YES  |     | NULL    |                |
| sclass    | int          | YES  |     | NULL    |                |
| sgender   | varchar(10)  | YES  |     | NULL    |                |
| sbirthday | date         | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

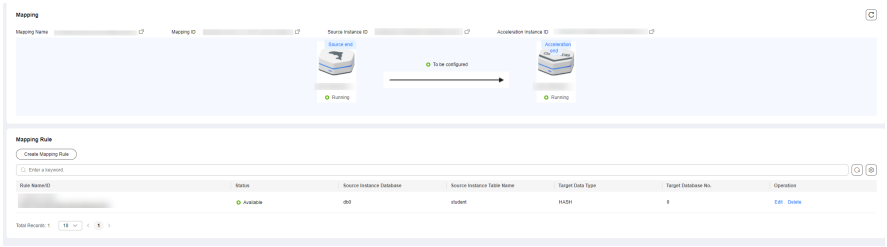
2. After the table is created, on the memory acceleration page, create a mapping rule to convert each row in the **students** table into a Redis hash. The key of a hash is in the format of *Database name*:*Data table name*:**sid:**<*sid value*>. The selected fields are **sname**, **sclass**, **sgender**, and **sbirthday**.

Figure 1-264 Configuring a mapping rule



3. After a mapping rule is created, check the mapping rule and information.

Figure 1-265 Mapping information



4. Insert a new data record to the **students** table in the MySQL instance.

```
mysql> INSERT INTO db1.students (sname, sclass, sgender, sbirthday) VALUES ('zhangsan', 1, 'male', '2015-05-20');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM db1.students;
+-----+----------+--------+---------+------------+
| sid | sname    | sclass | sgender | sbirthday  |
+-----+----------+--------+---------+------------+
|   1 | zhangsan |      1 | male    | 2015-05-20 |
+-----+----------+--------+---------+------------+
1 row in set (0.00 sec)
```

5. After the mapping rule is created, the data is automatically synchronized to the GeminiDB instance. Run commands in the GeminiDB instance to query the data.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:1"

127.0.0.1:6379> HGETALL db1:students:sid:1
1) "sbirthday"
2) "2015-05-20"
3) "sclass"
4) "1"
5) "sgender"
6) "male"
7) "sname"
8) "zhangsan"
```

6. Insert a new data record to the **students** table in the MySQL instance.


```
mysql> INSERT INTO db1.students (sname, sclass, sgender, sbirthday) VALUES ('lisi', 10, 'male',
'2015-05-22');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM db1.students;
+-----+----------+--------+---------+------------+
| sid | sname    | sclass | sgender | sbirthday  |
+-----+----------+--------+---------+------------+
|   1 | zhangsan |      1 | male    | 2015-05-20 |
|   2 | lisi     |     10 | male    | 2015-05-22 |
+-----+----------+--------+---------+------------+
2 rows in set (0.00 sec)
```

7. Check whether the new data is synchronized to the GeminiDB instance.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:1"
2) "db1:students:sid:2"

127.0.0.1:6379> HGETALL db1:students:sid:2
1) "sbirthday"
2) "2015-05-22"
3) "sclass"
4) "10"
5) "sgender"
6) "male"
7) "sname"
```

8. Update data in the **students** table in the MySQL instance.

```
mysql> UPDATE db1.students SET sclass=12, sname='wangwu' WHERE sid = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM db1.students;
+-----+--------+--------+---------+------------+
| sid | sname  | sclass | sgender | sbirthday  |
+-----+--------+--------+---------+------------+
|   1 | wangwu |     12 | male    | 2015-05-20 |
|   2 | lisi   |     10 | male    | 2015-05-22 |
+-----+--------+--------+---------+------------+
2 rows in set (0.00 sec)
```

9. Check whether the data is updated in the GeminiDB instance.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:1"
2) "db1:students:sid:2"

127.0.0.1:6379> HGETALL db1:students:sid:1
1) "sbirthday"
2) "2015-05-20"
3) "sclass"
4) "12"
5) "sgender"
6) "male"
7) "sname"
8) "wangwu"
```

10. Delete data from the **students** table in the MySQL instance.

```
mysql> DELETE FROM db1.students WHERE sid = 1;
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM db1.students;
+-----+-------+--------+---------+------------+
| sid | sname | sclass | sgender | sbirthday  |
+-----+-------+--------+---------+------------+
|   2 | lisi  |     10 | male    | 2015-05-22 |
+-----+-------+--------+---------+------------+
1 row in set (0.00 sec)
```

11. Check whether the data is deleted from the GeminiDB instance.

```
127.0.0.1:6379> KEYS *
1) "db1:students:sid:2"
```


1.23.3 Modifying and Deleting a Memory Acceleration Rule

A memory acceleration rule can enable automated data synchronization from MySQL to GeminiDB. You can also modify and delete this rule.

Precautions

- Currently, only hashes from MySQL can be converted to GeminiDB Redis API.
- If a table name of the MySQL instance in the memory acceleration rule is changed, you need to reconfigure the rule.
- Currently, the ENUM, SET, and JSON data cannot be synchronized.
- If you rename or delete one or more key fields of a memory acceleration rule, the rule becomes invalid.

Modifying a Mapping Rule

Step 1 **Log in to the Huawei Cloud console**.

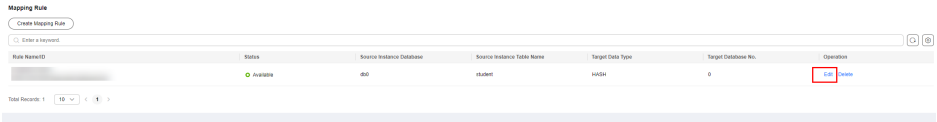
Step 2 Click  in the upper left corner and select a region and project.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

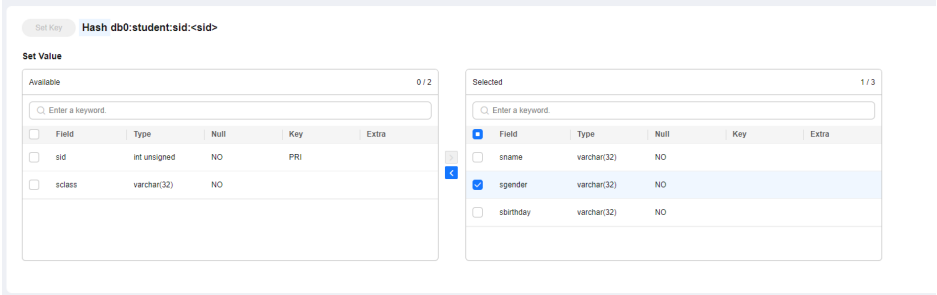
Step 5 In the navigation pane on the left, choose **Memory Acceleration**. In the **Mapping Rule** area, locate the target rule and click **Edit** in the **Operation** column.

Figure 1-266 The **Edit** button



Step 6 After editing the fields, click **Submit**.

Figure 1-267 Editing a mapping rule



----End

Deleting a Mapping Rule

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane on the left, choose **Memory Acceleration**. In the **Mapping Rule** area, locate the target rule and click **Delete** in the **Operation** column.

----End

1.23.4 Viewing and Removing Mappings

You can view the mapping list on the **Memory Acceleration Management** page and remove mappings.

Usage Notes

- After a mapping is removed, service applications cannot obtain the latest data of the source database from the acceleration instance.
- The corresponding mapping rule will be cleared after a mapping is removed.
- If the source instance or acceleration instance is not normal, the mapping cannot be removed.

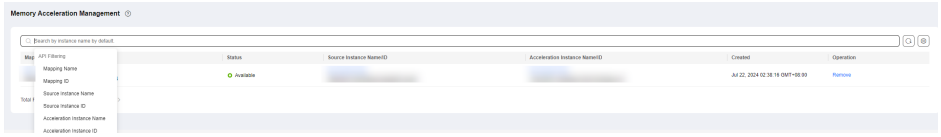
Querying the Mapping List

Step 1 Log in to the Huawei Cloud console.

Step 2 In the service list, choose **Databases** > **GeminiDB Redis API**.

Step 3 In the navigation pane on the left, choose **Memory Acceleration Management**. On the displayed page, search for your target mapping by keyword (such as the mapping name or mapping ID).

Figure 1-268 Mapping list



----End

Removing a Mapping

Step 1 Log in to the Huawei Cloud console.

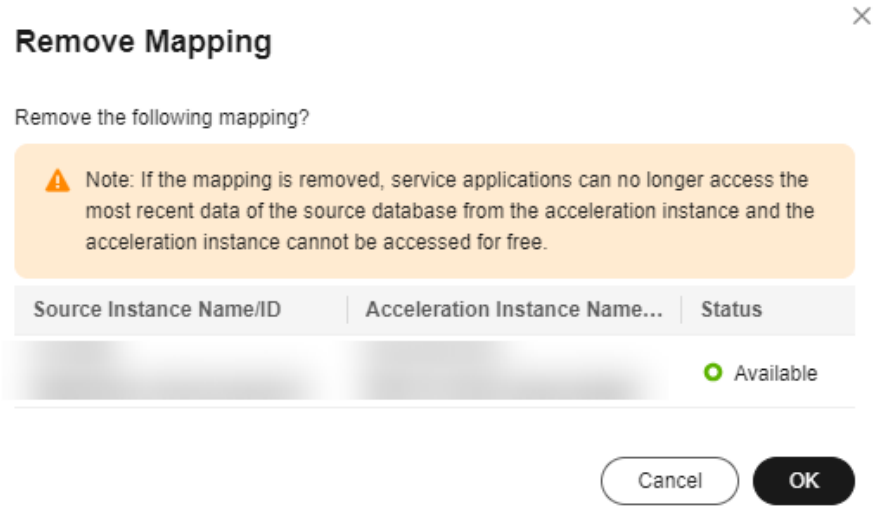
Step 2 In the service list, choose **Databases** > **GeminiDB Redis API**.

Step 3 In the navigation pane on the left, choose **Memory Acceleration Management**. On the displayed page, locate the target mapping and click **Remove** in the **Operation**.

Figure 1-269 Memory acceleration management

Step 4 In the displayed dialog box, click **OK**.

Figure 1-270 Removing a mapping



----End

2 Working with RDS for MariaDB

2.1 Suggestions on Using RDS for MariaDB

2.1.1 Instance Usage Suggestions

DB Instances

DB Instance Types

- Primary/Standby
  - A primary/standby pair provides an HA architecture. It is suitable for production databases of large and medium enterprises in the Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other sectors.
  - When a primary instance is being created, a standby instance is provisioned along with it to provide data redundancy. The standby instance is invisible to you after being created.
  - If a failover occurs due to a primary instance failure, your database client will be disconnected for a short period of time. The client needs to be able to reconnect to the instance.
- Single
  - A single-node architecture is more cost-effective than primary/standby pairs.
  - It is only recommended for development and testing of microsites, and small and medium enterprises, or for learning about RDS.
  - If a fault occurs on a single instance, the instance cannot recover in a timely manner.
- Read replica

  RDS for MariaDB supports single read replicas.

  NOTE

  If the replication between a read replica and the DB instance is abnormal, it can take a long time to rebuild and restore the read replica (depending on the data volume).

Instance Classes

- Dedicated

  The instance has dedicated CPU and memory resources to ensure stable performance. The performance of a dedicated instance is never affected by other instances on the same physical machine. This instance class is good when performance stability is important.

- General-purpose

  CPU resources are shared with other general-purpose DB instances on the same physical machine. CPU usage is maximized through resource overcommitment. This instance class is a cost-effective option and suitable for scenarios where performance stability is not critical.

Database Connection

- Configure RDS for MariaDB parameters for your workloads.
- Keep an appropriate number of active connections.
- Periodically release persistent connections because maintaining them may generate a large cache and use up memory.

Reliability and Availability

- Select primary/standby DB instances for production databases.
- Deploy primary and standby instances in different AZs.
- Create read replicas and enable read/write splitting for workloads involving frequent read/write operations.
- Change instance classes during off-peak hours.
- Select an instance class and storage space appropriate to your workloads.
- After scaling up your primary DB instance, scale up its read replicas in a timely manner to prevent service exceptions caused by insufficient storage of read replicas.

Backup and Restoration

- Perform manual backups during off-peak hours and change the backup time window (default setting: 01:00-02:00 (GMT+08:00)) for automated backup as required.
- Set the backup cycle to **All** for DB instances that process many write requests every day.
- Configure a backup retention period suited to your service demands. The default value is 7 days.
- If a DB instance is deleted, its automated full backups and binlog backups are also deleted. Perform manual backup for all data before deleting a DB instance.
- Configure a custom recycling policy to ensure that any instances that are deleted by mistake can be rebuilt.

Routine O&M

- Periodically check slow query logs and error logs to identify problems in advance.

- Periodically check the resource usage of DB instances. If the resources are insufficient, scale up the resources in a timely manner.
- Monitor instance metrics. If any metric is beyond its expected range, address related issues as soon as possible.
- Run the **SELECT** statement before deleting or modifying a record.

Security

- Prevent your database from being accessed from the Internet. If you want to allow the access from the Internet, bind an EIP to your DB instance and configure a whitelist.
- Use SSL to connect to your DB instance.

2.1.2 Database Usage Suggestions

Database Naming

- The names of database objects like databases, tables, and columns should be in lowercase. Different words in the name are separated with underscores (_).
- Reserved words and keywords cannot be used to name database objects in RDS for MariaDB.
- Each database object name must be explainable and contain a maximum of 32 characters.
- Each temporary table in databases is prefixed with **tmp** and suffixed with a date.
- Each backup table in databases is prefixed with **bak** and suffixed with a date.
- All columns storing the same data in different databases or tables must have the same name and be of the same type.

Database Design

- All tables use the InnoDB storage engine unless otherwise specified. InnoDB supports transactions and row locks. It delivers excellent performance, making it easy to restore data.
- Databases and tables all use the UTF8 character set to avoid characters getting garbled by character set conversion.
- All tables and fields require comments that can be added using the COMMENT clause to maintain the data dictionary from the beginning of the design.
- To avoid cross-partition queries, RDS for MariaDB partitioned tables are not recommended. Cross-partition queries will decrease the query efficiency. A partitioned table is logically a single table, but the data is actually stored in multiple different files. If you use partitioned tables for storage, store files from different partitions on different disk arrays.
- Do not create too many columns in one table. Store cold and warm data separately to reduce the width of a table. In doing so, more rows of data can be stored in each memory page, decreasing disk I/O and making more efficient use of the cache.
- Columns that are frequently used together should be in the same table to avoid JOIN operations.

- Do not create reserved fields in a table. Otherwise, modifying the column type will lock the table, which has a greater impact than adding a field.
- Do not store binary data such as images and files in databases.

Field Design

- Select a small data type for each column as much as possible. Numeric data is preferred, followed by dates or binary data, and the least preferred is characters. The larger the column data type, the more the space required for creating indexes. As a result, there are fewer indexes on a page and more I/O operations required, so database performance deteriorates.
- If the integer type is used as the database field type, select the shortest column type. If the value is a non-negative number, it must be the unsigned type.
- Ensure that each column has the NOT NULL attribute.
- Do not use the ENUM type. Instead, use the TINYINT type. Change ENUM values using ALTER. The ORDER BY operations on ENUM values are inefficient and require extra operations.

  If you have specified that ENUM values cannot be numeric, other data types (such as CHAR) can be used.
- If the numeric data type is required, use DECIMAL instead of FLOAT or DOUBLE. FLOAT and DOUBLE data cannot be stored precisely, and value comparison results may be incorrect.
- When you want to record a date or specific time, use the DATETIME or TIMESTAMP type instead of the string type.
- Store IP addresses using the INT UNSIGNED type. You can convert IP addresses into numeric data using function inet_aton or inet_ntoa.
- The VARCHAR data should be as short as possible. Although the VARCHAR data varies in length dynamically on disks, it occupies the maximum length in memory.
- Use VARBINARY to store variable-length character strings that are case-sensitive. VARBINARY is case-sensitive by default and quick to process because no character sets are involved.

Index Design

- Use no more than 5 indexes in a single table. Indexes speed up queries, but too many indexes may slow down writes. Inappropriate indexes sometimes reduce query efficiency.
- Do not create an independent index for each column in a table. A well-designed composite index is much more efficient than a separate index on each column.
- Create a primary key for each InnoDB table. Neither use a frequently-updated column as the primary key nor a multi-column primary key. Do not use the UUID, MD5, or character string column as the primary key. Use a column whose value can increment continuously as the primary key. So, the auto-increment ID column is recommended.
- Create an index on the following columns:
  - Columns specified in the WHERE clause of SELECT, UPDATE, or DELETE statements

  - Columns specified in ORDER BY, GROUP BY, or DISTINCT
  - Columns associated for joining multiple tables.
- The index column order is as follows:
  - Put the column with the highest selectivity on the far left when creating a composite index. Selectivity = Different values in a column/Total rows in the column
  - Put the column with the smallest field length on the far left of the composite index. The smaller length a field has, the more data one page stores, and the better the I/O performance is.
  - Put the most frequently used column on the left of the composite index, so you can create fewer indexes.
- Avoid using redundant indexes, such as primary key (id), index (id), and unique index (id).
- Avoid using duplicate indexes, such as index(a,b,c), index(a,b), and index(a). Duplicate and redundant indexes may slow down queries because the RDS for MariaDB query optimizer does not know which index it should use.
- When creating an index on the VARCHAR field, specify the index length based on selectivity. Do not index the entire field.

  If an index with the length of 20 bytes is the string type, its selectivity can reach 90% or above. In this case, use **count(distinct left(column name, index length))/count(*)** to check index selectivity.
- Use covering indexes for frequent queries.

  A covering index is a special type of index where all required fields for a query are included in the index. The index itself contains columns specified in WHERE and GROUP BY clauses, but also column combinations queried in SELECT, without having to execute additional queries.
- Constraints on foreign keys are as follows:

  The character sets of the columns for which a foreign key relationship is established must be the same, or the character sets of the parent and child tables for which a foreign key relationship is established must be the same.

SQL Statement Development

- Use prepared statements to perform database operations in programs. Prepared statements can be executed multiple times in a program once they are written. They are more efficient than SQL statements.
- Avoid implicit conversions because they may cause indexes to become invalid.

  Do not perform function conversions or math calculations on columns in the WHERE clause. Otherwise, the index becomes invalid.
- Do not use double percent signs (%%) or place % before a query condition, or the index cannot be used.
- Do not use **SELECT *** for queries because using **SELECT ***:
  - Consumes more CPUs, IP addresses, and bandwidth.
  - Causes covering indexes to become unavailable.
  - Increases the impact of table structure changes on code.

- Do not use subqueries. Subqueries generate temporary tables that do not have any indexes. If there is a lot of data, the query efficiency is severely affected. Convert subqueries into associated queries.
- Minimize the use of JOIN operations for more than 5 tables. Use the same data type for the fields that require JOIN operations.

  Each JOIN operation on a table occupies extra memory (controlled by **join_buffer_size**) and requires temporary table operations, affecting query efficiency.
- Reduce interactions with the same database as much as possible. The database is more suitable for processing batch operations.
- Replace OR clauses with IN clauses because IN clauses can effectively use indexes. Specify no more than 500 values for an IN clause.
- Do not perform reverse queries, for example, NOT IN and NOT LIKE.
- Do not use ORDER BY RAND() for random sorting.

  This operation loads all data that meets the conditions from the table to the memory for sorting, consuming more CPUs, I/O, and memory resources.

  Obtain a random value from the program and retrieve data from the involved database based on the value.
- If deduplication is not required, use UNION ALL instead of UNION.

  UNION ALL does not sort out result sets.
- Combine multiple operations and perform them in batches. The database is good for batch processing.

  This reduces interactions with the same database.
- If there are more than 1 million rows of write operations, perform them in multiple batches.

  A large number of batch writes may result in excessive primary/standby latency.
- If ORDER BY is used, use the order of indexes.
  - The last field of ORDER BY is a part of a composite index and is placed at the end of the composite index order.
  - Avoid file_sort to speed up queries.

  Correct example: in **WHERE a=? AND b=? ORDER BY c;**, index: **a_b_c**

  Wrong example: If an index supports range search, the index order cannot be used. For example, **WHERE a>10 ORDER BY b;**, index: **a_b** (sorting is not allowed)

2.2 Instance Connection

2.2.1 Connecting to an RDS for MariaDB Instance

You can connect to an RDS for MariaDB instance through a command-line interface (CLI), Data Admin Service (DAS), or using Java database connectivity (JDBC).

Table 2-1 Connection methods

Connection Method	Description
Connecting to an RDS for MariaDB Instance Through the MySQL CLI Client	In Linux, you need to install a MariaDB client on your device and connect to the instance through the MySQL CLI over a private or public network. • A floating IP address is provided by default. When your applications are deployed on an ECS that is in the same region and VPC as the RDS for MariaDB instance, you are advised to use a floating IP address to connect to the instance through the ECS. • If you cannot access your RDS for MariaDB instance through a floating IP address, bind an EIP to the instance and connect to the instance through the ECS.
Connecting to an RDS for MariaDB Instance Through JDBC	If you are connecting to an instance through JDBC, the SSL certificate is optional. For security reasons, you are advised to download the SSL certificate to encrypt the connection. SSL is disabled by default for RDS for MariaDB instances. You can enable SSL by referring to Configuring an SSL Connection. SSL encrypts connections to databases but it increases the connection response time and CPU usage. Therefore, you are advised not to enable SSL.
Connecting to an RDS for MariaDB Instance Through DAS	DAS enables you to manage databases on a web-based console and provides you with database development, O&M, and intelligent diagnosis to make it easy to use and maintain your databases. The permissions required for connecting to DB instances through DAS are enabled by default.

2.2.2 Connecting to an RDS for MariaDB Instance Through the MySQL CLI Client

2.2.2.1 Using MySQL CLI to Connect to an Instance Through a Private Network

If your applications are deployed on an ECS that is in the same region and VPC as your RDS for MariaDB instance, you are advised to connect to the DB instance through a floating IP address using the ECS.

This section describes how to connect a Linux ECS to a DB instance with SSL enabled or disabled through a floating IP address. SSL encrypts connections to the DB instance, making in-transit data more secure.

Prerequisites

1. You have logged in to an ECS.

- For details on how to create and log in to an ECS, see **Purchasing an ECS** and **Logging In to an ECS** in *Elastic Cloud Server Getting Started*.
- To connect to a DB instance through an ECS, you must ensure that:
  - The ECS and DB instance are in the same VPC.
  - The ECS is allowed by the security group to access the DB instance.
    - If the security group with which the 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 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.

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

      If the security group rules do not allow the access from the ECS, you need to add a security group rule, allowing the ECS to access the DB instance.

2. You have installed a database client to connect to DB instances.

   In Linux, install a **MariaDB client** on a device that can access RDS. It is recommended that you download a MariaDB client running a version later than that of the DB instance.

Connecting to a DB Instance Using Commands (SSL Connection)

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Information** area, check whether SSL is enabled.

- If yes, go to **7**.
- If no, click  . In the displayed dialog box, click **OK** to enable SSL. Then, go to **6**.

Step 6 Click  next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Step 7 Import the root certificate **ca.pem** to the Linux or Windows. For details, see **How Can I Import the Root Certificate to a Windows or Linux OS?**

Step 8 Connect to the RDS for MariaDB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p --ssl-ca=**<*caName*>

Example:

mysql -h 172.16.0.31 -P 3306 -u root -p --ssl-ca=ca.pem

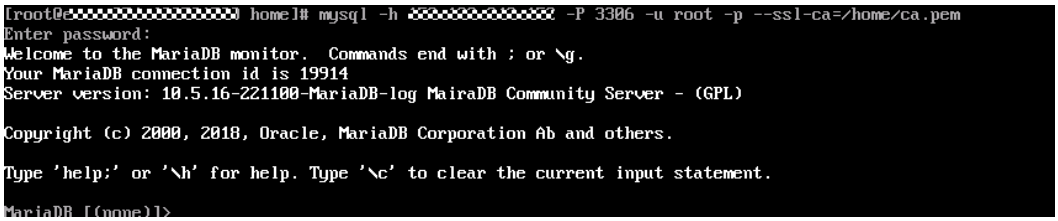
Table 2-2 Parameter description

Parameter	Description
<host>	Floating IP address. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the floating IP address in the **Connection Information** area.
<port>	Database port. By default, the value is **3306**. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the database port in the **Connection Information** area.
<userName>	Username of the database account used for logging in to the DB instance. The default value is **root**.
<caName>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

Step 9 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-1 Connection example



----End

Connecting to a DB Instance Using Commands (Non-SSL Connection)

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Information** area, check whether SSL is enabled.

- If yes, click . In the displayed dialog box, click **OK** to disable SSL. Then go to **6**.
- If no, go to **6**.

Step 6 Connect to the RDS for MariaDB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p**

Example:

mysql -h 172.16.0.31 -P 3306 -u root -p

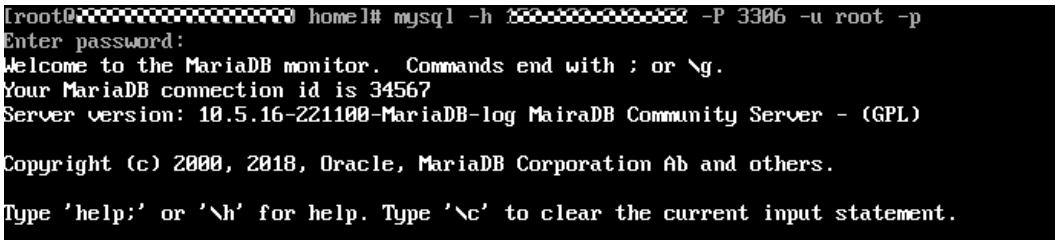
Table 2-3 Parameter description

Parameter	Description
<*host*>	Floating IP address. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the floating IP address in the **Connection Information** area.
<*port*>	Database port. By default, the value is **3306**. To obtain this parameter value, go to the **Basic Information** page of the DB instance. You can find the database port in the **Connection Information** area.
<*userName*>	Username of the database account used for logging in to the DB instance. The default value is **root**.

Step 7 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-2 Non-SSL connection example



----End

2.2.2.2 Using MySQL CLI to Connect to an Instance Through a Public Network

If you cannot access your DB instance through a floating IP address, bind an EIP to the DB instance and connect to it through the EIP.

This section describes how to connect a Linux ECS to a DB instance with SSL enabled or disabled through an EIP. SSL encrypts connections to the DB instance, making in-transit data more secure.

Prerequisites

1. You have bound an EIP to the target DB instance and configured security group rules.

a. Bind an EIP to the target DB instance.
b. Obtain the IP address of the ECS you use to connect to the DB instance.
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.
d. Run the **ping** command to ping the EIP bound in **1.a** to ensure that the EIP is accessible through the ECS.

2. You have installed a database client to connect to DB instances.

   In Linux, you need to install a **MariaDB client** on your device. It is recommended that you download a MariaDB client running a version later than that of the DB instance.

Connecting to a DB Instance Using Commands (SSL Connection)

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Information** area, check whether SSL is enabled.

- If yes, go to **Step 6**.
- If no, click  . In the displayed dialog box, click **OK** to enable SSL. Then go to **Step 6**.

Step 6 Click  next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Step 7 Import the root certificate **ca.pem** to the Linux or Windows. For details, see **How Can I Import the Root Certificate to a Windows or Linux OS?**

Step 8 Connect to the RDS for MariaDB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p --ssl-ca=**<*caName*>

Example:

mysql -h 172.16.0.31 -P 3306-u root -p --ssl-ca=ca.pem

Table 2-4 Parameter description

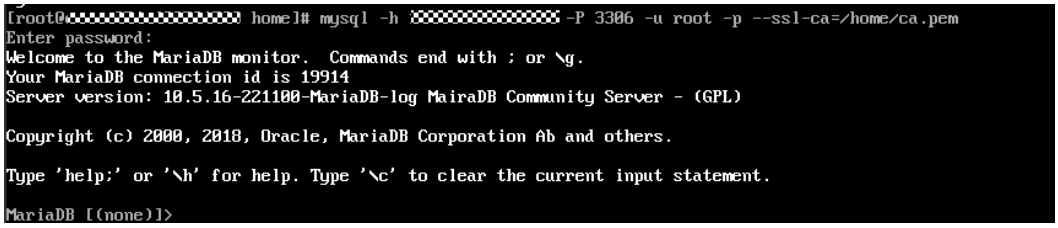
Parameter	Description
<*host*>	EIP of the DB instance to be connected.
<*port*>	Port of the DB instance to be connected.

Parameter	Description
<*userName*>	Username of the database account used for logging in to the DB instance. The default value is **root**.
<*caName*>	Name of the CA certificate. The certificate should be stored in the directory where the command is executed.

Step 9 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-3 Connection example



NOTE

If the connection fails, ensure that all **prerequisites** are correctly configured and try again.

----End

Connecting to a DB Instance Using Commands (Non-SSL Connection)

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

Step 5 In the **DB Information** area, check whether SSL is enabled.

- If yes, click . In the displayed dialog box, click **OK** to disable SSL. Then go to **6**.
- If no, go to **6**.

Step 6 Connect to the RDS DB instance. In Linux, for example, run the following command:

mysql -h <*host*> **-P** <*port*> **-u** <*userName*> **-p**

Example:

mysql -h 172.16.0.31 -P 3306 -u root -p

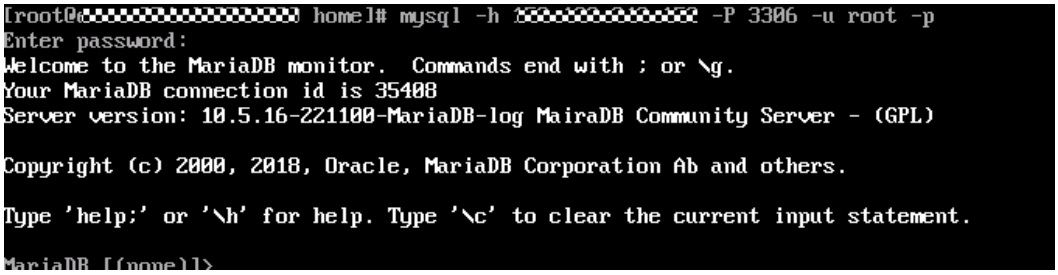
Table 2-5 Parameter description

Parameter	Description
<host>	EIP of the DB instance to be connected.
<port>	Port of the DB instance to be connected.
<userName>	Username of the database account. The default administrator is **root**.

Step 7 Enter the password of the database account if the following information is displayed:

```
Enter password:
```

Figure 2-4 Non-SSL connection example



NOTE

If the connection fails, ensure that preparations have been correctly made in **Prerequisites** and try again.

----End

2.2.3 Connecting to an RDS for MariaDB Instance Through JDBC

If you are connecting to an instance through JDBC, an SSL certificate is optional, but using an SSL certificate can improve the security of your data. SSL is disabled by default for RDS for MariaDB instances. It encrypts connections to databases but increases the connection response time and CPU usage. For this reason, you are advised not to enable SSL.

Prerequisites

You are familiar with:

- Computer basics.
- Java.
- JDBC.

Connection with the SSL Certificate

NOTE

Download the SSL certificate and verify it before connecting to your instance.

Step 1 Download the CA certificate or certificate bundle.

1. On the **Instances** page, click the instance name to go to the **Basic Information** page.
2. In the **DB Information** area, click on the right of the SSL switch.

Step 2 Use keytool to generate a truststore file using the CA certificate.

```
<keytool_installation_path> ./keytool.exe -importcert -alias <MariaDBCACert> -file <ca.pem> -keystore <truststore_file> -storepass <password>
```

Table 2-6 Parameter description

Parameter	Description
<keytool installation path>	Bin directory in the JDK or JRE installation path, for example, **C:\Program Files (x86)\Java\jdk11.0.7\bin**.
<MariaDBCACert>	Name of the truststore file. Set it to a name specific to the service for future identification.
<ca.pem>	Name of the CA certificate downloaded and decompressed in **Step 1**, for example, ca.pem.
<truststore_file>	Path for storing the truststore file.
<password>	Password of the truststore file.

Code example (using keytool in the JDK installation path to generate the truststore file):

```
Owner:  CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Issuer: CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Serial number: 1
Valid from: Thu Feb 16 11:42:43 EST 2017 until: Sun Feb 14 11:42:43 EST 2027
Certificate fingerprints:
    MD5: 18:87:97:37:EA:CB:0B:5A:24:AB:27:76:45:A4:78:C1
    SHA1: 2B:0D:D9:69:2C:99:BF:1E:2A:25:4E:8D:2D:38:B8:70:66:47:FA:ED
    SHA256:C3:29:67:1B:E5:37:06:F7:A9:93:DF:C7:B3:27:5E:09:C7:FD:EE:2D:18:86:F4:9C:40:D8:26:CB:DA:95:
A0:24
    Signature algorithm name: SHA256withRSA Subject Public Key Algorithm: 2048-bit RSA key
    Version: 1
    Trust this certificate? [no]: y
    Certificate was added to keystore
```

Step 3 Connect to your RDS for MariaDB instance through JDBC.

```
jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?param1=value1&param2=value2
```


Table 2-7 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** • If you are accessing the DB instance through an ECS, *instance_ip* is the floating IP address of the instance. You can view the floating IP address in the **Connection Information** area on the **Basic Information** or **Connectivity & Security** page. • If you are accessing the DB instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can view the EIP in the **Connection Information** area on the **Connectivity & Security** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can view the database port in the **Connection Information** area on the **Connectivity & Security** page.
<database_name>	Database name used for connecting to the DB instance. The default value is **MariaDB**.
<param1>	**requireSSL**, indicating whether the server supports SSL. Its value can be either of the following: • **true**: The server supports SSL. • **false**: The server does not support SSL. **NOTE** For details about the relationship between **requireSSL** and **sslmode**, see **Table 2-8**.
<param2>	**useSSL**, indicating whether the client uses SSL to connect to the server. Its value can be either of the following: • **true**: The client uses SSL to connect to the server. • **false**: The client does not use SSL to connect to the server. **NOTE** For details about the relationship between **useSSL** and **sslmode**, see **Table 2-8**.
<param3>	**verifyServerCertificate**, indicating whether the client verifies the server certificate. Its value can be either of the following: • **true**: The client verifies the server certificate. • **false**: The client does not verify the server certificate. **NOTE** For details about the relationship between **verifyServerCertificate** and **sslmode**, see **Table 2-8**.
<param4>	**trustCertificateKeyStoreUrl**. Its value is **file:<truststore_file>**. <truststore_file> is the path for storing the truststore file set in **Step 2**.

Parameter	Description
<param5>	**trustCertificateKeyStorePassword**. Its value is the password of the truststore file set in **Step 2**.

Table 2-8 Relationship between connection parameters and sslmode

useSSL	requireSSL	verifyServerCertificate	sslMode
false	N/A	N/A	DISABLED
true	false	false	PREFERRED
true	true	false	REQUIRED
true	N/A	true	VERIFY_CA

Code example (Java code for connecting to an RDS for MariaDB instance):

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.SQLException;

// There will be security risks if the username and password used for authentication are directly written into code. Store them in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.
public class JDBCTest {
    String USER = System.getenv("EXAMPLE_USERNAME_ENV");
    String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");

    public static void main(String[] args) {
        Connection conn = null;
        Statement stmt = null;
      // Set the required parameters in the URL based on the site requirements.
       String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
param1=value1&param2=value2";

       try {

          Class.forName("com.MariaDB.cj.jdbc.Driver");
          conn = DriverManager.getConnection(url, USER, PASS);

          stmt = conn.createStatement();
          String sql = "show status like 'ssl%'";
          ResultSet rs = stmt.executeQuery(sql);

          int columns = rs.getMetaData().getColumnCount();
          for (int i = 1; i <= columns; i++) {
              System.out.print(rs.getMetaData().getColumnName(i));
              System.out.print("\t");
          }

          while (rs.next()) {
              System.out.println();
              for (int i = 1; i <= columns; i++) {
                  System.out.print(rs.getObject(i));
```



```
                System.out.print("\t");
            }
        }
        rs.close();
        stmt.close();
        conn.close();
    } catch (SQLException se) {
        se.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        // release resource ....
    }
  }
}
```

----End

Connection Without the SSL Certificate

NOTE

You do not need to download the SSL certificate because certificate verification on the server is not required.

Connect to the RDS for MariaDB instance through JDBC.

```
jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?useSSL=false
```

Table 2-9 Parameter description

Parameter	Description
<instance_ip>	IP address of the DB instance. **NOTE** • If you are accessing the DB instance through an ECS, *instance_ip* indicates the floating IP address of the instance. You can view the floating IP address in the **Connection Information** area on the **Basic Information** or **Connectivity & Security** page. • If you are accessing the DB instance through a public network, *instance_ip* indicates the EIP that has been bound to the instance. You can view the EIP in the **Connection Information** area on the **Connectivity & Security** page.
<instance_port>	Database port of the DB instance. The default port is **3306**. **NOTE** You can view the database port in the **Connection Information** area on the **Connectivity & Security** page.
<database_name>	Database name used for connecting to the DB instance. The default value is **MariaDB**.

Code example (Java code for connecting to an RDS for MariaDB instance):

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
// There will be security risks if the username and password used for authentication are directly written into code. Store the username and password in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables. Before running this
```



```
example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.

public class MyConnTest {
    final public static void main(String[] args) {
        Connection conn = null;
                // Set the required parameters in the URL based on the site requirements.
                String url = "jdbc:mysql://<instance_ip>:<instance_port>/<database_name>?
param1=value1&param2=value2";
                String USER = System.getenv("EXAMPLE_USERNAME_ENV");
                String PASS = System.getenv("EXAMPLE_PASSWORD_ENV");
        try {
            Class.forName("com.MariaDB.jdbc.Driver");
            conn = DriverManager.getConnection(url,USER,PASS);
            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 ....
        }
    }
}
```

Related Issues

- Symptom

  When you use JDK 8.0 or a later version to connect to an RDS for MariaDB instance with an SSL certificate downloaded, an error similar to the following is reported:

  ```
  javax.net.ssl.SSLHandshakeException: No appropriate protocol (protocol is disabled or
  cipher suites are inappropriate)
      at sun.security.ssl.HandshakeContext.<init>(HandshakeContext.java:171) ~[na:1.8.0_292]
      at sun.security.ssl.ClientHandshakeContext.<init>(ClientHandshakeContext.java:98) ~
  [na:1.8.0_292]
      at sun.security.ssl.TransportContext.kickstart(TransportContext.java:220) ~
  [na:1.8.0_292]
      at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:428) ~
  [na:1.8.0_292]
      at
  com.MariaDB.cj.protocol.ExportControlled.performTlsHandshake(ExportControlled.java:316) ~
  [MariaDB-connector-java-8.0.17.jar:8.0.17]
      at
  com.MariaDB.cj.protocol.StandardSocketFactory.performTlsHandshake(StandardSocketFactory.java
  :188) ~[MariaDB-connector-java8.0.17.jar:8.0.17]
      at
  com.MariaDB.cj.protocol.a.NativeSocketConnection.performTlsHandshake(NativeSocketConnection.
  java:99) ~[MariaDB-connector-java8.0.17.jar:8.0.17]
      at
  com.MariaDB.cj.protocol.a.NativeProtocol.negotiateSSLConnection(NativeProtocol.java:331) ~
  [MariaDB-connector-java8.0.17.jar:8.0.17]
  ... 68 common frames omitted
  ```

- Solution

  Specify the corresponding parameter values in the code link of **Step 3** based on the JAR package used by the client. Example:

  – MariaDB-connector-java-5.1.*xx*.jar

In the database connection URL jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2, replace **param1=value1** with **enabledTLSProtocols=TLSv1.2**.

- MariaDB-connector-java-8.0.*xx*.jar
  In the database connection URL jdbc:**mysql**://*<instance_ip>*:*<instance_port>*/*<database_name>*?param1=value1&param2=value2, replace **param1=value1** with **tlsVersions=TLSv1.2**.

2.2.4 Connecting to an RDS for MariaDB Instance Through DAS

Scenarios

Data Admin Service (DAS) enables you to connect to and manage DB instances with ease on a web-based console. The permission required for connecting to DB instances through DAS has been enabled for you by default. Using DAS to connect to your DB instance is recommended, which is more secure and convenient.

Procedure

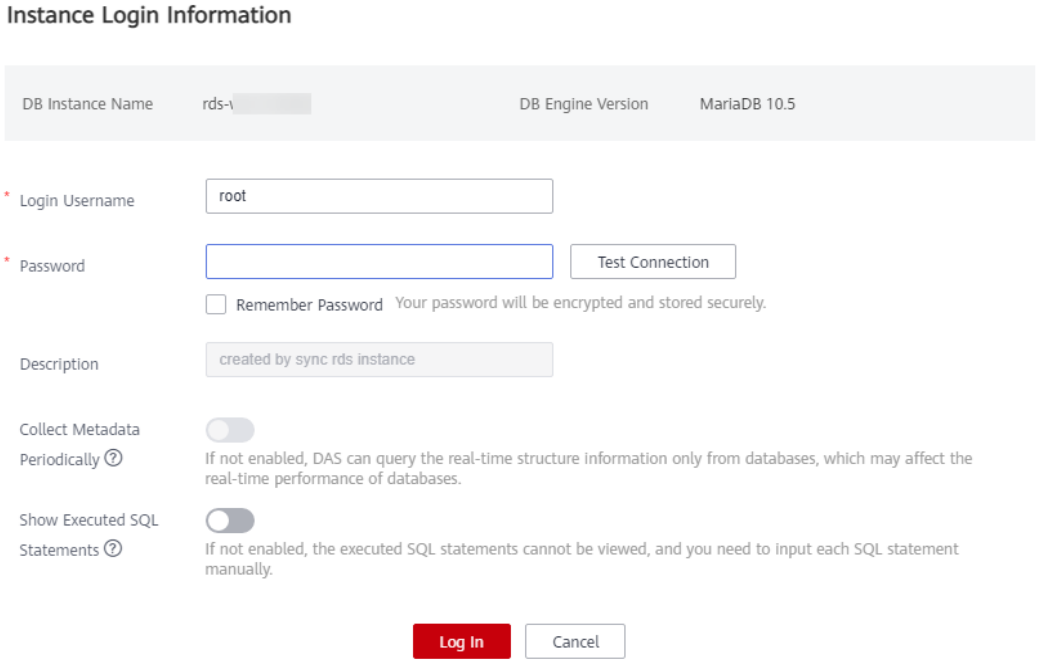
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Figure 2-5 Login page



Step 5 Enter the database username and password and click **Test Connection**.

Step 6 After the connection test is successful, click **Log In**.

For details about how to manage databases using DAS, see **RDS for MariaDB Database Management** in the *Data Admin Service User Guide*.

----End

2.3 Performance Tuning

2.3.1 What Is the Maximum Number of IOPS Supported by RDS?

The IOPS supported by RDS depends on the I/O performance of EVS disks. For details, see **Disk Types and Performance** in *Elastic Volume Service Service Overview*.

2.3.2 How Do I Improve the Query Speed of My RDS Database?

The following are some suggestions provided for you to improve the database query speed:

- View the slow query logs to check if there are any slow queries, and view their performance characteristics to locate the cause. For details about how to view RDS for MariaDB logs, see **Viewing and Downloading Slow Query Logs**.
- View the CPU usage of your DB instance to facilitate troubleshooting. For details, see **Viewing Monitoring Metrics**.
- Create read replicas to offload read pressure on the primary DB instance.
- Increase the CPU or memory specifications for DB instances with high loads. For details, see **Changing a DB Instance Class**.
- Add indexes for associated fields in multi-table association queries.
- Specify a field or add a WHERE clause, which will prevent full table scanning triggered by the SELECT statement.

2.3.3 Identifying Why CPU Usage of RDS for MariaDB Instances Is High and Providing Solutions

If the CPU usage of your RDS for MariaDB instance is high or close to 100%, database performance deteriorates. For example, data read/write becomes slow, connecting to the instance takes a longer time, or errors are reported when you are trying to delete data.

Solution

Analyze slow SQL logs and CPU usage to locate slow queries and then optimize them.

1. View the slow SQL logs to check for slowly executed SQL queries and view their performance characteristics (if any) to locate the cause.

For details about how to view RDS for MariaDB logs, see **Viewing and Downloading Slow Query Logs**.

2. View the CPU usage of your DB instance.

   For details, see **Viewing Monitoring Metrics**.

3. Create read replicas to reduce read pressure from primary DB instances.
4. Add indexes for associated fields in multi-table association queries.
5. Do not use the SELECT statement to scan all tables. You can specify fields or add the WHERE condition.

2.3.4 RDS for MariaDB Memory Usage Too High

For a DB instance storing mission-critical application data

Scale up the instance class.

For a DB instance not storing mission-critical application data

Check the memory usage of the local computer. If the memory usage curve is stable, no action is required.

For a DB instance storing mission-critical application data and configured with a large instance class

1. During off-peak hours, change the value of **performance_schema** to **OFF**. You need to reboot the instance for the change to take effect.
2. View the memory usage of your instance using DBA Assistant. For details, see **Viewing Performance Metrics**.

   If the space usage remains high, perform either of the following operations:

   - Scale up the instance class.
   - Change the **innodb_buffer_pool_size** value:
     - If the instance memory is 2 GB, change **innodb_buffer_pool_size** to **268,435,456** in byte (256 MB).
     - If the instance memory is 4 GB, change **innodb_buffer_pool_size** to **1,073,741,824** in byte (1 GB).
     - If the instance memory is 8 GB, change **innodb_buffer_pool_size** to **3,221,225,472** in byte (3 GB).
     - If the instance memory is greater than 8 GB, you do not need to adjust the **innodb_buffer_pool_size** value.

NOTICE

- Change the value of **innodb_buffer_pool_size** as needed.
- MariaDB has a dynamic memory balancing mechanism. If the memory usage is less than 90%, no action is required.
- RDS for MariaDB memory is allocated to the engine layer and server layer.
  - The memory allocated to the engine layer includes the InnoDB buffer pool, log buffer, and full text index cache. The InnoDB buffer pool is resident memory and accounts for a large proportion.

    The InnoDB buffer pool is a memory area that holds cached InnoDB data for tables, indexes, and other auxiliary buffers. You can use the **innodb_buffer_pool_size** parameter to define the buffer pool size.
  - The memory allocated to the server layer is occupied by the thread cache, binlog cache, sort buffer, read buffer, and join buffer. These caches and buffers are usually released when connections are closed.

  Such memory allocation keeps memory usage of a running RDS for MariaDB instance at about 80%.

2.3.5 What Should I Do If an RDS DB Instance Is Abnormal Due to Full Storage Space?

You can scale up storage space if it is no longer sufficient for your requirements. If the DB instance status is **Storage full** and no more data can be written to the database, the DB instance will be abnormal.

Solution

1. As your application data grows, the original storage space may be insufficient. You are advised to scale up storage space by referring to **Scaling Up Storage Space**.

   You can view the memory usage of your instance using DBA Assistant. For details, see **Viewing Storage Usage**.

   If the storage capacity has reached the upper limit of your DB instance class, change the instance class first.

   For details, see **Changing a DB Instance Class**.
2. Delete expired data files in a timely manner.
3. View performance metrics of your DB instance on the console, such as CPU, memory, storage, and connections. You can also set alarm rules for metric thresholds to identify risks in advance.

   For details, see **Viewing the Overall Status of a DB Instance**.

2.3.6 Troubleshooting Slow SQL Issues for RDS for MariaDB Instances

This section describes how to troubleshoot slow SQL statements on RDS for MariaDB instances. For any given service scenario, query efficiency depends on the architecture and on the database table and index design. Poorly designed architecture and indexes will cause many slow SQL statements.

Slow SQL Statements Caused by SQL Exceptions

- Causes and symptoms

  There are many causes for SQL exceptions, for example, unsuitable database table structure design, missing indexes, or too many rows that need to be scanned.

  On the **Slow Query Logs** page, you can download logs to identify the slow SQL statements and see how long they took to execute. For details, see **Viewing and Downloading Slow Query Logs**.

- Solution

  Optimize the SQL statements that you need to execute.

Slow SQL Statements Caused by DB Instance Limits

- Causes and symptoms

  DB instance performance can be limited because:

  - Your workloads have been increasing but the storage has not been scaled up accordingly.
  - The performance of your DB instance has been deteriorating as the physical server of the instance ages.
  - The amount of data has been increasing, and the data structure has been changing.

  You can view the resource usage of the DB instance on the console. If the values of all resource usage metrics are close to 100%, your DB instance may reach its maximum performance. For details, see **Viewing the Overall Status of a DB Instance**.

- Solution

  Upgrade the instance class. For details, see **Changing a DB Instance Class**.

Slow SQL Statements Caused by Inappropriate Parameter Settings

- Causes and symptoms

  Inappropriate settings of some parameters (such as **innodb_spin_wait_delay**) can impact performance.

  You can view parameter modifications on the console. For details, see **Viewing Parameter Change History**.

- Solution

  Modify related parameters based on your specific service scenario.

Slow SQL Statements Caused by Batch Operations

- Causes and symptoms

  A large number of operations are performed to import, delete, and query data.

  You can view **Total Storage Space**, **Storage Space Usage**, and **IOPS** on the console. For details, see **Viewing the Overall Status of a DB Instance**.

- Solution

  Perform batch operations during off-peak hours, or split them.

Slow SQL Statements Caused by Scheduled Tasks

- Causes and symptoms

  If the load of your DB instance changes regularly over time, there may be scheduled tasks causing this.

  You can view **DELETE Statements per Second**, **INSERT Statements per Second**, **INSERT_SELECT Statements per Second**, **REPLACE Statements per Second**, **REPLACE_SELECTION Statements per Second**, **SELECT Statements per Second**, and **UPDATE Statements per Second** on the console to determine whether the load has been changing regularly. For details, see **Viewing Monitoring Metrics**.

- Solution

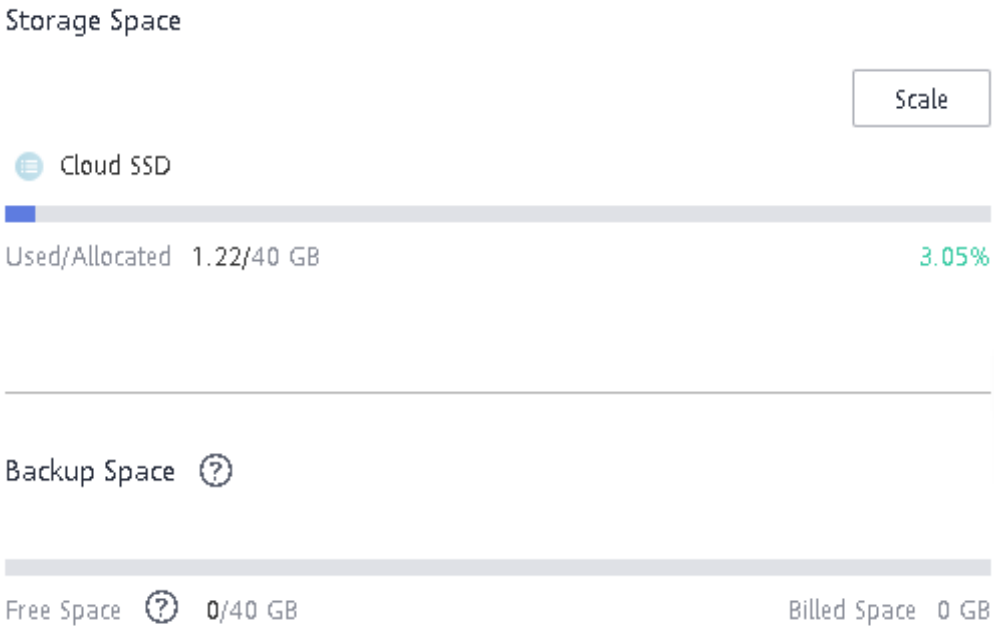
  Adjust the time when scheduled tasks are run. You are advised to run scheduled tasks during off-peak hours.

2.3.7 Resolving Insufficient Storage Issues for RDS for MariaDB Instances

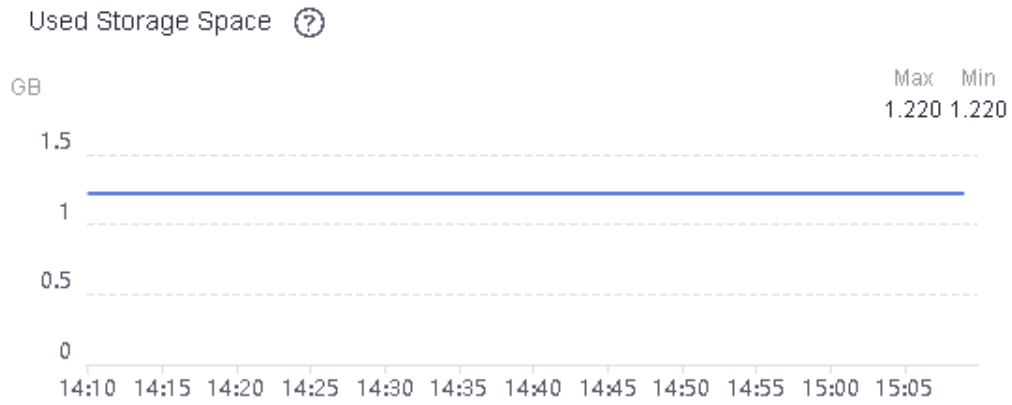
Storage usage is an important metric for measuring the performance of your DB instances. If the available storage space is insufficient, your DB instance may encounter serious issues. For example, data cannot be backed up or written into databases, or scaling up storage takes an extended period of time.

Viewing Storage Space Usage

- On the **Basic Information** page, you can see how much of your instance storage and backup space has been used. However, this page does not provide any details about what different types of data are being stored.



- To view the historical usage and the changes over time, click **View Metrics** on the **Basic Information** page.



Insufficient Storage Caused by Excessive Indexes

- Cause and symptom

  In most cases, a table contains primary key indexes and secondary indexes. More secondary indexes mean that the table takes up more space.

- Solution

  Optimize the data structure of the table to reduce the number of secondary indexes.

Insufficient Storage Caused by Large Fields

- Cause and symptom

  If large fields of the binary large object (BLOB), TEXT, or VARCHAR data type are defined in the schema of a table, the table takes up a lot of space.

- Solution

  Compress data before you insert the data into the table.

Insufficient Storage Caused by Excessive Idle Tablespaces

- Cause and symptom

  If the fragmentation ratio of an InnoDB table is high, there will be an excessive number of idle tablespaces. InnoDB manages tablespaces by page. If some records of a full page are deleted and no new records are inserted into the positions these records were deleted from, a large number of tablespaces will be idle.

- Solution

  Run the **show table status like** '<*Name of the table*>'**;** command to query idle tablespaces. If there are too many tablespaces, run the **optimize table** '<*Name of the table*>'**;** command to manage the tablespaces.

Insufficient Storage Caused by Excessively Large Temporary Tables

- Cause and symptom

  When you perform a semi-join, distinct, or sort operation without using an index on a table, a temporary table is created. If the temporary table contains an excessive amount of data, the storage usage for the temporary table may be excessively high.

When you execute data definition language (DDL) statements to rebuild tablespaces that are used to store the data of a large table, the temporary table that is generated from an index-based sort operation will also be large.

- Solution

  View the plans that the DDL statements were based on. Check whether the **Using Temporary** field was specified.

  Before you execute DDL statements on large tables, check whether your DB instance provides sufficient storage space. If the available storage space is insufficient, scale up the storage space of your DB instance before executing the statements.

2.4 Permissions Management

2.4.1 Creating a User and Granting Permissions

This chapter describes how to use **Identity and Access Management (IAM)** for fine-grained permissions management for your RDS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing RDS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your RDS resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

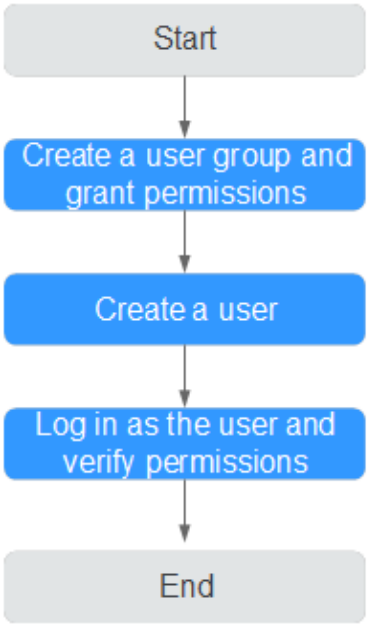
This section describes the procedure for granting permissions (see **Figure 2-6**).

Prerequisites

Learn about the permissions (see **Permissions**) supported by RDS and choose policies or roles according to your requirements. For the system policies of other services, see **System-defined Permissions**.

Process Flow

Figure 2-6 Process for granting RDS permissions



1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console, and attach the **RDS ReadOnlyAccess** policy to the group.

   NOTE

   To use some interconnected services, you also need to configure permissions of such services.

   For example, to connect to your DB instance through the console, configure the **DAS FullAccess** permission of Data Admin Service (DAS) besides **RDS ReadOnlyAccess**.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the RDS console by using the created user, and verify that the user only has read permissions for RDS.

   - Choose **Service List** > **Relational Database Service** and click **Buy DB Instance**. If a message appears indicating that you have insufficient permissions to perform the operation, the **RDS ReadOnlyAccess** policy has already been applied.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **RDS ReadOnlyAccess** policy has already taken effect.

2.4.2 RDS Custom Policies

Custom policies can be created to supplement the system policies of RDS.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

The following section contains examples of common RDS custom policies.

Example Custom Policies

- Example 1: Allowing users to create RDS DB instances

```
{
   "Version": "1.1",
   "Statement": [{
      "Effect": "Allow",
      "Action": ["rds:instance:create"]
   }]
}
```

- Example 2: Denying RDS DB instance deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user include both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **RDS FullAccess** policy to a user but you want to prevent the user from deleting RDS DB instances. Create a custom policy for denying RDS DB instance deletion, and attach both policies to the group the user belongs to. Then, the user can perform all operations on RDS DB instances except deleting RDS DB instances. The following is an example deny policy:

```
{
   "Version": "1.1",
   "Statement": [{
      "Action": ["rds:instance:delete"],
      "Effect": "Deny"
   }]
}
```

2.5 Instance Lifecycle

2.5.1 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.
- A reboot task configured during the current maintenance window will not be executed until the next maintenance window.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

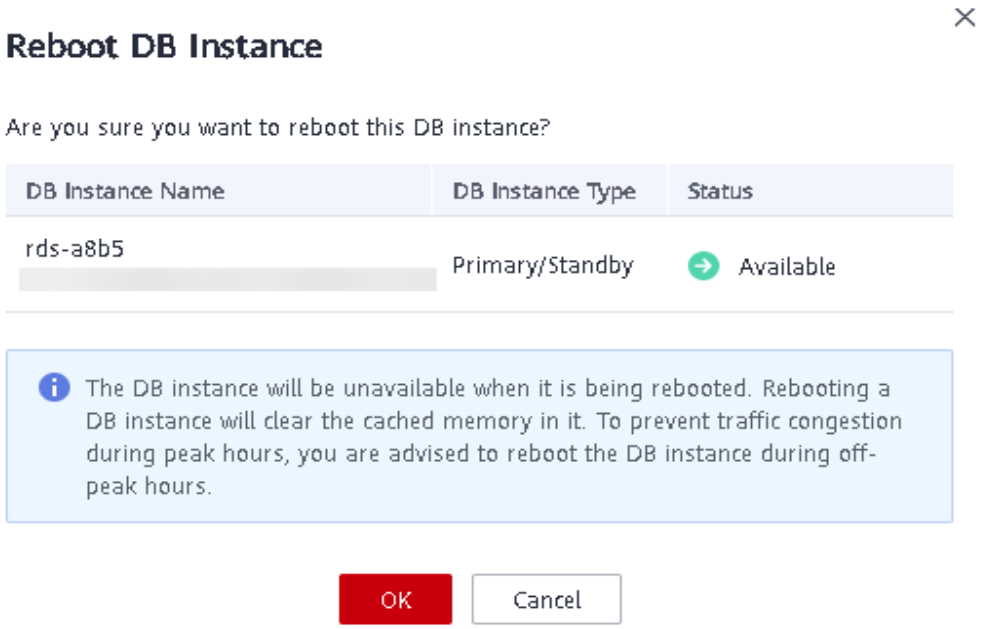
Step 4 On the **Instances** page, locate the target DB instance, or click and then locate the target read replica. Choose **More** > **Reboot** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** 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 **OK**.

Figure 2-7 Rebooting a DB instance



Step 6 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

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

----End

2.5.2 Selecting Displayed Items

Scenarios

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

Procedure

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click ⚙ above the instance list, select desired items from the custom columns, and click **OK**.

- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Disk Encryption** (submit a service ticket to apply for required permissions), **Billing Mode**, **Floating IP Address**, **Private Domain Name**, **IPv6 Address**, **Enterprise Project**, **Created**, **Database Port**, **Storage Type**, and **Operation**.

----End

2.5.3 Exporting DB Instance Information

Scenarios

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

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

Exporting Information About All DB Instances

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click **Export** above the DB instance list. By default, information about all DB instances are exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

Step 4 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, DB instance tag, enterprise project, or floating IP address, or select the DB instances to be exported, and click **Export** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

Step 4 Find a .csv file locally after the export task is completed.

----End

2.5.4 Deleting a Pay-per-Use DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas billed on the pay-per-use basis as required on the **Instances** page.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- A maximum of 50 pay-per-use DB instances can be deleted at a time.
- If you delete a pay-per-use DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and 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 a DB Instance from a Backup**.

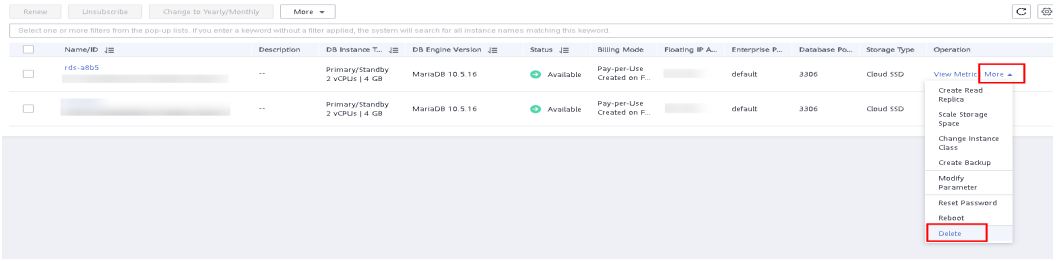
Deleting a Pay-per-Use DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.

Figure 2-8 Deleting a DB instance



Step 4 In the displayed dialog box, select **I have read this warning and understand the risks.** and click **Yes**.

Step 5 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Pay-per-Use Read Replica

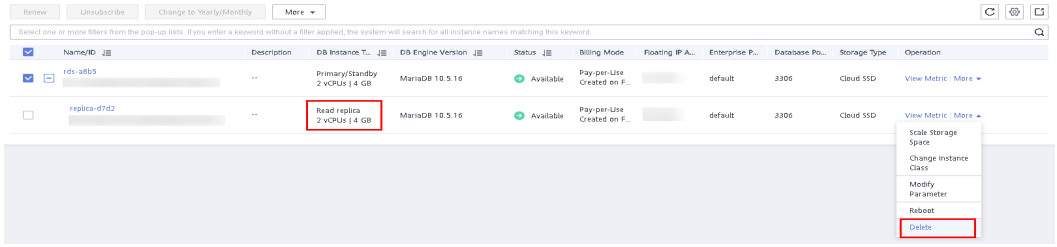
Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ⊞. All the read replicas created for the DB instance are displayed.

Step 4 Locate the read replica to be deleted and click **More** > **Delete** in the **Operation** column.

Figure 2-9 Deleting a read replica



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

Step 6 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list later to check that the deletion is successful.

----End

2.5.5 Modifying Recycling Policy

RDS allows you to move unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances to the recycle bin. You can rebuild a DB instance that was deleted up to 7 days ago from the recycle bin.

Constraints

- Read replicas cannot be moved to the recycle bin.
- A stopped instance will not be moved to the recycle bin after being deleted.
- The recycle bin is enabled by default and cannot be disabled.

Precautions

- The recycle bin is enabled by default and cannot be disabled. This function is free of charge.
- Instances in the recycle bin are retained for 7 days by default. 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.

Procedure

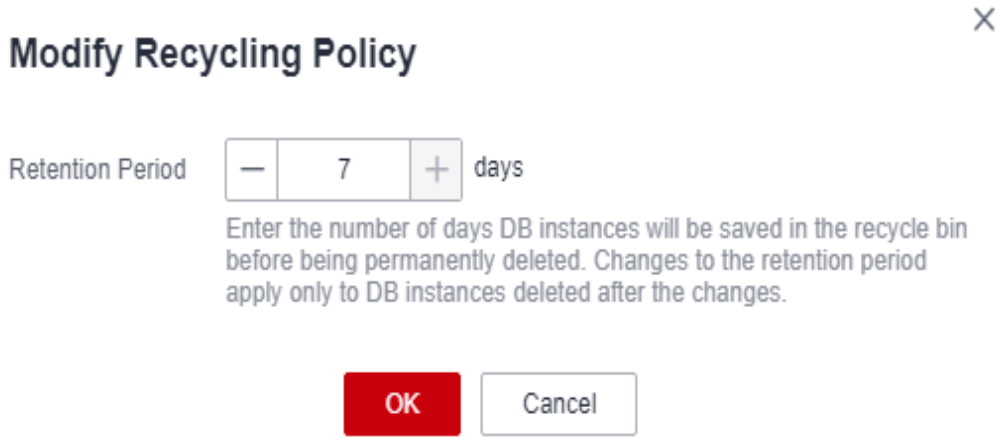
Step 1 **Log in to the management console**.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 Click **Modify Recycling Policy** and set the retention period of deleted instances. The value ranges from 1 to 7 days.

Figure 2-10 Setting the retention period



Step 5 Then, click **OK**.

----End

2.5.6 Rebuilding a DB Instance

You can rebuild DB instances that were deleted up to 7 days ago from the recycle bin. This section describes how to rebuild a DB instance.

Precautions

- Only primary/standby or single DB instances can be rebuilt.
- You can only rebuild DB instances within the retention period.
- After a DB instance is moved to the recycle bin, a full backup will be performed. You can rebuild the DB instance only after the full backup is complete.
- If resources are not renewed after expiration, you can rebuild DB instances from the recycle bin to restore data.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 In the DB instance list, locate the target DB instance and click **Rebuild** in the **Operation** column.

Step 5 On the displayed page, set required parameters and click **Next**.

- The DB engine and engine version of the new instance are the same as those of the original instance.

- The storage space of the new instance is the same as that of the original instance by default and the new instance must be at least as large as the original instance.
- Other settings are the same as those of the original instance by default and can be modified. For details, see **Buy a DB Instance**.

Step 6 Click **Submit**.

----End

2.6 Instance Modifications

2.6.1 Upgrading a Minor Version

RDS for MariaDB supports minor version upgrades. Upgrading a minor version not only fixes historical issues, but also enriches user experience. This section describes how to upgrade a minor version.

Precautions

- When any new minor version is released for addressing issues and vulnerabilities from the open source community, perform a minor version upgrade for your instance.
- The upgrade will cause the DB instance to reboot and interrupt services intermittently. To limit the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.
- If your RDS instance is involved in a DRS task, upgrading the minor version may cause the DRS task to fail.

  You are advised to check the retention period of RDS instance binlogs before upgrading the minor version.

  - If the binlogs are within the retention period, the DRS task will automatically restart after the minor version is upgraded.
  - If the binlogs are beyond the retention period, you need to reconfigure or recreate a DRS task.
- A minor version upgrade cannot be rolled back after the upgrade is complete. If the upgrade fails, the DB instance will be automatically rolled back to the source version.
- DDL operations on events, such as CREATE EVENT, DROP EVENT, and ALTER EVENT, are not allowed during a minor version upgrade.

Notes

- If the primary and standby instances are deployed in the same AZ, upgrading the minor version will trigger a primary/standby switchover. If they are deployed in different AZs, upgrading the minor version will trigger two switchovers.
- When you upgrade the minor version of a primary instance, the minor versions of read replicas (if any) will also be upgraded automatically. Read replicas cannot be upgraded separately.

- A minor version can be upgraded in minutes.
- For primary/standby DB instances, the standby DB instance is upgraded first and then the primary DB instance is upgraded afterwards.

Constraints

- If the replication delay between primary and standby DB instances is longer than 300 seconds, the minor version cannot be upgraded.
- Minor versions cannot be upgraded for DB instances with abnormal nodes.
- RDS for MariaDB DB instances with the event scheduler enabled do not support minor version upgrades. If you want to perform a minor version upgrade, disable the event scheduler first. For details, see **Enabling or Disabling Event Scheduler**.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the **DB Information** area on the **Basic Information** page, click **Upgrade Minor Version** next to the **DB Engine Version** field.

If the minor version of your DB instance is already the latest, there is no need to upgrade the minor version.

Figure 2-11 Upgrading a minor version

Step 5 In the displayed dialog box, select a scheduled time and click **OK**.

- Upon submission: The system upgrades the minor version immediately after you have submitted your upgrade request.
- In maintenance window: The system will upgrade the minor version during the maintenance window that you have configured.

----End

2.6.2 Changing a DB Instance Name

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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** 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 Information** area, click ✎ next to the **DB Instance Name** field to edit the DB instance name.

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

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

Step 4 View the results on the **Basic Information** page.

----End

2.6.3 Changing a DB Instance Description

Scenarios

After a DB instance is created, you can add a description.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the DB instance you wish to edit the description for and click ✎ in the **Description** column to make your modification. When you are finished, click **OK**.

Alternatively, click the target DB instance to go to the **Basic Information** page. In the **DB Information** area, click ✎ next to the **Description** field to edit the DB instance description.

NOTE

The DB instance description can include up to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.).

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

Step 4 View the results on the **Basic Information** page.

----End

2.6.4 Changing the Replication Mode

Scenarios

You can change the replication mode for primary/standby DB instances to **Asynchronous** or **Semi-synchronous**.

- **Asynchronous**:

  When applications update data, the primary DB instance responds to the applications immediately after data is updated. This mode provides better performance than the semi-synchronous mode.
- **Semi-synchronous** (default value):
  - When applications update data, the primary DB instance responds to the applications only after the standby DB instance receives logs, which affects database performance.
  - If the standby DB instance is abnormal, the primary DB instance waits for the response of the standby DB instance for several seconds and does not respond to write operations during this period.
    - If the standby DB instance is recovered during the waiting period, the primary DB instance starts to respond to write operations normally.
    - If the standby DB instance is not recovered during the waiting period, the replication mode is automatically switched to asynchronous. After the switchover is complete, the primary DB instance starts to respond to write operations.

Procedure

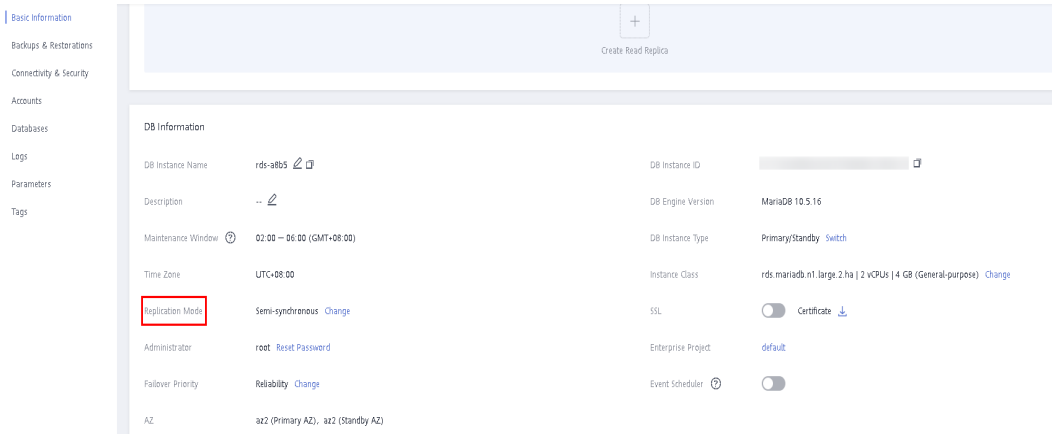
Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Replication Mode** field. In the displayed dialog box, select a mode and click **OK**.

Figure 2-12 Changing the replication mode



Step 5 On the **Basic Information** page, check for the new replication mode.

----End

2.6.5 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose highest priority is data consistency.
- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Constraints

The failover priority cannot be changed when the DB instance is stopped or its instance class is being changed.

Procedure

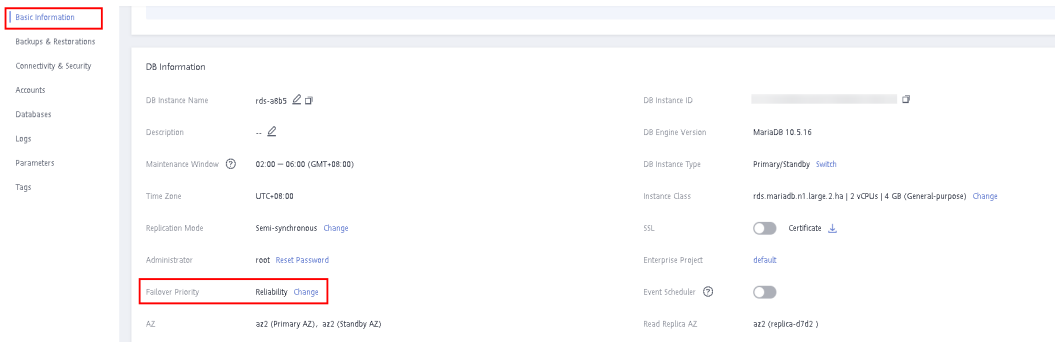
Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the primary instance name.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Change** next to the **Failover Priority** field. In the displayed dialog box, select a priority and click **OK**.

Figure 2-13 Changing the failover priority



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

----End

2.6.6 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPU 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

- You can change the DB instance class only when your account balance is greater than or equal to $0 USD.
- An instance cannot be deleted while its instance class is being changed.
- The following operations cannot be performed on an instance whose instance class is being changed: rebooting the instance, scaling up storage space, modifying the parameter template, creating a manual backup, creating a database account, and creating a database.
- You can scale up or down your RDS for MariaDB instance specifications.
- If there are any large transactions being processed during an instance class change, the change may fail.
- If the primary/standby replication delay of a DB instance is longer than 5 minutes, the instance class change will fail.
- Changing an instance class will interrupt services. Ensure that your applications support automatic reconnection. Perform this operation during off-peak hours because changing an instance class during peak hours takes much more time.
- Changing the instance class takes 5 to 15 minutes (during off-peak hours). If more time is required, submit a service ticket.

Procedure

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

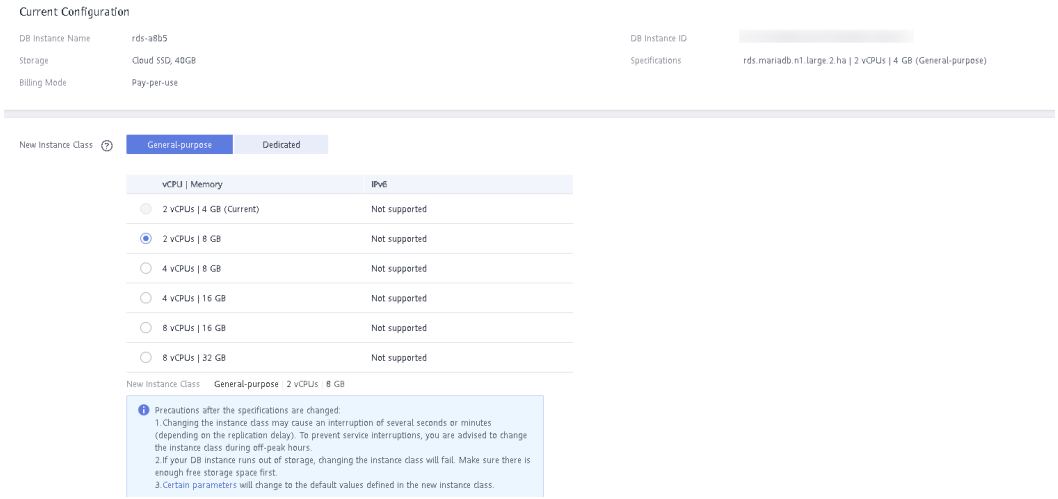
Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** 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** next to the **Instance Class** field.

Step 4 On the displayed page, specify the new instance class and click **Next**.

Figure 2-14 Changing a DB instance class



Step 5 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.

  To view the cost incurred by the DB instance class change, choose **Billing & Costs** > **Bills** in the upper right corner.
- For yearly/monthly DB instances:
  - If you intend to scale down the DB instance class, click **Submit**.

    The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 6 View the DB instance class change result.

Return to the **Instances** page and view the instance status. During the change period, the instance status is **Changing instance class**. You can view the execution progress of **Changing MariaDB instance class** on the **Task Center** page. After a few minutes, view the DB instance class on the **Basic Information** page to check that the change is successful.

----End

2.6.7 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 backup space increases with the instance scale-up.**

The DB instance needs to preserve at least 13% 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**.

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

- You can scale up storage space only when your account balance is greater than or equal to $0 USD.
- You can scale up storage space only when your instance status is **Available** or **Storage full**.
- The maximum allowed storage is 4,000 GB. If you want to increase the storage upper limit to 10 TB, submit a service ticket.
- 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.
- Reboot is not required during DB instance scaling-up.
- You cannot reboot or delete a DB instance that is being scaled up.
- Storage space can only be scaled up, not down.
- If you scale up a DB instance with the disk encrypted, the expanded storage space will be encrypted using the original encryption key.

Scaling Up a Primary DB Instance

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** 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 DB instance name to enter the **Basic Information** page. In the **Storage Space** area, click **Scale Storage Space**.
- If the storage space is full, locate the DB instance on the **Instances** page and click **Scale** in the **Status** column.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify the settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 6 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 **Instances** page will be **Scaling up**. Click the DB instance

and view the new storage space on the displayed **Basic Information** page to verify that the scale-up is successful.

For RDS for MariaDB instances, you can view the detailed progress 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ⊞ in front of it. Locate the 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 read replica name to enter the **Basic Information** page. In the **Storage Space** area, click **Scale Storage Space**.
- If the storage space is full, locate the read replica on the **Instances** page and click **Scale** in the **Status** column.

Step 4 On the displayed page, specify the new storage space and click **Next**.

The minimum increment for each scaling is 10 GB.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings and the read replica uses pay-per-use billing, click **Submit**.

Step 6 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 **Instances** page will be **Scaling up**. Click the read replica and view the new storage space on the displayed **Basic Information** page to verify that the scale-up is successful.

For RDS for MariaDB read replicas, you can view the detailed progress of the task on the **Task Center** page. For details, see section **Task Center**.

----End

2.6.8 Storage Autoscaling

With storage autoscaling enabled, when RDS detects that you are running out of database space, it automatically scales up your storage.

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

Constraints

- You can enable storage autoscaling only when your account balance is greater than or equal to $0 USD.
- The storage space can be scaled up automatically only when your instance status is **Available** or **Storage full**.
- To apply for the storage autoscaling permission, submit a service ticket by choosing **Service Tickets** > **Create Service Ticket** in the upper right corner of the management console.
- The maximum allowed storage is 4,000 GB.
- For a primary/standby DB instance, autoscaling the storage for the primary node will also autoscale the storage for the standby node.
- Storage autoscaling is unavailable when the DB instance is in any of the following statuses: changing instance class, upgrading a minor version, migrating the standby DB instance, and rebooting.

Procedure

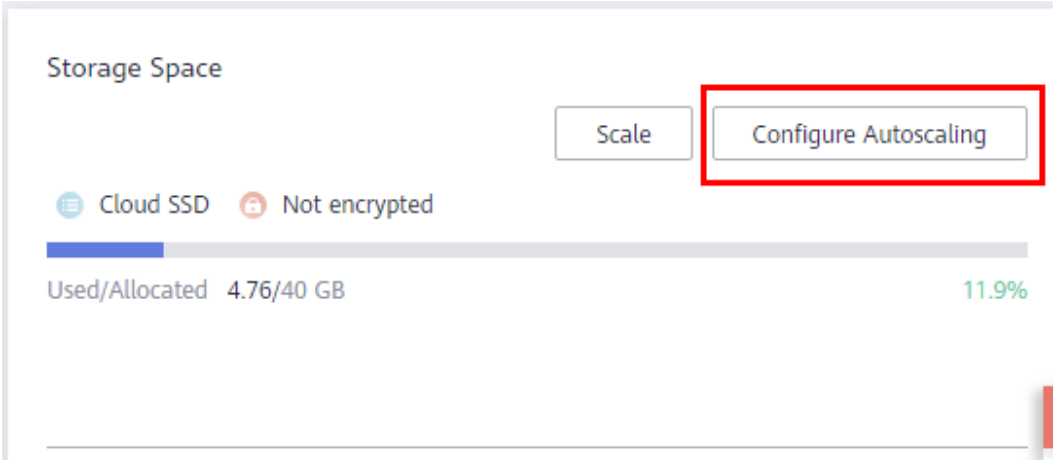
Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance or read replica (click ⊞ in front of a DB instance to locate the read replica).

Step 4 In the **Storage Space** area, click **Configure Autoscaling**. If the **Configure Autoscaling** option is not displayed, choose **Service Tickets** > **Create Service Ticket** in the upper right corner of the console to submit a request.

Figure 2-15 Storage space



Step 5 In the displayed dialog box, set the following parameters:

Figure 2-16 Configuring autoscaling

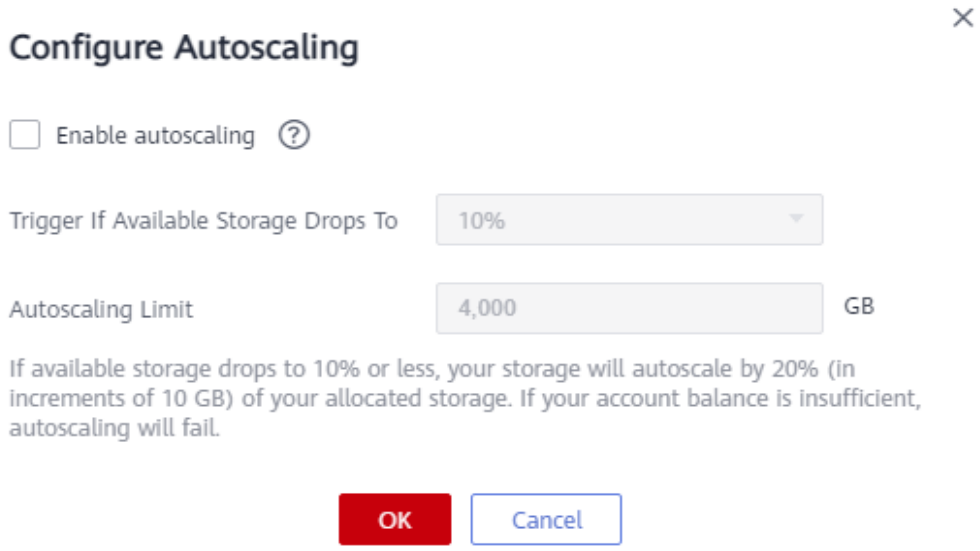


Table 2-10 Parameter description

Parameter	Description
Enable autoscaling	Select **Enable autoscaling**.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered.
Autoscaling Limit	The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance.

Step 6 Click **OK**.

----End

2.6.9 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

- The 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target primary/standby DB instance.

Step 4 In the **DB Information** area on the displayed **Basic Information** page, click **Switch** next to the **DB Instance Type** field.

NOTICE

A primary/standby switchover may cause service interruptions for several seconds or minutes (depending on the replication delay). To prevent traffic congestion, you are advised to perform a switchover during off-peak hours.

Step 5 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

Step 6 In the displayed dialog box, click **OK**.

Step 7 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, 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

2.6.10 Changing the Maintenance Window

Scenarios

The maintenance window is 02:00–06:00 by default and you can change it as required.

Precautions

- Before maintenance is performed, RDS will send SMS messages and emails to the contact person that has been set in the Huawei account.

- During the maintenance window, the DB instance will be intermittently disconnected for one or two times. Ensure that your applications support automatic reconnection.
- To prevent service interruption, you are advised to set the maintenance window to off-peak hours.

Procedure

Step 1 Log in to the management console.

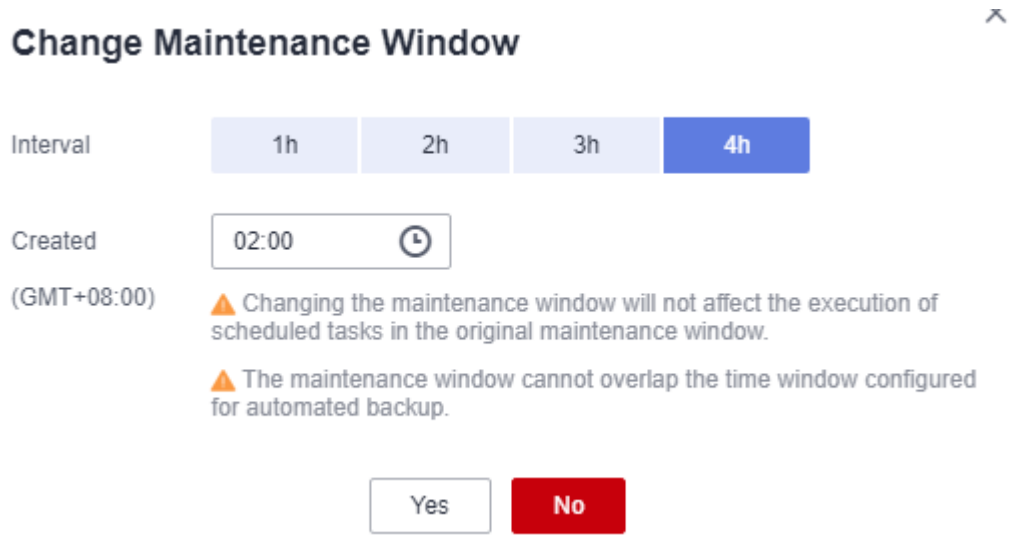
Step 2 Click in the upper left corner and select a region and a project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance. In the **DB Information** area on the **Basic Information** page, click **Change** next to the **Maintenance Window** field.

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

Figure 2-17 Changing the maintenance window



NOTE

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

----End

2.7 Read Replicas

2.7.1 Introducing Read Replicas

Introduction

RDS for MariaDB 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 offload read pressure on the primary DB instance, you can create one or more read replicas in the same region as the primary instance. 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 standby node). Changes to the primary DB instance are also automatically synchronized to all associated read replicas through the native MariaDB 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.

Applicable Scenario

In read-intensive scenarios, read replicas help offload read pressure from the primary instance.

Data is replicated from the primary instance to read replicas asynchronously. Although there is a replication delay, the data on read replicas will eventually be consistent with that on the primary instance. You can use read replicas if you do not mind such a replication delay.

Billing Standards

Read replicas are billed on a pay-per-use basis.

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.
- Pay-per-use billing is supported. You only pay for what you use.
- Read replicas support system performance monitoring.

  RDS provides up to 20 monitoring metrics, including storage space, IOPS, database connections, CPU usage, and network traffic. You can view these metrics to learn about the load on read replicas.

Constraints

- Up to five read replicas can be created for a DB instance. To create up to 10 read replicas for a DB instance, submit a service ticket to apply for the required permissions.
- Yearly/monthly billing is not supported.
- You can purchase read replicas only for your created DB instance.

- All databases and tables in the primary instance are synchronized to read replicas. Data of the primary instance, standby instance, and read replicas is consistent.
- 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. Create database accounts on the primary DB instance. For details, see **Creating a Database Account**.
- Read replicas cannot be recycled after they are deleted.

Creating and Managing a Read Replica

- **Creating a Read Replica**
- **Creating Read Replicas in Batches**
- **Managing a Read Replica**

2.7.2 Creating a Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

After an RDS instance is created, you can create read replicas for it as required.

Constraints

By default, up to five read replicas can be created for each DB instance.

For details about how to create read replicas in batches, see **Creating Read Replicas in Batches**.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** 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 4 On the displayed page, configure required parameters and click **Next**.

Table 2-11 Basic information

Parameter	Description
Billing Mode	Yearly/monthly billing and pay-per-use billing are supported.
Region	By default, read replicas are in the same region as your DB instance.
DB Instance Name	Must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
DB Engine	Same as the DB engine of your DB instance by default and cannot be changed.
DB Engine Version	Same as the DB engine version of your DB instance by default and cannot be changed.
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. **Cloud SSD**: cloud drives used to decouple storage from compute. The maximum throughput is 350 MB/s.
AZ	RDS allows you to deploy your DB instance and read replicas in a single AZ or across AZs to improve reliability.

Table 2-12 Instance specifications

Parameter	Description
Instance Class	Refers to the vCPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS. After a DB instance is created, you can change its instance class. For details, see **Changing a DB Instance Class**.
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.

Table 2-13 Network

Parameter	Description
VPC	Same as the primary DB instance's VPC.

Parameter	Description
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 security group.

Table 2-14 Enterprise project and tags

Parameter	Description
Enterprise Project	If your account has been associated with an enterprise project, select the target project from the **Enterprise Project** drop-down list. For more information about enterprise projects, see ***Enterprise Management User Guide***.
Tag	Optional. Tags help you easily identify and manage your read replicas. A maximum of 20 tags can be added for each read replica. After a read replica is created, you can view its tag details on the **Tags** page. For details, see **Managing Tags**.

Table 2-15 Yearly/monthly read replicas

Parameter	Description
Required Duration	The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	• By default, this option is not selected. • If you select this option, the auto-renew cycle is determined by the selected required duration.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for pay-per-use read replicas.
- For yearly/monthly read replicas, click **Pay Now**.

Step 6 After a read replica is created, you can view and manage it.

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

----End

Follow-up Operations

Managing a Read Replica

2.7.3 Creating Read Replicas in Batches

Scenarios

Read replicas are used to enhance read capabilities and reduce the load on primary DB instances. On the **Instances** page, you can select one or more DB instances and create read replicas for them in batches.

NOTE

- To create read replicas in batches, submit a service ticket to apply for required permissions.
- You can create a maximum of five read replicas for each primary DB instance.
- You can create read replicas for a maximum of 50 DB instances at a time.
- Read replicas can be created in batches only for RDS for MariaDB instances running the same database version.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, select one or more DB instances and choose **More** > **Create Read Replica** above the instance list.

Step 4 On the displayed page, configure required information and click **Next**.

- 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.
- Each account can create no more than 5 read replicas total for any given DB instance. In a batch creation, the number of read replicas you can create is limited by whichever DB instance already has the most replicas.

  For example, in a batch creation where most of the DB instances only have a single read replica, if any DB instance in the batch has more than one, for example, 3, you would only be able to add 2 more replicas for each DB instance in that particular batch operation.

Step 5 Confirm specifications.

- If you need to modify your settings, click **Previous**.

- If you do not need to modify your settings, click **Submit** for pay-per-use read replicas.
- For yearly/monthly read replicas, click **Pay Now**.

Step 6 After read replicas are created, you can view and manage them.

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

----End

Follow-up Operations

Managing a Read Replica

2.7.4 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click ⊞ in front of the DB instance and click the target read replica to go to the **Basic Information** page.

----End

Entering the Management Interface Through DAS

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and click ⊞ in front of it. In the expanded panel, locate the read replica you want to manage and click **Log In** in the **Operation** column.

Step 4 On the displayed page, enter the username and password and click **Log In**.

----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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 Click the name of the primary DB instance with which the target read replica is associated to go to the **Basic Information** page.

Step 4 In the DB instance topology, click the name of 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click  in front of a DB instance, locate the read replica to be deleted, and choose **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, select **I have read this warning and understand the risks.** and click **Yes**.

----End

2.8 Data Backups

2.8.1 Backup Solutions

RDS supports automated and manual backups. You can periodically back up databases. If a database is faulty or data is damaged, you can restore the database using backups to ensure data reliability.

RDS uses **sysbench** to import data models and a certain amount of data. After data is backed up, the compression ratio is about 80%. The more duplicate data there is, the higher the compression ratio is.

Compression ratio = Space occupied by backup files/Space occupied by data files x 100%

Backup Types

- Full backup: A full backup is to back up all data, even if no data has changed since the last backup.

  Full backups can be triggered automatically (by configuring an intra-region backup policy) or manually.
- Incremental backup (binlog backup): RDS automatically backs up data modifications made after the most recent full or incremental backup every five minutes.

How RDS Backs Up Data

- Single instance

  A single-node architecture, which is more cost-effective than mainstream primary/standby DB instances. After a backup is triggered, data is backed up from the primary instance and stored as a package on OBS. The backup does not take up storage space of the instance.

- Primary/standby instance

  An HA architecture. In a primary/standby pair, each instance has the same instance class. After a backup is triggered, data is backed up from the standby instance and stored as a package on OBS. The backup does not take up storage space of the instance.

  If a database or table in the primary instance is maliciously or mistakenly deleted, the database or table in the standby instance will also be deleted. In this case, you can only use backups to restore the deleted data.

Backup Solutions

Table 2-16 lists RDS backup solutions.

Table 2-16 Backup solutions

Task	Backup Type	Description
Backing up data	Automated backups	RDS automatically creates full backups for your instance during a backup window you specified and saves the backups based on the configured retention period. If necessary, you can restore data to any point in time within the backup retention period. Once the automated backup policy is enabled, a full physical backup is triggered immediately. After that, full backups will be created according to the specified time window and backup cycle. Incremental backups are automatically created every 5 minutes to ensure data reliability.
	Manual backups	Manual backups are user-initiated full backups of instances. The backup method is physical backup. Manual backups will not be deleted until you delete them manually.
	Incremental backups	Incremental backups are binary log (binlog) backups. Binary logging is enabled for RDS for MariaDB instances by default. You do not need to set an interval for incremental backups because RDS automatically backs up incremental data every 5 minutes. Incremental backups can be used to restore data to a specific point in time.
Downloading backups	Downloading a Full Backup File	You can use OBS Browser+, the browser, or the download URL to download a full backup.

Task	Backup Type	Description
	Downloading incremental backups	You can download a single binlog file or merged binlog file. To download a merged binlog file, use any of the following methods: OBS Browser+, the browser, or the download URL.

Billing

Backups are saved as packages in OBS buckets. Backups occupy backup space in OBS. If the free space RDS provides is used up, the additional space required will be billed. For the billing details, see **How Is RDS Backup Data Charged?**

Deleting Backups

- Manual backups and automated backups can be deleted in different ways:
  - Manual backups can only be manually deleted.
  - Automated backups cannot be manually deleted. You can adjust their retention period by referring to **Configuring an Intra-Region Backup Policy**, and backups that expire will be automatically deleted.
- Local binlogs

  If the retention period is set to **0**, the binlogs of your DB instance will be deleted once they are synchronized to the standby instance and read replicas and successfully backed up to OBS.

  If the retention period is set to a value greater than 0, for example, 1 day, the binlogs will be retained for one day after they are synchronized to the standby instance and read replicas from the primary instance and successfully backed up to OBS. After the retention period expires, the binlogs will be automatically deleted.

2.8.2 Configuring an Intra-Region Backup Policy

If a DB instance fails or its data is damaged, you can restore it from backups to ensure data reliability. You can customize an intra-region backup policy as required and then RDS backs up data based on the backup policy you configured. This section describes how to configure an intra-region backup policy.

Notes

- The backups generated using the intra-region backup policy are full backups. Binlog backups are incremental backups automatically generated by RDS every 5 minutes.
- RDS backs up data at the DB instance level, rather than the database level.
- Backups are saved as packages in OBS buckets to ensure data confidentiality and durability.

- When you create an RDS DB instance, intra-region backup is enabled by default. For security purposes, this function cannot be disabled after the instance is created.

Precautions

- Since backing up data affects database read and write performance, the backup time window should be set to off-peak hours.
- Intra-region backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your **intra-region backup policy**. Retained backup files will be automatically deleted at the end of the retention period.

Constraints

- Rebooting the instance is not allowed during full backup. Exercise caution when selecting a backup time window.
- 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.

Billing

Backups are saved as packages in OBS buckets.

Viewing or Modifying an Intra-Region Backup Policy

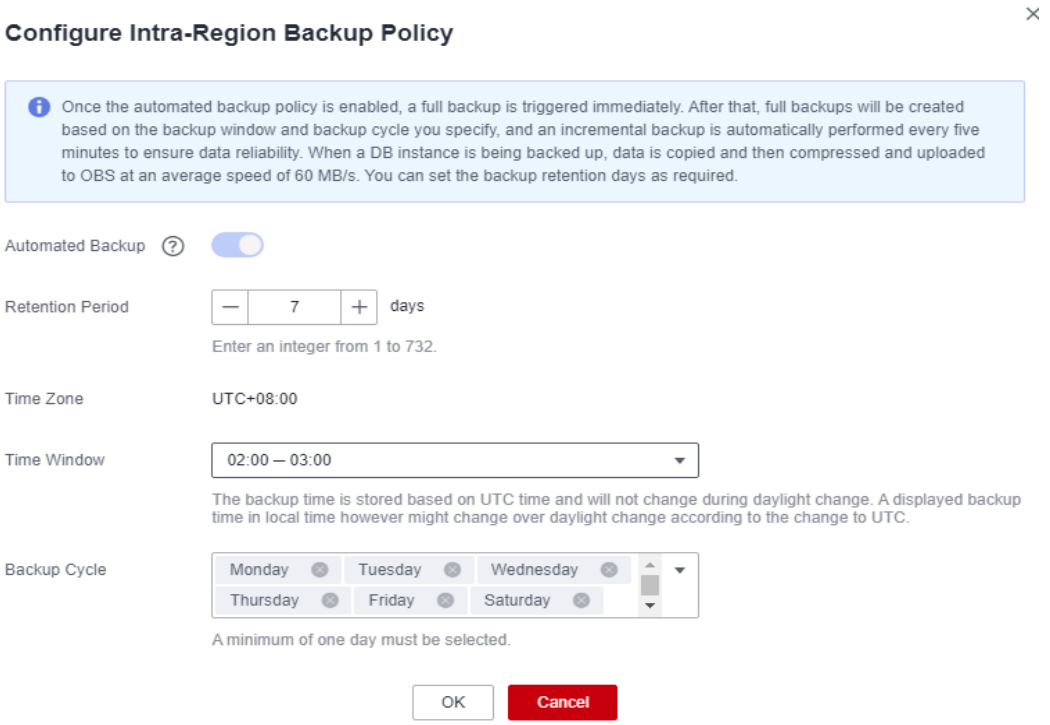
Step 1 **Log in to the management console**.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Intra-Region Backup Policies**.

Figure 2-18 Modifying a backup policy



Step 5 View the configured backup policy. To modify the backup policy, adjust the values of the following parameters:

Table 2-17 Parameter description

Parameter	Description
Retention Period	How many days your automated full backups and binlog backups can be retained. The retention period is from 1 to 732 days and the default value is **7**. • Extending the retention period improves data reliability. • Reducing the retention period takes effect for all backups. Any backups that have expired will be automatically deleted.
Time Window	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.
Backup Cycle	By default, each day of the week is selected. You can change the backup cycle and must select at least one day of the week.

Step 6 Click **OK**.

----End

2.8.3 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups for a running 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.

Billing

Backups are saved as packages in OBS buckets.

Method 1

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

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

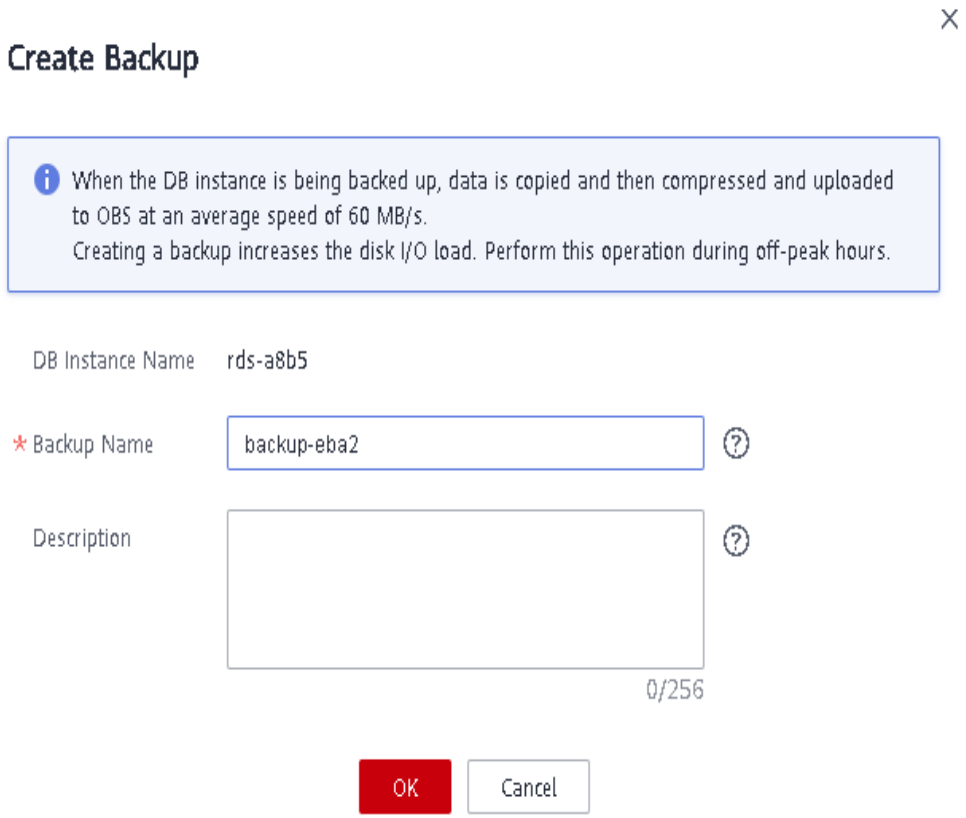
Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Create Backup** in the **Operation** column.

Step 4 In the displayed dialog box, enter a backup name and description.

- 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.

To check whether the backup has been created, you can click  in the upper right corner of the page to check the DB instance status. If the DB instance status becomes **Available** from **Backing up**, the backup has been created. You can manage the backup following the instructions provided in **Step 6**.

Figure 2-19 Creating a backup



Step 5 Click **OK**.

Step 6 After a manual backup has been created, view and manage it on the **Backups** page.

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

----End

Method 2

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 On the **Backups & Restorations** page, click **Create Backup**. In the displayed dialog box, enter a backup name and description.

- 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 5 Click **OK**.

Step 6 After a manual backup has been created, view and manage it on the **Backups** page.

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

----End

2.8.4 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

Procedure

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

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane, choose **Backups**. On the displayed page, select the backups you want to export and click **Export** to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.
- The backup information is exported to an Excel file for your further analysis.

Step 4 View the exported backup information.

----End

2.8.5 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 MariaDB allows you to download full backup files in .qp format.

Constraints

- Full backup files of frozen DB instances cannot be downloaded.
- When you use OBS Browser+ to download backup data, there is no charge for the outbound traffic from OBS.
- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.

Method 1: Using OBS Browser+

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the 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 backup to be downloaded and click **Download** in the **Operation** column.

Step 4 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 5 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and click **OK**.

Figure 2-20 Using OBS Browser+

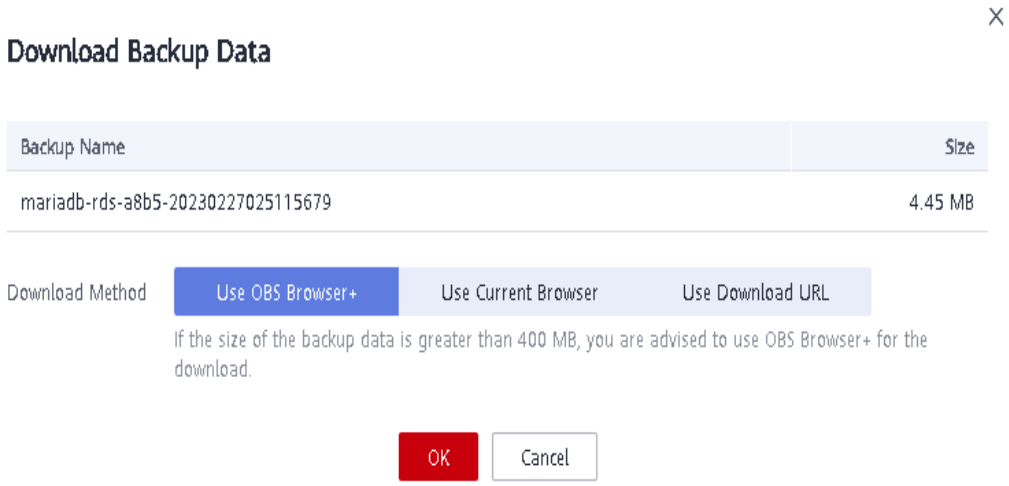
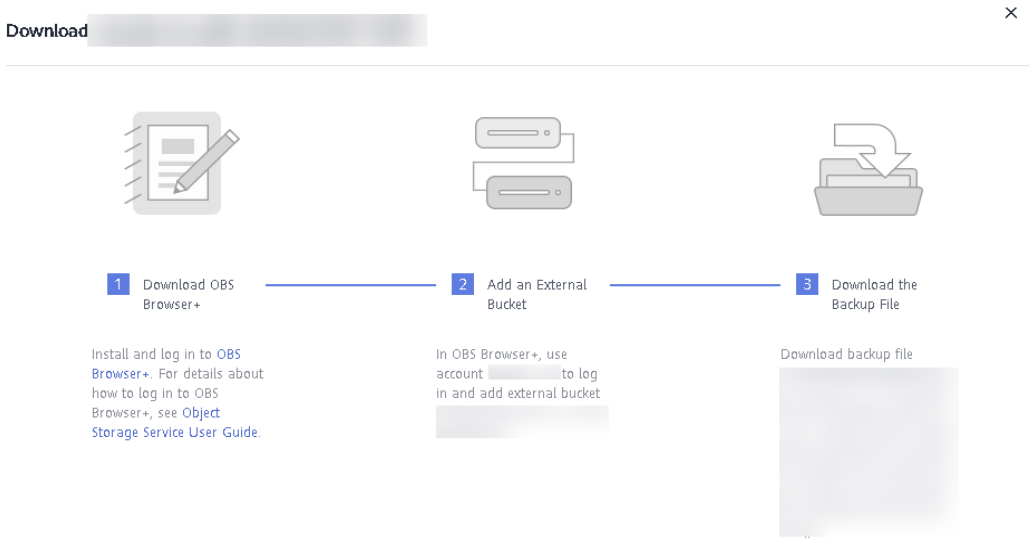
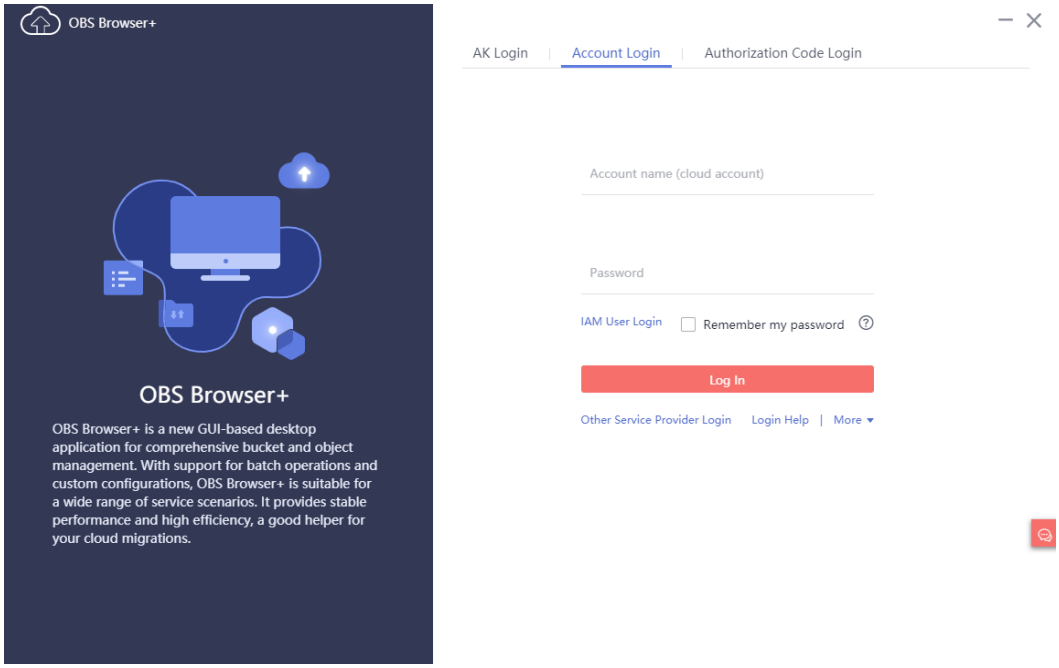


Figure 2-21 Download guide



1. Download OBS Browser+ following step 1 provided on the download guide page.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

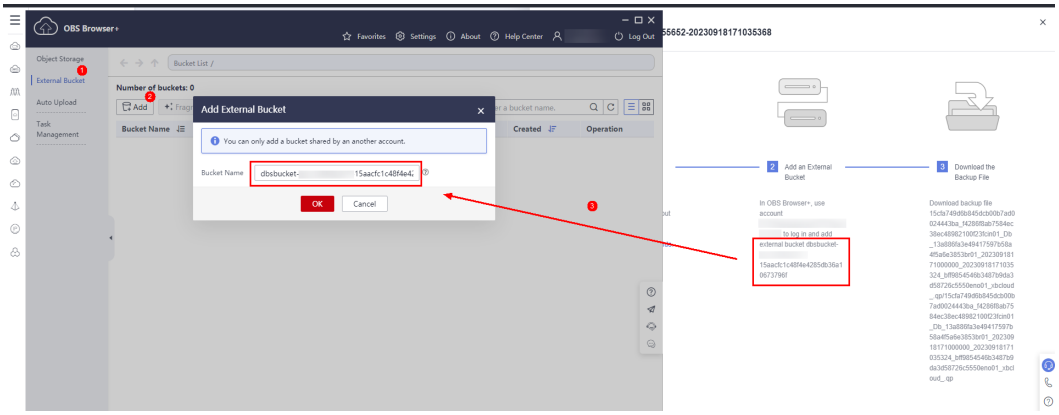
Figure 2-22 Logging in to OBS Browser+



For details about how to log in to OBS Browser+, see **Logging In to OBS Browser+** in the *Object Storage Service Tools Guide*.

4. Add an external bucket using the bucket name provided in step 2 on the download guide page.

Figure 2-23 Adding an external bucket



In the **Add Bucket** dialog box of OBS Browser+, select **Add external bucket** and enter the bucket name provided in step 2 "Add an External Bucket" on the RDS console.

NOTE

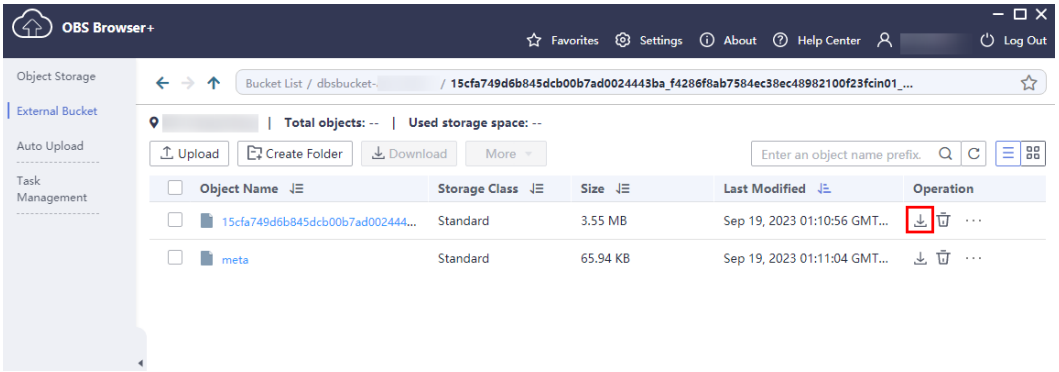
If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of OBS Browser+, enter the backup file name provided in step 3 "Download the Backup File" on the RDS console. In the search result, locate the target backup and download it.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name provided in step 3 on the download guide page. In the search result, locate the target backup and click  in the **Operation** column.

Figure 2-24 Downloading a backup



----End

Method 2: Using Current Browser

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

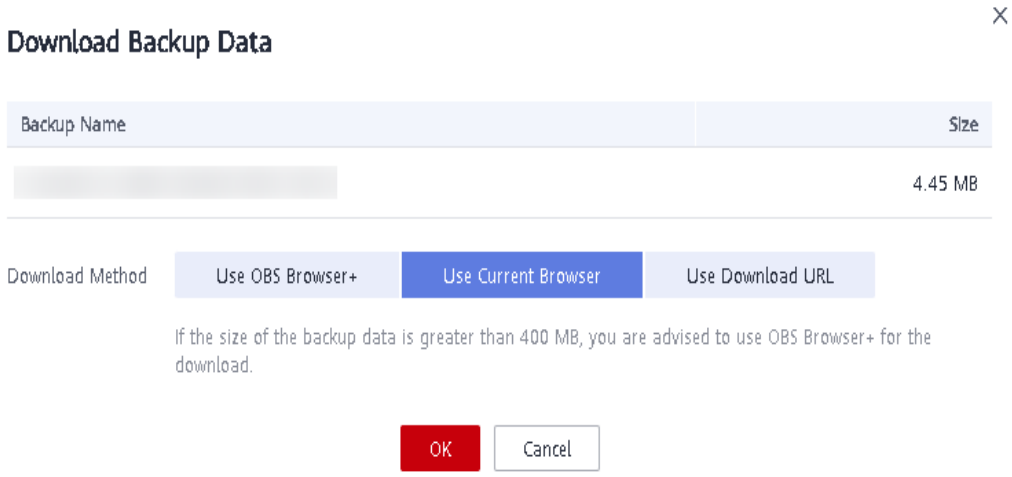
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 5 In the displayed dialog box, select **Use Current Browser** for **Download Method**.

Figure 2-25 Using the current browser



Step 6 Click **OK**.

----End

Method 3: Using Download URL

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

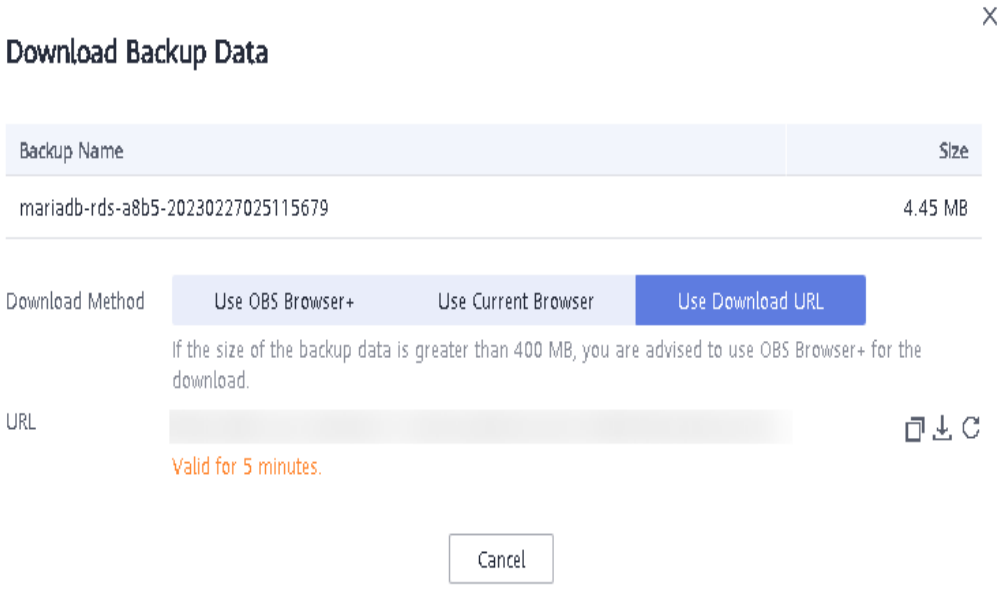
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 4 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 5 In the displayed dialog box, select **Use Download URL** for **Download Method**, click to copy the URL, and enter the URL in your browser.

Figure 2-26 Using the download URL



Step 6 A valid URL for downloading the backup data is displayed. Download the backup file in either of the following ways:

- Using other download tools, such as your browser or Thunder, to download the backup file
- Running the **wget** command to download the backup file

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

  Table 2-18 Parameter description

Parameter	Description
FILE_NAME	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.

Parameter	Description
DOWNLOAD_URL	The location of the backup file to be downloaded. If the location contains special characters, escape is required.

----End

2.8.6 Downloading a Binlog Backup File

Scenarios

RDS for MariaDB allows you to download binlog backup files to your client computer and use them to restore DB instances if necessary.

NOTE

The completion time displayed in the binlog backup file list indicates the time when the last transaction was committed.

Binlog backups on the management console are named in the format of "binlog name +timestamp" and use the row-based logging.

Binlog backup files of frozen DB instances cannot be downloaded.

Downloading a Binlog Backup File

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance. The **Basic Information** page is displayed.

Step 4 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 5 After the download is complete, you can view the binlog backups on your computer.

----End

2.8.7 Setting a Local Retention Period for RDS for MariaDB Binlogs

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

> **NOTE**
>
> Binary logging is enabled for RDS by default and uses row-based logging.
>
> Read replicas do not provide binlogs.

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

If the retention period is set to **0**, the binlogs of your DB instance will be deleted once they are synchronized to the standby instance and read replicas and successfully backed up to OBS. If the retention period is set to a value greater than 0, for example, 1 day, the binlogs will be retained for one day after they are synchronized to the standby instance and read replicas from the primary instance and successfully backed up to OBS. After the retention period expires, the binlogs will be automatically deleted. For details about how to view binlogs, see **Downloading a Binlog Backup File**.

Precautions

The binlog retention period is displayed in hour on the console. However, the value of **expire_logs_days** is displayed in day when you query the binlog retention period by running a command, which cannot be used as a reference. To check how long the binlogs can be retained, view the binlog retention period on the console.

Procedure

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

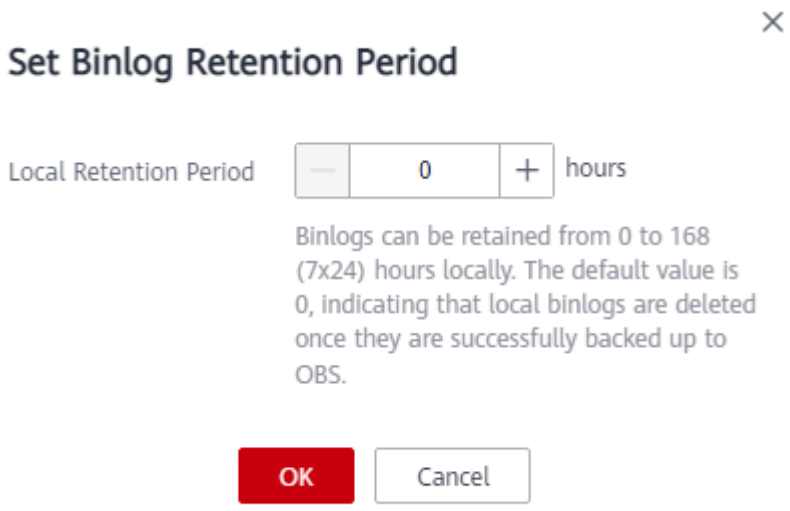
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the **Binlog Backups** page, click **Set Binlog Retention Period**.

Step 5 In the displayed dialog box, set the local retention period and click **OK**.

Figure 2-27 Setting the binlog retention period



NOTE

When binlogs are deleted depends on the local retention period you configure on the console.

----End

2.8.8 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 a DB instance is deleted, the automated backups created for it are also deleted.
- If an **automated backup policy** is enabled, the automated backups will be deleted after the backup retention period expires.
- 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.
- Replicating a backup does not interrupt your services.

Procedure

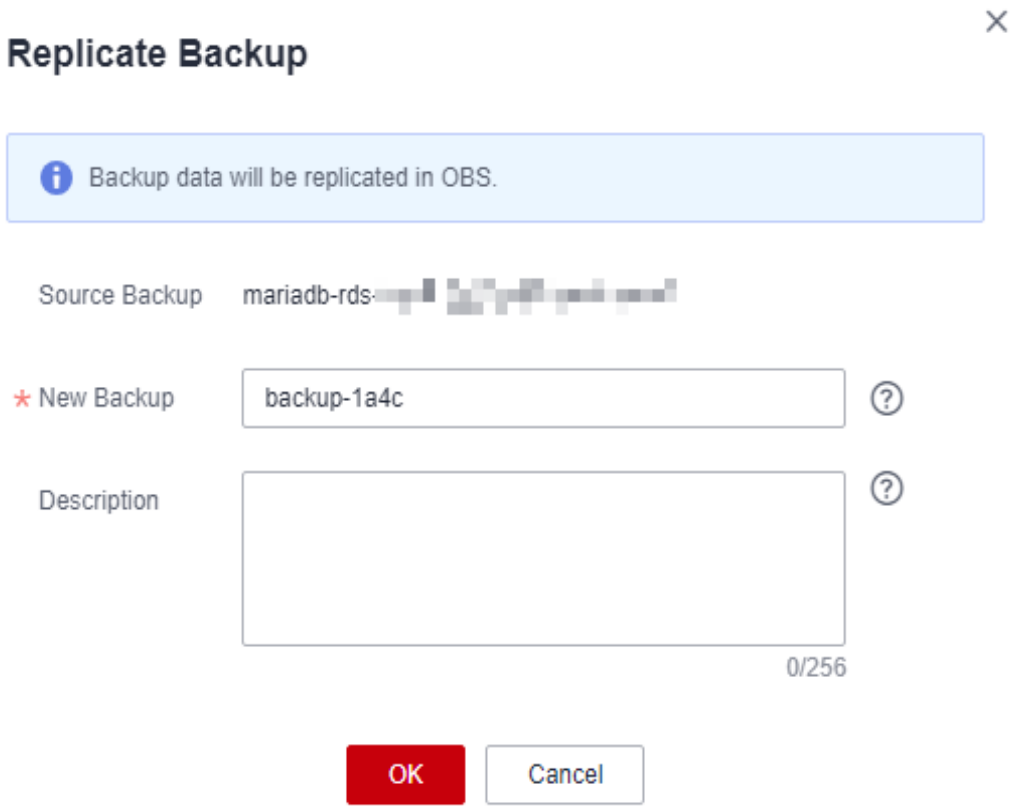
Step 1 **Log in to the management console**.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name. On the **Backups & Restorations** page, locate the backup to be replicated and click **Replicate** in the **Operation** column.

Alternatively, choose **Backups** in the navigation pane on the left. On the displayed page, locate the backup to be replicated and click **Replicate** in the **Operation** column.

Figure 2-28 Replicating a backup



Step 4 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 carriage return characters or the following special characters: >!<"&'=

Step 5 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

2.8.9 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

Step 4 In the displayed dialog box, click **Yes**.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

2.9 Data Restorations

2.9.1 Restoration Solutions

If your database is damaged or mistakenly deleted, you can restore it from backups.

Table 2-19 Restoring a DB instance

When you...	Following Steps In...
Restore data to an RDS for MariaDB instance	Restoring a DB Instance from a Backup
	Restoring a DB Instance to a Point in Time

2.9.2 Restoring a DB Instance from a Backup

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.

Constraints

- You can restore to new DB instances from backups only when your account balance is greater than or equal to $0 USD. You will pay for the new instance specifications.
- If transparent page compression is enabled by specifying attributes in the CREATE TABLE statement for the original DB instance, the restoration may fail due to insufficient storage space.

Procedure

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

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

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

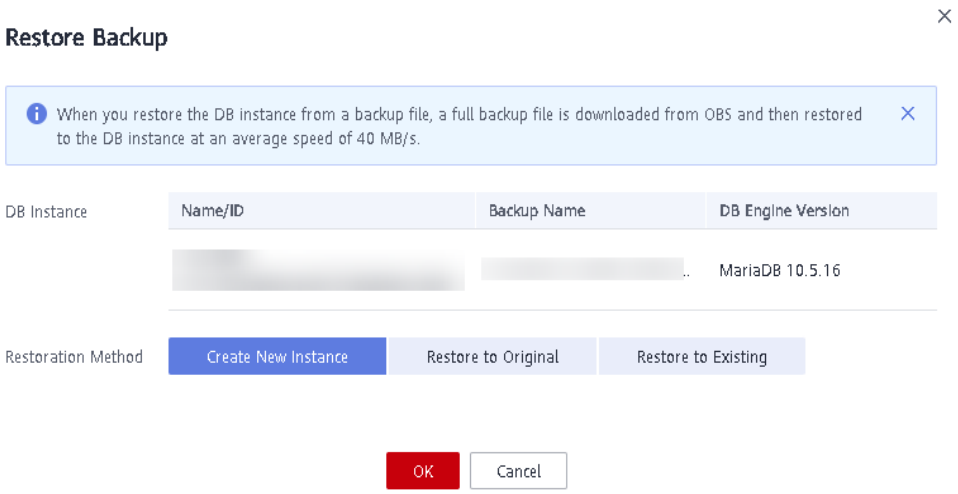
Step 4 Select a restoration method.

- Create New Instance

  Click **OK**. 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 **Buying a DB Instance**.

Figure 2-29 Restoring to a new DB instance



- Restore to Original
  a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.
  b. Confirm the information and click **OK**.

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 all existing data and the DB instance will be unavailable during the restoration process.

- Restore to Existing
  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored." and click **Next**.
  b. 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 use the same DB engine and the same or a later version than the original DB instance.
- Ensure that the storage space of the selected existing DB instance is greater than or equal to the storage space of the original DB instance. Otherwise, data will not be restored.

Step 5 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

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

NOTE

Restoring from backups does not affect the performance of original DB instances.

- 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 Original

  On the **Instances** page, the status of the original DB instance changes from **Restoring** to **Available**. If the original DB instance contains read replicas, the read replica status is the same as the original DB instance status.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** 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.

You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Viewing a Task**.

----End

FAQs

How Can I Restore Data If No Backup Is Available?

2.9.3 Restoring a DB Instance to a Point in Time

Scenarios

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

You can restore one or multiple DB instances at a 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.

Constraints

- You can restore to new DB instances from backups only when your account balance is greater than or equal to $0 USD. You will be billed for new instance specifications.
- Do not run the **reset master** command on RDS for MariaDB instances within their lifecycle. Otherwise, an exception may occur when restoring an RDS for MariaDB instance to a specified point in time.
- When you restore data to a new DB instance, large transactions in the original DB instance backup may cause a restoration failure. If the restoration fails, submit a service ticket.

Restoring a DB Instance

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Restore to Point in Time**.

Step 5 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 **Buying a DB Instance**.

- Restore to Original

  a. Select "I acknowledge that after I select Restore to Original, data on the original databases will be overwritten and the original DB instance will be unavailable during the restoration." and click **Next**.

  b. Confirm the information and click **OK**.

NOTICE

Restoring to the original DB instance will overwrite all existing data and the DB instance will be unavailable during the restoration process.

- Restore to Existing

  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored." 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 use the same DB engine and the same or a 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 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

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.

- Restore to Original

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  A new restoration time range is available. There will be a difference between the new and original time ranges. This difference reflects the duration of the restoration.

  After the restoration is complete, a full backup will be automatically triggered.

- Restore to Existing

  On the **Instances** page, the status of the DB instance changes from **Restoring** to **Available**.

  You can view the detailed progress and result of the task on the **Task Center** page. For details, see **Viewing a Task**.

  After the restoration is complete, a full backup will be automatically triggered.

----End

FAQs

How Can I Restore Data If No Backup Is Available?

2.10 Parameter Templates

2.10.1 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 section describes how to create a parameter template.

Scenarios

If you create a DB instance without specifying a custom DB parameter template, a default parameter template is used. 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.

Precautions

- 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 **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 **Replicating a Parameter Template**.
- 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

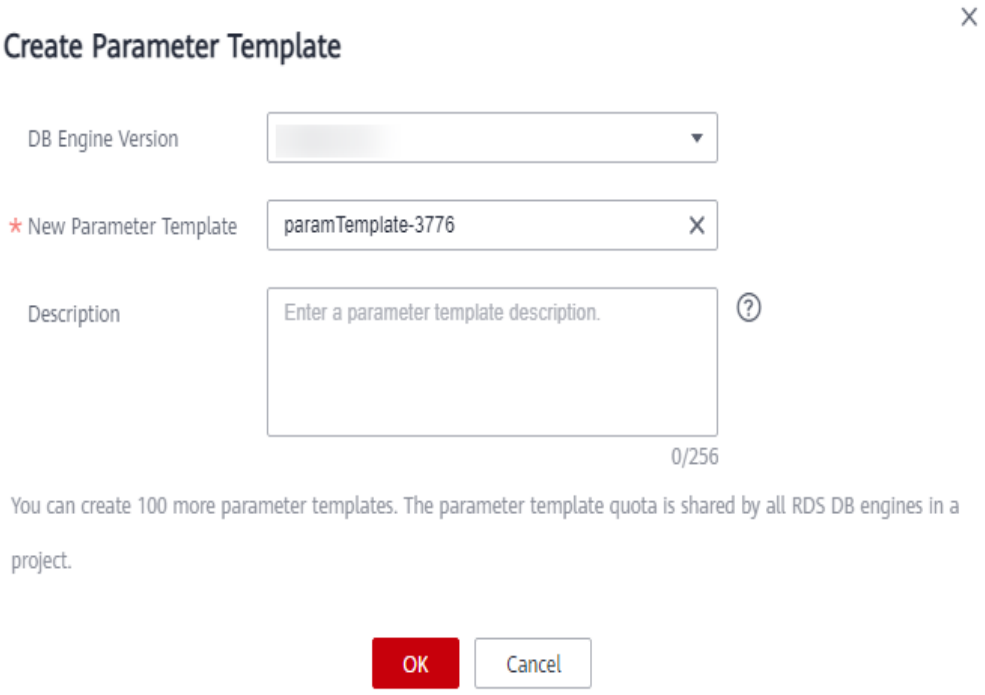
Step 1 **Log in to the management console**.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

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

Figure 2-30 Creating a parameter template



Step 5 In the displayed dialog box, configure required information.

- Select **MariaDB 10.5** for **DB Engine Version**.
- 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: >!<"&'=

Step 6 Click **OK** to create a parameter template.

----End

2.10.2 Modifying RDS for MariaDB Instance Parameters

You can modify parameters in a custom parameter template to optimize RDS database performance. This section describes how to modify parameters of an RDS for MariaDB instance.

Precautions

- You can only change parameter values in custom parameter templates. You cannot change the parameter values in default parameter templates.
- Pay attention to the following points when configuring parameters in a parameter template:
  - When you change a parameter value in a parameter template that has been applied to a DB instance and save the change, the change takes effect only to the DB instance and does not affect other DB instances.
  - When you modify dynamic parameters on the **Parameters** page of a DB instance and save the modifications, the modifications take effect immediately regardless of the **Effective upon Reboot** setting. However, when you modify static parameters on the **Parameters** page of a DB instance and save the modifications, the modifications do not take effect until you manually reboot the DB instance.
  - Modifying parameter template parameters: When you modify parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you have applied the template to your DB instances. When you modify static parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you have applied the template to your DB instances and manually rebooted those DB instances. For details, see **Applying a Parameter Template**.
  - 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.
- 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 a Custom Parameter Template and Applying It to DB Instances

Step 1 Log in to the management console.

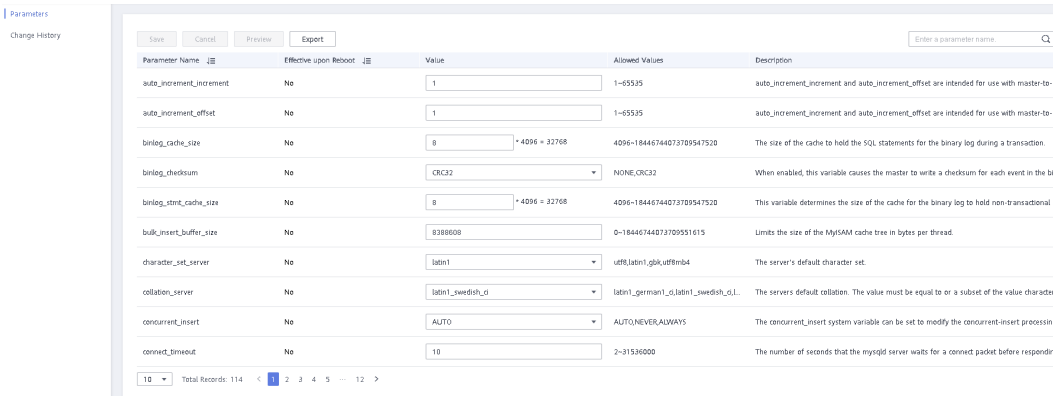
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 Choose **Parameter Templates** 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.

Figure 2-31 Modifying parameters in a parameter template



- 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 **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 Parameters of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the navigation pane, 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 **Instances** 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

2.10.3 Exporting a Parameter Template

To view and use parameters of a DB instance, you can export the parameter template. This section describes how to export 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 analysis.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 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**.

  - 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 Templates** page.

- Exporting to a file

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

  The file name must start with a letter and consist of 4 to 81 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

----End

2.10.4 Importing a Parameter Template

RDS allows you to import new parameter templates for future use. To apply an imported parameter template to new DB instances, see **Applying a Parameter Template**.

Constraints

- Only parameter templates that were exported from the **Parameter Templates** page on the RDS console can be imported.
- If any modification to an exported parameter template causes a change in the file format, the template may not be able to be imported.
- The parameter template to be imported cannot contain parameters related to specifications.

Procedure

Step 1 Log in to the management console.

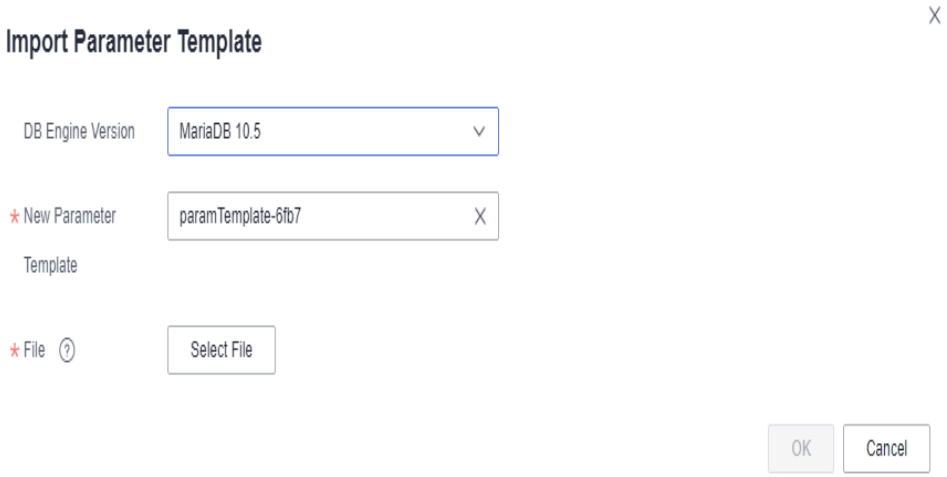
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Import Parameter Template**.

Step 4 In the displayed dialog box, click **Select File**, import the target parameter list (containing parameter names, values, and description), and click **OK**.

Only one file (CSV format) can be imported at a time. The file size cannot exceed 50 KB.

Figure 2-32 Importing a parameter template



----End

2.10.5 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

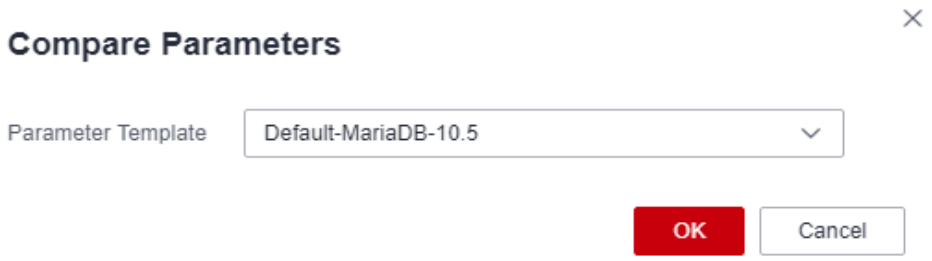
Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Compare** above the parameter list.

Figure 2-33 Comparing instance parameters with those in a specified parameter template



Step 5 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**.

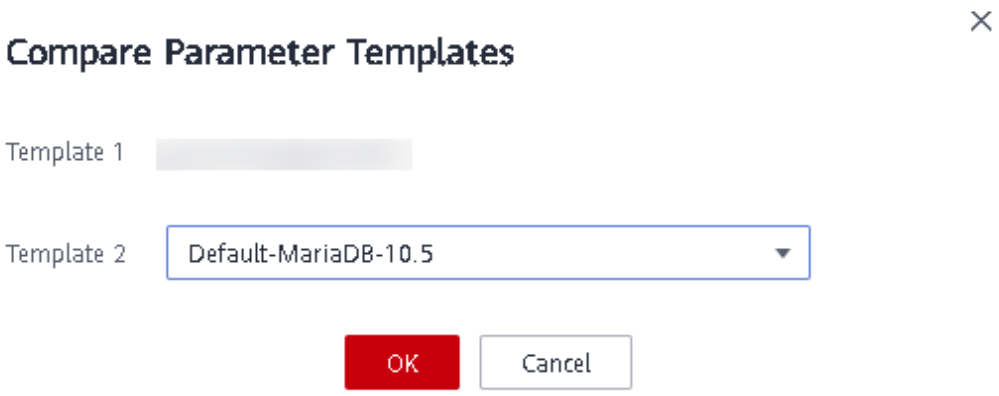
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** 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**.

Figure 2-34 Selecting a parameter template to be compared



- 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

2.10.6 Viewing Parameter Change History

Scenarios

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

NOTE

The change history for an exported or custom parameter template is initially blank.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**. The parameter change history of the last seven days is displayed.

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.

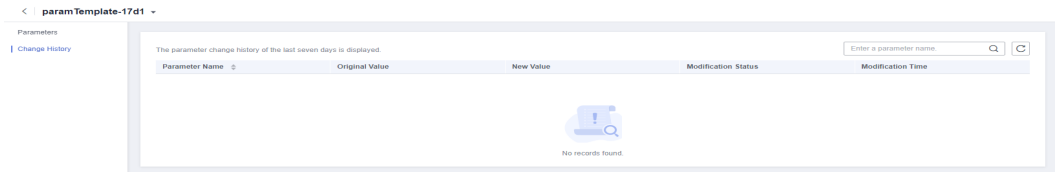
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 Choose **Parameter Templates** 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. The parameter change history of the last seven days is displayed.

Figure 2-35 Viewing parameter change history



You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

Viewing Parameter Changes

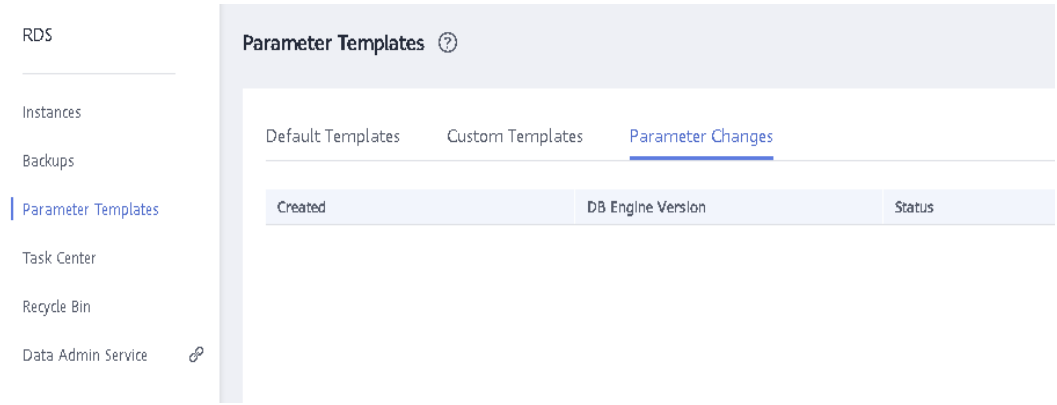
Step 1 **Log in to the management console**.

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click the **Parameter Changes** tab.

Figure 2-36 Viewing parameter changes



Step 5 Click **View Details** in the **Operation** column.

You can view detailed information about the modified parameters.

----End

2.10.7 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. You can also export the parameter template to generate a new parameter template for future use.

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 of the page and choose **Databases** > **Relational Database Service**.

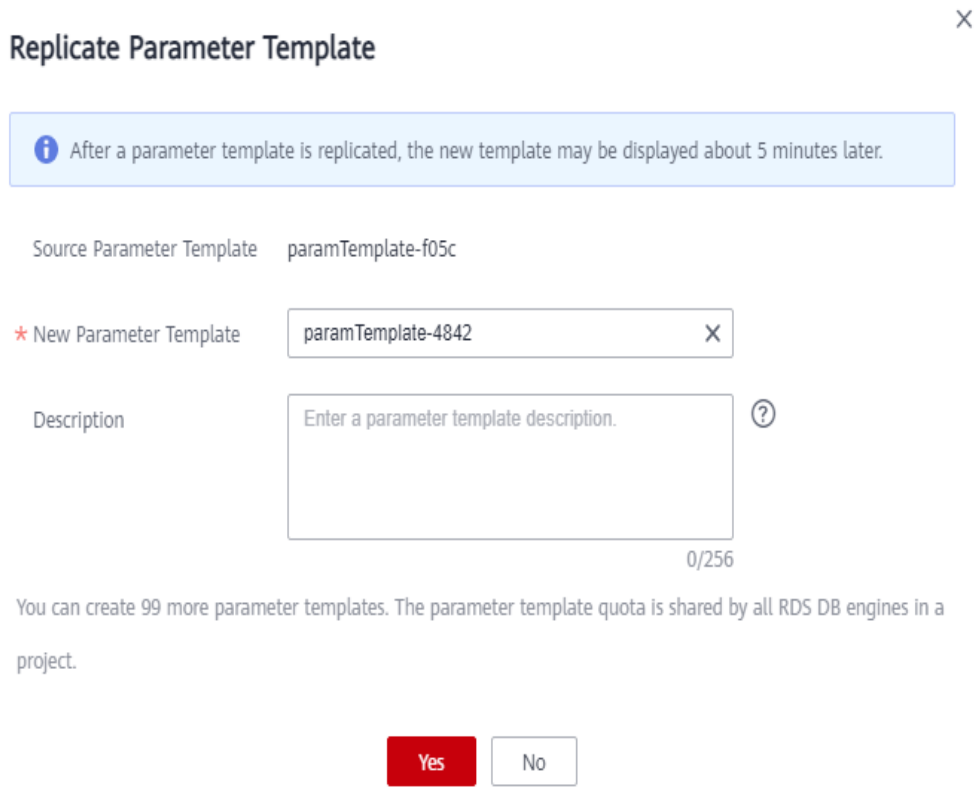
Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

NOTE

To ensure that your parameter templates are applicable to all types of DB instances and databases can be started normally, the values of **innodb_flush_log_at_trx_commit** and **sync_binlog** exported from primary DB instances or read replicas are **1** by default.

Figure 2-37 Replicating a parameter template



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

- 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 Templates** page.

----End

2.10.8 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

Figure 2-38 Confirming the reset

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

2.10.9 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 is applied, 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Parameter Templates** 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 **Viewing Application Records of a Parameter Template**.

----End

2.10.10 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**.

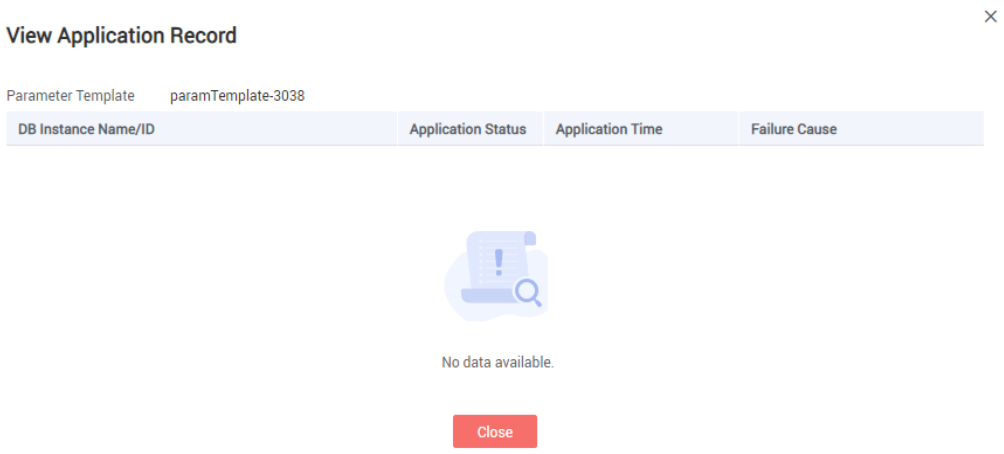
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

Step 4 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and choose **More** > **View Application Record** in the **Operation** column.

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

Figure 2-39 Viewing application records of a parameter template



----End

2.10.11 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

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

Step 5 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or 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

2.10.12 Deleting a Parameter Template

Scenarios

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

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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Parameter Templates**.

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

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

----End

2.11 Connection Management

2.11.1 Viewing and Changing a Floating IP Address

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

Constraints

Changing a floating IP address will interrupt the database connection. You are advised to change a floating IP address during off-peak hours.

Procedure

When you buy a DB instance, select a VPC and subnet on the **Buy DB Instance** page. Then, a floating IP address will be automatically assigned to your instance.

After the instance is created, you can change its floating IP address.

Step 1 Log in to the management console.

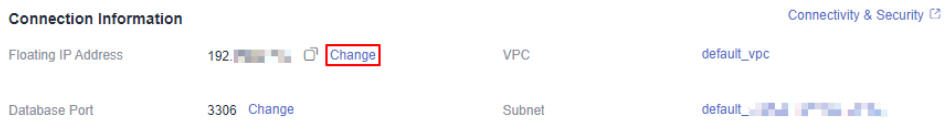
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Floating IP Address** field.

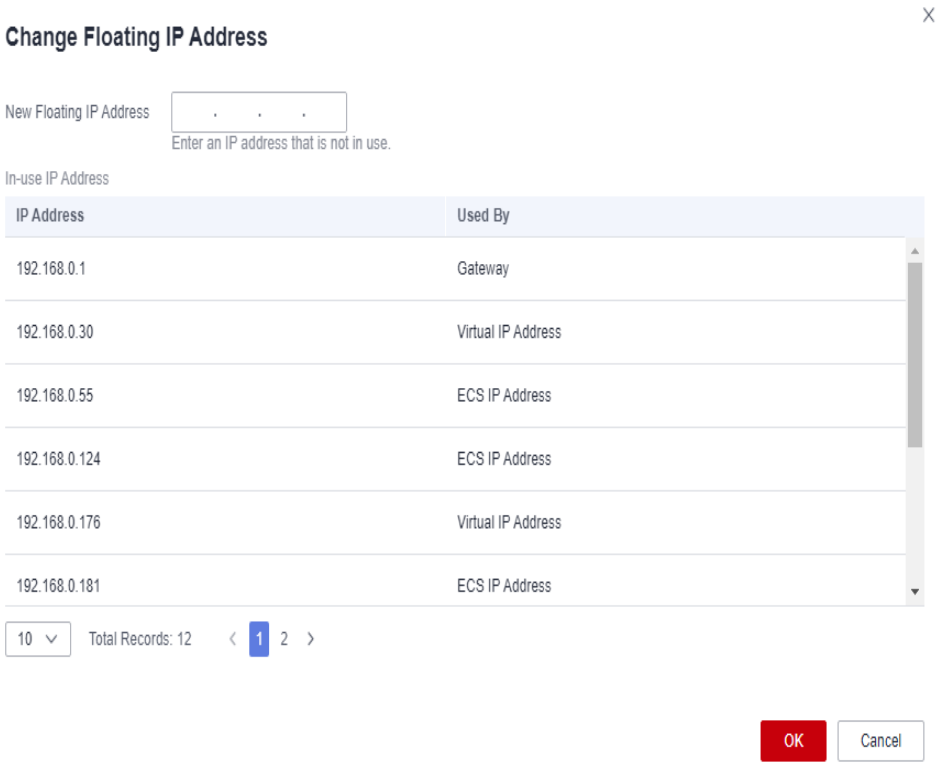
Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Change** next to the **Floating IP Address** field.

Figure 2-40 Floating IP address



Step 5 In the displayed dialog box, check the number of in-use IP addresses. If the in-use IP addresses are less than 254, there are unused floating IP addresses.

Figure 2-41 Changing a floating IP address



Step 6 Enter an available IP address and click **OK**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

2.11.2 Binding and Unbinding an EIP

Scenarios

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.

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, add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see **Configuring Security Group Rules**.
- You can buy an EIP on the **network console** and bind it to a DB instance. One EIP can be bound to only one DB instance. For pricing details, see **Elastic IP pricing details**.

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 of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Bind** above the connection topology.

Step 5 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **Yes**.

Step 6 On the **Connectivity & Security** 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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance that has an EIP bound.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** above the connection topology. In the displayed dialog box, click **Yes**.

Step 5 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 On the **Connectivity & Security** page, view the results.

You can also view the progress or the results 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

2.11.3 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.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

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 of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target DB instance or click  first and then click the target read replica.

Step 4 In the **Connection Information** area on the **Basic Information** page, click **Change** next to the **Database Port** field.

NOTE

RDS for MariaDB instances can use database ports 1024 to 65535, excluding 12017 and 33071, which are reserved for RDS system use.

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

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

If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

- 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.
- 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.
- This process takes 1 to 5 minutes.

Step 6 View the results on the **Basic Information** or **Connectivity & Security** page.

----End

2.11.4 Downloading a Certificate

RDS for MariaDB allows you to download a certificate.

Procedure

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 On the displayed page, click ⬇ next to the **SSL** field in the **DB Information** area to download the root certificate and certificate bundle.

Alternatively, choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, click ⬇ next to the **SSL** field to download the root certificate and certificate bundle.

----End

2.11.5 Configuring a Security Group Rule

Before you can connect to your DB instance, you need to create security group rules to enable specific IP addresses and ports to access your RDS DB instance. This section describes how to configure an inbound rule for a DB instance.

Context

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.

Scenarios

When you attempt to connect to an RDS DB instance through a private network, check whether the ECS and 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. Connect to the DB instance by referring to **Connecting to a DB Instance Using a MariaDB Client**.
- If they are in different security groups, 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 group 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.

For details about the requirements of security group rules, see **Adding a Security Group Rule** in the *Virtual Private Cloud User Guide*.

Constraints

The default security group rule allows all outgoing data packets. ECSs and RDS DB instances can access each other if they are 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 100 security groups in your cloud account.
- By default, you can add up to 50 security group rules to a security group.
- One RDS instance can be associated only with one security group, but one security group can be associated with multiple RDS instances.
- Too many security group rules will increase the first packet latency. You are advised to create no more than 50 rules for a 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

To ensure the security of your data and DB instances, you are advised to use the principle of least privilege for database access. Change the database port (default value: **3306**), and set the IP address to the remote server's address or any IP address in the remote server's smallest subnet to control the access from the remote server.

The default value of **Source** is **0.0.0.0/0**, indicating that 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Security Group Rules** area, view security group rules.

Step 6 Click **Add Inbound Rule** or **Allow All IP** to configure security group rules.

To add more inbound rules, click .

NOTE

Allow All IP allows all IP addresses to access RDS DB instances in the security group, which poses high security risks. Exercise caution when performing this operation.

Figure 2-42 Adding an inbound rule

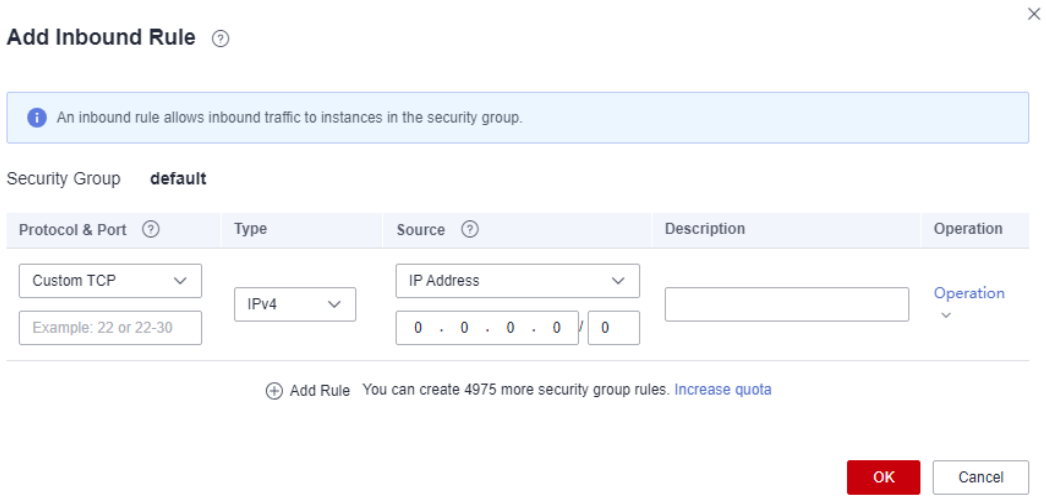


Table 2-20 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	**Protocol**: network protocol. Available options: **All ports**, **Custom TCP**, **Custom UDP**, **ICMP**, or **GRE**.	Custom TCP
	Port: the port over which the traffic can reach your DB instance. RDS for MariaDB instances can use database ports 1024 to 65535, excluding 12017 and 33071, which are reserved for RDS system use.	3306
Type	Supported source IP address type. Its value can be: • IPv4 • IPv6	IPv4
Source	The source in an inbound rule is used to match the IP address or address range of an external request. The source can be: • Single IP address: 192.168.10.10/32 (IPv4 address) • IP address segment: 192.168.1.0/24 (IPv4 address segment) • All IP addresses: 0.0.0.0/0 (any IPv4 address) • Security group: sg-abc • IP address group: ipGroup-test	0.0.0.0/0
Description	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

Step 7 Click **OK**.

----End

2.12 Database Management

2.12.1 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.
- Database names must be unique.
- After a database is created, the database name cannot be changed.

Creating a Database Through RDS

Step 1 Log in to the management console.

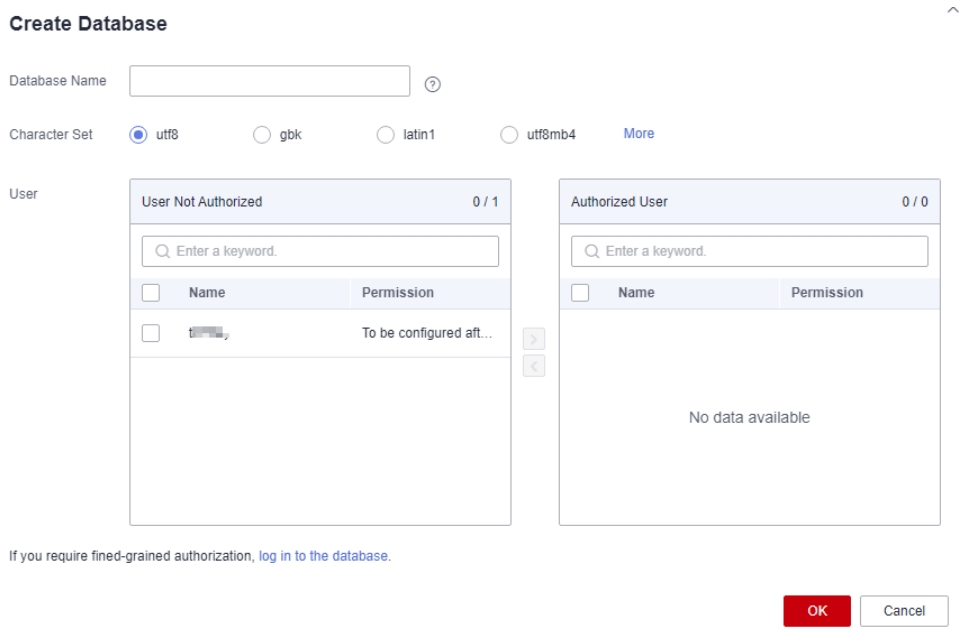
Step 2 Click in the upper left corner and select a region and a project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Databases** page, click **Create Database**. In the displayed dialog box, enter a database name, select a character set, and authorize permissions for users. Then, click **OK**.

Figure 2-43 Creating a database



- The database name can contain 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed. The total number of hyphens (-) cannot exceed 10.
- The default character set is **utf8**. You can click **More** to view more character sets.
- Select unauthorized users and click  to authorize permissions 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 6 After the database is created, manage it on the **Databases** page.

----End

Creating a Database Through DAS

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the instance you want to log in and click **Log In** in the **Operation** column.

Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Run the following command to create a database:

create database *database_name*;

Step 8 Run the following command to view the database:

show databases;

----End

2.12.2 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 no users are available, you can create one by referring to **Creating a Database Account**.

Step 7 In the displayed dialog box, click **OK**.

----End

2.12.3 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **OK**.

Step 6 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

2.12.4 Enabling or Disabling Event Scheduler

Scenarios

Event scheduler manages the scheduling and execution of events. The 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 MariaDB resolves this issue. With RDS for MariaDB, even if there is a failover or switchover, the events will still be properly scheduled. You can simply enable or disable the event scheduler on the RDS console.

Notes

- By default, the event scheduler is disabled after a DB instance is created.
- After a primary/standby failover or switchover is performed, the event scheduler 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 scheduler status is the same as that of the original DB instance.
- After a single DB instance is changed to a primary/standby DB instance, the event scheduler status is the same as that of the primary DB instance.

Constraints

Read replicas do not support this function.

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click  next to the **Event Scheduler** field.

NOTICE

After the event scheduler is enabled, reactivate the previously created events to ensure that the event statuses on the primary and standby instances are the same.

----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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the **DB Information** area on the displayed **Basic Information** page, click  next to the **Event Scheduler** field.

----End

FAQs

What Should I Do If I Do Not Have the Permission to Change the event_scheduler Settings?

Answer: You can only enable or disable the event scheduler on the console. For details, see this section.

2.13 Account Management (Non-Administrator)

2.13.1 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.

You can create a database account using RDS or DAS:

- RDS: RDS is easy to use. There are no special commands to remember.
- DAS: DAS is a powerful platform that offers more flexibility, but you need to be familiar with the creation commands. The process requires a bit more expertise.

Account Type

Table 2-21 Account description

Account Type	Description
Administrator account **root**	Only the administrator account **root** is provided on the instance creation page. For details about the supported permissions, see **RDS for MariaDB Constraints**. **NOTE** Running **revoke**, **drop user**, or **rename user** on **root** may cause service interruption. Exercise caution when running any of these statements.
System accounts	To provide O&M services, the system automatically creates system accounts when you create RDS for MariaDB DB instances. These system accounts are unavailable to you. • **mariadb.sys**: used to create views. • **rdsAdmin**: a management account with the highest permission. It is used to query and modify instance information, rectify faults, migrate data, and restore data. • **rdsRepl**: a replication account, used to synchronize data from the primary instance to the standby instance or read replicas. • **rdsBackup**: a backup account, used for backend backup. • **rdsMetric**: a metric monitoring account used by watchdog to collect database status data. • **dsc_readonly**: used to anonymize data.
Other accounts	Accounts created through the console, APIs, or SQL statements After an account is created, you can assign permissions to it as required. For details, see **Changing Permissions for a Database Account**.

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**.

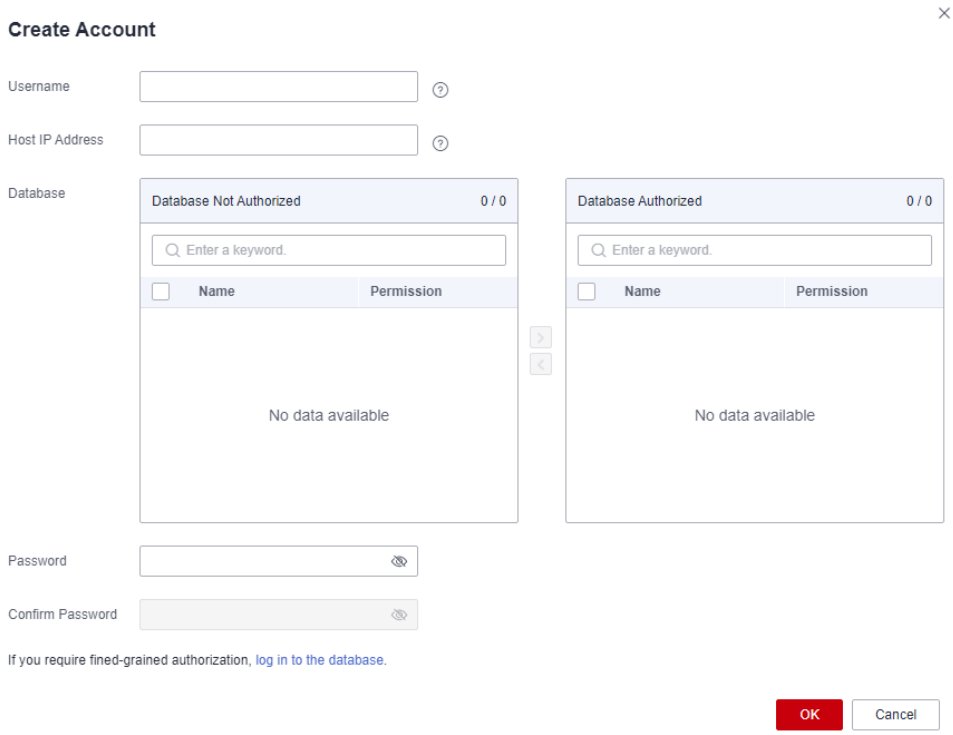
Step 2 Click in the upper left corner and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Accounts** page, click **Create Account**. In the displayed dialog box, specify **Username** and **Host IP Address**, authorize permissions for databases, enter a password, and confirm the password. Then, click **OK**.

Figure 2-44 Creating a database account



- The username consists of 1 to 32 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- 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 **Creating a Database**. You can also modify the permissions after the account creation by referring to **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 (~!@#$%^*-_=+?,()&).
- You can specify IP addresses that are allowed to access your DB instance.
  - To enable all IP addresses to access your instance, enter **%** for **Host IP Address**.
  - To enable all IP addresses in the subnet 10.10.10.*X* to access your instance, enter **10.10.10.%** for **Host IP Address**.

- To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.1,172.16.213.9** (no spaces before or after the comma).

Step 6 After the account is created, you can manage it on the **Accounts** page of the DB instance.

----End

Creating a Database Account Through DAS

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the instance you want to log in and click **Log In** in the **Operation** column.

Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 Create an account.

- On the top menu bar, choose **Account Management** > **User Management**. On the displayed page, click **Create User**. Then, configure basic information, advanced settings, global permissions, and object permissions, and click **Save**. In the displayed dialog box, click **OK**.

  For details about how to set permissions, see **Creating a User**.
- You can also choose **SQL Operations** > **SQL Query** from the top menu bar and run the following command to create an account:

  create user *username*;

----End

2.13.2 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.

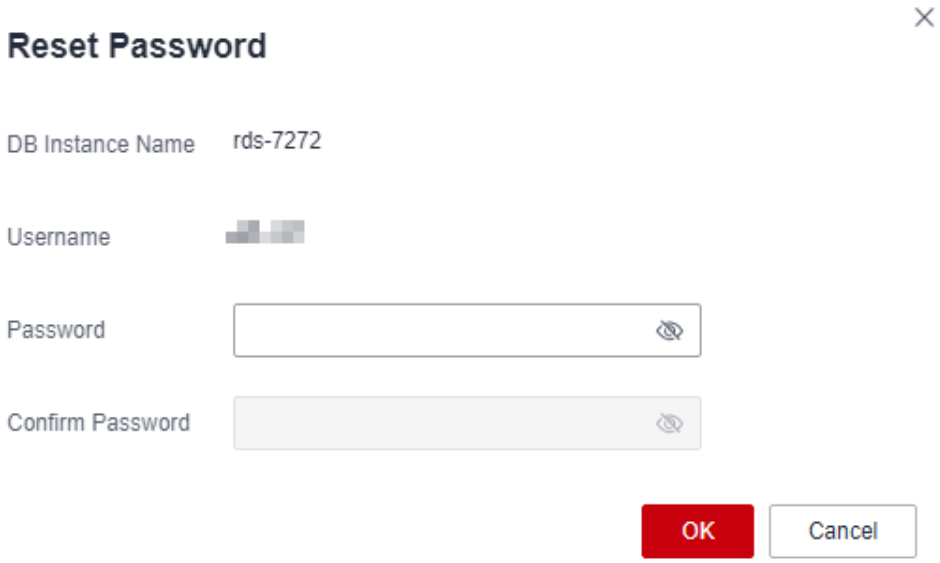
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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**.

Figure 2-45 Resetting a password



- The password must consist of 8 to 32 characters and contain all types of the following characters: uppercase letters, lowercase letters, digits, and special characters (~!@#$%^*-_=+?,()&).
- The password must be different from the username or username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.
- After the password is reset, the database will not be rebooted and permissions will not be changed.
- You can use Cloud Trace Service (CTS) to query the password reset records. For details, see **Viewing Traces**.

Step 7 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

2.13.3 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.

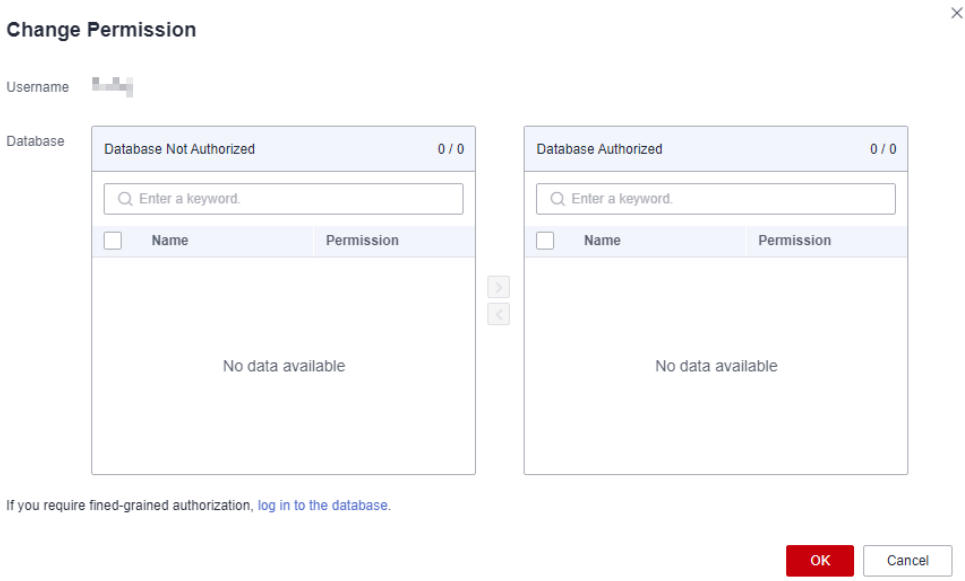
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **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.

Figure 2-46 Changing permissions



If there are no unauthorized databases, you can create one by referring to **Creating a Database**.

Step 7 Click **OK**.

----End

2.13.4 Modifying Host IP Addresses for a Database Account

Scenarios

You can modify the host IP addresses that are allowed to access your instance for database accounts you created.

Constraints

This operation cannot be performed for DB instances that are 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.

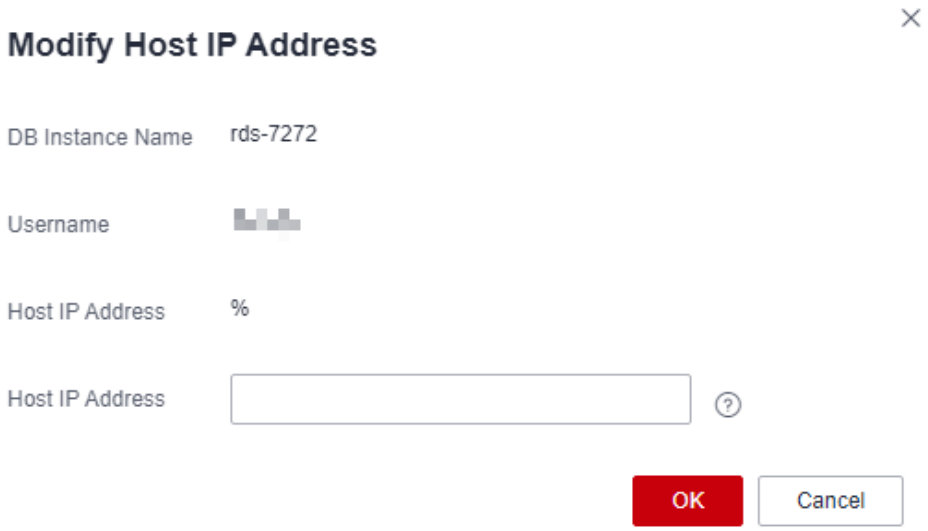
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Accounts**. Locate the target account and choose **More** > **Modify Host IP Address** in the **Operation** column.

Step 6 In the displayed dialog box, enter the new IP addresses.

Figure 2-47 Modifying host IP addresses



- To enable all IP addresses to access your instance, enter **%** for **Host IP Address**.
- To enable all IP addresses in the subnet 10.10.10.*X* to access your instance, enter **10.10.10.%** for **Host IP Address**.
- To specify multiple IP addresses, separate them with commas (,), for example, **192.168.0.1,172.16.213.9** (no spaces before or after the comma).

Step 7 Click **OK**.

Step 8 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

2.13.5 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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **OK**.

Step 7 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

2.14 Account and Network Security

2.14.1 Database Account Security

Setting the Account Password Complexity

For information about the database password strength requirements on the RDS console, see the database configuration table in **Buy a DB Instance**.

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 password.

Account Description

To provide O&M services, the system automatically creates system accounts when you create RDS for MariaDB instances. These system accounts are unavailable to you.

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: a management account, used to query and modify instance information, rectify faults, migrate data, and restore data.
- rdsRepl: the replication account, which is used to synchronize data from primary DB instances to standby DB instances or read replicas.
- rdsBackup: the backup account, which is used for backend backup.
- rdsMetric: the metric monitoring account, which is used by watchdog to collect database status data.

Setting Password Complexity

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance to navigate to the **Basic Information** page.

Passwords must:

- Consist of at least eight characters.
- Contain at least one uppercase letter, one lowercase letter, one digit, and one special character.
- Must be different from the username.

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

RDS for MariaDB allows you to modify the following parameters:

- **simple_password_check_minimal_length**: Set this parameter to **8**.
- **simple_password_check_letters_same_case**: Set this parameter to **1**.
- **simple_password_check_digits**: Set this parameter to **1**.
- **simple_password_check_other_characters**: Set this parameter to **1**.

NOTICE

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

- If the value is **Yes** and the DB instance status on the **Instances** 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.

Step 6 Perform the following operations as needed:

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

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

----End

2.14.2 Resetting the Administrator Password to Restore root Access

Scenarios

You can reset the administrator password of a primary instance.

If you forget the password of the administrator account **root**, you can reset the password.

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.
- 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 of the page and choose **Databases** > **Relational Database Service**.

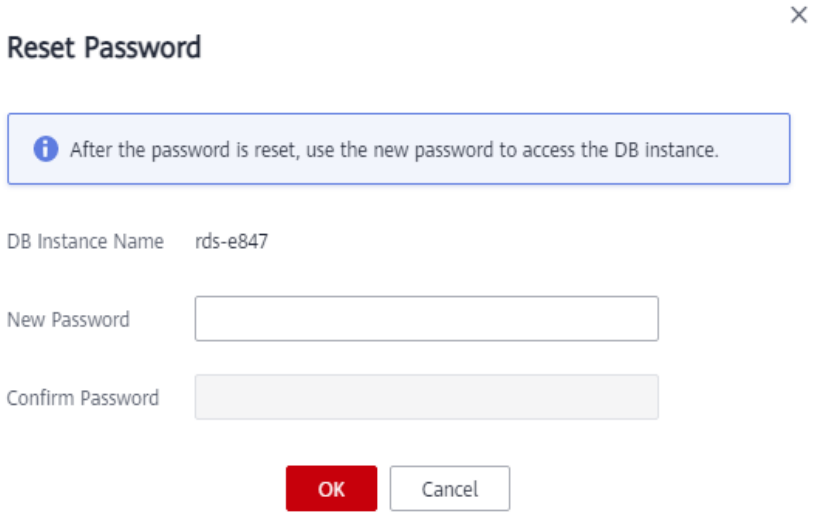
Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 4 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 5 Enter and confirm the new password.

Figure 2-48 Resetting the administrator 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.

Step 6 Click **OK**.

----End

Method 2

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

Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

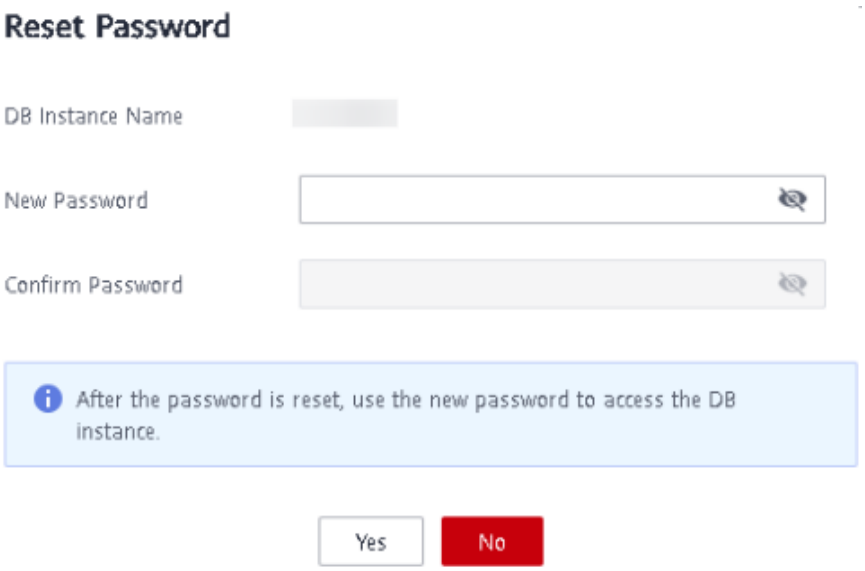
Step 4 In the **DB Information** area on the **Basic Information** page, click **Reset Password** next to the **Administrator** field.

Step 5 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 Enter and confirm the new password.

Figure 2-49 Resetting the administrator 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.

Step 7 Click **OK**.

----End

2.14.3 Configuring an SSL Connection

The Secure Socket Layer (SSL) connection encrypts data and is more secure. This section describes how to enable and disable SSL.

Context

SSL is an encryption-based Internet security protocol for establishing an encrypted link between a server and a client. It provides authenticated Internet connections to ensure the privacy and integrity of online communications. SSL:

- Authenticates users and servers, ensuring that data is sent to the correct clients and servers.
- Encrypts data, preventing it from being intercepted during transmission.
- Ensures data integrity during transmission.

Notes

By default, SSL is disabled for new RDS for MariaDB instances. If your client has no SSL compatibility issues, you can enable SSL by referring to **Enabling SSL**. Enabling SSL will increase the network connection response time and CPU resource consumption. Before enabling it, evaluate any potential impacts on service performance.

You can connect to a DB instance through a non-SSL connection or an SSL connection.

- If SSL is enabled, your connection will be more secure.
- If SSL is disabled, you can connect to a database using a non-SSL connection.

Precautions

Enabling or disabling SSL will cause DB instances to reboot and interrupt connections. Exercise caution when performing this operation.

Enabling SSL

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

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Step 5 In the displayed dialog box, click **OK**. Wait for some seconds and check that SSL has been enabled on the **Basic Information** page.

----End

Disabling SSL

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

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 4 In the **DB Information** area on the **Basic Information** page, click  next to the **SSL** field.

Step 5 In the displayed dialog box, click **OK**. Wait for some seconds and check that SSL has been disabled on the **Basic Information** page.

----End

2.14.4 Configuring a Password Expiration Policy

Using the same password too long makes it easier for hackers to crack or guess your password. Requiring password changes after a certain amount of time can improve security. This section describes how to configure a password expiration policy.

Precautions

- Once your password expires, you cannot log in to the database.
- After the password expiration policy is configured, you need to periodically check whether your password is about to expire.

Modifying the Database Parameter

Step 1 Log in to the management console.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane, choose **Parameters**.

Step 5 On the displayed page, change the value of **default_password_lifetime**.

The value of this parameter indicates how many days until a password expires. The default value is **0**, indicating that the created user password will never expire.

Step 6 Click **Save**. In the displayed dialog box, click **Yes**.

----End

Configuring the Password Expiration Policy Through DAS

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. On the displayed **Basic Information** page, click **Log In** in the upper right corner of the page.

Step 5 Enter the username and password and click **Log In**.

Step 6 Choose **SQL Operations** > **SQL Query**.

Step 7 In the editing area, compile the statement shown below. The unit of **password_life_time** is day. You are advised to set it to **180**.

ALTER USER *username* PASSWORD EXPIRE INTERVAL *password_life_time* DAY;

Step 8 Click **Execute SQL**. Then, view SQL execution status on the **Executed SQL Statements**, **Messages**, and **Result** tab pages.

----End

2.14.5 Unbinding an EIP

Scenarios

The Elastic IP (EIP) service enables your RDS instances to communicate with the Internet using static public IP addresses and scalable bandwidths. But this increases the risk of network-wide attacks on your instances. Using an EIP leaves you open to DoS or DDoS attacks.

As an internal component, the database can be accessed using an internal IP address. Therefore, you are advised to unbind the EIP from the database.

Prerequisites

An EIP has been bound to your DB instance. For details, see **Binding an EIP**.

Unbinding an EIP

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

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance that has an EIP bound.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Alternatively, in the **Connection Topology** area, click **Public Connection** and then **Unbind** above the connection topology. In the displayed dialog box, click **Yes**.

Step 5 (**Optional**) If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 On the **Connectivity & Security** page, view the results.

You can also view the progress or the results 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

2.14.6 Using DBSS (Recommended)

Database Security Service (DBSS) is an intelligent database security service. Based on the machine learning mechanism and big data analytics technologies, it can audit your databases, detect SQL injection attacks, and identify high-risk operations.

You are advised to use DBSS to provide extended data security capabilities. For details, see **Database Security Service**.

Advantages

- DBSS can help you meet security compliance requirements.
  - DBSS can help you comply with DJCP (graded protection) standards for database audit.
  - DBSS can help you comply with security laws and regulations, and provide compliance reports that meet data security standards (such as Sarbanes-Oxley).
- DBSS can back up and restore database audit logs and meet the audit data retention requirements.
- DBSS can monitor risks, sessions, session distribution, and SQL distribution in real time.
- DBSS can report alarms for risky behavior and attacks and respond to database attacks in real time.
- DBSS can locate internal violations and improper operations and keep data assets secure.

Deployed in bypass pattern, database audit can perform flexible audits on the database without affecting user services.

- Database audit monitors database logins, operation types (data definition, operation, and control), and operation objects based on risky operations to effectively audit the database.
- Database audit analyzes risks and sessions, and detects SQL injection attempts so you can stay apprised of your database status.
- Database audit provides a report template library to generate daily, weekly, or monthly audit reports according to your configurations. It sends real-time alarm notifications to help you obtain audit reports in a timely manner.

2.15 Metrics and Alarms

2.15.1 Configuring Displayed Metrics

You can use APIs provided by Cloud Eye to query the monitoring metrics and alarms generated for RDS. This section describes the RDS metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions.

Notes

- The RDS Agent monitors RDS DB instances and collects monitoring metrics only.
- The monitoring interval is 1 minute.

Namespace

SYS.RDS

DB Instance Monitoring Metrics

The following table lists the performance metrics of RDS for MariaDB instances.

Table 2-22 Performance metrics

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
1	rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0-100 %	RDS for MariaDB instance	1 minute
2	rds002_mem_util	Memory Usage	Memory usage of the monitored object	0-100 %	RDS for MariaDB instance	1 minute
3	rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0 counts/s	RDS for MariaDB instance	1 minute
4	rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
5	rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
6	rds006_conn_count	Total Connections	Total number of connections that attempt to connect to the MariaDB server	≥ 0 counts	RDS for MariaDB instance	1 minute
7	rds007_conn_active_count	Current Active Connections	Number of current active connections	≥ 0 counts	RDS for MariaDB instance	1 minute
8	rds008_qps	QPS	Query times of SQL statements (including stored procedures) per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
9	rds009_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0 transactions/s	RDS for MariaDB instance	1 minute
10	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	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
11	rds011_innodb_buf_hit	Buffer Pool Hit Ratio	Ratio of read hits to read requests in the InnoDB buffer	0–1	RDS for MariaDB instance	1 minute
12	rds012_innodb_buf_dirty	Buffer Pool Dirty Block Ratio	Ratio of dirty data to used pages in the InnoDB buffer	0–1	RDS for MariaDB instance	1 minute
13	rds013_innodb_reads	InnoDB Read Throughput	Number of read bytes per second in the InnoDB buffer	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
14	rds014_innodb_writes	InnoDB Write Throughput	Number of write bytes per second in the InnoDB buffer	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
15	rds015_innodb_read_count	InnoDB File Read Frequency	Number of times that InnoDB reads data from files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
16	rds016_innodb_write_count	InnoDB File Write Frequency	Number of times that InnoDB writes data to files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
17	rds017_innodb_log_write_req_count	InnoDB Log Write Requests per Second	Number of InnoDB log write requests per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
18	rds018_innodb_log_write_count	InnoDB Log Physical Write Frequency	Number of InnoDB physical write times to log files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
19	rds019_innodb_log_fsync_count	InnoDB Log fsync() Write Frequency	Number of completed fsync() write times to InnoDB log files per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
20	rds020_temp_tbl_rate	Temporary Tables Created per Second	Number of temporary tables created on hard disks per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
21	rds021_myisam_buf_usage	Key Buffer Usage	MyISAM key buffer usage	0–1	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
22	rds022_myisam_buf_write_hit	Key Buffer Write Hit Ratio	MyISAM key buffer write hit ratio	0–1	RDS for MariaDB instance	1 minute
23	rds023_myisam_buf_read_hit	Key Buffer Read Hit Ratio	MyISAM key buffer read hit ratio	0–1	RDS for MariaDB instance	1 minute
24	rds024_myisam_disk_write_count	MyISAM Disk Write Frequency	Number of times that indexes are written to disks per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
25	rds025_myisam_disk_read_count	MyISAM Disk Read Frequency	Number of times that indexes are read from disks per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
26	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	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
27	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	RDS for MariaDB instance	1 minute
28	rds028_comdml_del_count	DELETE Statements per Second	Number of DELETE statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
29	rds029_comdml_ins_count	INSERT Statements per Second	Number of INSERT statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
30	rds030_comdml_ins_sel_count	INSERT_SELECT Statements per Second	Number of INSERT_SELECT statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
31	rds031_comdml_rep_count	REPLACE Statements per Second	Number of REPLACE statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
32	rds032_comdml_rep_sel_count	REPLACE_SELECTION Statements per Second	Number of REPLACE_SELECTION statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
33	rds033_comdml_sel_count	SELECT Statements per Second	Number of SELECT statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
34	rds034_comdml_upd_count	UPDATE Statements per Second	Number of UPDATE statements executed per second	≥ 0 queries/s	RDS for MariaDB instance	1 minute
35	rds035_innodb_del_row_count	Row Delete Frequency	Number of rows deleted from the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute
36	rds036_innodb_ins_row_count	Row Insert Frequency	Number of rows inserted into the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute
37	rds037_innodb_read_row_count	Row Read Frequency	Number of rows read from the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
38	rds038_innodb_upd_row_count	Row Update Frequency	Number of rows updated into the InnoDB table per second	≥ 0 rows/s	RDS for MariaDB instance	1 minute
39	rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0-100 %	RDS for MariaDB instance	1 minute
40	rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4,000 GB	RDS for MariaDB instance	1 minute
41	rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4,000 GB	RDS for MariaDB instance	1 minute
42	rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
43	rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
44	rds072_conn_usage	Connection Usage	Percent of used MariaDB connections to the total number of connections	0-100 %	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
45	rds073_replication_delay	Real-Time Replication Delay	Real-time replication delay between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master.	≥ 0s	RDS for MariaDB instance	1 minute 5 seconds
46	rds074_slow_queries	Slow Query Logs	Number of slow query logs generated per minute by MariaDB	≥ 0	RDS for MariaDB instance	1 minute
47	rds075_avg_disk_ms_per_read	Disk Read Time	Average time required for each disk read in a specified period	≥ 0 ms	RDS for MariaDB instance	1 minute
48	rds076_avg_disk_ms_per_write	Disk Write Time	Average time required for each disk write in a specified period	≥ 0 ms	RDS for MariaDB instance	1 minute
49	rds077_vma	VMA	Virtual memory area size of an RDS process	≥ 0 counts	RDS for MariaDB instance	1 minute
50	rds078_threads	Threads	Number of threads in a process	≥ 0 counts	RDS for MariaDB instance	1 minute
51	rds079_vm_hwm	Peak Resident Set Size	Peak physical memory usage of an RDS process	≥ 0 KB	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
52	rds080_vm_peak	Peak Virtual Memory Size	Peak virtual memory usage of an RDS process	≥ 0 KB	RDS for MariaDB instance	1 minute
53	rds082_semi_sync_tx_avg_wait_time	Transaction Wait Time	Average wait time of transactions in semi-synchronous mode	≥ 0 μs	RDS for MariaDB instance	1 minute
54	rds173_replication_delay_avg	Average Replication Delay	Average replication delay within 60s between standby DB instances or read replicas and primary DB instances, corresponding to seconds_behind_master	≥ 0s	RDS for MariaDB instance	1 minute
55	rds_buffer_pool_wait_free	Dirty Pages to Be Flushed to Disks	When InnoDB needs to read or create a page and no clean pages are available, InnoDB flushes some dirty pages first and waits for that operation	≥ 0 counts	RDS for MariaDB instance	1 minute
56	rds_bytes_recv_rate	Received Bytes per Second	Number of bytes received by the database per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
57	rds_bytes_sent_rate	Sent Bytes per Second	Number of bytes sent from the database per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute
58	rds_conn_active_usage	Active Connection Usage	Usage of active connections	0-100 %	RDS for MariaDB instance	1 minute
59	rds_created_tmp_tables_rate	Temporary Tables Created per Second	Number of temporary tables created per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
60	rds_innodb_buffer_pool_pages_flushed_rate	Innodb_buffer_pool Page Flushes per Second	Number of innodb_buffer_pool page flushes per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
61	rds_innodb_buffer_pool_read_requests_rate	Innodb_buffer_pool Read Requests per Second	Number of innodb_buffer_pool read requests per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
62	rds_innodb_buffer_pool_write_requests_rate	Innodb_buffer_pool Write Requests per Second	Number of innodb_buffer_pool write requests per second	≥ 0 counts/s	RDS for MariaDB instance	1 minute
63	rds_innodb_lock_waits	Row Locks Waits Transactions	Number of InnoDB transactions waiting for row lock	≥ 0 counts	RDS for MariaDB instance	1 minute
64	rds_innodb_log_waits_count	Log Buffer Status	Number of times that the log buffer was too small and a wait was required for it to be flushed before continuing	≥ 0 counts	RDS for MariaDB instance	1 minute
65	rds_innodb_log_waits_rate	Flush Times to Disks Due to Insufficient Log Buffer	Times of transaction logs flushed to disks due to insufficient log buffer	≥ 0 counts/s	RDS for MariaDB instance	1 minute
66	rds_innodb_os_log_written_rate	Redo Log Size Written per Second	Size of redo logs written per second	≥ 0 bytes/s	RDS for MariaDB instance	1 minute

No.	Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Interval (Raw Data)
67	rds_innodb_pages_read_rate	Data Volume Read By InnoDB per Second	Data volume read by InnoDB per second	≥ 0 Pages/s	RDS for MariaDB instance	1 minute
68	rds_innodb_pages_written_rate	Data Volume Written by InnoDB per Second	Data volume written by InnoDB per second	≥ 0 Pages/s	RDS for MariaDB instance	1 minute
69	rds_innodb_row_lock_current_waits	Current Row Lock Waits	Number of current InnoDB row lock waits	≥ 0 counts	RDS for MariaDB instance	1 minute
70	rds_innodb_row_lock_time_avg	Row Lock Wait Time	Average wait time of InnoDB row locks	≥ 0 ms	RDS for MariaDB instance	1 minute
71	rds_wait_thread_count	Waiting Threads	Number of waiting threads	≥ 0 counts	RDS for MariaDB instance	1 minute

Dimension

Key	Value
mariadb_cluster_id	RDS for MariaDB instance ID

2.15.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS metrics on the management console. For details, see **Procedure**.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** in the upper right corner of the page.

Step 5 On the displayed page, view the instance monitoring metrics.

- On the Cloud Eye console, click **Select Metric** in the upper right corner. In the displayed dialog box, you can select the metrics to be displayed and sort them by dragging them to desired locations.
- You can sort graphs by dragging them based on service requirements.
- You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, and 7 days.

----End

2.15.3 Setting Alarm Rules

Scenarios

You can set alarm rules to customize the monitored objects and notification policies and keep track of the RDS running status.

RDS alarm rules include alarm rule names, resource types, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Procedure

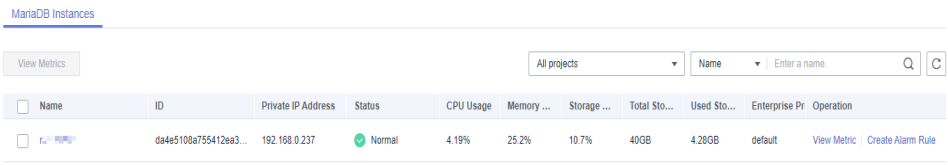
Step 1 Log in to the management console.

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

Step 3 Click **Service List**. Under **Management & Governance**, click **Cloud Eye**.

Step 4 In the navigation pane on the left, choose **Cloud Service Monitoring** > **Relational Database Service**.

Figure 2-50 Choosing a monitored object



Step 5 Locate the instance you want to add an alarm rule for and click **Create Alarm Rule** in the **Operation** column.

Step 6 On the displayed page, set required parameters.

- Basic information

  Table 2-23 Basic information

Parameter	Description
Name	Alarm rule name. The system generates a random name, which you can modify. Example value: alarm-wnat
Description	(Optional) Supplementary information about the alarm rule.

- Alarm parameters

Figure 2-51 Alarm parameters

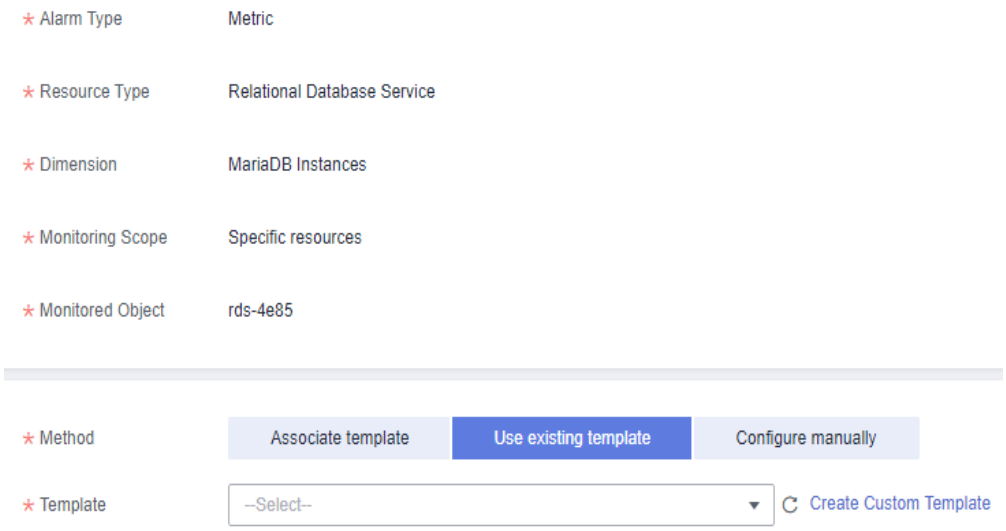


Table 2-24 Alarm parameters

Parameter	Description
Method	You are advised to select **Use existing template**. The existing templates already contain three common alarm metrics: CPU usage, memory usage, and storage space usage. **NOTE** If you select **Associate template**, after the associated template is modified, the policies contained in this alarm rule to be created will be modified accordingly.
Template	Select the template to be used. You can select a default alarm template or create a custom template.
Alarm Policy	If you select **Configure manually** for **Method**, you need to configure alarm policies. An alarm is triggered when the metric configured for this alarm reaches the preset threshold in consecutive periods. For example, Cloud Eye triggers an alarm if the average CPU usage of the monitored object is 80% or more for three consecutive 5-minute periods. **NOTE** A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm is triggered.
Alarm Severity	If you select **Configure manually** for **Method**, you need to configure alarm severity. The alarm severity can be **Critical**, **Major**, **Minor**, or **Informational**.

- Alarm notification parameters

Figure 2-52 Notification parameters

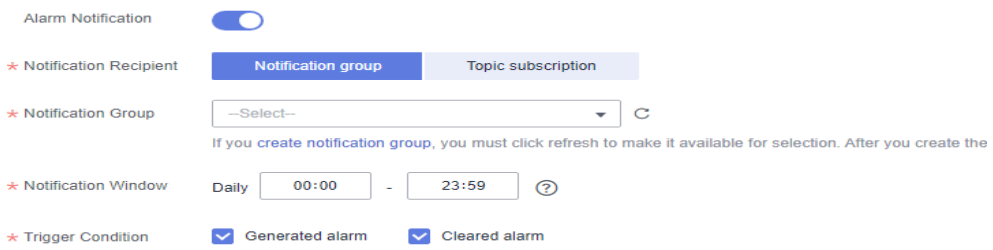


Table 2-25 Alarm notification parameters

Parameter	Description
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Recipient	You can select a notification group or topic subscription as required.
Notification Group	Notification group the alarm notification is to be sent to.
Notification Object	Object the alarm notification is to be sent to. You can select the account contact or a topic. – The account contact is the mobile phone number and email address of the registered account. – A topic is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.

- Advanced settings

Table 2-26 Advanced settings

Parameter	Description
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.

Parameter	Description
Tag	A tag is a key-value pair. Tags identify cloud resources so that you can easily categorize and search for your resources. – A key can contain a maximum of 128 characters, and a value can contain a maximum of 225 characters. – A maximum of 20 tags can be added.

Step 7 After the configuration is complete, click **Create**.

----End

2.15.4 Event Monitoring

2.15.4.1 Introduction to Event Monitoring

Event monitoring provides event data reporting, query, and alarm reporting. You can create alarm rules for both system and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on RDS resources that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, for example, resetting the administrator password or modifying the backup policy.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see **Events Supported by Event Monitoring**.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by services to Cloud Eye.

For details about how to report custom events, see **Reporting Events**.

2.15.4.2 Viewing Event Monitoring Data

Scenarios

This section describes how to view the event monitoring data.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using the following method:

On the **Instances** page, click the DB instance name. On the displayed **Overview** page, click **View Metrics** in the upper right corner.

Step 5 Click < to return to the main page of Cloud Eye.

Step 6 In the navigation pane on the left, choose **Event Monitoring**.

On the displayed **Event Monitoring** page, all system events generated in the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view the events generated in different periods.

Step 7 Click **View Graph**. On the details page, click **View Event** in the **Operation** column of a specific event to view details.

----End

2.15.4.3 Creating an Alarm Rule to Monitor an Event

Scenarios

This section describes how to create an alarm rule to monitor an event.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the **Create Alarm Rule** page, configure the parameters.

Table 2-27 Parameter description

Parameter	Description
Name	Specifies the alarm rule name. The system generates a random name, which you can modify.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create one.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Event Type	Specifies the event type of the metric corresponding to the alarm rule.

Parameter	Description
Event Source	Specifies the service the event is generated for. Select **Relational Database Service** or **Database Proxy Service**.
Monitoring Scope	Specifies the monitoring scope for event monitoring.
Method	Specifies the means you use to create the alarm rule.
Alarm Policy	**Event Name** indicates the instantaneous operations users performed on system resources, such as login and logout. For events supported by event monitoring, see **Events Supported by Event Monitoring**. You can select a trigger mode and alarm severity as needed.

Click  to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 2-28 Alarm notification

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Object	Specifies the object that receives alarm notifications. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification.

Step 6 Click **Create**.

----End

2.15.4.4 Events Supported by Event Monitoring

Table 2-29 Resource exception events

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
RDS	DB instance creation failure	createInstanceFailed	Major	A DB instance fails to create because the number of disks is insufficient, the quota is insufficient, or underlying resources are exhausted.	Check the number of disks and quota size. Release resources and create DB instances again.	DB instances cannot be created.
	Cross-region backup synchronization failure	crossRegionBackupSyncFailed	Minor	Generally, this problem is caused by insufficient underlying network and replication resources.	If this event is continuously reported, submit a service ticket to adjust the underlying resource allocation.	Backups cannot be used for restoration in the destination region.
	Full backup failure	fullBackupFailed	Major	A single full backup failure does not affect the files that have been successfully backed up, but prolongs the incremental backup restoration time during the point-in-time recovery (PITR).	Create a manual backup again.	Backup failed.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Primary/standby switchover failure	activeStandBySwitchFailed	Major	The standby DB instance does not take over workloads from the primary DB instance due to network or server failures. The original primary DB instance continues to provide workloads within a short time.	Check whether the connection between your application and the database is re-established.	None
	Replication status abnormal	abnormalReplicationStatus	Major	The possible causes are as follows: The replication delay between the primary and standby instances is too long, which usually occurs when a large amount of data is written to databases or a large transaction is processed. During peak hours, data may be blocked. The network between the primary and standby instances is disconnected.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Replication status recovered	replicationStatusRecovered	Major	The replication delay between the primary and standby instances is within the normal range, or the network connection between them has restored.	No action is required.	None
	DB instance faulty	faultyDBInstance	Major	A single or primary DB instance was faulty due to a disaster or a server failure.	Check whether an automated backup policy has been configured for the DB instance and submit a service ticket.	The database service may be unavailable.
	DB instance recovered	DBInstanceRecovered	Major	RDS rebuilds the standby DB instance with its high availability. After the instance is rebuilt, this event will be reported.	No action is required.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Failure of changing single DB instance to primary/ standby	singleToHaFailed	Major	A fault occurs when RDS is creating the standby DB instance or configuring replication between the primary and standby DB instances. The fault may occur because resources are insufficient in the data center where the standby DB instance is located.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write of the DB instance.
	Database process restarted	DatabaseProcessRestarted	Major	The database process is stopped due to insufficient memory or high load.	Log in to the Cloud Eye console. Check whether the memory usage increases sharply, the CPU usage is too high for a long time, or the storage space is insufficient. You can increase the CPU and memory specifications or optimize the service logic.	Downtime occurs. In this case, RDS automatically restarts the database process and attempts to recover the workloads.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Instance storage full	instanceDiskFull	Major	Generally, the cause is that the data space usage is too high.	Scale up the instance.	The DB instance becomes read-only because the storage space is full, and data cannot be written to the database.
	Instance storage full recovered	instanceDiskFullRecovered	Major	The instance disk is recovered.	No action is required.	The instance is restored and supports both read and write operations.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	MySQL instance connection limit reached	mysqlConnectionsFull	Major	The maximum number of connections supported has been reached as the workload was increasing.	• Release unnecessary connections. • Reduce the load by controlling concurrency. • Upgrade the instance class to allow more connections.	New connections cannot be established.
	MySQL instance connections full recovered	mysqlConnectionsFullRecovered	Major	The number of instance connections has been reduced to below the maximum number of connections.	Check whether the workload on your instance is running properly.	The number of instance connections is below the maximum allowed.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	New connection errors caused by a MySQL overload	highLoadInstanceConnectionsAbnormal	Major	New connections cannot be set up or are abnormal because resources such as CPUs, the memory, storage, or network bandwidth are insufficient.	• Scale up system resources like CPUs, the memory, and storage. • Adjust MySQL parameters, for example, increasing the connection pool size and adjusting the cache size. • Select the abnormal session you want to end and kill it for the databases to recover.	New connections cannot be set up or are abnormal.
	New connection failure caused by MySQL overload resolved	highLoadInstanceConnectionsAbnormalRevocered	Major	The new connection failure caused by MySQL overload has been resolved.	Check whether the workload on your instance is running properly.	The new connection failure has been resolved.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Kafka connection failed	kafkaConnectionFailed	Major	The network is unstable or the Kafka server does not work properly.	Check your network connection and the Kafka server status.	Audit logs cannot be sent to the Kafka server.
Database proxy	Proxy instance access to DB instance failure	proxy_connection_failure_cause_security_group	Major	No rules in the security group of the DB instance allow the proxy instance to access the DB instance.	Add the proxy instance address to the rules of the security group.	Service requests routed through the proxy instance are interrupted.
	Connection failure between proxy instance and DB instance	proxy_connection_failure_to_db	Major	The proxy instance failed to establish a new connection with the primary DB instance, and it may fail to establish a new connection with a read replica. The DB instance or proxy instance is overloaded, or the network between the them is abnormal.	Change values of related parameters based on metrics (connections, active connections, and CPU usage) of the DB instance and proxy instance. If the metrics are normal, submit a service ticket.	Service requests routed through the proxy instance are interrupted.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Connection failure between database proxy and read replica	proxy_connection_failure_to_replica	Minor	The proxy instance failed to establish a new connection with a read replica. The read replica is overloaded, or the network between the proxy instance and read replica is abnormal.	Change values of related parameters based on metrics (connections, active connections, and CPU usage) of the read replica. If the metrics are normal, submit a service ticket.	Read requests routed through the proxy instance are partially interrupted.

Table 2-30 Operation events

Event Source	Event Name	Event ID	Event Severity	Description
RDS	Reset administrator password	resetPassword	Major	The password of the database administrator is reset.
	Operate DB instance	instanceAction	Major	The storage space is scaled or the instance class is changed.
	Delete DB instance	deleteInstance	Minor	The DB instance is deleted.
	Modify backup policy	setBackupPolicy	Minor	The backup policy is modified.

Event Source	Event Name	Event ID	Event Severity	Description
	Modify parameter group	updateParameterGroup	Minor	The parameter group is modified.
	Delete parameter group	deleteParameterGroup	Minor	The parameter group is deleted.
	Reset parameter group	resetParameterGroup	Minor	The parameter group is reset.
	Change database port	changeInstancePort	Major	The database port is changed.
	Primary/standby switchover or failover	PrimaryStandbySwitched	Major	Only automatic failovers are monitored. Manual primary/standby switchovers are not supported.

2.16 Interconnection with CTS

2.16.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 2-31 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

Operation	Resource Type	Trace Name
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
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

Operation	Resource Type	Trace Name
Downloading a merged backup	backup	packBackupsDown-Load
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	rdsUnsubscribeIn-stance
Freezing a DB instance	all	rdsfreezeInstance

2.16.2 Viewing Traces

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.

NOTE

Before using CTS, you need to enable it. For details, see **Enabling CTS**.

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 In the upper left corner of the page, click and choose **Management & Governance** > **Cloud Trace Service**.

Step 4 Choose **Trace List** in the navigation pane on the left.

Step 5 Filter conditions to query traces. The details are described as follows:

- **Trace Type**, **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 6 Click **Query**.

Step 7 Click ⌄ on the left of the required trace to expand its details.

Step 8 Click **View Trace** in the **Operation** column. On the displayed dialog box, the trace structure details are displayed.

Step 9 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

2.17 Log Management

2.17.1 Viewing and Downloading Error Logs

RDS log management allows you to view database-level logs, including error logs and slow SQL query logs.

Error logs help you analyze problems with databases. You can download error logs for further analysis.

You can view error logs generated within the last month.

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 RDS for MariaDB instances, the following levels of logs are displayed:
  - All log levels
  - ERROR
  - WARNING
  - NOTE

- Currently, a maximum of 2,000 error log records can be displayed.
- You can click in the upper right corner to view logs generated in different time segments.
- Only error logs generated within the last one month can be viewed.
- 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane, choose **Logs**. On the **Error Logs** page, click the **Downloads** tab.

Step 6 Locate a log file whose status is **Preparation completed** and click **Download** in the **Operation** column.

- The system automatically loads the downloading preparation tasks. The time required depends on the log file size and the 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.
- If the size of a log to be downloaded is greater than 40 MB, you need to use OBS Browser+ to download it. For details, see **Method 1: Using OBS Browser +**.
- 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**.

- The downloaded logs contain only the logs of the primary node.

----End

2.17.2 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 to identify statements that are executing slowly and optimize the statements. You can also download slow query logs for service analysis.

Slow query logs generated within the last month can be viewed.

RDS supports the following statement types:

- All statement types
- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE

Parameter Description

Table 2-32 Parameters related to MariaDB slow queries

Parameter	Description
long_query_time	Specifies how many microseconds a SQL query has to take to be defined as a slow query log. The default value is 1s. When the execution time of an SQL statement exceeds the value of this parameter, the SQL statement is recorded in slow query logs. The recommended value is **1s**. Note: The lock wait time is not calculated into the query time.
log_queries_not_using_indexes	Specifies whether to record the slow queries without indexes. The default value is **OFF**.

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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.

NOTE

- You can view the slow query log records of a specified execution statement type or a specific time period.
- Only SELECT statements return the number of result rows. The number of result rows for the INSERT, UPDATE, DELETE, and CREATE statements is 0 by default.
- You can view slow query logs of a specified database name (which cannot contain any special characters). The database name supports only exact search.
- Slow query logs only record executed statements whose execution duration exceeds the threshold.
- 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, RDS starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. 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.
- A maximum of 2,000 slow log records can be displayed. To view more slow log records, submit a service ticket.
- If the length of a single line of an SQL statement exceeds 10 KB or the total number of lines exceeds 200, the SQL statement will be truncated. When you view slow query log details, the SQL statement may be incomplete after special processing and is for reference only.

----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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane, choose **Logs**. On the **Slow Query Logs** page, click the **Downloads** tab.

Step 6 Locate a log file 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.

- Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.
- It is impossible to generate a log file much larger than 40 MB, like 100 MB or 200 MB. If a log file that is a little larger than 40 MB is required, use OBS Browser+ to download it by referring to **Method 1: Using OBS Browser+**.
- 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**.
- The downloaded logs contain only the logs of the primary node.

----End

2.17.3 Enabling or Disabling SQL Audit

After you enable SQL audit, 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.

Notes

- Both DB instances and read replicas support SQL audit logging.
- Audit logs use the Coordinated Universal Time (UTC) format and are not affected by the time zone configuration.
- After SQL audit is enabled, RDS records SQL operations in audit logs. The generated audit log files are temporarily stored in the instance and then uploaded to OBS and stored in the backup space. If there is not enough free backup space available for generated audit logs, the additional space required is billed.
- Audit logs are cleared every hour. After you change the retention period of audit logs, expired audit logs will be deleted 1 hour later.
- After SQL audit is enabled, a large number of audit logs may be generated during peak hours. As a result, there are many audit log files temporarily stored in the instance, and the storage may be full. You are advised to enable **storage autoscaling**.

Precautions

- Enabling SQL audit deteriorates instance performance by about 5%.
- After SQL audit is disabled, all audit logs will be deleted immediately and cannot be recovered. Exercise caution when performing this operation.

Enabling SQL Audit

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

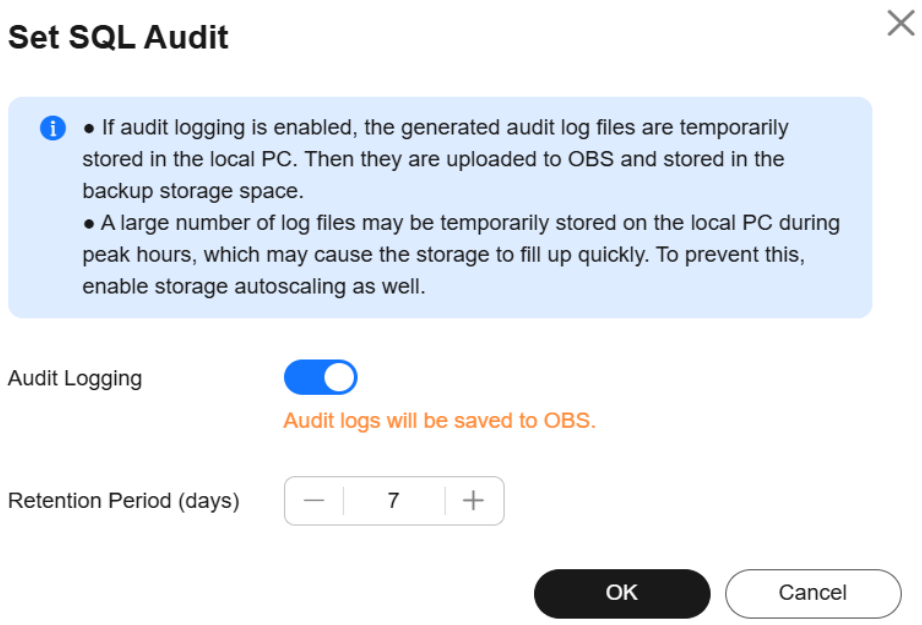
Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit**.

Figure 2-53 Setting SQL audit



Step 6 In the displayed dialog box, toggle on the **Audit Logging** switch and set the log retention period.

- To enable SQL audit, set  to  .
- Audit logs are retained for 7 days by default but can be retained from 1 to 732 days if needed.

Step 7 Click **OK**.

----End

Disabling SQL Audit

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

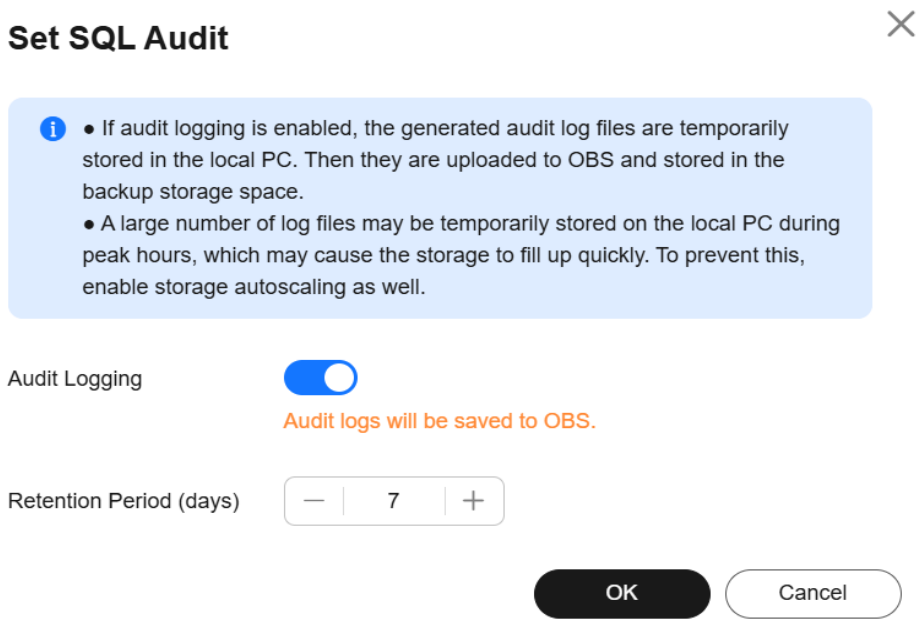
Step 2 Click  in the upper left corner and select a region and a project.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit**.

Figure 2-54 Setting SQL audit



Step 6 In the displayed dialog box, toggle off the **Audit Logging** switch and select the check box "I acknowledge that after audit log is disabled, all audit logs are deleted."

NOTICE

Deleted audit logs cannot be recovered. Exercise caution when performing this operation.

Step 7 Click **OK**.

----End

2.17.4 Downloading SQL Audit Logs

If you enable SQL audit, all SQL operations will be logged, and you can download audit logs to view details. The minimum time unit of audit logs is second. By default, SQL audit is disabled. Enabling this function may affect database 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 2-33**.

Figure 2-55 RDS for MariaDB audit logs

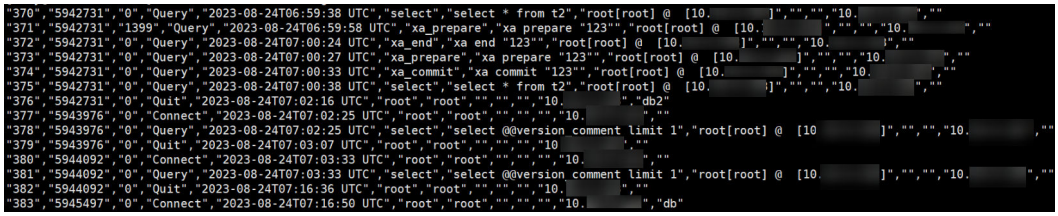


Table 2-33 Audit log field description

Parameter	Description
record_id	ID of a single record, which is the unique global ID of each SQL statement recorded in the audit log.
connection_id	ID of the session executed for 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	UTC time for the record.
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 connection is disconnected.)

Parameter	Description
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. For local connection, the field is empty.
default_db	Default database on which SQL statements are executed. **NOTE** Only when you have specified a database name using **-D** in the command for connecting to your DB instance, can the database name be queried in audit logs. If no database name has been specified, this parameter is left blank in audit logs. In the following example, the specified database name is **db**. **mysql -h** 10.10.0.233 **-P** 3306 **-u** root **-p -D** db

----End

2.18 DBA Assistant

2.18.1 Function Overview

Description

DBA Assistant provides visualized database O&M and intelligent diagnosis for developers and database administrators (DBAs), making database O&M easy and efficient. By analyzing alarms, resources, health data, performance metrics, and storage usage, it helps users quickly locate faults and keep track of instance status.

Scenarios

- Setting a slow session threshold can help you quickly identify abnormal sessions and kill the sessions when an exception occurs in your instance, so that your instance can recover quickly and ensure database availability.
- If your DB instance is unstable due to a large number of concurrent SQL requests from new services, you can set concurrency control rules for SQL statements to limit concurrent SQL statements and ensure instance stability.
- If your instance storage is full, you can learn about the storage usage and disk space distribution on the **Storage Analysis** page. Autoscaling is available for you to enable. After this function is enabled, the storage space is automatically scaled up when the storage space is too small.
- You can configure auto flow control to limit active connections in high burst traffic or abnormal read/write scenarios to ensure the availability of core workloads.

Supported Regions

You can view the supported regions in **Function Overview**.

Functions

Table 2-34 lists the functions supported by DBA Assistant.

Table 2-34 Function description

Function	Description	Reference
Dashboard	Shows the status of your instance, including alarms, resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.	**Viewing the Overall Status of a DB Instance**
Sessions	Displays a list of sessions and allows you to filter sessions. You can set a slow session threshold to identify abnormal sessions for urgent instance recovery, ensuring database availability.	• **Viewing Session Statistics** • **Setting a Slow Session Threshold**
Performance	Displays key metrics of your instance and provides metric comparison between different days. You can keep track of metric changes and detect exceptions in a timely manner.	**Viewing Performance Metrics**
Storage Analysis	Storage occupied by data and logs and historical changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DBA Assistant can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner.	• **Viewing Storage Usage** • **Viewing Table Diagnosis Results** • **Setting a Diagnosis Threshold** • **Viewing Top Databases and Tables by Physical File Size**
Slow Query Log	Displays slow queries within a specified time period. You can view top 5 slow query logs by user or client IP address, sort statistics, and identify sources of slow SQL statements.	**Viewing Slow Query Logs**

Function	Description	Reference
Concurrency Control	Restricts the execution of concurrent SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.	**Concurrency Control**
Auto Flow Control	Automatically detects database exceptions such as high vCPU usage and excessive active sessions, and limits traffic based on specified priorities. You can control traffic by database or user as required. Limiting traffic of non-core databases or from non-core users can ensure that core workloads remain stable.	**Auto Flow Control**
Daily Reports	Provides overall information about your instance status of the previous day, including slow SQL analysis and performance & storage analysis. You can download and subscribe to analysis reports. A daily diagnosis is recommended.	• **Viewing Diagnosis Reports** • **Subscribing to Diagnosis Reports**

2.18.2 Viewing the Overall Status of a DB Instance

On the **Dashboard** page, you can get knowledge of the overall status of your instance, including alarms, resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.

Functions

Table 2-35 lists the functions provided by Dashboard.

Table 2-35 Function description

Function	Description
Alarms	Shows alarms of different severities in the last 7 days. After you click the number next to an alarm severity, the **Alarm Rules** page is displayed, showing all alarm rules of the severity.
Resources	Shows the vCPU usage, memory usage, storage usage, and disk IOPS of your instance.
Key Performance Metrics	Shows vCPU utilization & slow query logs, connections, memory utilization, and disk reads/writes of your instance in the last hour.

Function	Description
Health	Shows the health status of your instance based on operational data analytics and intelligent algorithms.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

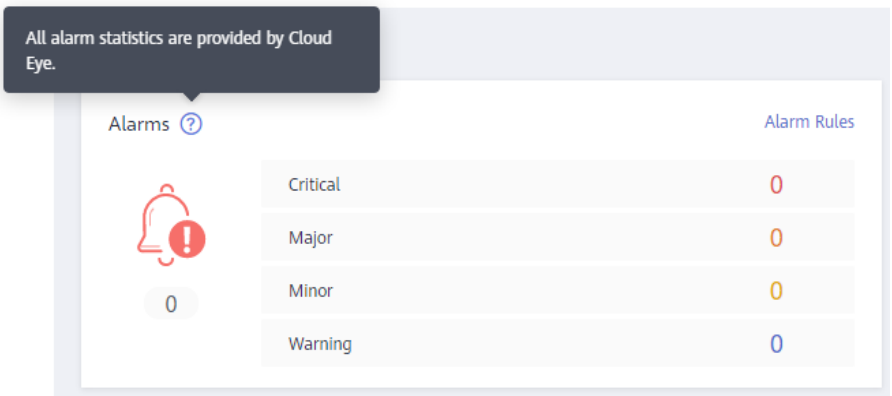
Step 6 On the **Dashboard** page, view the status of your instance.

- In the **Alarms** area, view alarm information of your instance. To view alarm details, click the number next to an alarm severity.

  If attention needs to be paid to specific metrics such as CPU usage, storage space usage, and connection usage, you can create alarm rules and configure alarm policies for these metrics.

  For example, if you want to pay special attention to session traffic for new services, you can configure an alarm policy for connection usage by modifying the alarm rule of your instance. For details about how to modify an alarm rule, see Modifying an Alarm Rule.

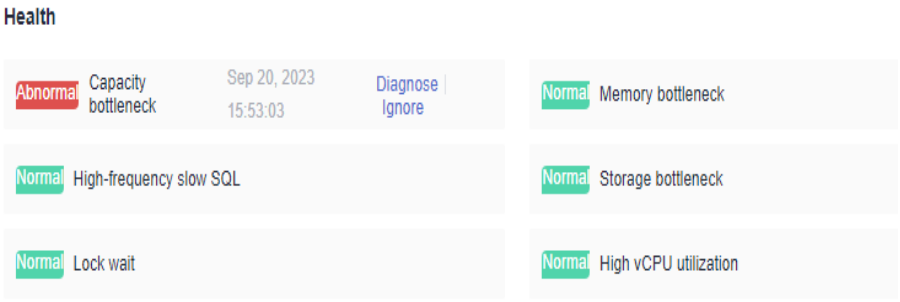
  Figure 2-56 Alarms

- In the **Health** area, view the health diagnosis results of your instance.

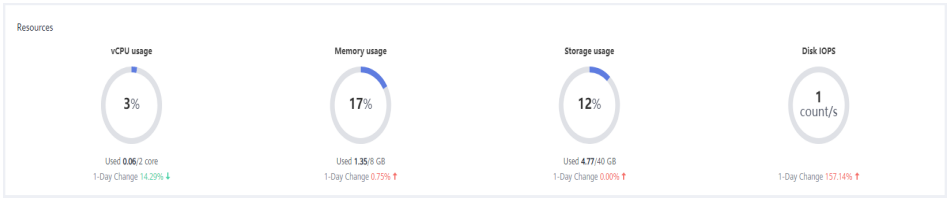
  For example, if an exception is detected for **Capacity bottleneck**, click **Diagnose** to view the storage usage and disk space distribution, and scale up the storage or delete unnecessary data as required. For details, see Viewing Storage Usage.

Figure 2-57 Health



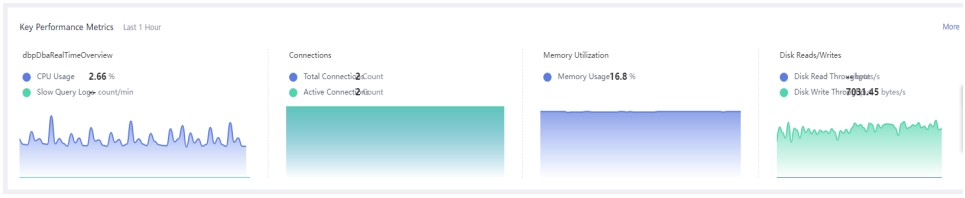
- In the **Resources** area, view the resource usages of your instance.

Figure 2-58 Resources



- In the **Key Performance Metrics** area, view the key performance metrics of your instance in the last hour.

Figure 2-59 Key Performance Metrics



----End

2.18.3 Managing Real-Time Sessions

2.18.3.1 Viewing Session Statistics

DBA Assistant allows you to view session statistics of your instance, including slow sessions, active sessions, and total sessions, helping you learn about the distribution of sessions in different dimensions.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

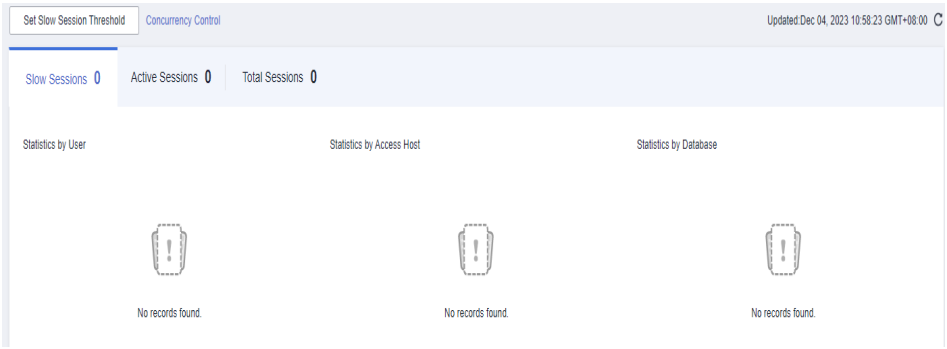
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Sessions** tab. You can view slow sessions, active sessions, and total sessions by the following three dimensions:

- User
- Access host
- Database

Figure 2-60 Session statistics



----End

2.18.3.2 Setting a Slow Session Threshold

You can set a slow session threshold to identify sessions whose execution time is longer than the threshold. This allows you to identify abnormal sessions and kill the sessions to restore the database.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

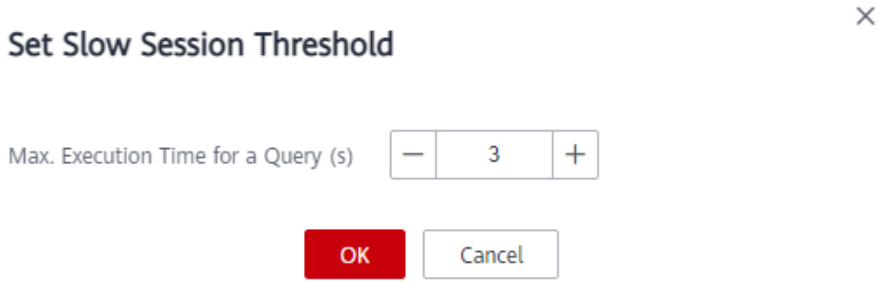
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Sessions** tab.

Step 7 Click **Set Slow Session Threshold** and specify **Max. Execution Time for a Query (s)**.

Max. Execution Time for a Query (s) indicates how long a session has been executed before it can be considered as a slow session. The default value is **3**. The value ranges from 1 to 86,400, in seconds.

Figure 2-61 Setting a slow session threshold



Step 8 Click **OK**. Sessions whose execution time exceeds the threshold are automatically displayed.

Step 9 In the **Sessions** area, select the target session IDs based on the instance status and service requirements, and click **Kill Session**.

Figure 2-62 Viewing the session list

NOTE

The **Kill Session** operation ends only the selected sessions. It does not affect other sessions or services.

Step 10 Click **OK**.

----End

2.18.4 Viewing Performance Metrics

DBA Assistant allows you to view the performance metrics of your DB instance. Historical trends of performance metrics within a specified time period help you learn about the status and resource usage of your DB instance. If any alarm is reported, you can take actions timely.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

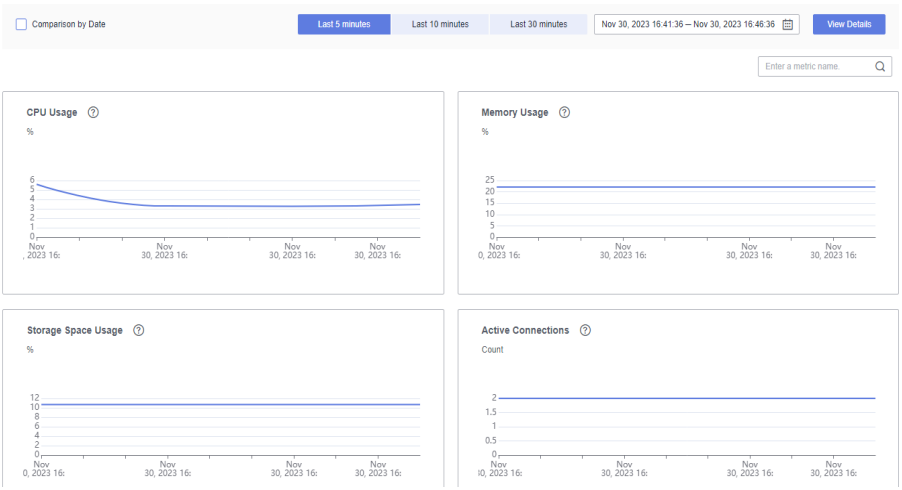
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Performance** tab to view the performance metrics of your instance.

- By default, **Comparison by Date** is selected. The system automatically displays the comparisons of the performance metrics between the last hour and the same time period of the previous day for your instance.

  You can also select different days and time ranges and click **View Details**.
- If you deselect **Comparison by Date**, you can view the dynamic trend of each performance metric of your instance in a specified period of the current day. By default, the metric trends in the last 5 minutes are displayed. You can select **Last 10 minutes**, **Last 30 minutes**, or customize a time period and click **View Details**. The system automatically refreshes and displays the dynamic trend of each metric of the instance in the specified time period.

Figure 2-63 Performance metric trends



----End

2.18.5 Subscribing to Intelligent O&M

To use the **Tablespaces**, **Slow Query Logs**, and **Auto Flow Control** functions, you need to subscribe to Intelligent O&M first. This section describes how to subscribe to Intelligent O&M.

Procedure

Step 1 Log in to the management console.

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

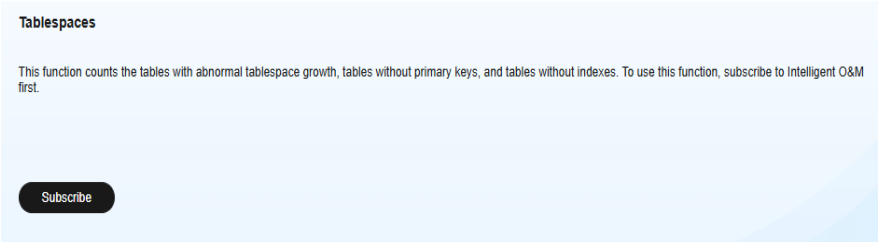
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab.

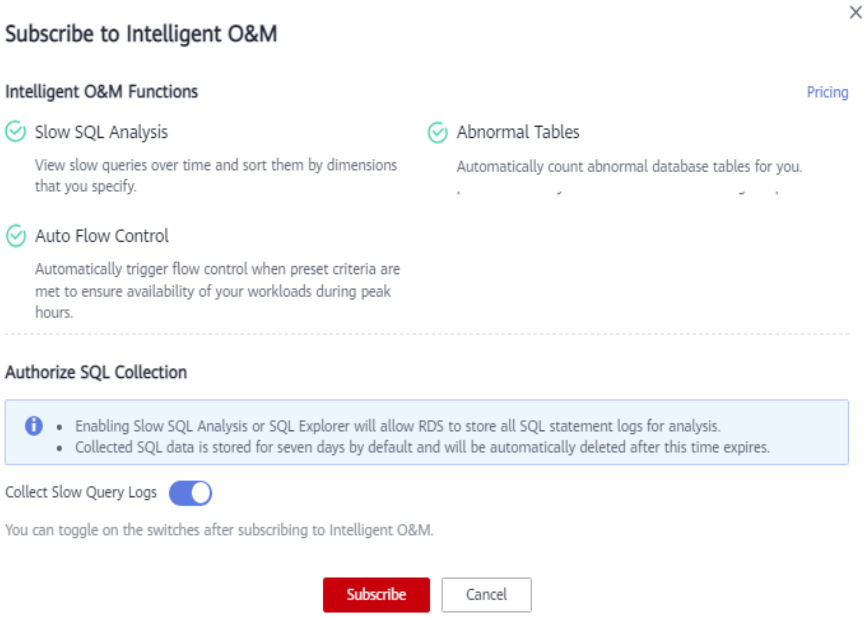
Figure 2-64 Tablespaces



Step 7 Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.

If you select **Collect Slow Query Logs**, slow SQL statements will be collected and analyzed. For details, see **Viewing Slow Query Logs**.

Figure 2-65 Subscribing to Intelligent O&M



Step 8 Select "I have read and understand the billing rules." and click **Subscribe**.

----End

2.18.6 Viewing Storage Usage

DBA Assistant allows you to view the storage usage of your DB instance in real time to prevent full storage space.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance name.

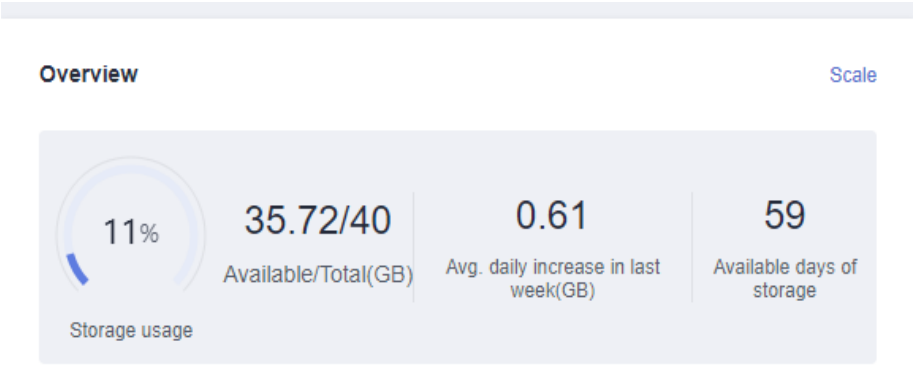
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab.

- In the **Overview** area, view the storage usage, including the available storage space and total storage space.

  If the storage usage reaches 87% or higher, you can click **Scale** to scale up the storage. For details about constraints and billing, see **Scaling Up Storage Space**.

  Figure 2-66 Storage usage

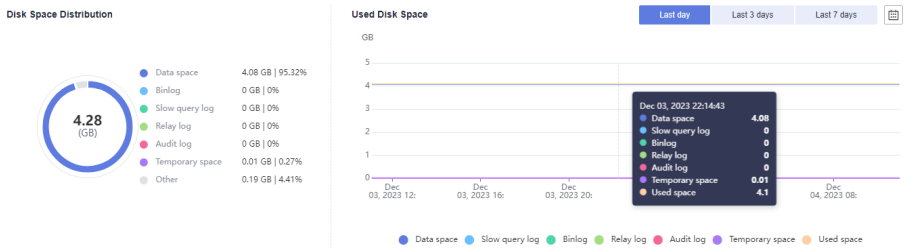
  

  NOTE

  If the average daily increase in last week is 0 GB, the estimated available days of storage are unlimited and are not displayed.

- In the **Disk Space Distribution** area, view the space distribution of your instance. For details, see **Table 2-36**.

  Figure 2-67 Disk space distribution

NOTE

If the total number of files in your disk space (including data space, binlog space, slow query log space, relay log space, audit log space, temporary space, and other space) exceeds 10,000, RDS will not collect information about the files or display disk space distribution and usages over time on the console. This prevents performance slowdowns caused by collecting statistics on too many files. If this happens, submit a service ticket.

Table 2-36 Parameter description

Parameter	Description
Data space	Disk space for storing user data
Binlog	Disk space for storing binlogs
Slow query log	Disk space for storing slow logs
Relay log	Disk space for storing relay logs
Audit log	Disk space for storing audit logs
Temporary space	Disk space for storing temporary files
Other	Disk space for storing files such as **ib_buffer_pool**, **ib_doublewrite** and **error.log** generated by the instance.

----End

FAQ

Q: What can I do if the storage space of my DB instance is full?

A: Reduce the storage usage to below 87% so that the DB instance becomes available and data can be written to the instance. You can use either of the following methods to reduce the storage usage:

- Scale up the storage space: **Services are not interrupted during storage scale-up**. You can also enable autoscaling. When the available storage of a DB instance drops to the threshold, autoscaling is triggered.
- Reduce data: Delete useless historical data.
  a. If your instance becomes read-only, you need to submit a service ticket to cancel the read-only status first. If your instance is not in the read-only state, you can delete data directly.
  b. Check the top 50 databases and tables with large physical files and identify the historical table data that can be deleted. For details, see **Viewing Top Databases and Tables by Physical File Size**.
  c. To clear up space, you can optimize tables with a high fragmentation rate during off-peak hours.

     To delete data of an entire table, run **DROP** or **TRUNCATE**. To delete part of table data, run **DELETE** and **OPTIMIZE TABLE**.

- If temporary files generated by sorting queries occupy too much storage space, optimize your SQL query statements.

  You can query **slow query logs**, and analyze and optimize the problematic SQL statements.

2.18.7 Viewing Table Diagnosis Results

Intelligent table diagnosis can diagnose tables with abnormal tablespace growth, tables without primary keys, and tables without indexes, helping you quickly locate abnormal tables.

To use this function, you need to subscribe to Intelligent O&M first. For details, see **Subscribing to Intelligent O&M**.

Procedure

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

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

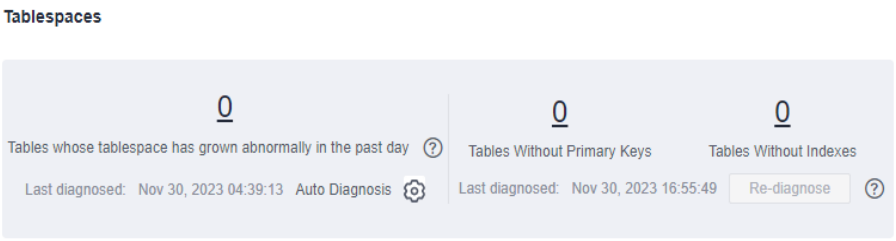
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab to view the table diagnosis results of your instance.

Figure 2-68 Table diagnosis results



----End

2.18.8 Setting a Diagnosis Threshold

You can set a diagnosis threshold to identify abnormal tables whose tablespace is above the threshold.

To use this function, you need to subscribe to Intelligent O&M first. For details, see **Subscribing to Intelligent O&M**.

Procedure

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

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

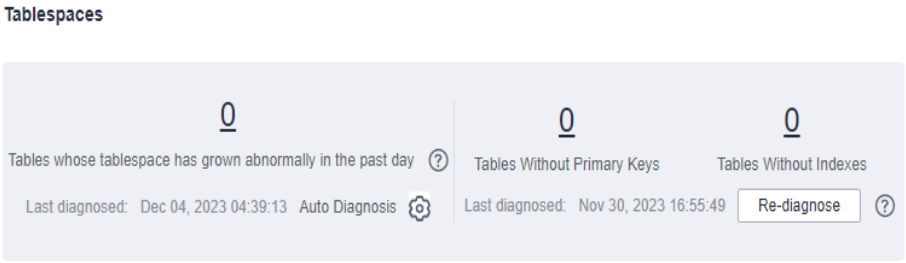
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab to view the table diagnosis results of your instance.

Figure 2-69 Table diagnosis results



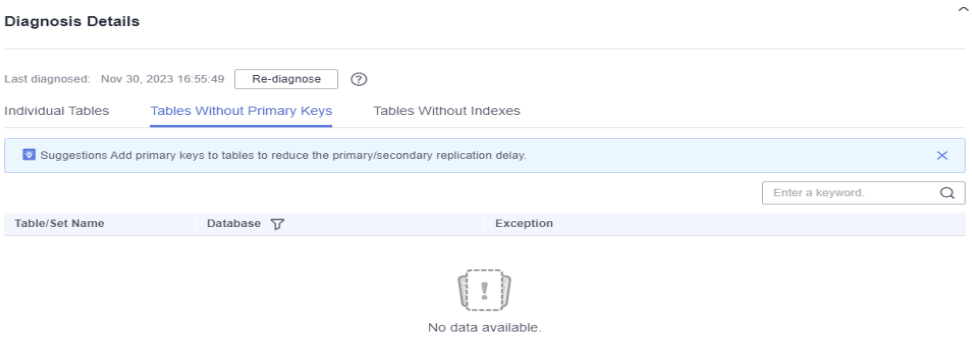
Step 7 Click next to **Auto Diagnosis**. In the displayed dialog box, configure a daily tablespace increase limit. The value ranges from 1 to 100,000,000, in MB.

Figure 2-70 Configuring a daily tablespace increase limit

Step 8 Click **OK**. The system automatically identifies the tables whose tablespace exceeds the specified threshold in the past day and counts the number of those tables in the **Tablespaces** area.

Step 9 Click a number and view the details about abnormal tables on the **Diagnosis Details** page.

Figure 2-71 Diagnosis details



----End

FAQ

- Q: What can I do if there are tables whose tablespace has grown abnormally in the past day?

  A: Check tablespace fragmentation and reclaim fragmented space. Do not use **DELEDTE** to clear data. If you have any other questions, submit a service ticket.

- Q: What is the impact of tables without primary keys on my DB instance?

  A: Tables without primary keys can cause slow SQL statements, affecting instance stability. You are advised to add primary keys to such tables to reduce the primary/standby replication delay.

- Q: What is the impact of tables without indexes on my DB instance?

  A: Tables without indexes can cause slow SQL statements, affecting instance stability. You are advised to add indexes to table fields for more efficient query.

2.18.9 Viewing Top Databases and Tables by Physical File Size

In combination with disk space distribution, top 50 databases and tables by physical file size help you identify the databases and tables with high disk space usage.

Procedure

Step 1 Log in to the management console.

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

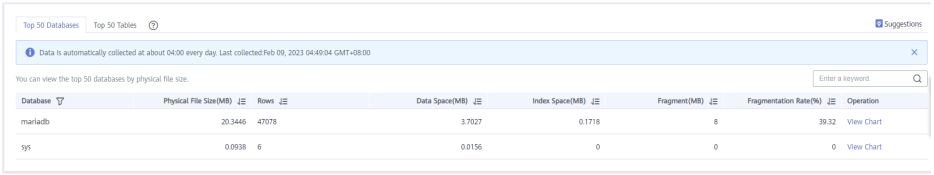
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Storage Analysis** tab to view top 50 databases and tables.

Figure 2-72 Top 50 databases and tables

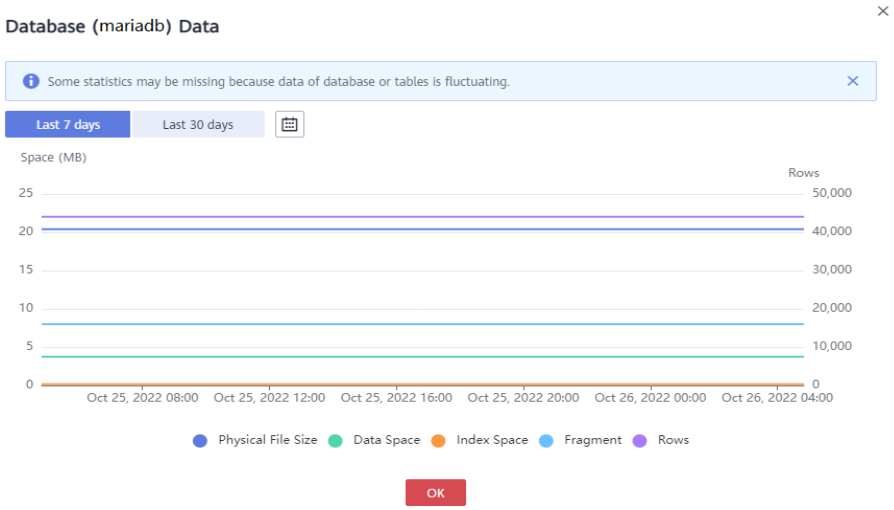


NOTE

- Physical file sizes are precisely recorded, but other fields' values are estimated. If there is a large gap between a file size and another field, run ANALYZE TABLE on the table.
- A database or table whose name contains special characters, including slashes (/) and #p#p, is not counted.
- If there are more than 50,000 tables in your instance, to prevent data collection from affecting the instance performance, top databases and tables will not be counted.

Step 7 Locate a database and click **View Chart** in the **Operation** column. You can view data volume changes in the last 7 days, last 30 days, or a custom time period (spanning no more than 30 days).

Figure 2-73 Database Data



----End

2.18.10 Viewing Slow Query Logs

Slow query logs help you locate SQL statements that process a large amount of data, scan a large number of rows, or run for a long time, so that you can optimize them to improve database performance.

To use this function, you need to subscribe to Intelligent O&M first. For details, see **Subscribing to Intelligent O&M**.

If you did not subscribe to Intelligent O&M, you can view only the data of the last hour. The data will be automatically deleted when it expires.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

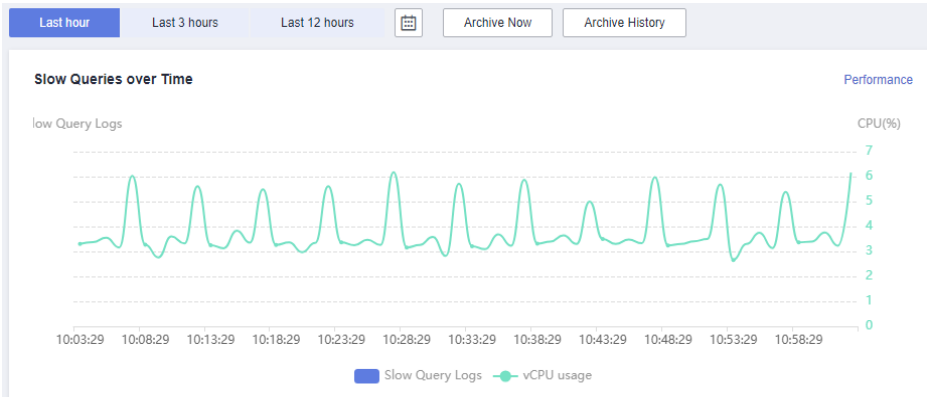
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Slow Query Log** tab.

Step 7 In the **Slow Queries over Time** area, you can view the slow query log and vCPU usage trends of your DB instance.

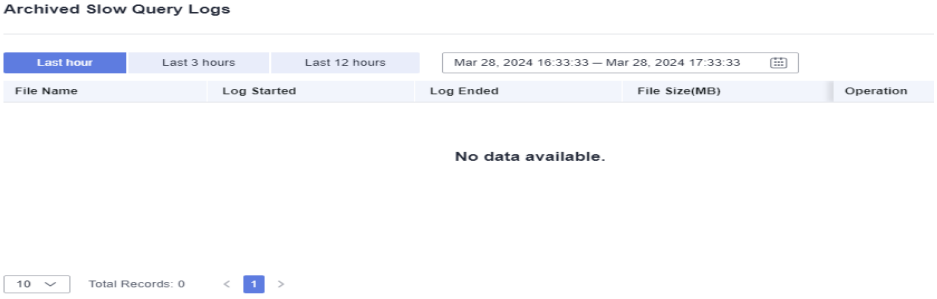
You can view the **Slow Queries over Time** chart in the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day).

Figure 2-74 Slow Queries over Time



Step 8 Click **Archive History** and view slow query log data on the displayed page, including **Log Started**, **Log Ended**, and **File Size(MB)**.

Figure 2-75 Archived Slow Query Logs

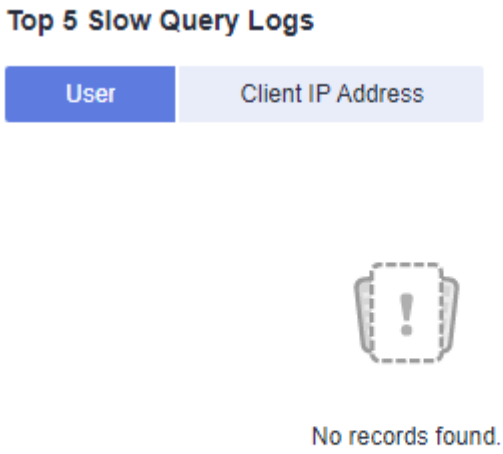


NOTE

Slow query logs are automatically archived every three minutes. Alternatively, you can click **Archive Now** and then view the latest statistics.

Step 9 In **Top 5 Slow Query Logs** area, view the top 5 slow SQL statements sorted by user or client IP address.

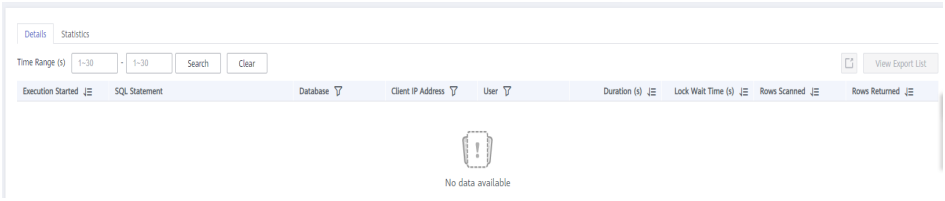
Figure 2-76 Top 5 Slow Query Logs



Step 10 In the details list, view details about slow query logs.

- You can filter slow query logs by SQL statement, database, client IP address, user, execution duration, and scanned rows.
- To export the slow query log list, click **Export**.
- To view log export history, click **View Export List**.

Figure 2-77 Details



----End

2.18.11 Concurrency Control

You can create rules to control concurrent execution of SQL statements by specifying SQL type, keywords, and maximum concurrency. To maintain better performance at high concurrency, SQL statements that meet the specified SQL type and keyword and exceed the maximum concurrency will not be executed.

Constraints

- The rule you are creating will be applied only to the current instance.
- If a SQL statement matches multiple concurrency control rules, only the most recently added rule is applied.

- SQL statements that have been executed before a concurrency control rule is added are not counted.
- If the replication delay is too long, adding or deleting a concurrency control rule for a read replica does not take effect immediately.
- You are advised to upgrade the minor kernel version to the latest version.
- Too many concurrency control rules affect the database performance. Delete unnecessary rules after using them.
- This function controls how many statements can run at the same time. However, it does not limit concurrency for:
  - System tables
  - Queries where no database data is involved, such as **SELECT sleep(*xxx*);**
  - Account **root**
  - SQL statements in stored procedures, triggers, and functions

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Concurrency Control**.

Step 7 Click **Add Rule**. Configure the parameters listed in Table 2-37.

Figure 2-78 Adding a rule

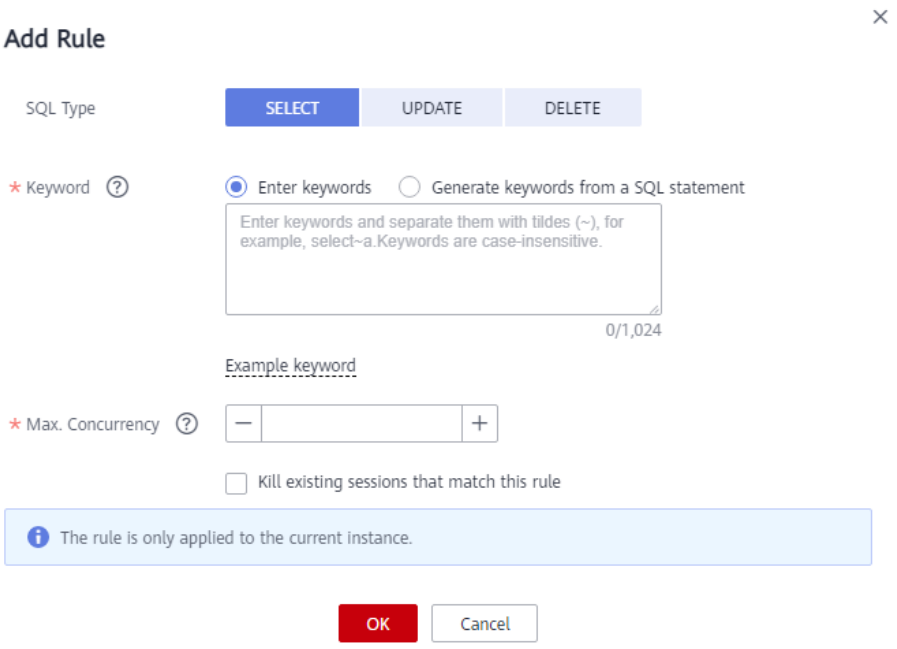


Table 2-37 Parameter description

Parameter	Description
SQL Type	There are three options: **SELECT**, **UPDATE**, and **DELETE**.
Keyword	A maximum of 128 keywords (case-insensitive) are supported. You can specify keywords in either of the following ways: • **Enter keywords**: Take **select~a** as an example. **select** and **a** are two keywords contained in a concurrency control rule. The keywords are separated by a tilde (~). In this example, the rule restricts the execution of only the SQL statements containing keywords **select** and **a**. • **Generate keywords from a SQL statement**: You can enter a SQL statement and then click **Generate Keyword**. The generated keywords are for reference only. Exercise caution when using them. SQL statements match the keywords from first to last. For example, if one rule contains the keyword **a~and~b**, the statement **xxx a>1 and b>2** can match the keyword, but **xxx b>2 and a>1** cannot.
Max. Concurrency	If the number of concurrent SQL statements matching the keyword exceeds this limit, the SQL statements will not be executed. The value ranges from 0 to 1,000,000,000.
Kill existing sessions that match this rule	Selecting this option will not kill the connection sessions of user **root**.

Step 8 Confirm the settings and click **OK**.

Step 9 Toggle on the concurrency control switch  .

NOTE

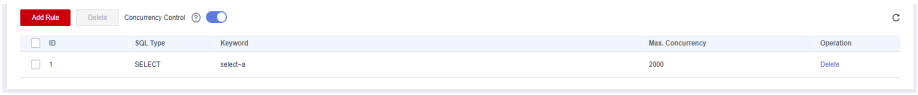
Concurrency control rules take effect only after concurrency control is enabled.

----End

Related Operations

To delete a concurrency control rule, locate it in the rule list and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

Figure 2-79 Deleting a rule



2.18.12 Auto Flow Control

Auto Flow Control helps ensure availability of your workloads. It restricts the execution of SQL statements during peak hours or when there are read/write exceptions by limiting how many SQL statements can be executed at the same time.

After it is enabled, the system performs flow control on sessions when the criteria you specify for your instance are met (for example, the number of active connections to your instance exceeds the **Max. Active Connections** parameter value).

To use this function, **subscribe to Intelligent O&M** first.

Constraints

To use auto flow control, submit a service ticket to apply for required permissions.

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Auto Flow Control**.

Step 7 Click **Auto Flow Control**.

Step 8 Toggle on and configure required parameters. For details about the parameters, see **Table 2-38**.

Example for setting Auto Flow Control parameters

Set **Time Window** to **15:00-18:00**, **Max. Duration** to **Last 5 minutes**, **vCPU usage ≥** to **90%**, **Active sessions ≥** to **20**, and **and Duration (min)** to **5**. When all the criteria are met, Auto Flow Control is triggered. If your vCPU usage or number of active sessions falls below the threshold during the time window, flow control ends.

Figure 2-80 Example

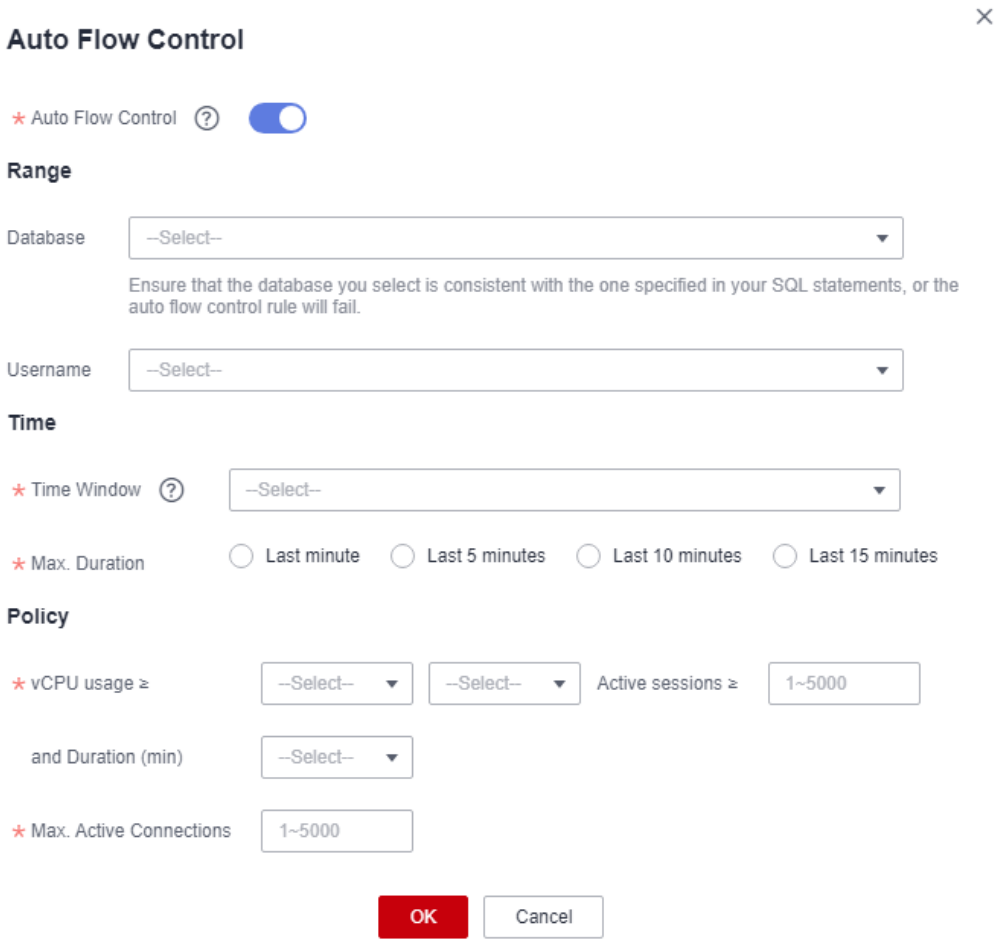


Table 2-38 Parameter description

Parameter	Description
Database	The name of the database for which auto flow control needs to be enabled. Ensure that the database you select is consistent with the one specified in your SQL statements, or the auto flow control policy will fail.
Username	The name of the user that auto flow control is applied. If no username is specified, auto flow control will be applied to user **root**.
Time Window	The time when flow control is applied. Auto flow control can be triggered only once within the time window.
Max. Duration	Maximum length of time that SQL statements matching the auto flow control policy can be throttled within the time window.

Parameter	Description
vCPU usage	vCPU usage threshold for the instance. You also need to specify the relationship between the vCPU usage and active sessions. Their relationship can be **and** or **or**.
Active sessions	Threshold for active sessions. Value range: 1 to 5000
Duration (min)	How long the vCPU usage and active sessions exceed the specified values. For example, if you set **vCPU usage ≥** to **90%**, **Active sessions ≥** to **1000**, and **Duration (min)** to **30**, auto flow control will be triggered only when the vCPU usage and active sessions exceed 90% and 1,000 for 30 minutes.
Max. Active Connections	Maximum number of active connections allowed. Value range: 1 to 5000 For example, if you set **Max. Active Connections** to **500**, the system will automatically end some active connections when necessary to keep the number of active sessions within 500.

Step 9 Click **OK**. You will see a record generated on the page every time Auto Flow Control is triggered. You can see historical details, too.

----End

2.18.13 Managing Diagnosis Reports

2.18.13.1 Viewing Diagnosis Reports

You can start a health diagnosis on your DB instance and view the current and historical diagnosis reports.

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

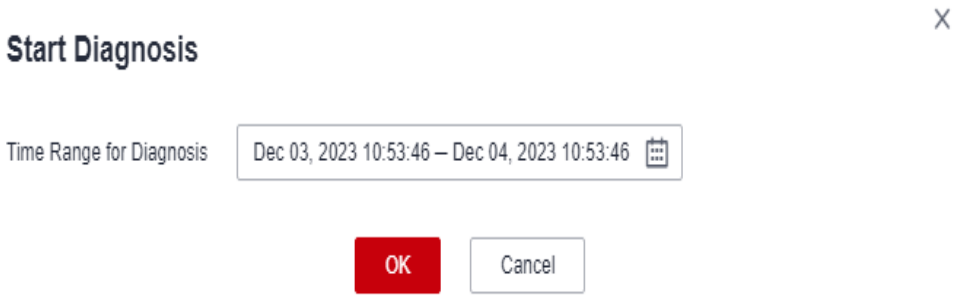
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Daily Reports** tab.

Step 7 Click **Start Diagnosis** and select a time range for the diagnosis.

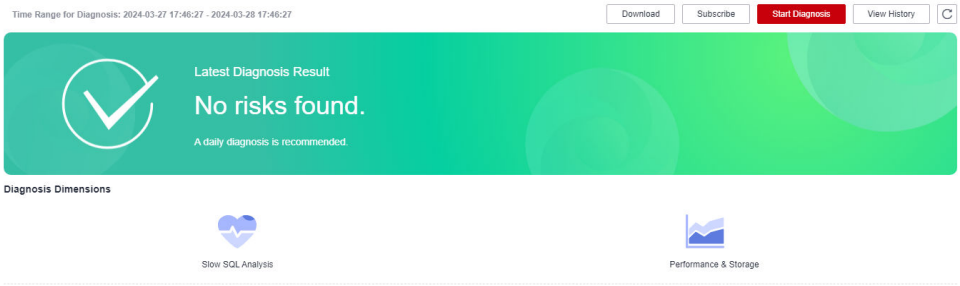
Figure 2-81 Start Diagnosis



Step 8 Click **OK**. You can also view historical diagnosis reports or download a report to your local PC.

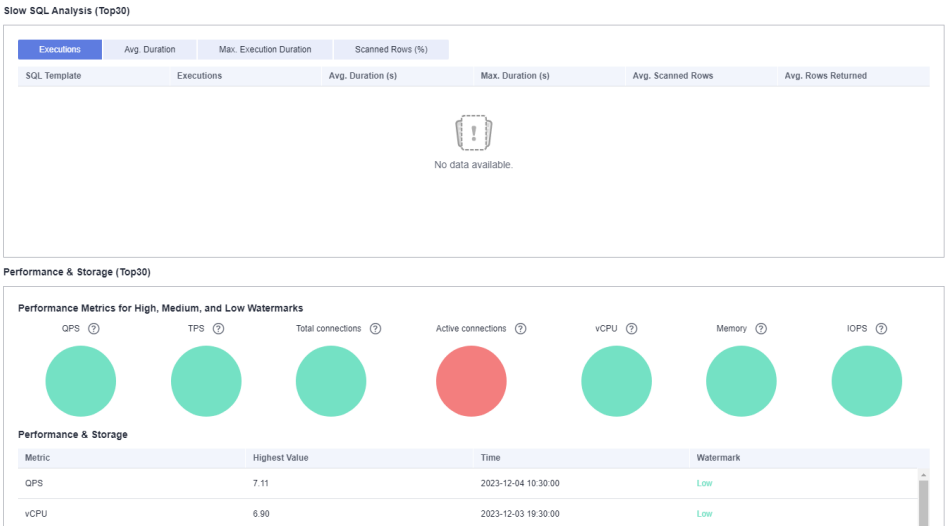
- To view historical diagnosis reports, click **View History** in the upper right corner of the page.
- To download a report to your local PC, click **Download** in the upper right corner of the page.

Figure 2-82 Daily Reports



Step 9 In the **Diagnosis Dimensions** area, click **Slow SQL Analysis** or **Performance & Storage** to view details.

Figure 2-83 Viewing report analysis details



----End

2.18.13.2 Subscribing to Diagnosis Reports

After you subscribe to diagnosis reports, Simple Message Notification (SMN) will send diagnosis exception reports to the preset email address so that you can learn about the overall health status of your DB instance in real time.

Billing

When you use SMN, only pay for what you use. There are no minimum fees. For details, see **Billing**.

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Daily Reports** tab.

Step 7 In the upper right corner of the page, click **Subscribe** and set subscription parameters. For details about the parameters, see **Table 2-39**.

Table 2-39 Subscription parameters

Parameter	Description
Subscription	Select **By topic** or **By email**.
Topics	A topic is used to publish messages and subscribe to notifications. It serves as a message transmission channel between publishers and subscribers. If there are no topics you want to select, **create one**. After a topic is created, click **Add Subscription** in the **Operation** column of the topic. In the displayed dialog box, specify a protocol (only **Email** is supported) and an endpoint.
Email Addresses	If you select **By email** for **Subscription**, you need to specify **Email Addresses**.

Step 8 Click **OK**.

----End

Related Operations

If you want to unsubscribe from diagnosis reports, click **Unsubscribe** in the upper right corner of the page. In the displayed dialog box, confirm the information and click **OK**.

2.19 Task Center

2.19.1 Viewing a Task

You can view the progresses and results of scheduled and instant tasks on the **Task Center** page.

Task Details

RDS allows you to view and manage the following instant tasks:

- Creating DB instances
- Creating read replicas
- Scaling up storage space
- Switching primary/standby DB instances
- Rebooting DB instances
- Binding EIPs to DB instances
- Unbinding EIPs from DB instances
- Restoring data to new DB instances

RDS allows you to view and manage the following scheduled tasks:

- Changing MariaDB instance classes
- Rebooting MariaDB 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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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, order ID, 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 can only show up to 30 days of past tasks.

- 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

2.19.2 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.

Precautions

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 ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed **Instant Tasks** page, locate the task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

You can delete the records of instant tasks in any of the following statuses:

- 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 ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the 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 **OK** to cancel the task. Then, click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

You can delete the records of scheduled tasks in any of the following statuses:

- Completed
- Failed
- Canceled
- To be executed
- To be authorized

----End

2.20 Managing Tags

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

Constraints

- Log in to the management console. Click **Service List** and choose **Management & Governance** > **Tag Management Service**. Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- A maximum of 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 Log in to the management console.

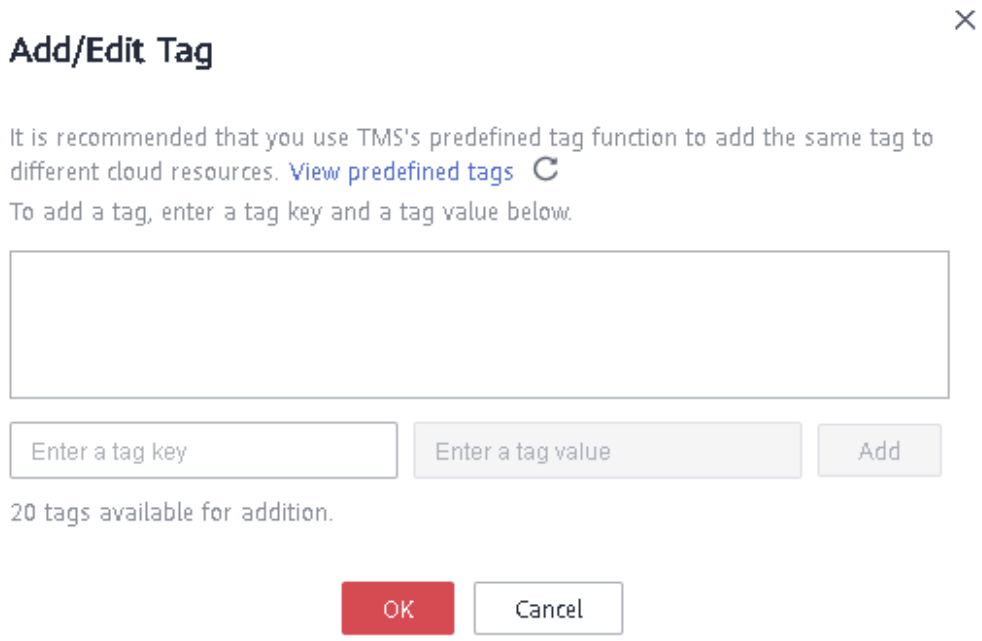
Step 2 Click in the upper left corner and select a region and a project.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane, choose **Tags**. On the displayed page, click **Add/Edit Tag**.

Figure 2-84 Adding/Editing a tag



Step 6 In the displayed dialog box, enter a tag key and value, click **Add** and then click **OK**.

- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with your DB instances (except the current instance).
- The tag key must be unique. It must consist of 1 to 128 characters, including letters, digits, spaces, and the following characters: _ **. : = + - @**. However, it cannot start or end with a space, or start with **_sys_**.

- The tag value (optional) can consist of up to 255 characters, including letters, digits, spaces, and the following characters: _ **. : / = + - @**. However, it cannot start or end with a space.

View and manage the tag on the **Tags** page.

----End

Deleting a Tag

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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Tags** page, locate the tag to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.

Verify that the tag is no longer displayed on the **Tags** page.

----End

2.21 Managing Quotas

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of RDS quotas include the maximum number of DB instances that you can create. Quotas are put in place to prevent excessive resource usage.

If a quota cannot meet your needs, apply for a higher quota.

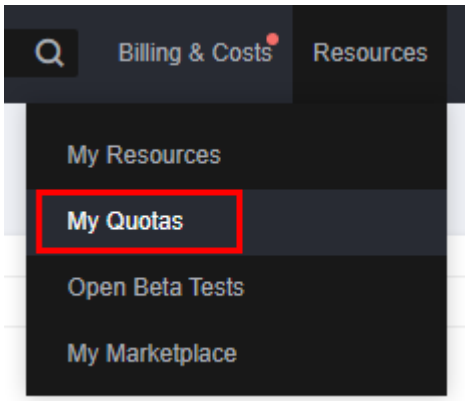
Viewing Quotas

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

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

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Figure 2-85 My quotas



Step 4 On the **Quotas** page, view the used and total quotas of each type of resources.

----End

Increasing Quotas

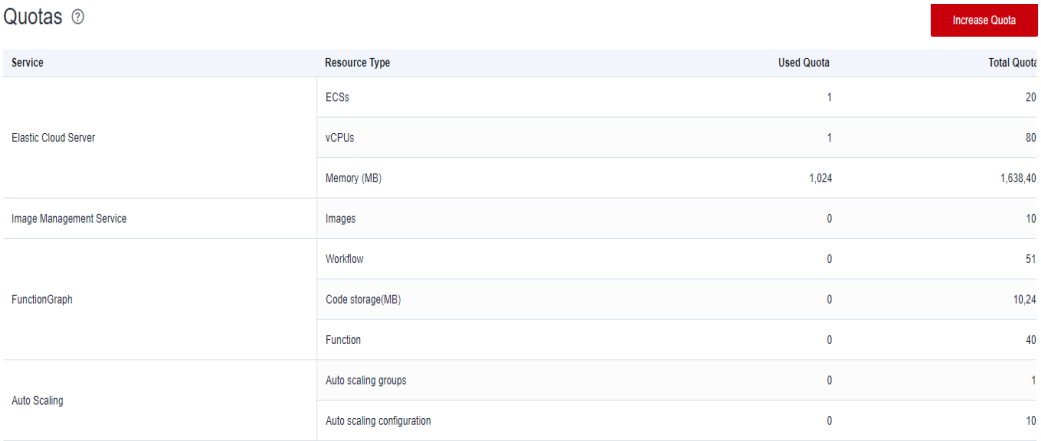
Step 1 **Log in to the management console**.

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

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 2-86 Increasing quotas



Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for adjustment.

Step 6 After all required parameters are configured, select the agreement and click **Submit**.

----End

3 Working with RDS for PostgreSQL

3.1 Using IAM to Grant Access to RDS

3.1.1 Creating a User and Granting Permissions

This chapter describes how to use **Identity and Access Management (IAM)** for fine-grained permissions management for your RDS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing RDS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your RDS resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

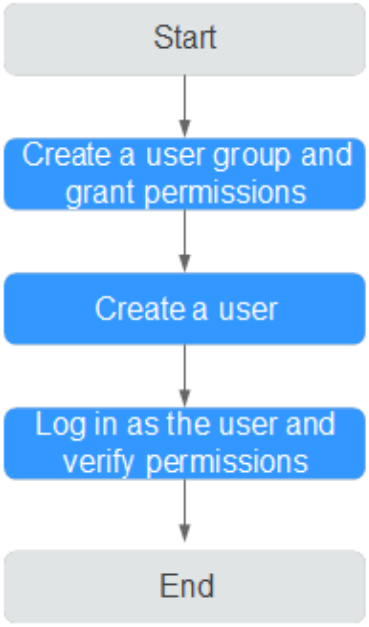
This section describes the procedure for granting permissions (see **Figure 3-1**).

Prerequisites

Learn about the permissions (see **Permissions Management**) supported by RDS and choose policies or roles according to your requirements. For the system policies of other services, see **System Permissions**.

Process Flow

Figure 3-1 Process for granting RDS permissions



1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console, and attach the **RDS ReadOnlyAccess** policy to the group.

   NOTE

   To use some interconnected services, you also need to configure permissions of such services.

   For example, to connect to your DB instance through the console, configure the **DAS FullAccess** permission of Data Admin Service (DAS) besides **RDS ReadOnlyAccess**.

2. **Create an IAM user and add it to the user group**.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the RDS console by using the created user, and verify that the user only has read permissions for RDS.

   - Choose **Service List** > **Relational Database Service** and click **Buy DB Instance**. If a message appears indicating that you have insufficient permissions to perform the operation, the **RDS ReadOnlyAccess** policy has already been applied.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **RDS ReadOnlyAccess** policy has already taken effect.

3.1.2 RDS Custom Policies

Custom policies can be created to supplement the system policies of RDS. For the actions supported for custom policies, see **Permissions Policies and Supported Actions**.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. The following section contains examples of common RDS custom policies.

Example Custom Policies

- Example 1: Allowing users to create RDS DB instances

```
{
    "Version": "1.1",
    "Statement": [{
        "Effect": "Allow",
        "Action": ["rds:instance:create"]
    }]
}
```

- Example 2: Denying RDS DB instance deletion

  A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user include both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

  The following method can be used if you need to assign permissions of the **RDS FullAccess** policy to a user but you want to prevent the user from deleting RDS DB instances. Create a custom policy for denying RDS DB instance deletion, and attach both policies to the group the user belongs to. Then, the user can perform all operations on RDS DB instances except deleting RDS DB instances. The following is an example deny policy:

```
{
    "Version": "1.1",
    "Statement": [{
        "Action": ["rds:instance:delete"],
        "Effect": "Deny"
    }]
}
```

3.2 Buying an RDS for PostgreSQL DB Instance

Scenarios

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

RDS for PostgreSQL supports the yearly/monthly and pay-per-use billing modes. RDS allows you to tailor your compute resources and storage space to your business needs.

Prerequisites

- You have **registered a Huawei ID and enabled Huawei Cloud services**.

Procedure

Step 1 Go to the **Buy DB Instance** page.

Step 2 On that page, click the **Custom Config** tab, select a billing mode, and configure information about your DB instance. Then, click **Buy**.

- Billing Mode
  - **Yearly/Monthly**: If you select this mode, skip **Step 3** and go to **Step 4**.
  - **Pay-per-use**: If you select this mode, go to **Step 3**.
- Engine Options

Figure 3-2 Billing mode and basic information

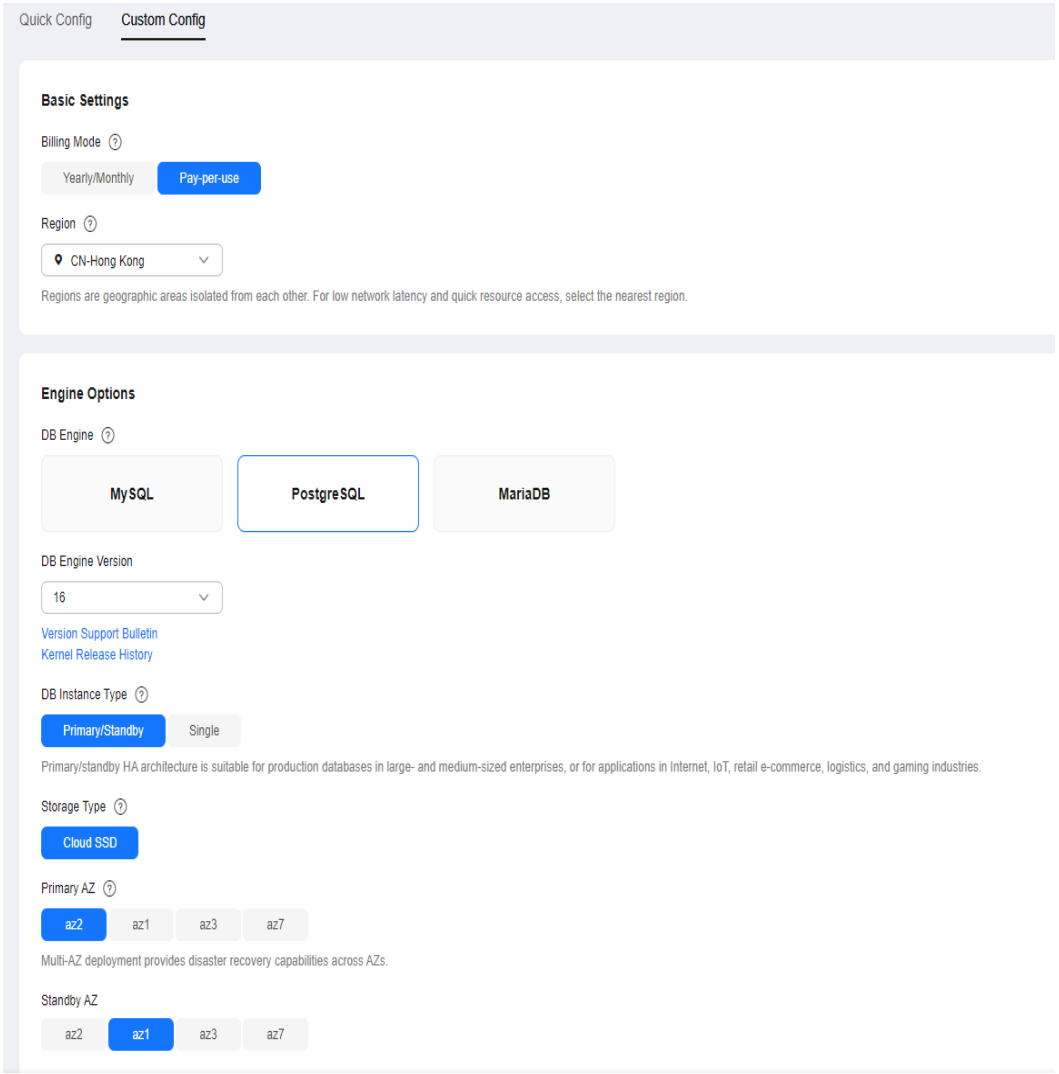


Table 3-1 Basic information

Parameter	Description
Region	Region where your resources are located. **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 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 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. Some regions support both single AZs and multiple AZs and some only support single AZs. To achieve high reliability, RDS will automatically deploy your primary and standby instances in different physical servers even if you deploy them in the same AZ. If you attempt to create primary/standby DB instances in the same AZ in a Dedicated Computing Cluster (DCC) and there is only one physical server available, the creation will fail. 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 less expensive 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.

Parameter	Description
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. – **Cloud SSD**: cloud disks used to decouple storage from compute. – **Extreme SSD**: uses 25GE network and RDMA technologies to provide you with up to 1,000 MB/s throughput per disk and sub-millisecond latency. **NOTE** – The cloud SSD and extreme SSD storage types are supported with general-purpose, dedicated, and Kunpeng general-enhanced DB instances. – If you have purchased the Dedicated Distributed Storage Service (DSS), only the storage type that you have selected when you buy the DSS service is displayed. – The IOPS supported by cloud SSDs depends on the I/O performance of Elastic Volume Service (EVS) disks. For details, see the description about ultra-high I/O in **Disk Types and Performance** of *Elastic Volume Service Service Overview*. – The IOPS supported by extreme SSDs depends on the I/O performance of Elastic Volume Service (EVS) disks. For details, see the description about extreme SSDs in **Disk Types and Performance** of *Elastic Volume Service Service Overview*.

- Instance Configuration

Figure 3-3 DB instance specifications

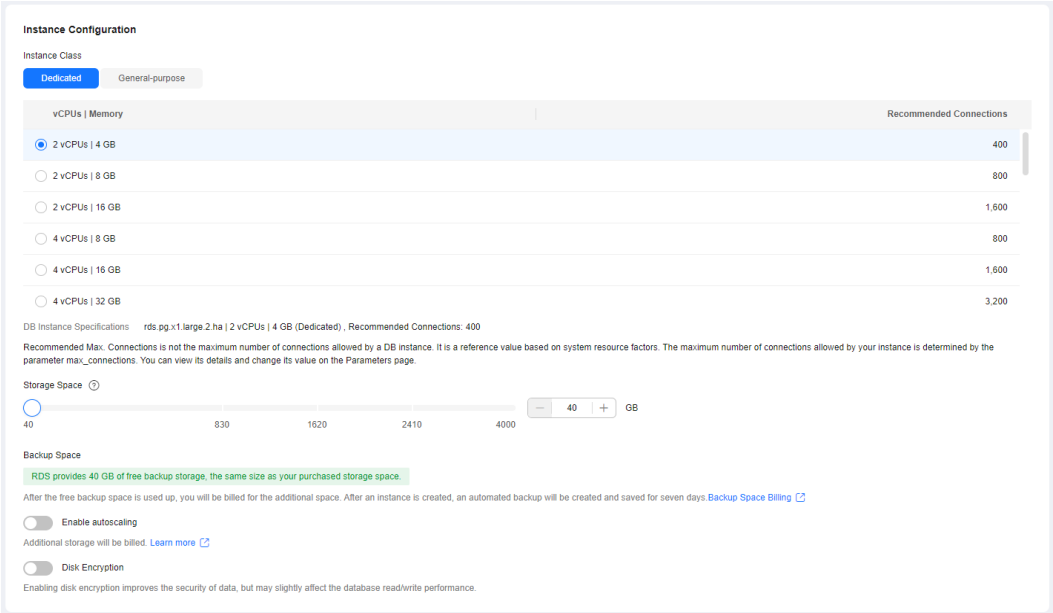


Table 3-2 Instance specifications

Parameter	Description
Instance Class	Refers to the vCPU and memory of a DB instance. Different instance classes support different numbers of database connections and maximum IOPS. After a DB instance is created, you can change its vCPU and memory. For details, see **Changing a DB Instance Class**. **NOTE** Only general-enhanced DB instances are allowed for a DCC. Select **4 vCPUs | 8 GB** or above for production environments. Smaller instance classes can only be used for test environments.
Resource Type	– **EVS** – **DSS** **NOTE** This option is displayed only when you have purchased Dedicated Distributed Storage Service (DSS).
Storage Pool	Displayed only when you select **DSS** for **Resource Type**. The storage pool is secure because it is physically isolated from other pools.
Storage Space (GB)	Contains the system overhead required for inodes, reserved blocks, 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. If the storage type is cloud SSD or extreme SSD, you can enable storage autoscaling. If the available storage drops to a specified threshold, autoscaling is triggered. If you specify a read replica when creating a primary DB instance and enable storage autoscaling for the primary DB instance, storage autoscaling is also enabled for the read replica by default. – **Enable autoscaling**: If you select this option, autoscaling is enabled. – **Trigger If Available Storage Drops To**: If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered. – **Autoscaling Limit**: The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance. After a DB instance is created, you can scale up its storage space. For details, see **Scaling up Storage Space**.

Parameter	Description
Disk Encryption	– **Disable**: indicates the encryption function is disabled. – **Enable**: indicates the encryption function is enabled, improving data security but affecting system performance. ▪ **Key Name**: indicates the tenant key. Select one from the drop-down list. ▪ To create a key, click **Create Key** and configure parameters in the displayed dialog box. For more information, see **Creating a Key** in the *Data Encryption Workshop User Guide*. **NOTE** ▪ If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. Disk encryption will also encrypt backup data stored in OBS. ▪ If disk encryption is enabled, keep the key properly. Once the key is disabled, deleted, or frozen, the database will be unavailable and data may not be restored. ▪ If a shared KMS key is used, the corresponding CTS events are **createdatakey** and **decrydatakey**. Only the key owner can receive the events.

- Basic Settings and Connectivity

Figure 3-4 Basic settings and connectivity

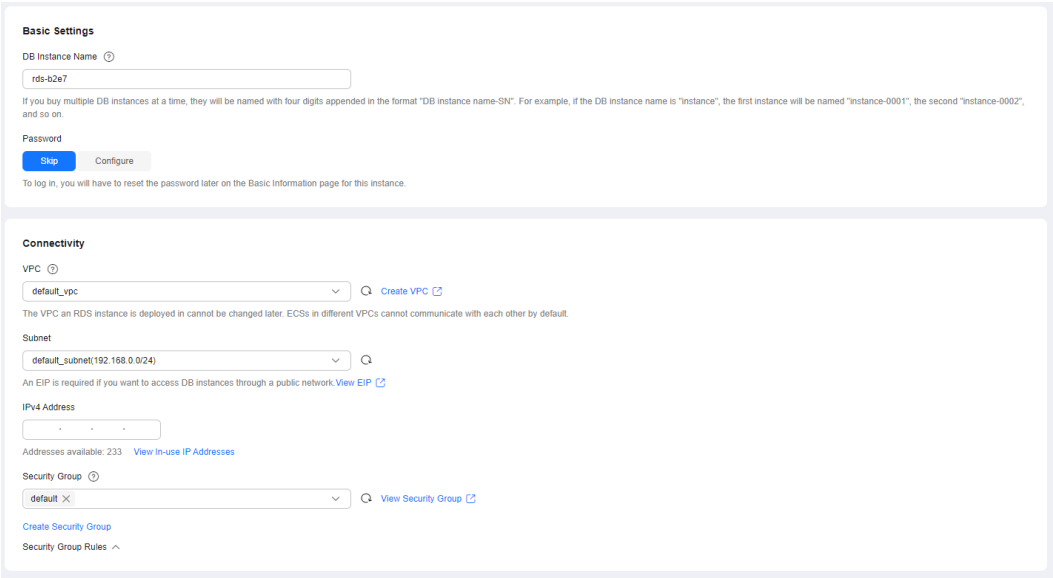


Table 3-3 Network

Parameter	Description
DB Instance Name	The instance name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed. – If you intend to buy multiple DB instances at a time, the allowed length for each instance name will change. – If you buy multiple instances at a time, a hyphen (-) followed by a number with four digits will be appended to the instance name, starting with -0001. For example, if you enter **instance**, the first instance will be named instance-0001, the second instance-0002, and so on.
Password	– **Configure** (default setting): Configure a password for your DB instance during the creation process. – **Skip**: Configure a password later after the DB instance is created. **NOTICE** If you select **Skip** for **Password**, you need to reset the password before you can log in to the instance. After a DB instance is created, you can reset the password. For details, see **Resetting the Administrator Password**.
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. If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password. Keep this password secure. The system cannot retrieve it. After a DB instance is created, you can reset this password. For details, see **Resetting the Administrator Password**.
Confirm Password	Must be the same as **Administrator Password**.

Parameter	Description
VPC	A virtual network in which your RDS DB instances are located. A VPC can isolate networks for different workloads. 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 *Virtual Private Cloud User Guide*. If no VPC is available, RDS allocates a VPC to you by default. To use a shared VPC, select a VPC that another account shares with the current account from the drop-down list. VPC owners can share the subnets in a VPC with one or multiple accounts through **Resource Access Manager (RAM)**. Through VPC sharing, you can easily configure, operate, and manage multiple accounts' resources at low costs. For more information about VPC and subnet sharing, see **VPC Sharing**. **NOTICE** After a 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. – IPv4 address: A floating IPv4 address is automatically assigned when you create a DB instance. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the DB instance is created, you can change the floating IP address. – IPv6 address: A DB instance assigned a floating IPv6 address will be created only when the vCPUs and memory you selected support IPv6 addresses. A floating IPv6 address is automatically assigned during instance creation and cannot be specified. After the DB instance is created, this floating IP address cannot be changed.

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. In addition, a network **access control list (ACL)** can help control inbound and outbound traffic of subnets in your VPC. 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. When creating a DB instance, you can select multiple security groups. For better network performance, you are advised to select no more than five security groups. In such a case, the access rules of all the selected security groups apply on the instance. To use multiple security groups, choose **Service Tickets > Create Service Ticket** in the upper right corner of the management console to apply for the required permissions. If no security group is available, RDS allocates a security group to you by default.

- Additional Options

Figure 3-5 Additional options

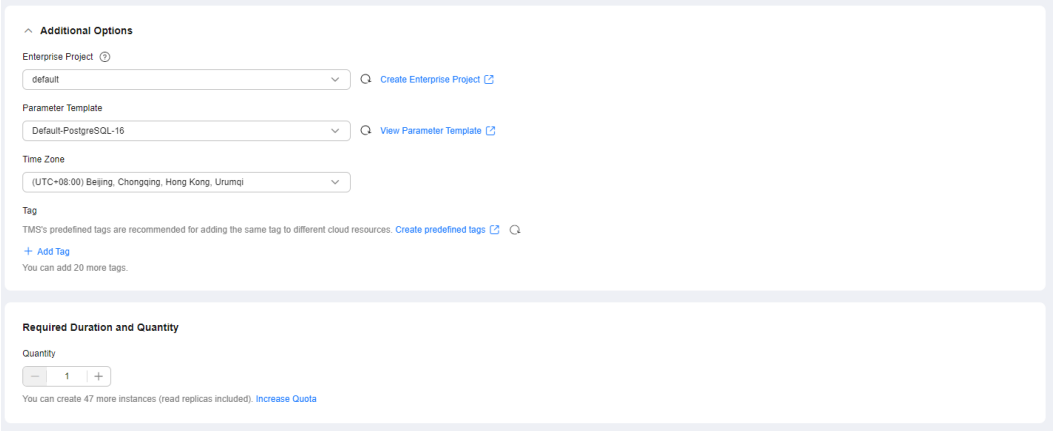


Table 3-4 Additional options

Parameter	Description
Enterprise Project	If your account has been associated with an enterprise project, select the target project from the **Enterprise Project** drop-down list. For more information about enterprise projects, see *Enterprise Management User Guide*.

Parameter	Description
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 section **Modifying Parameters in a Parameter Template**.
Time Zone	You need to select a time zone for your instance based on the region hosting your instance. You can select a time zone during instance creation and change it later as needed.
Tag	Tags an RDS DB instance. This parameter is optional. Adding tags to RDS DB instances helps you better identify and manage the DB instances. A maximum of 20 tags can be added for each DB instance. If your organization has configured tag policies for RDS, add tags to DB instances based on the policies. If a tag does not comply with the policies, DB instance creation may fail. Contact your organization administrator to learn more about tag policies. After a DB instance is created, you can view its tag details on the **Tags** page. For details, see **RDS for PostgreSQL Tags**.

- Required Duration and Quantity

Table 3-5 Required duration and quantity

Parameter	Description
Required Duration	This option is available only for yearly/monthly DB instances. The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	– This option is available only for yearly/monthly DB instances and is not selected by default. – If you select this option, the auto-renew cycle is determined by the selected required duration.

Parameter	Description
Quantity	RDS supports batch creation of DB instances. If you intend to create primary/standby DB instances and set **Quantity** to **1**, a primary DB instance and a synchronous standby DB instance will be created.

If you have any questions about the price, click **Pricing details** at the bottom of the page.

NOTE

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

Step 3 Confirm the specifications for pay-per-use DB instances.

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

Skip **Step 4** and **Step 5** and go to **Step 6**.

Step 4 Confirm the order for yearly/monthly DB instances.

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

Step 5 Select a payment method and complete the payment.

NOTE

This operation applies only to the yearly/monthly billing mode.

Step 6 To view and manage your DB instance, go to the **Instances** page.

- When your DB instance is being created, the status is **Creating**. The status changes to **Available** after the instance is created.
- The automated backup policy is enabled by default. You can change it after the DB instance is created. An automated full backup is immediately triggered once your DB instance is created.
- After a DB instance is created, you can enter a description for it.
- The default database port is **5432**. You can change it after a DB instance is created.

  NOTE

  You are advised to change the database port in a timely manner.

  For details, see **Changing a Database Port**.

----End

Related Operations

Creating a DB Instance Using an API

3.3 Instance Connection

3.3.1 Overview

Before connecting to a DB instance, you must create one first. For details about how to create a DB instance, see **Buying an RDS for PostgreSQL DB Instance**. You can connect to an RDS for PostgreSQL instance through a command-line interface (CLI), graphical user interface (GUI), Data Admin Service (DAS), or using Java database connectivity (JDBC).

Connecting to a DB Instance over a Private or Public Network Using CLI

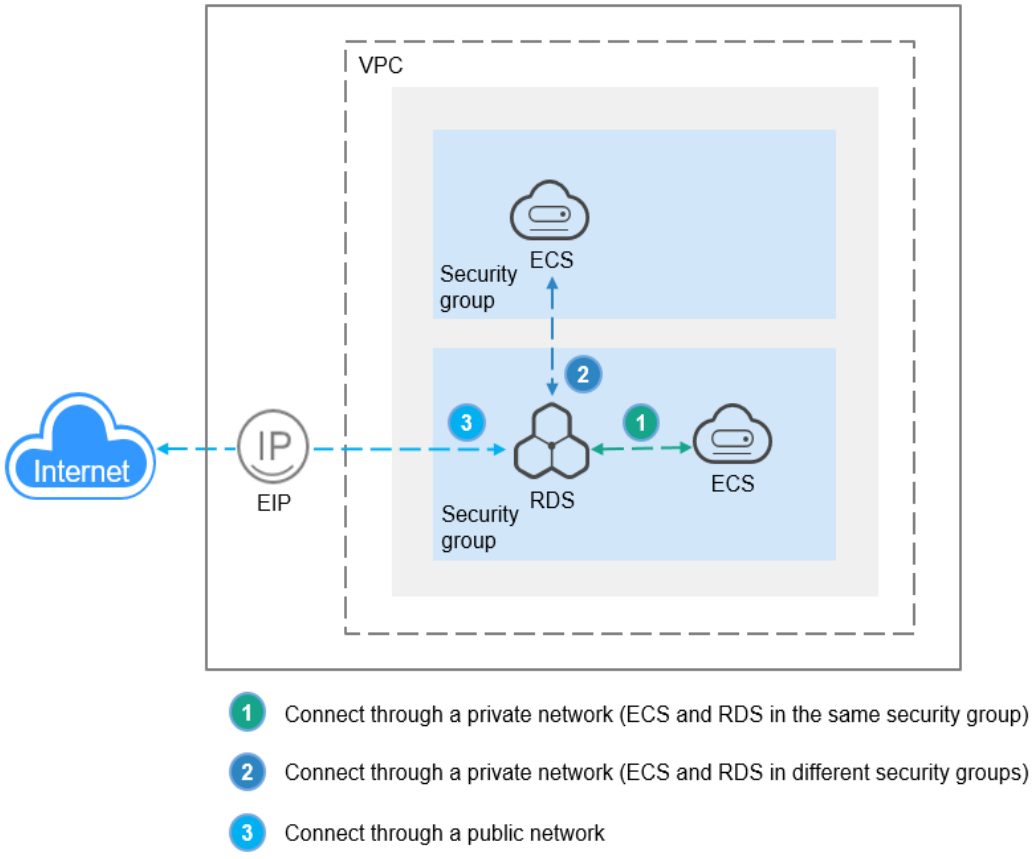
Table 3-6 lists how to use CLI to connect to an RDS for PostgreSQL instance over a private or public network.

Table 3-6 Connecting to a DB instance over a private or public network

Connection Method	IP Address	Security Group Rules	Description
Private network	Private IP address	• If the ECS and RDS DB instance are in the same security group, they can communicate with each other over a private network by default. No security group rules need to be configured. • If they are in different security groups, 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. For details, see **Configuring a Security Group Rule**. – ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security group 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.	• Secure and high-performance • Recommended

Connection Method	IP Address	Security Group Rules	Description
Public network	You need to purchase an EIP. For pricing details, see **EIP Billing**.	To access a DB instance from resources outside the security group that the DB instance is associated with, you need to configure an **inbound** rule for the security group. For details, see **Configuring a Security Group Rule**.	• Less secure • 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 private IP address to access the DB instance.

Figure 3-6 Connecting to a DB instance over a private or public network



Connection Methods

Table 3-7 Connection methods

Connection Method	Description
Connecting to an RDS for PostgreSQL Instance Through DAS (Recommended)	DAS enables you to manage databases on a web-based console and provides you with database development, O&M, and intelligent diagnosis to make it easy to use and maintain your databases. The permissions required for connecting to DB instances through DAS are enabled by default.
Connecting to an RDS for PostgreSQL Instance Through the psql CLI Client	In Linux, you need to install a PostgreSQL client on the ECS and connect to the instance through the psql CLI over a private or public network. • A private IP address is provided by default. When your applications are deployed on an ECS that is in the same region and VPC as the RDS for PostgreSQL instance, you are advised to use a floating IP address to connect to the instance through the ECS. • If you cannot access your RDS for PostgreSQL instance through a floating IP address, bind an EIP to the instance and connect to the instance through the EIP.
Connecting to an RDS for PostgreSQL Instance Through the GUI	In Windows, you can use the **pgAdmin** client to connect to an RDS for PostgreSQL instance.
Connecting to an RDS for PostgreSQL Instance Through JDBC	RDS for PostgreSQL is compatible with the community ecosystem and does not provide the driver service. You can select a community driver version as required. If you are connecting to an instance through JDBC, the SSL certificate is optional. For security reasons, you are advised to download the SSL certificate to encrypt the connection.

3.3.2 Logging In to an RDS for PostgreSQL Instance and Creating a Database Through DAS (Recommended)

Scenarios

Data Admin Service (DAS) enables you to connect to and manage DB instances with ease on a web-based console. The permissions required for connecting to DB instances through DAS are enabled by default. Using DAS to connect to your DB instance is recommended, which is more secure and convenient.

Step 1: Log In to an RDS for PostgreSQL Instance

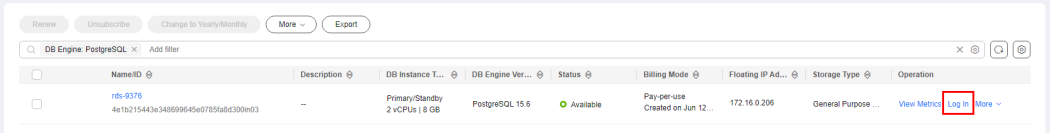
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

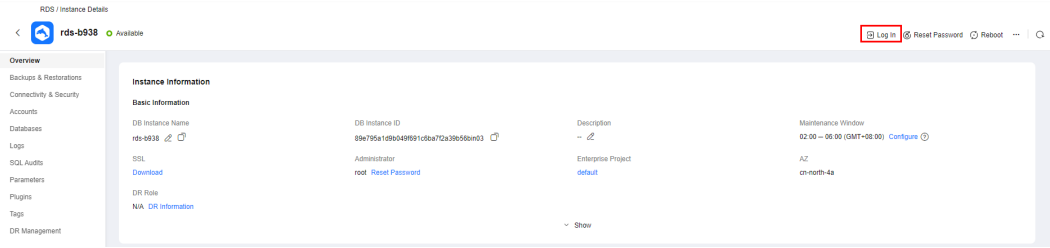
Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Figure 3-7 Logging in to an instance



Alternatively, click the DB instance name on the **Instances** page. On the displayed **Overview** page, click **Log In** in the upper right corner.

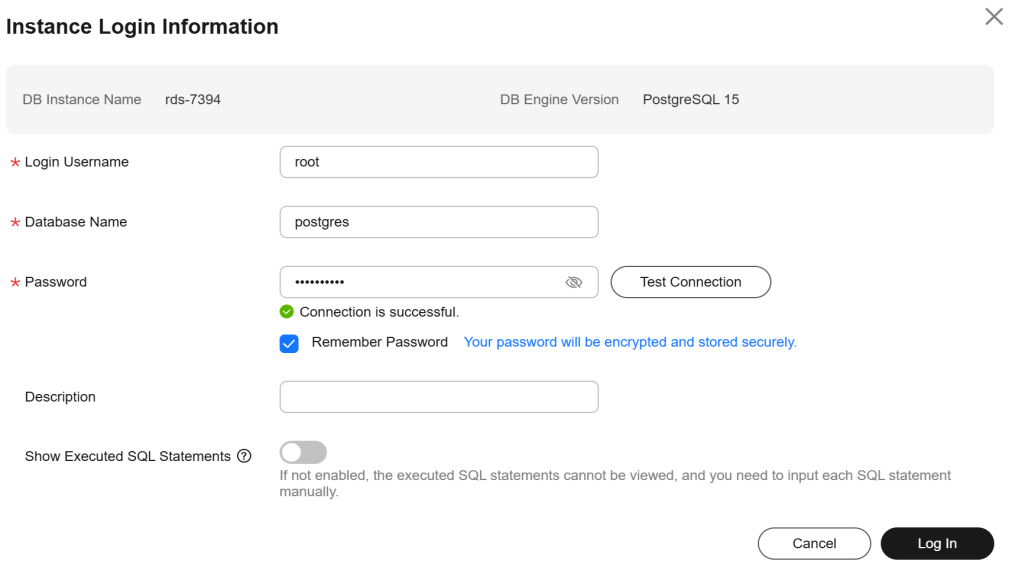
Figure 3-8 Logging in to an instance



Step 5 On the displayed login page, enter the username and password and click **Log In**.

- **Login Username**: Enter **root**.
- **Database Name**: Enter **postgres**.
- **Password**: Enter the **root** password you specified during instance creation. If you forget the password, you can reset it. For details, see **Resetting the Administrator Password to Restore Root Access**.

Figure 3-9 Logging in to an instance



----End

Step 2: Create a Database

Step 1 On the DAS console, choose **SQL Operations** > **SQL Query**.

Step 2 In the SQL window, run the following command to create a database named **db1**:

```
create database db1;
```

You can also specify a template database and set properties such as character sets, LC_COLLATE (character collation), and LC_CTYPE (character classification) for each database. For details, see **Syntax**.

----End

Syntax

```
CREATE DATABASE name
[ [ WITH ] [ OWNER [=] user_name ]
      [ TEMPLATE [=] template ]
      [ ENCODING [=] encoding ]
      [ LC_COLLATE [=] lc_collate ]
      [ LC_CTYPE [=] lc_ctype ]
      [ TABLESPACE [=] tablespace_name ]
      [ ALLOW_CONNECTIONS [=] allowconn ]
      [ CONNECTION LIMIT [=] connlimit ]
      [ IS_TEMPLATE [=] istemplate ] ]
```

- TEMPLATE

  RDS for PostgreSQL has two database templates: **template0** and **template1**. The default template is **template1**. When you use **template1** to create a database, do not specify a new character set for the database. Otherwise, an error will be reported. You can also specify a custom template to create a database.

- ENCODING

When creating a database, you can specify a character set using **WITH ENCODING**. For details about the supported character sets, see the **PostgreSQL community documentation**.

- LC_COLLATE

  String sort order. The default value is **en_US.utf8**.

  Comparison of the same string in different collations may have different results.

  For example, after you execute **SELECT 'a'>'A';**, the result is **false** if this parameter is set to **en_US.utf8** and the result is **true** if this parameter is set to **C**. If you need to migrate a database from Oracle to RDS for PostgreSQL, set **LC_COLLATE** to **C**. You can query the supported collations from the **pg_collation** table.

- LC_CTYPE

  It is used to classify if a character is a digit, uppercase letter, lowercase letter, and so on. You can query the supported character classifications from the **pg_collation** table.

- For details about other parameters, see the **PostgreSQL community documentation**.

Examples

- Using **TEMPLATE** to specify a database template
  - When template1 is used, the character set or collation defined in this template cannot be changed. For details about collations, see **Configuring the Collation of a Database in a Locale**.

    ```
    CREATE DATABASE my_db WITH TEMPLATE template1 ;
    ```

  - When template0 is used, you can change the character set and collation. For details, see **Configuring the Collation of a Database in a Locale**.

    ```
    CREATE DATABASE my_db WITH ENCODING = 'UTF8' LC_COLLATE ='zh_CN.utf8' LC_CTYPE ='zh_CN.utf8' TEMPLATE = template0 ;
    ```

  - If no template is specified during database creation, template1 is used by default. You can also specify a custom template to create a database.

    ```
    CREATE DATABASE my_db WITH TEMPLATE = mytemplate;
    ```

- Using **WITH ENCODING** to specify a character set

  ```
  CREATE DATABASE my_db WITH ENCODING 'UTF8';
  ```

- **LC_COLLATE** and **LC_CTYPE**
  - Querying character sets (encodings) supported by LC_COLLATE and LC_CTYPE

    ```
    SELECT pg_encoding_to_char(collencoding) AS encoding,collname,collcollate AS "LC_COLLATE",collctype AS "LC_CTYPE" FROM pg_collation;
    ```

    If **encoding** is empty, **LC_COLLATE** supports all character sets.

	encoding	collname	LC_COLLATE	LC_CTYPE
1		default		
2		C	C	C
3		POSIX	POSIX	POSIX
4	UTF8	ucs_basic	C	C
5	LATIN1	aa_DJ	aa_DJ	aa_DJ
6	LATIN1	aa_DJ.iso88591	aa_DJ.iso88591	aa_DJ.iso88591
7	UTF8	aa_DJ.utf8	aa_DJ.utf8	aa_DJ.utf8
8	UTF8	aa_ER	aa_ER	aa_ER

- Configuring the collation of a database in a locale

  Run the following command to create a database with **LC_COLLATE** and **LC_CTYPE** set to **zh_CN.utf8**:

  ```
  CREATE DATABASE my_db WITH ENCODING = 'UTF8' LC_COLLATE ='zh_CN.utf8' LC_CTYPE ='zh_CN.utf8' TEMPLATE = template0 ;
  ```

  If the specified **LC_COLLATE** is incompatible with the character set, error information similar to the following is displayed:

NOTE

1. The specified **LC_COLLATE** and **LC_CTYPE** must be compatible with the target character set. Otherwise, an error is reported. For details, see **Querying LC_COLLATE and LC_CTYPE Settings Supported by a Character Set**.
2. The **LC_COLLATE** and **LC_CTYPE** settings of an existing database cannot be changed by running the **ALTER DATABASE** statement. You can change them while creating a new database and then import your data to the new database.

FAQs

- **What Should I Do If I Can't Connect to My DB Instance Due to Insufficient Permissions?**
- **What Should I Do If I Can't Connect to My RDS for PostgreSQL Instance?**

What Can I Do If the DAS Console Is Not Displayed After I Click Log In in the Operation Column of an Instance on the Instances Page?

Solution: Set your browser to allow pop-ups and try again.

How Do I View the Created Databases and the Character Sets, LC_COLLATE, and LC_CTYPE Information of the Databases?

- To view the created databases, run the psql meta-command **\ l**.

```
postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 db3       | postgres | SQL_ASCII | en_US.UTF-8 | en_US.UTF-8 |
 mydb      | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |
 mydb1     | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |           |             |             | postgres=CTc/postgres
 template1 | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
           |          |           |             |             | postgres=CTc/postgres
```

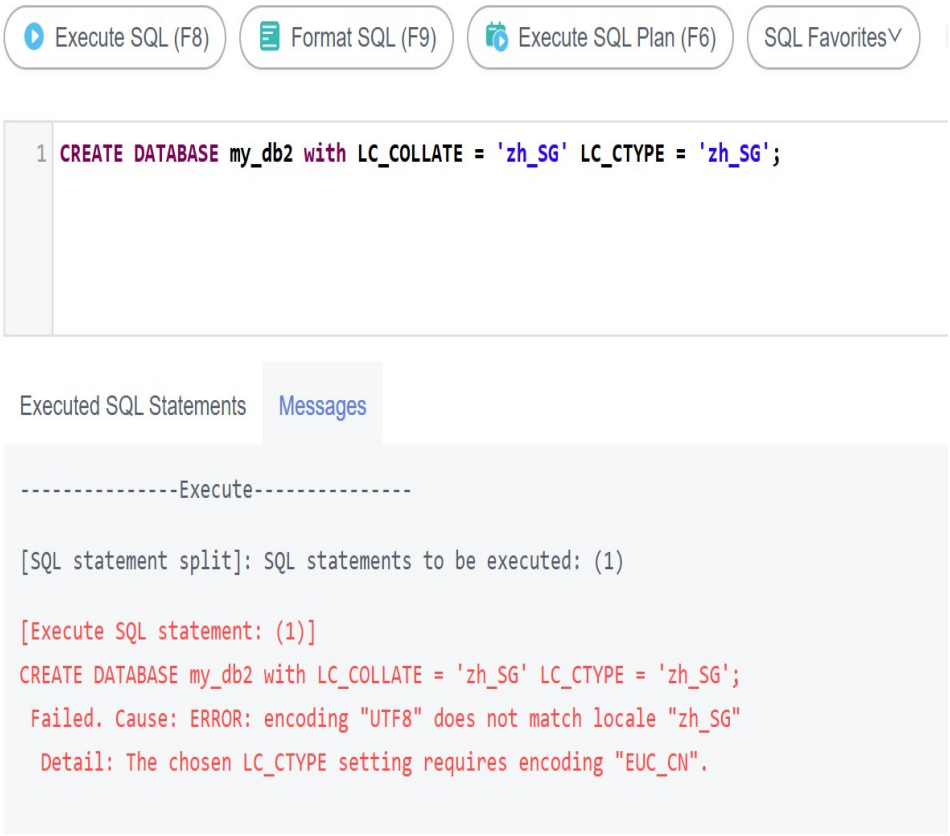
- To view the character sets, LC_COLLATE, and LC_CTYPE information of databases, query the **pg_database** system catalog.

```
postgres=# select datname,pg_encoding_to_char(encoding),datcollate,datctype from pg_database;
  datname  | pg_encoding_to_char | datcollate  |  datctype
-----------+---------------------+-------------+-------------
 postgres  | UTF8                | en_US.UTF-8 | en_US.UTF-8
 test      | UTF8                | en_US.UTF-8 | en_US.UTF-8
 template1 | UTF8                | en_US.UTF-8 | en_US.UTF-8
 template0 | UTF8                | en_US.UTF-8 | en_US.UTF-8
 db3       | SQL_ASCII           | en_US.UTF-8 | en_US.UTF-8
 mydb      | UTF8                | en_US.UTF-8 | en_US.UTF-8
 mydb1     | UTF8                | en_US.UTF-8 | en_US.UTF-8
 testdb1   | SQL_ASCII           | en_US.UTF-8 | en_US.UTF-8
(8 rows)
```

What Do I Do If the Character Set Does Not Match the Locale During Database Creation?

If the **LC_COLLATE** you specified does not match the character set during database creation, the following error message is displayed:

```
CREATE DATABASE my_db2 WITH LC_COLLATE ='zh_SG' LC_CTYPE ='zh_SG' ;
```

Solution:

1. Query the character set supported by the template database. For details, see **How Do I View the Created Databases and the Character Sets, LC_COLLATE, and LC_CTYPE Information of the Databases?**. The default template database is **template1**.
2. Query the **LC_COLLATE** value supported by the character set. For details, see **Examples**.
3. Change the value of **LC_COLLATE** to match the character set and create the database again.

Follow-up Operations

After logging in to the DB instance, you can create or migrate your databases.

- **Creating a PostgreSQL Database Using an API**
- **Managing PostgreSQL Databases Using DAS**
- **Migration Solution Overview**

3.3.3 Connecting to an RDS for PostgreSQL Instance Through the psql CLI Client

3.3.3.1 Connecting to a DB Instance from a Linux ECS over a Private Network

You can connect to your DB instance using a Linux ECS with a PostgreSQL client installed over a private network.

You can use the PostgreSQL client psql to connect to your DB instance over a Secure Sockets Layer (SSL) connection. SSL encrypts connections to your DB instance, making in-transit data more secure.

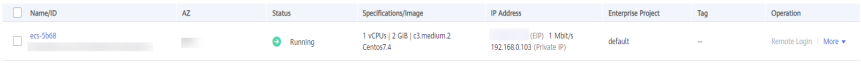
SSL is enabled by default when you create an RDS for PostgreSQL DB instance and cannot be disabled after the instance is created.

Enabling SSL reduces the read-only and read/write performance of your instance by about 20%.

Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Linux ECS, go to **3**.
   - If no Linux ECS is available, go to **2**.

   Figure 3-10 ECS

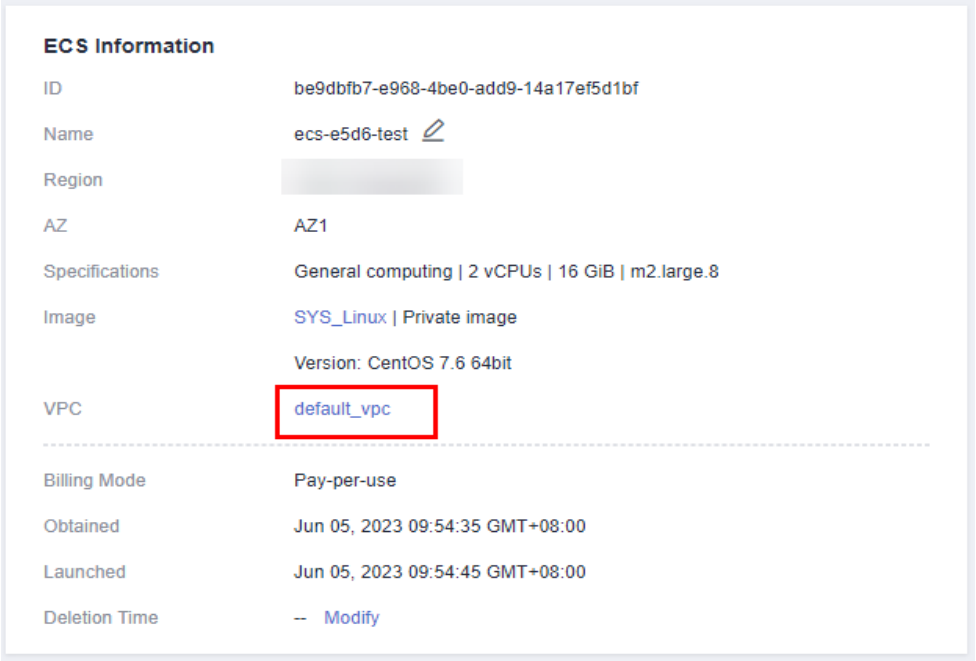
2. Buy an ECS and select Linux (for example, CentOS) as its OS.

   To download a PostgreSQL client to the ECS, bind an EIP to the ECS. The ECS must be in the same region, VPC, and security group as the RDS for PostgreSQL DB instance for mutual communications.

   For details about how to purchase a Linux ECS, see **Purchasing a Custom ECS** in *Elastic Cloud Server User Guide*.

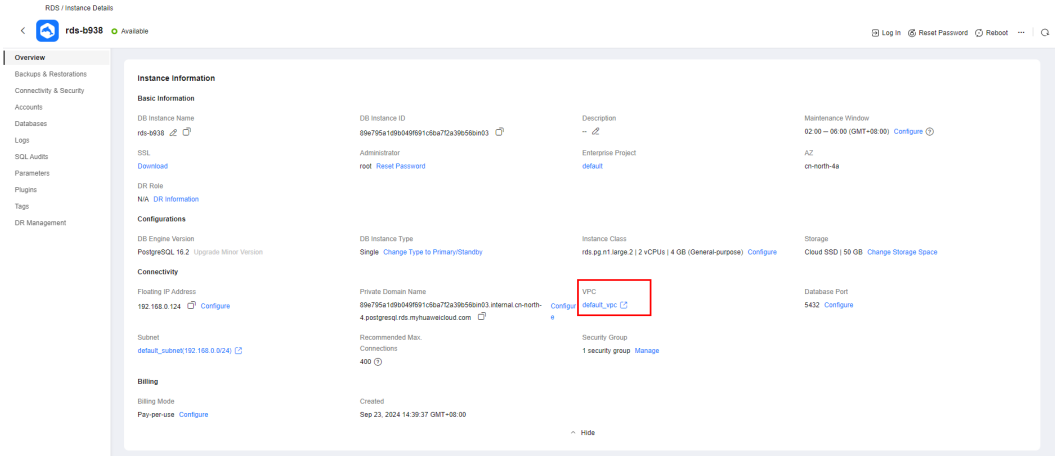
3. On the **ECS Information** page, view the region and VPC of the ECS.

Figure 3-11 ECS information



4. On the **Overview** page of the RDS for PostgreSQL instance, view the region and VPC of the DB instance.

Figure 3-12 Overview



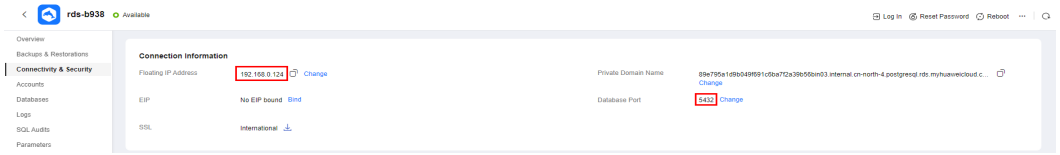
5. Check whether the ECS and RDS for PostgreSQL instance are in the same region and VPC.
   - If yes, go to **Installing the PostgreSQL Client (PostgreSQL 15 and Earlier)**.
   - If they are not in the same region, purchase another ECS or DB instance. The ECS and DB instance in different regions cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.
   - If the ECS and DB instance are in different VPCs, change the VPC of the ECS to that of the DB instance. For details, see **Changing a VPC**.

Step 2: Test Connectivity and Install the PostgreSQL Client

Installing the PostgreSQL Client (PostgreSQL 15 and Earlier)

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name.
3. Choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, obtain the floating IP address and database port of the DB instance.

   Figure 3-13 Connection information

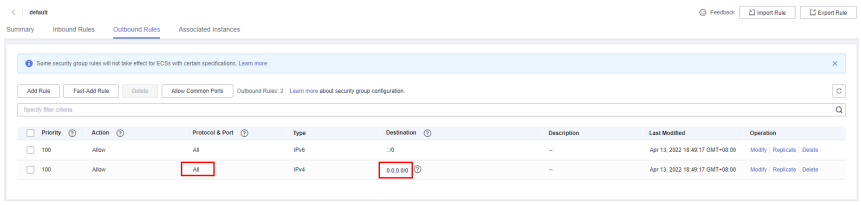
   

4. On the ECS, check whether the floating IP address and database port of the DB instance can be connected.

   ```
   curl -kv 192.168.0.7:5432
   ```

   - If yes, network connectivity is normal.
   - If no, check the security group rules.
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the floating IP address and port of the DB instance.

       Figure 3-14 ECS security group

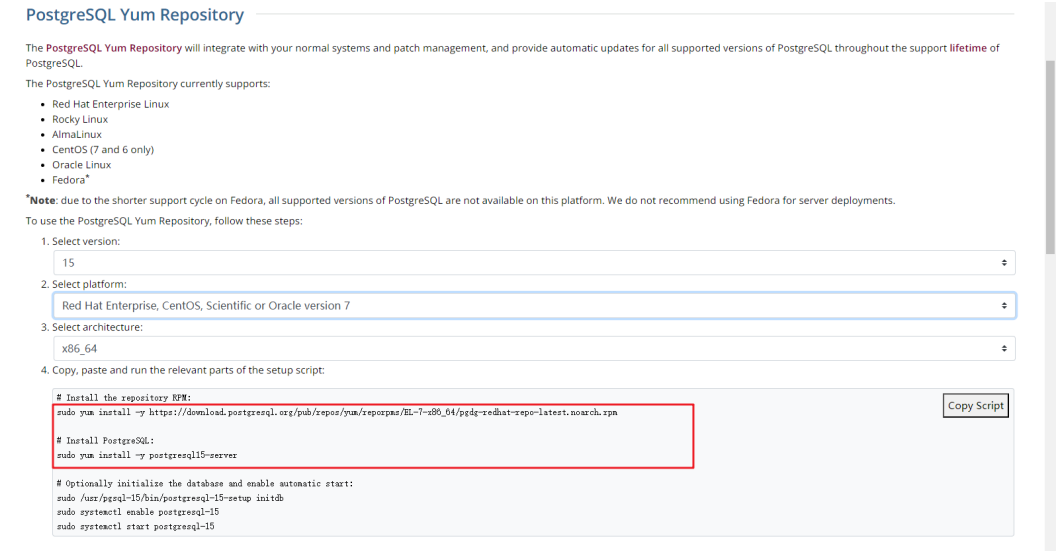
       

     - If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS.
5. Install the PostgreSQL client.

   The PostgreSQL community provides **client installation methods** for different OSs. You can download and install the client using the installation tool of the OS. This installation method is simple but can be used only for the OSs supported by the PostgreSQL community.

   In this example, CentOS 7 is used. Use the default installation tool of the OS to install the client (PostgreSQL 15 or earlier).

Figure 3-15 Obtaining the installation tool



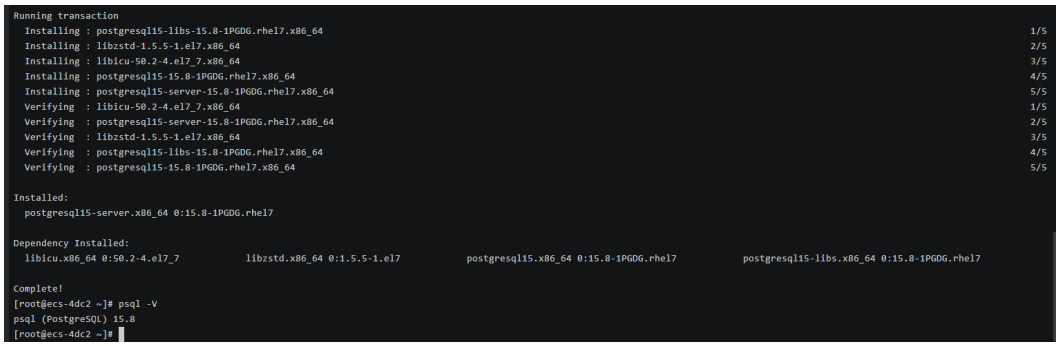
Run the following commands:

```
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 postgresql15-server
```

Check whether the installation is successful.

```
psql -V
```

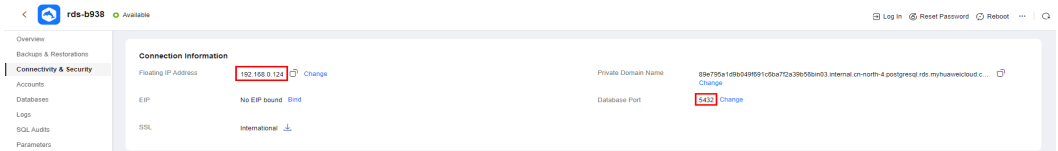
Figure 3-16 Successful installation



Installing the PostgreSQL Client (No Restrictions on PostgreSQL Versions)

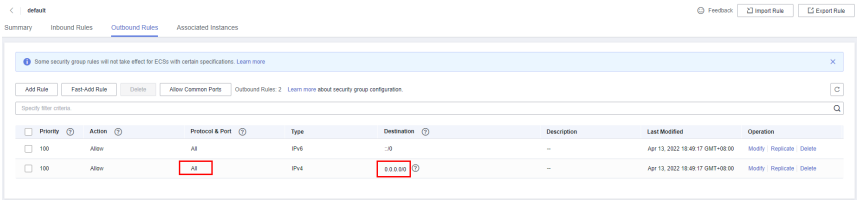
1. Log in to the ECS. For details, see **Login Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name.
3. Choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, obtain the floating IP address and database port of the DB instance.

Figure 3-17 Connection information



4. On the ECS, check whether the floating IP address and database port of the DB instance can be connected.
   ```
   curl -kv 192.168.0.7:5432
   ```
   - If yes, network connectivity is normal.
   - If no, check the security group rules.
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the floating IP address and port of the DB instance.

       Figure 3-18 ECS security group

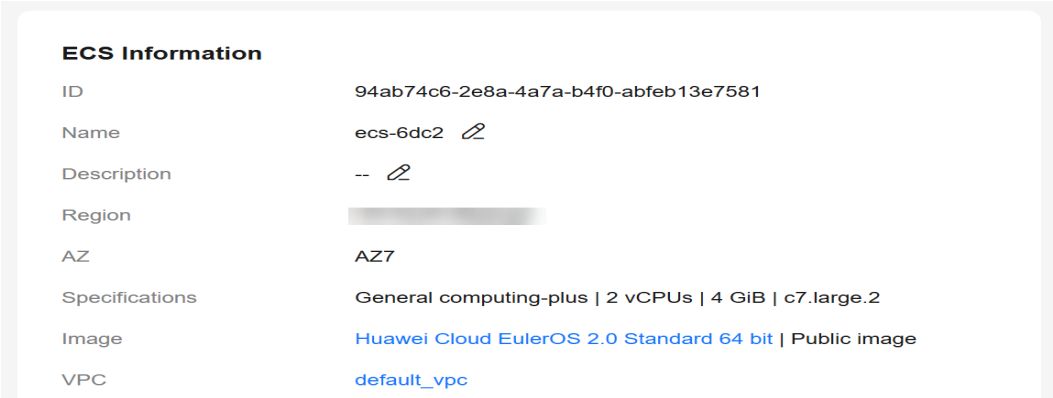
       

     - If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS.
5. Install the PostgreSQL client.

   Installation from source code: This installation method has no restrictions on RDS for PostgreSQL instance versions and ECS OS types.

   The following uses an ECS using the Huawei Cloud EulerOS 2.0 image as an example to describe how to install a PostgreSQL 16.4 client.

   Figure 3-19 Checking the ECS image

   a. To use SSL, download OpenSSL to the ECS in advance.
      ```
      sudo yum install -y openssl-devel
      ```
   b. Obtain the **code download link**, run **wget** to download the installation package to the ECS, or download the installation package to the local PC and upload it to the ECS.
      ```
      wget https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.gz
      ```
   c. Decompress the installation package.
      ```
      tar xf postgresql-16.4.tar.gz
      ```


d. Compile the source code and then install the client.

```
cd postgresql-16.4
./configure --without-icu --without-readline --without-zlib --with-openssl
make -j 8 && make install
```

NOTE

If **--prefix** is not specified, the default path is **/usr/local/pgsql**. The client can be installed in the simplest way.

Figure 3-20 Compilation and installation

```
make -C ../../../src/common all
make[4]: Entering directory '/root/postgresql-16.4/src/common'
make[4]: Nothing to be done for 'all'.
make[4]: Leaving directory '/root/postgresql-16.4/src/common'
make[3]: Leaving directory '/root/postgresql-16.4/src/interfaces/libpq'
make -C ../../../src/port all
make[3]: Entering directory '/root/postgresql-16.4/src/port'
make[3]: Nothing to be done for 'all'.
make[3]: Leaving directory '/root/postgresql-16.4/src/port'
make -C ../../../src/common all
make[3]: Entering directory '/root/postgresql-16.4/src/common'
make[3]: Nothing to be done for 'all'.
make[3]: Leaving directory '/root/postgresql-16.4/src/common'
/usr/bin/mkdir -p '/usr/local/pgsql/lib/pgxs/src/test/isolation'
/usr/bin/install -c  pg_isolation_regress '/usr/local/pgsql/lib/pgxs/src/test/isolation/pg_isolation_regress'
/usr/bin/install -c  isolationtester '/usr/local/pgsql/lib/pgxs/src/test/isolation/isolationtester'
make[2]: Leaving directory '/root/postgresql-16.4/src/test/isolation'
make -C test/perl install
make[2]: Entering directory '/root/postgresql-16.4/src/test/perl'
make[2]: Nothing to be done for 'install'.
make[2]: Leaving directory '/root/postgresql-16.4/src/test/perl'
/usr/bin/mkdir -p '/usr/local/pgsql/lib/pgxs/src'
/usr/bin/install -c -m 644 Makefile.global '/usr/local/pgsql/lib/pgxs/src/Makefile.global'
/usr/bin/install -c -m 644 Makefile.port '/usr/local/pgsql/lib/pgxs/src/Makefile.port'
/usr/bin/install -c -m 644 ./Makefile.shlib '/usr/local/pgsql/lib/pgxs/src/Makefile.shlib'
/usr/bin/install -c -m 644 ./nls-global.mk '/usr/local/pgsql/lib/pgxs/src/nls-global.mk'
make[1]: Leaving directory '/root/postgresql-16.4/src'
make -C config install
make[1]: Entering directory '/root/postgresql-16.4/config'
/usr/bin/mkdir -p '/usr/local/pgsql/lib/pgxs/config'
/usr/bin/install -c -m 755 ./install-sh '/usr/local/pgsql/lib/pgxs/config/install-sh'
/usr/bin/install -c -m 755 ./missing '/usr/local/pgsql/lib/pgxs/config/missing'
make[1]: Leaving directory '/root/postgresql-16.4/config'
```

e. Add the following code to the **/etc/profile** file to configure environment variables:

```
export PATH=/usr/local/pgsql/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/pgsql/lib:$LD_LIBRARY_PATH
source /etc/profile
```

f. Test whether the psql is available.

```
psql -V
```

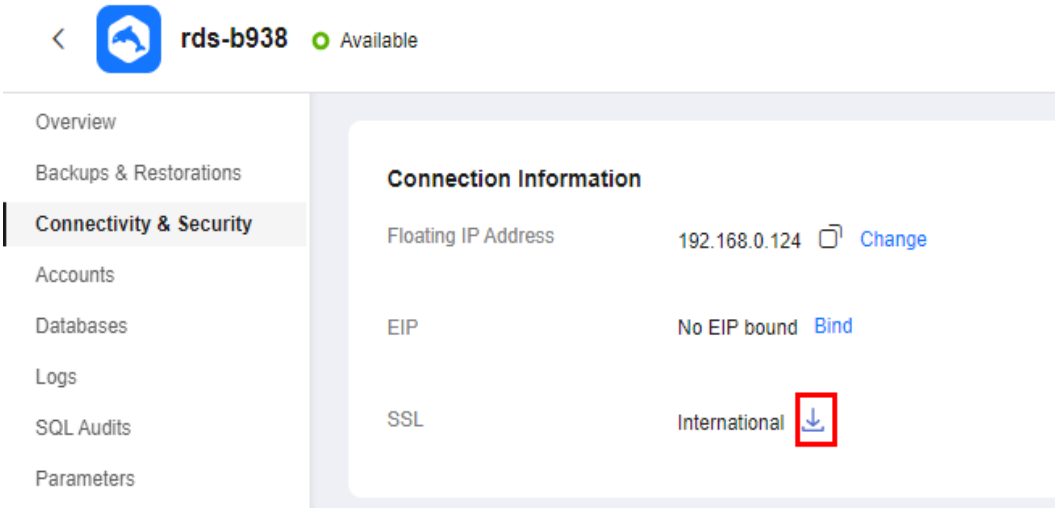
Figure 3-21 Testing psql

```
                . /etc/bashrc
        fi
fi
export PATH=/usr/local/pgsql/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/pgsql/lib:$LD_LIBRARY_PATH
[root@ecs-88a7 pgsql]# source /etc/profile
[root@ecs-88a7 pgsql]# psql -V
psql (PostgreSQL) 16.4
[root@ecs-88a7 pgsql]#
```


Step 3: Connect to the DB Instance Using Commands (SSL Connection)

1. On the **Instances** page, click the DB instance name.
2. In the navigation pane, choose **Connectivity & Security**.
3. In the **Connection Information** area, click next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Figure 3-22 Downloading a certificate



4. Upload **ca.pem** to the ECS.

   NOTE

   - TLS v1.2 or later is recommended. Versions earlier than TLS v1.2 have security risks.
   - The recommended protocol algorithm is EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EDH+aRSA+AESGCM:EDH+aDSS+AESGCM:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!SRP:!RC4. Using other options have security risks.
   - **ca-bundle.pem** contains both the new certificate provided as of April 2017 and the old certificate.
   - Both **ca.pem** and **ca-bundle.pem** can be used for SSL connections because **ca-bundle.pem** contains **ca.pem**.

5. Run the following command on the ECS to connect to the DB instance:

   psql --no-readline -h *<host>* **-p** *<port>* **"dbname=***<database>* **user=***<user>* **sslmode=verify-ca sslrootcert=***<ca-file-directory>***"**

   Example:

   psql --no-readline -h 192.168.0.7 -p 5432 "dbname=postgres user=root sslmode=verify-ca sslrootcert=/root/ca.pem"

Table 3-8 Parameter description

Parameter	Description
<host>	Floating IP address obtained in **3**.
<port>	Database port obtained in **3**. The default value is **5432**.

Parameter	Description
<database>	Name of the database to be connected. The default database name is **postgres**.
<user>	Administrator account **root**.
<ca-file-directory>	Directory of the CA certificate used for the SSL connection. This certificate should be stored in the directory where the command is executed.
sslmode	SSL connection mode. Set it to **verify-ca** to use a CA to check whether the service is trusted.

6. Enter the password of the database account as prompted.
   ```
   Password:
   ```
   If the following information is displayed, the connection is successful.
   ```
   SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
   ```

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Creating a PostgreSQL Database Using an API**
- **Managing PostgreSQL Databases Using DAS**
- **Migration Solution Overview**

3.3.3.2 Connecting to a DB Instance from a Linux ECS over a Public Network

You can connect to your DB instance using a Linux ECS installed with a PostgreSQL client over a public network.

You can use the PostgreSQL client psql to connect to your DB instance over a Secure Sockets Layer (SSL) connection. SSL encrypts connections to your DB instance, making in-transit data more secure.

SSL is enabled by default when you create an RDS for PostgreSQL DB instance and cannot be disabled after the instance is created.

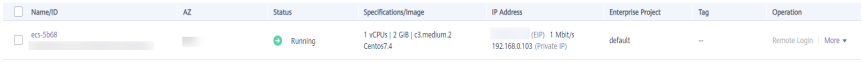
Enabling SSL reduces the read-only and read/write performance of your instance by about 20%.

You can also access your DB instance through Network Address Translation (NAT). If you have configured both NAT and EIP, the EIP is preferentially used.

Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Linux ECS, go to **3**.
   - If no Linux ECS is available, go to **2**.

Figure 3-23 ECS



2. Buy an ECS and select Linux (for example, CentOS) as its OS.

   To download a PostgreSQL client to the ECS, bind an EIP to the ECS.

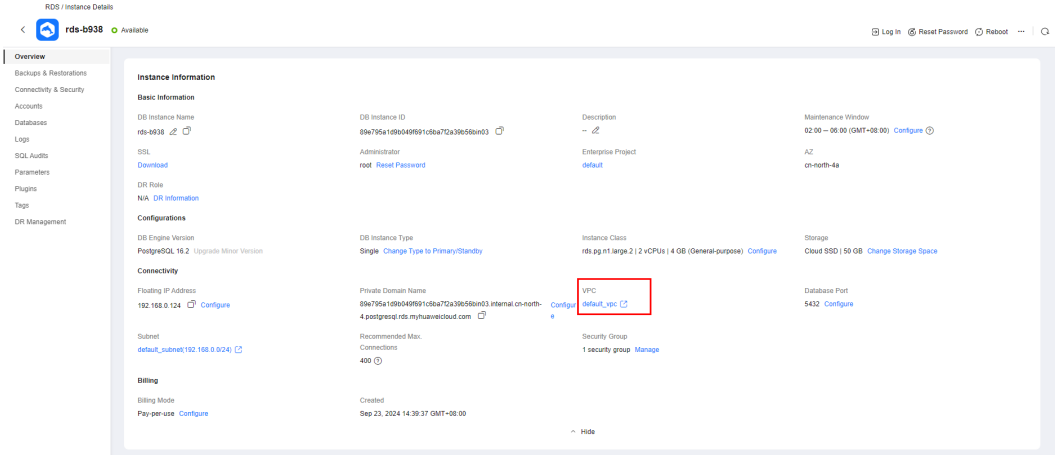
   For details about how to purchase a Linux ECS, see **Purchasing a Custom ECS** in *Elastic Cloud Server User Guide*.

3. On the **ECS Information** page, view the region and VPC of the ECS.

Figure 3-24 ECS information

4. On the **Overview** page of the RDS for PostgreSQL instance, view the region and VPC of the DB instance.

Figure 3-25 Overview

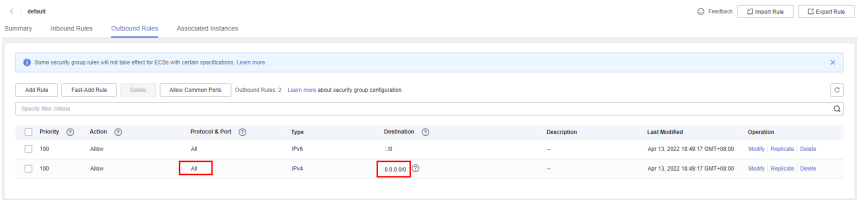


Step 2: Test Connectivity and Install the PostgreSQL Client

Installing the PostgreSQL Client (PostgreSQL 15 and Earlier)

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name.
3. Choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, obtain the EIP and database port of the DB instance.
4. On the ECS, check whether the EIP and database port of the DB instance can be connected.
   ```
   curl -kv EIP:5432
   ```
   - If yes, network connectivity is normal.
   - If no, check the security group rules.
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the EIP and port of the DB instance.

       Figure 3-26 ECS security group

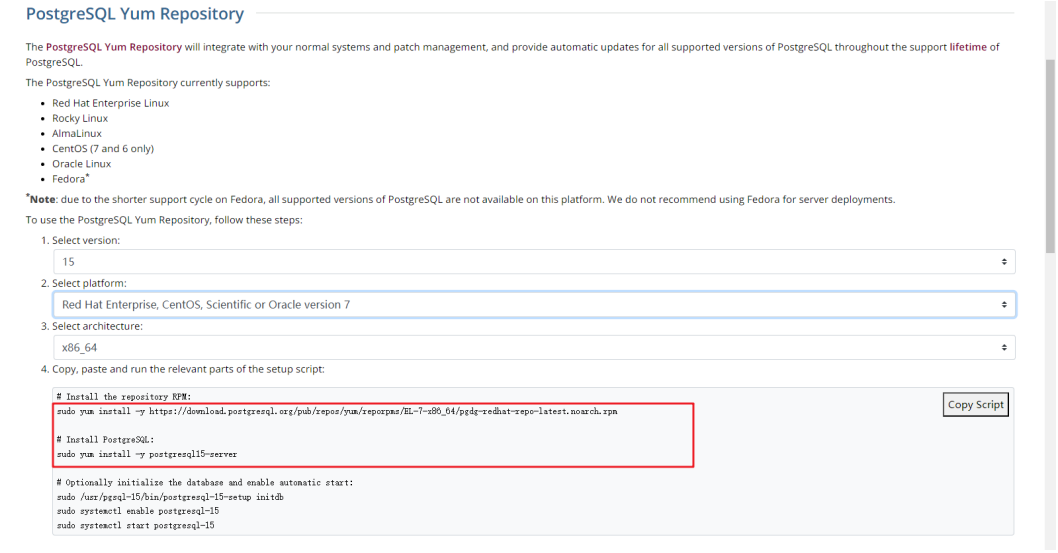
       

     - If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS.
5. Install the PostgreSQL client.

   The PostgreSQL community provides **client installation methods** for different OSs. You can download and install the client using the installation tool of the OS. This installation method is simple but can be used only for the OSs supported by the PostgreSQL community.

   In this example, CentOS 7 is used. Use the default installation tool of the OS to install the client (PostgreSQL 15 or earlier).

Figure 3-27 Obtaining the installation tool



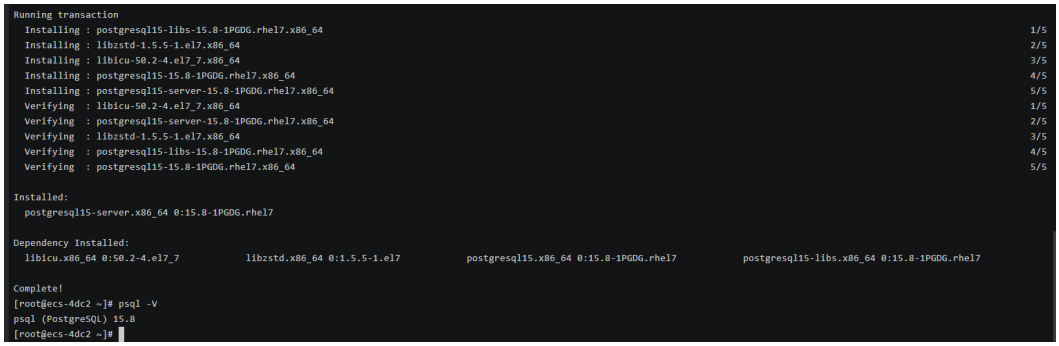
Run the following commands:

```
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 postgresql15-server
```

Check whether the installation is successful.

```
psql -V
```

Figure 3-28 Successful installation

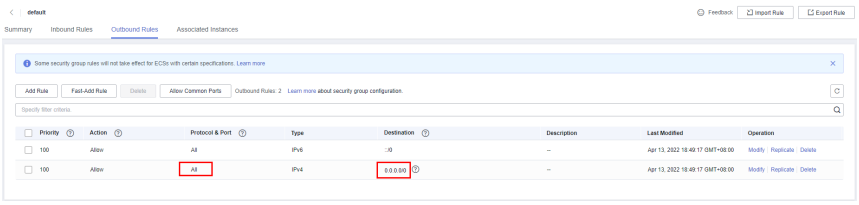


Installing the PostgreSQL Client (No Restrictions on PostgreSQL Versions)

1. Log in to the ECS. For details, see **Logging In to a Linux ECS Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name.
3. Choose **Connectivity & Security** from the navigation pane. In the **Connection Information** area, obtain the EIP and database port of the DB instance.
4. On the ECS, check whether the EIP and database port of the DB instance can be connected.
   curl -kv *EIP:5432*
   - If yes, network connectivity is normal.
   - If no, check the security group rules.

- If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the EIP and port of the DB instance.

  Figure 3-29 ECS security group

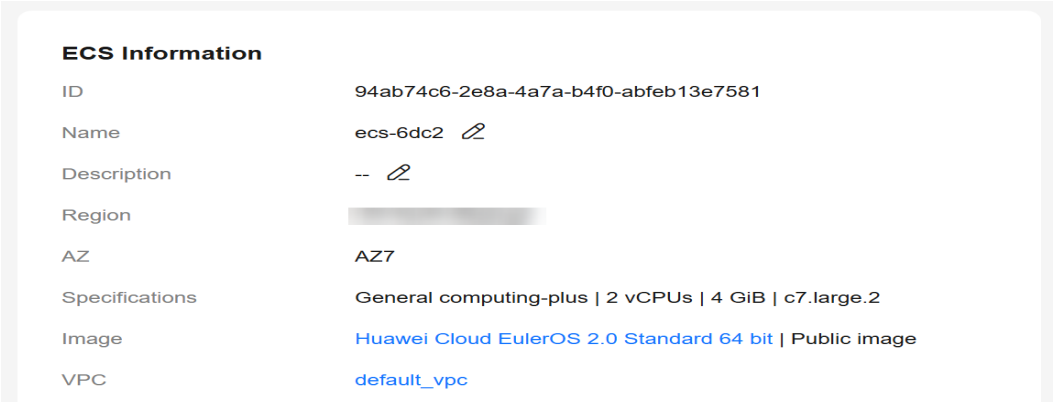
- If in the security group of the DB instance, there is no inbound rule allowing the access from the private IP address and port of the ECS, add an inbound rule for the private IP address and port of the ECS.

5. Install the PostgreSQL client.

   Installation from source code: This installation method has no restrictions on RDS for PostgreSQL instance versions and ECS OS types.

   The following uses an ECS using the Huawei Cloud EulerOS 2.0 image as an example to describe how to install a PostgreSQL 16.4 client.

   Figure 3-30 Checking the ECS image

   a. To use SSL, download OpenSSL to the ECS in advance.
   ```
   sudo yum install -y openssl-devel
   ```

   b. Obtain the **code download link**, run **wget** to download the installation package to the ECS, or download the installation package to the local PC and upload it to the ECS.
   ```
   wget https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.gz
   ```

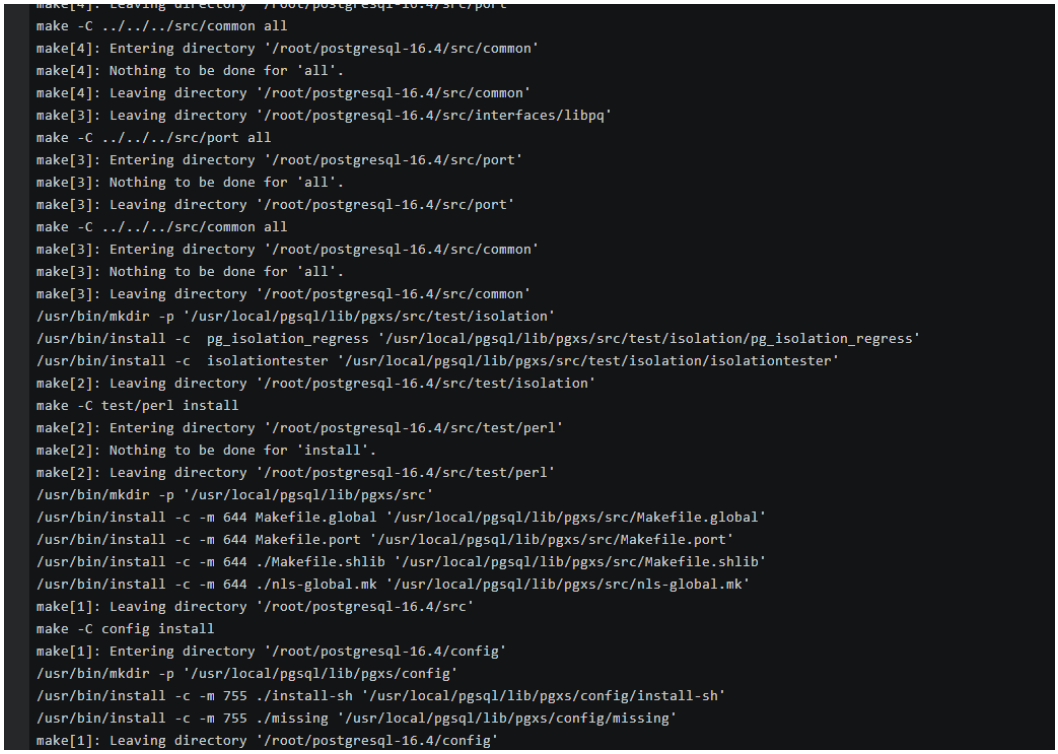
   c. Decompress the installation package.
   ```
   tar xf postgresql-16.4.tar.gz
   ```

   d. Compile the source code and then install the client.
   ```
   cd postgresql-16.4
   ./configure --without-icu --without-readline --without-zlib --with-openssl
   make -j 8 && make install
   ```

   NOTE

   If **--prefix** is not specified, the default path is **/usr/local/pgsql**. The client can be installed in the simplest way.

Figure 3-31 Compilation and installation



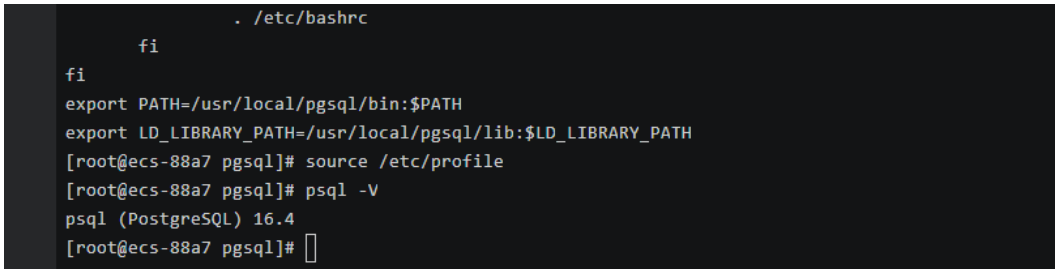
e. Add the following code to the **/etc/profile** file to configure environment variables:

```
export PATH=/usr/local/pgsql/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/pgsql/lib:$LD_LIBRARY_PATH
source /etc/profile
```

f. Test whether the psql is available.

```
psql -V
```

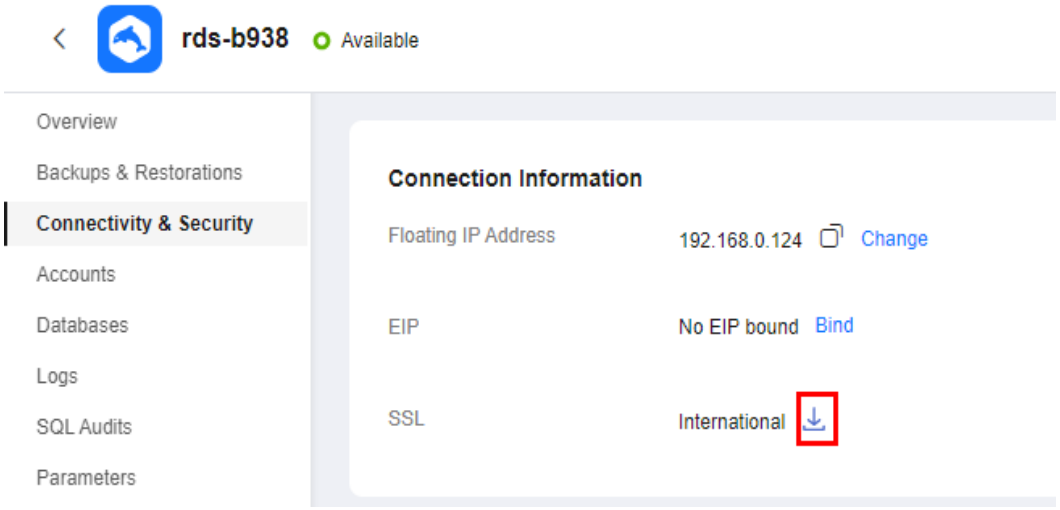
Figure 3-32 Testing psql



Step 3: Connect to the DB Instance Using Commands (SSL Connection)

1. On the **Instances** page, click the DB instance name.
2. In the navigation pane, choose **Connectivity & Security**.
3. In the **Connection Information** area, click next to the **SSL** field to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Figure 3-33 Downloading a certificate



4. Upload **ca.pem** to the ECS.

   NOTE

   - TLS v1.2 or later is recommended. Versions earlier than TLS v1.2 have security risks.
   - The recommended protocol algorithm is EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EDH+aRSA+AESGCM:EDH+aDSS+AESGCM:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!SRP:!RC4. Using other options have security risks.
   - **ca-bundle.pem** contains both the new certificate provided as of April 2017 and the old certificate.
   - Both **ca.pem** and **ca-bundle.pem** can be used for SSL connections because **ca-bundle.pem** contains **ca.pem**.

5. Run the following command on the ECS to connect to the DB instance:

   psql --no-readline -h *<host>* **-p** *<port>* **"dbname=***<database>* **user=***<user>* **sslmode=verify-ca sslrootcert=***<ca-file-directory>***"**

   Example:

   psql --no-readline -h 192.168.0.44 -p 5432 "dbname=postgres user=root sslmode=verify-ca sslrootcert=/root/ca.pem"

Table 3-9 Parameter description

Parameter	Description
<host>	EIP obtained in 3.
<port>	Database port obtained in 3. The default value is **5432**.
<database>	Name of the database to be connected. The default database name is **postgres**.
<user>	Administrator account **root**.
<ca-file-directory>	Directory of the CA certificate used for the SSL connection. This certificate should be stored in the directory where the command is executed.

Parameter	Description
sslmode	SSL connection mode. Set it to **verify-ca** to use a CA to check whether the service is trusted.

6. Enter the password of the database account as prompted.
   ```
   Password:
   ```
   If the following information is displayed, the connection is successful.
   ```
   SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
   ```

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Creating a PostgreSQL Database Using an API**
- **Managing PostgreSQL Databases Using DAS**
- **Migration Solution Overview**

3.3.4 Connecting to an RDS for PostgreSQL Instance Using pgAdmin

pgAdmin is an administration and development tool for PostgreSQL. Using pgAdmin, you can connect to specific databases from your clients, create tables, and run simple and complex SQL statements. pgAdmin can be used on Windows, Linux, macOS, and other operating systems. The latest version of pgAdmin is based on the browser/server (B/S) architecture. For more information, see the **pgAdmin documentation**.

This section uses pgAdmin 4-4.17 as an example to describe how to use pgAdmin to connect to an RDS for PostgreSQL instance and create databases and tables.

NOTICE

The pgAdmin version must be 4 or later.

Procedure

Step 1 Obtain the pgAdmin installation package.

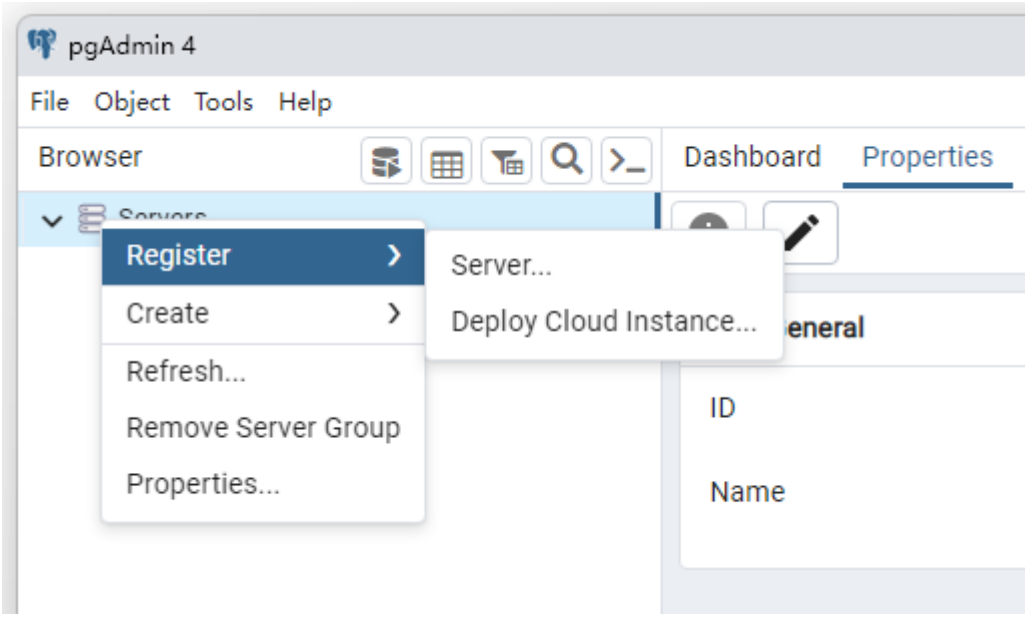
Download the pgAdmin installation package for Windows from the **pgAdmin official website**.

Step 2 Double-click the installation package and complete the installation as instructed.

Step 3 Start the pgAdmin client after the installation.

Step 4 In the displayed login window, right-click **Servers** and choose **Register** > **Server** from the shortcut menu.

Figure 3-34 Login information



Step 5 On the **General** page, specify **Name**. On the **Connection** page, specify information about the DB instance to be connected. Then, click **Save**.

Figure 3-35 Entering basic information

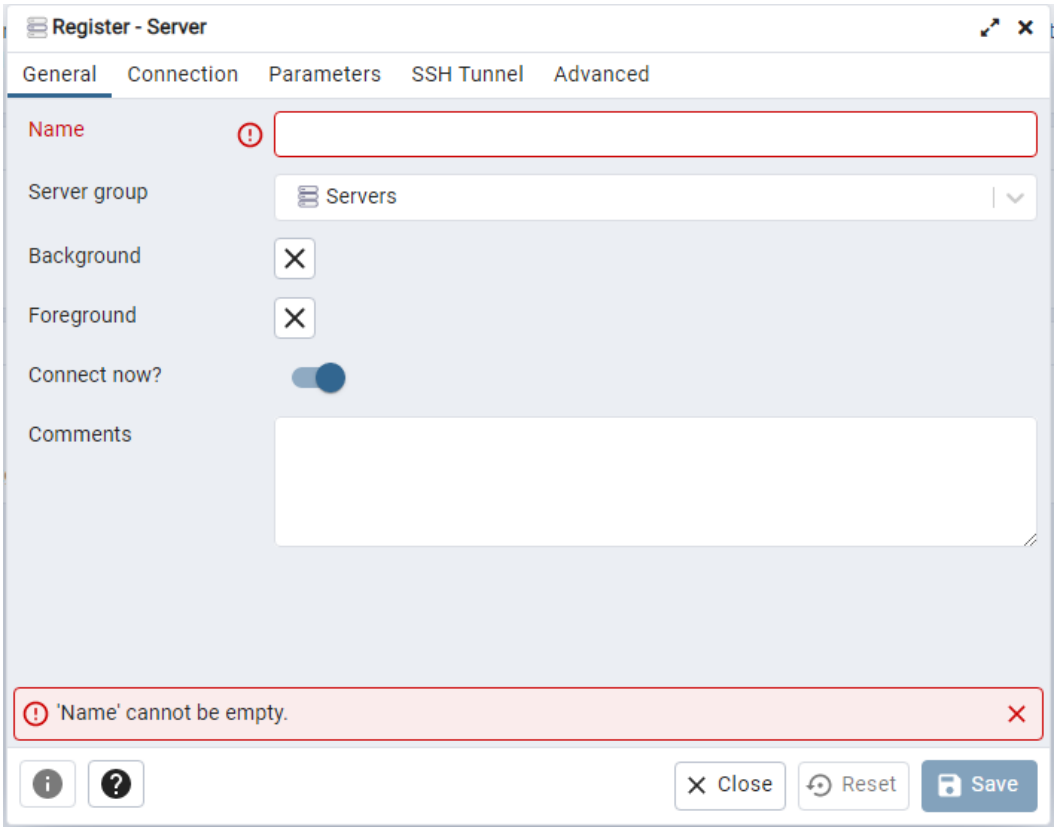
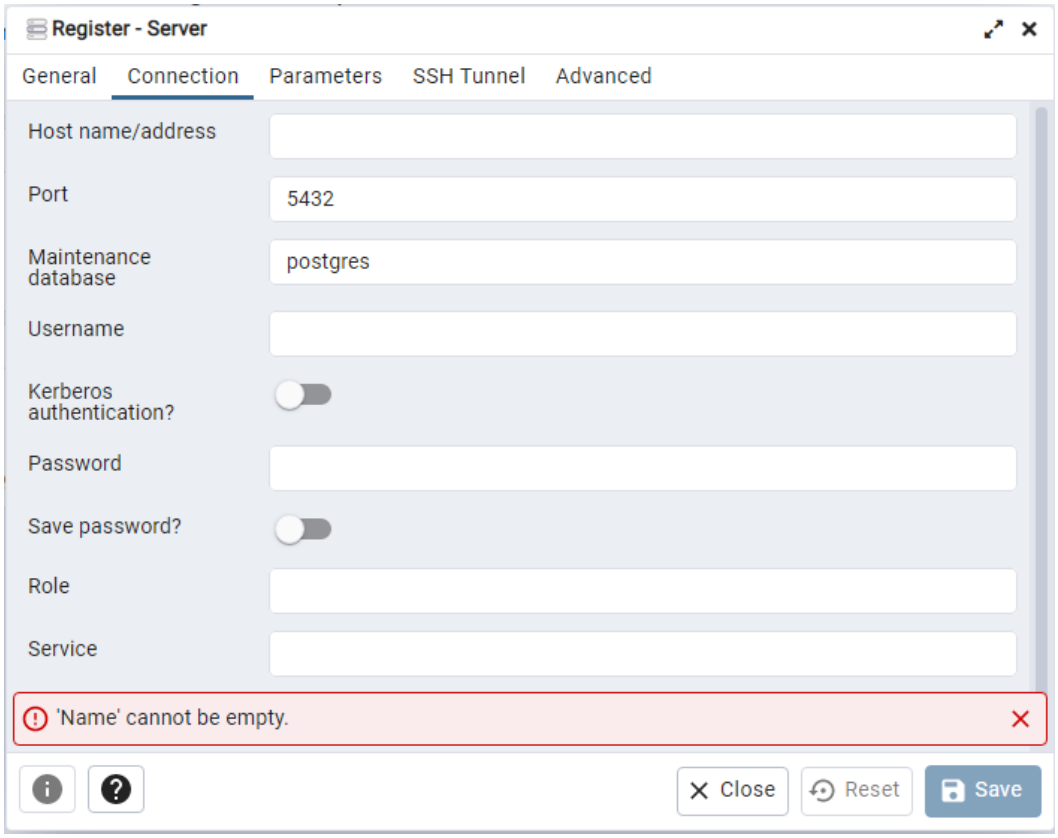


Figure 3-36 Entering connection information



Parameter description:

- **Host name/address**: indicates the EIP bound to the target DB instance.
- **Port**: indicates the database port. By default, the value is **5432**.
- **Username**: indicates the username. By default, the value is **root**.
- **Password**: indicates the password of the target database username.

Step 6 In the login window, check that the connection information is correct. The target DB instance is successfully connected.

----End

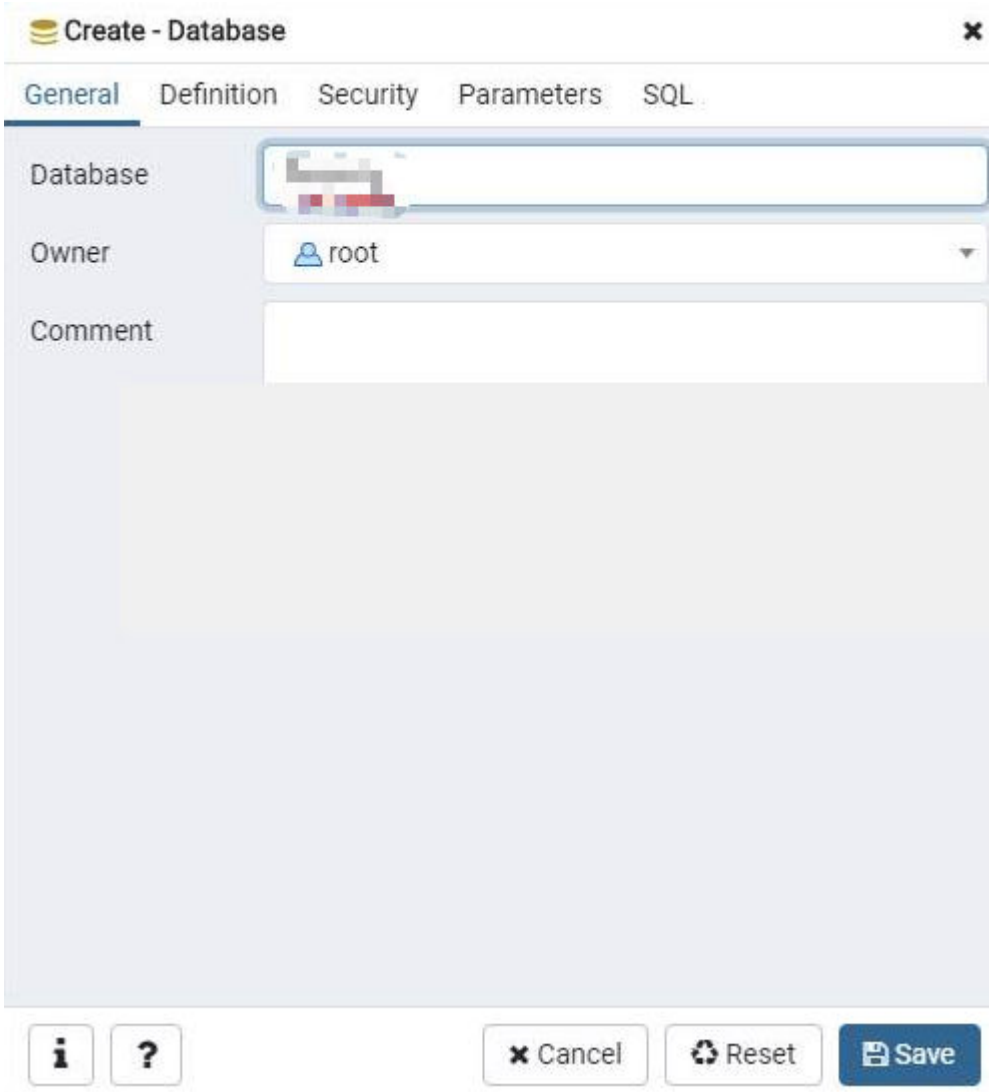
Basic Database Operations

Creating a Database

Step 1 In the navigation pane on the left of pgAdmin, right-click the target instance node and choose **Create** > **Database** from the shortcut menu.

Step 2 On the **General** tab, specify **Database** and click **Save**.

Figure 3-37 Creating a database



----End

Creating a Table

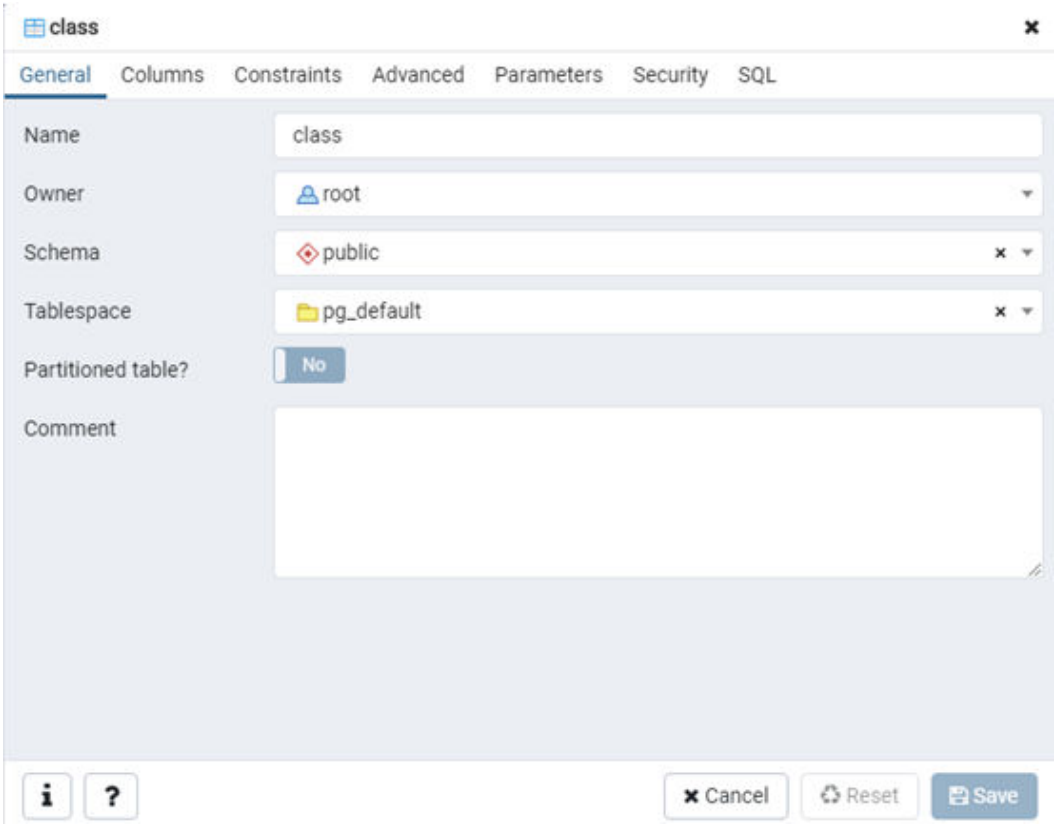
Step 1 Access the created database. In the navigation pane on the left, right-click **Tables** and choose **Create** > **Table** from the shortcut menu.

NOTE

Create tables in the schema of the database created by the current user.

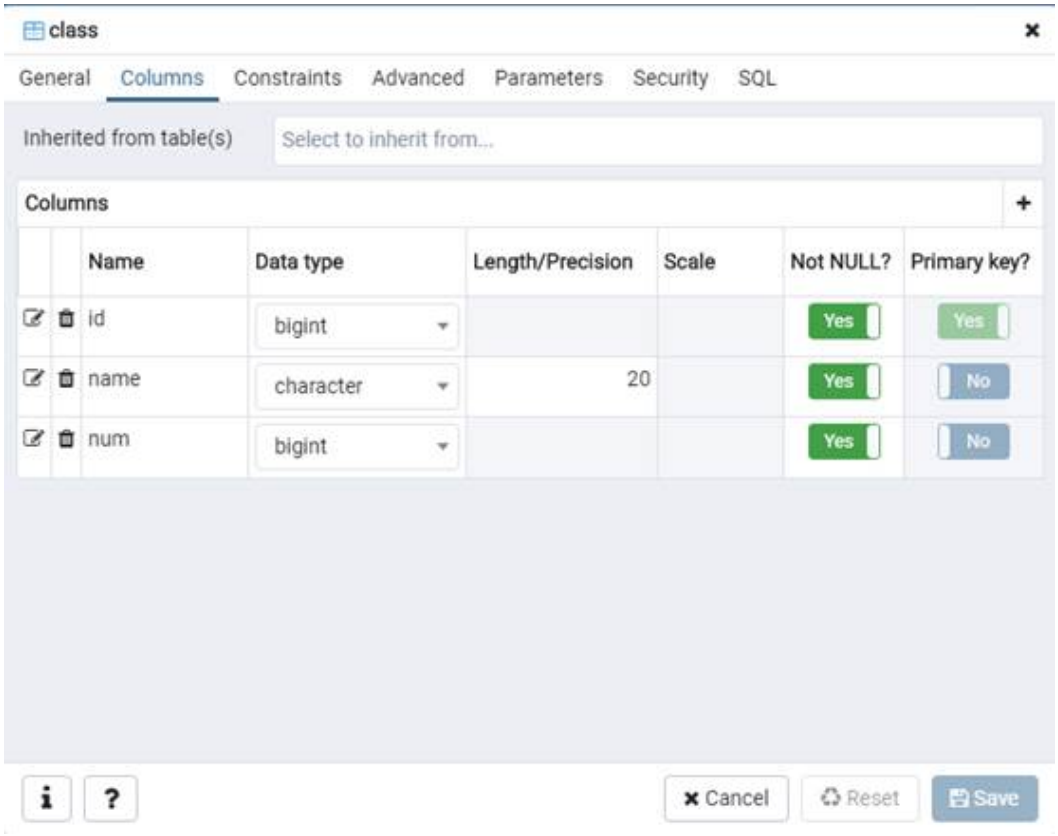
Step 2 On the **General** tab, enter required information and click **Save**.

Figure 3-38 Basic information



Step 3 On the **Columns** tab, add table columns and click **Save**.

Figure 3-39 Adding table columns



----End

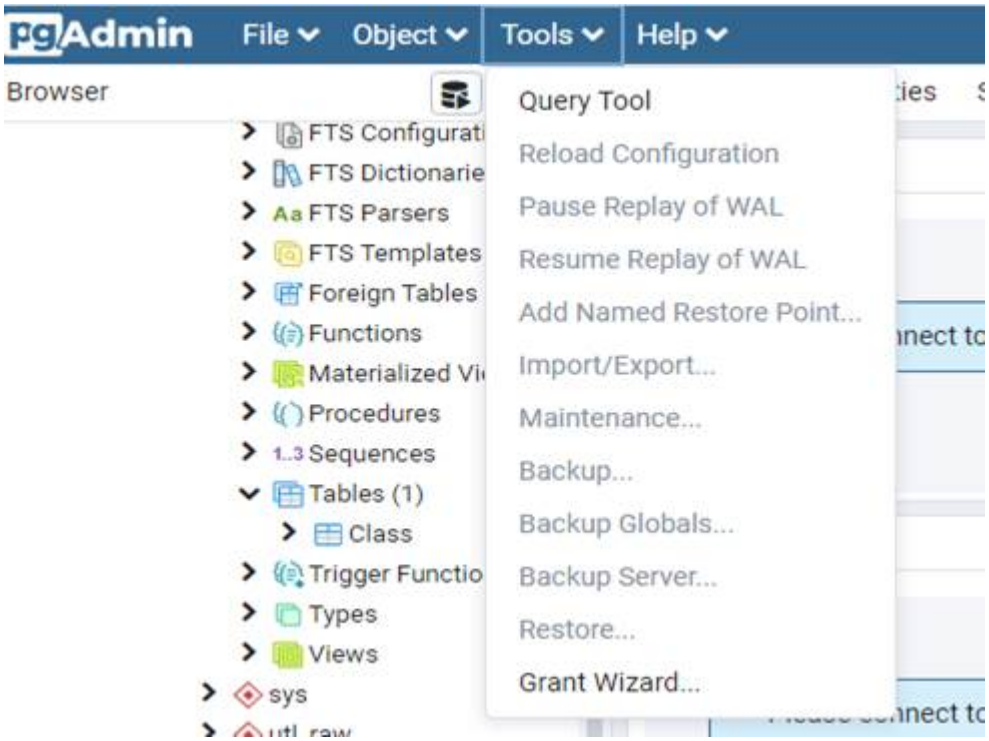
Executing SQL Statements

On the top menu bar, choose **Tools** > **Query Tool**. The SQL CLI is displayed.

NOTE

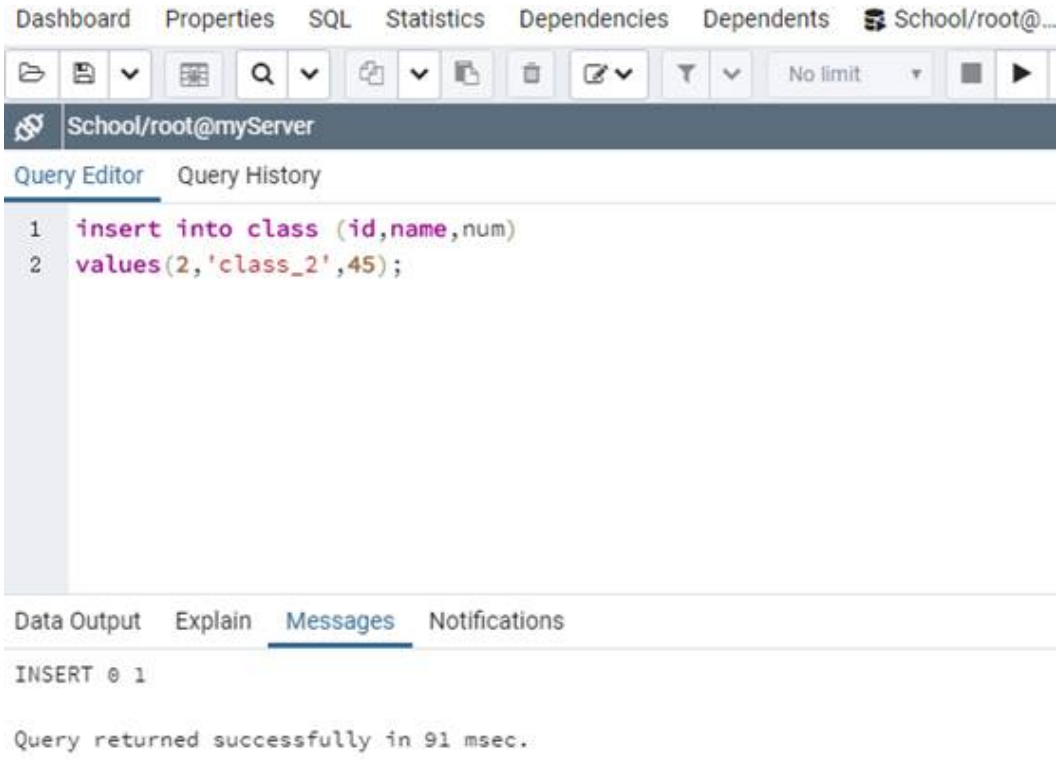
Take care when adding, deleting, or modifying data in the instance. Improper operations can cause instance or service exceptions.

Figure 3-40 SQL execution



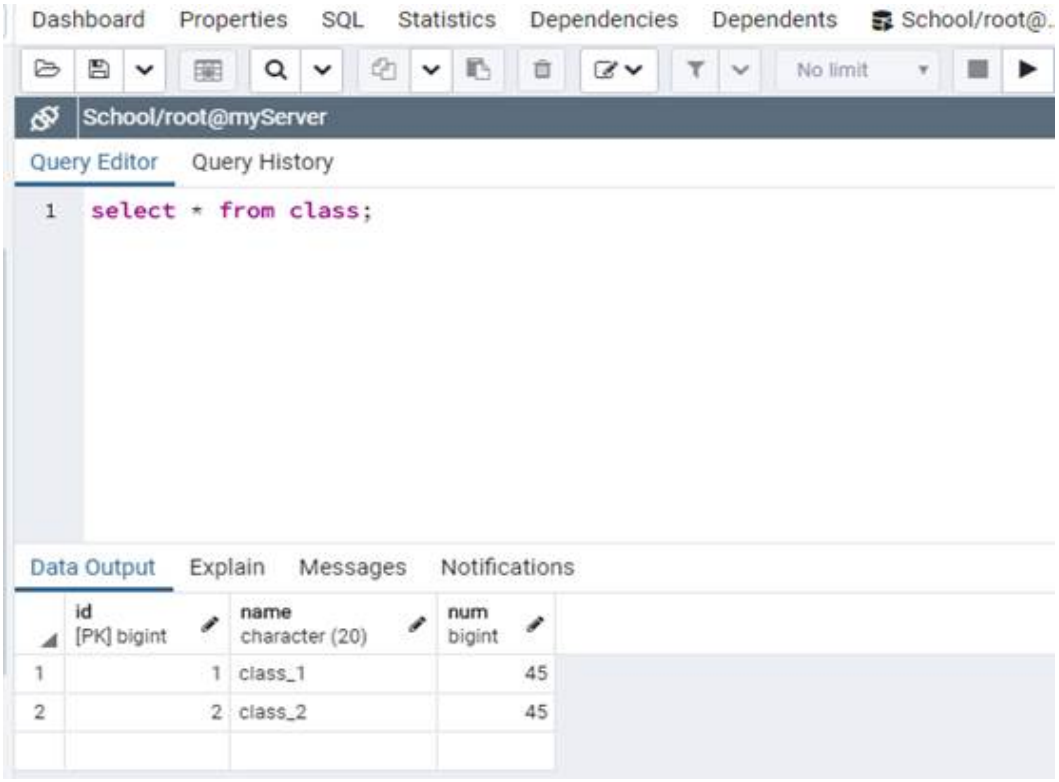
- Enter an **INSERT** command and click **Execute** to insert data into the table.

Figure 3-41 Inserting data



- Enter an **SELECT** command and click **Execute** to query data in the table.

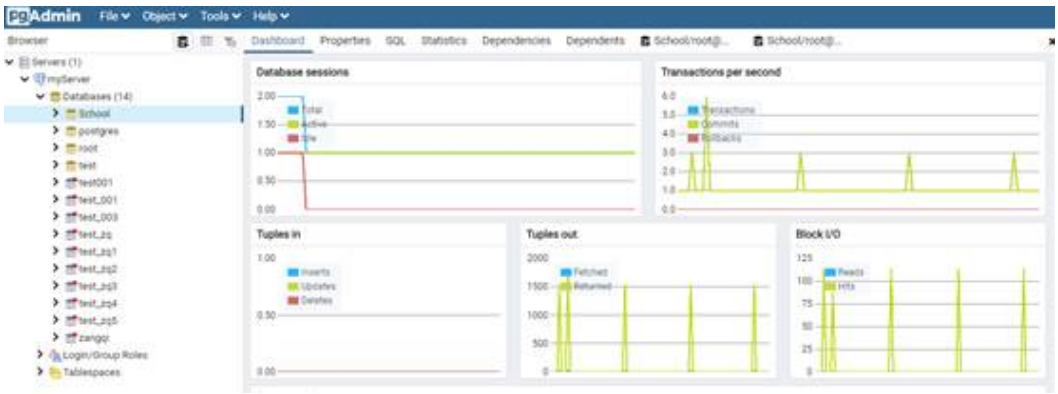
Figure 3-42 Querying data



Viewing Monitoring Data

In the navigation pane on the left, select a database and click the **Dashboard** tab on the right pane to view database metrics, including **Database sessions**, **Transactions per second**, **Tuples in**, **Tuples out**, and **Block I/O**.

Figure 3-43 Viewing metrics



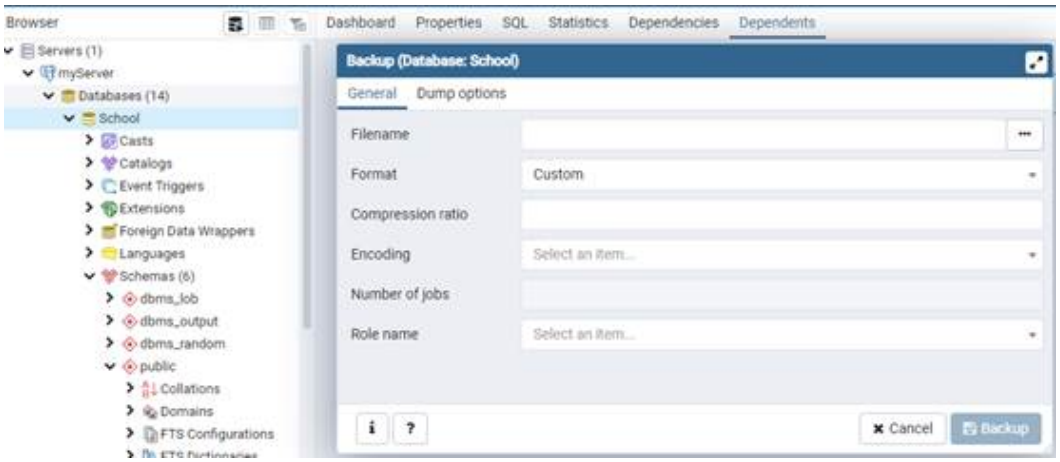
NOTE

Cloud Eye monitors operating statuses of DB instances. You can view the DB instance metrics on the management console. For details, see **Viewing Monitoring Metrics**.

Backing Up Data

1. In the navigation pane on the left, right-click the database to be backed up and choose **Backup** from the shortcut menu.
2. On the **General** tab, enter required information and click **Backup**.

Figure 3-44 Backing up data



Restoring Data

1. In the navigation pane on the left, right-click the database to be restored and choose **Restore** from the shortcut menu.
2. On the displayed tab in the right pane, select a file name and click **Restore**.

Figure 3-45 Restoring data

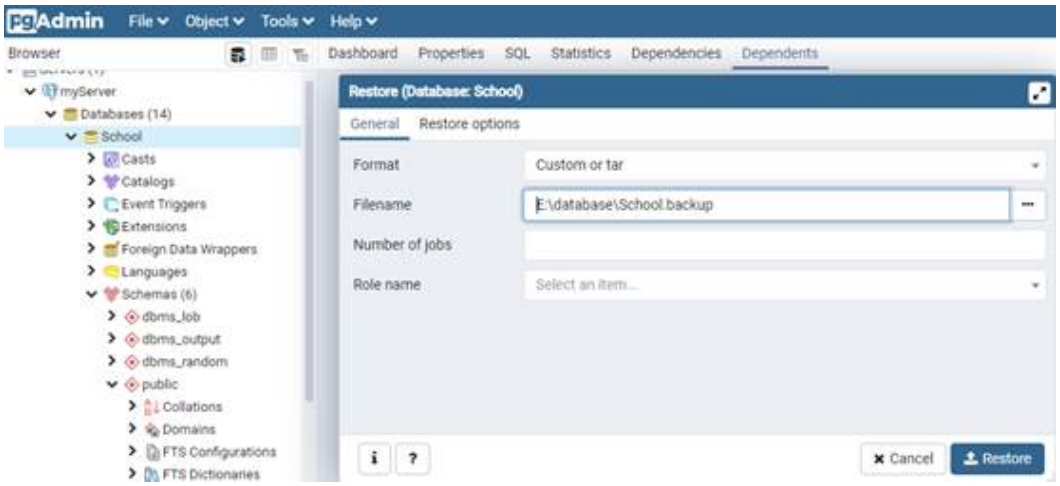
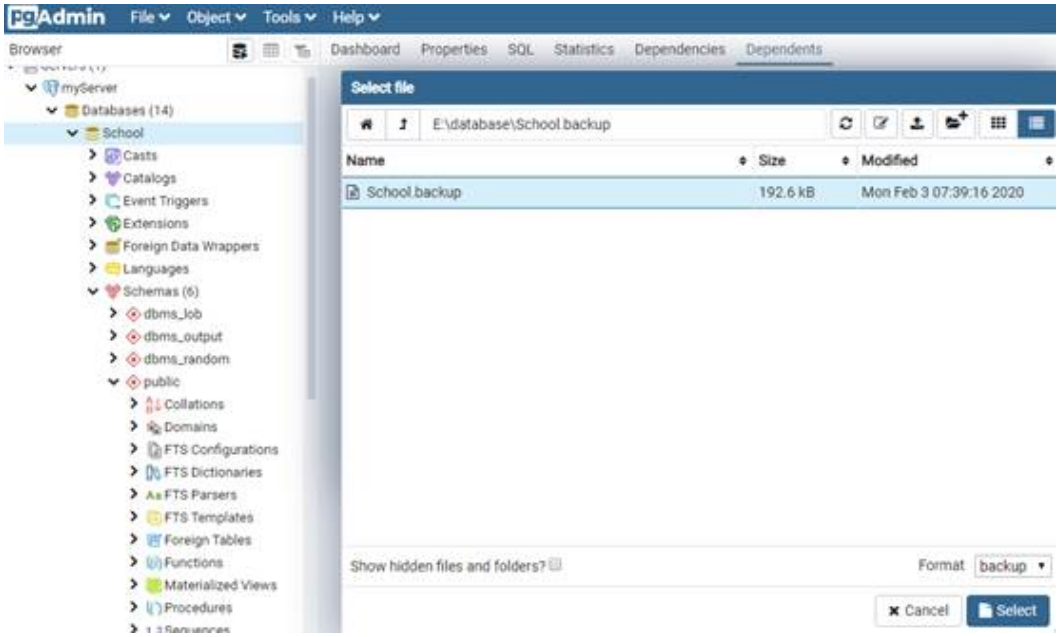


Figure 3-46 Restoration completed



3.3.5 Connecting to an RDS for PostgreSQL Instance 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

You are familiar with:

- Computer basics.
- Java.
- JDBC.

Obtaining and Using JDBC

- JDBC driver download address: **https://jdbc.postgresql.org/download/**
- JDBC API: **https://jdbc.postgresql.org/documentation/**

Connection with the SSL Certificate

NOTE

Download the SSL certificate and verify the certificate before connecting to databases.

On the **Instances** page, click the instance name to go to the **Overview** page. Under **SSL**, click **Download** to download the root certificate or certificate bundle.

Step 1 Connect to the RDS for PostgreSQL DB instance through JDBC.

jdbc:postgresql://*<instance_ip>***:***<instance_port>***/***<database_name>***?sslmode=verify-ca&sslrootcert=***<ca.pem>*

Table 3-10 Parameter description

Parameter	Description
<instance_ip>	If you attempt to access the RDS DB instance through an ECS, set *instance_ip* to the floating IP address displayed on the **Overview** page of the DB instance to which you intend to connect.
	If you attempt to access the RDS DB instance through an EIP, set *instance_ip* to the EIP that has been bound to the DB instance.
<instance_port>	Enter the database port displayed on the **Overview** 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. **verify-ca**: I want my data encrypted, and I accept the overhead. I want to be sure that I connect to a server that I trust. For details about other options, see **https://jdbc.postgresql.org/documentation/use/#connection-parameters/**.
sslrootcert	Path of the CA certificate for the SSL connection. For details, see **https://jdbc.postgresql.org/documentation/use/#connection-parameters/**.

Example script in Java:

```
// There will be security risks if the username and password used for authentication are directly written into code. Store the username and password in ciphertext in the configuration file or environment variables.
// In this example, the username and password are stored in the environment variables. Before running this example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.

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://<instance_ip>:<instance_port>/<database_name>?sslmode=verify-ca&sslrootcert=/home/Ruby/ca.pem";
        String userName = System.getenv("EXAMPLE_USERNAME_ENV");
        String password = System.getenv("EXAMPLE_PASSWORD_ENV");

        try {
            Class.forName("org.postgresql.Driver");
            conn = DriverManager.getConnection(url, userName, 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 for PostgreSQL DB instance through JDBC.

jdbc:postgresql:// *<instance_ip>*:*<instance_port>*/*<database_name>***?sslmode=disable**

Table 3-11 Parameter description

Parameter	Description
<instance_ip>	If you attempt to access the RDS DB instance through an ECS, set *instance_ip* to the floating IP address displayed on the **Overview** page of the DB instance to which you intend to connect.
	If you attempt to access the RDS DB instance through an EIP, set *instance_ip* to the EIP that has been bound to the DB instance.
<instance_port>	Enter the database port displayed on the **Overview** 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**: I don't care about security and don't want to pay the overhead for encryption. For details about other options, see **https://jdbc.postgresql.org/documentation/use/#connection-parameters/**.

Example script in Java:

```
// There will be security risks if the username and password used for authentication are directly written into code. Store the username and password in ciphertext in the configuration file or environment variables.
```



```
// In this example, the username and password are stored in the environment variables. Before running this example, set environment variables EXAMPLE_USERNAME_ENV and EXAMPLE_PASSWORD_ENV as needed.

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://<instance_ip>:<instance_port>/<database_name>?sslmode=disable";
                String userName = System.getenv("EXAMPLE_USERNAME_ENV");
        String password = System.getenv("EXAMPLE_PASSWORD_ENV");
        try {
            Class.forName("org.postgresql.Driver");
            conn = DriverManager.getConnection(url, userName, 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

3.3.6 Connecting to an RDS for PostgreSQL Instance Using Python

Prerequisites

You should be familiar with:

- Computer basics.
- Python programming language.
- How to use psycopg2.

Installation Dependency

Install psycopg2 to connect to PostgreSQL databases.

```
pip install psycopg2
```


Parameters

Table 3-12 Parameter description

Parameter	Description
dbname	The name of the database to be connected. The default database name is **postgres**.
user	The username used for connecting to the database.
password	The password of the username.
host	• If you access the database from an ECS, it is the floating IP address shown on the **Overview** page of the DB instance. • If you access the database over the Internet, it is the EIP that has been bound to the DB instance.
port	The port number shown on the **Overview** page of the DB instance. The default port number is **5432**.
sslmode	The SSL connection mode. • **disable**: No certificate is used for connection and security is not concerned. • **verify-ca**: CA authentication is used. • For details about other options, see the **community documentation**.
sslrootcert	The path of the server certificate.

Connecting to the Instance with an SSL Certificate

The following code is an example of how to use the **psycopg2.connect** function to connect to an RDS for PostgreSQL instance based on SSL certificate authentication and how to use the **cursor.execute** method to run INSERT and UPDATE statements on the instance.

NOTICE

An error "permission deny for schema public" may be reported when you create tables on certain instance versions. To resolve this problem, use the **grant create on SCHEMA public to root;** command.

```
import psycopg2
db_params
={'database':'postgres','user':'root','password':'****','host':'xxx.xxx.xxx.xxx','port':'5432','sslmode':'verify-
ca','sslrootcert':'/path/to/CA/ca.pem',}
conn=psycopg2.connect(**db_params)
print("Connection established")
cursor = conn.cursor()
# Drop previous table of same name if one exists
cursor.execute("DROP TABLE IF EXISTS inventory;")
```



```
print("Finished dropping table (if existed)")
# Create a table
cursor.execute("grant create on SCHEMA public to root;")
cursor.execute("CREATE TABLE inventory (id serial PRIMARY KEY, name VARCHAR(50), quantity INTEGER);")
print("Finished creating table")
# Insert some data into the table
cursor.execute("INSERT INTO inventory (name, quantity) VALUES (%s, %s);",("banana",150))
cursor.execute("INSERT INTO inventory (name, quantity) VALUES (%s, %s);",("orange",154))
cursor.execute("INSERT INTO inventory (name, quantity) VALUES (%s, %s);",("apple",100))
print("Inserted 3 rows of data")
cursor.execute("SELECT * FROM inventory;")
result = cursor.fetchall()
for row in result:
    print(row)
# Clean up
conn.commit()
cursor.close()
conn.close()
```

The command output is as follows:

```
Connection established
Finished dropping table(if existed)
Finished creating table
Inserted 3 rows of data
(1,'banana',150)
(2,'orange',154)
(3,'apple',100)
```

Connecting to the Instance Without an SSL Certificate

The following code is an example of how to connect to an RDS for PostgreSQL instance without using an SSL certificate.

```
import psycopg2
db_params
={'database':'postgres','user':'root','password':'****','host':'xxx.xxx.xxx.xxx','port':'5432','sslmode':'disable'}
conn=psycopg2.connect(**db_params)
print("Connection established")
cursor = conn.cursor()
# Drop previous table of same name if one exists
cursor.execute("DROP TABLE IF EXISTS inventory;")
print("Finished dropping table (if existed)")
# Create a table
cursor.execute("grant create on SCHEMA public to root;")
cursor.execute("CREATE TABLE inventory (id serial PRIMARY KEY, name VARCHAR(50), quantity INTEGER);")
print("Finished creating table")
# Insert some data into the table
cursor.execute("INSERT INTO inventory (name, quantity) VALUES (%s, %s);",("banana",150))
cursor.execute("INSERT INTO inventory (name, quantity) VALUES (%s, %s);",("orange",154))
cursor.execute("INSERT INTO inventory (name, quantity) VALUES (%s, %s);",("apple",100))
print("Inserted 3 rows of data")
cursor.execute("SELECT * FROM inventory;")
result = cursor.fetchall()
for row in result:
    print(row)
# Clean up
conn.commit()
cursor.close()
conn.close()
```

3.3.7 Connection Management

3.3.7.1 Viewing 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

To use floating IP addresses, **submit a service ticket** to apply for the required permissions.

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.

Only floating IPv4 addresses can be changed.

Procedure

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

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

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

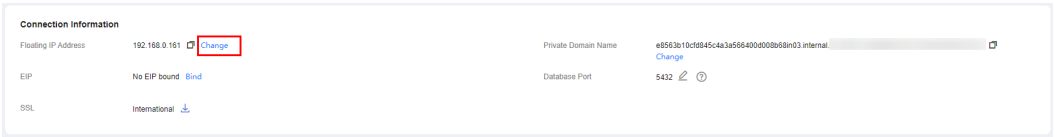
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

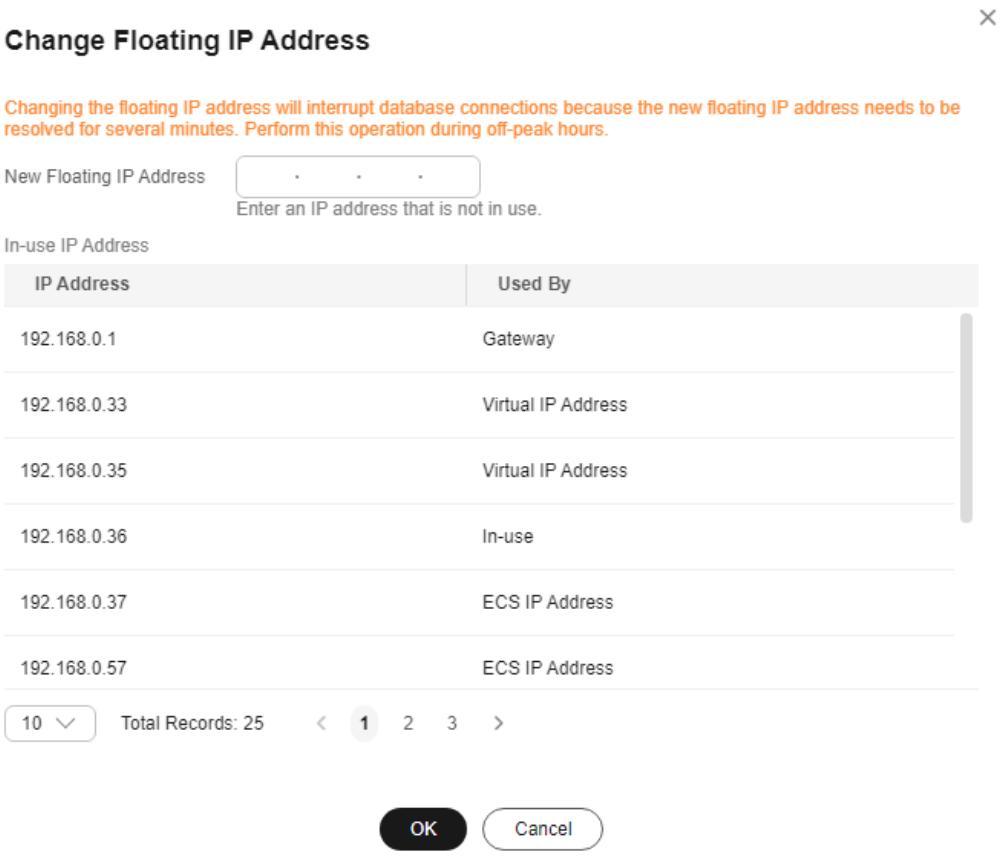
Step 5 Under **Floating IP Address**, click **Configure**.

Figure 3-47 Floating IP address



Step 6 In the displayed dialog box, check the number of in-use IP addresses. If the in-use IP addresses are less than 254, there are unused floating IP addresses.

Figure 3-48 Changing a floating IP address



Step 7 Enter an available IP address and click **Yes**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

Step 8 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

3.3.7.2 Changing a Private Domain Name

You can change the private domain name of your DB instance and connect to the instance using the new domain name.

Procedure

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **Private Domain Name**, click **Configure**.

Step 6 In the displayed dialog box, enter a new private domain name. Click **OK**.

NOTE

- Only the prefix of a private domain name can be modified.
- The prefix of a private domain name can contain 8 to 63 characters, and can include only letters and digits.
- The new private domain name must be different from the existing ones.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

3.3.7.3 Configuring SSL Encryption

SSL is enabled by default when you create an RDS for PostgreSQL DB instance and cannot be disabled after the instance is created. SSL encryption ensures that all communications between a client and server are encrypted, preventing data leakage and tampering and ensuring data integrity.

Impact of SSL Encryption on Database Performance

Enabling SSL reduces the read-only and read/write performance of your instance by about 20%.

The impact varies depending on the service model. SSL encryption has little impact on database performance if there are complex SQL statements being executed because the execution of such statements takes much time. But SSL encryption will decrease the performance if simple SQL statements are being executed because the execution is fast.

Checking Whether SSL Is Enabled on the Server

By default, SSL is enabled on the RDS for PostgreSQL instance server. You can log in to the instance and run the following SQL command to check whether SSL is enabled:

```
show ssl;
```

- If the **ssl** value is **on**, SSL is enabled on the server.
- If the **ssl** value is **off**, SSL is disabled on the server.

NOTE

SSL is enabled on the server by default and cannot be disabled.

Checking Whether SSL Is Enabled on the Client

You can check whether the client uses SSL encryption in either of the following ways:

- Check whether the following information is displayed when you use psql to connect to the DB instance:
  ```
  SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
  ```
  - **protocol** indicates the SSL connection protocol, which is **TLSv1.2**.
  - **cipher** indicates the encryption algorithm used for SSL connection, which is **ECDHE-RSA-AES256-GCM-SHA384**.
  - **bits** indicates the key length, which is **256** bits.
- Query the **pg_stat_ssl** view to check whether the client uses SSL connection. If yes, corresponding connection information is displayed in the view.
  ```
  SELECT * FROM pg_stat_ssl;
  ```
  This query returns the statistics of all current SSL connections, including the process ID, client IP address, SSL protocol version, SSL encryption algorithm, and validity and expiration date of the client certificate. If the client uses SSL connection, you can view the related information in this view.

Parameters Related to SSL Encryption on the Server

Table 3-13 Parameters related to SSL encryption on the server

Parameter	Value	Description
ssl	on	SSL is enabled by default and **cannot be disabled**.
ssl_cert_file	/CA/server.pem	Location of the SSL certificate file on the server, **which cannot be changed**.
ssl_ciphers	ALL:!ADH:!LOW:!EXP:!MD5:!3DES:!DES:@STRENGTH;	SSL cipher list for secure connection. You can change the value based on security requirements. Recommended cipher list: EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EDH+aRSA+AESGCM:EDH+aDSS+AESGCM:!aNULL:!eNULL:!LOW:!3DES:!MD5:!EXP:!SRP:!RC4
ssl_key_file	/CA/server.key	Location of the SSL private key file on the server, **which cannot be changed**.
ssl_min_protocol_version	TLSv1.2	Minimum SSL/TLS protocol version to be used. You can change the value based on security requirements. TLSv1.2 or later is recommended.

Parameters Related to SSL Encryption on the Client

After SSL is enabled for an RDS for PostgreSQL instance, the client can connect to the instance through SSL.

When the client connects to the instance, you can set **sslmode** based on the site requirements.

- If SSL connection is used, **sslmode** can be set to **allow**, **prefer**, **Require**, **Verify-CA**, or **Verify-Full**. The default value is **prefer**.
- If SSL connection is not used, set **sslmode** to **Disable**.

NOTE

If **sslmode** is set to **Verify-CA** or **Verify-Full**, you need to set the **Root certificate** parameter, which indicates the path of the database CA certificate. The CA certificate can be downloaded from the console.

Table 3-14 sslmode values

Value	Description
disable	The client does not use the SSL connection.
allow	The client attempts to establish an SSL or TLS connection. If the server does not support the SSL or TLS connection, the client connects to the server in common text mode.
prefer	Default value. The client attempts to establish an SSL connection first. If the server does not support the SSL connection, the client connects to the server in common text mode.
require	The client only attempts to establish an SSL connection, encrypts the data link, and does not verify the validity of the server certificate.
verify-ca	The client uses SSL to connect to the server and verifies the validity of the server certificate.
verify-full	The client uses SSL to connect to the server, verifies the validity of the server certificate, and checks whether the CN or DNS in the certificate is consistent with the database connection address configured during the connection.

Related Operations

Connecting to an RDS for PostgreSQL Instance Through JDBC

Using psql CLI to Connect to an Instance Through a Private Network

Using psql CLI to Connect to an Instance Through a Public Network

3.3.7.4 Binding and Unbinding an EIP

Scenarios

You can bind an EIP to your DB instance to enable public network access and can unbind the EIP later if it is not 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 5432, ensure that the security group allows access over the 5432 port.

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, add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see **Configuring Security Group Rules**.
- Traffic generated by the public network is charged. You can unbind the EIP from your DB instance when the EIP is no longer used.

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**.

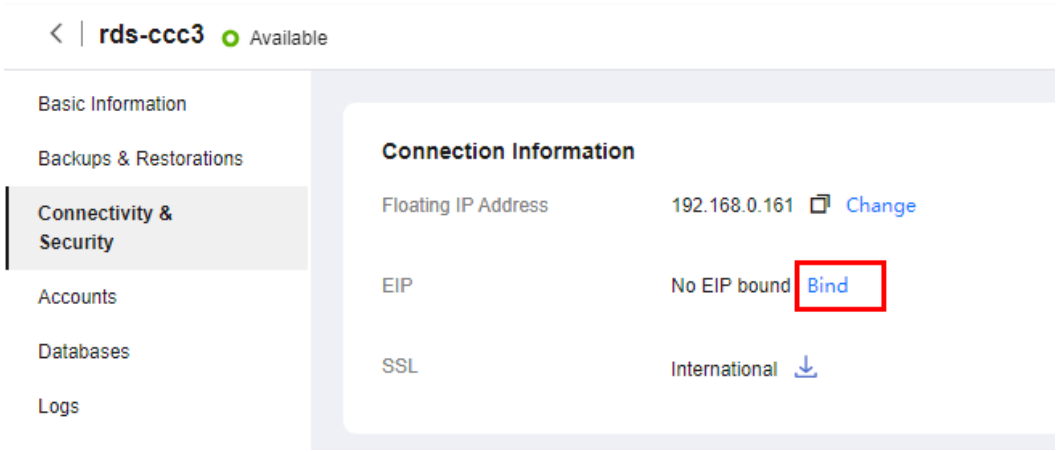
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Figure 3-49 Connectivity & Security



Step 6 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **Yes**. If no available EIPs are displayed, click **View EIP** and obtain an EIP.

Step 7 On the **EIPs** page, On the **Connectivity & Security** 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance that has an EIP bound.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

Step 7 On the **Connectivity & Security** page, view the results.

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

3.3.7.5 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.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

RDS for PostgreSQL 11 DB instances will not be rebooted after you change the database port.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Overview** page, find **Database Port** and click **Configure** under it.

NOTE

RDS for PostgreSQL instances can use database ports 2100 to 9500.

- In the displayed dialog box, enter a new port and click **Yes**.

  If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

  Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

  a. 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.

  b. 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.

  c. This process takes 1-5 minutes.

- In the displayed dialog box, click **No** to cancel the modification.

Step 6 View the result on the **Overview** page.

----End

3.4 Database Usage

3.4.1 Suggestions on Using RDS for PostgreSQL

3.4.1.1 Instance Usage Suggestions

Database Connection

RDS for PostgreSQL uses a process architecture, providing a backend service process for each client connection.

- Set **max_connections** depending on the type of your application. Use the parameter settings provided on pgtune as examples:
  - Set **max_connections** to **200** for web applications.
  - Set **max_connections** to **300** for OLTP applications.
  - Set **max_connections** to **40** for data warehouses.
  - Set **max_connections** to **20** for desktop applications.
  - Set **max_connections** to **100** for hybrid applications.
- Limit the maximum number of connections allowed for a single user based on workload requirements.
  ```
  ALTER ROLE xxx CONNECTION LIMIT xxx;
  ```
- Set the number of active connections to two to three times the number of vCPUs.
- Avoid long transactions, which may block autovacuum and affect database performance.
- Periodically release persistent connections because maintaining them may generate a large cache and use up memory. You can configure parameters such as **idle_session_timeout** and **idle_in_transaction_session_timeout** to release idle connections.
- Check the application framework to prevent the application from automatically starting transactions without performing any operations.

Read Replicas

- Avoid long transactions, which may cause query conflicts and affect playback.
- Configure **hot_standby_feedback** for instances requiring real-time data and set **max_standby_streaming_delay** to a proper value.
- Monitor long transactions, long connections, and replication delay and address all issues in a timely manner.
- Ensure that applications connected to a read replica can be switched to other nodes because read replicas are single-node instances incapable of providing high availability.

Reliability and Availability

- Select primary/standby DB instances for production databases.
- Keep vCPU, memory, and storage usage less than 85% for production databases to prevent problems such as out of memory (OOM) and full storage.
- Deploy primary and standby instances in different AZs to improve availability.
- Set the time window for automated backup to off-peak hours. Do not disable full backup.
- Configure asynchronous replication between primary and standby DB instances to prevent workloads on the primary instance from being blocked due to a fault on the standby instance.
- Pay attention to the size of temporary files and the generation rate. Too many temporary files affect database performance, slow down database startup, and even cause service unavailability.
- Do not create too many objects in one instance. Generally, the number of tables in a single instance does not exceed 20,000, and that in a single database does not exceed 4,000. This prevents service unavailability caused by long-time table file scanning during database startup.

Logical Replication

- Keep the name of a logical replication slot less than 40 bytes to prevent full backup failures.
- Delete replication slots that are no longer used for logical replication to prevent database bloat.
- Replication slots will be lost after a primary/standby switchover is performed (due to an instance class change, a minor version upgrade, or a host failure). When this occurs, you need to create replication slots again.
- Use failover slots for RDS for PostgreSQL 12.6 and later minor versions, and all minor versions of RDS for PostgreSQL 13 and 14 to prevent replication slot loss after a primary/standby switchover or instance reboot.
- When using logical replication, avoid long transactions and commit discarded two-phase transactions in a timely manner to prevent stacked WAL logs from occupying too much storage space.
- When using logical replication, avoid using a large number of sub-transactions (such as savepoints and exception clauses in a transaction) to prevent high memory usage.
- When using Data Replication Service (DRS) to synchronize or migrate data, delete the logical replication slots contained in the databases that are rarely accessed or add heartbeat tables to periodically push the replication slots to prevent stacked WAL logs.

Database Age

- Definition of database age:
  - Database age is a PostgreSQL-specific concept. It refers to the latest transaction ID minus oldest transaction ID in the database.
  - As defined in the Multi-Version Concurrency Control (MVCC) mechanism of RDS for PostgreSQL, the maximum age allowed for a database is 2

billion transactions old. When a database reaches the maximum age, it will be forcibly shut down. In this case, contact technical support to vacuum the database.

  - To view the age of a database, run the following SQL statement:

    select datname, age(datfrozenxid) from pg_database;

- You are advised to use the **db_max_age** metric to monitor the database age and set the alarm threshold to 1 billion.

Stability

- Commit or roll back two-phase transactions in a timely manner to prevent database bloat.
- Change the table structure, for example, adding fields or indexes, during off-peak hours.
- To create indexes during peak hours, use the CONCURRENTLY syntax to avoid blocking the DML of the table.
- Before modifying the structure of a table during peak hours, perform a verification test to prevent the table from being rewritten.
- Configure a lock wait timeout duration for DDL operations to avoid blocking operations on related tables.
- Partition your database if its capacity exceeds 2 TB.
- If a frequently accessed table contains more than 20 million records or its size exceeds 10 GB, split the table or create partitions.
- To prevent replication exceptions on the standby instance or read replicas, control the data write speed of the primary instance under 50 MB/s. That's because the standby instance or read replica replays WAL logs in a single process at a maximum speed of 50 MB/s to 70 MB/s.

Routine O&M

- Periodically download and view slow query logs on the **Logs** page to identify and resolve performance issues in a timely manner.
- Periodically check the resource usage of your instance. If the service pressure fluctuates greatly, you are advised to configure resource alarms and upgrade the instance specifications when necessary. High write pressure will slow down database reboots and affect service availability.
- Run the **SELECT** statement before deleting or modifying a record.
- After a large amount of data is deleted or updated in a table, run VACUUM on the table.
- Note the number of available replication slots and ensure that at least one replication slot is available for database backup.
- Remove any replication slots that are no longer used to prevent the replication slots from blocking log reclaiming.
- Do not use unlogged tables because data in these tables will be lost after a database exception (such as OOM or underlying faults) or primary/standby switchover.
- Do not run VACUUM FULL on system catalogs. If necessary, run VACUUM. Running VACUUM FULL on system catalogs causes the instance to reboot and the instance cannot be connected for a long time.

Security

- Avoid enabling access to your database from the Internet. If you do need to enable Internet access, bind an EIP to your DB instance and configure a whitelist.
- Use SSL to connect to your DB instance.

3.4.1.2 Database Usage Suggestions

Naming

- The names of objects (such as databases, tables, and indexes) must be no more than 63 bytes. Note that some characters (such as Chinese characters) may occupy multiple bytes.
- Do not use reserved database keywords in object names or start an object name with pg or a digit.
- A database name can contain 1 to 63 characters. Only letters, digits, and underscores (_) are allowed. It cannot start with **pg** or a digit and cannot be the same as RDS for PostgreSQL template database names. RDS for PostgreSQL template databases include postgres, template0, and template1.

Table Design

- The table structure must be designed in advance to avoid frequent structure changes, such as adding fields or changing data types.
- There cannot be more than 64 fields in a single table.
- Create partitioned tables for the tables whose data needs to be deleted periodically. For example, you can create partitions by time and delete data from the partitions using DROP or TRUNCATE.
- Use appropriate data types for table fields. For example, do not use the character type for numeric or date data.
- When using the numeric data type, ensure that the values are within allowed ranges and meet precision requirements.

Index Design

- Design primary keys or unique keys for tables that require logical replication.
- When creating a foreign key, specify the action for deleting or updating the foreign key, for example, ON DELETE CASCADE.
- Create indexes for fields that are frequently used (such as fields for data query and sorting).
- Create partial indexes for queries using fixed conditions.
- Create expression indexes for queries using conditional expressions.
- A single table cannot contain too many indexes because indexes also occupy storage. For example, there should be fewer than 5 single-column indexes and fewer than 3 composite indexes.

SQL Design

- Specify the required fields to be returned in a query.

- Only use IS NULL or IS NOT NULL to determine whether a field is NULL.
- Use NOT EXISTS instead of NOT IN in a query.
- Use UNION ALL instead of UNION to concatenate result sets.
- Use TRUNCATE instead of DELETE to delete an entire table.
- Submit data changes in large transactions in batches to prevent high pressure during transaction commit or rollback.
- When creating a function, define the volatility of the function as the strictest category, instead of the default VOLATILE. Too many concurrent calls of VOLATILE functions may result in failures to establish new connections.

Security

- Do not assign the public role to the owner of an application database object. Assign a specific role to the owner.
- A database password must meet complexity requirements.
- Allocate a unique database account for each service.
- When accessing an object, explicitly specify the schema of the object to avoid accessing objects with the same name in other schemas.

3.4.2 Databases

3.4.2.1 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 being restored.
- Database names must be unique.
- After a database is created, its name cannot be changed.

Procedure

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

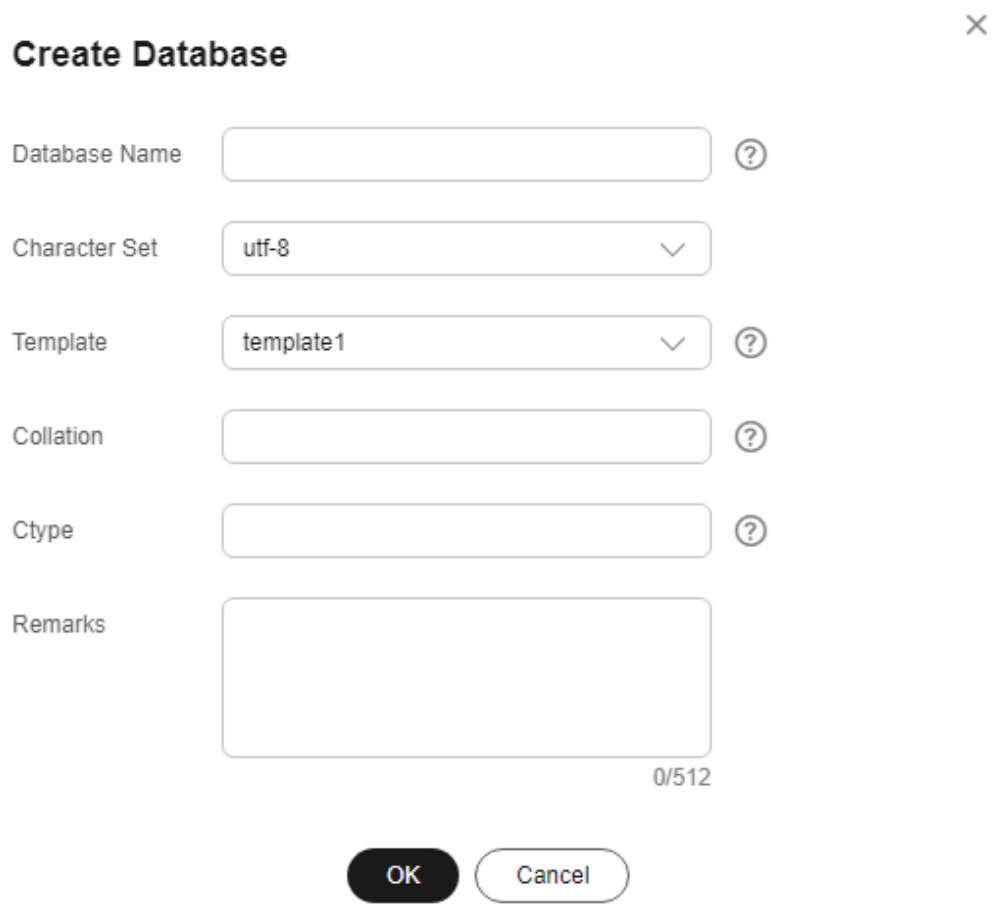
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Databases** page, click **Create Database**. In the displayed dialog box, configure required parameters and click **OK**.

Figure 3-50 Creating a database



- The database name can contain 1 to 63 characters. Only letters, digits, and underscores (_) are allowed. It cannot start with pg or a digit and cannot be the same as RDS for PostgreSQL template database names. RDS for PostgreSQL template databases include postgres, template0, and template1.
- The default character set is **utf-8**. You can change it as required.
- You can specify a template database. A new database will be created using this template. Option template1 (default option) is adapted to RDS and template0 complies with the PostgreSQL community settings.
- A collation defines how characters are ordered. en_US.UTF-8 is used by default. Different collations may result in varied ordering results. For example, select 'a'>'A'; is false under en_US.UTF-8, but true under 'C', and also, 'C' must be used to obtain the migration results as expected when data is migrated from Oracle to RDS for PostgreSQL. Collations supported by a database can be queried from system catalog **pg_collation**.
- Ctype refers to character classification to be used in the new database (LC_CTYPE). The use of this parameter affects the classification of characters, such as lowercase letters, uppercase letters, and digits. By default, the character classification of the template database is used.
- The remarks can contain 0 to 512 characters.

Step 6 After the database is created, manage it on the **Databases** page.

----End

3.4.2.2 Modifying Database Remarks

Scenarios

RDS allows you to modify remarks for databases.

Constraints

The remarks of system database **postgres** cannot be modified.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Databases**.

Step 6 Locate the target database and click in the **Remarks** column.

NOTE

The database remarks can be empty or contain up to 512 characters.

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

----End

3.4.2.3 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

Databases cannot be deleted from DB instances that are being restored.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, click **Databases**.

Step 6 Locate the target database and click **Delete** in the **Operation** column.

Step 7 In the displayed dialog box, click **OK**.

----End

3.4.3 Accounts (Non-Administrator)

3.4.3.1 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

- The instance must be in the running state.
- This operation is not allowed for DB instances that are being restored.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Accounts** page, click **Create Account**.

Step 6 In the displayed dialog box, set required parameters and click **OK**.

Figure 3-51 Creating a database account

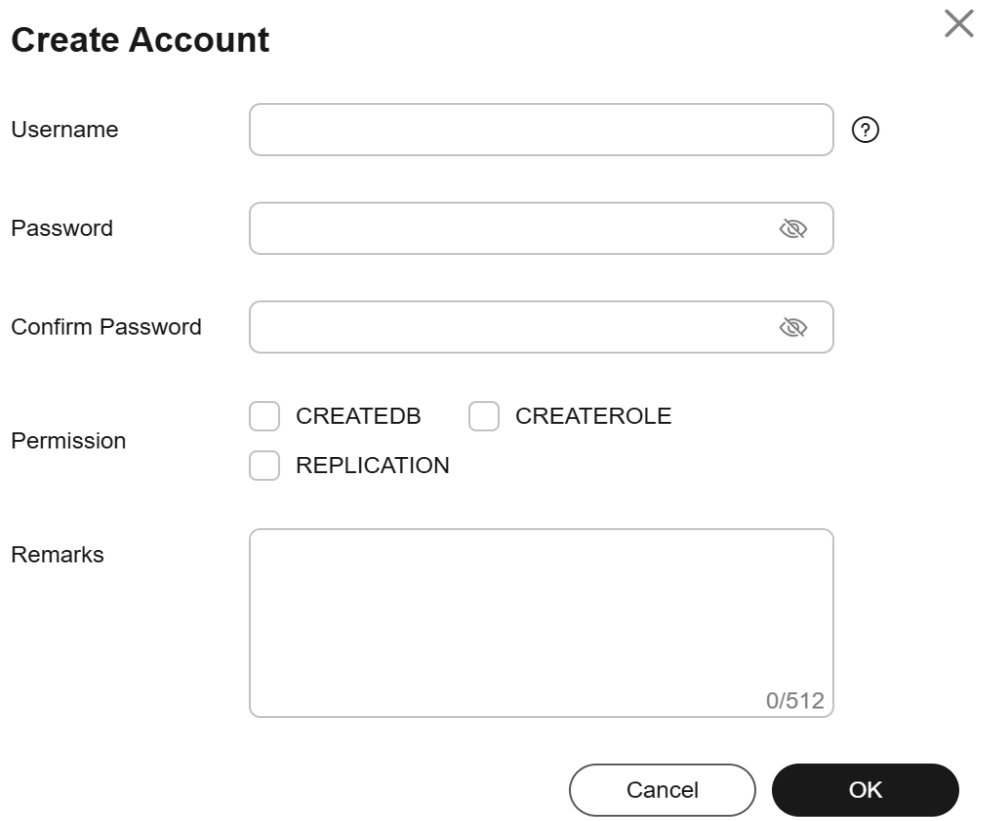


Table 3-15 Parameter description

Parameter	Description
Username	The username can contain 1 to 128 characters. It can include letters, digits, hyphens (-), and underscores (_), and it must be different from system accounts. System accounts include **rdsadmin**, **rdsuser**, **rdsbackup**, and **rdsmirror**.
Password	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 cannot contain the username or the username spelled backwards. You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.

Parameter	Description
Permission	You can assign permissions, including **CREATEDB**, **CREATEROLE**, and **REPLICATION**, to the user. • **CREATEDB**: indicates that the user has the permission to create a database. If this attribute is not specified, the user cannot create databases by default. • **CREATEROLE**: indicates that the user has the permission to create other users. If this attribute is not specified, the user cannot be used to create new users by default. • **REPLICATION**: indicates that the user can use streaming replication or logical replication. If this attribute is not specified, the user cannot be used to set up streaming replication or logical replication by default.
Remarks	The remarks can contain 0 to 512 characters.

Step 7 After the account is created, manage it on the **Accounts** page.

----End

Privileges of the root User

RDS for PostgreSQL provides permissions for the **root** user. To create objects on an RDS for PostgreSQL database without operation risks, escalate your account to root privileges when necessary.

The following table describes root privilege escalation in different versions.

Table 3-16 Privileges of the **root** user

Version	Whether to Escalate Privileges	Initial Version for Privilege Escalation
pgcore9	No	N/A
pgcore10	No	N/A
pgcore11	Yes	11.11
pgcore12	Yes	12.6
pgcore13	Yes	13.2
pgcore14	Yes	14.4
pgcore15	Yes	15.4
pgcore16	Yes	16.2

Escalate to root privileges when you need to:

- Create an event trigger.
- Create a wrapper.
- Create a logical replication publication.
- Create a logical replication subscription.
- Query and maintain replication sources.
- Create a replication user.
- Create a full-text index template and parser.
- Run the **vacuum** command on a system table.
- Run the **analyze** command on a system table.
- Create an extension.
- Grant an object permission to a user.

3.4.3.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect your instance against brute force cracking, change your password periodically, such as every three or six months.

Procedure

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

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

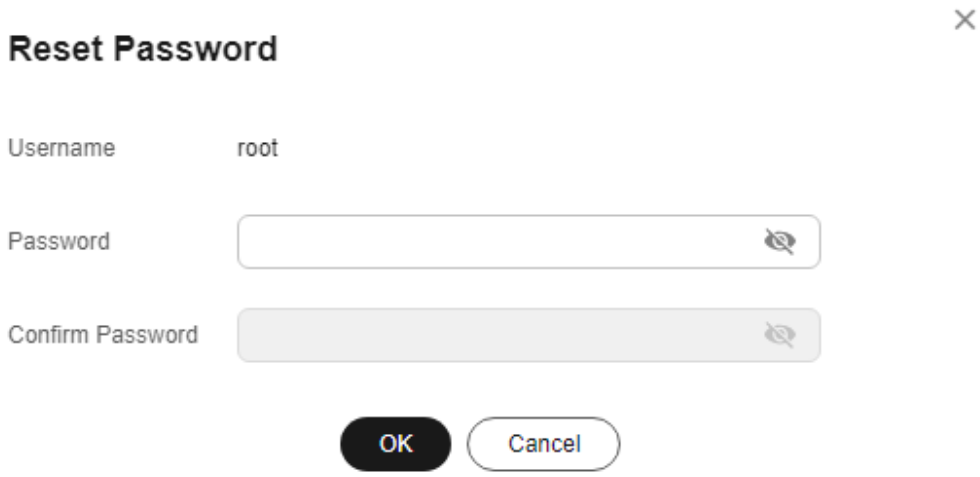
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the target username and click **Reset Password** in the **Operation** column.

Step 6 In the displayed dialog box, enter a new password, confirm the password, and click **OK**.

Figure 3-52 Resetting a password



- 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 cannot contain the username or the username spelled backwards.
- You are advised to enter a strong password to improve security and prevent security risks such as brute force cracking.
- You can use Cloud Trace Service (CTS) to query the password reset records. For details, see **Viewing Tracing Events**.

----End

3.4.3.3 Modifying Remarks of a Database Account

Scenarios

RDS allows you to modify remarks of database accounts.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Accounts**.

Step 6 Locate the target username and click in the **Remarks** column.

NOTE

The remarks can contain 0 to 512 characters.

- To submit the modification, click ✓.
- To cancel the modification, click ✗.

----End

3.4.3.4 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 being restored.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the target username and click **Delete** in the **Operation** column.

Step 6 In the displayed dialog box, click **OK**.

----End

3.4.3.5 Modifying pg_hba.conf

Scenarios

You can modify parameters in the **pg_hba.conf** file for your DB instance to ensure access to your database.

Constraints

- This operation is not allowed when another operation is being performed on the DB instance.
- All restored DB instances will not inherit the **pg_hba.conf** files of the original instances. You need to reconfigure the **pg_hba.conf** file.
- Only the **md5**, **reject**, and **scram-sha-256** authentication methods are supported.

Procedure

Step 1 Log in to the management console.

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

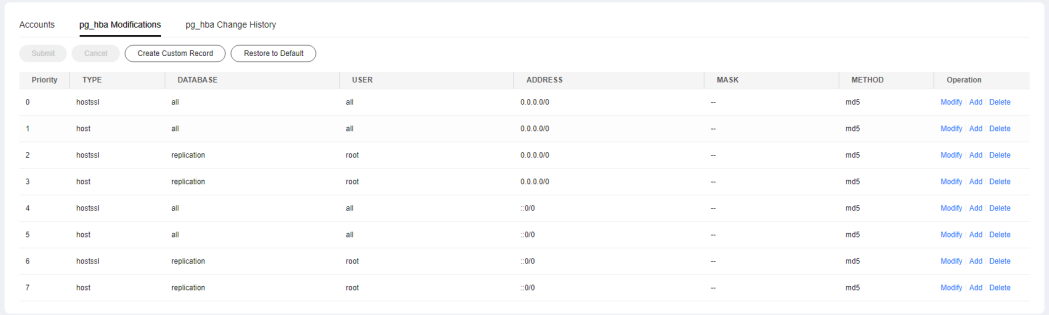
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Accounts** and click **pg_hba Modifications**.

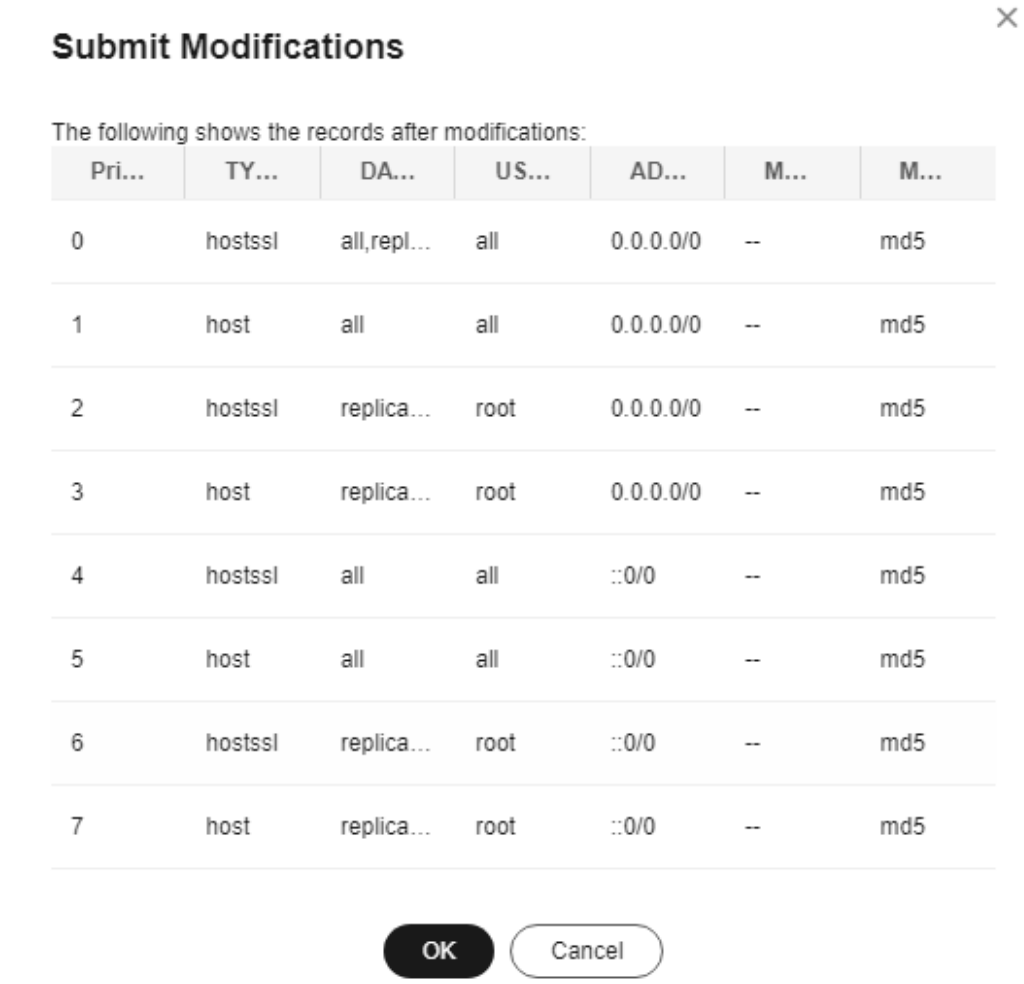
There are four default rules.

Figure 3-53 pg_hba Modifications



Step 6 To modify or delete a default rule or add a new rule, locate the rule and click **Modify**, **Delete**, or **Add** in the **Operation** column. Then click **Submit** above the list. In the displayed dialog box, click **OK**. For the description of each parameter, see **Table 3-17**.

Figure 3-54 Submit Modifications



NOTE

After the modification is submitted, the new configurations take effect only for new connections. For old connections, you need to disconnect the connections and reconnect them for the modification to take effect.

Table 3-17 Parameter description

Parameter	Example Value	Description
Priority	0	Priority of the record. The value **0** indicates the highest priority.

Parameter	Example Value	Description
TYPE	host	Type of the record. Valid values: • **host**: The record matches connection attempts made using TCP/IP. **host** records match either SSL or non-SSL connection attempts. • **hostssl**: The record matches connection attempts made using TCP/IP, but only when the connection is made with SSL encryption. • **hostnossl**: The record only matches connection attempts made over TCP/IP that do not use SSL.
DATABASE	all	Database that can be accessed by the user. The value **all** indicates that the user can access all databases. If multiple databases are specified, separate them with commas (,). The specified databases must have been created and cannot be **template0** or **template1**.
USER	user0	User who is allowed to access the database. Set this parameter to the username created in **Creating a Database Account**. If multiple usernames are configured, separate them with commas (,). The specified users must have been created and cannot be built-in users such as **rdsAdmin**, **rdsMetric**, **rdsBackup**, **rdsRepl**, and **rdsProxy**.
ADDRESS	0.0.0.0/0 ::0/0	IP address that the user can access the database from. **0.0.0.0/0** (IPv4) or **::0/0** (IPv6) indicates that the user can access the database from any IP address. **NOTE** The IP addresses that are not in this CIDR block cannot access the database. Exercise caution when modifying this parameter.
MASK	Empty	Subnet mask. If **ADDRESS** is set to an IP address, you can use this parameter to specify the subnet mask of the IP address.

Parameter	Example Value	Description
METHOD	md5	Authentication method. Valid values: • **reject** • **scram-sha-256** • **md5**

Step 7 To import rules in batches, click **Create Custom Record** above the list. In the displayed dialog box, configure new rules and click **OK**.

- **Append records with lowest priorities**: The new rules are added below the existing rules and have the lowest priorities.
- **Append records with highest priorities**: The new rules are added above the existing rules and have the highest priorities.
- **Overwrite existing records**

Step 8 To restore the default **pg_hba.conf** configurations, click **Restore to Default** above the list.

Step 9 Run the **psql** command to connect to the database and test the connectivity you have specified for **pg_hba.conf**.

```
psql -h <instance_connection_address> -U <specified_username> -p 5432 -d <specified_database_name>
```

----End

3.4.3.6 Viewing the pg_hba.conf Change History

Scenarios

You can query the change history of **pg_hba.conf**.

Procedure

Step 1 Log in to the management console.

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

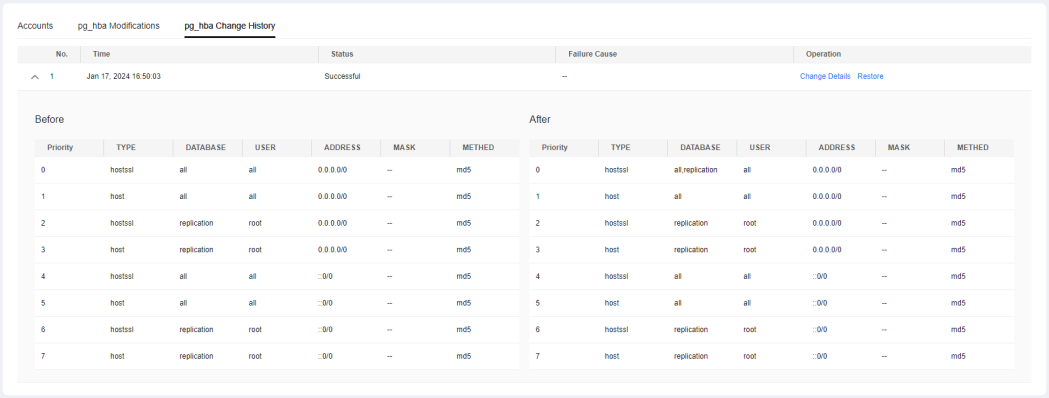
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Accounts** and click **pg_hba Change History**.

Step 6 To view the comparisons before and after the change, click **Change Details**.

Figure 3-55 Viewing the pg_hba change history



Step 7 To restore the parameter settings, click **Restore**. Exercise caution when performing this operation.

----End

3.4.4 Tablespace Management

Scenarios

RDS provides the PostgreSQL tablespace management solution based on user **root**.

Creating a Tablespace

Step 1 Connect to the database as user **root** and create a tablespace.

 # psql --host=*<RDS_ADDRESS>* **--port=***<DB_PORT>* **--dbname=***<DB_NAME>* **--username=root -c** "**select control_tablespace** (**'create'**, '*<TABLESPACE_NAME>*');"

Table 3-18 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
DB_NAME	Indicates the database name.
TABLESPACE_NAME	Indicates the tablespace name.

Step 2 Enter the password of user **root** when prompted.

Log in to the **my_db** database and create the **tbspc1** tablespace. Example:

 # psql --host=192.168.6.141 --port=5432 --dbname=my_db --username=root -c "select control_tablespace('create', 'tbspc1');"

```
Password for user root:
        control_tablespace
```



```
------------------------------
create tablespace tbspc1 successfully.
(1 row)
```

If the creation fails, view error logs of the DB instance.

NOTE

To ensure performance, a maximum of 100 tablespaces can be created.

----End

Granting Tablespace Permissions

Step 1 Connect to a database as user **root** and grant the tablespace usage permissions to specified users.

psql --host=*<RDS_ADDRESS>* **--port=***<DB_PORT>* **--dbname=***<DB_NAME>* **--username=root -c** "**select control_tablespace ('alter',** '*<TABLESPACE_NAME>*', '*<USER_NAME>*');"

Table 3-19 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
DB_NAME	Indicates the database name.
TABLESPACE_NAME	Indicates the tablespace name.
USER_NAME	Indicates the tablespace username.

Step 2 Enter the password of user **root** when prompted.

Log in to the **my_db** database and grant permissions to tablespace **tbspc1**, for example:

 # psql --host=192.168.6.141 --port=5432 --dbname=my_db --username=root -c "select control_tablespace('alter', 'tbspc1', 'user1');"

```
Password for user root:
      control_tablespace
----------------------------
alter tablespace tbspc1 successfully.
(1 row)
```

If the permissions fail to be granted, view error logs of the DB instance.

----End

Deleting a Tablespace

Step 1 Connect to a database as user **root** and delete a tablespace.

 # psql --host=<*RDS_ADDRESS*> **--port=**<*DB_PORT*> **--username=root --dbname=**<*DB_NAME*> **-c** "**select control_tablespace**('**drop',** '<*TABLESPACE_NAME*>');"

Table 3-20 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
DB_NAME	Indicates the database name.
TABLESPACE_NAME	Indicates the tablespace name.

Step 2 Enter the password of user **root** when prompted.

Example:

 # psql --host=192.168.6.141 --port=8635 --dbname=my_db --username=root -c "select control_tablespace('drop', 'tbspc1');"

```
Password for user root:
        control_tablespace
----------------------------
drop tablespace tbspc1 successfully.
(1 row)
```

Before deleting the tablespace, ensure that it is empty. If the deletion fails, view error logs of the DB instance.

----End

3.5 Database Migration

3.5.1 Migration Solution Overview

You can migrate data from RDS for MySQL, self-managed MySQL databases, MySQL databases built on other clouds, self-managed Oracle databases, self-managed PostgreSQL databases, or PostgreSQL databases built on other clouds to RDS for PostgreSQL, or from one RDS for PostgreSQL instance to another RDS for PostgreSQL instance.

Data migration tools include Data Replication Service (DRS), pg_dump, and Data Admin Service (DAS). You are advised to use DRS because it is easy to use and can complete a migration task in minutes. DRS facilitates data transfer between databases, helping you reduce DBA labor costs and hardware costs.

DRS provides real-time synchronization. Real-time synchronization refers to the real-time flow of workload data from sources to destinations through a synchronization instance while consistency of data is ensured. It is different from migration. Migration means moving entire data of a database to another. Synchronization refers to the continuous flow of data between different applications.

For more information, see **What Is DRS?**

Migration Solutions

Table 3-21 RDS for PostgreSQL migration solutions

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
RDS for PostgreSQL	Small	One-time	Some time	Use pg_dump to copy data from the source to the destination RDS for PostgreSQL instance.	**Migrating Data to RDS for PostgreSQL Using psql**
	Medium	One-time	Some time	Use DAS to export data from the source and then import the data to the destination RDS for PostgreSQL instance.	**Migrating Data to RDS for PostgreSQL Using the Export and Import Functions of DAS**
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from the source to the destination RDS for PostgreSQL instance.	**From PostgreSQL to RDS for PostgreSQL**
• On-premises PostgreSQL databases • PostgreSQL databases on ECSs	Any	One-time or continuous	Minimal	Use DRS to synchronize data from self-managed PostgreSQL databases to RDS for PostgreSQL.	**From PostgreSQL to RDS for PostgreSQL**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
PostgreSQL databases on other clouds	Any	One-time or continuous	Minimal	Use DRS to synchronize data from PostgreSQL databases on other clouds to RDS for PostgreSQL.	**From PostgreSQL to RDS for PostgreSQL**
• On-premises Oracle databases • Oracle databases on ECSs	Any	One-time or continuous	Minimal	Use DRS to synchronize data from self-managed Oracle databases to RDS for PostgreSQL.	**From Oracle to RDS for PostgreSQL**
RDS for MySQL	Any	One-time or continuous	Minimal	Use DRS to synchronize data from RDS for MySQL to RDS for PostgreSQL.	**From MySQL to RDS for PostgreSQL**
• On-premises MySQL databases • MySQL databases on ECSs	Any	One-time or continuous	Minimal	Use DRS to synchronize data from self-managed MySQL databases to RDS for PostgreSQL.	**From MySQL to RDS for PostgreSQL**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
MySQL databases on other clouds	Any	One-time or continuous	Minimal	Use DRS to synchronize data from MySQL databases on other clouds to RDS for PostgreSQL.	From MySQL to RDS for PostgreSQL

3.5.2 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 the 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.

   For details, see **Step 2: Test Connectivity and Install the PostgreSQL Client**.

   NOTE

   The PostgreSQL client version must be the same as the DB engine version of your RDS for PostgreSQL instance. A PostgreSQL database or client will provide pg_dump and psql.

Exporting Data

Before migrating an existing PostgreSQL database 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 have to stop all applications using the source database.
- Take care when exporting or importing data. Improper operations can cause instance or workload exceptions.

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 as prompted.

NOTE

If the exported SQL file uses INSERT statements, you can easily edit and modify the file. However, the speed of importing data may be slower than that of using COPY statements. You are advised to select a right statement format as needed.

- If both the source and destination databases are PostgreSQL databases, you are advised to export COPY statements (default). For details, see **Example 1: Exporting the source database to an SQL file (COPY)**.
- If either of the source and destination databases is a non-PostgreSQL database, you are advised to export INSERT statements. For details, see **Example 2: Exporting the source database to an SQL file (INSERT)**.

For more information, see **pg_dump options**.

Examples:

- Example 1: Exporting the source database to an SQL file (COPY)

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db

  Password for user root:
- Example 2: Exporting the source database to an SQL file (INSERT)

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --inserts --file=backup.sql my_db

  Password for user root:
- Example 3: Exporting all table structures from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --schema-only --file=backup.sql my_db

Password for user root:

- Example 4: Exporting all table data from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --data-only --file=backup.sql my_db

  Password for user root:

After the commands in any of the above examples are 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
```

Step 3 Use pg_dump to export tables from the source database to an SQL file.

pg_dump --username=*<DB_USER>* **--host=***<DB_ADDRESS>* **--port=***<DB_PORT>* **--format=plain --file=***<BACKUP_FILE> <DB_NAME>* **--table=***<TABLE_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 be exported.
- ***DB_NAME*** indicates the name of the database to be migrated.
- ***TABLE_NAME*** indicates the name of the specified table in the database to be migrated.

Enter the database password as prompted.

Examples:

- Example 1: Exporting one table from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db --table=test

  Password for user root:

- Example 2: Exporting multiple tables from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db --table=test1 --table=test2

  Password for user root:

- Example 3: Exporting all tables starting with ts_ from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db --table=ts_*

  Password for user root:

- Example 4: Exporting all tables except those starting with ts_ from the source database to an SQL file

  $ pg_dump --username=root --host=192.168.151.18 --port=5432 --format=plain --file=backup.sql my_db -T=ts_*

  Password for user root:

After the commands in any of the above examples are 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

3.5.3 Migrating Data to RDS for PostgreSQL Using the Export and Import Functions of DAS

Scenarios

To back up or migrate data, you can use Data Admin Service (DAS) to export data from the source database first and then import the data to from your local PC or OBS bucket to the destination database.

For more information, see **Import and Export**.

Constraints

- Take care when exporting or importing data. Improper operations can cause instance or workload exceptions.
- Only one file that is no larger than 1 GB can be imported at a time.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- If there are more than 100,000 tables in an RDS for PostgreSQL instance, an error will be reported when you export data using the **Export Database** function of DAS. In this case, use the **Export SQL Result** function instead.

Exporting Data

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

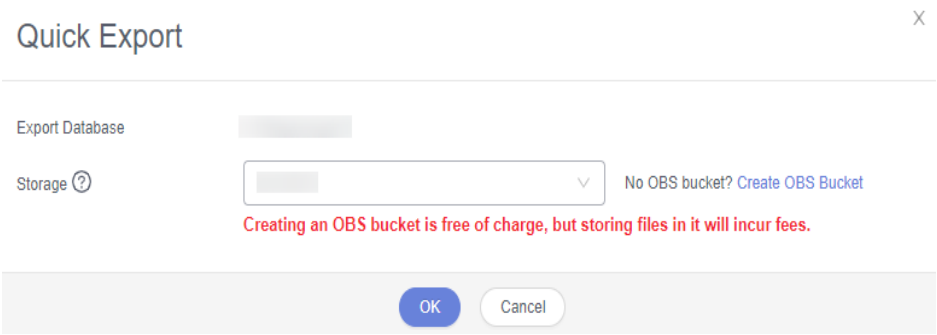
Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.

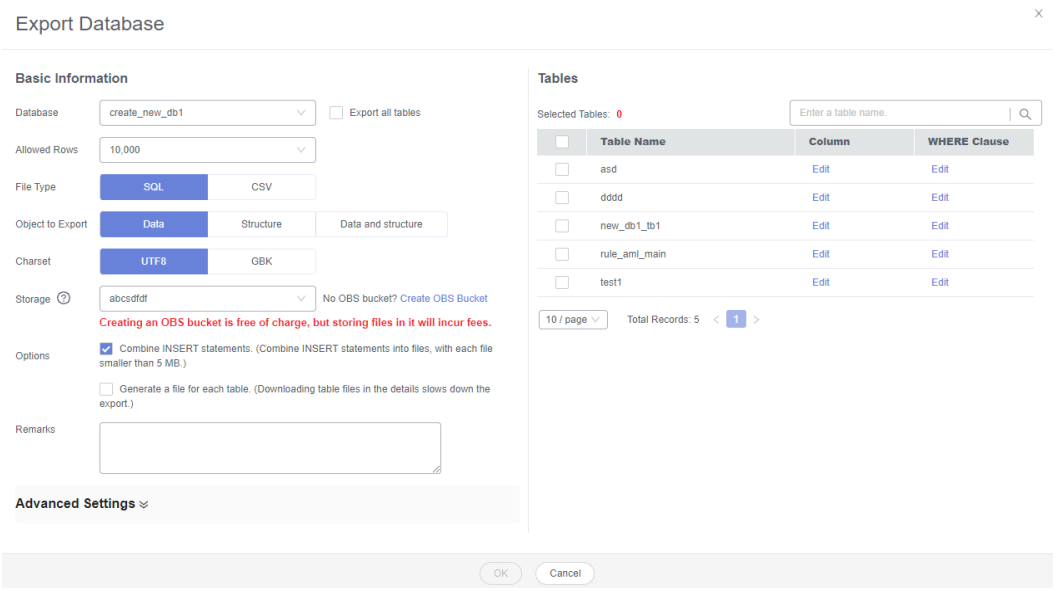
Alternatively, click **Quick Export** and select the target database. On the displayed page, select a storage path and click **OK**.

Figure 3-56 Quick export



Step 8 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

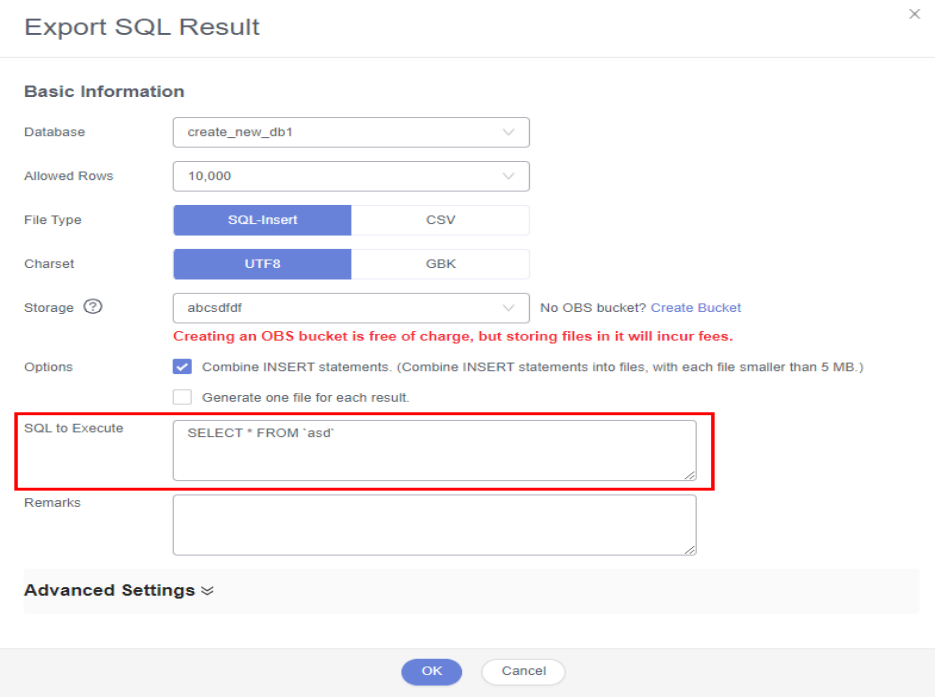
Figure 3-57 Creating an export task



NOTE

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.

Figure 3-58 Export SQL result



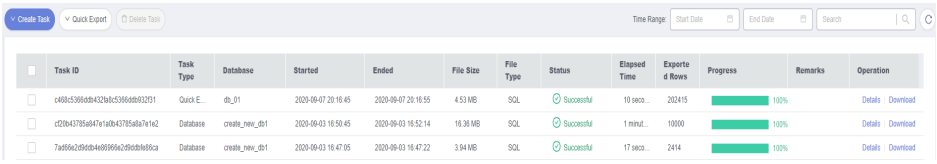
- Databases are classified into user databases and system databases. System databases cannot be exported. If system database data is required, deploy system database services in a created user database, so that you can export the system database data from the user database.
- DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

Figure 3-59 Task list



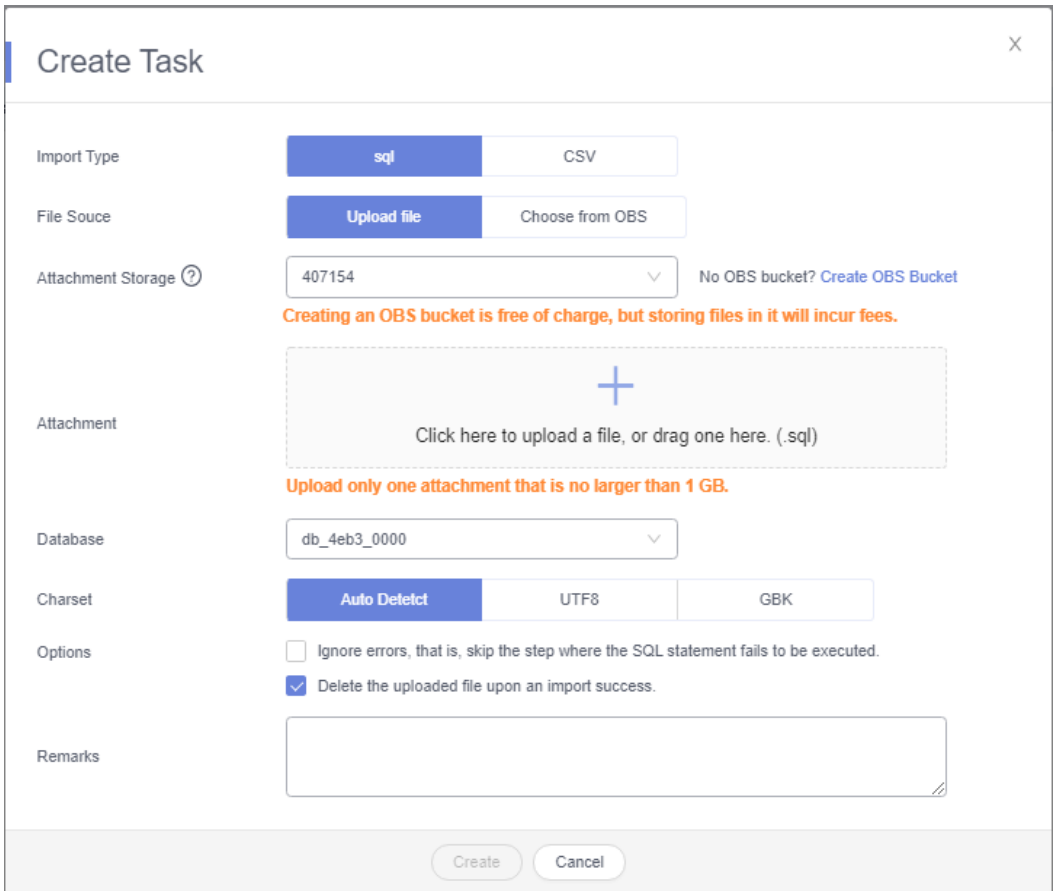
----End

Importing Data

Step 1 On the top menu bar, choose **Import and Export** > **Import**.

Step 2 Import a file from your local PC or an OBS bucket.

Figure 3-60 Creating an import task



- From your local PC

In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.

For security purposes, imported files are stored in OBS buckets.

NOTE

- To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
- If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database.

- From an OBS bucket

  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

NOTE

The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting the import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

Step 4 View the import progress in the task list or check task details.

----End

3.6 Version Upgrade

3.6.1 Upgrading a Minor Version

Scenarios

RDS for PostgreSQL supports minor version upgrades to improve performance, add new functions, and fix bugs.

Precautions

- When any new minor version is released to address vulnerabilities and other issues from the open source community, **perform a minor version upgrade** for your instance.
- The upgrade will cause the instance to reboot and interrupt services for a period of time. The length of the interruption depends on service volume. To minimize the impact of the upgrade, perform the upgrade during off-peak hours, or ensure that your applications support automatic reconnection.
- When you upgrade the minor version of a primary instance, the minor versions of read replicas (if any) will also be upgraded automatically. Read replicas cannot be upgraded separately.
- A minor version upgrade cannot be rolled back after the upgrade is complete. If the upgrade fails, the DB instance will be automatically rolled back to the source version.
- You are advised to perform a full backup before upgrading a minor version.
- If the storage is insufficient before a minor version upgrade, **scale up storage space** first. If storage autoscaling is triggered during the upgrade, both of them will fail.
- You need to re-establish a DR relationship after upgrading the minor version of a DR instance.
- After a minor version upgrade, you may need to update extensions. For details, see **Installing and Uninstalling an Extension on the RDS Console**.
- Before upgrading minor versions earlier than RDS for PostgreSQL 12.6, you need to stop all logical replications and delete all logical replication slots. Otherwise, the upgrade will fail.
  - Querying a replication slot: **select * from pg_replication_slots;**
  - Deleting a replication slot: **select pg_drop_replication_slot('SLOT_NAME');**
- If the kernel version of your instance has potential risks or major defects, has expired, or has been brought offline, the system will notify you by SMS message or email and deliver an upgrade task during the maintenance window.

Constraints

- The minor version cannot be upgraded for instances with abnormal nodes.
- For primary/standby DB instances, the standby DB instance is upgraded first and then the primary DB instance is upgraded afterwards.
- The following minor versions cannot be upgraded:
  - Versions earlier than 11.2 for RDS for PostgreSQL 11
  - Versions earlier than 1.0.12 for RDS for PostgreSQL Enhanced Edition
- The upgrade will be performed immediately upon the submission of your request. Delayed upgrade of minor versions is not supported.
- Read replicas cannot be upgraded independently.
- DB instances of the latest version cannot be upgraded.

Procedure

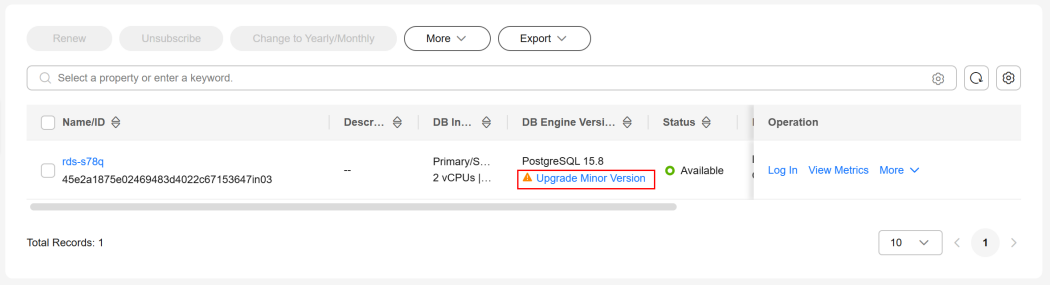
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

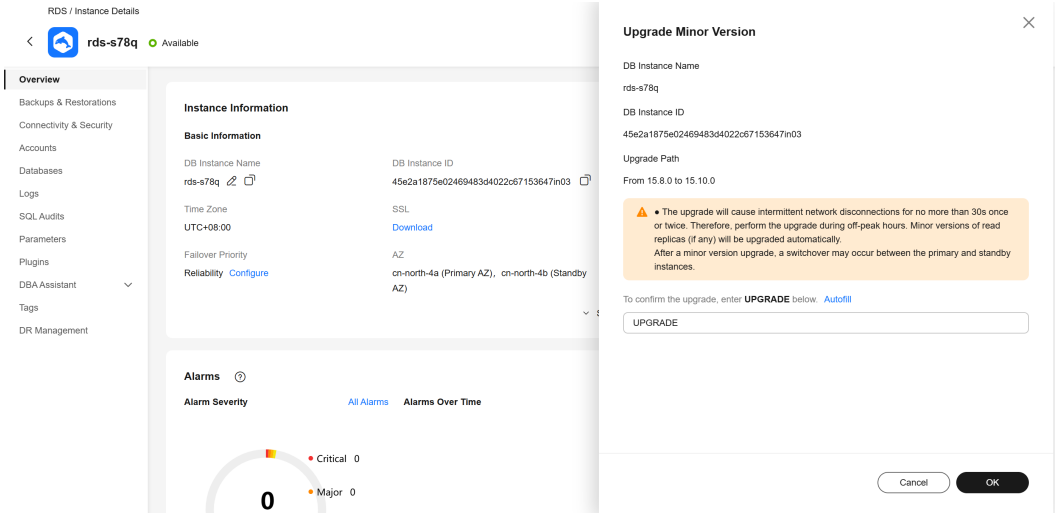
Step 4 On the **Instances** page, locate the target primary/standby instance and click **Upgrade Minor Version**.

Figure 3-61 Upgrading a minor version



Step 5 In the displayed dialog box, enter **UPGRADE** and click **OK**. The system immediately upgrades the kernel to the latest version.

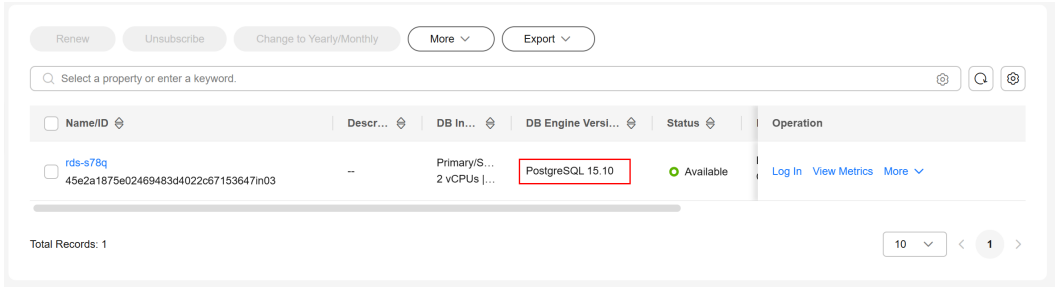
Figure 3-62 Confirming upgrade information



Step 6 Check the result.

During the upgrade, the instance status is **Upgrading minor version**. After the upgrade is complete, the instance status changes to **Available**, and the version is the latest kernel minor version.

Figure 3-63 Checking the new minor version



----End

3.6.2 Upgrading the Major Version of a DB Instance Using SQL Commands

Scenarios

You can upgrade the RDS for PostgreSQL major version to enjoy more functions and higher performance and security. Major version upgrades may introduce changes that are backward incompatible with existing versions and affect service running. Therefore, you need to test services on the target version before the upgrade.

In this section, the source instance indicates the DB instance that runs the source version, and the target instance indicates the DB instance that runs the target version.

RDS for PostgreSQL Version Description

- RDS for PostgreSQL v10 and later versions consist of a major version and a minor version. A major version upgrade refers to the upgrade of the major version, such as from 11.x to 12.x.
- Versions earlier than RDS for PostgreSQL v10 consist of two major versions and a minor version. A major version upgrade refers to the upgrade of the major versions, such as from 9.5.x to 9.6.x or from 9.x.x to 10.x.

Preparations

1. View information about the RDS for PostgreSQL DB instance to be upgraded.
   a. On the **Instances** page, click the DB instance to be upgraded.
   b. On the **Overview** page, view the region, AZ, VPC, subnet, and security group of the DB instance.
2. Prepare an ECS.

   To connect to a DB instance through an ECS, you must first create an ECS.

   The region, AZ, VPC, subnet, and security group of the ECS are the same as those of the RDS for PostgreSQL DB instance to be upgraded.
3. Install a PostgreSQL client on the ECS created in 2.

   For details, see **How Can I Install the PostgreSQL Client?**

   NOTE

   The version of the RDS for PostgreSQL client must be the same as that of the RDS for PostgreSQL instance. The RDS for PostgreSQL instance or client provides **pg_dump**, **pg_restore**, and **psql**.
4. **Connect to the source instance through psql** and run the following SQL statement on each database to obtain the used extension list:

   select extname from pg_extension;
5. Select a target version that contains all extensions based on the used extension list.

   For details about the extensions supported by different RDS for PostgreSQL versions, see **Supported Extensions**.
6. Create a parameter template that is compatible with the source version by referring to **Creating a Parameter Template**.
7. Create an RDS for PostgreSQL instance running the target version.
   - For details on how to create a DB instance, see **Buying a DB Instance**.
   - The region, AZ, VPC, subnet, and security group of the target instance are the same as those of the source instance.
8. On the ECS prepared in 2, **use psql to connect to the target instance** and check that the connection is successful.

Procedure

Perform the following operations on the prepared ECS.

Step 1 Use psql to connect to the source instance and run the following SQL statement to obtain the database list:

postgres=# \l

Step 2 Use psql to connect to the target instance and run the following SQL statement to check whether all databases obtained in **Step 1** exist on the target instance:

postgres=# \l

- If yes, go to **Step 3**.
- If no, run the following SQL statement to create databases that does not exist on the target instance and go to **Step 3**.

  postgres=# create database my_target_db;

  NOTE

  - The template databases template0 and template1 do not need to be migrated.
  - The postgres database is created by default and does not need to be migrated unless it stores service data.

Step 3 Use pg_dump to dump the source instance and use pg_restore to restore data to the target instance. Repeat **Step 3** to **Step 4** on each service database.

- For versions other than RDS for PostgreSQL 11, run the following dump command:

  pg_dump -Fc -v --host=source_IP --port=source_port --username=my_user --dbname=my_source_db | pg_restore -v --no-owner --host=target_IP --port=target_port --username=my_user --dbname=my_target_db

- For RDS for PostgreSQL 11, run the following dump command:

  pg_dump -Fc -v --host=source_IP --port=source_port -Ndbms_lob -Ndbms_output -Ndbms_random -Nsys -Nutl_raw -Npg_catalog --username=my_user --dbname=my_source_db | pg_restore -v --no-owner --host=target_IP --port=target_port --username=my_user --dbname=my_target_db

NOTE

- The login user using **pg_dump** must have the permission to access all objects in the database.
- The login user using **pg_restore** must have all operation permissions on the database.
- For details about how to grant permissions, see **GRANT**.
- If the **pg_dump** command uses the **-N** parameter, blobs will not be exported.
- If the pg_dump command uses the **-Fc** parameter, the exported file is in binary format. To export SQL files, use the **-Fp** parameter.

Step 4 After a database is migrated, test services on the target database to ensure that the services are running properly on it.

Step 5 Check that services are running properly on the target databases. Then, switch services to the target instance and delete the source instance.

----End

3.6.3 Upgrading the Major Version of a DB Instance on the Console

Scenarios

RDS for PostgreSQL allows you to upgrade the major version of your DB instance in either of the following methods:

- Upgrade without cutover: You can use it to test service compatibility of a new version. Upgrading a major version may cause compatibility issues. A compatibility test is strongly recommended. After the test is passed, perform an upgrade in cutover mode. An upgrade without workload cutover will not affect the original instance.
- Upgrade with cutover: During an upgrade, the original instance is set to read-only and workloads are interrupted for minutes. After the upgrade is complete, the original and new instances automatically exchange their virtual IP addresses and the application connection will be switched to the new instance. No changes need to be made to your application.

Constraints

- To use this function, **submit a service ticket** to request required permissions.
- Major version upgrades are available to the following versions:
  - RDS for PostgreSQL 9.5: 9.5.25 or later
  - RDS for PostgreSQL 9.6: 9.6.24 or later
  - RDS for PostgreSQL 10: 10.21 or later
  - RDS for PostgreSQL 12: 12.7 or later
  - RDS for PostgreSQL 13: 13.3 or later
  - RDS for PostgreSQL 14: 14.4 or later
  - RDS for PostgreSQL 15: 15.4 or later
  - Major version upgrades are unavailable to RDS for PostgreSQL 11 and Enhanced Edition.
- Due to OS restrictions, some instances do not support major version upgrades. To learn which versions your instance can be upgraded to, see the list of available versions on the **Major Version Upgrade** page.
- DR instances do not support major version upgrades.
- Before a major version upgrade, perform an upgrade check. If there is no successful upgrade check in the validity period, a major version upgrade is not allowed.

Extensions Unsupported for Major Version Upgrades

If any extension listed in **Table 3-22** is detected in the upgrade path during a major version upgrade pre-check, uninstall the extension before the upgrade is performed and reinstall it after the upgrade is complete. If you perform the upgrade without removing this extension, the upgrade will fail, or the extension cannot be used after the instance is upgraded.

Table 3-22 Extensions unsupported for major version upgrades

Source Version	Target Version	Extension That Can Cause an Upgrade Failure	Extension That Cannot Be Used After an Instance Upgrade
12	13	orafce, postgis_sfcgal	address_standardizer_data_us, pgaudit
	14	orafce, postgis_sfcgal	anon, pgaudit
	15	orafce, postgis_sfcgal	anon, pgaudit
	16	orafce, postgis_sfcgal, pgl_ddl_deploy	anon, pgaudit
13	14	-	anon, pgaudit, pg_stat_kcache
	15	-	anon, pgaudit, pg_stat_kcache
	16	pgl_ddl_deploy	anon, pgaudit, pg_stat_kcache
14	15	-	pgaudit, pg_stat_kcache
	16	pgl_ddl_deploy	pgaudit, pg_stat_kcache
15	16	pgl_ddl_deploy	pgaudit

Billing

The new instance generated after a major version upgrade is billed in pay-per-use mode. After confirming that the workloads are running stably, you can perform any of the following operations:

- Change the billing mode of the new instance to yearly/monthly. For details, see **Changing the Billing Mode from Pay-per-Use to Yearly/Monthly**.
- If the original instance is billed in pay-per-use mode, release the original instance to reduce fees. For details, see **Deleting a Pay-per-Use DB Instance or Read Replica**.
- If the original instance is billed in yearly/monthly mode, unsubscribe from it. For details, see **Unsubscribing from a Yearly/Monthly Instance**. Unsubscribing from a yearly/monthly instance before it expires may cause fee loss. Before an upgrade, learn about the unsubscription rules in **Unsubscribing from In-Use Resources**.

Precautions

- If the major version upgrade is complete and the application is cut over to the new instance but the workloads are incompatible with the new version, you need to roll back the upgrade. **Submit a service ticket** to remove the read-only status of the original instance. Then, you can continue to use the original instance.

NOTICE

The data added after an upgrade is complete will not be automatically synchronized to the original instance.

- After a major version upgrade is complete, a new DB instance is created. The original DB instance is still retained and billed. You can release the original instance when the workloads on the new instance run stably.
- After a major version upgrade, the audit logs, error logs, and slow query logs of the original instance are still stored in the original instance. You can only view the logs generated after the upgrade on the new instance.
- Read replicas do not support major version upgrades. If your DB instance has a read replica, the read replica will not be upgraded synchronously. You need to recreate one after a major version upgrade. For details, see **Creating a Read Replica**.
- If your DB instance has a DR instance, the DR instance will not be upgraded synchronously and the DR relationship will be interrupted. You need to recreate a DR instance of the new version after a major version upgrade.
- A major version upgrade has the following impacts:
  - If you upgrade your instance with service cutover, the instance will be set to read-only during the upgrade and services will be interrupted for minutes. Perform the upgrade during off-peak hours. If you upgrade your instance without service cutover, there is no impact on your services.

    NOTICE

    The **default_transaction_read_only** parameter controls the read-only settings. Before the upgrade, check whether any modification has been made to this parameter. If yes, the data inserted into the instance during the cutover will be lost after the upgrade.

  - After a major version upgrade:
    - Parameter modifications in the original instance are automatically synchronized to the new instance. For parameters that were not modified, default values for the new version are used.
    - If the original instance uses any parameter that is not supported by the new version, the parameter will be automatically deleted from the new instance.
    - If the value of any parameter in the original instance is not within the valid range configured in the new version, the new instance will use the default value defined in the parameter template of the new version.
- Upgrading a major version will not upgrade the versions of extensions. For details about supported extension versions, see **Supported Extensions**. If the new instance supports any extension of a later version, you can run **ALTER EXTENSION extension_name UPDATE TO 'new_version';** to update the extension, or uninstall and reinstall the extension of the latest version.

NOTICE

Certain extensions (such as postgis) will cause the upgrade task to fail. In this case, uninstall the extensions before performing a major version upgrade.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Major Version Upgrade**.

NOTE

If the **Major Version Upgrade** tab is not displayed, submit a service ticket to apply for required permissions.

Step 6 On the displayed page, select the target version, click **Check Now**, and wait for several minutes.

NOTE

- If extensions are installed on the instance after a successful upgrade check, an upgrade may fail due to compatibility issues. If this happens, perform a check again.
- If the upgrade check fails, you can view the check report by clicking **View Report** on the **Upgrade Checks** tab page and rectify the incompatibility based on the report.

Step 7 After the check is successful, click **Next**. Select "I understand Precautions for the Upgrade." and click **Upgrade Now**.

Step 8 Specify **Cutover** and **Collect Statistics**.

NOTE

During a major version upgrade, optimizer statistics are not automatically synchronized. Statistics need to be collected after the upgrade is complete.

- **Before cutover**: Service stability is ensured. If your instance has a large amount of data, the upgrade may take a longer time.
- **After cutover**: Faster upgrade is ensured. After the upgrade, accessing tables that no statistics have been generated for may cause inaccurate execution plans and even instance breakdowns during peak hours.

----End

Upgrade Check Reports and Upgrade Reports

If an upgrade check or an upgrade fails, you can analyze the causes based on the upgrade check report or upgrade report. The procedure is as follows:

1. View the **pg_upgrade_internal.log** file.

The **pg_upgrade_internal.log** file is the main log file of an upgrade check report or upgrade report. If the operation fails, check for errors in this file. Common errors are as follows:

- A list of problem libraries is in the file: loadable_libraries.txt

  It means there are extensions that are incompatible with the target version. They are listed in **loadable_libraries.txt**.

- A list of tables with the problem is in the file: tables_with_oids.txt

  It means there are tables that are created with the WITH OIDS clause and such tables are recorded in **tables_with_oids.txt**. The WITH OIDS clause is not supported by RDS for PostgreSQL 12 or later.

- Consult the last few lines of "pg_upgrade_server.log" for the probable cause of failure.

  It means that the target version failed to start during the upgrade check and you can check **pg_upgrade_server.log** for the causes.

- Consult the last few lines of "pg_upgrade_dump_xxxx.log" for the probable cause of failure.

  It means that pg_dump failed to back up data during the upgrade and you can check **pg_upgrade_dump_xxxx.log** for the causes.

2. Analyze causes based on the report items.

   - loadable_libraries.txt

     This item displays incompatible libraries, which usually correspond to incompatible extensions. Check the extensions listed in **loadable_libraries.txt** and determine whether to delete them. Delete them before the upgrade if you are sure that the deletion will not affect workloads.

   - tables_with_oids.txt

     This item displays tables created with the WITH OIDS clause. Check the tables listed in **tables_with_oids.txt** and evaluate whether the workload code depends on the OIDs. If stripping OIDs from the tables does not affect workloads, run the following SQL statement:

     ```
     ALTER TABLE {table_name} SET WITHOUT OIDS;
     ```

   - pg_upgrade_server.log

     Check the last several lines of the **pg_upgrade_server.log** file. If an error similar to the following appears, the extension displayed in this error does not exist in the target version. Delete it from **shared_preload_libraries** as required and then perform the upgrade.

     ```
     FATLA: could not access file "xxx": No such file or directory.
     ```

     Example:

     ```
     FATLA: could not access file "pg_pathman": No such file or directory.
     ```

   - pg_upgrade_dump_xxxx.log

     - Check the last several lines of **pg_upgrade_dump_*xxxx*.log**. If an error similar to the following is displayed, there are too many tables in the current instance. In this case, increase the value of **max_locks_per_transaction** and perform the upgrade again.

       ```
       pg_dump: error: query failed: ERROR: out of shared memory
       HINT: You might need to increase max_locks_per_transaction.
       pg_dump:error: query was: LOCK TABLE "xxx"."xxx" IN ACCESSSHARE MODE
       ```


- Check the last several lines of **pg_upgrade_dump_xxxx.log**. If an error similar to the following is displayed, the **pgl_ddl_deploy** extension exists in the current instance. This extension is incompatible with the target version, so the upgrade failed. Check whether there are any other incompatible third-party extensions in the instance based on **Extensions Unsupported for Major Version Upgrades** (some incompatible third-party extensions cannot be identified through an upgrade check). Delete them as required and then perform the upgrade.

  ```
  pg_restore: error: could not execute query: ERROR: could not find function "xxx" in file xxx
  Command was: CREATE FUNCTION "pgl_ddl_deploy"."xxx"
  ```

3.7 Instance Management

3.7.1 Instance Lifecycle

3.7.1.1 Buying a Same DB Instance as an Existing DB Instance

Scenarios

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

NOTE

- You can buy 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Buy 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 how to buy an RDS for PostgreSQL DB instance, see **Buying a DB Instance**.

Step 6 Confirm the instance specifications.

- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**.

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 **Instances** page.

----End

3.7.1.2 Stopping an Instance

Scenarios

If you use DB instances only for routine development, you can temporarily stop pay-per-use instances to save money. You can stop an instance for up to 15 days.

Billing

After a DB instance is stopped, the VM where the DB instance is located is no longer billed. Other resources, including EIPs, storage resources, and backups, are still billed.

Constraints

- Only cloud SSD and extreme SSD pay-per-use instances can be stopped. RDS instances in a DCC cannot be stopped.
- If you stop a primary instance, read replicas (if there are any) will also be stopped. They are stopped for up to 15 days. You cannot stop a read replica without stopping the primary instance.
- A stopped instance cannot be deleted through the console.
- Stopping a DB instance will also stop its automated backups. After the DB instance is started, a full backup is automatically triggered.
- If you do not manually start your stopped DB instance after 15 days, your DB instance is automatically started during the next maintenance window. For details about the maintenance window, see **Changing the Maintenance Window**. To start a DB instance, see **Starting an Instance**.
- A stopped pay-per-use instance may fail to start due to insufficient ECS resources. If this happens, try again later. If you need assistance, **submit a service ticket**.

Procedure

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

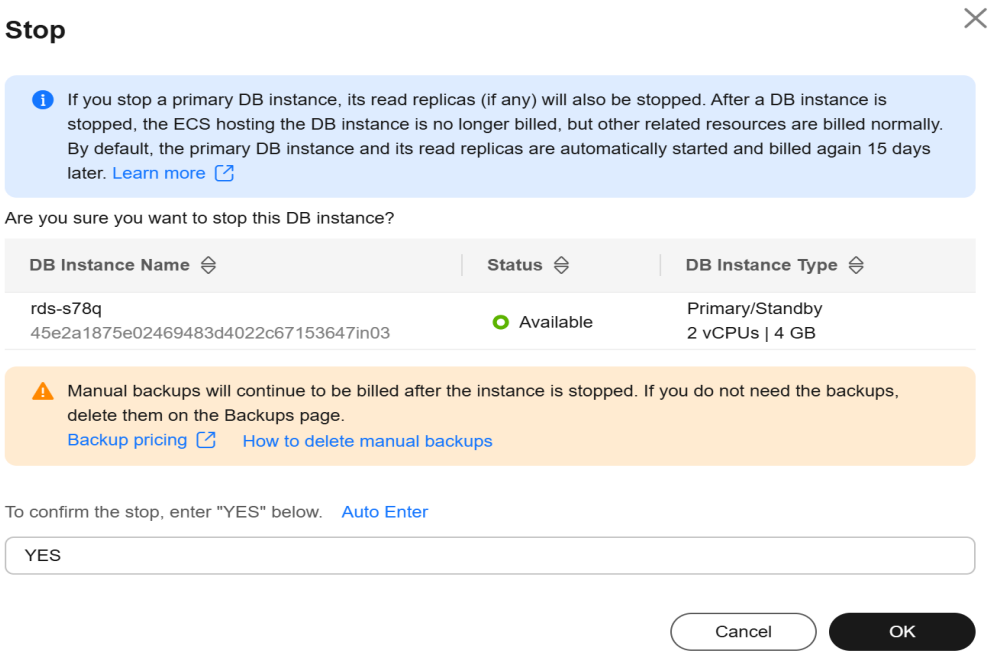
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary instance that you want to stop and choose **More** > **Stop** in the **Operation** column.

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

Figure 3-64 Stopping an instance



Step 6 Refresh the instance list and view the status of the instance. If the status is **Stopped**, the instance is stopped successfully.

----End

3.7.1.3 Starting an Instance

Scenarios

You can stop your instance temporarily to save money. After stopping your instance, you can restart it to begin using it again.

Billing

After a DB instance is started, the VM where the DB instance is located is billed again.

Constraints

- If you start a primary instance, read replicas (if there are any) will also be started.
- When a stopped DB instance is started, a full backup is automatically triggered.
- Only instances in **Stopped** state can be started.
- A stopped pay-per-use instance may fail to start due to insufficient ECS resources. If this happens, try again later. If you need assistance, **submit a service ticket**.

Procedure

Step 1 Log in to the management console.

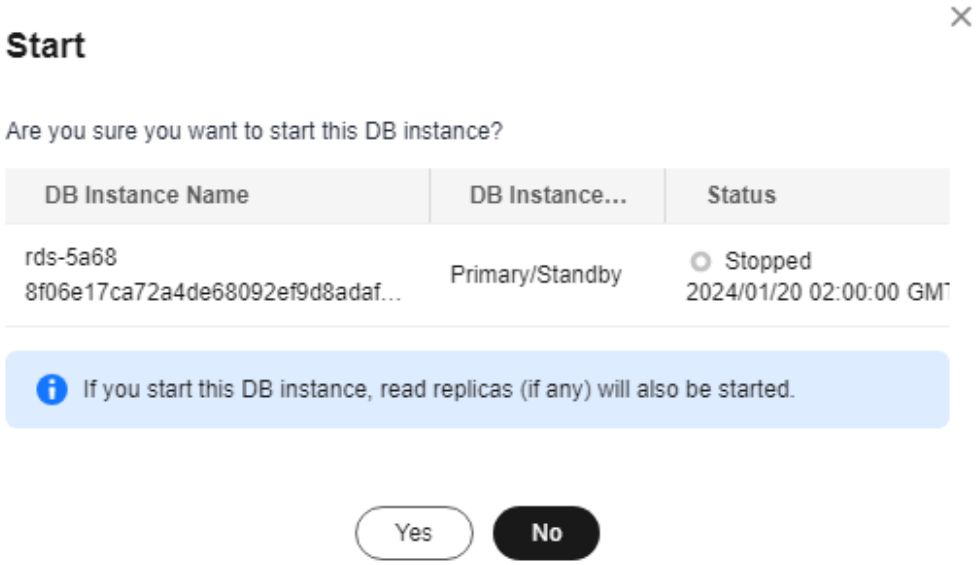
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary instance that you want to start and choose **More** > **Start** in the **Operation** column.

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

Figure 3-65 Starting an instance



Step 6 Refresh the instance list and view the status of the instance. If the status is **Available**, the instance is started successfully.

----End

3.7.1.4 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.
- An instance cannot be rebooted if its storage is full.

- 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.

Rebooting a DB Instance or Read Replica

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **Instances** page to go to the **Overview** 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 the check box and click **OK**.

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

Rebooting DB Instances or Read Replicas in Batches

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 Choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select one or more DB instances or read replicas (maximum: 50) to be rebooted and choose **More** > **Reboot** above the DB instance list.

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

3.7.1.5 Selecting Displayed Items

Scenarios

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

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

- **Table Text Wrapping**: If you enable this function, excess text will move down to the next line.
- **Operation Column**: If you enable this function, the **Operation** column is always fixed at the rightmost position of the table.
- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Disk Encryption** (submit a service ticket to apply for required permissions), **Billing Mode**, **Floating IP Address**, **Private Domain Name**, **IPv6 Address**, **Read/Write Splitting Address**, **Proxy ID**, **Enterprise Project**, **Created**, **Database Port**, **Storage Type**, **Tags**, and **Operation**.

----End

3.7.1.6 Exporting DB Instance Information

Scenarios

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

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click **Export** above the DB instance list. By default, information about all DB instances is exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, DB instance tag, enterprise project, or floating IP address, or select the DB instances to be exported, and click **Export** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

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

----End

3.7.1.7 Deleting a Pay-per-Use DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas billed on the pay-per-use basis as required on the **Instances** page. (To delete DB instances or read replicas billed on the yearly/monthly basis, you need to unsubscribe from the order. For details, see **Unsubscribing from a Yearly/Monthly Instance**.)

Billing

- You will not be billed for the instances that were not successfully created.
- If you delete a pay-per-use DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be deleted.
- A stopped instance cannot be deleted through the console.
- 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 **rebuild a DB instance** that was deleted up to 7 days ago from the recycle bin.
- You can use a manual backup to restore a DB instance. For details, see **Restoring a DB Instance from Backups**.

Deleting a Pay-per-Use DB Instance

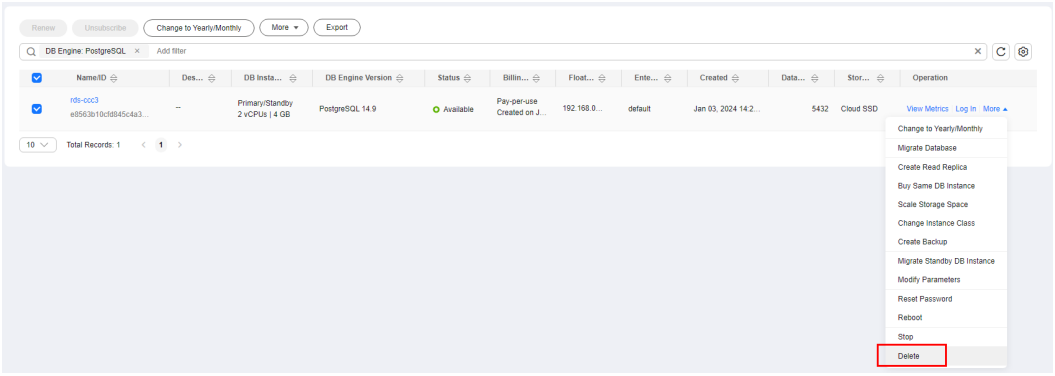
Step 1 **Log in to the management console**.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.

Figure 3-66 Deleting a DB instance



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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Pay-per-Use Read Replica

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

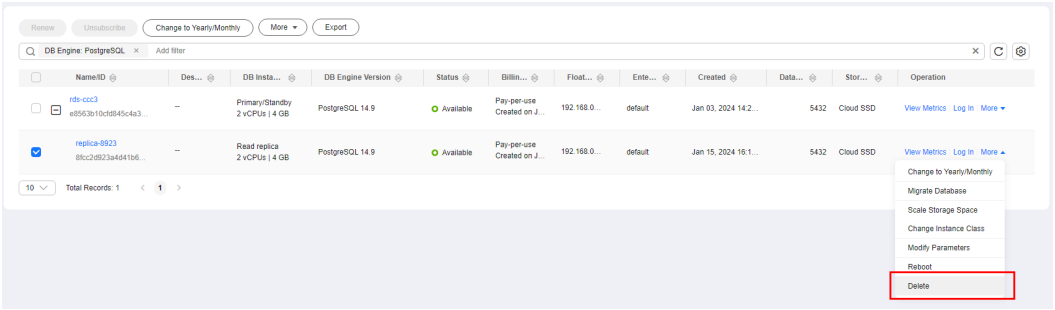
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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.

Figure 3-67 Deleting a read replica



Step 6 In the displayed dialog box, click **Yes**.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 8 Refresh the DB instance list later to check that the deletion is successful.

----End

3.7.1.8 Recycling a DB Instance

Scenarios

RDS allows you to move unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances to the recycle bin. You can rebuild a DB instance that was deleted up to 7 days ago from the recycle bin.

If resources are not renewed after expiration, you can rebuild DB instances from the recycle bin to restore data.

Constraints

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

- After you submit a deletion request for your DB instance, a full backup will be performed. You can rebuild the DB instance only after the full backup is complete.

Modifying Recycling Policy

NOTICE

Instances in the recycle bin are retained for 7 days by default. 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.

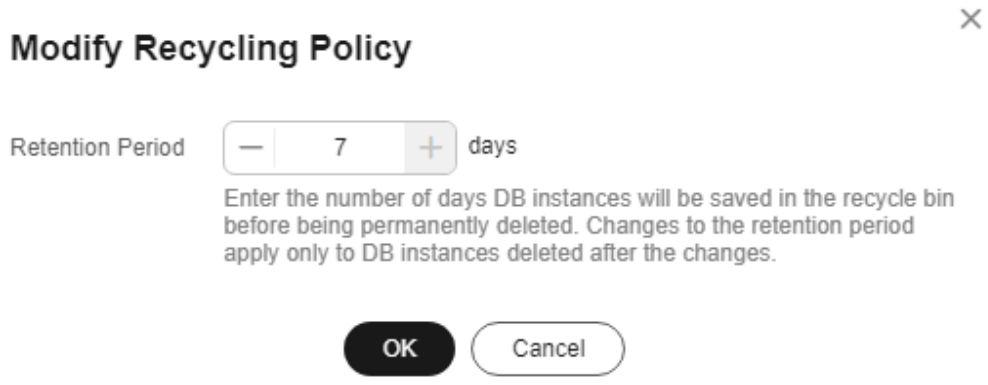
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances to 1 to 7 days.

Step 6 Then, click **OK**.

Figure 3-68 Modifying the recycling policy



----End

Rebuilding a DB Instance

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

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 On the **Recycle Bin** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.

Step 6 On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see **Restoring a DB Instance from Backups**.

----End

3.8 Instance Modifications

3.8.1 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **DB Instance Name**, click to edit the instance name.

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

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

Step 5 View the results on the **Overview** page.

----End

3.8.2 Changing a DB Instance Description

Scenarios

After a DB instance is created, you can add a description.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance you wish to edit the description for and click in the **Description** column to make your modification. When you are finished, click **OK**.

Alternatively, click the instance name to go to the **Overview** page. Under **Description**, click to edit the instance description.

NOTE

The DB instance description can include up to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.).

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

Step 5 View the results on the **Overview** page.

----End

3.8.3 Changing the Replication Mode

Scenarios

RDS allows you to change the replication mode between primary and standby DB instances. Data can be asynchronously or synchronously replicated from the primary instance to the standby instance.

- **Asynchronous** (default): When an application writes data to the primary instance, the primary instance returns a response to the application immediately without waiting for the standby instance to receive logs.
  - Advantages: Asynchronous replication involves low overhead and ensures that write operations are not blocked during a failover of your primary/standby instances.
  - Disadvantages: In rare cases, replication is delayed between the primary and standby instances, and data may be lost after the failover.
- **Synchronous**: When an application writes data to the primary instance, the primary instance returns a response to the application only after the standby instance receives logs (which are flushed to the disk).

  - Advantages: Data remains strongly consistent between the primary and standby instances, and no data loss occurs after a failover.
  - Disadvantages: Synchronous replication involves high overhead and causes write operations to be blocked when the primary or standby instance is faulty.

NOTE

- Asynchronous replication is recommended for applications requiring a guarantee of high availability.
- Synchronous replication is recommended for applications that require strong data consistency and can tolerate a short-time blocking of write operations.
- Write operations refer to non-SELECT operations, such as DDL and DML.

Procedure

Step 1 Log in to the management console.

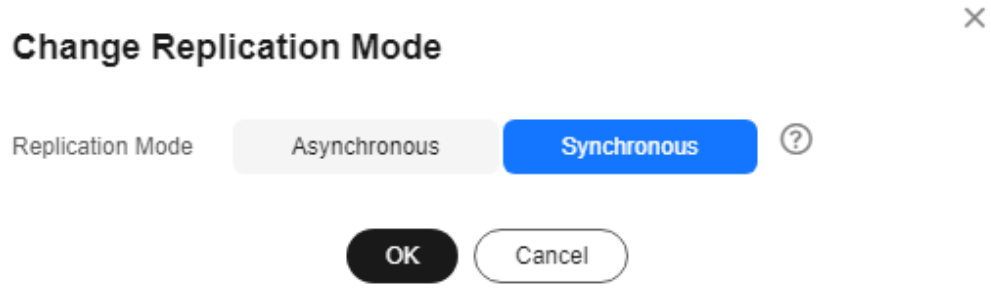
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 On the displayed **Overview** page, find **Replication Mode** and click **Configure** under it. In the displayed dialog box, select a mode and click **OK**.

Figure 3-69 Changing the replication mode



Step 6 On the **Overview** page, check for the new replication mode.

----End

3.8.4 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose

highest priority is data consistency. In extreme scenarios, there may be a small amount of data lost if your instance uses asynchronous replication.

- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Constraints

The failover priority cannot be changed when the DB instance is stopped or its instance class is being changed.

Procedure

Step 1 Log in to the management console.

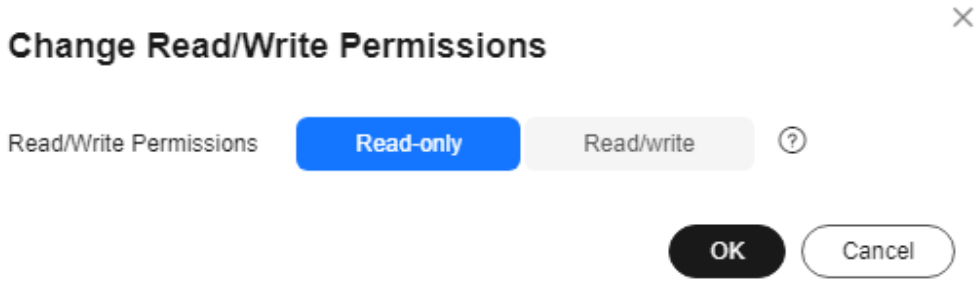
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 On the displayed **Overview** page, find **Failover Priority** and click **Configure** under it. In the displayed dialog box, select a priority and click **OK**.

Figure 3-70 Changing the failover priority



Step 6 View the results on the **Overview** page.

----End

3.8.5 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPUs and memory) of a DB instance as required.

Constraints

- You can change the DB instance class only when your account balance is greater than or equal to $0 USD.
- The instance class can be changed only when the DB instance is available.
- The instance class of a DR instance cannot be changed.
- A DB instance cannot be deleted when its instance class is being changed.
- You can scale up or down the compute and memory capacity of RDS for PostgreSQL DB instances as required.
- If a DB instance has a read replica, the new instance class must be no larger than the read replica class. When changing the read replica class, ensure that the selected class is no smaller than the DB instance class.

  To change a read replica to an instance class smaller than that of the primary instance, you need to **submit a service ticket** to apply for required permissions.
- After the instance class is changed, some parameters are automatically changed to the default values defined in the new instance class. The parameters are **max_worker_processes**, **max_wal_senders**, **max_prepared_transactions**, and **max_locks_per_transaction**.
- After the instance class is changed, the value of **max_connections** is the larger one between the default value defined in the new instance class and the value specified in the current instance.
- After you change instance classes, the DB instances will be rebooted and services will be interrupted. You are advised to change instance classes during off-peak hours.
- The time required for changing an instance class (during off-peak hours) is as follows:
  - This process takes 5 to 15 minutes.
  - When you are changing an instance class, service downtime only occurs during the primary/standby switchover. The duration of the downtime varies based on the replication delay and the number of temporary files.
  - If the change takes an extended period of time, **submit a service ticket**.

Billing

Table 3-23 Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Instance class upgrade	After an instance class is upgraded, the new instance class takes effect in the original usage period. You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are subject to the price displayed on the console. Suppose you purchased a one-month RDS for PostgreSQL 14 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The instance price was $59.56 USD per month. On June 15, 2023, you changed the instance class to 4 vCPUs | 8 GB. The instance price became $121.56 USD per month. **Price difference of upgrade = Price for the new instance class × Remaining period - Price for the original instance class × Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Price difference of the upgrade = $121.56 USD x 0.5 - $59.56 USD x 0.5 = $31 USD

Billing Mode	Operation	Impact on Fees
	Instance class downgrade	After an instance class is downgraded, the new instance class takes effect in the original usage period. RDS refunds the difference in price based on the remaining period. The following prices are for reference only. The actual prices are subject to the price displayed on the console. Suppose you purchased a one-month RDS for PostgreSQL 14 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The instance price was $59.56 USD per month. On June 15, 2023, you changed the instance class to 2 vCPUs | 4 GB. The instance price became $50.56 USD per month. **Refunded fees = Price for the original instance class × Remaining period - Price for the new instance class × Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and refunded fees are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Refunded fees = $59.56 USD x 0.5 - $50.56 USD x 0.5 = $4.5 USD
Pay-per-use	Instance class upgrade	After an instance class is changed, the new instance class is billed by hour. For details, see **Product Pricing Details**.
	Instance class downgrade	

Parameter Changes

After the instance class is changed, RDS will change the values of the following parameters accordingly:

- shared_buffers
- max_connections
- maintenance_work_mem
- effective_cache_size

For RDS for PostgreSQL 11 and later versions, in addition to the preceding parameters, the values of the following parameters will also be changed:

- max_prepared_transactions
- max_wal_size
- work_mem

Procedure

Step 1 Log in to the management console.

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

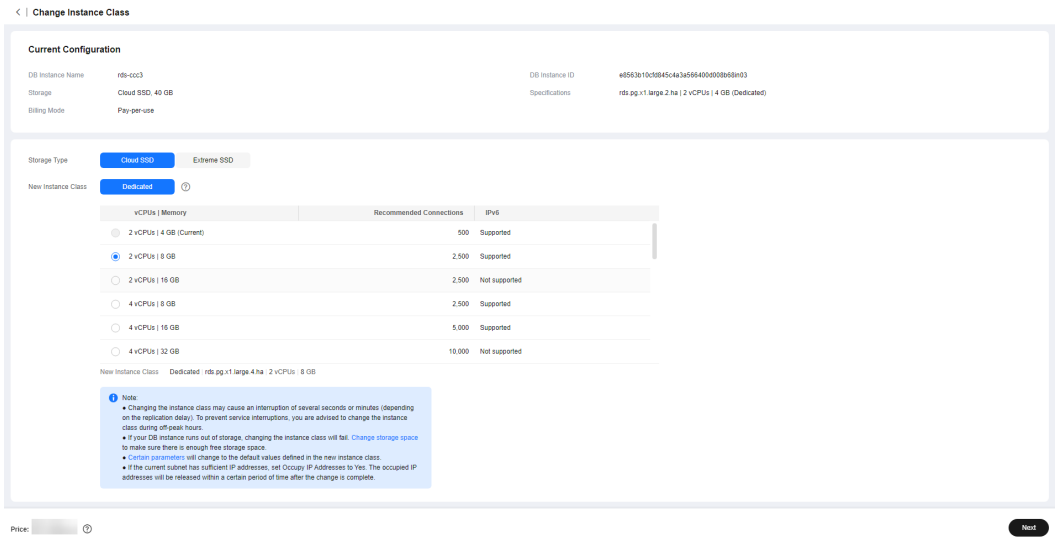
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **Instance Class**, click **Configure**.

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

Figure 3-71 Changing a DB instance class



DB instances in a DCC only support the general-enhanced instance class.

Step 6 Confirm the specifications.

- If you need to modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.

  To view the cost incurred by the DB instance class change, choose **Billing & Costs** > **Bills** in the upper right corner.
- For yearly/monthly DB instances:

- If you intend to scale down the DB instance class, click **Submit**.

  The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
- If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 7 View the DB instance class change result.

Return to the **Instances** page and view the instance status. During the change period, the instance status is **Changing instance class**. After a few minutes, click the DB instance and view its instance class on the displayed **Overview** page to check that the change is successful.

----End

3.8.6 Changing a Storage Type

Scenarios

If the storage type of your RDS DB instance does not match your workloads, you can change it as needed.

Constraints

- To change a storage type, **submit a service ticket** to request required permissions.
- Both ultra-high I/O and cloud SSD can be changed to extreme SSD.
- If the storage type of a read replica is different from that of its associated DB instance, the data synchronization may be affected. To change the storage type of a DB instance, change that of its read replica (if any) first to ensure that the storage type of the read replica is the same as that of the DB instance.
- During the storage type change, operations such as changing the instance class and scaling up storage space cannot be performed on the instance.
- Changing the storage type may affect the disk performance, so you need to change the type during off-peak hours.
- Changing the storage type may take several minutes, hours, or even days. The time required depends on the throughput, storage space, and original storage type. This operation cannot be paused.
- In rare cases, the change may fail due to resource problems. If this happens, you are advised to perform the change again.
- Due to **Elastic Volume Service (EVS) constraints**, there can be a maximum of 10 EVS disks with their types being changed at the same time. You may have to wait in a queue when you change your storage type.

Procedure

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

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

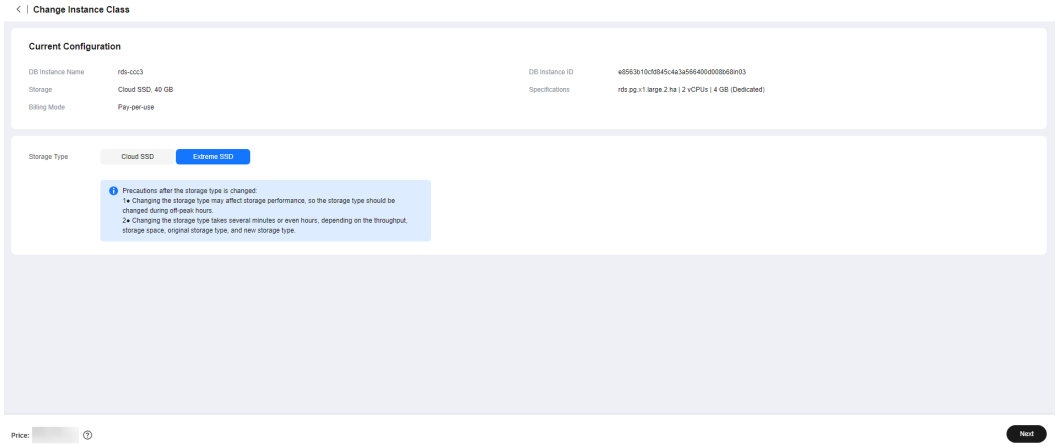
Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **Instance Class**, click **Configure**.

Step 5 On the displayed page, select a new storage type and click **Next**.

Figure 3-72 Changing the storage type



Step 6 Confirm the new storage type.

- To modify your settings, click **Previous**.
- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**. The change starts only after the payment is successful.

Step 7 Check the result.

Return to the **Instances** page. After a few minutes, click the instance name and check the new storage type on the displayed **Overview** page.

----End

3.8.7 Scaling Storage Space

Scenarios

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

When the storage usage is greater than or equal to 97%, the DB instance status changes to **Storage full**. Your connection to the instance is intermittently interrupted and the instance becomes read-only. When this happens, data cannot be written to the instance.

A 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** metric to learn about the storage usage in a timely manner. For details, see **Setting Alarm Rules**.

During the scale-up period, services are not interrupted.

Constraints on Scale-up

- You can scale up storage space only when your account balance is greater than or equal to $0 USD.
- 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.
- A DB instance cannot be deleted during scale-up.
- The storage space of a DB instance can be scaled up or down.
- If you scale up a DB instance with the disk encrypted, the expanded storage space will be encrypted using the original encryption key.

Constraints on Scale-down

- The storage space of a DB instance can be scaled only when the instance is **Available**.
- Only instances whose storage type is cloud SSD or ultra-high I/O can be scaled down.
- If the specifications of a read replica are less than those of the primary instance, scaling down the storage space of the primary instance may fail. It is recommended that the specifications of read replicas be no less than those of the primary instance.
- Scaling down storage space of a DB instance during peak hours may fail due to insufficient storage space. Scale down storage space during off-peak hours.
- To restore from a backup created before scale-down to an existing instance, select an instance whose storage is at least equal to that of the original instance before the scale-down.
- DB instances with snapshot backup enabled cannot be scaled down.
- During scale-down, 40 GB space is reserved to prevent the disk from becoming read-only, so your DB instance can be scaled down only when its available storage space is at least 40 GB.
- For primary/standby instances, scaling down the primary instance will cause the standby instance to also be scaled down.
- A DB instance cannot be deleted during scale-down.
- If you scale down a DB instance with the disk encrypted, the new storage space will be encrypted using the original encryption key.
- The time required for storage scale-down (during off-peak hours) is as follows:
  - The time required depends on how much data there is in your instance. More data means more time.
  - If the scale-down takes an extended period of time, **submit a service ticket**.

Billing

Table 3-24 Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Storage scale-up	You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are subject to the price displayed on the console. Suppose you purchased a one-month RDS for PostgreSQL 14 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The unit price of storage space is $0.214 USD/GB per month. On June 15, 2023, you scaled up the storage by 60 GB. The total storage after scale-up is 100 GB. **Price difference = Scale-up volume x Unit price x Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Price difference = 60 GB x $0.214 USD x 0.5 = $6.42 USD
Yearly/ Monthly	Storage scale-down	After the scale-down is successful, RDS refunds the difference in price based on the new storage space and the remaining period. The calculation method is the same as that for scaling up storage space.
Pay-per-use	Storage scaling	The new storage space is billed by hour. For details, see **Product Pricing Details**.

Scaling the Storage Space of a DB Instance

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Or use either of the following methods:

- Click the instance name to go to the **Overview** page. In the **Storage & Backup** area, click **Scale Storage Space**.
- If the instance storage is full, locate the instance on the **Instances** page and click **Scale** in the **Status** column.

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

You can increase or decrease the storage by at least 10 GB. Enter a value that is a multiple of 10. The instance supports a storage space range from 40 GB to 4,000 GB.

The final storage space range on the console may differ slightly. It varies with the storage usage of the instance.

Step 6 Confirm the information.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 7 Check the result.

Scaling storage space takes 3-5 minutes. During this period, the status of the DB instance on the **Instances** page will be **Scaling up** or **Scaling down**. After a while, click the instance name and check that the new value for storage space appears on the **Overview** page.

----End

Scaling the Storage Space of a Read Replica

Scaling the storage space of a read replica does not affect that of the primary DB instance. You can separately scale read replicas to meet service requirements. New storage space of read replicas after scaling 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Or use either of the following methods:

- Click the read replica name to go to the **Overview** page. In the **Storage & Backup** area, click **Scale Storage Space**.
- If the storage of the read replica is full, locate the read replica on the **Instances** page and click **Scale** in the **Status** column.

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 or down only by a multiple of 10 GB. The allowed minimum and maximum storage spaces are 40 GB and 4,000 GB, respectively.

The storage space range varies with the storage usage of the read replica. For details, see the information displayed on the console.

Step 6 Confirm the information.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for a pay-per-use read replica or click **Pay Now** for a yearly/monthly read replica.

Step 7 Check the result.

Scaling storage space takes 3-5 minutes. During this period, the status of the read replica on the **Instances** page will be **Scaling up** or **Scaling down**. After a while, click the read replica name and check the new storage space on the displayed **Overview** page to verify that the scaling is successful.

----End

3.8.8 Configuring Storage Autoscaling

Scenarios

With storage autoscaling enabled, when RDS detects that you are running out of database space, it automatically scales up your storage.

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

You can enable storage autoscaling in either of the following ways:

- Enable this function when you create a DB instance. For details, see **Buy a DB Instance**.
- Enable this function after you create a DB instance. See the operations provided in this section.

Constraints

- You can enable storage autoscaling only when your account balance is greater than or equal to $0 USD.
- The storage space can be scaled up automatically only when your instance status is **Available** or **Storage full**.

- Storage autoscaling for RDS for PostgreSQL DB instances is supported only for cloud SSD and extreme SSD storage types. For details about storage types, see **DB Instance Storage Types**.
- The maximum allowed storage is 4,000 GB.
- For a primary/standby DB instance, autoscaling the storage for the primary node will also autoscale the storage for the standby node.
- Storage autoscaling is unavailable when the DB instance is in any of the following statuses: changing instance class, upgrading a minor version, migrating the standby DB instance, and rebooting.
- If a yearly/monthly DB instance has pending orders, it will not be autoscaled.

Procedure

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

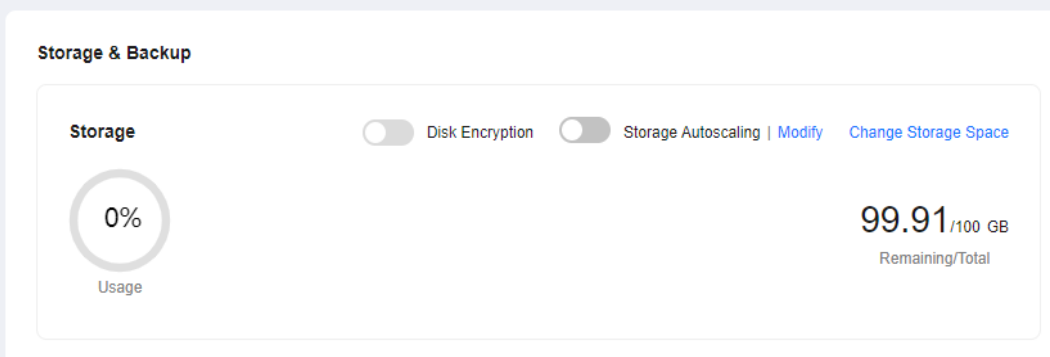
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance or read replica name (click in front of a DB instance to locate its read replica).

Step 5 In the **Storage & Backup** area, toggle on the **Storage Autoscaling** switch.

Figure 3-73 Storage space



Step 6 In the displayed dialog box, set the following parameters:

Figure 3-74 Configuring autoscaling

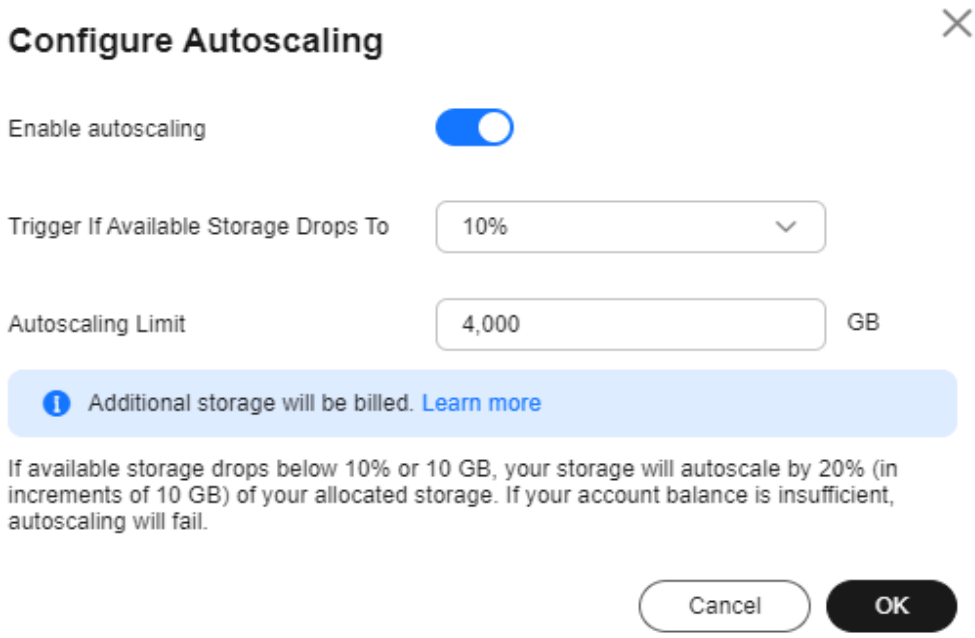


Table 3-25 Parameter description

Parameter	Description
Enable autoscaling	If you select this option, autoscaling is enabled.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold (10%, 15%, or 20%) or 10 GB, autoscaling is triggered.
Autoscaling Limit	The default value range is from 40 GB to 4,000 GB. The limit must be no less than the storage of the DB instance.

Step 7 Click **OK**.

----End

3.8.9 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 once or twice. Ensure that your applications support automatic reconnection.
- If you set the time for changing your instance class to the maintenance window, the change will fail when the instance is frozen or any of the following operations is being performed during this time:

  Changing the instance class, automatically backing up the instance, creating a manual backup, scaling up or down storage, changing the storage type, changing the billing mode from pay-per-use to yearly/monthly, stopping the instance, rebooting the instance, creating a read replica, changing the database port, binding an EIP, unbinding an EIP, changing the floating IP address, modifying instance parameters, changing the instance type from single to primary/standby, resetting the administrator password, establishing a DR relationship, upgrading the major version, upgrading the minor version, and switching between primary and standby instances.

Procedure

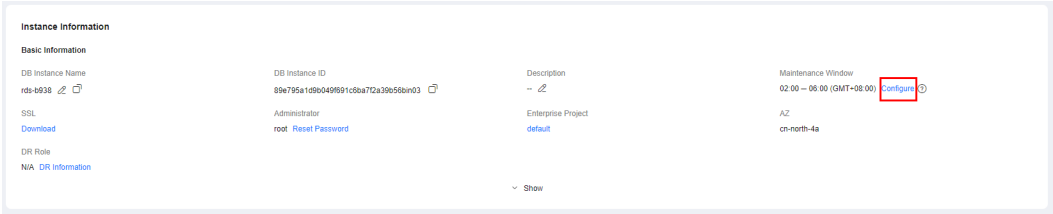
Step 1 **Log in to the management console**.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

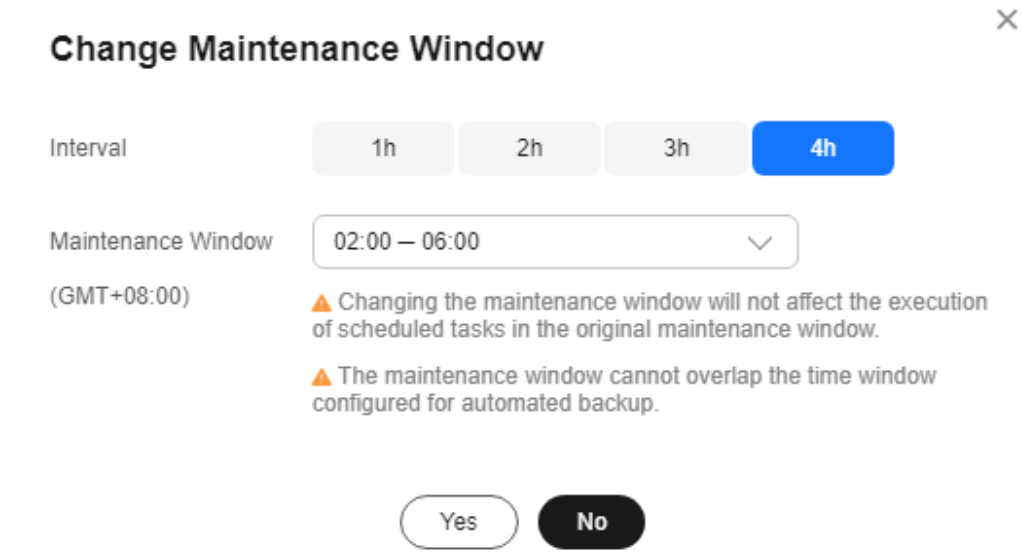
Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page. Under **Maintenance Window**, click **Configure**.

Figure 3-75 Changing the maintenance window



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

Figure 3-76 Changing the maintenance window



NOTE

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

----End

3.8.10 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 affects network and disk I/O operations of 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.
- Anti-affinity deployment is supported for primary/standby DB instances to prevent the entire instance unavailability due to the failure of a single host.

Precautions

RDS single DB instances can be changed to primary/standby DB instances, but not the other way around. You can use Data Replication Service (DRS) or the export and import tool of the client to migrate data from primary/standby DB instances to single DB instances.

Changing a single-node instance to primary/standby does not change its networking information, including the VPC, subnet, security group, private IP address, private domain name, and database port.

Billing

Table 3-26 Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Changing a single DB instance to primary/ standby	You need to pay for the difference in price based on the remaining period. The following prices are for reference only. The actual prices are subject to the price displayed on the console. Suppose you purchased a one-month RDS for PostgreSQL 14 single DB instance (instance class: general-purpose, 2 vCPUs | 8 GB; storage: cloud SSD, 40 GB) in CN-Hong Kong on June 1, 2023. The instance price was $59.56 USD per month. On June 15, 2023, you changed the instance type from single to primary/standby. The instance price became $155.69 USD per month. **Price difference = Price for the primary/ standby instance × Remaining period - Price for the single instance × Remaining period** The remaining period is the remaining days of each calendar month divided by the maximum number of days in each calendar month. In this example, the remaining period and price difference are calculated as follows: Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 Price difference = $155.69 USD x 0.5 - $59.56 USD x 0.5 = $48.06 USD
Pay-per-use	Changing a single DB instance to primary/ standby	The primary/standby DB instance is billed by hour. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

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**.

It is recommended that the standby AZ be different from the primary AZ to provide failover and high availability.

Step 6 Check the instance status on the **Instances** 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

3.8.11 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.

Precautions

For primary/standby switchovers or other operations involving switchovers, such as instance class changes and minor version upgrades:

If there is any slow SQL statement still running during a primary/standby switchover, the slow SQL connection may be suspended (new connections and other idle connections are not affected). After a period of time, an error is returned to the client. The time when the error is reported is related to the settings of TCP parameters such as **keepalives_idle**, **keepalives_interval**, and **keepalives_count** on the client. For details, see the **official documentation**.

Constraints

A manual switchover does not change the connection information of the DB instance, including its VPC, subnet, security group, floating IP address, private domain name, and database port.

You can switch the primary and standby instances only when the following conditions are met:

- The primary/standby instance is running properly.

- The primary/standby replication is normal.
- The replication delay is less than 5 minutes, and the data on the primary and standby instances is consistent.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target primary/standby instance name to go to the **Overview** page.

Step 5 Under **DB Instance Type**, click **Switch**.

NOTICE

A primary/standby switchover may cause a brief interruption of several seconds or minutes (depending on the replication delay). If transaction logs are generated at a speed higher than 30 MB/s, services will probably be interrupted for several minutes. To prevent traffic congestion, perform a switchover during off-peak hours.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the ***Identity and Access Management User Guide***.

Step 7 In the displayed dialog box, click **OK**.

Step 8 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, 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

3.8.12 Changing the AZ of a Standby DB Instance

Scenarios

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

For details about regions and AZs, see **Regions and AZs**.

Constraints

Only when a DB instance is available and its storage is not full, you can migrate its standby instance to another AZ.

Precautions

- Before the migration, check the resource usage of your DB instance to prevent resource overload from affecting workloads and the migration progress.
- During the migration, if there is a large amount of data being written to the primary instance (in synchronous replication mode), the write operations may be blocked after the migration.
- DDL operations will be suspended during the migration. To prevent service interruption, perform the migration during off-peak hours.
- After the migration, check your workloads and verify data.
- The migration duration is in direct proportion to the instance data volume.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 Check the DB instance status on the **Instances** 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 **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**.
- On the **Overview** page, find **AZ** and check the AZ hosting the standby DB instance.

----End

3.8.13 Updating the OS of a DB Instance

To improve database performance and security, the OS of an RDS for PostgreSQL instance needs to be updated timely.

Every time you upgrade the kernel version of your instance, RDS for PostgreSQL determines whether to update the OS and selects the right cold patch to upgrade the OS if necessary.

Updating the OS does not change the DB instance version or other information.

In addition, RDS for PostgreSQL installs hot patches as required to fix major OS vulnerabilities within the maintenance window you specified.

3.9 Data Backups

3.9.1 Introduction to Backups

What Are Database Backups?

RDS for PostgreSQL automatically creates backups for DB instances during the backup time window. The backups are stored based on a preset retention period (1 to 732 days).

A backup file is generated each time a backup is complete. If the instance fails or data is damaged, you can use the backup file to restore the instance, ensuring data reliability.

Backup Types

RDS for PostgreSQL supports multiple backup types. For details, see **Backup Types**.

- Full backup: A full backup is to back up all data, even if no data has changed since the last backup.

  Full backup can be initiated manually or automatically.
- Incremental backup: Incremental backups refer to Write-Ahead Logging (WAL) backups. RDS performs an incremental backup every 5 minutes.

Where Data Is Backed Up

- Single-node instance

  A single-node architecture, which is more cost-effective than mainstream primary/standby DB instances. After a backup is triggered, data is backed up from the primary instance and stored as a package on OBS. The backup does not take up storage space of the instance.
- Primary/standby instance

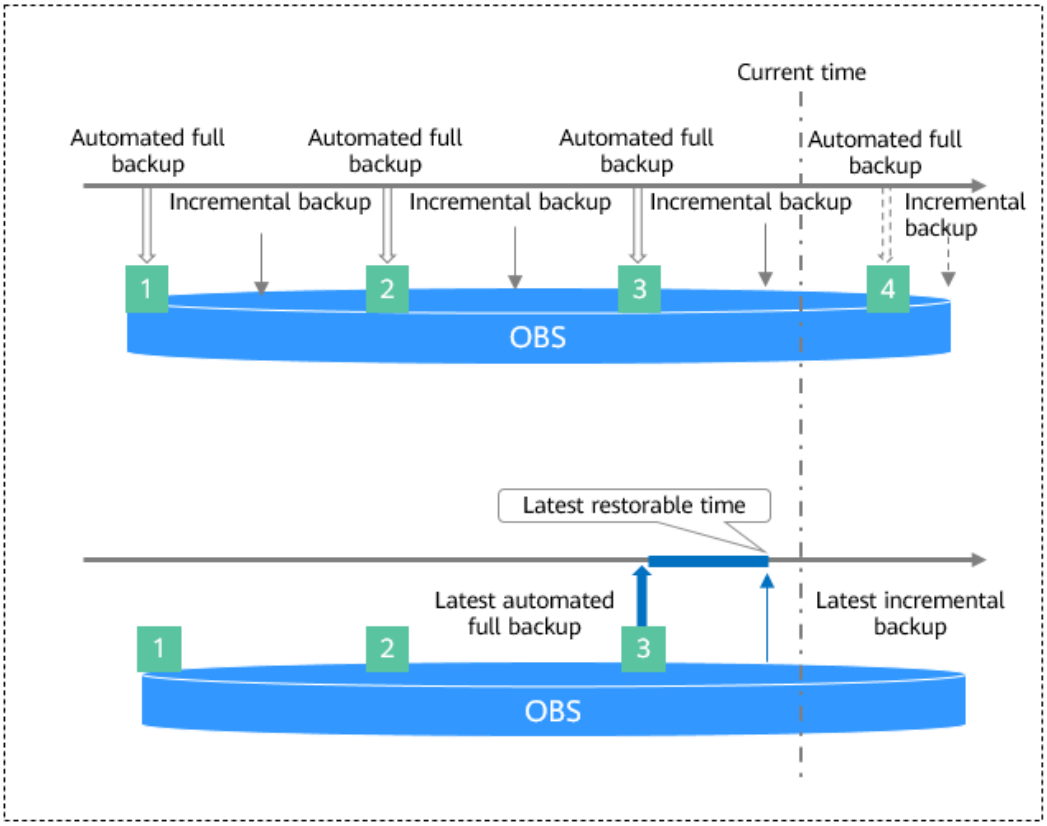
  An HA architecture. In a primary/standby pair, each instance has the same instance class. After a backup is triggered, data is backed up from the primary instance and stored as a package on OBS. The backup does not take up storage space of the instance.

  If a database or table in the primary instance is maliciously or mistakenly deleted, the database or table in the standby instance will also be deleted. In this case, you can only use backups to restore the deleted data.

How Data Is Backed Up

RDS for PostgreSQL automated backup is enabled by default and cannot be disabled. RDS for PostgreSQL performs automated full backups based on the time window and interval specified in the backup policy. It also backs up data modifications made after the most recent automated full or incremental backup every five minutes. When you restore an instance to a point in time, the most recent full backup will be downloaded from OBS for restoration. After the restoration is complete, incremental backups will be replayed to the specified point in time.

Figure 3-77 How data is backed up



Backup Storage and Billing

Backups are saved as packages in OBS buckets. Backups occupy backup space in OBS. If the free space RDS provides is used up, the additional space required will be billed. For details, see **How Is RDS Backup Data Billed?**

Deleting Backups

Backups can be deleted in different ways:

- Manual backups can only be manually deleted.
- Automated backups cannot be manually deleted. To delete them, set the retention period specified in your **automated backup policy**. Retained backups will be automatically deleted at the end of the retention period.

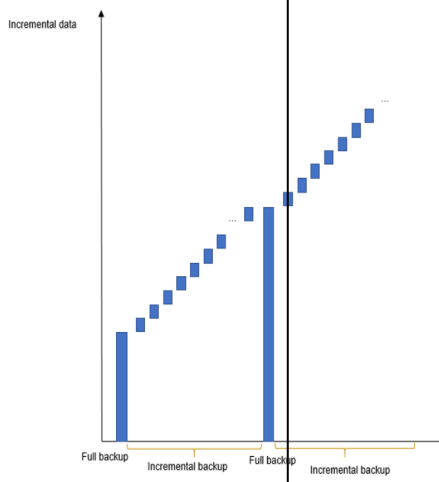
3.9.2 Backup Types

RDS for PostgreSQL supports multiple backup types. Based on different dimensions, there are the following backup types:

Full Backups, Database-Level Backups, and Incremental Backups Based on Data Volume

Table 3-27 Comparison between full backups and incremental backups

Backup Type	Full backups	Database-level backups	Incremental backups
Description	All data in an instance is backed up.	Specific databases in an instance are backed up.	Only data changes within a certain period of time are backed up.
Enabled by Default	Yes	No. To use this function, **submit a service ticket** to request required permissions.	Yes
Retention Period	• You can specify how many days automated backups can be retained for. If you shorten the retention period, the new backup policy takes effect for existing backups. • Manual backups will not be deleted until you delete them manually.	Database-level backups are the backups you create for specific databases. They are retained until you delete them manually.	Incremental backups will be deleted along with automated full backups.

Characteristics	• A full backup is to back up all data of your DB instance in the current point of time. • You can use a full backup to restore the complete data generated when the backup was created. • Full backups can be created automatically or manually.	• Database-level backups are triggered manually. • By default, a maximum of 50 databases can be backed up at a time. • If a database fails or its data is damaged, you can restore it from a database-level backup to ensure data reliability.	• The system automatically backs up data modifications made after the most recent automated or incremental backup every 5 minutes. • Incremental backups can only be created automatically. • Restoring data from incremental backups relies on the most recent full backup, as shown in **Figure 3-78**, so the most recent full backup that exceeds the retention period will not be deleted to ensure successful restoration. **Figure 3-78** Incremental data restoration 
How to Check Backup Size	Click the target instance name. On the **Backups & Restorations** page, click the **Full Backups** tab and check the backup size.	Click the target instance name. On the **Backups & Restorations** page, click the **Database Backups** tab and check the backup size.	Click the target instance name. On the **Backups & Restorations** page, click the **Incremental Backups** tab and check the backup size.

Automated Backups and Manual Backups Based on Backup Methods

Table 3-28 Comparison between automated backups and manual backups

Backup Type	Automated backups	Manual backups
Description	• You can set an automated backup policy on the console, and the system will back up your instance data based on the time window and backup cycle you set in the backup policy and will store the backups for the retention period you specified. • Automated backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your automated backup policy. Retained backups (including full and incremental backups) will be automatically deleted at the end of the retention period. **NOTE** Once a DB instance is created, a backup is restored to a new instance, or a minor version is upgraded, a full backup is automatically triggered by default and this function cannot be disabled. Backups that are being created can be stopped. If such a backup task is stopped and the first full backup of your instance is not complete, no backup is available for restoration. To stop a backup, **submit a service ticket** to apply for required permissions.	• Manual backups are user-initiated full backups of your DB instance. They are retained until you delete them manually. • Regularly backing up your DB instance is recommended, so if your DB instance fails or data is corrupted, you can restore it using backups.
Enabled by Default	Yes	Yes

Retention Period	Automated backups are retained for the number of days you specified. The retention period ranges from 1 to 732 days.	Manual backups are always retained until you delete them manually.
How to Configure	See **Configuring an Intra-Region Backup Policy**.	See **Creating a Manual Backup**.

3.9.3 Instance-Level Backups

3.9.3.1 Configuring an Intra-Region 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.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

You do not need to set an interval for incremental backup because RDS automatically backs up incremental data every 5 minutes. Incremental backups can be used to restore data to a specific point in time.

Constraints

You can only configure an automated backup policy for your DB instance, but not for read replicas.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

Modifying an Automated Backup Policy

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

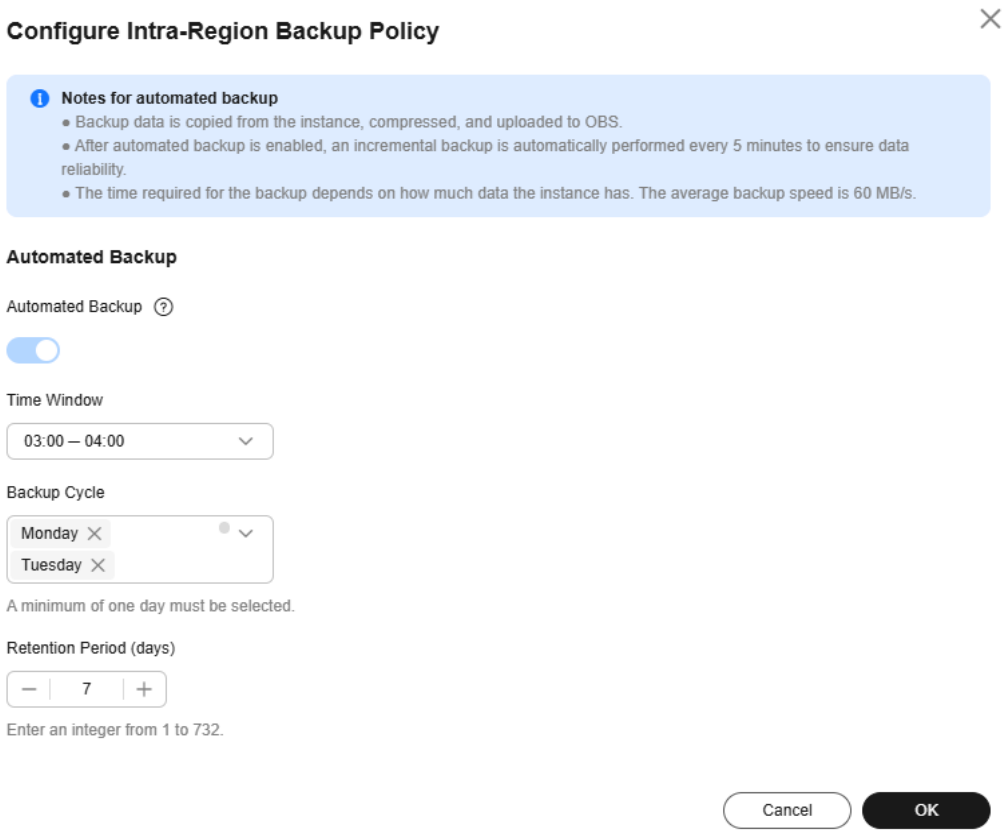
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Backups & Restorations** page, click **Intra-Region Backup Policies**. On the displayed page, you can view the existing backup policy. If you want to modify the policy, adjust the values of the following parameters:

Figure 3-79 Modifying an automated backup policy



- **Retention Period**: How many days your automated full backups and incremental backups can be retained. The retention period is from 1 to 732 days and the default value is **7**. To extend the retention period, **submit a service ticket**.
  - Extending the retention period improves data reliability.
  - Reducing the retention period takes effect for existing backups. Any backups (except manual backups) that have expired will be automatically deleted. Exercise caution when performing this operation.

  Policy for automatically deleting automated full backups:

  To ensure data integrity, even after the retention period expires, the most recent backup will be retained, for example, if **Backup Cycle** was set to **Monday** and **Tuesday** and **Retention Period** was set to **2**:

  - The full backup generated on Monday will be automatically deleted on Thursday because:

    The backup generated on Monday expires on Wednesday, but it is the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday.
  - The full backup generated on Tuesday will be automatically deleted on the following Wednesday because:

    The backup generated on Tuesday will expire on Thursday, but as it is the last backup, it will be retained until a new backup expires. The next backup will be generated on the following Monday and will expire on the following Wednesday, so the full backup generated on Tuesday will not be automatically deleted until the following Wednesday.

- **Time Window**: A one-hour period the backup will be scheduled for each day, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.

  NOTE

  To minimize the potential impact on services, set the time window to off-peak hours. The backup time is in UTC format. The backup time segment changes with the time zone during the switch between the DST and standard time.
- **Backup Cycle**: Daily backups are selected by default, but you can change it. At least one day must be selected.

Step 6 Click **OK**.

----End

3.9.3.2 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups for an available DB instance. You can use these backups to restore data.

Constraints

- When you delete a DB instance, its automated backups are also deleted but its manual backups are retained.
- You can create manual backups only when your account balance is no less than $0 USD.
- The backup name must be unique.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

Method 1

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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**.

- 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.

To check whether the backup has been created, you can click in the upper right corner of the page to check the DB instance status. If the DB instance status becomes **Available** from **Backing up**, the backup has been created.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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**.

- 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 **Backups** page.

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

----End

3.9.3.3 Replicating a Backup

Scenarios

RDS supports replication of automated and manual backups.

Constraints

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

Snapshot-based backups, including CBR snapshot-based backups, cannot be replicated.

Backup Retention Policy

- If a DB instance is deleted, the automated backups created for it are also deleted.
- If an **automated backup policy** is enabled, the automated backups will be deleted after the backup retention period expires.
- 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.
- Replicating a backup does not interrupt your services.

Billing

Backups are saved as packages in OBS buckets. For details, see **How Is RDS Backup Data Billed?**

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name. On the **Backups & Restorations** page, locate the backup to be replicated and click **Replicate** in the **Operation** column.

Alternatively, choose **Backups** in the navigation pane on the left. On the displayed page, locate the 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 carriage return characters or the following special characters: >!<"&'=

Step 6 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

3.9.4 Creating a Database-Level Backup

Scenarios

You can create database-level backups for DB instances running properly. The database-level backups can be used to restore data, ensuring data reliability.

Constraints

- To use this function, **submit a service ticket** to request required permissions.
- When you delete a DB instance, its automated backups will also be deleted but its database-level backups will not.
- You can create database-level backups only when your account balance is no less than $0 USD.
- The backup name must be unique.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

Procedure

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**.

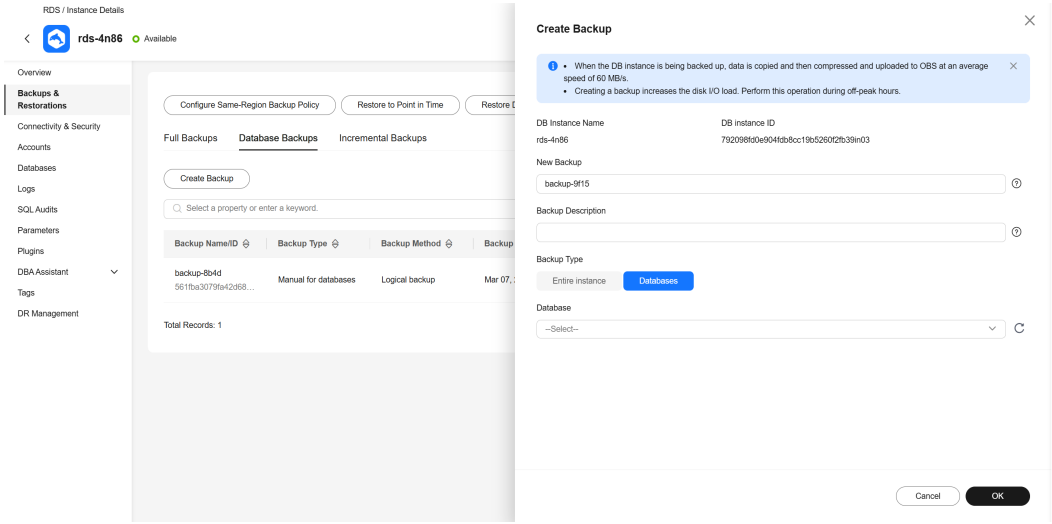
Step 6 Click the **Database Backups** tab and click **Create Backup**.

Alternatively, click **Create Backup** on the **Full Backups** tab page. In the dialog box displayed on the right, select **Databases** for **Backup Type**.

Step 7 In the displayed dialog box, enter a backup name and description, set **Backup Type** to **Databases** (default), select the database to be backed up, and click **OK**.

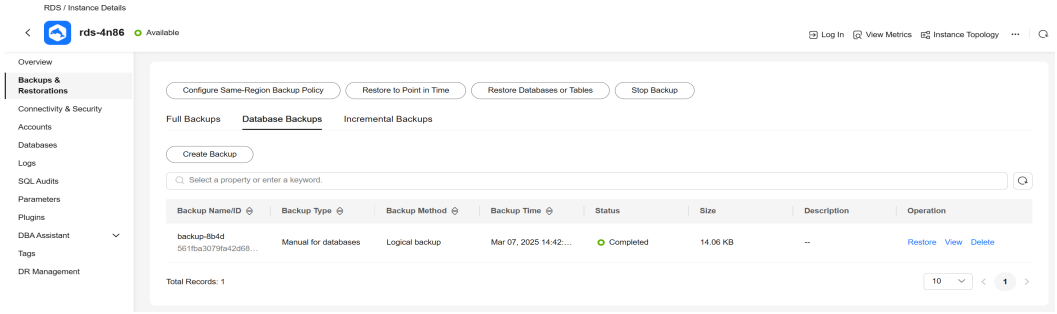
- The backup name can contain 4 to 64 characters and must start with a letter. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
- The description consists of a maximum of 256 characters and cannot contain carriage return characters or the following special characters: >!<"&'=
- The time required for creating a database-level backup depends on how much data there is in the database.

Figure 3-80 Creating a database-level backup



Step 8 After the creation is complete, click **View**. In the dialog box displayed on the right, you can download the backup of the specified database and perform a database-level restoration.

Figure 3-81 Backup created



----End

3.9.5 Managing Backups

3.9.5.1 Downloading an Instance-Level Backup

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.

Constraints

- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.
- When you use OBS Browser+ to download backup data, there is no charge for the generated outbound traffic.

Method 1: Using OBS Browser+

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

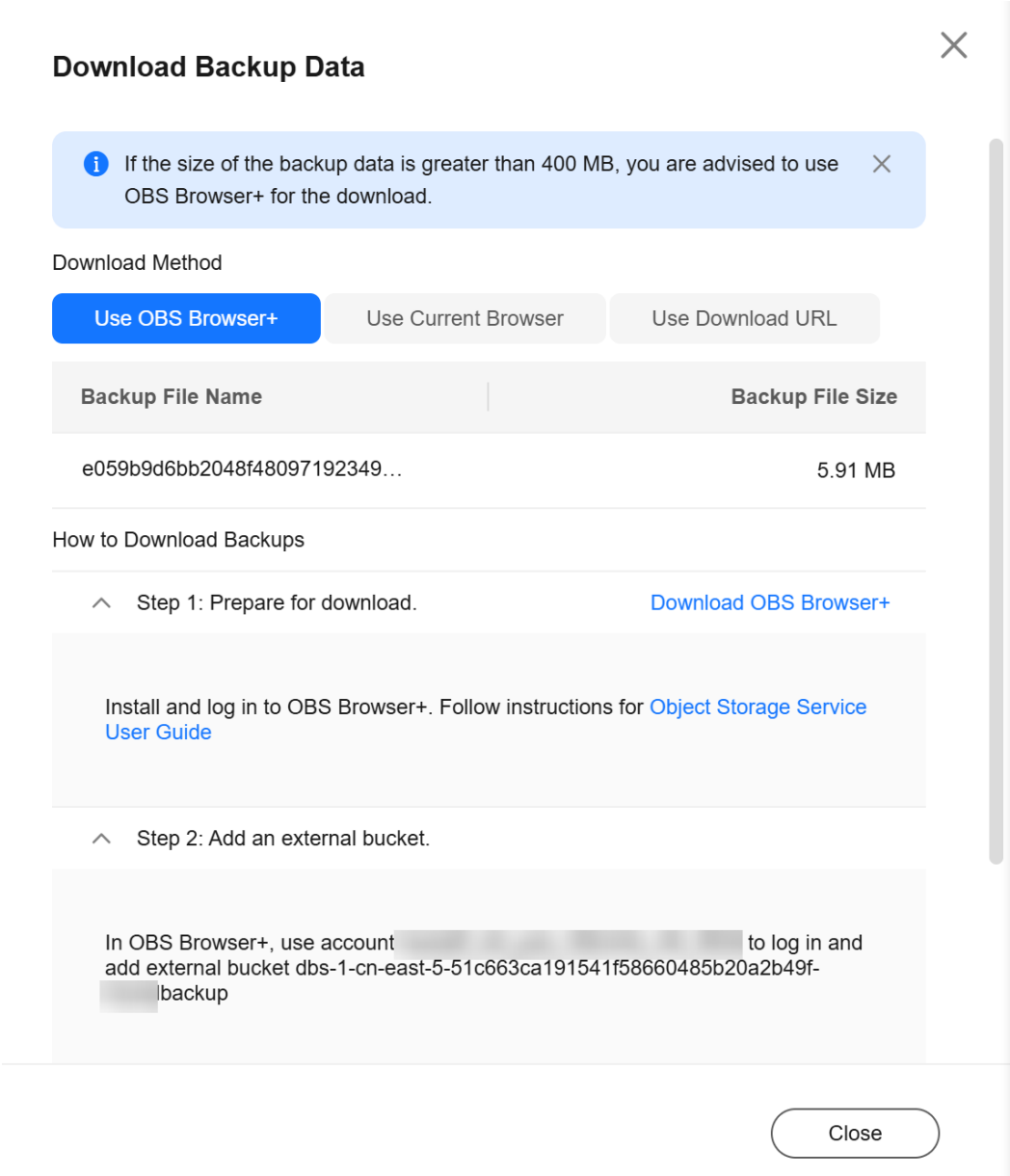
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

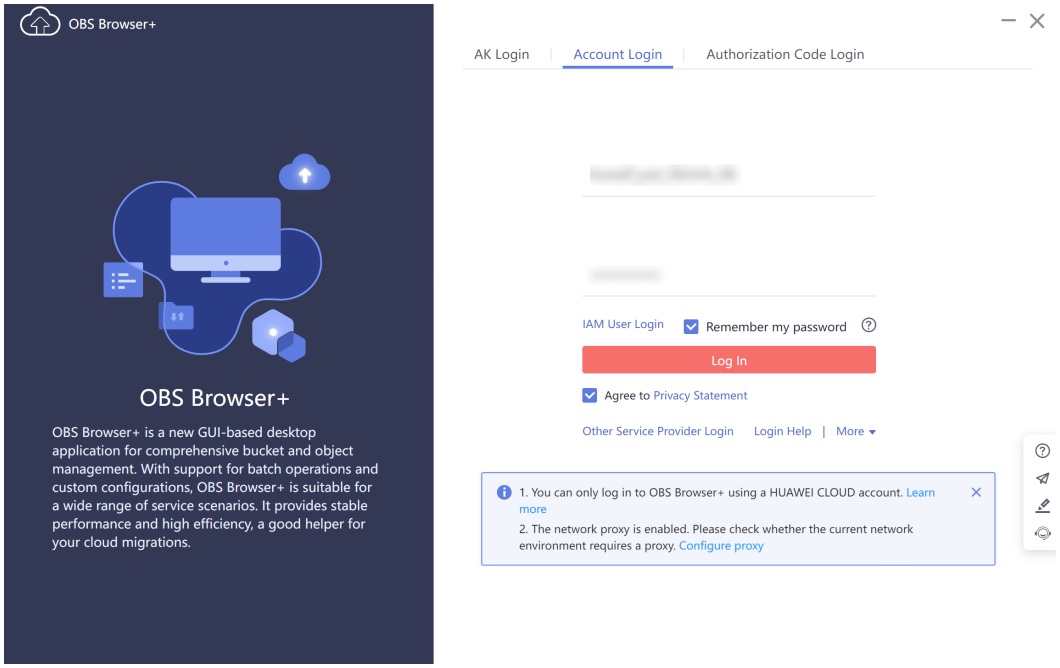
Step 6 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and download the backup as prompted.

Figure 3-82 Using OBS Browser+



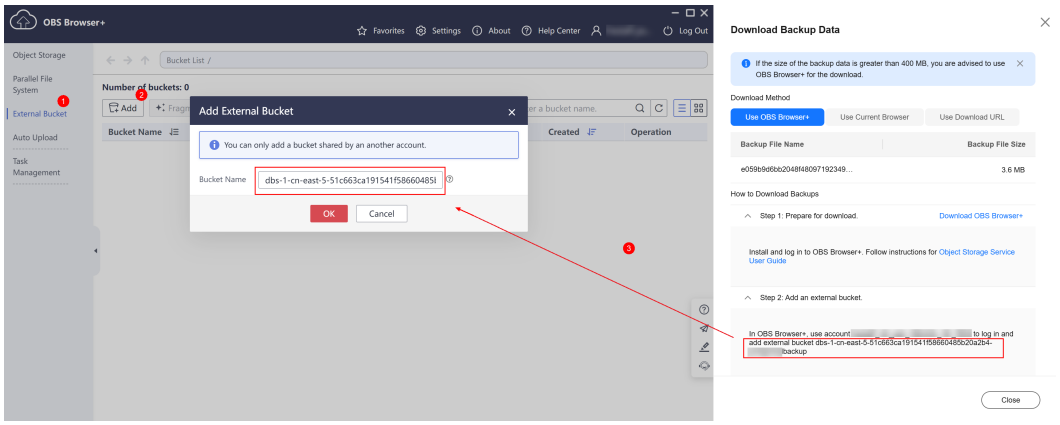
1. Download OBS Browser+.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

Figure 3-83 Logging in to OBS Browser+



4. Add an external bucket using the bucket name provided in step 2 on the download guide page.

Figure 3-84 Adding an external bucket



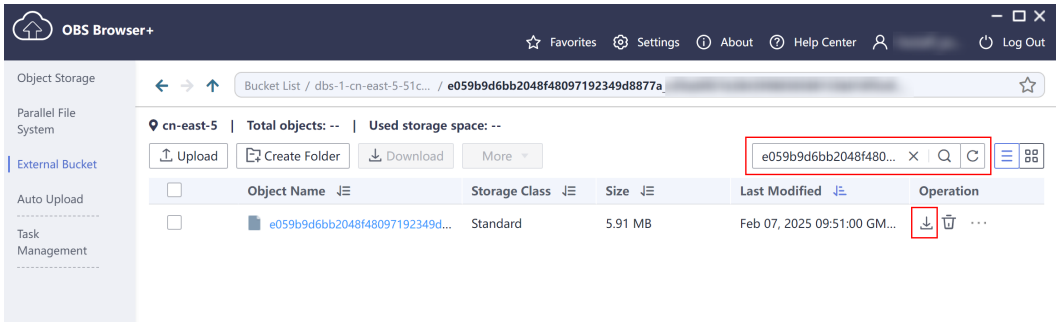
NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name and start a search. In the search result, locate the target backup and click in the **Operation** column.

Figure 3-85 Downloading a backup



----End

Method 2: Using Current Browser

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

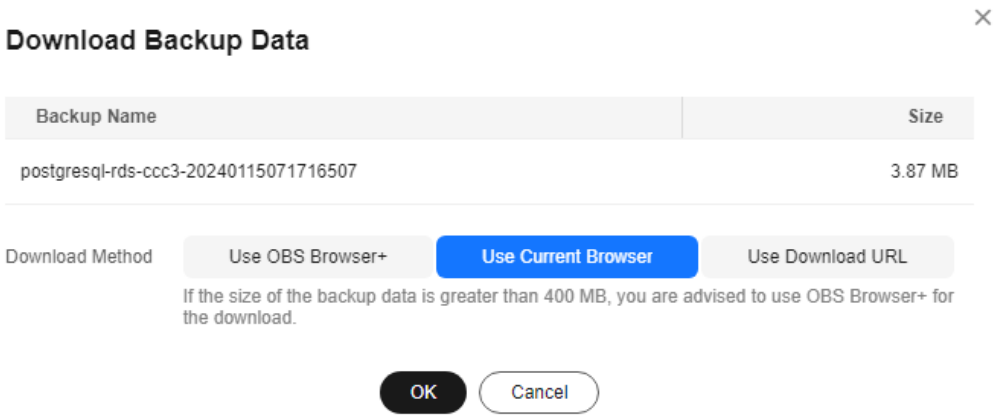
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 In the displayed dialog box, select **Use Current Browser** for **Download Method** and click **OK**.

Figure 3-86 Using the current browser



----End

Method 3: Using Download URL

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

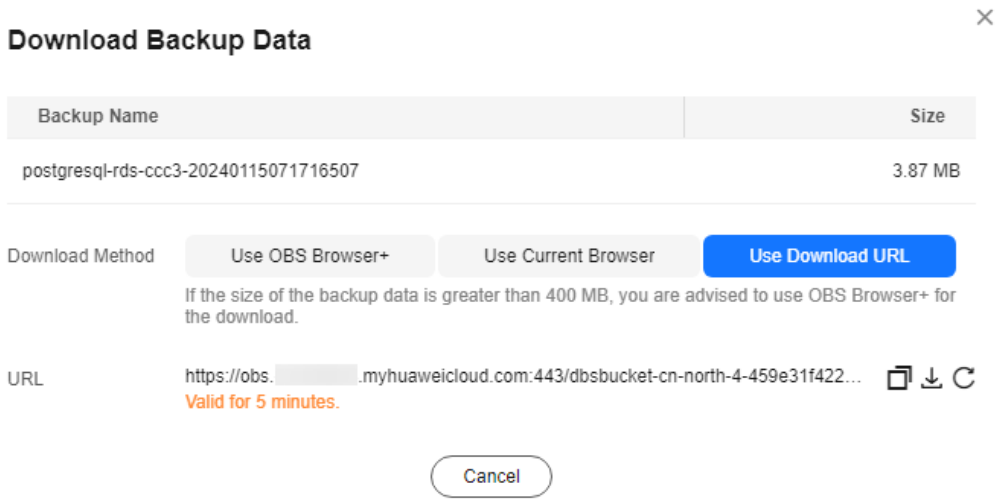
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the **Full Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 In the displayed dialog box, select **Use Download URL** for **Download Method**, click to copy the URL, and enter the URL in your browser.

Figure 3-87 Using the download URL



A valid URL for downloading the backup data is displayed.

- You can use various download tools, such as your browser 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

3.9.5.2 Downloading a Database-Level Backup

Scenarios

You can download a database-level backup for local storage or restoration.

Constraints

- If the size of a backup is greater than 400 MB, you are advised to use OBS Browser+ to download the backup.
- When you use OBS Browser+ to download a backup, there is no charge for the generated outbound traffic.

Method 1: Using OBS Browser+

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

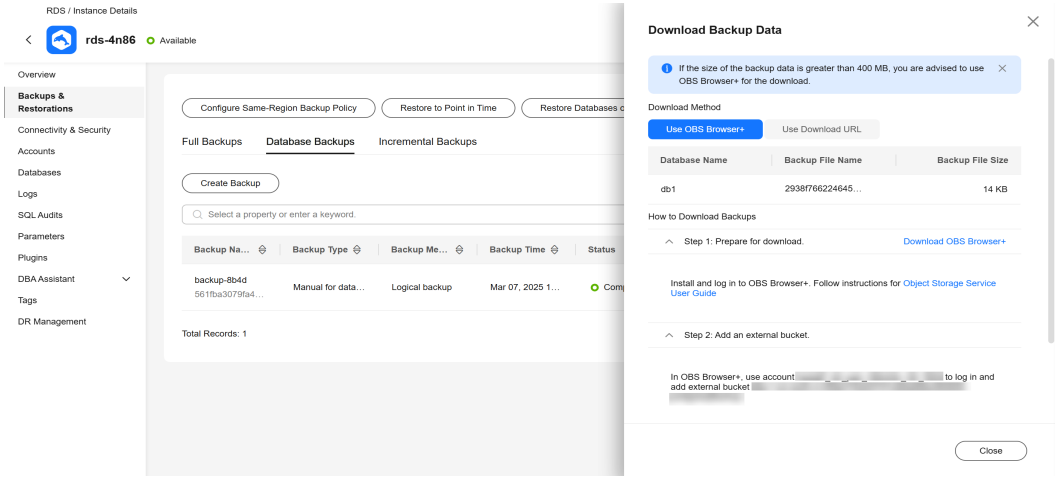
Step 5 In the navigation pane on the left, choose **Backups & Restorations**.

Step 6 Click the **Database Backups** tab. Locate a backup and click **View** in the **Operation** column.

Step 7 In the displayed dialog box, locate the backup to be downloaded and click **Download** in the **Operation** column.

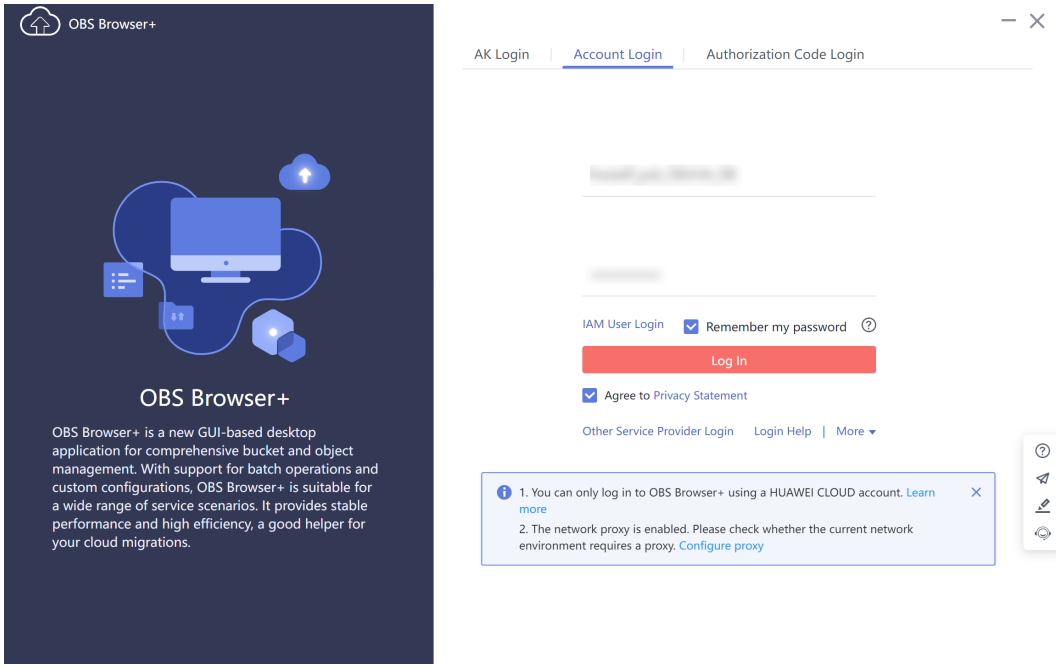
Step 8 In the displayed dialog box, select **Use OBS Browser+** for **Download Method**.

Figure 3-88 Using OBS Browser+



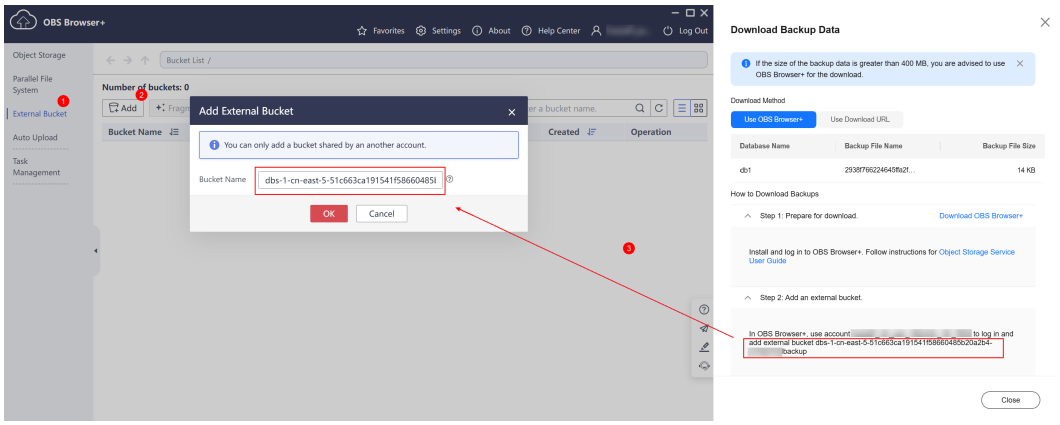
1. Download OBS Browser+.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+ using the username provided in Step 2 on the download guide page.

Figure 3-89 Logging in to OBS Browser+



4. Add an external bucket using the bucket name provided in Step 2 on the download guide page.

Figure 3-90 Adding an external bucket



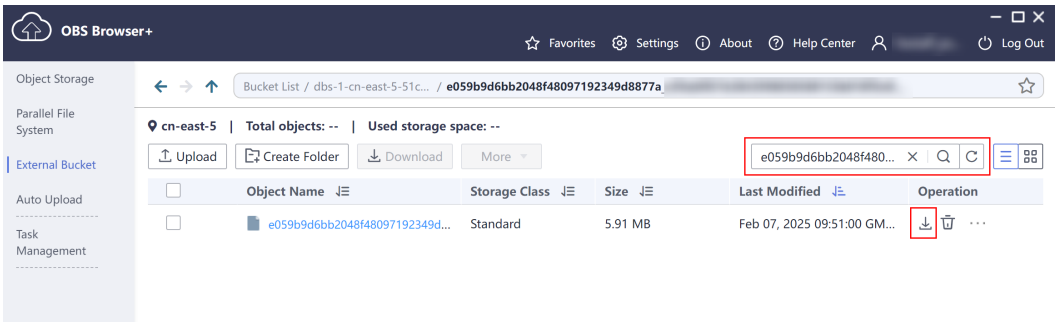
NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

5. Download the backup.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name and start a search. In the search result, locate the target backup and click in the **Operation** column.

Figure 3-91 Downloading a backup



----End

Method 2: Using Download URL

Step 1 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 2 In the navigation pane on the left, choose **Backups & Restorations**.

Step 3 Click the **Database Backups** tab. Locate a backup and click **View** in the **Operation** column.

Step 4 In the displayed dialog box, locate the 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**.

Step 6 Click **Download** to download the backup using your browser.

The URL is valid for 5 minutes.

----End

3.9.5.3 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name. Choose **Backups & Restorations** in the navigation pane on the left. On the **Incremental Backups**

page, locate the 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 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 After the download is complete, you can view the incremental backups on your computer.

----End

3.9.5.4 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

For details about how to export backup data, see **Downloading an Instance-Level Backup** and **Downloading an Incremental Backup File**.

Procedure

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

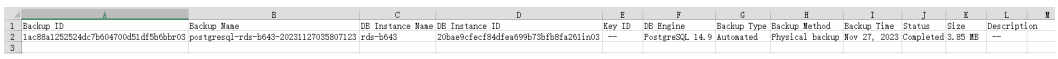
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, select the backups you want to export and click  to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.
- The backup information is exported to an Excel file for your further analysis.

Figure 3-92 RDS for PostgreSQL backup information



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Backup ID	Backup Name	DB Instance Name	DB Instance ID	Key ID	DB Engine	Backup Type	Backup Method	Backup Time	Status	Size	Description	
2	1ac88a1252524dc7b604700d51df5b6bbr03	postgresql-rds-b643-20231127035807123	rds-b643	20bae9cfecf84dfea699b75bfb8fa261in03	--	PostgreSQL 14.9	Automated	Physical backup	Nov 27, 2023	Completed	3.85 MB	--	
3													

Step 5 View the exported backup information.

----End

3.9.5.5 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

To delete a database-level backup, click the **Database Backups** tab, locate the backup, and click **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

3.9.5.6 Stopping a Backup

Scenarios

You can stop a backup that is being created and then delete it.

Constraints

- Only automated and manual backups that are being created can be stopped.

- After a backup is stopped, it will not be billed and can be deleted.
- A backup that is being replicated or a CBR snapshot-based backup that is being created cannot be stopped.
- This function can only be used when there is much data to be backed up and creating the backup takes too long.
- You are advised not to stop the first automated backup after an instance is changed or restored. Forcibly stopping that backup may cause incremental and differential backups between the current time and the next automated full backup to fail, and point-in-time recovery (PITR) may be unavailable. Do not stop that backup unless absolutely necessary.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Backups & Restorations** and click **Stop Backup**.

Step 6 In the displayed dialog box, click **OK** to stop the backup of the current instance.

After the backup is stopped, its status changes to **Stopped** and you can delete it.

----End

3.10 Data Restorations

3.10.1 Restoration Solutions

If your database is damaged or mistakenly deleted, you can restore it from backups.

Restoring a Mistakenly Deleted Instance

- RDS moves unsubscribed yearly/monthly instances and deleted pay-per-use instances to the recycle bin. You can **rebuild an instance** that was deleted up to 7 days ago from the recycle bin.
- You can restore a deleted instance from its retained manual backups. For details, see **Restoring a DB Instance from Backups**.

Restoring Mistakenly Deleted or Modified Data of a DB Instance

Table 3-29 Restoration solutions

Solution	Category	Storage Type		Restoration Time Point	Scope		Restore To			Time Required
		Cloud SSD	Extreme SSD	Point in Time When a Backup Was Generated	All Databases and Tables	Certain Databases and Tables	New Instance	Original Instance	Existing Instance Other than the Original Instance	
Restoring an entire instance	Restoration from backups	√	√	x	√	x	√	x	√	Depending on how much data there is in the instance
	Point-in-time recovery (PITR)	√	√	√	√	x	√	x	√	Depending on how much data there is in the instance

Restoring databases and tables	Restoration of databases and tables	√	√	√	x	√	x	√	x	Depending on how much data there is in the instance and in the databases and tables

Restoring or Migrating Data to an RDS for PostgreSQL Instance

- You can restore data to an RDS for PostgreSQL instance from backups. For details, see **Restoring Data to RDS for PostgreSQL**.
- You can migrate data to an RDS for PostgreSQL instance using Data Replication Service (DRS), pg_dump, or Data Admin Service (DAS). For details, see **Migration Solution Overview**.

Restoring or Migrating Data to a Self-Managed PostgreSQL Database

- You can restore data to a self-managed PostgreSQL database from backups. For details, see **Restoring Data to an On-Premises PostgreSQL Database from a Full Backup**.
- You can migrate data to a self-managed PostgreSQL database using DRS. For details, see **From PostgreSQL to PostgreSQL**.

3.10.2 Restoring Data to RDS for PostgreSQL

3.10.2.1 Restoring a DB Instance from Backups

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.

Function Description

Table 3-30 Function description

Item	Description
Restoration scope	The entire instance
Instance data after restoration	The instance data after restoration is consistent with that in the full backup used for the restoration. • Restoring data to a new instance creates an instance with the same data as that in the backup. • Restoring data to an existing instance will overwrite the instance data.
Restoration type	• Restoration to a new instance • Restoration to an existing instance other than the original one
Configurations for restoring to a new instance	• The DB engine and engine version of the new instance are the same as those of the original instance. • The storage space of the new instance is the same as that of the original instance by default and the new instance must be at least as large as the original instance. • Other parameters need to be reconfigured.
Time required	The time required depends on how much data there is in the instance. The average restoration speed is 40 MB/s.

Constraints

- You can restore to new DB instances from backups only when your account balance is greater than or equal to $0 USD. You will pay for the new instance specifications.
- RDS for PostgreSQL supports restoration to the original DB instance from backups. To use this function, **submit a service ticket**.
- Constraints on restoring data to an existing DB instance (other than the original instance):
  - 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 DB engine version as the original DB

instance. For example, a backup of PostgreSQL 16.5 can be restored only to an existing instance running PostgreSQL 16.5 instead of 16.6.

- 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.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, select the backup to be restored and click **Restore** in the **Operation** column.

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

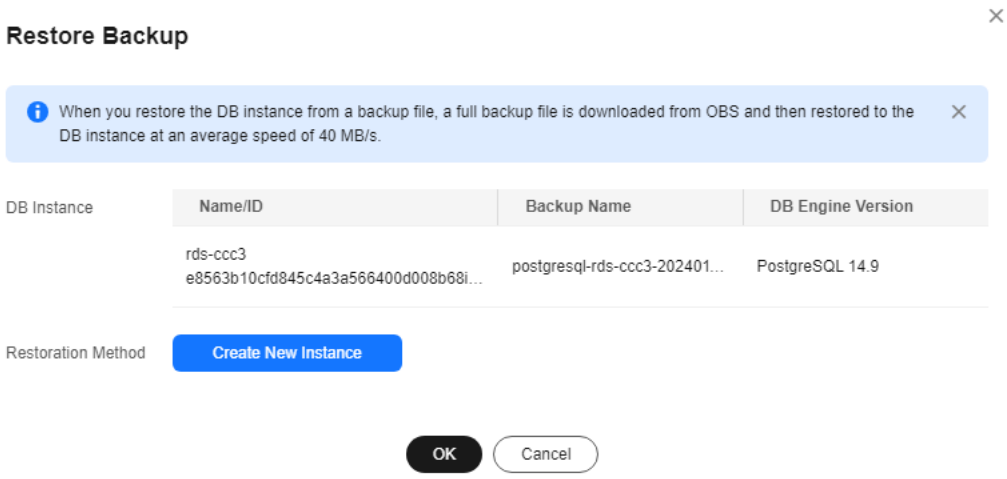
Step 5 Select a restoration method and click **OK**.

NOTICE

RDS for PostgreSQL does not support restoration to the original DB instance. If you intend to restore to the original DB instance, restore to a new DB instance first and then change the IP address to that of the original DB instance.

- Create New Instance

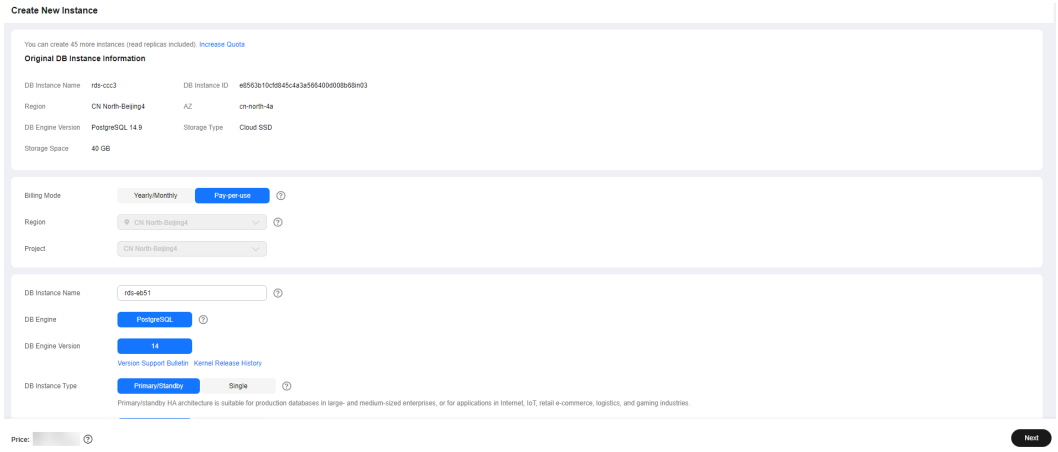
Figure 3-93 Restoring to a new DB instance



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

- The DB engine and engine version of the new instance are the same as those of the original instance.
- 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 **Buy a DB Instance**.

Figure 3-94 Creating a new instance



- Restore to Existing
  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored." and click **Next**.
  b. Confirm the information and click **OK**.
  c. If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.
  d. Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

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 **Instances** 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

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

Backup data cannot be restored to original RDS for PostgreSQL DB instances. If you need to restore data to your original DB instance, restore backup data to a new DB instance and then **migrate the data to the original instance using DRS** or **change the floating IP address of the new DB instance to that of the original instance**.

FAQs

How Can I Restore Data If No Backup Is Available?

3.10.2.2 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.

Function Description

Table 3-31 Function description

Item	Description
Restoration scope	The entire instance
Instance data after restoration	The instance data after restoration is consistent with that in the full backup plus the incremental backup used for the restoration. • Restoring data to a new instance creates an instance with the same data as that generated by that time point. • Restoring data to an existing instance will overwrite the instance data.

Item	Description
Restorable time point	Any time point within the retention period after the earliest full backup is generated
Scenario	• Restoration to a new instance • Restoration to an existing instance other than the original one
Configurations for restoring to a new instance	• The DB engine and engine version of the new instance are the same as those of the original instance. • Other parameters need to be reconfigured.
Time required	The time required depends on how much data there is in the instance. The average restoration speed is 30 MB/s.

Constraints

- You can restore to new DB instances from backups only when your account balance is greater than or equal to $0 USD. A new DB instance is billed based on the instance specifications.
- RDS for PostgreSQL does not support restoration to the original DB instance. If you intend to restore to the original DB instance, restore to a new DB instance first and then change the IP address to that of the original DB instance.
- Constraints on restoring data to an existing DB instance (other than the original instance):
  - 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 DB engine version as the original DB instance. For example, a backup of PostgreSQL 16.5 can be restored only to an existing instance running PostgreSQL 16.5 instead of 16.6.
  - Ensure that the storage space of the selected DB instance is no less than that of the original DB instance. Otherwise, data will not be restored.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **Buying a DB Instance**.

- Restore to Existing

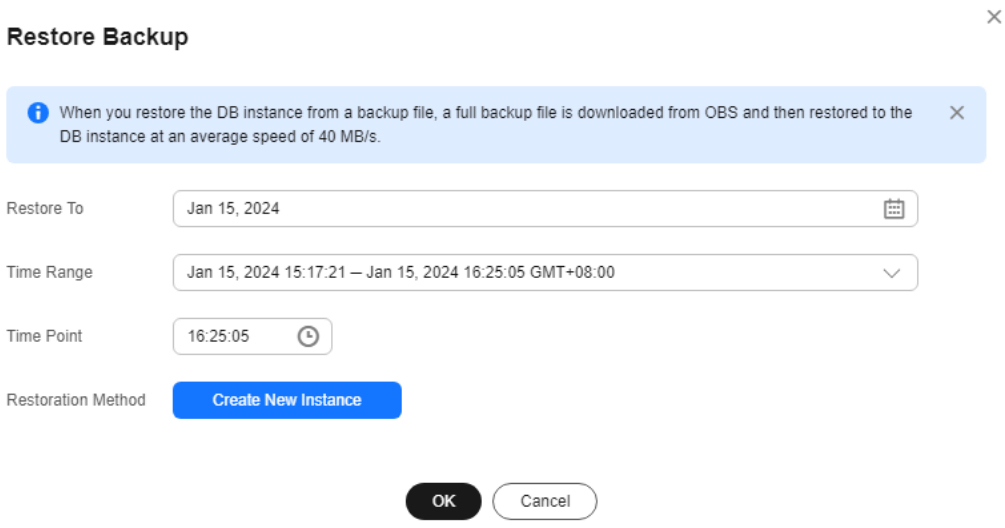
  a. Select "I acknowledge that restoring to an existing DB instance will overwrite data on the instance and will cause the existing DB instance to be unavailable during the restoration. Only DB instances that can be used as target instances for the restoration are displayed here. Eligible instances must have the same DB engine type, version, and at least as much storage as the instance being restored." and click **Next**.

  b. Confirm the information and click **OK**.

  c. If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

  d. Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Figure 3-95 Restoring data to a point in time



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.

  A full backup is triggered after the new DB instance is created.

- Restore to Existing

  On the **Instances** 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

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

FAQs

How Can I Restore Data If No Backup Is Available?

3.10.2.3 Restoring Databases or Tables to a Point in Time

Scenarios

RDS allows you to restore databases or tables using point-in-time recovery (PITR). This ensures your data integrity and minimizes impact on the original instance performance. You can select databases or tables and restore them to a specified point in time. During database or table PITR, RDS downloads the most recent full backup from OBS and restores it to a temporary DB instance, and then replays WAL to the specified point in time on the temporary instance. After that, data on the temporary instance is written to the target databases or tables of the original instance. The time required depends on how much data needs to be restored.

The time required depends on the amount of data to be restored on the DB instance. Restoring databases or tables will not overwrite data in the DB instance. You can select the databases or tables to be restored.

RDS for PostgreSQL supports restoration of databases or tables of only one DB instance at a time.

Constraints

- Take care when restoring tables. Improper operations can cause instance or service exceptions.
- To prevent restoration failures and impact on original data, table-level restoration removes foreign key constraints, inheritance relationships,

partition relationships and triggers, and renames indexes and associated sequences. Database-level restoration does not restore subscriptions.

- During table restoration, a maximum of 20,000 tables can be restored for one instance at a time. If the number of tables to be restored exceeds 20,000, you can restore the entire instance using PITR. For details, see **Restoring a DB Instance to a Point in Time**.
- During database restoration, a maximum of 2,000 databases and 20,000 tables can be restored for one instance at a time. If you want to restore more databases or tables at a time, you can restore the entire instance using PITR. For details, see **Restoring a DB Instance to a Point in Time**.
- During a database or table PITR, DB instances and read replicas cannot be rebooted or deleted, and their instance specifications cannot be modified.
- In a database or table PITR, the database or table information to be restored is read from the latest full backup before the selected time point. You can select any time point within the restorable time period. Therefore, a database or table can be restored to the earliest full backup time point when its information exists.
- If there is no backup data about the specified tables at the point in time, the restoration will still be completed, but no data of the tables is restored.
- RDS for PostgreSQL Enhanced Edition does not support database or table PITR.

Prerequisites

After the restoration, a new database or table will be generated in the DB instance. Ensure that the DB instance has sufficient storage space for the generated database or table.

Restoring Databases or Tables of a DB Instance

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

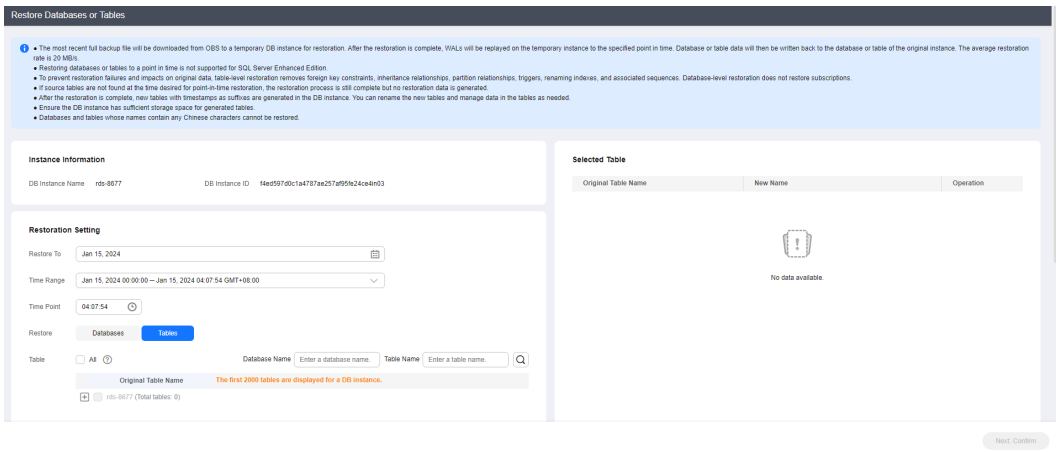
Step 5 In the navigation pane, choose **Backups & Restorations**. On the displayed page, click **Restore Databases or Tables**.

Step 6 Specify restoration information and click **Next: Confirm**.

- To facilitate your operations, you can search for the databases or tables to be restored.
- After the restoration is complete, new databases or tables with timestamps appended as suffixes to original database or table names are generated in the DB instance. You can rename the new databases or tables.
- The new table name must be unique and consist of 1 to 64 characters. Only letters, digits, underscores (_), hyphens (-), and dollar signs ($) are allowed.

- Databases whose names contain periods (.) cannot be restored.
- To prevent data loss, new databases with unique names must be specified for database PITR.
- During database PITR, a maximum of 2,000 databases and 20,000 tables can be restored for a single instance at a time.

Figure 3-96 Restoring databases or tables to a point in time



Step 7 On the displayed page, confirm the information and click **Submit**.

Step 8 On the **Instances** page, check that the DB instance status is **Restoring**. During the restoration, services are not interrupted.

You can also view the progress and result of restoring databases or tables to a specified point in time on the **Task Center** page.

After the restoration is successful, you can manage data in the databases or tables as required.

----End

NOTE

- Data is restored at an average speed of 20 MB/s.
- Restoring databases or tables to a specified point in time does not affect new data. The restored database or table is a temporary database or table with a timestamp suffix. You can manage the data in the temporary database or table as required.

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

FAQs

How Can I Restore Data If No Backup Is Available?

3.10.2.4 Restoring Databases from Database-Level Backups

Scenarios

You can restore a database to its original instance from a database-level backup. After the restoration is successful, a new database and tables are generated on the instance and they have timestamps appended to their names. You can rename the tables or manage the table data as required.

Constraints

- Any database or table whose name contains any Chinese characters cannot be restored.
- Database- or table-level restoration is not supported for databases or tables that contain JSON virtual columns.
- A maximum of 50 databases can be restored for an instance at a time.

Prerequisites

After the restoration, new databases and tables will be generated in the original DB instance. Ensure that the DB instance has sufficient storage space for them.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

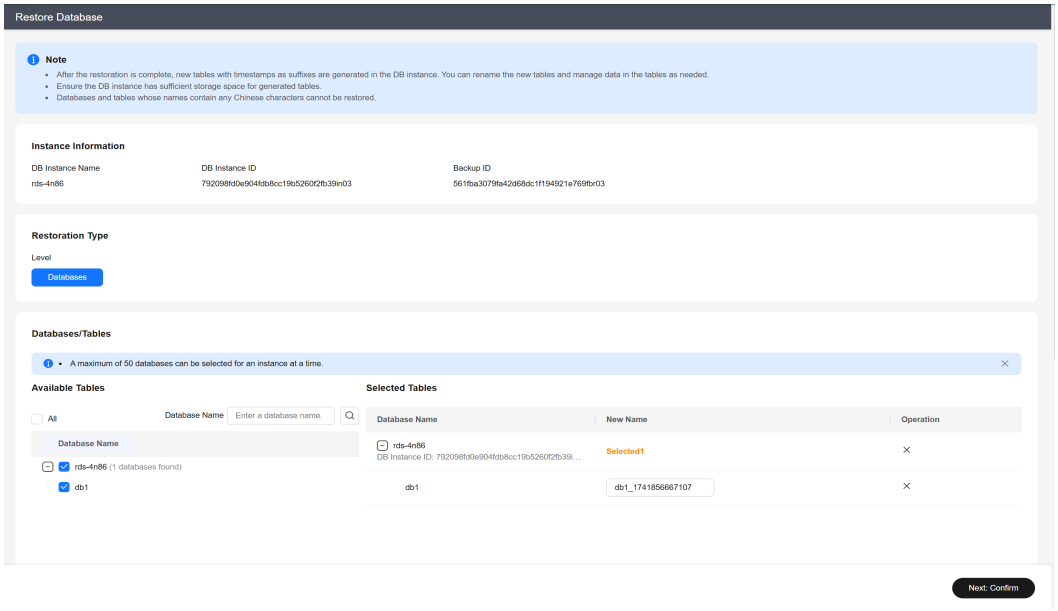
Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 In the navigation pane on the left, choose **Backups & Restorations**.

Step 6 Click the **Database Backups** tab. Locate a backup and click **Restore** in the **Operation** column.

Step 7 Select the databases to be restored and click **Next: Confirm**.

Figure 3-97 Database-level restoration



Step 8 On the displayed page, confirm the information and click **Submit**.

- On the **Instances** page, check that the DB instance status is **Restoring**. During the restoration, workloads are not interrupted.
- After the restoration is successful, new databases and tables are generated on the instance and they have timestamps appended to their names. You can rename the tables or manage the table data as required.

----End

3.10.3 Restoring Data to an On-Premises PostgreSQL Database from a Full Backup

This section describes how to restore an RDS for PostgreSQL instance to an on-premises PostgreSQL database from a full backup.

If you want to migrate all data from your RDS for PostgreSQL DB instance to an on-premises PostgreSQL database, you can download a full backup in **.tar.gz** format and restore it to the on-premises database following the steps described in this section.

- **Step 1: Download a Full Backup of Your RDS for PostgreSQL DB Instance**
- **Step 2: Restore the Backup to the On-Premises PostgreSQL Database**
- **FAQ**

Operation Process

1. Download the target full backup of your RDS for PostgreSQL DB instance.
2. Upload the full backup to the on-premises PostgreSQL database.
3. Use the tar utility to decompress the full backup.
4. Store the configuration files of the on-premises database in a temporary directory and run the OS commands to copy the extracted full backup files to the **data** directory of the on-premises database.

5. Restart the database and wait until the database restoration is complete.

Constraints

- This section does not apply to restoration using incremental backups.
- **The minor version of the on-premises PostgreSQL database must be the same as that of your RDS for PostgreSQL DB instance.**

  To view the PostgreSQL kernel version, run **psql -V** or **psql --version**.
- The backup can be used to restore only to an on-premises database running Linux. The tar utility must be installed on the OS.

  To install this tool, run **sudo yum install tar**.
- During the restoration, do not run any other services on or store service data to the on-premises database.
- RDS for PostgreSQL has certain enhanced features, such as failover slots, which may cause SQL errors on the restored on-premises database. You need to delete such enhanced features from the on-premises database. For details, see **FAQ**.
- The OS of the on-premises database may be different from that of RDS. The sorting rules of some PostgreSQL indexes are subject to the OS. After data is restored to the on-premises database, you need to rebuild the indexes. To find out which indexes need to be rebuilt, see **Locale data changes**.

Step 1: Download a Full Backup of Your RDS for PostgreSQL DB Instance

RDS for PostgreSQL DB instances automatically perform a full backup at a scheduled time you specified. You can also create manual backups. When a full backup is created, download the generated **.tar.gz** file.

1. On the **Instances** page, click the instance name. On the displayed page, choose **Backups & Restorations** > **Full Backups** and click **Download**. For details, see **Downloading an Instance-Level Backup**.
2. Use a file transfer tool (such as WinSCP) to upload the full backup file to the Linux device where the on-premises PostgreSQL database is running.

Step 2: Restore to the On-Premises PostgreSQL Database from the Backup

Notes

Modify certain information based on your site requirements:

1. Store the RDS for PostgreSQL backup file in different directories before and after decompression.
   - Before: **/home/postgres/*Full backup*.tar.gz**
   - After: **/home/postgres/backuprds**
2. Store the **postgresql.conf** and **pg_hba.conf** configuration files in the **data** directory of the on-premises PostgreSQL database to the **/home/postgres/backuplocal** directory.
3. Use the **postgres** user to install the on-premises PostgreSQL database.
4. Replace *$PGDATA* in the commands described below with the **data** directory of your on-premises PostgreSQL database. To query the **data** directory, run the following commands:

su - postgres

psql --host=localhost --port=<*DB_PORT*>** --dbname=postgres --username=postgres -c "show data_directory;"**

DB_PORT indicates the port number of the on-premises database. The default value is **5432**.

Procedure

1. Switch to the **postgres** user and create a temporary directory **backuprds**. Perform all the following steps as **postgres**.

   su - postgres

   mkdir /home/postgres/backuprds

2. Stop the on-premises PostgreSQL database.

   pg_ctl stop -D *$PGDATA*

3. Create a temporary directory to store the **postgresql.conf** and **pg_hba.conf** configuration files in the **data** directory of the on-premises PostgreSQL database.

   mkdir /home/postgres/backuplocal

   cp *$PGDATA*/**pg_hba.conf** *$PGDATA*/**postgresql.conf /home/postgres/backuplocal**

4. Clear the **data** directory of the on-premises database.

   NOTICE

   Before performing this operation, ensure that the data in the *$PGDATA/* directory is no longer needed.

   To view files in the *$PGDATA/* directory, run the **ls -l** *$PGDATA* command.

   rm -rf *$PGDATA/**

5. Run the following command to decompress the backup to the directory prepared in **1**:

   NOTE

   If you upload the backup file to **/home/postgres/***Full backup***.tar.gz** as user **root**, you need to change the owner of the file.

   1. Run the **sudo su** command to switch to user **root**.
   2. Run the **chown -R postgres:postgres /home/postgres/***Full backup***.tar.gz** command to change the file owner to the **postgres** user.
   3. Run the **su - postgres** command to switch back to **postgres**.

   tar -zxf /home/postgres/*Full backup***.tar.gz -C /home/postgres/backuprds**

   After the decompression, the following folders are generated in **/home/postgres/backuprds**:

   - **base**: stores full backup files.
   - **pg_wal**: stores incremental backup files. For PostgreSQL 9.*x*, the generated directory is **pg_xlog**.
   - Tablespace folders named using digits (if the original backup contains tablespace files)

6. Copy the files in **5** and **3** to the specified directories of the on-premises database in sequence.
   a. Copy all files in the **base** directory to the **data** directory, and then replace the two files in the **data** directory with the configuration files in **3**.

      cp -r /home/postgres/backuprds/base/* *$PGDATA*

      cp -r /home/postgres/backuplocal/* *$PGDATA*

   b. Copy the files in the **pg_wal** directory (or the **pg_xlog** directory for PostgreSQL 9.*x*) to **pg_wal** (or **pg_xlog** for PostgreSQL 9.*x*) under the **data** directory of the on-premises database.

      cp -r /home/postgres/backuprds/pg_wal/* *$PGDATA*/**pg_wal**

   c. (Optional) If there are tablespace files in the original backup, modify the tablespace soft link information in the **data/tablespace_map** file.

      - Copy the tablespace files to the **/tmp/tblspc/** directory.

        If the decompressed file contains multiple tablespace directories, run the **cp -r /home/postgres/backuprds/$table_space /tmp/tblspc** command repeatedly to ensure that all tablespaces are copied to the **/tmp/tblspc** directory.

        mkdir /tmp/tblspc

        cp -r /home/postgres/backuprds/*$table_space* **/tmp/tblspc**

        $table_space indicates the name of the tablespace folder obtained in **5**.

      - Delete the **/tablespace_map** file from the **data** directory of the on-premises database.

        rm -rf *$PGDATA*/**tablespace_map**

      - Add a new **/tablespace_map** file to the **data** directory of the on-premises database. If the decompressed file contains multiple tablespace directories, run the following command repeatedly to ensure that the soft link information of the tablespaces is complete:

        echo "*$table_space* **/tmp/tblspc/***$table_space***" >>** *$PGDATA*/**tablespace_map**

7. Restart the database and wait until the database restoration is complete.

   pg_ctl start -D *$PGDATA*

   NOTE

   If the cloud database is processing many write requests during the backup, there will be a large number of WAL logs in the **pg_wal** directory. It may take a long time to replay WAL logs when the database is started. As a result, the startup command may time out and fail.

   To check the restoration progress, run the **ps uxwwf | grep 'startup'** command.

FAQ

Data Restoration Issues

Q: How do I restore data if no backup is available?

A: You can migrate data using Data Replication Service (DRS). For details, see **From PostgreSQL to PostgreSQL**.

Backup and Restoration Issues

- Q1: After I restored data to an on-premises database using an RDS backup, the database failed to start and the error message "replication slot file xxx has corrupted length xxx" was displayed. What should I do?

  A: Delete all files and folders under the **pg_replslot** directory and then restart the database.

- Q2: What are the causes for the error "could not locate a valid checkpoint record"?

  A: This error usually indicates that the checkpoint record in the database is damaged or lost. As a result, the database cannot be restored. Generally, it is because WAL logs are not properly loaded. Handle the problem by referring to **6.b**.

RDS for PostgreSQL 11 Data Restoration Issues

- Q1: What should I do if the error message "ERROR: internal function "int4_text" is not in internal lookup table" is displayed during the conversion from **int4** to **text** when an RDS for PostgreSQL 11 instance is restored to a local PostgreSQL 11 database?

  A: Connect to the on-premises PostgreSQL 11 database as the installation user (such as **postgres**) and run the following command to delete the conversion rule:

  delete from pg_cast where castsource = 'int4'::regtype and casttarget = 'text':: regtype;

- Q2: What can I do if multiple type conversion functions generate errors when an RDS for PostgreSQL 11 instance is restored to a local PostgreSQL 11 database?

  A: Run the following SQL statement on the on-premises PostgreSQL 11 database and RDS for PostgreSQL 11 instance, respectively, and compare the results:

  select oid, * from pg_cast order by 1;

  For the new type conversion rules of RDS for PostgreSQL 11, run the following SQL statement on the local PostgreSQL 11 database to delete the rules:

  delete from pg_cast where castsource = *xxx* **and casttarget =** *xxx***;**

3.11 Read Replicas

3.11.1 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 offload read pressure on the primary DB instance, you can create one or more read replicas in the same region as the primary instance. 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 standby node). RDS automatically synchronizes changes made on the primary DB instance to all associated read replicas using the PostgreSQL replication. If there is a high network latency on the primary instance, data synchronization to read replicas is affected. Read replicas and the primary DB instance must be in the same region but can be in different AZs.

Billing Standards

Read replicas are billed on a yearly/monthly or pay-per-use basis.

Functions

- The specifications of a read replica must be at least equal to those of the primary DB instance to prevent long delay and high load.
- Read replicas are billed on a yearly/monthly or pay-per-use basis. Yearly/monthly billing provides a larger discount than pay-per-use billing and is recommended for long-term users.
- 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.
- You can bind EIPs to read replicas and unbind EIPs from read replicas.
- 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

- Primary DB instances and read replicas billed on the yearly/monthly basis can be unsubscribed only through the RDS console.
- If you want to unsubscribe both primary DB instances and read replicas that are billed on the yearly/monthly basis, unsubscribe the read replicas first.
- Expired primary DB instances are recycled in the recycle bin and deleted after the retention period expires. The associated read replicas must be unsubscribed before the DB instances are deleted. After the DB instances are deleted, its associated read replicas cannot be unsubscribed.

- You can purchase read replicas only for your created primary DB instance.
- A maximum of five read replicas can be created for a DB instance.

Creating and Managing a Read Replica

- **Creating a Read Replica**
- **Managing a Read Replica**

3.11.2 Creating a Read Replica

Scenarios

Read replicas enhance the read capabilities and reduce the load on your DB instances.

You can create read replicas as needed.

NOTE

Up to five read replicas can be created for a DB instance.

The specifications of a read replica must be at least equal to the specifications of the DB instance. If you want the specifications of a read replica to be smaller than those of the DB instance, **submit a service ticket**.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Create Read Replica** in the **Operation** column.

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

Figure 3-98 Basic information

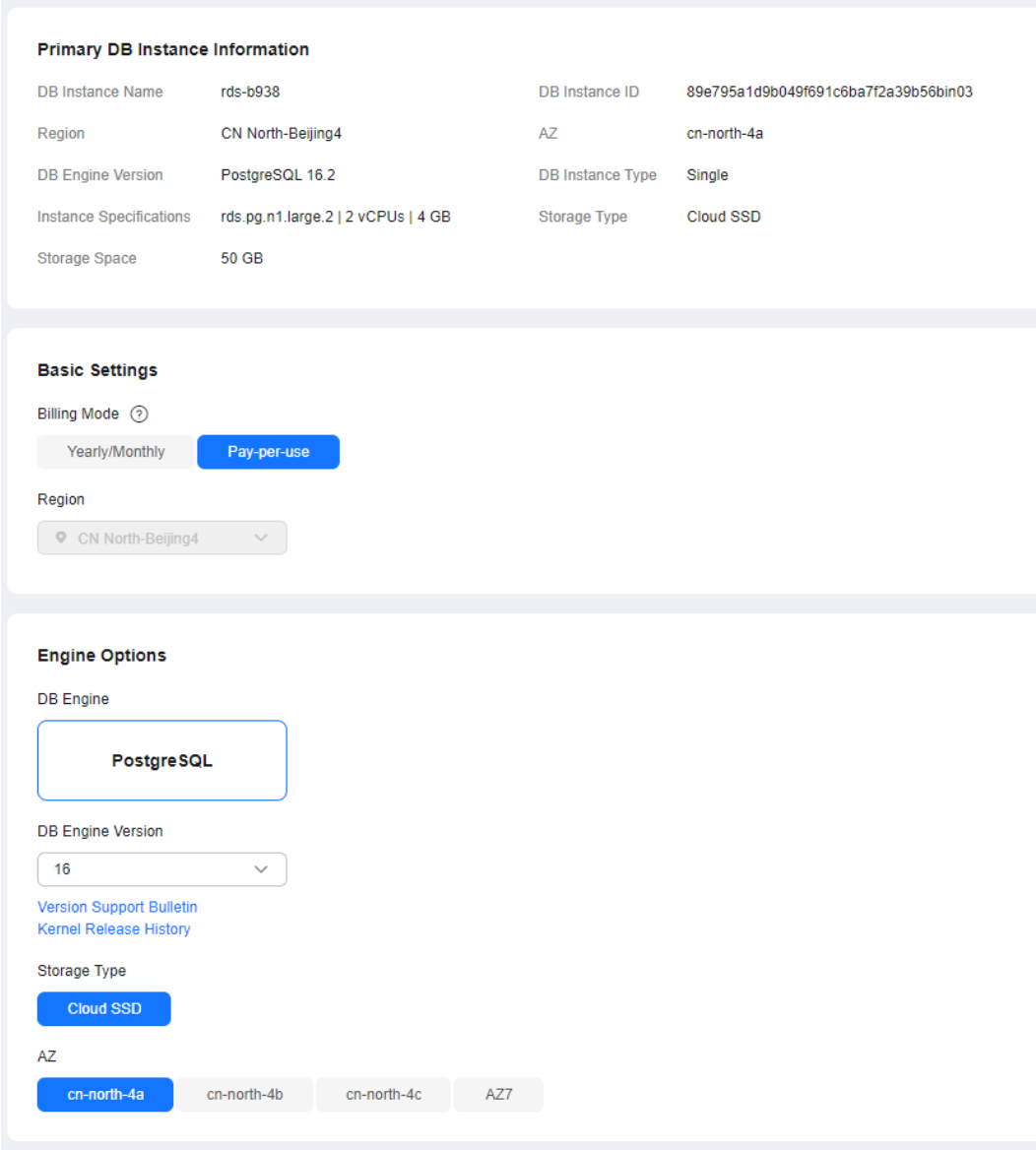


Table 3-32 Basic information

Parameter	Description
Billing Mode	Yearly/monthly billing and pay-per-use billing are supported.
Region	By default, read replicas are in the same region as your DB instance.
DB Engine	Same as the DB engine of your 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
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. • **Cloud SSD**: cloud disks used to decouple storage from compute. • **Extreme SSD**: uses the 25GE network and RDMA technology to provide you with up to 1 million random read/write performance per disk and low latency per channel. **NOTE** If you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default.
AZ	RDS allows you to deploy your DB instance and read replicas in a single AZ or across AZs to improve reliability.

Figure 3-99 Specifications

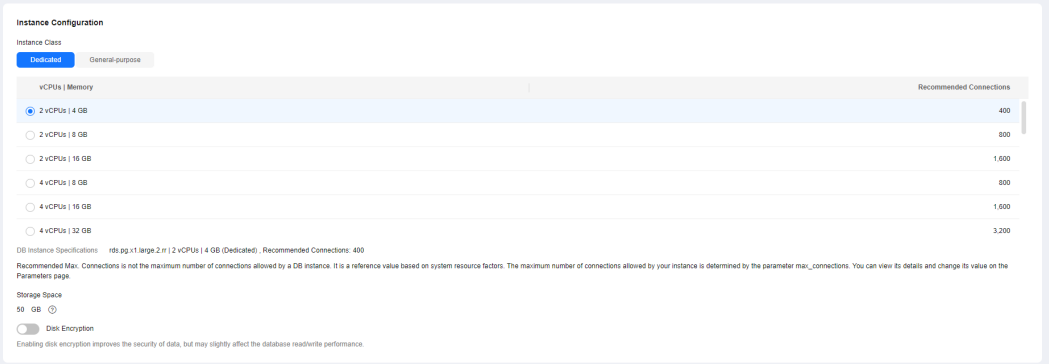


Table 3-33 Instance specifications

Parameter	Description
Instance Class	Refers to the vCPU and memory of a DB instance. Different instance classes have different numbers of database connections and maximum IOPS. After a DB instance is created, you can change its CPU and memory. For details, see section **Changing a DB Instance Class**. DB instances in a DCC only support the general-enhanced instance class.
Storage Space	Contains the system overhead required for inodes, reserved blocks, and database operation. By default, storage space of a read replica is the same as that of the primary DB instance.

Parameter	Description
Disk Encryption	• **Disable**: indicates the encryption function is disabled. • **Enable**: indicates the encryption function is enabled. Enabling disk encryption improves security but affects system performance. **Key Name**: indicates the tenant key. You can select an existing key or create a new one. **NOTE** – If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. – After an RDS DB instance is created, do not disable or delete a key that is currently in use. Otherwise, RDS will be unavailable and data cannot be restored. – For details about how to create a key, see "**Creating a CMK**" in *Data Encryption Workshop User Guide*.

Figure 3-100 Connectivity and additional options

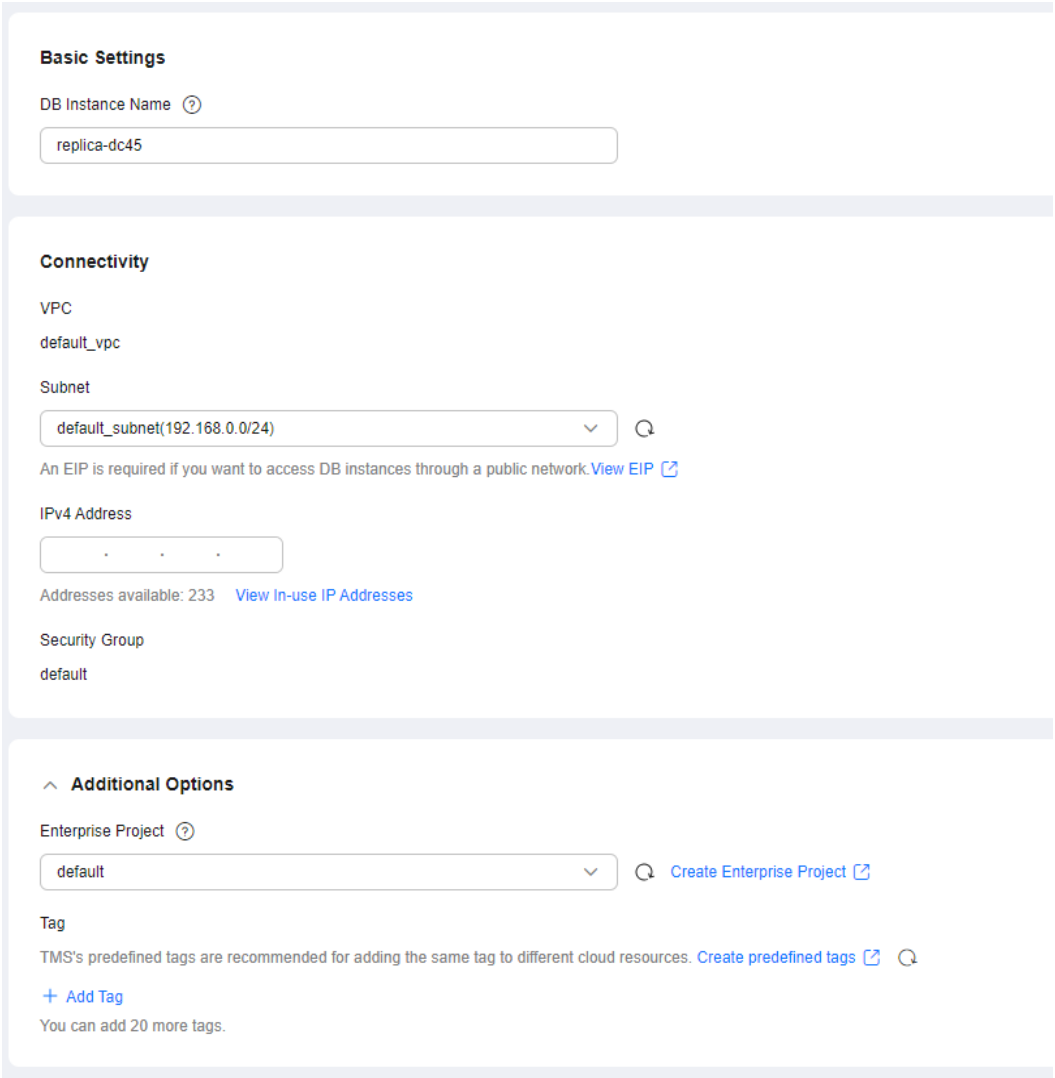


Table 3-34 Connectivity

Parameter	Description
DB Instance Name	Different DB instances can have the same name. The instance name must start with a letter and consist of 4 to 64 characters. Only letters (case-sensitive), digits, hyphens (-), and underscores (_) are allowed.
VPC	Same as the primary DB instance's VPC.
Subnet	Same as the primary instance's subnet. • IPv4 address: A floating IPv4 address is automatically assigned when you create a read replica. You can also enter an unused floating IPv4 address in the subnet CIDR block. After the read replica is created, you can change the floating IP address. • IPv6 address: A read replica assigned a floating IPv6 address will be created only when the vCPUs and memory you selected support IPv6 addresses. A floating IPv6 address is automatically assigned during read replica creation and cannot be specified. After the read replica is created, this floating IP address cannot be changed.
Security Group	Same as the primary DB instance's security group.

Table 3-35 Additional options

Parameter	Description
Enterprise Project	If your account has been associated with an enterprise project, select the target project from the **Enterprise Project** drop-down list. For more information about enterprise projects, see ***Enterprise Management User Guide***.
Tag	Optional. Tags help you easily identify and manage your read replicas. A maximum of 20 tags can be added for each read replica. After a read replica is created, you can view its tag details on the **Tags** page. For details, see **Managing Tags**.

Table 3-36 Yearly/monthly read replicas

Parameter	Description
Required Duration	The system will automatically calculate the configuration fee based on the selected required duration. The longer the required duration is, the larger discount you will enjoy.
Auto-renew	• By default, this option is not selected. • If you select this option, the auto-renew cycle is determined by the selected required duration.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit** for pay-per-use read replicas.
- For yearly/monthly read replicas, click **Pay Now**.

Step 7 After a read replica has been created, you can view and manage it on the **Instances** page by clicking on the left of the DB instance to which it belongs.

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

----End

FAQ

Q: Does creating read replicas during peak hours increase the load on my primary instance when my primary instance's CPU usage is high?

A: Yes. When a read replica is created, it synchronizes data from the primary instance, which consumes I/O and CPU resources of the primary instance. To avoid this impact, you can create read replicas during off-peak hours.

Follow-up Operations

Managing a Read Replica

3.11.3 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

----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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

3.11.4 Configuring Replication Delay for a Read Replica

Scenarios

You can configure a replication delay for a read replica. The read replica replays the WAL logs received from the primary instance after the specified delay.

If any data is deleted from your DB instance by mistake, it will be deleted from the read replica after the specified delay. During this time period, you can stop WAL log replay on the read replica, dump the data from the read replica, and insert it into the primary instance to restore the data.

Constraints

- To use this function, **submit a service ticket** to request required permissions.
- This function is only available to read replicas of RDS for PostgreSQL 12 and later versions.
- After you enable WAL replay to a specific time, the replication between the primary instance and read replica will be interrupted. During this period, if the WAL logs of the primary instance are rotated, the read replica will be disconnected from the primary instance (data of the primary instance cannot be synchronized to the read replica). To reconnect the read replica to the primary instance, reset the replay time and **submit a service ticket** to restore the read replica. The time required depends on how much data there is in the read replica and how many available resources there are in the primary instance.

Configuring Read Replica Delay

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the instance list, click ⊞ in front of the target DB instance and then click the name of the target read replica.

Step 5 Go to the **Read Replica Delay** page or refresh the page. The read replica delay details at the current time are displayed.

Figure 3-101 Read Replica Delay Details

Table 3-37 Read replica delay details

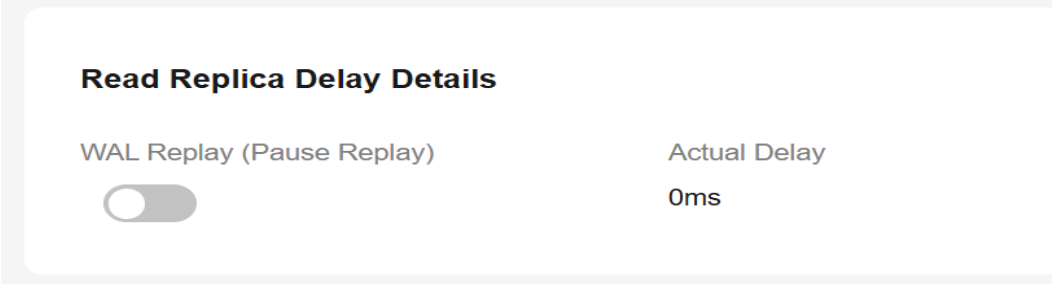
Parameter	Description
WAL Replay	The WAL replay status of the read replica. You can pause or resume WAL replay by toggling the switch. • Toggling on the switch: WAL replay starts. • Toggling off the switch: WAL replay pauses.
Actual Delay	The actual delay for WAL replay on the read replica during page loading or refreshing, in ms. The value of this parameter is slightly different from that of **Custom Delay**. For example, if there are no data writes to the primary instance and no WAL logs are synchronized, the actual delay is 0 ms.
Custom Delay	User-defined delay, in ms. Click ✎ to edit it. The delay cannot be too long. Otherwise, if the primary instance is heavily loaded, a large number of WAL logs are stacked on the read replica, causing full storage issues or even read replica disconnection.
Latest Replayed WAL Log Position	The latest WAL log position replayed by the read replica.
Latest Received WAL Log Position	The latest WAL log position received by the read replica.

----End

Replaying WAL Data to a Specific Time

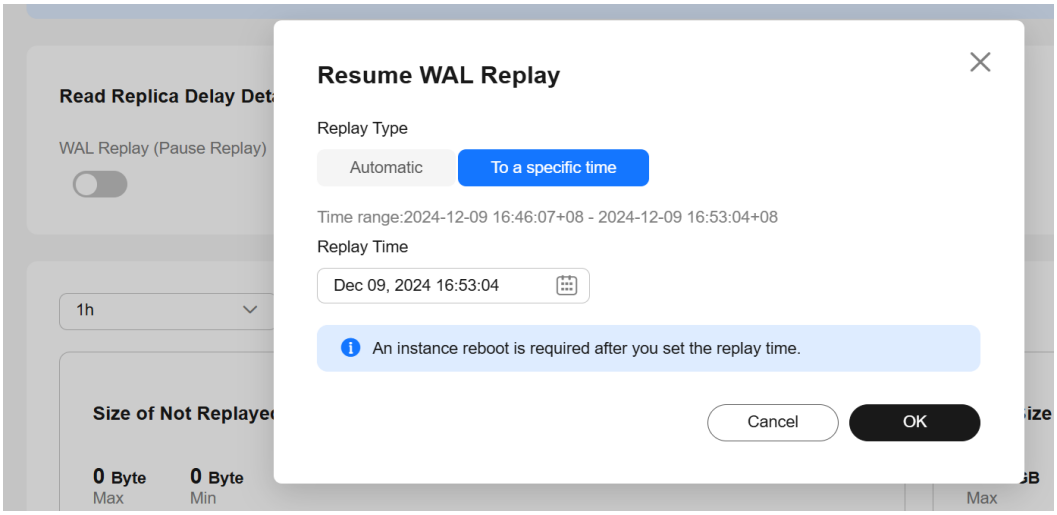
Step 1 On the **Read Replica Delay** page of the read replica, pause WAL replay.

Figure 3-102 Pausing replay



Step 2 Toggle on the replay switch and select **To a specific time** for **Replay Type**. The replay will be paused after the WAL logs are replayed to the specified time.

Figure 3-103 Replaying to a specific time



NOTE

Automatic indicates the normal replay state. The read replica will continuously replay the received WAL logs.

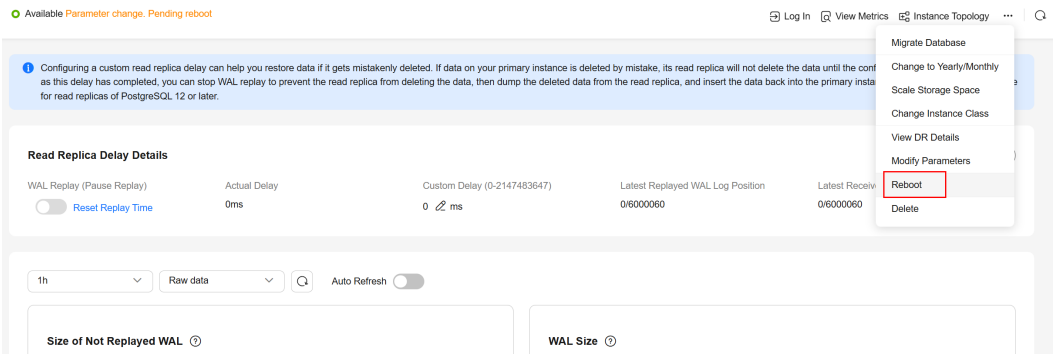
Step 3 Specify **Replay Time** and click **OK**.

NOTE

The read replica replays WAL logs to the closest transaction commit time before the target replay time.

Step 4 Click **Reboot** to replay WAL logs to the specified time.

Figure 3-104 Rebooting the read replica



Step 5 Check the replay status.

Figure 3-105 Replaying

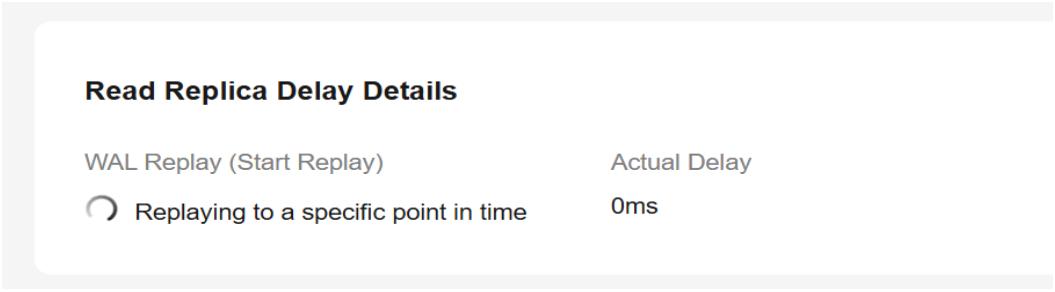
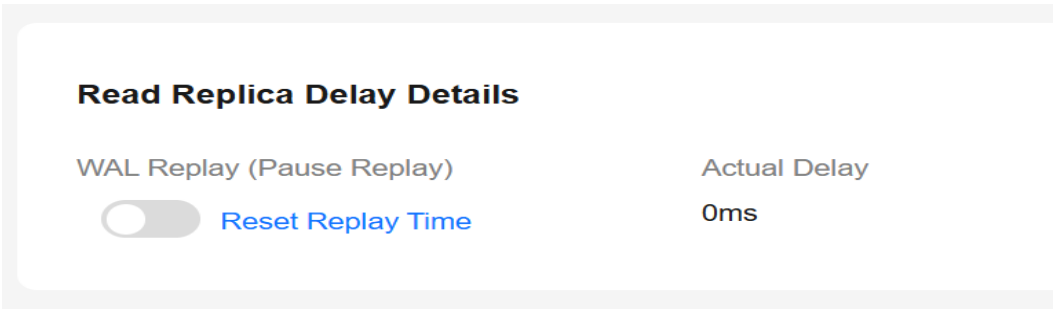


Figure 3-106 Replay completed



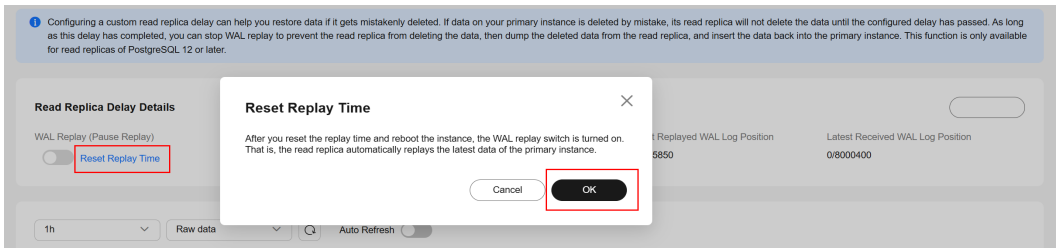
Step 6 To restore the replay, reset the replay time and reboot the read replica.

1. Click **Reset Replay Time**. In the displayed dialog box, click **OK**.

   NOTE

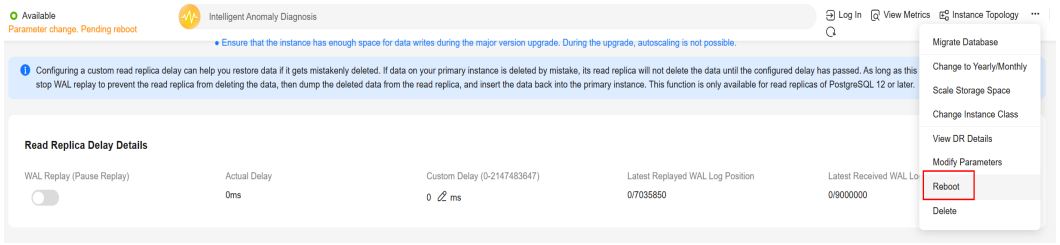
   After the read replica is rebooted, there will still be a replay delay. The delay depends on the value of **recovery_min_apply_delay**. If you want to immediately replay the WAL logs to the specified time, set this parameter to **0**.

Figure 3-107 Resetting the replay time



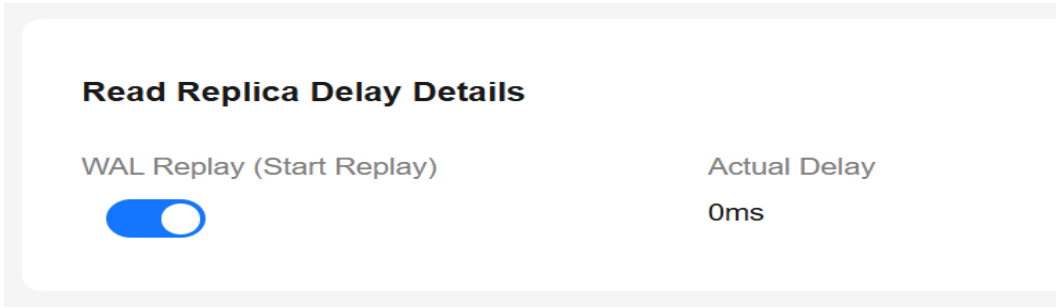
2. Click **Reboot**.

Figure 3-108 Rebooting the read replica



3. Check whether the replay status becomes normal.

Figure 3-109 Normal replay



----End

3.12 DR Management

3.12.1 Creating a DR Relationship

Scenarios

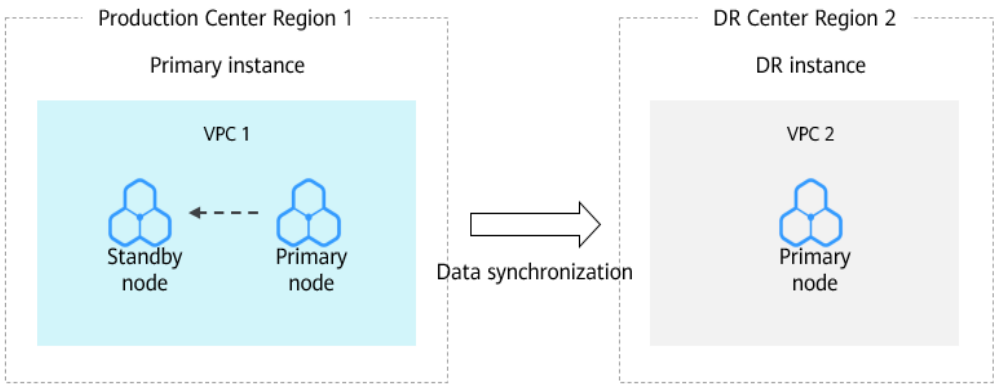
After a cross-region DR relationship is created, if the region where the primary instance is located encounters a natural disaster and the primary instance cannot be connected, you can promote the DR instance in another region to primary. To connect to the new primary instance, you only need to change the connection address on the application side.

How Cross-Region DR Works

Two RDS for PostgreSQL instances are deployed in two data centers, one in the production center and the other in the DR center. The RDS DR function replicates

data from the primary instance in the production center to the DR instance in the DR center, keeping data synchronous across the regions.

Figure 3-110 Topology



Precautions

- Before using this function, ensure that the network between the DB instances in two different regions is connected. You can **create a cloud connection** to connect the VPCs in different regions.
- Before using this function, ensure that the primary instance and DR instance are available and are deployed in different regions. The primary instance uses a primary/standby deployment and the DR instance uses a single-node deployment.
- The CPUs, memory, and storage space of the DR instance must be greater than or equal to those of the primary instance.
- The underlying architecture and major version of the DR instance must be the same as those of the primary DB instance.
- Cross-cloud or cross-region DR relationships cannot be established across major versions.
- After the API for configuring DR for the primary instance is called, you cannot change the instance class or perform a primary/standby switchover until the DR relationship is set up.
- After a DR relationship is set up, you can change the instance class of the DR instance. To use this function, **submit a service ticket**.
- After changing the database port or private IP address of the primary instance, you need to re-establish the DR relationship.
- After a DR instance is set up, a minor version upgrade cannot be performed.
- RDS for PostgreSQL 12 and later versions support cross-region DR.
- Modifying a parameter of the primary instance does not modify that of the DR instance synchronously. You need to modify the parameter of the DR instance separately based on service requirements.
- RDS for PostgreSQL DR instances do not support point-in-time recovery (PITR) or CBR snapshot-based backups. Perform such operations on the primary instance if needed.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Paste the information of the DR instance to the primary instance and configure DR for the primary instance.

1. On the **Instances** page, click the name of an instance (to act as the DR instance) that you want to create a DR relationship for.
2. Click **DR Information**.
3. In the displayed dialog box, click **Copy**.
4. On the **Instances** page, select another instance (to act as the primary instance) and choose **More** > **Create DR Relationship**.
5. On the displayed page, confirm the information and click **Create DR Relationship**.
6. In the displayed dialog box, paste the information you copied in **Step 4.3** to the text box and click **OK**.
7. Check the task execution result on the **Task Center** page. If the task status is **Completed**, the DR settings are configured for the primary instance. Perform subsequent operations only after this task is executed.

Step 5 Paste the information of the primary instance to the DR instance and configure DR for the DR instance.

1. On the **Instances** page, click the name of the primary instance.
2. Click **DR Information**.
3. In the displayed dialog box, click **Copy**.
4. On the **Instances** page, select the DR instance and choose **More** > **Create DR Relationship**.
5. On the displayed page, confirm the information and click **Create DR Relationship**.
6. In the displayed dialog box, paste the information you copied in **Step 5.3** to the text box and click **OK**.
7. Check the task execution result on the **Task Center** page. If the task status is **Completed**, the DR settings are configured for the DR instance. After this task is executed, the DR relationship is created.

Step 6 On the **Instances** page, click the name of the DR instance.

Step 7 In the navigation pane on the left, choose **DR Management**. The DR replication status of the DR instance is displayed.

----End

3.12.2 Promoting a DR Instance to Primary

Scenarios

If the region where the primary instance is located suffers a natural disaster and the primary instance cannot be connected, you can promote the DR instance in another region to primary. To connect to the new primary instance, you only need to change the connection address on the application side.

Precautions

After a DR instance is promoted to primary, the DR relationship between it and the original primary instance is removed.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the name of the DR instance.

Step 5 In the navigation pane, choose **DR Management**.

Step 6 In the DR relationship list, click **Promote DR Instance to Primary** in the **Operation** column.

Step 7 In the displayed dialog box, click **OK**.

Step 8 Check the task execution result on the **Task Center** page. If the task status is **Completed**, the promotion is successful.

Step 9 Switch your workload to the new primary instance.

----End

3.12.3 Removing a DR Relationship

Scenarios

If a DR relationship is not required, you can remove it.

Precautions

Only the DR relationship that has been successfully established can be removed. You must first remove the DR relationship of the DR instance and then that of the primary instance. Otherwise, an alarm may be generated.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Remove the DR relationship of the DR instance.

1. On the **Overview** page of the primary instance, click **DR Information**.
2. In the displayed dialog box, click **Copy**.
3. On the **Instances** page, click the name of the DR instance.
4. In the navigation pane, choose **DR Management**.
5. In the DR relationship list, click **Remove DR Relationship** in the **Operation** column.
6. Paste the copied DR information to the dialog box.
7. Check the task execution result on the **DR Management** page. If the list is deleted, the task is successfully executed.

Step 5 Remove the DR relationship of the primary instance by referring to **Step 4**.

----End

3.13 Extension Management

3.13.1 Installing and Uninstalling an Extension on the RDS Console

Scenarios

RDS allows you to install and uninstall extensions on the console.

RDS for PostgreSQL extensions only take effect on the databases you created the extensions for. To use an extension on databases, it has to be created separately for each database.

Prerequisites

Before installing or uninstalling extensions, ensure that there are databases in your instance. For details about how to log in to a DB instance, see **Logging In to an RDS for PostgreSQL Instance and Creating a Database Through DAS (Recommended)**.

Precautions

- plpgsql is a built-in extension and cannot be uninstalled.

- Logical replication extensions, such as decoderbufs (available only in RDS for PostgreSQL 11 to 14) and wal2json (available only in RDS for PostgreSQL 11 and later versions), can be directly used without installation.
- Some extensions depend on the **shared_preload_libraries** parameter. They can be installed only after related libraries are loaded.
- pg_cron is only available to RDS for PostgreSQL 12 (12.11.0 and later), 13, and later versions. Before using this extension, change the value of **cron.database_name** to the name of the database this extension is used for (only one database is supported), and change the value of **cron.use_background_workers** to **on**.
- pltcl is not supported for RDS for PostgreSQL 13.2. To use this extension, upgrade your instance to the latest minor version.
- Installing or uninstalling some extensions will cause their dependent extensions and tables to be installed or uninstalled synchronously. For example, when you install or uninstall postgis, postgis_sfcgal will be installed or uninstalled at the same time.
- Some extensions cannot be upgraded after a minor version upgrade. To upgrade them, uninstall them first and install them again.

Modifying the shared_preload_libraries Parameter

Some extensions require corresponding parameter values to be loaded before the extensions can be installed.

You can modify the **shared_preload_libraries** parameter to load parameter values in batches or load each required parameter value independently before installing an extension.

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

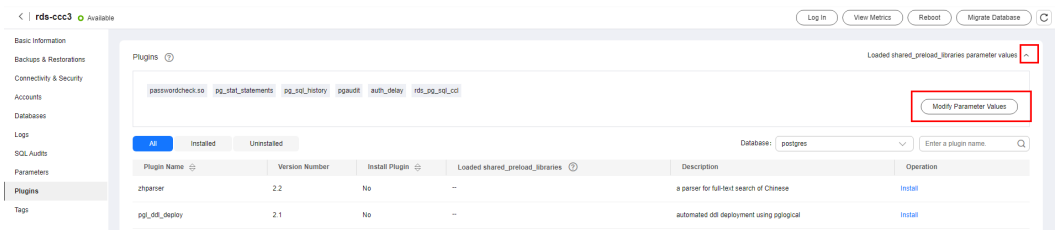
Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Plugins**.

Step 6 On the **Plugins** page, click next to **Loaded shared_preload_libraries parameter values** to view the loaded parameter values.

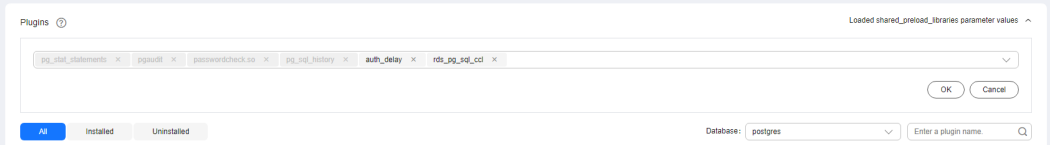
Step 7 Click **Modify Parameter Values**.

Figure 3-111 Viewing loaded parameter values



Step 8 Select the parameter values to be loaded from the drop-down list box and click **OK**.

Figure 3-112 Selecting parameter values to be loaded



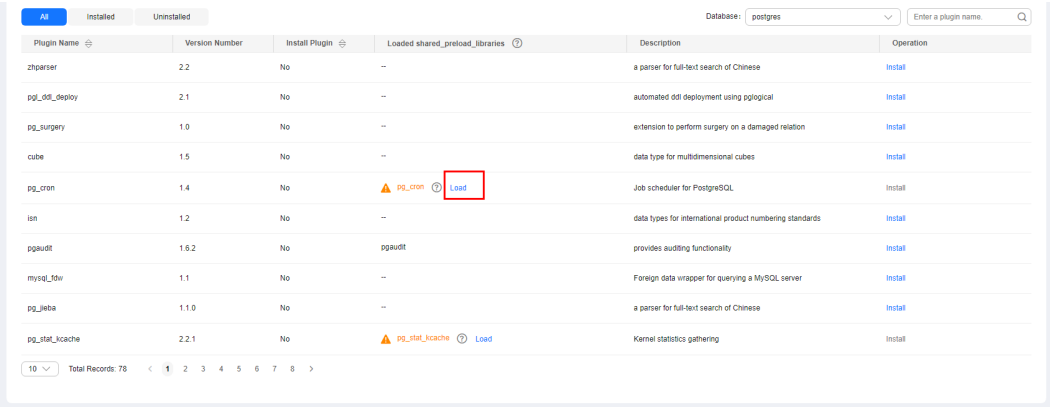
Step 9 In the displayed dialog box, click **OK**.

NOTE

- The modified parameter values take effect only after the instance is rebooted. If your instance has read replicas, the parameter values for the read replicas are also modified. You also need to reboot the read replicas.
- To ensure security and O&M functions of RDS for PostgreSQL, the following parameter values are loaded by default and cannot be deleted:
  - **passwordcheck.so**
  - **pg_stat_statements**
  - **pg_sql_history**
  - **pgaudit**

Step 10 You can also load each parameter value independently before installing an extension.

Figure 3-113 Loading a parameter value



----End

Installing and Uninstalling an Extension

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

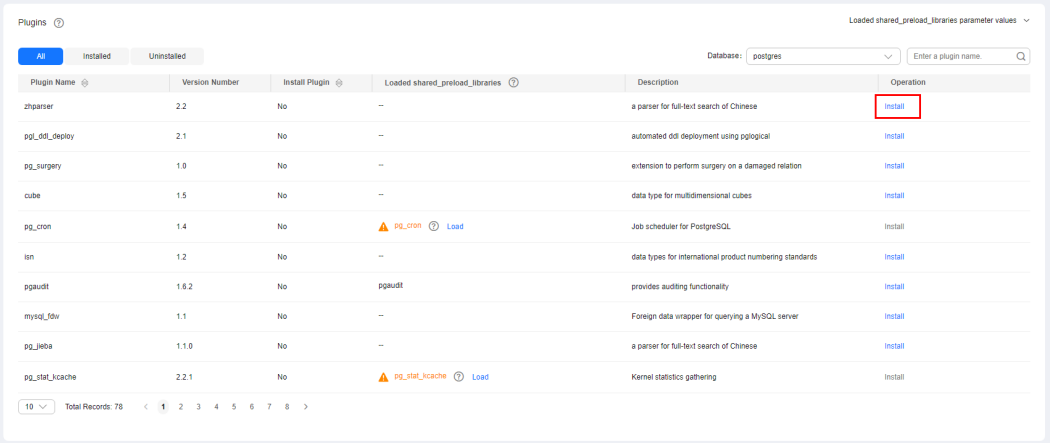
Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane, choose **Plugins**.

Step 6 In the **Database** drop-down list above the extension list, select the database where the extension is to be installed.

Step 7 Locate the extension to be installed and click **Install** in the **Operation** column.

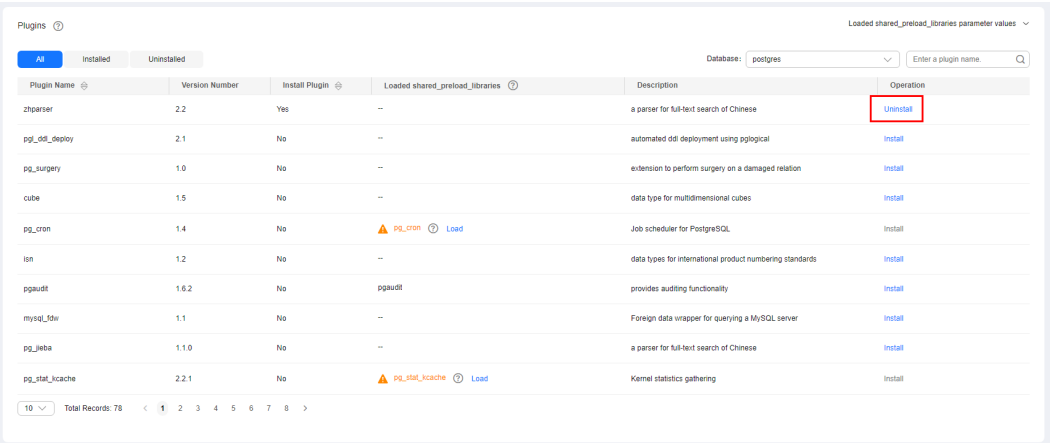
Figure 3-114 Installing an extension



Step 8 After a minor version upgrade, click **Update** next to the extension to be updated.

Step 9 To uninstall an extension, click **Uninstall**.

Figure 3-115 Uninstalling an extension



----End

3.13.2 Installing and Uninstalling an Extension Using SQL Commands

RDS provides the PostgreSQL extension management solution for user **root**. Except the following extensions, you need to manually create other extensions by referring to this section.

- auto_explain

- passwordcheck
- pg_profile_pro
- pg_sql_history
- plpgsql
- wal2json
- test_decoding

NOTE

RDS for PostgreSQL extensions only take effect on the databases you created the extensions for. To use an extension on databases, it has to be created separately for each database.

The latest minor versions of RDS for PostgreSQL 11 and later versions allow user **root** to create extensions (create extension) or delete extensions (drop extension).

Creating an Extension

Connect to the database where an extension needs to be created as user **root** and run the following SQL statements:

select control_extension('create','*<EXTENSION_NAME>***', '***<SCHEMA>***');**

- *EXTENSION_NAME* indicates the extension name. For more information, see **Supported Extensions**.
- *SCHEMA* indicates the name of the schema where the extension is created. If this parameter is not specified, the **public** schema is used by default.

Example:

Create postgis in the public schema.

```
-- Specify the public schema for creating the extension.
select control_extension('create','postgis', 'public');
      control_extension
 ------------------------------
  create postgis successfully.
 (1 row)
-- If the schema parameter is not specified, the default schema is public.
select control_extension('create', 'postgis');
     control_extension
------------------------------
 create postgis successfully.
(1 row)
```

Deleting an Extension

Connect to the database where an extension needs to be created as user **root** and run the following SQL statements:

select control_extension('drop','*<EXTENSION_NAME>***', '***<SCHEMA>***');**

- *EXTENSION_NAME* indicates the extension name. For more information, see **Supported Extensions**.
- SCHEMA indicates the schema name. This parameter does not matter much when you delete an extension, so you do not need to specify this parameter.

Example:


```
select control_extension('drop','postgis');
     control_extension
 ---------------------------
  drop postgis successfully.
 (1 row)
```

Common Errors

- Error 1
  ```
  ERROR:  permission denied for function control_extension
  ```
  Solution: Use **root** to run the **control_extension** function.
- Error 2
  ```
  ERROR:  function control_extension(unknown, unknown) is not unique
  ```
  Solution: Add the schema parameter in the function. If the schema is not specified, there may be functions with the same name, causing execution failures.
- Error 3
  ```
  ERROR:  function control_extension(unknown, unknown) does not exist
  ```
  Solution: Do not create extensions in the **postgres** database. The **control_extension** function does not exist in the **postgres** database because this database is used as an O&M database.

3.13.3 Supported Extensions

NOTE

The following table lists the extensions supported by the latest minor versions of RDS for PostgreSQL. You can use **SELECT name FROM pg_available_extensions;** to view the extensions supported by your DB instance. If any of the following extensions is not supported by your PostgreSQL version, **migrate your data to an RDS for PostgreSQL instance of the latest version**.

The extensions mysql_fdw, dblink, pgsql-ogr-fdw, postgres_fdw, and tds_fdw are used to access data stored in remote database servers. Before using any of them, ensure that the server IP addresses of the two DB instances are in the same VPC and subnet.

Extensions of RDS for PostgreSQL 13, 14, and 15 are available only for users with the open beta test (OBT) permission. You can **submit a service ticket** to apply for the permission.

Table 3-38 Supported extensions

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
address_standardizer	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
address_standardizer_data_us	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
amcheck	-	-	-	1.1	1.2	1.2	1.3	1.3	1.3
auth_delay	-	-	-	-	2	2	2	2	2

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
auto_explain	2	2	2	2	2	2	2	2	2
autoinc	-	-	-	-	1	1	1	1	1
bloom	-	-	-	1.0	1.0	1.0	1.0	1.0	1
btree_gin	1.0	1.0	1.2	1.3	1.3	1.3	1.3	1.3	1.3
btree_gist	1.1	1.2	1.5	1.5	1.5	1.5	1.6	1.7	1.7
citext	1.1	1.3	1.4	1.5	1.6	1.6	1.6	1.6	1.6
cube For details, see **cube**.	1.0	1.2	1.2	1.4	1.4	1.4	1.5	1.5	1.5
dblink	1.1	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
dict_int	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
dict_xsyn	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
earthdistance For details, see **earthdistance**.	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
fuzzystrmatch	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.2
hll	2.12	2.12	2.12	2.12	2.14	2.18	2.18	2.18	2.18
hstore	1.3	1.4	1.4	1.5	1.6	1.7	1.8	1.8	1.8
hypopg	-	-	-	1.4.0	1.4.0	1.4.0	1.4.0	1.4.0	1.4.0
icu	-	-	-	1.0	1.0	1.0	1.0	1.0	1
insert_username	-	-	-	-	1	1	1	1	1
intagg	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
intarray	1.0	1.2	1.2	1.2	1.2	1.3	1.5	1.5	1.5
ip4r	-	-	-	-	2.4.2	2.4.2	2.4.2	2.4.2	2.4.2
isn	1.0	1.1	1.1	1.2	1.2	1.2	1.2	1.2	1.2

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
jsonb_plperl	-	-	-	-	1	1	1	1	1
lo	-	-	-	-	1.1	1.1	1.1	1.1	1.1
ltree	1.0	1.1	1.1	1.1	1.1	1.2	1.2	1.2	1.2
moddatetime	-	-	-	-	1	1	1	1	1
mysql_fdw	-	-	-	2.9.1	2.9.1	2.9.1	2.9.1	2.9.1	2.9.1
old_snapshot	-	-	-	-	-	-	1.0	1.0	1
orafce	3.8.0	3.8.0	3.8.0	3.8.0	3.8.0	3.14.0	3.21.1	4.4.0	4.4.0
pageinspect	1.3	1.5	1.6	1.7	1.7	1.8	1.9	1.11	1.12
passwordcheck	2	2	2	2	2	2	2	2	2
pgAudit	-	-	-	-	1.6.2	1.6.2	1.6.2	1.7.0	16
pg_bigm	-	-	-	1.2_20200228	1.2_20200228	1.2_20200228	1.2_20200228	1.2_20200228	-
pg_buffercache	1.1	1.2	1.3	1.3	1.3	1.3	1.3	1.3	1.4
pg_cron	-	-	-	-	1.6.2	1.6.2	1.6.2	1.6.2	1.6.2
pg_freespacemap	1.0	1.1	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pg_hint_plan	1.1.5	1.2.0	1.3.0	1.3.5	1.3.9	1.3.9	1.4.2	1.5.1	1.6.0
pg_jieba	1.1.0	1.1.0	1.1.0	1.1.0	1.1.0	2.0.1	1.1.0	1.1.0	-
pg_partman	-	-	-	-	-	-	5.0.1	5.0.1	5.0.1
pg_pathman	1.5.8	1.5.8	1.5.8	1.5.8	1.5.12	1.5.12	-	-	-
pg_prewarm	1.0	1.1	1.1	1.2	1.2	1.2	1.2	1.2	1.2
pg_qualstats	-	-	-	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0	2.1.0

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
pg_repack	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0	1.5.0
pg_roaringbitmap	-	-	-	0.5.4	0.5.4	0.5.4	0.5.4	0.5.4	0.5.4
pg_stat_kcache	-	-	-	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3	2.2.3
pg_stat_statements	1.3	1.4	1.6	1.6	1.7	1.8	1.9	1.10	1.1
pg_surgery	-	-	-	-	-	-	1.0	1.0	1
pg_tle	-	-	-	-	-	1.2.0	1.2.0	1.2.0	1.2.0
pg_track_settings	-	-	-	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2	2.1.2
pg_trgm	1.1	1.3	1.3	1.4	1.4	1.5	1.6	1.6	1.6
pg_visibility	-	-	-	1.2	1.2	1.2	1.2	1.2	1.2
pg_wait_sampling	-	-	-	1.1.5	1.1.5	1.1.5	1.1.5	1.1.5	1.1.5
pgcrypto	1.2	1.3	1.3	1.3	1.3	1.3	1.3	1.3	1.3
pgl_ddl_deploy	-	-	-	-	2.1.0	2.1.0	2.1.0	2.1.0	2.2.1
pglogical	-	-	-	2.4.4	2.4.4	2.4.4	2.4.4	2.4.4	2.4.4
pg_profile_pro For details, see **pg_profile_pro**.	-	-	-	-	1.0	-	-	-	-
pgrouting	-	-	-	3.1.0	3.1.0	3.1.4	3.3.1	3.5.0	3.6.1
pgrowlocks	1.1	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
pg_sql_history For details, see •**pg_sql_history**.	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2
pgsql-ogr-fdw	-	-	-	1.1.3	1.1.3	1.1.3	-	1.1.3	1.1.4
pgstattuple	1.3	1.4	1.5	1.5	1.5	1.5	1.5	1.5	1.5
pgvector	-	-	-	-	0.6.1	0.6.1	0.6.1	0.6.1	0.6.1
plpgsql For details, see **plpgsql**.	1.0	1.0	1.0	1.0	1.0	1.0	1	1.0	1
plperl	-	-	-	1.0	1.0	1.0	1.0	1.0	1
plprofiler	-	-	-	-	4.2.4	4.2.4	4.2.4	4.2.4	4.2.4
plproxy	-	-	-	2.11.0	2.11.0	2.11.0	2.11.0	2.11.0	2.11.0
plv8	-	-	-	2.3.15	2.3.15	2.3.15	-	-	-
postgis For details, see **postgis**.	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_raster	Integrated to postgis	Integrated to postgis	Integrated to postgis	Integrated to postgis	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_sfcgal	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_tiger_geocoder	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgis_topology	2.5.1	2.5.1	2.5.1	2.5.1	3.0.0	3.1.0	3.2.6	3.4.1	3.4.1
postgres_fdw	1.0	1.0	1.0	1.0	1.0	1.0	1.1	1.1	1.1

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
postgres-decoderbufs	-	-	-	1.7.0	1.7.0	1.7.0	1.7.0	-	-
postgresql_anonymizer	-	-	-	0.7.1	0.7.1	0.7.1	1.1.0	1.1.0	1.1.0
q3c	-	-	-	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1	2.0.1
rum	-	-	-	1.3.13	1.3.13	1.3.13	1.3.13	1.3.13	1.3.13
seg	-	-	-	-	1.3	1.3	1.4	1.4	1.4
sslinfo	-	-	-	1.2	1.2	1.2	1.2	1.2	1.2
tablefunc	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
tcn	-	-	-	-	1	1	1	1	1
tds_fdw	-	-	2.0.3	2.0.3	2.0.3	2.0.3	2.0.3	2.0.3	2.0.3
test_decoding	2	2	2	2	2	2	2	2	2
TimescaleDB For details, see **TimescaleDB**.	0	1.3.2	1.3.2	1.3.2	1.7.0	2.1.0	2.7.0	2.11.1	2.14.2
tsm_system_rows	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
tsm_system_time	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1
unaccent	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
uuid-ossp	1.0	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1
wal2json For details, see **wal2json**.	-	-	-	2.5	2.5	2.5	2.5	2.5	2.5
xml2	-	-	-	1.1	1.1	1.1	1.1	1.1	1.1
zhparser	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2

Extension Name	PostgreSQL 9.5	PostgreSQL 9.6	PostgreSQL 10	PostgreSQL 11	PostgreSQL 12	PostgreSQL 13	PostgreSQL 14	PostgreSQL 15	PostgreSQL 16
pg_stat_monitor	-	-	-	-	2.0.4	2.0.4	2.0.4	2.0.4	2.0.4

Extension Description

- **postgis**
  - Creating postgis_topology and postgis_tiger_geocoder will change the **search_path** settings. However, this change will not take effect for established connections. To use the two extensions, re-establish a connection to update the **search_path** settings.
- **plpgsql**

  plpgsql 1.0 provides the SQL procedural language and is installed by default.
- **earthdistance**

  To install the earthdistance extension, you must install the cube extension first.
- **cube**

  If the earthdistance extension has been installed, deleting the cube extension will cause the earthdistance extension to be unavailable.
- **TimescaleDB**

  The **TimescaleDB** extension of RDS for PostgreSQL supports only the features of the Apache protocol. It does not support the features of the TSL protocol. For details, see **TimescaleDB Apache 2 and TimescaleDB Community Edition**.
- **wal2json**

  This extension is a logical replication extension. You can directly use it without installing it through control_extension.

  This extension cannot be queried from the **pg_available_extensions** view. You can run the following statement to check whether **wal2json** is supported. If no error is reported, **wal2json** is supported.

  select pg_create_logical_replication_slot('tst_wal2json', 'wal2json');

  After the statement is executed successfully, delete the slot to prevent stacked WAL logs.

  select pg_drop_replication_slot('tst_wal2json');
- **pg_profile_pro**

  This extension is not supported temporarily due to its defects. It will be open to you after the defects are rectified. We are sorry for any inconvenience caused.
- **pg_sql_history**

  This extension is used by **Database Security Service (DBSS)** to audit SQL operations of RDS for PostgreSQL instances. The SQL statements queried by this extension can be truncated because the space allocated to each record is

fixed at 4 KB, which covers not only the SQL statement, but also information such as the database name, username, and SQL type.

3.13.4 pg_profile_pro

Scenarios

pg_profile_pro is an extension used to monitor the performance and status of RDS for PostgreSQL DB instances. It provides monitoring data reports for SQL statements, tables, indexes, functions, transactions, and vacuum operations to detect existing or potential performance problems of databases.

Based on the pg_stat_statements view of PostgreSQL, this extension creates historical statistics in your DB instance and generates statistics samples. Periodic statistics samples are used to generate monitoring reports, helping identify resource-consuming activities.

CAUTION

This extension is not supported temporarily due to its defects. It will be open to you after the defects are rectified. We are sorry for any inconvenience caused.

Constraints

Only RDS for PostgreSQL 12 supports this extension.

Procedure

Step 1 Run the following command to connect to the **postgres** database as user **root** and obtain the sample list.

 # psql --host=<*RDS_ADDRESS*> **--port=**<*DB_PORT*>**--dbname=postgres --username=root -c "select * from profile.show_samples();"**

Table 3-39 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.

Enter the password of user **root** when prompted.

```
Password for user root:
```

Information similar to the following is displayed:

```
 sample |      sample_time       | dbstats_reset | clustats_reset | archstats_reset
--------+------------------------+---------------+----------------+-----------------
      1 | 2021-04-02 17:15:49+08 |               |                |
      2 | 2021-04-02 17:25:57+08 |               |                |
```



```
     3 | 2021-04-02 17:36:04+08 |           |           |
(3 rows)
```

Use the actual query result.

Step 2 Connect to the **postgres** database as user **root** and obtain the report using either of the following methods:

Method 1: Obtain the report based on the sample ID.

 # psql --host=<*RDS_ADDRESS*> --port=<*DB_PORT*> --dbname=postgres --username=root -Aqtc "select profile.get_report(<*sample_start_id*>, <*sample_end_id*>)" -o <*filename*>.html

Table 3-40 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
sample_start_id	Indicates the start sample ID contained in the report.
sample_end_id	Indicates the end sample ID contained in the report.
filename	Indicates the name of the file where the report content is to be saved. You can specify a relative path or an absolute path for the file.

NOTE

The value of *sample_start_id* must be smaller than that of *sample_end_id*. Otherwise, the report cannot be generated.

Method 2: Obtain the report by specifying a time period.

 # psql --host=<*RDS_ADDRESS*> --port=<*DB_PORT*> --dbname=postgres --username=root -Aqtc "select profile.get_report(tstzrange('*sample_start_time*', '*sample_end_time*'))" -o <*filename*>.html

Table 3-41 Parameter description

Parameter	Description
RDS_ADDRESS	Indicates the IP address of the RDS DB instance.
DB_PORT	Indicates the port of the RDS DB instance.
sample_start_time	Indicates the sample start time contained in the report.
sample_end_time	Indicates the sample end time contained in the report.

Parameter	Description
filename	Indicates the name of the file where the report content is to be saved. You can specify a relative path or an absolute path for the file.

NOTE

Retain the default values of the following parameters:

- Sampling period (unit: minute): 10
- Sample retention period (unit: day): 7
- Number of displayed records: 20

----End

Parameter Configuration

Table 3-42 Parameters affecting the sample report

Parameter	Default Value	Restart Required	Function	Remarks
pg_profile_pro.topn	20	No	Controls the number of objects (such as statements and relationships) to be reported in each sorting report.	This parameter affects the sample size. If you want to display more objects in the report, more objects need to be retained in the sample.
pg_profile_pro.max_sample_age	3	No	Indicates the sample retention period, in days. Samples whose retention period has exceeded the value of this parameter will be automatically deleted during the next sampling.	The minimum parameter granularity is day.

Parameter	Default Value	Restart Required	Function	Remarks
pg_profile_pro.track_sample_timings	off	No	Controls whether pg_profile_pro traces the detailed sampling time.	After this parameter is set to **on**, the time consumed by each sampling is recorded, which increases the space usage.
pg_profile_pro.period	3600	No	Indicates the sampling period, in seconds. Controls the sampling period.	This parameter affects the sample size. The shorter the sampling period is, the more objects the report reserves due to more samples in unit time.
pg_profile_pro.enable	on	No	Controls whether the extension collects samples. • **on**: yes. • **off**: no.	If this parameter is set to **off**, the latest report cannot be generated, but the historical report can be viewed.

NOTE

The default values in the table are for the latest RDS version and may vary in other versions.

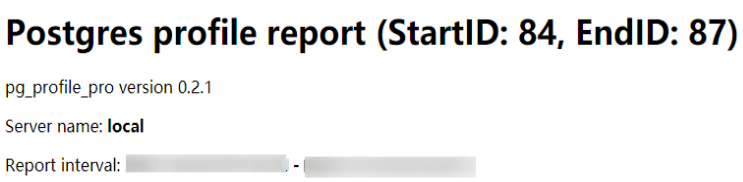
Report Example

A report consists of title, contents, and tables.

- Title

  The title contains start and end sample IDs, pg_profile_pro kernel version, server name, and report interval.

  Figure 3-116 Report title

- Contents

  Each layer of contents is linked to a table. You can get an overview of all tables from the contents or click the links to view each table.

  The following figure shows a contents example.

- Tables

  The tables display the database performance from different dimensions and focuses.

  – Each table is followed by suggestions on the focuses.

  – In each table, you can move the mouse pointer to a column title to view the comments (both in Chinese and English).

3.13.5 pg_repack

Scenarios

pg_repack can reorganize tables and indexes with minimal locks to restore the physical order. Unlike CLUSTER and VACUUM FULL it works online, without holding an exclusive lock on the processed tables during processing.

Constraints

- Only the **root** user can use pg_repack.

- The target table must have a primary key or at least a unique total index on a NOT NULL column.
- Performing a full-table repack requires free disk space about twice as large as the target table and its indexes.
- pg_repack cannot reorganize temp tables or cluster tables by GiST indexes.
- You will not be able to perform DDL commands of the target table except VACUUM or ANALYZE while pg_repack is working.
- pg_repack can be used only after a client is deployed locally. For details, see the official documentation at **https://reorg.github.io/pg_repack/**.

How to Use

- Install the extension.
  ```
  select control_extension('create', 'pg_repack');
  ```
- Delete the extension.
  ```
  select control_extension('drop', 'pg_repack');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

Example

Use pg_repack to repack a table.

1. **Create a test table pg_repack_test.**
   ```
   create table pg_repack_test(id bigint primary key, name varchar);
   insert into pg_repack_test select i , to_char(random()*100000, 'FM000000') from generate_series(1, 1000000) i;
   delete from pg_repack_test where id in (select i  from generate_series(1, 600000, 2) i);
   select pg_size_pretty(pg_relation_size('pg_repack_test'));
   ```
2. **Repack the test table.**
   ```
   pg_repack --host=<RDS_ADDRESS> --port=<DB_PORT> --dbname=<DB_NAME> --username=root --no-superuser-check --no-kill-backend -t pg_repack_test
   ```
   - *RDS_ADDRESS*: IP address of the RDS DB instance.
   - *DB_PORT*: Port of the RDS DB instance.
   - *DB_NAME*: Name of the database where the pg_repack_test table is located.
3. **Check the size of the repacked table.**
   ```
   select pg_size_pretty(pg_relation_size('pg_repack_test'));
   ```

FAQs

Table 3-43 Common error information and solutions

Error Information	Solution
ERROR: pg_repack failed with error: ERROR: permission denied for schema repack	Use the **root** user.
ERROR: pg_repack failed with error: You must be a superuser to use pg_repack	Add **--no-superuser-check** to skip superuser checks.

Error Information	Solution
NOTICE: Waiting for 1 transactions to finish. First PID: xxxx	Wait until the transaction is complete.

3.13.6 pgl_ddl_deploy

Introduction

There are many databases that require replicating data to other databases for various purposes. One of the most useful database technologies that is used to move data from point A to point B is called "logical replication". In database jargon, there are two categories of SQL statements: DML and DDL. For a number of reasons, DDL has to be handled separately. During the migration, the DBA is required to manually deploy the SQL in the correct order for all involved database clusters, manage locking contention, and add new tables to replication if necessary. Built on top of pglogical, pgl_ddl_deploy enables any DDL SQL statement to be directly propagated to subscribers. This solves the problem that pglogical cannot synchronize DDL statements.

For more information, see **official pgl_ddl_deploy documentation**.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 12 and later versions. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'pgl_ddl_deploy';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

For details about the extensions supported by RDS for PostgreSQL, see **Supported Extensions**.

Features

RDS for PostgreSQL supports the **pgl_ddl_deploy** extension, which is used to automatically synchronize DDL statements. In many cases, most DDL statements executed in application environments can be synchronized.

- Any DDL statement can be synchronized to subscribers.
- Tables can be automatically added to replication upon creation.
- Filtering by regular expression or a specific set of tables is supported.
- There is an option to deploy in a lock-safe way on subscribers.
- There is an option to fail certain events on the subscriber to be retried later.
- In some edge cases, alerting can be built around provided logging for the DBA to then handle possible manual deployments.

- **ALTER TABLE** statements can be filtered by subcommand tags.
- Support for automatically killing blocking processes that are preventing DDL execution on the subscriber system is optional.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'pgl_ddl_deploy');
  ```
- Deleting the extension
  ```
  SELECT control_extension ('drop', 'pgl_ddl_deploy');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

Basic Usage

This extension involves publication and subscription and depends on pglogical. You need to add and configure parameters.

```
wal_level = 'logical'
shared_preload_libraries = 'pglogical'
```

For details about how to modify the **shared_preload_libraries** parameter, see **Modifying the shared_preload_libraries Parameter**.

```
-- Configuring parameters on the provider
SELECT control_extension ('create', 'pglogical');
SELECT control_extension ('create', 'pgl_ddl_deploy');
CREATE TABLE foo (id INT PRIMARY KEY);
-- Creating a publication
CREATE PUBLICATION testpub FOR TABLE foo;
-- Configuring a replication set
INSERT INTO pgl_ddl_deploy.set_configs (set_name, include_schema_regex, driver) VALUES
('testpub', '.*', 'native'::pgl_ddl_deploy.driver);
-- Deploying the publication
SELECT pgl_ddl_deploy.deploy('testpub');
-- Adding roles for the user
SELECT pgl_ddl_deploy.add_role(oid) FROM pg_roles WHERE rolname='root';
-- Configuring parameters on the subscriber
SELECT control_extension ('create', 'pglogical');
SELECT control_extension ('create', 'pgl_ddl_deploy');
CREATE TABLE foo (id INT PRIMARY KEY);
-- Creating a subscription
CREATE SUBSCRIPTION testsub CONNECTION conninfo PUBLICATION testpub;
ALTER SUBSCRIPTION testsub REFRESH PUBLICATION;
```

After the configuration is complete, run the following DDL statements on the provider:

```
ALTET TABLE foo ADD COLUMN bla INT;
CREATE TABLE bra (id INT PRIMARY KEY);
```

You can verify the following on the subscriber:

```
\d foo
                Table "public.foo"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           | not null |
 bla    | integer |           |          |
```



```
Indexes:
    "foo_pkey" PRIMARY KEY, btree (id)
\dt
          List of relations
 Schema |  Name   | Type  |  Owner
--------+---------+-------+----------
 public | bar     | table | root
 public | foo     | table | root
(2 rows)
```

Restrictions

This extension has some limitations. Although most DDL statements executed in application environments can be synchronized, it does not cover 100% of edge cases.

DDL Involving Multiple Tables

A single DDL SQL statement which alters both replicated and non-replicated tables cannot be supported. For example, if you set the **include_schema_regex** parameter to **'^replicated.*'**:

```
DROP TABLE replicated.foo, notreplicated.bar;
```

The following message will be displayed on the provider:

```
WARNING: Unhandled deployment logged in pgl_ddl_deploy.unhandled
DROP TABLE
```

The **replicated.foo** table exists on the subscriber.

```
\d replicated.foo
              Table "replicated.foo"
Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           | not null |
Indexes:
    "foo_pkey" PRIMARY KEY, btree (id)
```

Similarly, if filtered replication is used, an error may occur when you run the following statement:

```
ALTER TABLE replicated.foo ADD COLUMN bar_id INT REFERENCES notreplicated.bar (id);
```

The statement is not synchronized to the subscriber.

```
\d replicated.foo
              Table "replicated.foo"
Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
id     | integer |           | not null |
Indexes:
    "foo_pkey" PRIMARY KEY, btree (id)
```

Unsupported Commands

CREATE TABLE AS and SELECT INTO are not supported to replicate DDL due to limitations on transactional consistency. That is, if a table is created from a set of data on the provider, running the same SQL on the subscriber will in no way guarantee consistent data. For example:


```
CREATE TABLE foo AS
SELECT field_1, field_2, now() AS refreshed_at
FROM table_1;
```

Similar to CREATE TABLE AS, the following message will be displayed for SELECT INTO:

```
WARNING:  Unhandled deployment logged in pgl_ddl_deploy.unhandled
```

Multi-Statement Client SQL Limitations

The complexities and limitations come when the client sends all SQL statements as one single string to PostgreSQL. Assume the following SQL statements:

```
CREATE TABLE foo (id serial primary key, bla text);
INSERT INTO foo (bla) VALUES ('hello world');
```

If this was in a file that was called using psql, it would run as two separate SQL command strings. However, if in Python or Ruby's ActiveRecord you create a single string as above and execute it, then it would be sent to PostgreSQL as one single SQL command string. The replication depends on the value of the **allow_multi_statements** parameter:

- If the value is **false**, pgl_ddl_deploy will only auto-replicate a client SQL statement containing one command tag that matches the event trigger command tag. That is really safe, but it means you may have a lot more unhandled deployments.
- If the value is **true**, pgl_ddl_deploy will only auto-replicate DDL that contains **safe** command tags to propagate. For example, mixed DDL and DML is forbidden. If a command contains more than two DDL statements and the statements are used on both replicated and non-replicated tables, the problem described in **DDL Involving Multiple Tables** occurs.

In any case that a SQL statement cannot be automatically run on the subscriber based on these analyses, instead it will be logged as a WARNING and put into the unhandled table for manual processing. For more details and solutions to problems that occur during replication, see **official pgl_ddl_deploy documentation**.

3.13.7 pgvector

Introduction

RDS for PostgreSQL supports the **pgvector** extension, which allows for vector data type and vector similarity search. This extension supports:

- Exact and approximate nearest neighbor search
- L2 distance, inner product, and cosine distance
- Any language with a PostgreSQL client

For more information, see **official pgvector documentation**.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 12 and later versions. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'vector';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

For details about the extensions supported by RDS for PostgreSQL, see **Supported Extensions**.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'vector');
  ```
- Deleting the extension
  ```
  SELECT control_extension ('drop', 'vector');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

Basic Functions

- Creating a vector column with 3 dimensions
  ```
  CREATE TABLE items (id bigserial PRIMARY KEY, embedding vector(3));
  ```
- Inserting vectors
  ```
  INSERT INTO items (embedding) VALUES ('[1,2,3]'), ('[4,5,6]');
  ```
- Getting the nearest neighbors by L2 distance
  ```
  SELECT * FROM items ORDER BY embedding <-> '[3,1,2]';
  ```
- Getting the nearest neighbors by cosine distance
  ```
  SELECT * FROM items ORDER BY embedding <=> '[3,1,2]';
  ```
- Getting the nearest neighbors by inner product

  <#> returns the negative inner product since PostgreSQL only supports ASC order index scans on operators.

  ```
  SELECT * FROM items ORDER BY embedding <#> '[3,1,2]';
  ```

Advanced Functions

- Getting the distance
  ```
  SELECT embedding <-> '[3,1,2]' AS distance FROM items;
  SELECT (embedding <#> '[3,1,2]') * -1 AS inner_product FROM items;
  SELECT 1 - (embedding <=> '[3,1,2]') AS cosine_similarity FROM items;
  ```
- Averaging vectors
  ```
  SELECT AVG(embedding) FROM items;
  ```
- Exact search providing perfect recall

  You can add an index to use approximate nearest neighbor search, which trades some recall for performance.

  ```
  CREATE INDEX ON items USING ivfflat (embedding vector_l2_ops) WITH (lists = 1);
  INSERT INTO items (embedding) VALUES ('[1,2,4]');
  SELECT * FROM items ORDER BY embedding <-> '[3,3,3]';
  ```


3.13.8 pgAudit

Introduction

Financial institutions, government agencies, and many industries need to keep audit logs to meet regulatory requirements. By using the PostgreSQL Audit Extension (pgAudit) with your RDS for PostgreSQL instance, you can capture detailed records that auditors usually need to meet compliance regulations. For example, you can use pgAudit to track changes made to specific databases and tables, as well as record users who make such changes and many other details.

For more information, see the **official pgAudit documentation**.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 12 and later versions. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'pgaudit';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'pgaudit');
  ```
- Deleting the extension
  ```
  SELECT control_extension ('drop', 'pgaudit');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

How to Use

Configuring the pgAudit

1. First, preload pgAudit on the **Plugins** page of your instance because pgAudit installs event triggers for auditing data definition language (DDL) statements. By default, pgAudit is preloaded. To check whether it is successfully loaded, you can run the following command:
   ```
   show shared_preload_libraries;
                          shared_preload_libraries
   ------------------------------------------------------------------------------
    pg_stat_statements,pgaudit,passwordcheck.so,pg_sql_history,auth_delay,pglogical
   (1 row)
   ```
2. After the extension is loaded, install it by referring to **Extension Installation and Uninstallation**.
3. After the extension is installed, enable audit logging.

NOTE

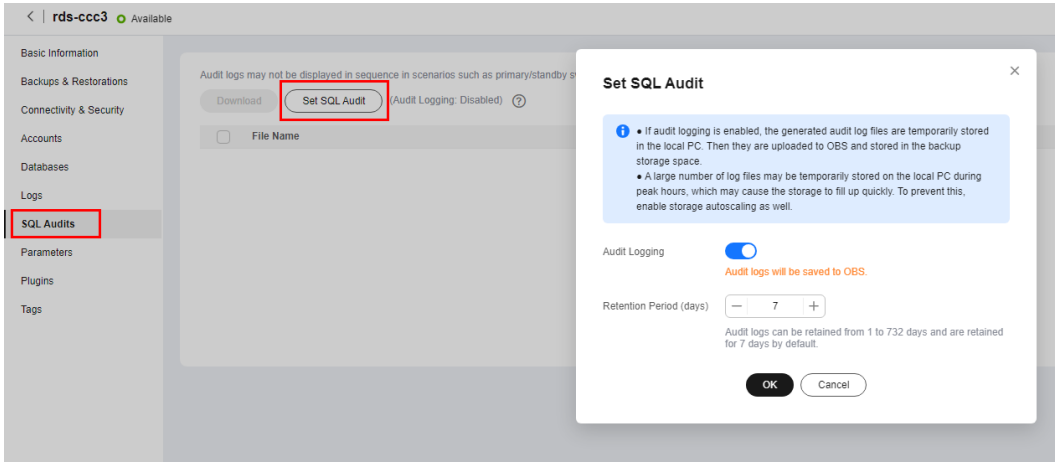
To enable audit logging, **submit a service ticket** to request required permissions.

a. On the RDS console, click the DB instance name. On the displayed page, click **SQL Audits**.

b. Click **Set SQL Audit**.

c. In the displayed dialog box, toggle on the audit log switch and set the number of days to retain audit logs.

Figure 3-117 Setting SQL audit



4. Configure parameters.

   Go to the **Parameters** page, search for the **pgaudit.log** parameter (that specifies which types of statements will be logged by session audit logging), and set it to an appropriate value to capture log insertions, updates, deletions, and other changes. The following table explains the values of **pgaudit.log**.

Table 3-44 Parameter description

Value	Description
NONE	(Original value) Specifies that no changes to the database will be recorded.
ALL	Specifies that all changes will be recorded, including READ, WRITE, FUNCTION, ROLE, DDL, and MISC.
DDL	Specifies that all DDL statements (excluding those in the ROLE class) will be recorded.
FUNCTION	Specifies that function calls and DO blocks will be recorded.
MISC	Specifies that commands such as DISCARD, FETCH, CHECKPOINT, VACUUM, and SET will be recorded.
READ	Specifies that SELECT and COPY will be recorded when the source is a relationship (for example, a table) or query.

Value	Description
role	Specifies that statements related to roles and permissions will be recorded, for example, GRANT, REVOKE, CREATE ROLE, ALTER ROLE, and DROP ROLE.
WRITE	Specifies that INSERT, UPDATE, DELETE, TRUNCATE, and COPY will be recorded when the destination is a relationship (table).

The following table lists other parameters related to pgAudit. You can set them on the console as needed.

Table 3-45 Parameter description

Parameter	Description
pgaudit.log	Specifies which types of statements will be logged by session audit logs.
pgaudit.log_catalog	Specifies that session logging should be enabled if all relations in a statement are in pg_catalog.
pgaudit.log_client_authentication	Controls whether to record user authentication information.
pgaudit.log_extra_field	Controls whether to record fields such as PID, IP, user name, and database.
pgaudit.log_file_rotation_age	Sets the rotation interval for separate audit logs.
pgaudit.log_parameter	Specifies that audit logging should include the parameters that were passed with the statement.
pgaudit.log_relation	Specifies whether session audit logging should create a separate log entry for each relationship (such as a table and view) referenced in a SELECT or DML statement.
pgaudit.log_rows	Sets the retrieved or affected rows that audit logs should include.
pgaudit.log_write_txid	Controls whether to record the TXID of write operations (such as INSERT and UPDATE).
pgaudit.logstatementonce	Controls whether audit logs include statements, text, and parameters.
pgaudit.log_client	Controls whether audit logs are sent to clients.
pgaudit.log_level	Sets the log level for log entries.

Parameter	Description
pgaudit.write_into_pg_log_file	Controls whether to write audit information into PostgreSQL run logs.

To display audit logs on your client, configure the following parameters:

- Set both **pgaudit.write_into_pg_log_file** and **pgaudit.log_client** to **on** and select a log level (for example, **notice**) to be displayed on the client based on the value of **pgaudit.log_level**. When you query audit logs on your client again, the logs of the corresponding level are displayed.
- If either **pgaudit.write_into_pg_log_file** or **pgaudit.log_client** is set to **off**, audit logs will not be displayed on the client.
- **pgaudit.log_level** is available only when **pgaudit.log_client** is set to **on**.

SQL Audit Verification

1. Execute SQL statements.

```
create table t1 (id int);

insert into t1 values (1);

select * from t1;
 id
----
  1

(1 rows)
```

2. On the **SQL Audits** page, download the audit log.

   The audit log contains the following information:

```
AUDIT: OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
```

   - **AUDIT** indicates an audit log entry.
   - **OBJECT** indicates an object-level audit log.
   - The first 1 indicates the object ID.
   - The second 1 indicates the sub-ID of the object.
   - **READ** indicates a read operation.
   - **SELECT** indicates a SELECT query.
   - **TABLE** indicates that the object type is table.
   - **public.t1** indicates the name and schema of the table.
   - **select * from t1** indicates the executed SQL query statement.

3.13.9 pglogical

Introduction

The pglogical extension provides logical streaming replication for PostgreSQL using a publish/subscribe model.

pglogical is a logical replication system implemented entirely as a PostgreSQL extension. It is fully integrated and does not require triggers or external programs. It provides an efficient way to selectively replicate data.

For more information, see the **official pglogical documentation**.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 12 and later versions. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'pglogical';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

For details about the extensions supported by RDS for PostgreSQL, see **Supported Extensions**.

Features

Use cases supported are:

- Upgrades between major versions (given the above restrictions)
- Full database replication
- Selective replication of sets of tables using replication sets
- Data gather/merge from multiple upstream servers

Requirements:

- The pglogical extension must be installed on both the provider and subscriber.
- Tables on the provider and subscriber must have the same names and be in the same schema.
- Tables on the provider and subscriber must have the same columns, with the same data types in each column.
- Tables must have the same primary key. It is not recommended to add additional unique constraints other than the primary key.
- To replicate multiple databases, you must set up individual provider/subscriber relationships for each. There is no way to configure replication for all databases in a PostgreSQL install at once.

Extension Installation and Uninstallation

- Installing the extension
  ```
  select control_extension('create', 'pglogical');
  ```
- Deleting the extension
  ```
  select control_extension('drop', 'pglogical');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

Basic Usage

To use the pglogical extension, you need to modify the configuration parameters.

wal_level = 'logical'

shared_preload_libraries = 'pglogical'

For details about how to modify the **shared_preload_libraries** parameter, see **Modifying the shared_preload_libraries Parameter**.

1. To configure a logical streaming replication, create a provider node.
```
SELECT pglogical.create_node(
    node_name := 'provider',
    dsn := 'host=127.0.0.1 port=5432 dbname=test user=provider_user'
);
```
2. To configure a replication set, add all tables in **public** to the **default** replication set.
```
SELECT pglogical.replication_set_add_all_tables('default', ARRAY['public']);
```

   NOTE

   Replication sets provide a mechanism to control which tables in the database will be replicated and which actions on those tables will be replicated.

   default indicates that all tables and all operations on these tables will be replicated.

   For more information about replication sets, see **official pglogical documentation**.
3. Create a subscriber node. Once the provider node is configured, the subscriber can subscribe to it.
```
SELECT pglogical.create_node(
    node_name := 'subscriber',
    dsn := 'host=127.0.0.1 port=5432 dbname=test user=subscriber_user'
);
```
4. Create a subscription on the subscriber. After the subscription is created, the synchronization and replication processes are started.
```
SELECT pglogical.create_subscription(
    subscription_name := 'subscription',
    provider_dsn := 'host=providerhost port=5432 dbname=test user=provider_user'
);
SELECT pglogical.wait_for_subscription_sync_complete('subscription');
```

Advanced Usage

1. Create a table to be replicated on the provider and subscriber, respectively.
```
create table test(id int primary key, name text, reg_time timestamp);
```

   NOTE

   The names and structures of the tables on the provider and subscriber must be the same.
2. Insert data to the table on the provider.
```
insert into test select generate_series(1,10000),'test',now();
```
3. Add the table to the replication set on the provider.
```
-- Adding all tables to the replication set
SELECT pglogical.replication_set_add_all_tables('default', ARRAY['public']);
-- Adding specified tables to the replication set
SELECT pglogical.replication_set_add_table( set_name := 'default', relation :=
'test',synchronize_data := true);
```


> **NOTE**
>
> If you add all tables to the replication set, run the following statement on the subscriber to synchronize data. Otherwise, data cannot reach the subscriber and the subscription status is unknown.
>
> ```
> select pglogical.alter_subscription_synchronize('subscription1');
> ```
>
> If you add specified tables to the replication set, the tables are automatically synchronized by default.

4. Verify that the table has been added to the replication set.
   ```
   select * from pglogical.replication_set_table ;
   ```
5. Query the subscription status on the subscriber.
   ```
   select * from pglogical.show_subscription_table('subscription1','test');
   ```
6. Check whether the table data is synchronized on the subscriber.
   ```
   select count(*) from test;
   ```

3.13.10 pg_stat_statements

Introduction

The pg_stat_statements extension provides a means for tracking planning and execution statistics of all SQL statements executed by a server.

For more information, see **pg_stat_statements** in the PostgreSQL documentation.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 10 and later versions.

You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name  = 'pg_stat_statements';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Extension Installation and Uninstallation

To check whether pg_stat_statements is installed in a database, run the following SQL statement:

```
select * from pg_extension  where extname = 'pg_stat_statements';
```

If the command output is empty, the extension is not installed. If the extension information is displayed, the extension is installed.

By default, pg_stat_statements is preloaded in the **shared_preload_libraries** parameter. Perform the following steps to install or delete the extension:

- Installing the extension
  ```
  SELECT control_extension('create', 'pg_stat_statements');
  ```


- Deleting the extension
  ```
  SELECT control_extension('drop', 'pg_stat_statements');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

Basic Functions

1. After the pg_stat_statements extension is installed, configure the parameters below. You can adjust the values based on your workload requirements.

   Table 3-46 Parameter description

Parameter	Reboot Required	Default Value	Allowed Values	Description
pg_stat_statements.max	Yes	5000	100-5,000,000	Specifies the maximum number of statements tracked by pg_stat_statements.
pg_stat_statements.save	No	on	on, off	Specifies whether to save statement statistics across server shutdowns.
pg_stat_statements.track	No	top	top, all, none	Controls which statements are counted by pg_stat_statements.
pg_stat_statements.track_planning	No	off	on, off	Controls whether planning duration is tracked by pg_stat_statements.
pg_stat_statements.track_utility	No	on	on, off	Controls whether utility commands are tracked by pg_stat_statements.

2. Query the pg_stat_statements view to obtain statistics.
   ```
   select * from pg_stat_statements;
   ```
3. Query the SQL statements with high I/O consumption.
   ```
   -- Top 5 SQL statements
   select userid::regrole, dbid, query
   from pg_stat_statements
   order by (blk_read_time+blk_write_time) desc limit 5;
   ```
4. Query the SQL statements with high consumption of shared memory.
   ```
   select userid::regrole, dbid, query
   from pg_stat_statements
   order by (shared_blks_hit+shared_blks_dirtied) desc limit 5;
   ```
5. Reset the statistics.
   ```
   select pg_stat_statements_reset();
   ```


Advanced Functions

You can use pg_stat_statements to troubleshoot high CPU usage. The process is as follows:

1. Reset the pg_stat_statements counter.
   ```
   select pg_stat_statements_reset();
   ```
   Leave enough time for pg_stat_statements to collect information.
2. Obtain the most time-consuming SQL statements.
   ```
   select * from pg_stat_statements order by total_exec_time desc limit 10;
   ```
   The obtained SQL statements have been occupying the user-mode CPU for a long time. Analyze these SQL statements.
3. Obtain the SQL statements that read the buffer for the most times.
   ```
   select * from pg_stat_statements order by shared_blks_hit + shared_blks_read desc limit 10;
   ```
   The obtained SQL statements may cause too many buffer reads due to a lack of indexes, consuming a large number of CPU resources.
4. Obtain the SQL statements that have been executed for the most times.
   ```
   select * from pg_stat_statements order by calls desc limit 10;
   ```
   It takes a short time to execute some simple SQL statements separately. However, in some cases (for example, cyclic executions in a transaction or concurrent executions), the CPU usage increases.

3.13.11 rds_hwdrs_ddl

Introduction

RDS for PostgreSQL supports the rds_hwdrs_ddl extension. This extension can be used when you do not have sufficient permissions to create objects for **PostgreSQL incremental DDL synchronization by creating triggers and functions**.

Supported Versions

This extension is available to RDS for PostgreSQL 9.5 to 15.

You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'rds_hwdrs_ddl';
```

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension('create', 'rds_hwdrs_ddl');
  ```
  After the extension is installed, you can view the objects created by the extension in the system view.
  ```
  -- Viewing the hwdrs_ddl_info table
  select relname, relowner::regrole, relacl from pg_class where relname = 'hwdrs_ddl_info';
     relname     | relowner |            relacl
  ---------------+----------+------------------------------
  ```



```
 hwdrs_ddl_info | root     | {root=arwdDxt/root,=ard/root}

-- Viewing the hwdrs_ddl_function function
select proname, proowner::regrole from pg_proc where proname = 'hwdrs_ddl_function';
      proname       | proowner
--------------------+----------
 hwdrs_ddl_function | root
(1 row)
-- Viewing the hwdrs_ddl_event trigger
select evtname, evtevent from pg_event_trigger;
     evtname     |     evtevent
-----------------+-----------------
 hwdrs_ddl_event | ddl_command_end
(1 row)
```

- Uninstalling the extension

```
SELECT control_extension('drop', 'rds_hwrds_ddl');
```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

How to Use

This extension creates all objects required for **PostgreSQL incremental DDL synchronization by creating triggers and functions** and grants corresponding permissions. You only need to install this extension to perform synchronization tasks.

To use this extension to complete a DDL synchronization task, perform the following steps:

1. **Install the extension**.
2. Create a **synchronization task from PostgreSQL to RDS for PostgreSQL**.
3. After the synchronization task is complete, **uninstall the extension** to delete the hwdrs_ddl_info table, the hwdrs_ddl_function function, and the hwdrs_ddl_event trigger.

3.13.12 rds_hwdrs_privs

Introduction

The rds_hwdrs_privs extension is used to escalate **root** privileges for some RDS for PostgreSQL versions, including:

- Granting the SELECT permission on pg_catalog.pg_authid
- Granting the BYPASSRLS and REPLICATION permissions
- Granting the permissions required to create a publication for ALL TABLES
- Granting the permissions required to execute some pg_replication_origin_xxx functions

Supported Versions

This extension is available to RDS for PostgreSQL 9.5 to 15. This extension is used to escalate **root** privileges for RDS for PostgreSQL 9.5, 9.6, 10, 11.5, and earlier versions. For versions later than 11.5, you can perform operations granted by this extension as user **root** directly.

You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'rds_hwdrs_privs';
```

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'rds_hwdrs_privs');
  ```
- Uninstalling the extension
  ```
  SELECT control_extension ('drop', 'rds_hwdrs_privs');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

How to Use

This extension can be used only by user **root** or a member user of user **root**.

You can directly perform operations as user **root** or grant **root** permissions to another user (for example, **drs_sync**) by running the following command:

```
grant root to drs_sync;
```

and use user **drs_sync** to perform operations.

- Granting a user the SELECT permission on pg_catalog.pg_authid
  ```
  select control_select_on_pg_authid('grant', 'drs_sync');
  ```
  The first parameter can be **grant** or **revoke**. The second parameter indicates a specific user, which must have been created.
- Granting the BYPASSRLS and REPLICATION permissions to a user

  select control_user_privilege('bypassrls', 'drs_sync');

  The first parameter indicates the permission to be assigned. The options are BYPASSRLS, NOBYPASSRLS, REPLICATION, and NOREPLICATION. The second parameter indicates a specific user, which must have been created.
- Granting the permissions required to create a publication for all tables
  ```
  select create_publication_for_all_tables('foo_pub', 'insert, update');
  select create_publication_for_all_tables('foo_pub');
  ```

  NOTE

  This function creates only a publication for all tables. You can use SQL statements to create a publication for a specific table.

  The first parameter is the publication name, which must be different from any existing publication name. The second parameter indicates which DML operations will be published by the new publication to the subscribers. By default, the value is the same as that of **create publication foo_pub for all tables**. In RDS for PostgreSQL 10, the allowed operations are **insert**, **update**, and **delete**. In RDS for PostgreSQL 11, **truncate** is also supported.

  The owner of the new publication is **root**. You can perform subsequent operations on the publication using SQL statements as **root** or a member user of **root**.

- Granting the permissions required to execute some **pg_replication_origin_xxx** functions

```
-- Creating a replication origin
select exec_pg_replication_origin_func('pg_replication_origin_create', 'foo_repl_origin');
-- Deleting a replication origin
select exec_pg_replication_origin_func('pg_replication_origin_create', 'foo_repl_origin');
-- Checking whether the current session is bound to a replication origin
select exec_pg_replication_origin_func('pg_replication_origin_session_is_setup');
```

  The first parameter indicates the name of the function to be executed. The options are **pg_replication_origin_create**, **pg_replication_origin_drop**, **pg_replication_origin_oid**, **pg_replication_origin_session_setup**, **pg_replication_origin_session_reset**, and **pg_replication_origin_session_is_setup**. The second parameter can be left blank. Whether the second parameter is left blank depends on whether the function to be executed requires a parameter value.

3.13.13 HypoPG

Introduction

HypoPG is an extension of RDS for PostgreSQL. It helps you understand whether a specific index can improve problematic queries. It allows you to rapidly create virtual indexes that have no resource cost (CPU, storage, or other resources).

For more information, see **HypoPG**.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 11 and later versions. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'hypopg';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Features

HypoPG is a third-party open-source extension supported by RDS for PostgreSQL. The virtual indexes created by HypoPG do not exist in any system catalog but are stored in the private memory of the connection. Because virtual indexes do not actually exist in any physical file, HypoPG ensures that the virtual indexes can be used only by a simple **EXPLAIN** statement (excluding the **ANALYZE** option). Virtual indexes are not real indexes and therefore do not consume CPU, storage, or other resources.

NOTE

HypoPG supports the following index types:

- BTREE
- BRIN
- HASH
- BLOOM (The bloom extension must be installed first.)

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'hypopg');
  ```
- Uninstalling the extension
  ```
  SELECT control_extension ('drop', 'hypopg');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

How to Use

1. Install the HypoPG extension.
   ```
   SELECT control_extension ('create', 'hypopg');
   ```
2. Create a table and insert test data.
   ```
   CREATE TABLE t (id int, col text) ;
   INSERT INTO t select x as id,'col '||x from generate_series(1,100000) as x;
   ```
3. View the default execution plan.
   ```
   EXPLAIN SELECT * FROM t WHERE id = 1;
                          QUERY PLAN
   ------------------------------------------------------------
    Seq Scan on t  (cost=0.00..1399.84 rows=344 width=36)
      Filter: (id = 1)
   (2 rows)
   ```
4. Create a virtual index.
   ```
   SELECT hypopg_create_index('CREATE INDEX ON t (id)') ;
    hypopg_create_index
   ----------------------------
   (14737,<14737>btree_t_id)
   (1 row)
   ```

   Table 3-47 Parameter description

Parameter	Description
14737	Identifier of the virtual index.
<14737>btree_t_id	Name of the virtual index.

5. Run EXPLAIN again to check that your DB instance uses the virtual index.
   ```
   EXPLAIN SELECT * FROM t WHERE id = 1;
                                  QUERY PLAN
   ------------------------------------------------------------------------------
    Index Scan using "<14737>btree_t_id" on t  (cost=0.04..2.26 rows=1 width=13)
      Index Cond: (id = 1)
   (2 rows)
   ```


6. Virtual indexes are "virtual" and are not used when SQL statements are actually run. View the actual execution plan.

```
EXPLAIN ANALYZE SELECT * FROM t WHERE id = 1;
                                                  QUERY PLAN
-------------------------------------------------------------------------------------------------------------
 Seq Scan on t  (cost=0.00..1791.00 rows=1 width=13) (actual time=0.010..5.378 rows=1
loops=1)
   Filter: (id = 1)
   Rows Removed by Filter: 99999
 Planning Time: 0.036 ms
 Execution Time: 5.401 ms
(5 rows)
```

3.13.14 pg_cron

Introduction

The pg_cron extension is a cron-based job scheduler. It uses the same syntax as regular cron, but it allows you to schedule PostgreSQL commands directly from the database. For more information, see the **official pg_cron documentation**.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 12 and later versions. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'pg_cron';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Features

The standard cron syntax is used, in which the asterisk (*) means "run every time period", and a specific number means "but only at this time".

```
┌──────────────── min (0 - 59)
│ ┌────────────── hour (0 - 23)
│ │ ┌──────────── day of month (1 - 31)
│ │ │ ┌────────── month (1 - 12)
│ │ │ │ ┌──────── day of week (0 - 6) (0 to 6 are Sunday to
│ │ │ │ │         Saturday, or use names; 7 is also Sunday)
│ │ │ │ │
```

For example, the syntax for 9:30 a.m. (GMT) every Saturday is as follows:

```
30 9 * * 6
```

Precautions

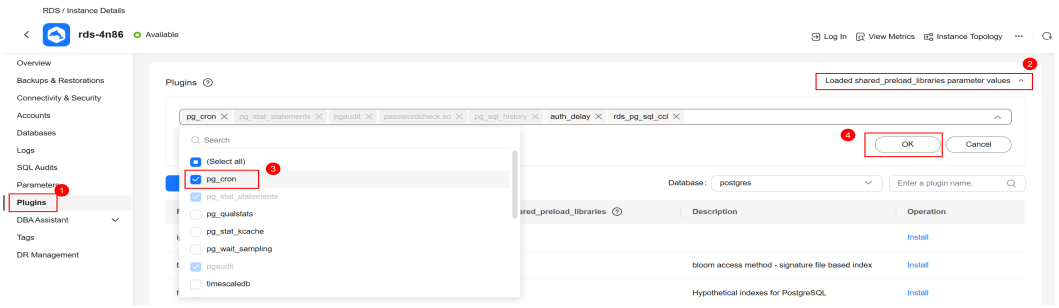
- pg_cron requires a daemon process. Therefore, before starting a database, you need to add pg_cron to the **shared_preload_libraries** parameter value.
- Scheduled jobs do not run on the standby instance. However, if the standby instance is promoted to primary, scheduled jobs automatically start.

- Scheduled jobs are executed with the permissions of the job creator.
- Scheduled jobs are executed using the GMT time.
- An instance can run multiple jobs concurrently, but one job can only run once at a given time.
- If a job needs to wait for the completion of the previous scheduled job, it will enter the wait queue and will be started as soon as possible after the previous job completes.
- Before using this extension, you need to change the value of **cron.database_name** to the name of the database where scheduled jobs are created. This parameter can only be set to one database name instead of multiple names.

Installing the Extension

1. In the instance list, click the target instance name to go to the **Overview** page.
2. Choose **Plugins** and add pg_cron to the **shared_preload_libraries** parameter values.

   Figure 3-118 Plugins

3. Reboot the DB instance for the modification to be applied.

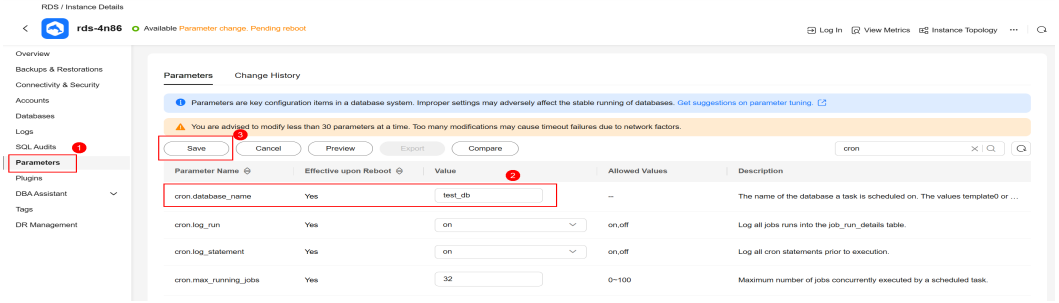
   Figure 3-119 Extension added

4. If the extension is not to be installed in the default database postgres, modify the **cron.database_name** parameter.

   The value of **cron.database_name** must be changed to the name of the database where pg_cron is to be used. For example, if you want to install the extension in the test_db database, change the value of **cron.database_name** to **test_db**.

Figure 3-120 Changing the parameter value



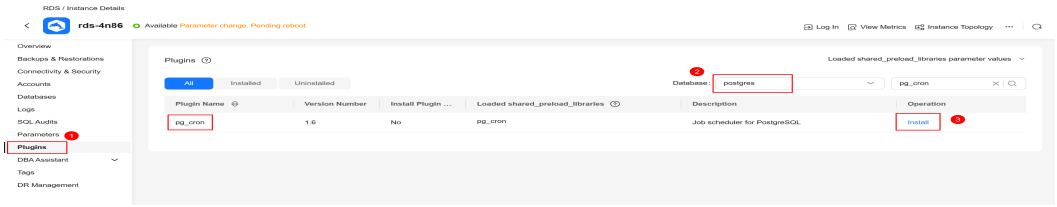
NOTE

After modifying **cron.database_name**, reboot the instance for the modification to be applied.

5. Install the pg_cron extension.
   - Installing pg_cron on the console

     On the **Plugins** page, select the desired database, search for pg_cron, and click **Install**.

     Figure 3-121 Installing pg_cron

   - Installing pg_cron using SQL statements

     Log in to the desired database and run the following SQL statement to create the extension:

     ```
     CREATE EXTENSION IF NOT EXISTS pg_cron;
     ```

Basic Usage

Create jobs.

```
-- Job 1: Delete old data at 03:30 a.m. (GMT) every Saturday.
SELECT cron.schedule('30 3 * * 6', $$DELETE FROM events WHERE event_time < now() - interval '1 week'$$);
-- Job 2: Run the VACUUM command at 10:00 a.m. (GMT) every day. The job is named nightly-vacuum.
SELECT cron.schedule('nightly-vacuum', '0 10 * * *', 'VACUUM');
```

Advanced Usage (Configuring Scheduled Jobs for Databases Other Than postgres)

Prerequisites: The pg_cron extension has been installed in the postgres database, and the test_db database has been created. Configure scheduled jobs for the test_db database.

1. Log in to the postgres database.
2. Create scheduled jobs.


```
SELECT cron.schedule('create', '10 * * * *', 'create table test (a int);');
SELECT cron.schedule('insert', '15 * * * *', 'insert into test values(1);');
SELECT cron.schedule('drop', '20 * * * *', 'drop table test;');
```

NOTE

Each scheduled job name must be unique. Otherwise, the job will be overwritten.

3. Update the scheduled jobs to run them in the test_db database.

```
UPDATE cron.job SET database = 'test_db' WHERE jobid = 1;
UPDATE cron.job SET database = 'test_db' WHERE jobid = 2;
UPDATE cron.job SET database = 'test_db' WHERE jobid = 3;
```

To query the job IDs, run the following commands:

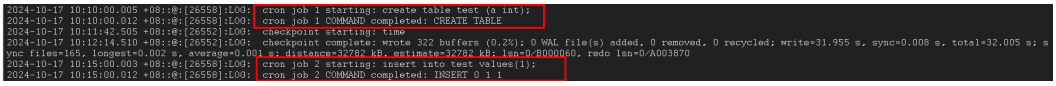
```
postgres=> select * from cron.job;
 jobid |  schedule  |           command           | nodename  | nodeport | database  | username | active |
jobname
-------+------------+-----------------------------+-----------+----------+-----------+----------+--------+---------
     1 | 10 * * * * | create table test (a int);  | localhost |     5432 | postgres  | root     | t      | create
     2 | 15 * * * * | insert into test values(1); | localhost |     5432 | postgres  | root     | t      | insert
     3 | 20 * * * * | drop table test;            | localhost |     5432 | postgres  | root     | t      | drop
```

4. In the command output, the value of **database** is **test_db**, indicating that the jobs are executed in the test_db database.

```
postgres=> select * from cron.job;
 jobid |  schedule  |           command           | nodename  | nodeport | database  | username | active |
jobname
-------+------------+-----------------------------+-----------+----------+-----------+----------+--------+---------
     1 | 10 * * * * | create table test (a int);  | localhost |     5432 | test_db   | root     | t      | create
     2 | 15 * * * * | insert into test values(1); | localhost |     5432 | test_db   | root     | t      | insert
     3 | 20 * * * * | drop table test;            | localhost |     5432 | test_db   | root     | t      | drop
```

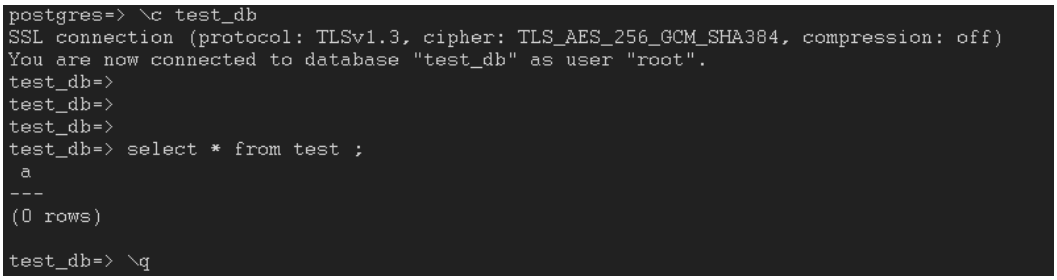
5. Check whether the scheduled jobs have been successfully executed.
   - Check the logs for successful job execution.

     Figure 3-122 Checking logs

   - Check whether the **test** table has been created in the test_db database.

     Figure 3-123 Querying the database

6. If you do not want to use the scheduled jobs anymore, delete them based on the result obtained in **4**:
   - Delete a job using the job ID.

     ```
     SELECT cron.unschedule(1);
     ```

   - Delete a job using the job name.

     ```
     SELECT cron.unschedule('create');
     ```


Parameters Related to the pg_cron Extension

Table 3-48 Parameter description

Parameter	Description	Default Value	Reboot Required
cron.database_name	The database in which scheduled job metadata is kept.	postgres	Yes
cron.log_statement	Whether to log all cron statements before running them.	true	Yes
cron.log_run	Whether to log every job that runs in the job_run_details table.	true	Yes
cron.host	The name of the host where scheduled jobs are to be executed.	localhost	Yes
cron.use_background_workers	Whether to use background work processes instead of client sessions to run jobs.	false	Yes
cron.max_running_jobs	The number of jobs that can run concurrently.	5	Yes

3.13.15 dblink

Introduction

dblink is an extension module that supports connections to RDS for PostgreSQL databases. dblink is mainly used for distributed query. It allows you to query data across databases. It considers different databases as a whole, so that you can query data or perform other operations across databases.

It can also be used for data backup and restoration, and data synchronization. It greatly improves the flexibility and scalability of your databases, helping you use databases more efficiently and reliably. For more information, see the **official dblink documentation**.

Supported Versions

This extension is available to RDS for PostgreSQL 9.5 to 15.

You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'dblink';
```

If this extension is not supported, **upgrade the minor version of your DB instance**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Extension Installation and Uninstallation

- Installing the extension
  ```
  SELECT control_extension ('create', 'dblink');
  ```
- Uninstalling the extension
  ```
  SELECT control_extension ('drop', 'dblink');
  ```

For more information, see **Installing and Uninstalling an Extension on the RDS Console** and **Installing and Uninstalling an Extension Using SQL Commands**.

NOTE

When dblink is used to perform cross-database operations, the server IP addresses of the two DB instances involved must be in the same VPC.

How to Use

Scenario 1:

If certain operations need to be performed across databases in an instance, you can use dblink.

```
-- 1. Create databases.
create database homedb;
create database db1;

-- 2. Switch to homedb and create a dblink connection.
select dblink_connect('connect_db1', 'dbname=db1 port=5432 user=root password=******** host=127.0.0.1');
dblink_connect
----------------
OK
(1 row)

-- 3. Run SQL statements.
-- Query data.
SELECT * FROM dblink('connect_db1', 'select * from test') as test(id integer, info varchar(8));
id | info
----+------
1 | a
2 | b
(2 rows)

-- Insert data.
SELECT dblink_exec('connect_db1', 'insert into test values(3,'c')');
dblink_exec
-------------
INSERT 0 1
(1 row)

-- 3. Switch to db1 and view the result.
select * from test;
id | info
----+------
1 | a
2 | b
3 | c
(3 rows)

-- 4. Close the remote connection.
SELECT dblink_disconnect('connect_db1');
dblink_disconnect
----------------
```



```
OK
(1 row)
```

Scenario 2:

If there are two RDS for PostgreSQL DB instances in the same VPC, one production instance (db1) and one test instance (db2), and data in the test instance needs to be synchronized to the production instance, you can use dblink to synchronize the data.

```
-- 1. Log in to the production instance and create a database.
create database db1;

-- 2. Switch to the production database db1 and connect to the test database db2.
select dblink_connect('connect_db2', 'dbname=db2 port=5432 user=root password=********
host=10.29.182.247');
dblink_connect
----------------
OK
(1 row)

-- 3. Query data of table test1 in test database db2.
SELECT * FROM dblink('host=10.29.182.247 port=5432 user=root password=******** dbname=db2', 'select *
from test1') as test1(id int, name text);
id | name
----+------
1 | a
2 | b
(2 rows)

-- 4. Synchronize data of table test1 in test database db2 to table backup1 in production database db1.
insert into backup1 SELECT * FROM dblink('dbname=db2 port=5432 user=root password=********
host=10.29.182.247', 'select * from test1') as backup1(id int, name text);
dblink
-------------
INSERT 0 2
(2 row)

--5. Query data of table backup1 in production database db1.
select * from backup1;
id | name
----+------
1 | a
2 | b
(2 rows)

--6. Close the connection.
SELECT dblink_disconnect('connect_db2');
dblink_disconnect
----------------
OK
(1 row)
```

3.13.16 rds_pg_sql_ccl

Introduction

To prevent highly concurrent SQL statements that consume too many resources from causing instance instability, RDS for PostgreSQL provides rds_pg_sql_ccl, a Huawei-developed extension. ccl is short for concurrent control. SQL statement concurrency control can ensure instance stability, optimize performance, and guarantee resources for critical tasks in the following scenarios:

- When workloads increase sharply, the instance stability is ensured by limiting the execution of a certain type of SQL statements.

- When there are not enough resources, the success of core tasks is ensured by limiting the execution of other SQL statements to reduce resource consumption.

This extension provides two concurrency control methods:

- Method 1: It limits the number of SQL statements that can be executed at the same time. This number is specified by the **rds_pg_sql_ccl.max_concurrent_sql** parameter. The default value is **-1**, indicating that the number is not limited.
- Method 2: It limits the number of a certain type of SQL statements (with the same query ID) that can be executed at the same time. This number is specified by concurrency control rules. For details about concurrency control rules, see below.

Supported Versions

This extension is available to the latest minor versions of RDS for PostgreSQL 11.20, 12.15, 13.11, 14.8, 15.4, 16.2, and above. You can run the following SQL statement to check whether your DB instance supports this extension:

```
SELECT * FROM pg_available_extension_versions WHERE name = 'rds_pg_sql_ccl';
```

If this extension is not supported, **upgrade the minor version of your DB instance** or **upgrade the major version using dump and restore**.

To see more extensions supported by RDS for PostgreSQL, go to **Supported Extensions**.

Notes

SQL statement concurrency control rules must be configured based on workloads and resource usage.

Creating Rules

1. In any given database, each rule must have a unique query ID, but rules in different databases can have the same query ID.
2. A rule does not take effect immediately after being created. You need to call the **enable_ccl_rule** function to make it applied.
3. The **get_query_id** function cannot obtain the query IDs of SQL statements using bind variables, and the **add_ccl_rule_by_query** function cannot limit the execution of such SQL statements.
4. You can obtain the query ID of a SQL statement with bind variables using the pg_stat_statements extension. Then, you can use **add_ccl_rule_by_queryid** to create rules. For details, see "Concurrency Control for SQL Statements with Bind Variables".

How Rules Take Effect

1. Method 1 limits the number of SQL statements that can be concurrently executed. Rules creating using this method take effect preferentially. On the basis of method 1, you can use method 2 to further limit concurrent execution of a specific type of SQL statements.

2. After an instance reboot, none of the rules are applied.
3. Read replicas synchronize the rules of the primary instance and call the **enable_ccl_rule** function to make the rules apply.
4. A rule is only applied to SQL statements executed after the rule was enabled.

Concurrency Control for SQL Statements Without Bind Variables

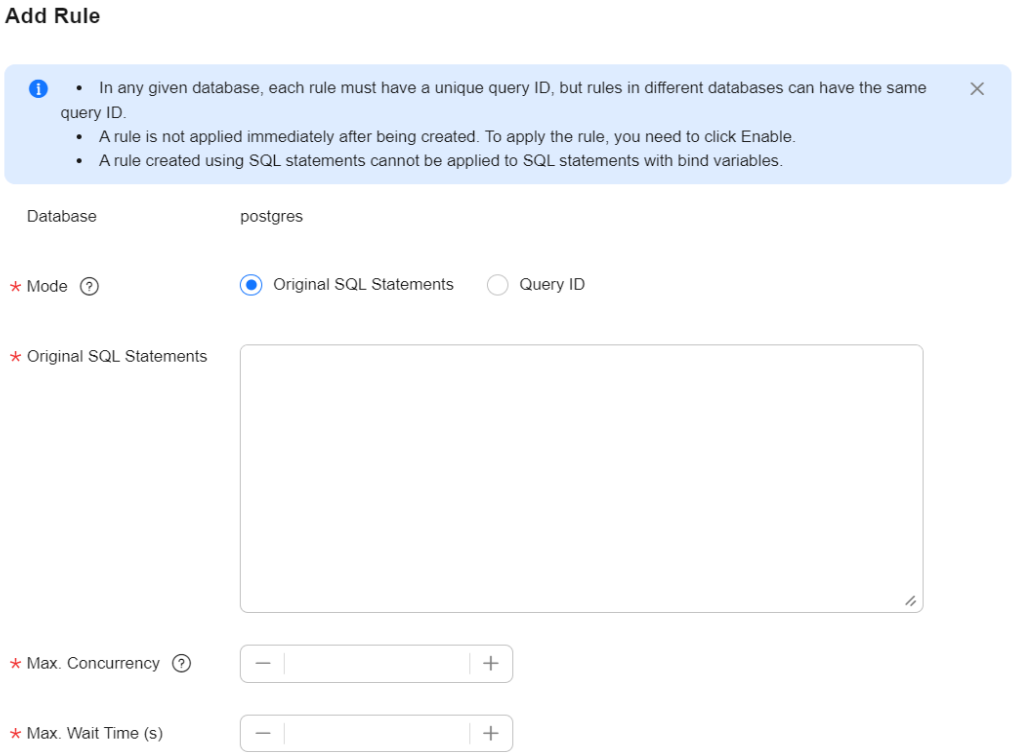
Prerequisites:

1. Install the kernel extension rds_pg_sql_ccl on the console or by running SQL statements.
   ```
   SELECT control_extension ('create', 'rds_pg_sql_ccl');
   ```
2. Set kernel parameters.
   ```
   rds_pg_sql_ccl.enable_ccl = on
   ```

Then, perform the following operations:

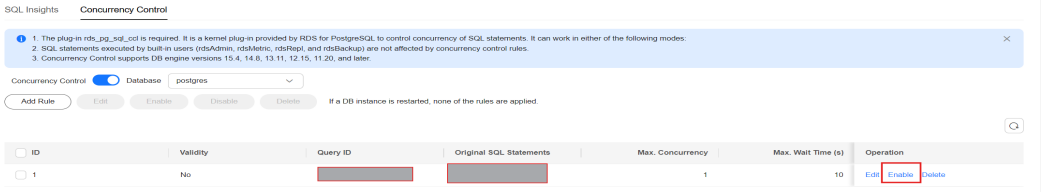
1. On the **Instances** page, click the instance name to go to the **Overview** page.
2. In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.
3. Choose **SQL Explorer** > **Concurrency Control**.
4. Toggle on the concurrency control switch.
5. Click **Add Rule** and configure required parameters.

Figure 3-124 Adding a concurrency control rule



   - Locate the rule and click **Enable** in the **Operation** column to enable the rule.

Figure 3-125 Enabling a rule



- To disable a rule, click **Disable**.

Figure 3-126 Disabling a rule

- To delete a rule, click **Delete**. After a concurrency control rule that is enabled is deleted, the rule is not applied anymore.

Figure 3-127 Deleting a rule

Concurrency Control for SQL Statements with Bind Variables

Drivers such as JDBC support prepared statements. They precompile parameterized SQL statements and execute them after parameters are entered. In the pg_stat_statements view, the statements are displayed as bind variables. For SQL statements using bind variables, the query IDs calculated by the kernel are different from those of the same statements using actual parameter values. The concurrent execution of such statements cannot be limited by adding the statements.

Such SQL statements can only be limited by executing them first and then adding concurrency control rules.

1. Execute a SQL statement with bind variables. In this way, the kernel calculates its query ID. An example of the JDBC-based prepared statement program is as follows:
   ```
   String sql = "select pg_sleep(?);";
   PreparedStatement preparedStatement = conn.prepareStatement(sql);
   preparedStatement.setInt(1, 500);
   ResultSet resultSet = preparedStatement.executeQuery();
   ```
2. Query the query ID of the SQL statement in the pg_stat_statements view.
   ```
   select queryid from pg_stat_statements where query like '%select pg_sleep%';
   ```
3. Add a concurrency control rule based on the query ID.
   ```
   select rds_pg_sql_ccl.add_ccl_rule_by_queryid($queryid);
   ```
4. Enable the rule based on the return value (**rule_id**) of the previous SQL statement.
   ```
   select rds_pg_sql_ccl.enable_ccl_rule($rule_id);
   ```
5. Obtain all rules applied from the get_all_enabled_rule view provided by the extension.
   ```
   select * from rds_pg_sql_ccl.get_all_enabled_rule;
   ```


Parameters

Table 3-49 Parameter description

Parameter	Data Type	Default Value	Maximum Value	Minimum Value	Description
rds_pg_sql_ccl.enable_ccl	bool	false	-	-	Whether to enable a concurrency control rule.
rds_pg_sql_ccl.max_enabled_rules	int	5000	500000	0	The number of rules that take effect at the same time.
rds_pg_sql_ccl.max_concurrent_sql	int	-1	50000	-1	The number of SQL statements that can be concurrently executed (its priority is higher than that of concurrency control rules). If the value is less than 0, the number of SQL statements is not limited.

Function Interface Description

Table 3-50 Function interface description

No.	Function	Parameter	Return Value	Description
1	rds_pg_sql_ccl.get_query_id	query_string text, search_path text default 'public'	queryid	Calculates the query ID of a SQL statement.
2	rds_pg_sql_ccl.add_ccl_rule_by_query	query_string text, max_concurrency int default 0, max_waiting int default 0, search_path text default 'public'	ruleid	Adds a concurrency control rule using SQL statements.
3	rds_pg_sql_ccl.add_ccl_rule_by_queryid	query_id bigint, max_concurrency int default 0, max_waiting int default 0, search_path text default 'public'	ruleid	Adds a concurrency control rule based on the query ID.
5	rds_pg_sql_ccl.enable_ccl_rule	rule_id bigint	bool	Enables a concurrency control rule based on the rule ID.
6	rds_pg_sql_ccl.disable_ccl_rule	rule_id bigint	bool	Disables a concurrency control rule based on the rule ID.
7	rds_pg_sql_ccl.disable_all_ccl_rule	-	void	Disables all rules.
8	rds_pg_sql_ccl.delete_ccl_rule	rule_id bigint	void	Deletes a concurrency control rule based on the rule ID.

No.	Function	Parameter	Return Value	Description
9	rds_pg_sql_ccl.update_ccl_rule	new_rule_id bigint, new_max_concurrency int, new_max_waiting int	void	Updates a concurrency control rule based on the rule ID.

Description of some parameters:

- **max_concurrency**: The maximum number of SQL statements that can be concurrently executed.
- **max_wait**: The maximum waiting time after which new SQL statements of a specified type will fail to be executed when the maximum number of concurrent SQL statements is reached.
- **new_max_concurrency**: The new maximum number of concurrent SQL statements.
- **new_max_wait**: The new maximum waiting time.

View Interface Description

Table 3-51 View interface description

No.	View	Column	Description
1	rds_pg_sql_ccl.get_all_enabled_rule	dbid oid, queryid bigint, max_concurrency int, max_wait int	Displays all concurrency control rules applied.
2	rds_pg_sql_ccl.get_activity_query_status	queryid bigint, wait_start_time timestamptz, pid int, dbid oid	Displays the status of each SQL statement in the current instance, such as the query ID and whether the SQL statement is waiting.

No.	View	Column	Description
3	rds_pg_sql_ccl.get_current_db_ccl_rule	rule_id bigint, query_id bigint, query_string, max_concurrency int, max_waiting int, search_path text, create_time timestamptz, enabled bool	Displays the concurrency control rules created for the current database (whether applied or not).

3.14 Problem Diagnosis and SQL Analysis

3.14.1 Function Overview

DBA Assistant provides visualized database O&M and intelligent diagnosis for developers and database administrators (DBAs), making database O&M easy and efficient. By analyzing alarms, resources, and performance metrics, it helps users quickly locate faults and keep track of instance status.

NOTE

To use DBA Assistant on the RDS console, IAM users must have the **RDS FullAccess**, **DAS FullAccess**, **DAS Administrator**, and **CES FullAccess** permissions. For details, see **Creating a User and Granting Permissions**.

Scenarios

- In emergency cases, you can manually terminate slow sessions to recover your instance, improving database availability.
- If your DB instance is unstable due to a large number of concurrent SQL requests from new services, you can set concurrency control rules for SQL statements to limit concurrent SQL statements and ensure instance stability.
- If your instance storage is full, you can learn about the storage usage and disk space distribution on the **Storage Analysis** page. You can enable storage autoscaling. When the available storage of your instance drops to the threshold, autoscaling is triggered. For details, see **Configuring Storage Autoscaling**.

Supported Regions

DBA Assistant is available in the following regions: CN North-Beijing4, CN East-Shanghai1, CN East2, CN South-Guangzhou, AF-Johannesburg, CN Southwest-Guiyang1, CN-Hong Kong, AP-Bangkok, AP-Singapore, ME-Riyadh, TR-Istanbul, and LA-Sao Paulo1.

To use DBA Assistant in any other regions, **submit a service ticket** to apply for required permissions.

Functions

Table 3-52 lists the functions supported by DBA Assistant.

Table 3-52 Function description

Function	Description	Reference
Overview	Shows the status of your instance, including alarms, resource usages, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.	**Viewing the Overall Status of a DB Instance**
Performance	Displays key metrics of your instance and provides metric comparison between different days. You can keep track of metric changes and detect exceptions in a timely manner.	**Viewing Performance Metrics of a DB Instance**
Killing Sessions	If the maximum number of connections for an instance has been reached and the instance cannot be logged in to, you can view and kill unnecessary sessions through the emergency channel.	**Killing Sessions**
Real-Time Sessions	You can query session snapshots of your instance while sorting, filtering, and displaying the snapshots as needed. You can also view session statistics by user, source, and database. Sessions can be killed for urgent instance recovery to ensure database availability.	**Managing Real-Time Sessions**
Slow Query Log	Displays slow queries within a specified time period. You can view top 5 slow query logs by user or client IP address, sort statistics, and identify sources of slow SQL statements.	**Viewing Slow Query Logs of a DB Instance**
SQL Insights	After **Collect All SQL Statements** is enabled, you can gain a comprehensive insight into SQL statements on the **SQL Explorer** page. Top SQL helps you locate exceptions.	**Creating a SQL Insights Task**
Concurrency Control	Concurrency Control restricts the execution of SQL statements based on specified rules when there are SQL statements that cannot be optimized timely or a resource (for example, vCPU) bottleneck occurs.	**Creating a Concurrency Control Rule**

3.14.2 Performance Monitoring

3.14.2.1 Viewing the Overall Status of a DB Instance

On the **Overview** page, you can get knowledge of the overall status of your RDS for PostgreSQL instance, including alarms, intelligent anomaly diagnosis, and key performance metrics. DBA Assistant diagnoses instance health using operational data analytics and intelligent algorithms, and provides you with solutions and suggestions for handling detected exceptions.

Functions

Table 3-53 lists the functions provided on the **Overview** page.

Table 3-53 Function description

Function	Description
Alarms	To view alarm details, click the number next to an alarm severity.
Intelligent Anomaly Diagnosis	Shows the health status of your instance based on operational data analytics and intelligent algorithms.
Performance Monitoring	Shows key performance metrics of the instance, including the CPU usage, memory usage, number of SQL statements executed for more than 3s, and connections.

Alarms

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

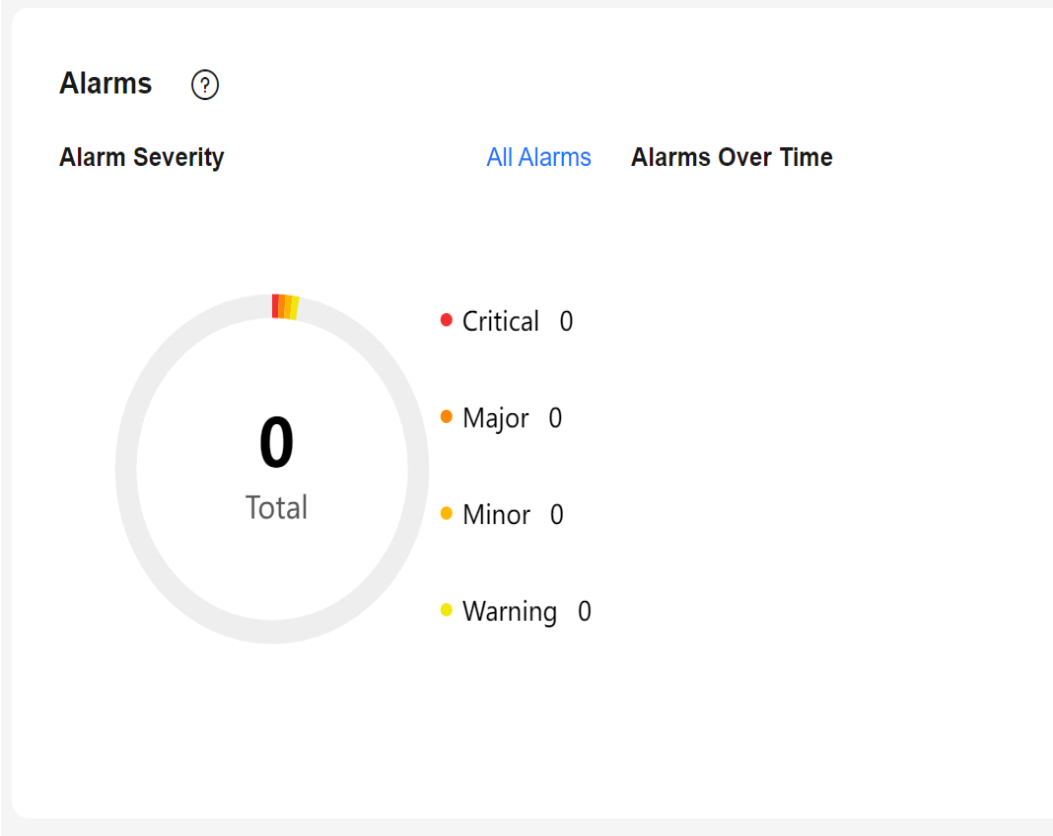
Step 4 On the **Instances** page, click the DB instance name.

Step 5 On the **Overview** page, view the status of your instance.

- In the **Alarms** area, view alarm information of your instance.

  To view the list of all alarms, click **All Alarms**. To view alarm details, click the number next to an alarm severity.

Figure 3-128 Alarms

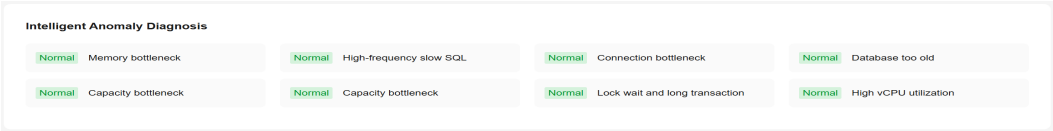


- In the **Intelligent Anomaly Diagnosis** area, view the health diagnosis results of your instance.

  NOTE

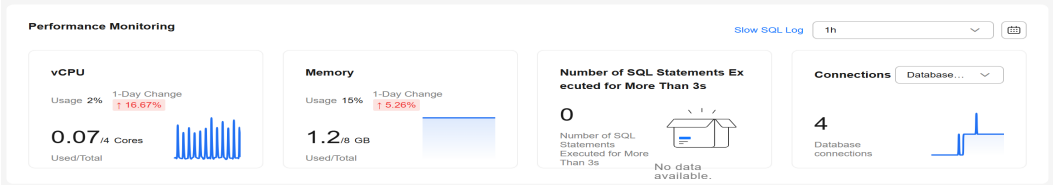
  Intelligent Anomaly Diagnosis provides diagnosis results for the check items in the past 5 minutes. If any diagnosis result is abnormal, the check item is abnormal in the past 5 minutes.

Figure 3-129 Intelligent Anomaly Diagnosis



- In the **Performance Monitoring** area, view key performance metrics of your instance.

Figure 3-130 Performance Monitoring



----End

3.14.2.2 Viewing Performance Metrics of a DB Instance

Scenarios

DBA Assistant allows you to view the performance metrics of your DB instance. Historical trends of performance metrics within a specified time period help you learn about the status and resource usage of your DB instance. If any alarm is reported, you can take actions timely.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

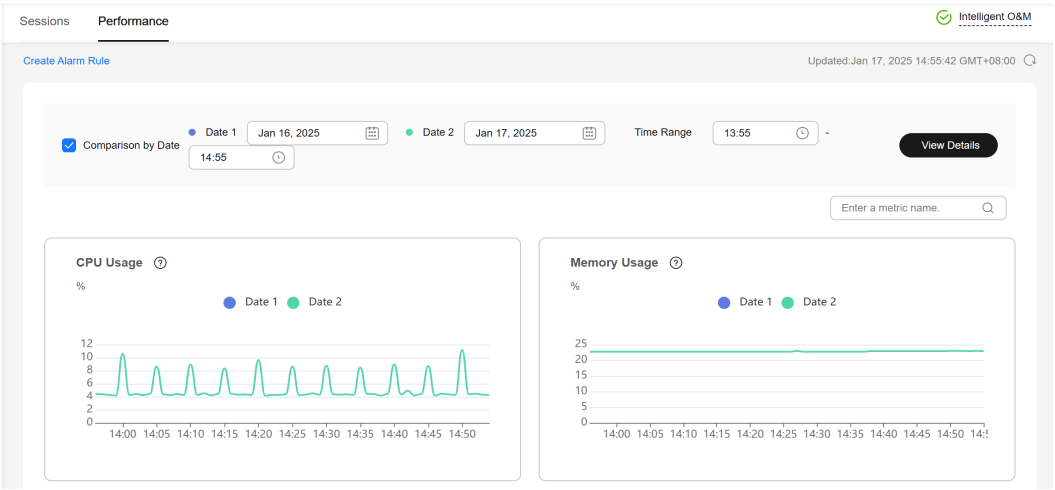
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Performance** tab. You can perform the following operations on this tab page:

- Click **View Details** to view metric changes of your instance in the same period on different days.

  To view more metrics, see **Viewing Monitoring Metrics**.

  Figure 3-131 Performance

- Click **Create Alarm Rule** to go to the Cloud Eye console and customize alarm rules and notification policies. By this way, you can learn about your instance status in a timely manner.

  For details, see **Setting Alarm Rules**.

----End

3.14.3 Problem Diagnosis

3.14.3.1 Killing Sessions

Scenarios

You can kill sessions when necessary on the **Sessions that Can Be Killed If Necessary** tab page. This page covers:

- **Emergency Channel**: If the maximum number of connections for an instance has been reached and the instance cannot be logged in to, you can view and kill unnecessary sessions through this channel.
- **History Logs**: You can view history logs to learn details of the kill operations that you performed using the emergency channel.

Precautions

Killing a session may cause the application to disconnect from the instance. Your application should be able to reconnect to the instance.

Constraints

- Do not kill sessions unless you really need to. All your kill operations will be logged.
- Sessions of sensitive users such rdsAdmin, rdsBackup, rdsMetric, and rdsRepl, and sessions whose username is null cannot be killed.
- You may fail to refresh the session list if your instance has a heavy load. Minimize the resources occupied by the emergency channel. Wait a few seconds and try again.
- If the CPU usage reaches 100%, requests to kill sessions may fail. You may have to try more than once.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

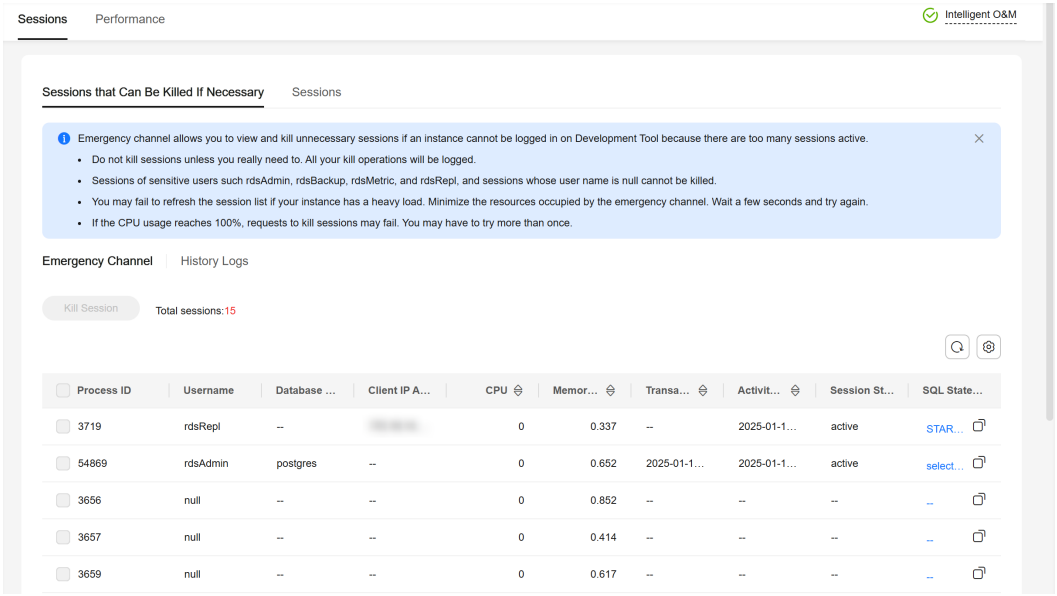
Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Choose **Sessions** > **Sessions that Can Be Killed If Necessary**.

- To kill a session, click the **Emergency Channel** tab, select the session, and click **Kill Session**.

Figure 3-132 Emergency Channel



- To check the kill history, click the **History Logs** tab.

----End

3.14.3.2 Managing Real-Time Sessions

Scenarios

You can query session snapshots of your instance while sorting, filtering, and displaying the snapshots as needed. You can also view session statistics by user, source, and database. Sessions can be killed for urgent instance recovery to ensure database availability.

Precautions

Killing a session may cause the application to disconnect from the instance. Your application should be able to reconnect to the instance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

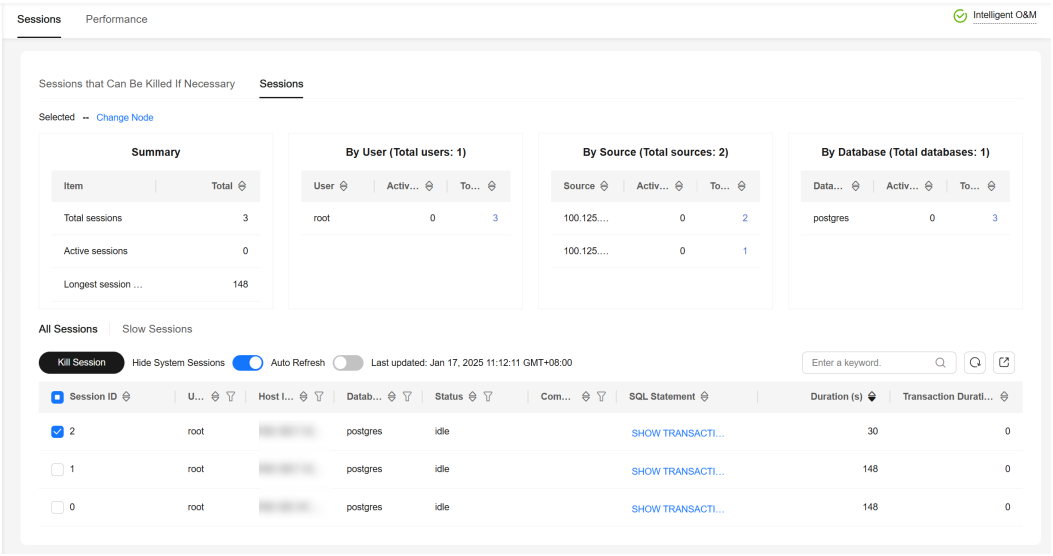
Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Sessions** tab. You can perform the following operations on this tab page:

- Viewing session statistics

In the session statistics area, you can view the session summary and session statistics by user, source, and database.

Figure 3-133 Sessions



- Killing abnormal sessions

  In the session list, select the abnormal session you want to end and click **Kill Session** to recover the database.

- Exporting the session list

  To export the session list, click above it.

----End

3.14.4 SQL Analysis

3.14.4.1 Viewing Slow Query Logs of a DB Instance

Scenarios

Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, client IP address, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

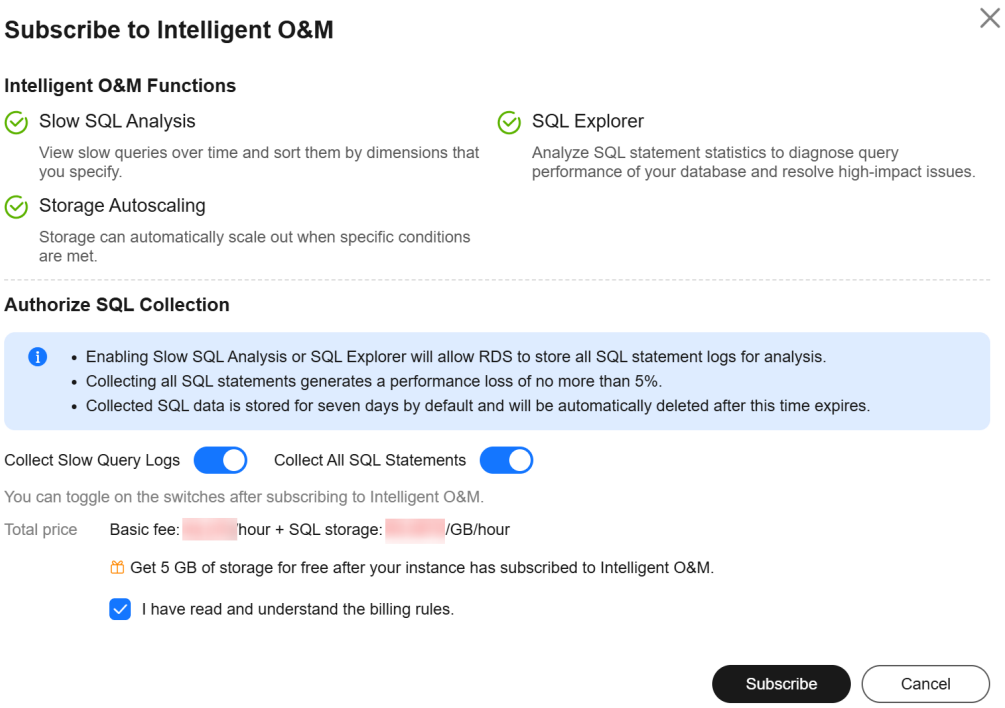
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Slow Query Log** tab.

NOTE

Slow SQL Analysis needs to be purchased separately. To use this function, subscribe to Intelligent O&M first.

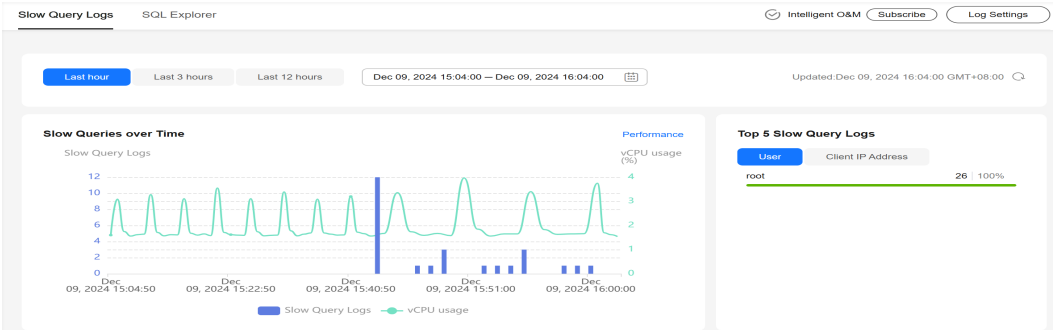
Step 7 Click **Subscribe**. In the displayed dialog box, you can learn about Intelligent O&M functions and pricing.



Step 8 After subscribing to Intelligent O&M, view slow queries over time of your instance.

Step 9 You can view slow queries over time and you can see the slow log archive history for the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day).

Figure 3-134 Viewing slow queries over time



Step 10 View slow query log details and template statistics.

- To export slow query log information, click **Export**.
- To view log export history, click **View Export List**.

----End

3.14.4.2 Creating a SQL Insights Task

Scenarios

SQL Insights allows you to not only query all executed SQL statements, but also analyze and search for the tables that are accessed and updated most frequently, and the SQL statements that have the longest lock wait, helping you quickly identify exceptions.

Constraints

- You need to enable **Collect All SQL Statements** before using SQL Insights.
- After **Collect All SQL Statements** is disabled, new SQL statements will not be collected anymore and the collected SQL data will be deleted.
- Only the tasks created in the last two days can be displayed.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click **SQL Explorer** and then **SQL Insights**.

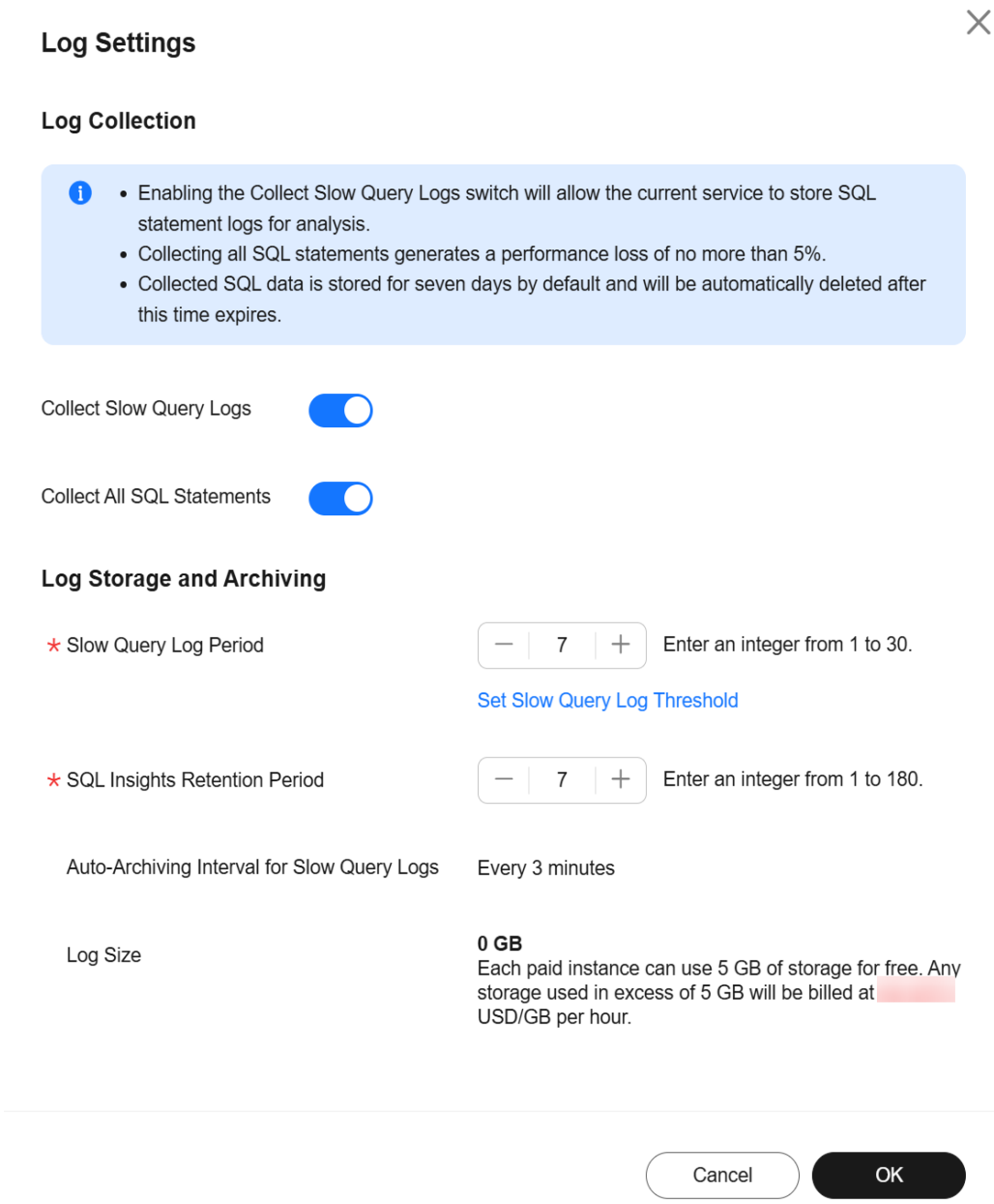
Step 7 Click next to **Collect All SQL Statements**.

To disable this function, click **Log Settings** in the upper right corner, toggle off the **Collect All SQL Statements** switch, and click **OK**.

NOTE

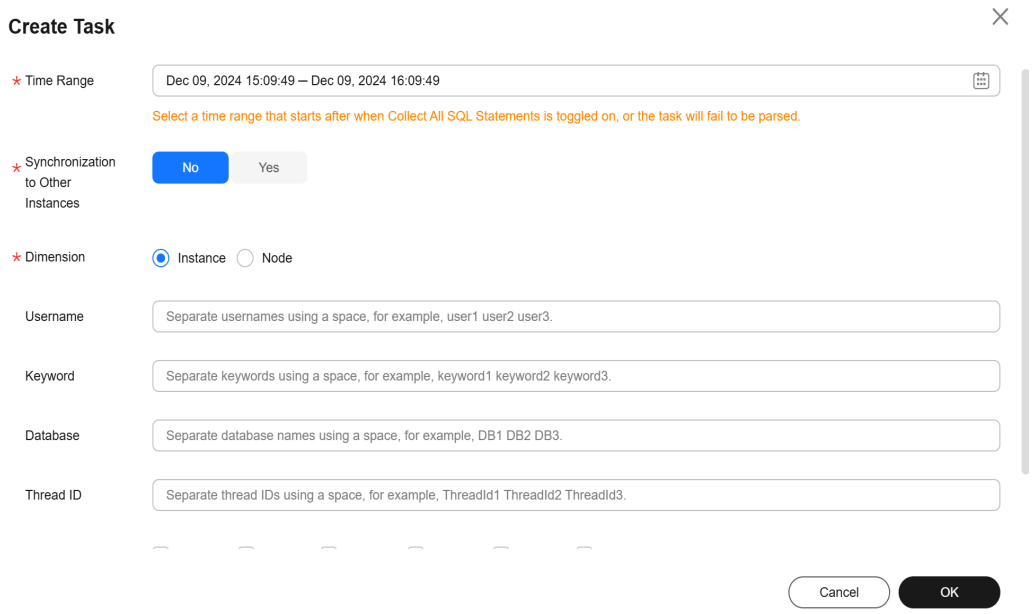
Collecting all SQL statements generates a performance loss of no more than 5%.

Figure 3-135 Log settings



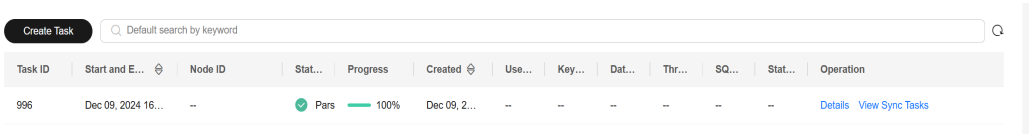
Step 8 Click **Create Task**. In the displayed dialog box, specify **Time Range**, **Dimension**, and other configuration items, and click **OK**.

Figure 3-136 Creating a SQL Insights Task



Step 9 In the task list, click **Details** in the **Operation** column to view task details.

Figure 3-137 Viewing task details



----End

3.14.4.3 Creating a Concurrency Control Rule

Scenarios

You can create rules to control concurrent execution of SQL statements by specifying original SQL statements or query ID. When the number of matched SQL statements exceeds the maximum number of concurrent SQL statements allowed, the DB instance rejects the SQL statements to ensure stability. SQL statement concurrency control of RDS for PostgreSQL is implemented using the Huawei-developed extension rds_pg_sql_ccl.

It can be used in the following scenarios:

- When workloads increase sharply, the instance stability is ensured by limiting the execution of a certain type of SQL statements.
- When there are not enough resources, the success of core tasks is ensured by limiting the execution of other SQL statements to reduce resource consumption.

Supported Versions

SQL statement concurrency control is available in the following versions:

- RDS for PostgreSQL 15: 15.4 and later
- RDS for PostgreSQL 14: 14.8 and later
- RDS for PostgreSQL 13: 13.11 and later
- RDS for PostgreSQL 12: 12.15 and later
- RDS for PostgreSQL 11: 11.20 and later

Constraints

- SQL statements executed by built-in users (rdsAdmin, rdsMetric, rdsRepl, and rdsBackup) are not affected by concurrency control rules.
- To use SQL statement concurrency control, the rds_pg_sql_ccl extension must be installed. For details, see **Installing and Uninstalling an Extension on the RDS Console**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Choose **SQL Explorer** > **Concurrency Control**.

Step 7 Toggle on the concurrency control switch .

NOTE

Concurrency control rules take effect only after concurrency control is enabled.

Step 8 Click **Add Rule**. Configure the parameters listed in **Table 3-54**.

NOTE

Configure SQL statement concurrency control rules based on workloads and resource usage.

Figure 3-138 Creating a Concurrency Control Rule

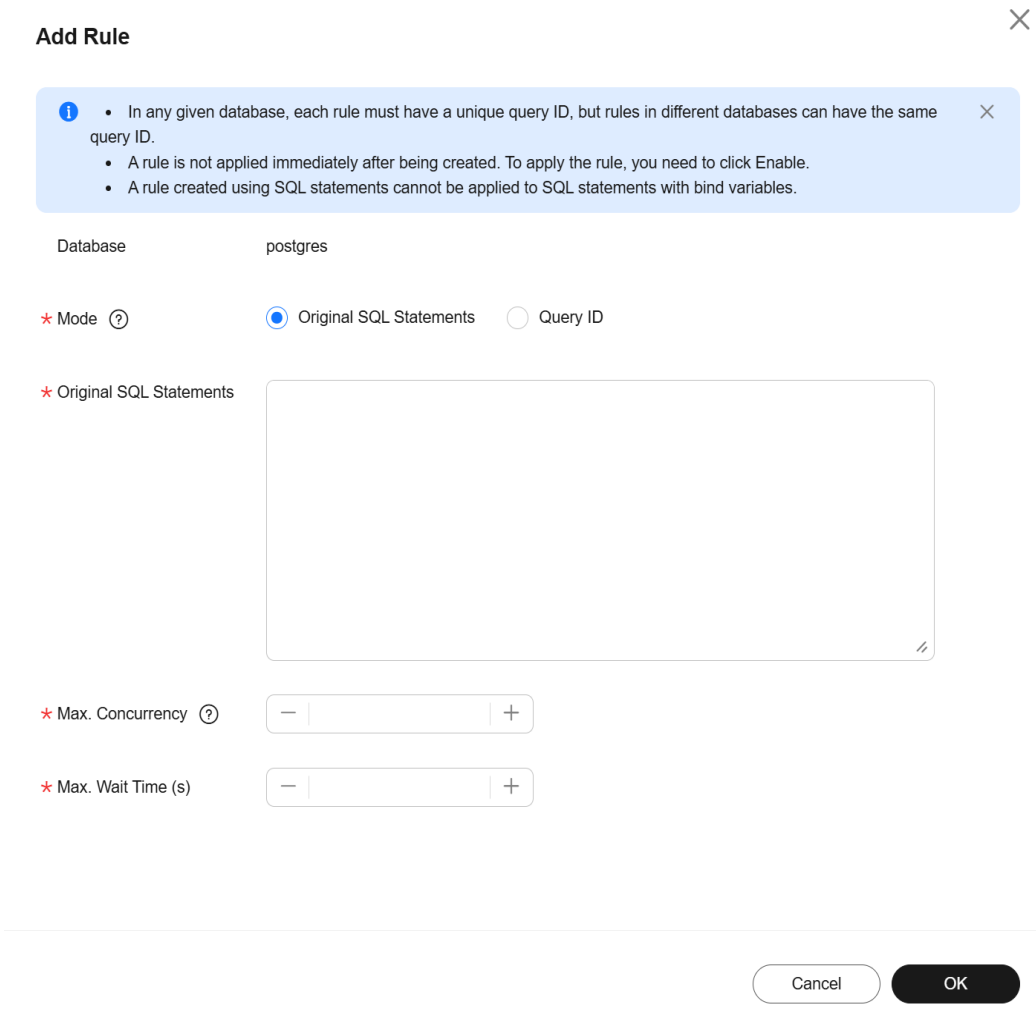


Table 3-54 Parameter description

Parameter	Description
Mode	• Original SQL Statements A rule created using SQL statements cannot be applied to SQL statements with bind variables. • Query ID In any given database, each rule must have a unique query ID, but rules in different databases can have the same query ID.
Max. Concurrency	The maximum allowed number of concurrent SQL statements that match the rule. SQL statements exceeding this upper limit will not be executed. The value ranges from 0 to 50000. **0** indicates that the concurrency of SQL statements is not limited.
Max. Wait Time	Wait time before the SQL statements can be executed. The value ranges from 0 to 1000000000.

Step 9 Confirm the settings and click **OK**.

Step 10 A rule is not applied immediately after being created. To apply the rule, click **Enable**.

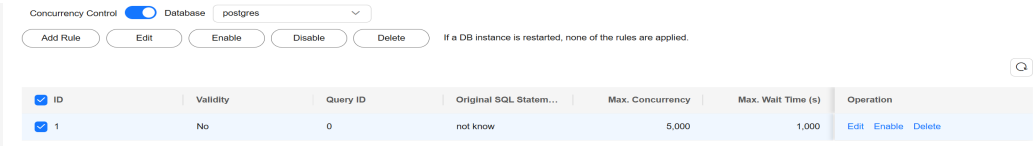
----End

Follow-up Operations

If a concurrency control rule is not required, you can disable or delete it in the **Operation** column.

- To disable a rule, click **Disable**.
- To delete a rule, click **Delete**. After a concurrency control rule that is enabled is deleted, the rule is not applied anymore.

Figure 3-139 Deleting a rule



3.14.5 Common Performance Problems

3.14.5.1 Troubleshooting High CPU Usage

Description

CPU usage refers to the percentage of CPU time occupied when the system is running.

CPU usage includes the user-mode CPU time percentage and kernel-mode CPU time percentage.

- User mode is the mode in which user programs are running.
- Kernel mode is the mode in which OS management programs are running, including system calls, kernel threads, and interrupts.

When the CPU is full, services slow down.

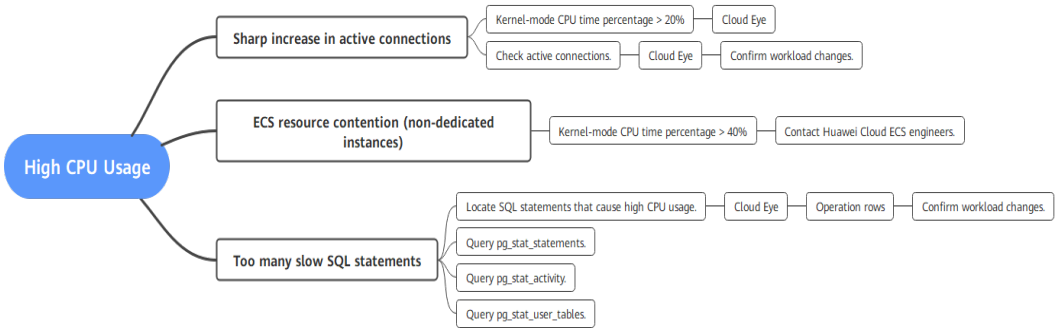
Cause Analysis

There are three possible causes:

- Sharp increase in active sessions
- ECS underlying resource contention (non-dedicated instances)
- Too many slow SQL statements

The following figure shows the troubleshooting methods for the three possible causes.

Figure 3-140 Cause analysis



Troubleshooting

- **Sharp increase in active connections**

  Check whether the percentage of kernel-mode CPU time is greater than 20% or whether the **Active Connections** metric increases sharply.

  – Checking the percentage of kernel-mode CPU time

    On the Cloud Eye console, check the **Kernel-mode CPU Time Percentage** metric in the last one hour.

    If this metric is higher than **20%**, there may be a large number of system calls or interrupts. In this case, a large number of processes are running in the system.

    NOTE

    When the number of active connections exceeds the instance specifications, the system continuously switches the processes running in the CPU. The kernel program switches the CPU to different address spaces. As a result, the kernel-mode CPU time percentage increases.

  – Checking active connections

    On the Cloud Eye console, check whether the **Active Connections** metric surged at a certain time point within the last 24 hours or last 7 days.

    NOTE

    In normal cases, the number of active sessions should be twice the number of vCPUs to maximize CPU usage.

- **ECS resource contention (non-dedicated instances)**

  A rare cause for kernel-mode CPU time percentage greater than 20% is ECS resource contention. This mainly occurs in non-dedicated instances (including general-purpose and general-enhanced instances).

  Generally, the kernel-mode CPU time percentage of RDS for PostgreSQL instances is lower than 10%. If this percentage is greater than 10%, check whether it is caused by ECS resource contention. **Submit a service ticket** to confirm this problem.

- **Too many slow SQL statements**

  RDS for PostgreSQL provides slow query logs. You can use these logs to locate the time-consuming SQL statements for further analysis. However, a slow SQL statement may also cause other SQL statements to run slowly, so you may find many slow SQL statements in the logs. It is hard to find the target SQL statement.

In addition to slow SQL statements, there are some simple SQL statements that may cause abruptly high CPU usage in some cases (for example, cyclic execution in a transaction or a large number of concurrent executions).

The following methods are recommended for tracing slow SQL statements:

a. Use the **pg_stat_statements** extension to locate the SQL statements that cause high CPU usage. For details, see **pg_stat_statements**.

b. Query the **pg_stat_activity** view to find the SQL statements that have been running for a long time.

```
SELECT  *,
(now() - backend_start) AS proc_duration,
(now() - xact_start) AS xact_duration,
 (now() - query_start) AS query_duration,
 (now() - state_change) AS state_duration
FROM pg_stat_activity
 WHERE pid<>pg_backend_pid()
ORDER BY state_duration DESC limit 10;
```

c. Query the **pg_stat_user_tables** view to find the tables that are being scanned in full mode and the corresponding SQL statements.

```
select * from pg_stat_user_tables order by seq_tup_read desc, seq_scan desc limit 10;
```

d. Check whether there are any slow SQL statements based on the **pg_stat_statements** or **pg_stat_activity** view.

NOTE

The **pg_stat_statements** extension must be installed first.

Check for slow SQL statements based on **pg_stat_statements**:

```
select * from pg_stat_statements where query like '%tablename%' order by shared_blks_hit + shared_blks_read desc;
```

Check for slow SQL statements based on **pg_stat_activity**:

```
select
  *,
  (now() - backend_start) AS proc_duration,
  (now() - xact_start) AS xact_duration,
  (now() - query_start) AS query_duration,
  (now() - state_change) AS state_duration
from pg_stat_activity
where pid<>pg_backend_pid() and query like '%tablename%'
ORDER BY state_duration DESC;
```

These slow SQL statements are usually caused by a lack of indexes. As a result, too many buffer reads are executed, consuming a large number of CPU resources.

Solution

- Sharp increase in active connections

  Check whether the sharp increase in active connections is necessary for your workload. If yes, upgrade the instance specifications. If no, optimize the workload to address the sharp increase. You can also kill unnecessary sessions to reduce the CPU usage. For details, see **Killing Sessions**.

  NOTE

  Killing a session may cause the application to disconnect from the instance. Your application should be able to reconnect to the instance.

- ECS resource contention (non-dedicated instances)

  Change your instance to a dedicated instance.

- Too many slow SQL statements
  Locate the problematic SQL statements and optimize them.

3.14.5.2 Troubleshooting High Memory Usage

Description

The memory usage of an RDS for PostgreSQL instance includes the usage of both shared memory and local memory.

- Shared memory: It is mainly used for the data buffer and WAL buffer to improve the read and write performance. It also stores some global information, such as process and lock information.
  The value of **shared_buffers** determines the size of the initial shared memory you can request. The initial value for this parameter is set to 25% of the physical memory for an RDS for PostgreSQL instance. The value ranges from 25% to 40%. If the value exceeds 40% of the physical memory, the buffer effect is not obvious. This is because RDS for PostgreSQL runs on the file system and if the file system also has a buffer, there will be two buffers, causing negative impacts.
- Local memory: Backend services need local memory to temporarily store data that does not need to be stored globally. Local memory is specified by the following parameters:
  - **temp_buffers** specifies the maximum amount of memory used for temporary buffers within each database session.
  - **work_mem** specifies the base maximum amount of memory to be used by a query operation (such as a sort or hash table) before writing to temporary disk files. Note that each sort operation instead of each SQL statement will use as much memory as the value of **work_mem**.
  - **maintenance_work_mem** specifies the maximum amount of memory to be used by maintenance operations.

Impact

Redundancy is required for the memory of a production DB instance. In normal cases, the memory usage must be less than 70%. If the memory usage continues to be higher than this limit, you are advised to upgrade the memory specifications. High memory usage may trigger an alarm and cause the following problems:

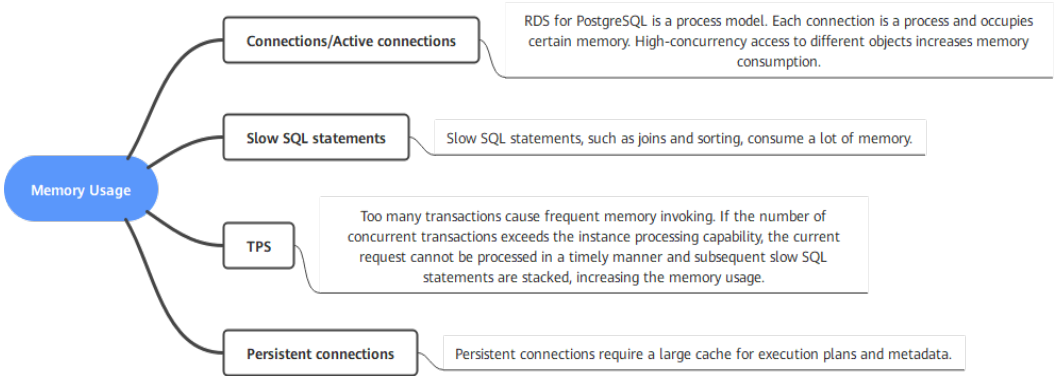
- Data is frequently swapped between memory and disks, which consumes a large number of CPU resources. As a result, the database performance deteriorates and data reads and writes are affected.
- In severe cases, an out of memory (OOM) problem may occur. If an OOM problem occurs, the database service process restarts, existing database connections are interrupted, and new connections cannot be established. Then the HA process restarts the DB instance. During this period, the instance replays the WAL logs generated from the last checkpoint to the time when the OOM problem occurred to ensure transaction consistency.

Cause Analysis

High memory usage is generally caused by an increase in connections, active connections, slow SQL queries, TPS, or persistent connections. If the memory

usage increases sharply or does not meet your expectation, analyze the causes as follows:

Figure 3-141 Cause analysis



Troubleshooting

- **Connections/Active Connections**
  - On the Cloud Eye console, check whether the memory usage, connection usage, number of database connections, and number of active connections increase or decrease simultaneously in the target period.
  - Run the following SQL statement to check the maximum number of connections allowed for the instance:
    ```
    show max_connections;
    ```
  - Run the following SQL statement to check the number of active connections to the instance:
    ```
    select count(1) from pg_stat_activity where state <> 'idle';
    ```
  - Run the following SQL statement to check the number of idle connections:
    ```
    select count(1) from pg_stat_activity where state = 'idle';
    ```
- **Slow SQL Statements**
  - On the Cloud Eye console, check whether the memory usage, number of SQL statements executed for more than 1s, number of SQL statements executed for more than 3s, and number of SQL statements executed for more than 5s increase or decrease simultaneously in the target period.
  - Run the following SQL statement to view the top three slow SQL statements (for RDS for PostgreSQL 10 and later versions) and check whether the SQL statements in the returned **query** field use the JOIN or ORDER syntax:

    select (extract(epoch from now() - query_start)) query_time, datname, usename, client_addr, wait_event, state, query from pg_stat_activity where state not like 'idle%' and query_start is not null and backend_type = 'client backend' and pid <> pg_backend_pid() order by 1 desc limit 3;
  - Query the pg_stat_statements view to obtain statistics and query the SQL statement that consumes the most shared memory. For details, see **4**.
- **TPS**

  On the Cloud Eye console, check whether the memory usage and TPS increase or decrease simultaneously in the target period.

- **Persistent Connections**
  - Run the SQL statement shown below to view the top three persistent connections (for RDS for PostgreSQL 10 and later versions). In the command output, the **conn_time** field indicates the connection lifetime, and the **query** field indicates the SQL statement executed by the process.

    ```
    select (extract(epoch from now()-backend_start)) conn_time, datname, pid, usename, client_addr, wait_event_type, wait_event, state, query from pg_stat_activity where backend_type = 'client backend' order by conn_time desc nulls last limit 3;
    ```

  - Persistent connections cache certain information, such as query results, transaction information, and lock information, in the database. If many persistent connections are maintained for a long period of time, the cached information increases accordingly, occupying more memory. To further locate the fault, query the pg_stat_statements view based on the value of the **query** field obtained in the last step and check how much shared memory the SQL statement has used.

    ```
    select userid::regrole, dbid, shared_blks_hit, shared_blks_dirtied from pg_stat_statements where query =  'query';
    ```

Solution

- **Too Many Connections or Active Connections**

  If there are too many connections or idle connections, run the SQL statement shown below or configure connection timeout for clients to release idle connections, or use a connection pool to reduce the overhead of establishing new database connections. If there are too many active connections, reduce the number of concurrent requests or upgrade the memory specifications. You can also kill unnecessary sessions to reduce the memory usage. For details, see **Killing Sessions**.

  NOTE

  Killing a session may cause the application to disconnect from the instance. Your application should be able to reconnect to the instance.

  ```
  select pg_terminate_backend(pid) from pg_stat_activity where state = 'idle';
  ```

- **Too Many Slow SQL Statements**

  Locate the SQL statements that consume much memory, optimize the SQL statements or upgrade the memory specifications.

- **High TPS**

  Reduce the number of transactions or upgrade the memory specifications.

- **Too Many Persistent Connections/Long Connection Lifetime**

  Periodically release persistent connections because maintaining them may generate a large cache and use up memory.

FAQ

Q: Why does the memory usage increase when pg_dump is used to export data? How do I avoid this problem?

A: When pg_dump is used to export data, a process accesses all objects such as tables and indexes in the target database to obtain structure data. If the accessed tables or indexes are too large, there may be large RelCache (relational table caches) or CatCache (system catalog table caches) that cannot be released. As a result, the memory usage increases and even an OOM problem occurs.

Suggestions for executing a pg_dump task:

1. Do not perform DDL operations.
2. Monitor the metric of slow SQL statements. If there is a lock conflict, kill the conflicting process.
3. Execute the pg_dump task during off-peak hours.
4. Decrease the values of **shared_buffers** and **work_mem** to 1/2 or 1/4 of the current values or less. After the task is complete, roll back the parameters.
5. Upgrade the memory specifications.

3.14.5.3 Troubleshooting Database Age Increase Problem

Description

In a given database, the maximum age between the earliest and latest transactions is 2 billion (2^31). When the age of a table is greater than the value of **autovacuum_freeze_max_age** (400 million by default for an RDS for PostgreSQL instance), the autovacuum process freezes the table.

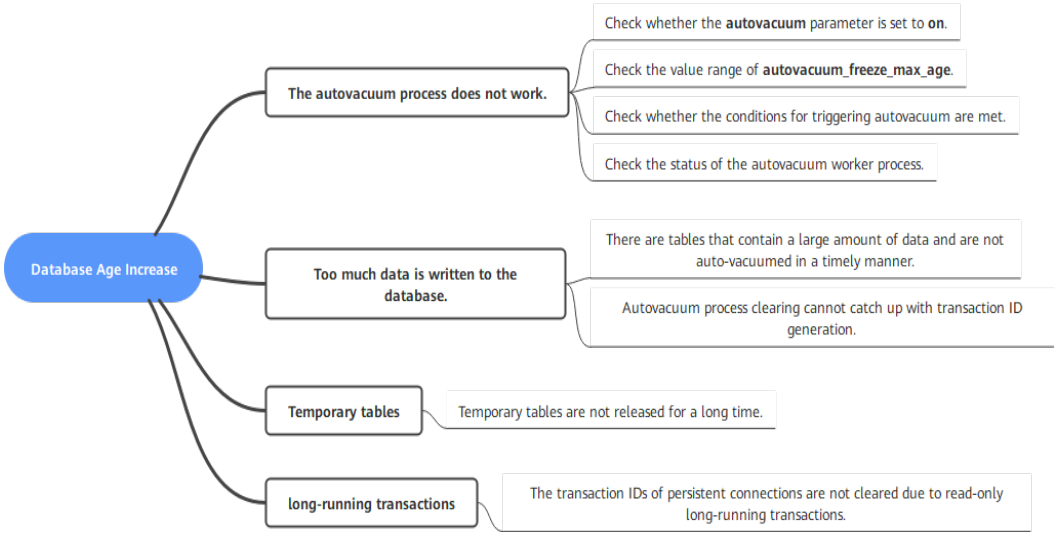
Once the database age exceeds 2 billion, the database breaks down and does not accept new transactions. You need to run VACUUM FULL in single-user mode to rectify the fault.

Cause Analysis

There are several possible causes:

1. The autovacuum process does not work.
2. Too much data is written to the database.
3. Temporary tables are not released for a long time.
4. There are read-only long-running transactions.

Figure 3-142 Cause analysis



Troubleshooting

- **The autovacuum process does not work.**

  There are many dead tuples in the database, and the vacuum operation is not performed. Do as follows:

  a. Check whether the **autovacuum** parameter is set to **on**.

  b. Check the value of **autovacuum_freeze_max_age**. The default value is 400 million for an RDS for PostgreSQL instance. If you change the value to a value greater than 1 billion, you are advised to decrease the value.

  c. Check whether the conditions for triggering autovacuum are met.

     The **autovacuum_vacuum_threshold** parameter specifies the minimum number of updated or deleted tuples needed to trigger a VACUUM in any one table.

     The **autovacuum_vacuum_scale_factor** parameter specifies a fraction of the table size to add to autovacuum_vacuum_threshold when deciding whether to trigger a VACUUM.

  d. Run the following SQL statement to check whether the autovacuum process is normal:

     ```
     select * from pg_stat_activity where backend_type like '%vacuu%';
     ```

- **Too much data is written to the database.**

  a. Check the database age.

     ```
     select datname, age(datfrozenxid) from pg_database where datname <> 'template1' and datname <> 'template0' order by age desc;
     ```

  b. Check whether the autovacuum parameters are properly set and compare them with those in the parameter template.

     ```
     SELECT name, setting FROM pg_settings WHERE name like '%vacuum%';
     ```

  c. Query the five oldest tables in the database.

     ```
     select relname, relfrozenxid,  age(relfrozenxid) aa from pg_class where relfrozenxid != 0 order by aa desc limit 5;
     ```

  d. Query the autovacuum status of these tables.

     ```
     SELECT schemaname, relname, last_vacuum, last_autovacuum, vacuum_count, autovacuum_count FROM pg_stat_all_tables WHERE relname='pg_toast_1335431529';
     ```

  e. Query the sizes of these tables.

     ```
     select pg_size_pretty(pg_relation_size(pg_toast_1335431536));
     ```

  f. Run the following command twice to check the value of **heap_blks_scanned** in the two execution results. If the value increases normally, autovacuum is running properly.

     ```
     select * from pg_stat_progress_vacuum;
     ```

     If autovacuum is running properly, check the disk read/write throughput and IOPS metrics in the last seven days. If the storage is fully occupied for a long period of time, the disk I/O is too high. Autovacuum process clearing cannot catch up with transaction ID generation and the database age increases.

     For details about the storage types and maximum throughput, see **Performance Comparison of DB Instance Storage Types**.

Figure 3-143 Viewing disk read/write throughput

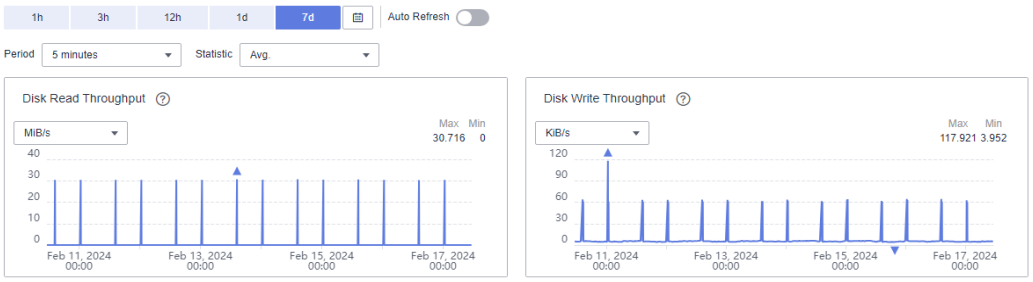
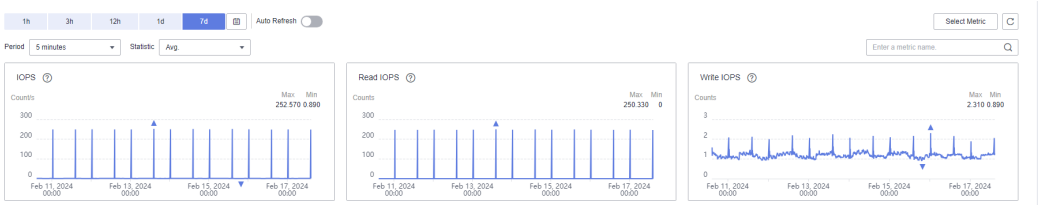


Figure 3-144 Viewing IOPS



- **Temporary tables are not released for a long time.**

  a. For details about the troubleshooting method, see **a** to **c**. If the oldest table in the query result starts with **tmp_%**, check whether the table is a temporary table by viewing its properties.

  b. View the properties of the oldest table. If the value of the **relpersistence** field is **t**, the table is a temporary table.

  ```
  select * from pg_class where relname ='tmp_table_pu';
  ```

  NOTICE

  In a database, temporary tables are not vacuumed, but their lifecycles are not long.

  Once the connection is released, the temporary tables are reclaimed.

  Therefore, you need to check whether there are persistent connections in the database by running the following statement:

  ```
  select (now()-backend_start) duration, *from pg_stat_activity  where backend_type = 'client backend' order by duration desc nulls  last;
  ```

  When any persistent connection is found, release it and then release the temporary table. Check whether the database age decreases.

- **There are long-running transactions.**

  Long-running transactions are also a cause for database age increase. You can query long-running transactions of an RDS for PostgreSQL instance on the Cloud Eye console or using SQL statements.

  a. Run the following statement to check whether there are long-running transactions:

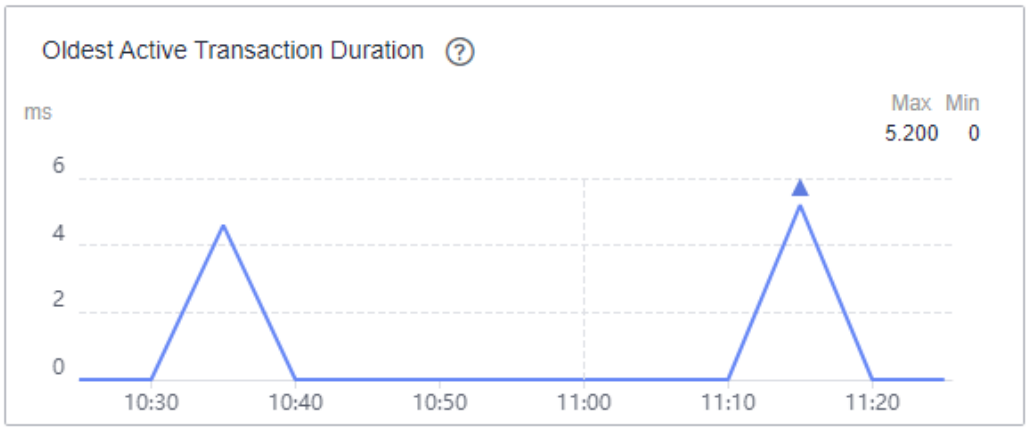
  ```
  select * from pg_stat_activity where state <> 'idle' order by xact_start;
  ```

  b. Alternatively, check the oldest active transaction duration on the Cloud Eye console and determine whether there are long-running transactions.

NOTICE

You can only determine whether there are long-running transactions, but cannot view details about the long-running transactions on the Cloud Eye console. You are advised to use both SQL statements and Cloud Eye to identify long-running transactions.

Figure 3-145 Viewing the oldest active transaction duration



c. If there is a long-running transaction, run the following SQL statement to cancel the long-running transaction:

```
-- Recommended:
select pg_cancel_backend($PID);
-- If the preceding statement is invalid, run the following statement:
select pg_terminate_backend($PID);
```

d. After canceling the long-running transaction, perform the vacuum operation on the oldest table in the database.

```
vacuum "Test20231127";
```

e. After the tablespace is cleared, run the following SQL statement for verification. If the value of **n_dead_tup** returns to **0** or is small, the restoration is complete.

```
SELECT schemaname, relname, n_live_tup, n_dead_tup,
FROM pg_stat_all_tables WHERE relname = 'Test20231127';
```

Solution

- The autovacuum process does not work.
  a. Check whether the **autovacuum** parameter is set to **on**. If no, set it to **on** and observe the database age.
  b. Check the value of **autovacuum_freeze_max_age**. The default value is 400 million for an RDS for PostgreSQL instance. If you change the value to a value greater than 1 billion, decrease the value and observe the database age.
- Too much data is written to the database.

  If the disk throughput reaches the performance upper limit, change the storage type.

  Run the VACUUM command to clear old tables.

- Temporary tables are not released for a long time.

  Temporary tables will not be auto-vacuumed. If the database age increases due to temporary tables, release the client connection to reclaim the temporary tables.
- There are long-running transactions.

  Cancel the long-running transactions and then run VACUUM on the oldest table in the database.

  a. Cancel the long-running transactions.
  ```
  select pg_cancel_backend($PID);
  ```
  b. Clear the table.
  ```
  vacuum table_name;
  ```

Reference

PostgreSQL official documentation

3.14.5.4 Troubleshooting High Storage Space Usage

Description

Redundancy is required for the storage space of a production DB instance. If the storage space usage is too high, handle the problem in a timely manner to prevent the instance from being damaged due to full storage.

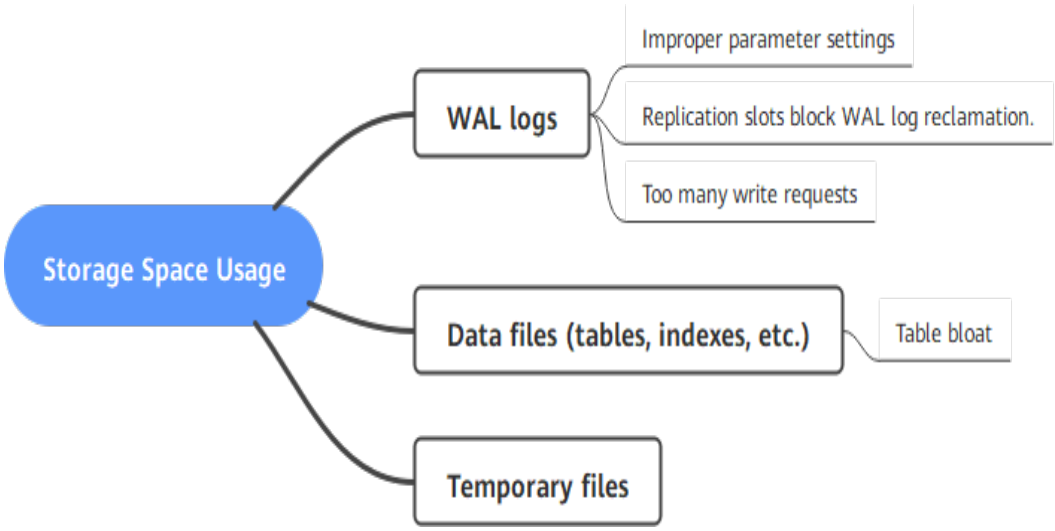
You need to pay attention to the following key metrics:

- Storage usage: rds039_disk_util
- Total storage: rds047_disk_total_size
- Used storage: rds048_disk_used_size
- Transaction logs (WAL logs) usage: rds040_transaction_logs_usage
- Oldest replication slot lag (WAL logs accumulated due to replication slot problems): rds045_oldest_replication_slot_lag

Cause Analysis

In an RDS for PostgreSQL instance, data files (such as tables and indexes), WAL logs, and temporary files may occupy the most storage space. If the storage usage increases unexpectedly, analyze the causes as follows:

Figure 3-146 Cause analysis



Troubleshooting and Solution

NOTICE

SQL statements for querying the storage usage of databases, tables, or WAL logs occupy a large amount of disk I/O. Therefore, run such SQL statements during off-peak hours.

- **Check whether the WAL log size is within its allowed range. If no, rectify the fault.**
  - Check the WAL log size.

    View the **rds040_transaction_logs_usage** metric or run the following SQL statement to check the WAL log size. If there are many WAL logs, perform the following steps to locate the fault.

    ```
    select round(sum(size)/1024/1024/1024,2) "GB" from pg_ls_waldir();
    ```

    NOTICE

    The **pg_ls_waldir()** function is available only in RDS for PostgreSQL 12 and later versions.

    User **root** is required to execute the **pg_ls_waldir** function.

  - Check the WAL log retention parameter.
    - For RDS for PostgreSQL 12 or earlier versions, check the value of **wal_keep_segments** (unit: MB). For later versions, check the value of **wal_keep_size** (unit: MB).
    - The value of the WAL log retention parameter should be greater than 4 GB but less than 10% of the total storage. Otherwise, the primary instance may clear the WAL logs required by the standby instance, causing exceptions on the standby instance.

  - Check the replication slot status and the size of logs that are not cleared.

    Replication slots block WAL reclamation. If inactive or unnecessary replication slots are found, delete them as required.

    Run the following SQL statement to query the status of a replication slot and uncleared WAL logs:

    ```
    select slot_name, active,
    pg_size_pretty(pg_wal_lsn_diff(b, a.restart_lsn)) as slot_latency
    from pg_replication_slots as a, pg_current_wal_lsn() as b;
    ```

    Run the following SQL statement to delete a slot:

    ```
    select pg_drop_replication_slot('slot_name');
    ```

  - Check how busy write services are.

    View the **rds044_transaction_logs_generations** metric to determine how busy write services are. This metric indicates the average size of transaction logs (WAL logs) generated per second.

    If the value of this metric is large, there are a large number of write services. In this case, the database kernel reserves more WAL logs for reclamation, and the storage usage of WAL logs increases. You are advised to scale up storage to ensure storage redundancy.

- **Check whether the size of data files is normal. If no, rectify the fault.**
  - Query the top 10 databases with the highest storage usage.

    ```
    select datname, pg_database_size(oid)/1024/1024 as dbsize_mb from pg_database order by dbsize_mb desc limit 10;
    ```

  - View the top 10 objects (tables/indexes) with the highest storage usage.

    You can use the **relpages** field of pg_class to estimate the size of a table or index. The SQL statement is as follows:

    ```
    select relname, relpages*8/1024 as tablesize_mb from pg_class order by tablesize_mb desc limit 10;
    ```

    To obtain the exact size of a table or index, use any of the following functions:

    Table 3-55 Function description

Name	Return Type	Description
pg_relation_size(relation regclass, fork text)	bigint	Storage space used by a specified fork ('main', 'fsm', 'vm', or 'init') of a specified table or index
pg_relation_size(relation regclass)	bigint	Shorthand for pg_relation_size(..., 'main')
pg_table_size(regclass)	bigint	Storage space used by a specified table, excluding indexes (but including TOAST, free space map, and visibility map)
pg_total_relation_size(regclass)	bigint	Total storage space used by the specified table, including all indexes and TOAST data

- Check whether there is any table bloat.

  Once the table that occupies a large amount of storage space is determined, you can use the pgstattuple extension to analyze whether the table is bloated. The extension can be installed by running the following statements:

  ```
  create control_extension('create', 'pgstattuple');
  select * from pgstattuple('table_name');
  ```

  NOTICE

  Some kernel versions do not support the pgstattuple extension. For details, see **Supported Extensions**.

  For details about how to use this extension, see **https://www.postgresql.org/docs/15/pgstattuple.html**.

- Clear table data.

  - If any table bloat is found, you can vacuum the table in the maintenance time window.

    NOTICE

    VACUUM FULL locks the table. Ensure that no DML operation is being performed during the operation.

    ```
    vacuum full table_name;
    ```

  - If any unnecessary table or data is found, you can use the **truncate table** or **drop table** statement to delete unnecessary data.

    ```
    truncate table table_name;
    ```

  - **The DELETE operation does not release storage space. Instead, there will be a large number of WAL logs generated, which increases the storage space consumption. Do not use DELETE to release storage space when the storage is full.**

    According to the Multi-Version Concurrency Control (MVCC) mechanism of PostgreSQL, the DELETE operation does not release storage space (deleted data is marked as invisible). The storage space can be released only after VACUUM FULL is executed. The VACUUM FULL operation also consumes storage space and locks the table. Therefore, perform this operation during off-peak hours and reserve at least twice the size of the table.

  - If only a small amount of data needs to be retained, you can create a new table and transfer the data to the table. The procedure is as follows:

    1) Store information such as the structure and indexes of the original table.
    2) Create a new table.
    3) Insert data into the new table.

4) Check whether the data in the new table meets the expectation. If yes, go to the next step. If no, check whether the previous operations are successful.
5) Delete the original table.
6) Rename the new table and create indexes.

NOTE

VACUUM FULL rebuilds the table and its indexes. During this period, WAL logs are generated. Sufficient storage space needs to be reserved. (Assume that the size of the rebuilt table is 1 GB and the size of indexes is 0.5 GB. You are advised to reserve at least 2.5 GB of storage space.)

For details about vacuum, see **https://www.postgresql.org/docs/current/routine-vacuuming.html**.

- If the storage usage exceeds 97%, the instance becomes read-only and data cannot be cleared using **drop** or **truncate**. To solve this problem, use either of the following methods:
  - **Scale up storage space**. If the storage capacity has reached the upper limit of your DB instance class, **upgrade the instance class** first.

    After the storage usage drops below 87%, the instance becomes readable and writable. Then, delete unnecessary data. You can enable **storage autoscaling** for instances using cloud disks. When the storage usage reaches the threshold, autoscaling is triggered.
  - If you do not want to scale up the storage, **submit a service ticket** to remove the read-only status and then delete unnecessary data. Before removing the read-only status, stop your workloads. If data continues to be written to the disk after the read-only status is removed, the storage will be getting full again.
- **Check whether the size of temporary files is normal. If no, rectify the fault.**

  If high storage usage is not caused by data files or WAL logs, temporary files may occupy a large amount of storage space. Run the following SQL statement to check the size of temporary files:

```
select round(sum(size)/1024/1024/1024,2) "GB" from pg_ls_tmpdir();
```

NOTICE

- The **pg_ls_waldir()** function is available only in RDS for PostgreSQL 12 and later versions.
- User **root** is required to execute the **pg_ls_waldir** function.
- When there are a large number of temporary files, the SQL statement execution is slow.

Generally, temporary files are released after complex SQL statements are executed. However, if an OOM exception occurs, temporary files may fail to be released. To reduce the generated temporary files, analyze and optimize slow SQL statements. Or you can reboot the instance in the maintenance time window to delete all temporary files.

3.14.5.5 Troubleshooting Abnormal Connections and Active Connections

Description

When the number of database connections reaches the upper limit allowed for a DB instance, subsequent connections will be rejected. Abnormal changes in the number of connections and the number of active connections can indicate workload changes and database status to some extent.

RDS for PostgreSQL provides two related metrics:

- Database connections: number of backend connections to a DB instance.
- Active connections: number of active connections to a DB instance.

Cause Analysis

Connection pool parameter modification and workload change are considered as normal changes.

If you cannot determine whether there are normal changes or abruptly high concurrency, check database connection information.

- The number of database connections suddenly decreases and then is restored to its normal range.

  Possible cause: Some connections are interrupted, or the DB instance reboots unexpectedly due to an OOM problem or crash.
- The number of database connections increases suddenly or reaches the maximum.

  Possible cause: The number of new connections is greater than that of closed connections per unit time.
- The concurrency is reduced and connections are not released in a timely manner if any of the following issues occurs:
  - There are slow SQL statements.
  - There are lock conflicts.
  - There are long-running transactions.

Troubleshooting

- **Query database connection information.**

  You can sort out the database connection information using the information combination in the pg_stat_activity view.

  The following shows an example.

  ```
  -- Database connection information is filtered by database name, username, client IP address, and status and then sorted by the number of client connections in descending order.
  SELECT datname, usename, client_addr, state, count(*) AS client_number
  FROM pg_stat_activity
  WHERE state <> 'idle'
  GROUP BY datname, usename, client_addr, state
  ORDER BY client_number DESC;
  ```

  The preceding query result shows where the most connections come from. In this way, you can identify workload changes and high concurrency.
- **Check whether the maximum number of connections has been reached.**

If your instance cannot be connected and the following error logs are generated, the number of database connections reaches the upper limit.

```
FATAL: remaining connection slots are reserved for non-replication superuser connections.
FATAL: sorry, too many clients already.
```

- **Check for any abnormal reboot.**
  a. Check the memory usage metric for any abnormal changes in memory usage.
  b. Download error logs of the corresponding time period by referring to **Viewing and Downloading Error Logs**.
  c. Use the keyword **killed** or **the database system is in recovery mode** to determine the time when the reboot occurred.
- **Slow SQL statements**

  In most cases, slow SQL statements are accompanied by high CPU usage. You can locate slow SQL statements by referring to **Troubleshooting High CPU Usage**.
- **Lock conflicts**
  a. Query the lock status of the current database connection using the pg_stat_activity view and the pg_blocking_pids function.

```
-- Query the lock statuses of the earliest five PIDs (client connections) for the current transaction.
SELECT pg_blocking_pids(pid), array_length(pg_blocking_pids(pid), 1) blocking_num, *
FROM pg_stat_activity
WHERE pid IN (select pid FROM pg_stat_activity WHERE state <> 'idel'
AND xact_start IS NOT NULL ORDER BY xact_start DESC LIMIT 5)
AND pid <> pg_backend_pid()
ORDER BY blocking_num DESC NULLS LAST;
```

  b. Based on the preceding query result, determine whether there are many lock conflicts that prevent the connections from being released.
- **Long-running transactions**

  Locate long-running transactions by referring to **Troubleshooting Long-Running Transactions**.

Solution

- **Normal workload change or increased concurrency**

  Upgrade the DB instance specifications.
- **Excessive database connections**

  Temporary solution:
  a. Run SQL statements to release idle connections as user **root**.

     For example, to query the idle connection of the user **user**, run the following SQL statement to obtain the process ID:

     select * from pg_stat_activity where state = 'idle' and usename = 'user';

     Release the idle connection.

     select pg_terminate_backend(*pid*);
  b. Increase the value of **max_connections** and reboot your instance for the new value to be applied.

  Permanent solution:

a. Reduce database connections.
b. If database connections cannot be reduced, upgrade the instance specifications.

- **OOM exception or crash**

  If the memory usage is high for a long time, upgrade the instance specifications or optimize the workloads to reduce the resident memory usage. If the DB instance reboots due to SQL statements, optimize the SQL statements.

- **Slow SQL statements**

  Locate the slow SQL statements and optimize them.

- **Lock conflicts**

  Check whether applications can be disconnected. If yes, use the pg_cancel_backend function to disconnect the applications from the database.

- **Long-running transactions**

  Handle long-running transactions by referring to **Troubleshooting Long-Running Transactions**.

3.14.5.6 Troubleshooting Long-Running Transactions

What Is a Long-Running Transaction?

A long-running transaction refers to a transaction in which DDL or DML operations are being performed and are not committed for a long time. Long-running transactions may:

- Exhaust I/O resources.
- Occupy a large number of CPUs.
- Lock resources and reduce concurrency.
- Cause table bloats.

Metrics Related to Long-Running Transactions

There are two long-running transaction metrics: **oldest_transaction_duration** and **oldest_transaction_duration_2pc**. The latter is used for two-phase transactions. You can check the metrics using either of the following methods:

- Method 1: **Go to the Cloud Eye console**.
- Method 2: Use SQL statements.
  - To check the longest transaction lifetime (**oldest_transaction_duration**), log in to the instance and run the following SQL statement in any database:

    ```
    select EXTRACT (EPOCH FROM max(now()-xact_start)) from pg_stat_activity where backend_type = 'client backend' and state <> 'idle';
    ```

  - To check the longest pending transaction lifetime (**oldest_transaction_duration_2pc**), log in to the instance and run the following SQL statement in any database:

    ```
    select coalesce(EXTRACT (EPOCH FROM now() - min(prepared)), 0) from pg_prepared_xact();
    ```

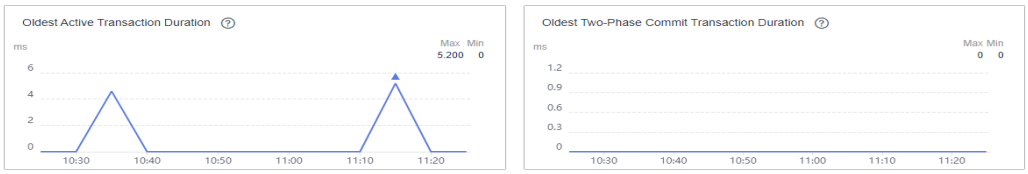

Cause Analysis

- There are batch operations.
- There are a lot of lock contentions.

Troubleshooting

1. Check **oldest_transaction_duration** and **oldest_transaction_duration_2pc** for long-running transactions. For details, see **Viewing Monitoring Metrics**.

   Figure 3-147 Checking long-running transaction metrics

2. Log in to the instance and run the following SQL statement in any database to obtain transaction activities from pg_stat_activity:

   ```
   select (now() - xact_start) trans_time, pid, datname, usename, client_addr, wait_event, state, substring(query, 1,50) from pg_stat_activity where state <> 'idle' and xact_start is not null and backend_type = 'client backend' and pid <> pg_backend_pid() order by 1 desc limit 3;
   ```

3. Run the following SQL statement in any database to query two-phase long-running transactions from pg_prepared_xacts:

   ```
   select * from pg_prepared_xacts order by 3 desc;
   ```

Solution

- Terminate long-running transactions by running the following SQL statement:

  ```
  select pg_cancel_backend($PID);
  ```

  If the **pg_cancel_backend** statement is invalid, run the following SQL statement:

  ```
  select pg_terminate_backend($PID);
  ```

  NOTE

  $PID: long transaction process ID, which can be obtained from **2**.

- Perform batch operations during off-peak hours.
- **Set alarm rules** for **oldest_transaction_duration_2pc** and commit pending transactions in a timely manner.

3.14.5.7 Troubleshooting Inactive Logical Replication Slots

Description

Inactive logical replication slots must be cleared in a timely manner for a production DB instance. If the value of the **Inactive Logical Replication Slots** metric is no less than 1 in three consecutive periods, there are inactive logical replication slots.

Inactive logical replication slots have the following impacts:

- Residual inactive logical replication slots retain resources required for logical replication. WAL logs cannot be cleared, occupying much storage space or even causing full storage.

- Applications may not run as expected, which may cause risks.

Troubleshooting

To troubleshoot inactive logical replication slots, perform the following steps:

1. Check whether any logical replication slot is not in use.
2. Determine whether the logical replication slot you found is still needed.
3. If it is no longer used, delete it.

Solution

1. Check whether any logical replication slot is not in use.

   Run the SQL statement shown below on the publisher to check whether any logical replication slot is not in use:

   If command output is displayed, there is an inactive logical replication slot in the instance. **slot_name** in the command output indicates the name of the inactive logical replication slot.

   ```
   select slot_name,database,active from pg_replication_slots where active ='f' and slot_type='logical';
   ```

2. Determine whether the logical replication slot you found is still needed.

   If no, go to **3**.

   If yes, do as follows:

   a. Check whether the replication of the logical replication slot is not enabled during subscription creation on the subscriber.

      Run the following SQL statement on the subscriber to check the return value in the **subenabled** column:

      ```
      select  subname,subenabled from pg_subscription;
      ```

      - If the return value is **f**, logical replication is not enabled for the subscription. Run the following SQL statement to enable logical replication:

        ```
        ALTER SUBSCRIPTION sub_name ENABLE;
        ```

      - If the return value is **t**, go to the next step.

   b. Check whether the logical replication slot is not cleared after any other tool is used to execute a task due to task interruption or exceptions in source or destination database operations, such as backup and index rebuilding.

      Run the SQL statement shown below on the publisher and check whether the return value of **slot_name** starts with **drs**. If yes, the logical replication slot is generated during DRS task execution. In this case, you can clear the slot as needed.

      ```
      select slot_name,database,active from pg_replication_slots where active ='f' and
      slot_type='logical';
      ```

3. If the logical replication slot is no longer used, delete it.

   ```
   select pg_drop_replication_slot('slot_name');
   ```


3.14.5.8 Troubleshooting High Oldest Replication Slot Lag or Replication Lag

Description

- Oldest replication slot lag: lagging size of the most lagging replica in terms of WAL data received. You can run the following SQL statement to view the lags of replication slots used by replicas:

```
select slot_name, temporary, active,restart_lsn, confirmed_flush_lsn, master_lsn,
pg_size_pretty(pg_wal_lsn_diff(master_lsn, a.restart_lsn)) as latency from pg_replication_slots a,
pg_current_wal_lsn() as master_lsn;
```

- Replication lag: delay between the time when data is written to the primary instance and the time when data is replicated to a replica.

A high oldest replication slot lag or replication lag may have the following impacts:

- The primary database server retains required WAL logs. WAL logs are stacked, occupying much storage space or even causing full storage space.
- If the replication lag is high, WAL log playback on the replica is slower than log generation on the primary instance, so data cannot be synchronized to the replica in real time.

Cause Analysis

In RDS for PostgreSQL, pay attention to the oldest replication slot lag and replication lag between the primary instance and read replicas. The possible causes for increase in the metrics are the following:

- Heavy workload on the primary instance
- High playback delay for read replicas
- Network delay between the primary instance and read replicas

Troubleshooting and Solution

1. Heavy workload on the primary instance

   Check whether there are many data writes or updates in the primary instance.

   On the Cloud Eye console, view the **Transaction Logs Generation** metric of the primary instance. Check whether the metric exceeds 40 MB/s in a given period. (Generally, WAL logs are replayed at a speed of 40 MB/s on read replicas.)

   – If the metric exceeds this threshold for a long period of time, there is a heavy workload on the primary instance. In this case, optimize the workload.

     If the workload on the primary instance is heavy, the value of **sent_lsn** in the **pg_stat_replication** view is greatly different from the query result of **select pg_current_wal_lsn();** executed on the primary instance. You can run the following SQL statements to check the difference:

     i. Confirm the read replica node information. Run the following SQL statement on the primary instance and record the value in the **sent_lsn** column as **lsn1**.

```
select * from pg_stat_replication;
```

     ii. Query the current WAL write location on the primary instance and record it as **lsn2**.


```
select pg_current_wal_lsn();
```

iii. Calculate the distance between the WAL write location and the sent WAL location.

```
select  pg_size_pretty(pg_wal_lsn_diff(lsn1,lsn2));
```

– If the metric value does not exceed the threshold, go to the next step.

2. High playback delay for read replicas

   If the read replica has long-running transactions or is heavily loaded, the query on the read replica conflicts with the log replay. The read replica fails to send the query result to the primary instance, causing a replay delay. You can download the error logs of the read replica and check whether such errors have been logged.

```
ERROR: canceling statement due to conflict with recovery
Detail:  User query might have needed to see row versions that must be removed
```

   – If yes, perform the following operations:

     ▪ Avoid long-running transactions. For details about how to troubleshoot long-running transactions, see **Troubleshooting Long-Running Transactions**.

     ▪ Set **hot_standby_feedback** to **on** for the read replica to minimize query conflicts.

   – If no, go to the next step.

3. Network delay between the primary instance and read replicas

   If the value of **sent_lsn** queried on the primary instance is greatly different from that of **pg_last_wal_receive_lsn** queried on the read replica, there is a long network delay between the primary instance and read replica. In this case, **submit a service ticket** to locate the fault.

   To check the difference, run the following SQL statements:

   a. Run the following SQL statement on the primary instance and record the value in the **sent_lsn** column as **lsn1**:

```
select * from pg_stat_replication;
```

   b. Query the value of **pg_last_wal_receive_lsn()** on the read replica and record it as **lsn2**.

```
select pg_last_wal_receive_lsn();
```

   c. Calculate the distance between the sent WAL location of the primary instance and the received WAL location of the read replica.

```
select  pg_size_pretty(pg_wal_lsn_diff(lsn1,lsn2));
```

3.14.5.9 Troubleshooting SQL Statements That Have Been Executed for 3s or 5s

Description

If there are SQL statements that have been executed for 3s or 5s in a DB instance, the SQL statements are running slowly and cannot obtain results in a timely manner.

Cause Analysis

The possible causes are as follows:

- A SQL statement is blocked and waits for the execution of other transactions.
- The usage of system resources such as CPUs is high.
- There are no indexes.
- Sorting in a SQL statement slows down the I/O execution.

Troubleshooting

- **Blocked SQL statement**

  a. Find the process waiting for a lock.

  ```
  SELECT locktype, relation, relation::regclass AS relname, virtualxid, transactionid,
  virtualtransaction, pid, mode, granted FROM pg_locks where granted = 'f';
  ```

  b. Obtain the PID of this process from **a** and find the process that blocks this process (value of the **pg_blocking_pids** field in the query result).

  ```
  select pg_blocking_pids(pid), array_length(pg_blocking_pids(pid), 1) blocking_num, * from
  pg_stat_activity  where pid = <pid from step 1> order by blocking_num;
  ```

  c. Query the blocked SQL statement.

  ```
  select * from pg_stat_activity where pid = <pid from step 2>;
  ```

  d. Determine the blocked SQL statement, for example, **update xxx;**. View the PID of the blocked process through the **pg_stat_activity** view.

  - **pg_blocking_pids**: ID of the blocking process
  - **wait_event_type**: type of the wait event

  ```
  select pg_blocking_pids(pid), array_length(pg_blocking_pids(pid), 1) blocking_num, *
  from pg_stat_activity
  where query like '%update xxx%'
  and pid <> pg_backend_pid()
  order by blocking_num desc NULLS LAST;
  ```

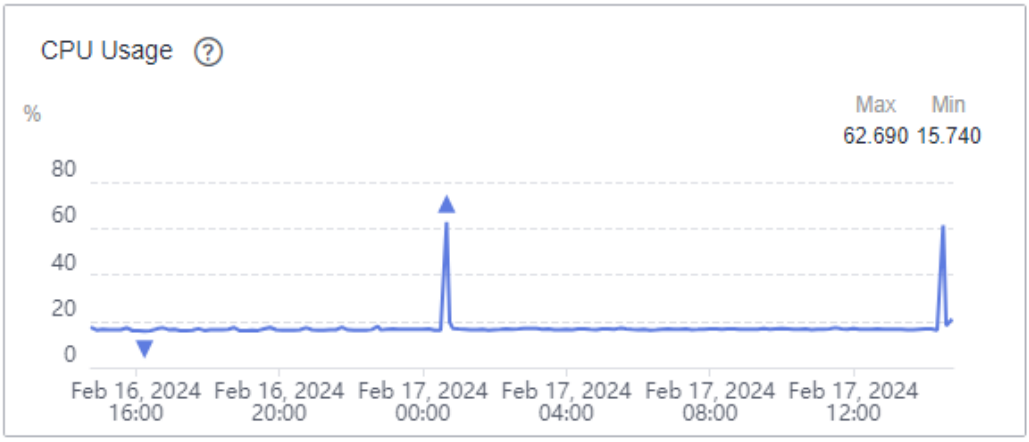
  After finding the PID of the blocked SQL statement, you can run the **SELECT pg_terminate_backend($PID);** command to end the process.

- **High usage of system resources**

  If SQL statements are running slowly but no blocking is found, you can view monitoring metrics.

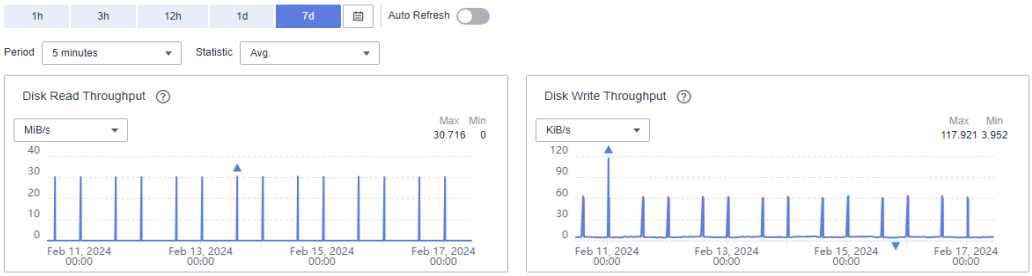
  On the management console, check whether the CPU usage is always high in the last seven days.

  Figure 3-148 Viewing CPU usage



If the CPU usage remains above 80%, check the disk read throughput and disk write throughput metrics to see whether read and write operations are performed continuously.

Figure 3-149 Viewing the disk throughput



Find the SQL statement that is running slowly and check whether its execution is also slow during off-peak hours. If it runs quickly during off-peak hours, the slow execution of the SQL statement is caused by high system resource usage.

- **Missing indexes**

  Locate the slow SQL statement and view the execution plan by running **explain (ANALYZE, VERBOSE, BUFFERS)**.

  ```
  xxx1:15:33.140 +08:10.51.11.21(53162):xxx:LOG:  duration: 22035.462 ms  plan:
     Query Text:  xxxxxxxxxxxxx
     Nested Loop  (cost=1.07..892.50 rows=1 width=806)
       Join Filter: ((xxx))
       ->  Nested Loop  (cost=1.07..25.41 rows=1 width=565)
             ->  Nested Loop  (cost=0.66..16.97 rows=1 width=523)
                   ->  Index Scan using xxx  (cost=0.25..8.54 rows=1 width=318)
                         Filter: ((itype = 1) AND (dr = 0))
                   ->  Index Scan using xxxx  (cost=0.41..8.43 rows=1 width=247)
                         Index Cond: (xxx)
             ->  Index Scan using xxx  (cost=0.41..8.43 rows=1 width=84)
                   Index Cond: (xxx)
       ->  Seq Scan on xxx log2  (cost=0.00..867.00 rows=1 width=66)
  ```

  The execution plan shows that a full table scan was performed on a table and took the most time.

  ```
  Seq Scan on xxx log2
  ```

  Confirm whether:

  a. The table has indexes.

  b. A large amount of data was added, deleted, modified, or queried during the execution of the SQL statement.

  Re-create indexes and run the SQL statement again to check whether the problem persists.

- **Sorting used in a SQL statement**

  If the running SQL statement contains the **GROUP BY** operation, similar to the following:

  ```
  select xx1,xx2,xx3,xx4,xx5,xx6
        from tbl_xxx
        GROUP BY xx1,xx2,xx3,xx4,xx5,xx6;
  ```

  the execution can take a long time. You can view the execution time of the SQL statement in the execution plan by running **explain (ANALYZE, VERBOSE, BUFFERS)**.


```
xxxx  (cost=439756.01..470149.19 rows=189957 width=28) (actual time=18072.697..20311.874
rows=323770 loops=1)
  Group Key: xxxx, xxxx
  Filter: (xxxxxx)
  Rows Removed by Filter: 192
  ->  Sort  (cost=439756.01..444504.94 rows=1899574 width=20) (actual time=18072.671..19960.595
rows=1834158 loops=1)
        Sort Key: xxxxxxxxxx
        Sort Method: external merge  Disk: 61056kB
        ->  Result  (cost=0.00..163739.61 rows=1899574 width=20) (actual time=0.009..927.709
rows=1834158 loops=1)
              ->  Append  (cost=0.00..144743.87 rows=1899574 width=20) (actual time=0.008..791.301
rows=1834158 loops=1)
                    ->  Seq Scan on xxxx  (cost=0.00..0.00 rows=1 width=212) (actual time=0.004..0.004
rows=0 loops=1)
```

Sort Method: external merge Disk: 61056kB indicates that the sorting operation occupies the disk I/O. In this case, you can run the following SQL statement to disable the sequential query and check the execution plan again:

```
set enable_seqscan = off;
```

If the sorting operation still uses the disk I/O, you can adjust the value of **work_mem** to increase the query memory.

NOTICE

Set **work_mem** to match your workloads. If the value is too large, an OOM problem may occur.

Solution

- **Blocked SQL statement**

  When you find the blocked SQL statement, run **SELECT pg_terminate_backend($PID);** to end the process.

  After it is ended, perform **a** again to check whether there is any other lock conflict.

- **High usage of system resources**

  Optimize workloads to reduce the number of concurrent requests.

- **Missing indexes**

  Re-create indexes and use the indexes for query.

- **Sorting used in a SQL statement**

  Increase the value of **work_mem** to reduce the I/O execution time for sorting.

3.15 Security and Encryption

3.15.1 Database Account Security

Password Strength Requirements

- For information about the database password strength requirements on the RDS console, see the database configuration table in **Buy a DB Instance**.

- 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.

SSL Encryption

SSL is enabled by default for RDS for PostgreSQL DB instances and cannot be disabled.

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 RDS for PostgreSQL DB instances. These system accounts are unavailable to you.

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.
- pg_execute_server_program: account that allows users who run the database to execute programs on the database server to cooperate with COPY and other functions that allow the execution of server programs.
- pg_read_all_settings: account that reads all configuration variables, even those that are usually visible only to the super user.
- pg_read_all_stats: account that reads all **pg_stat_*** views and uses various extension-related statistics, even those that are usually visible only to the super user.
- pg_stat_scan_tables: account that executes a monitoring function that may obtain an **ACCESS SHARE lock** on the table (and may hold the lock for a long time).
- pg_signal_backend: account that sends a signal (for example, a signal for canceling a query operation or an abortion signal) to another backend.

- pg_read_server_files: account that allows a database user to use the COPY and other file access functions to read files from any accessible directory on a server.
- pg_write_server_files: account that allows a database user to use the COPY and other file access functions to write files to any accessible directory on a server.
- pg_monitor: account that reads and executes various monitoring views and functions. It is a member of pg_read_all_settings, pg_read_all_stats, and pg_stat_scan_tables.
- 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.
- __rds_pg_profile_user_: metric monitoring account, which is used by the pg_profile_pro extension to collect database status data. This account is available only for the latest version of RDS for PostgreSQL 12 and is automatically created after pg_profile_pro is created.

CAUTION

pg_profile_pro is not supported temporarily due to its defects. Therefore, this account will not be automatically created for new instances.

3.15.2 Resetting the Administrator Password to Restore Root Access

Scenarios

If you forget the password of the administrator account **root**, you can reset the password. The new password is applied immediately without rebooting the instance. You can only reset the administrator password from the primary instance.

Precautions

- If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password.
- 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.
- 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

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 (~ ! @ # $ % ^ * - _ = + ? ,). 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 On the **Overview** page, find **Administrator** and click **Reset Password** under it.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Enter and confirm the new password.

NOTICE

Keep this password secure. The system cannot retrieve it.

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 (~ ! @ # $ % ^ * - _ = + ? ,). 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

3.15.3 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 be changed as well.

Precautions

You can add or modify rules for the security group associated with your RDS instance, but cannot disassociate or delete the security group.

Managing Security Groups

Step 1 On the **Instances** page, click the DB instance or read replica.

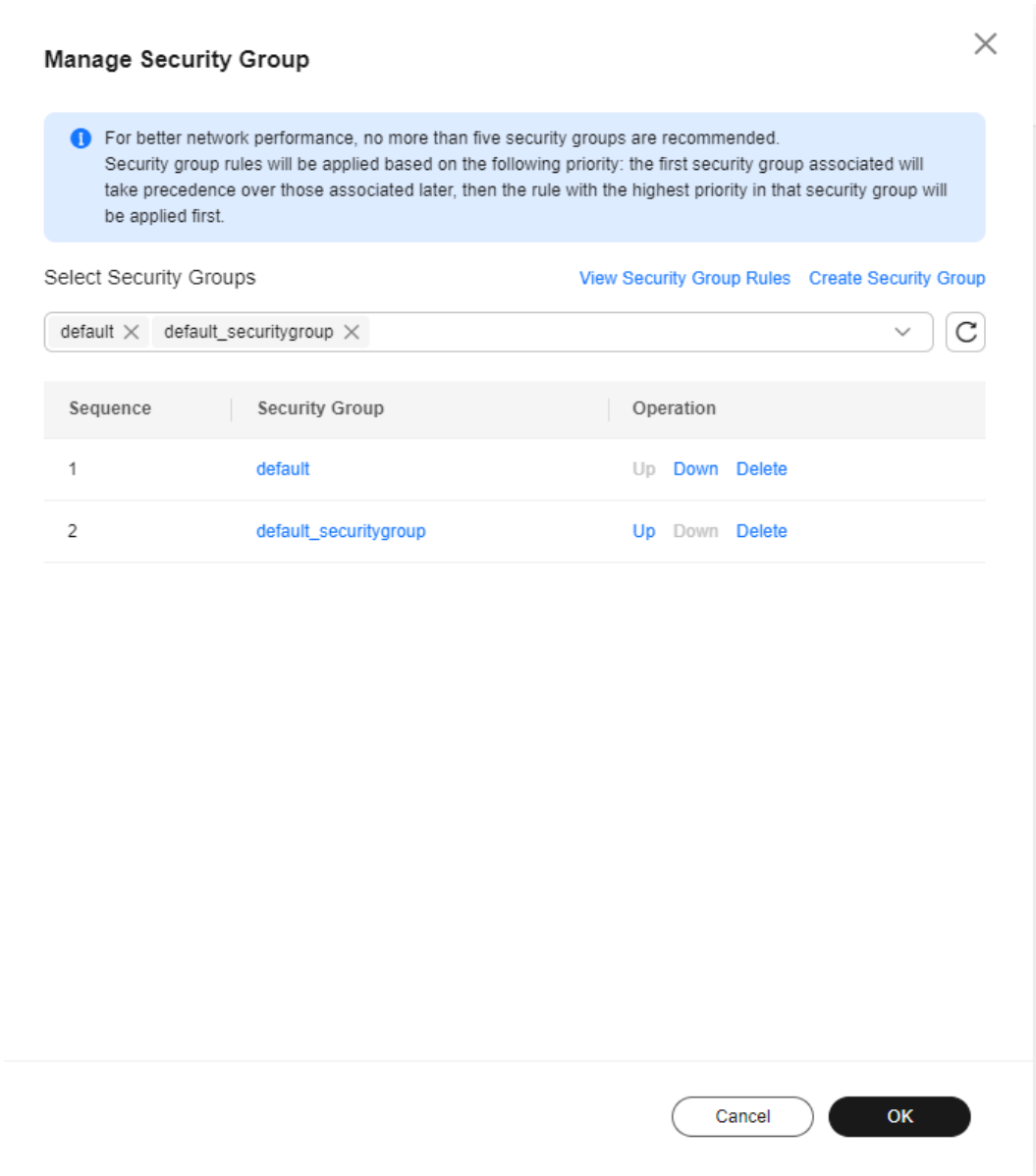
Step 2 On the **Overview** page, click **Manage** under **Security Group**.

- You can select multiple security groups at a time. The security group rules will be applied based on the following sequence: the first security group associated will take precedence over those associated later, then the rule with the highest priority in that security group will be applied first.
- To create a new security group, click **Create Security Group**.

NOTE

Using multiple security groups may impact the network performance. Selecting more than five security groups is not recommended.

Figure 3-150 Managing security groups



Step 3 Click **OK**.

----End

3.15.4 Performing a Server-Side Encryption

Introduction

The RDS console provides server-side encryption with Data Encryption Workshop (DEW)-managed keys.

DEW uses a third-party hardware security module (HSM) to protect keys, enabling you to easily create and control encryption keys. For security reasons, keys are not displayed in plaintext outside of HSMs. With DEW, all operations on keys are controlled and logged, and usage records of all keys can be provided to meet regulatory compliance requirements.

If server-side encryption is enabled, disk data will be encrypted and stored on the server when you create a DB instance or expand disk capacity. When downloading encrypted objects, the encrypted data will be decrypted on the server and displayed to you in plaintext.

Encrypting Disks Using Server-Side Encryption

For server-side encryption, you need to first create a key using Data Encryption Workshop (DEW) or use the default key that DEW comes with. When creating a DB instance, select **Enable** for disk encryption and select or create a key. The key is the end tenant key and is used for server-side encryption.

- You will need the KMS administrator permission for the region where RDS is deployed. This permission can be granted using Identity and Access Management (IAM). On the IAM console, add permission policies to user groups. For details, see **Creating a User Group and Assigning Permissions**.
- If you want to use a user-defined key to encrypt objects to be uploaded, create a key using DEW. RDS supports only symmetric keys. For details, see **Creating a CMK**.
- If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. Disk encryption will not encrypt backup data stored in OBS. To enable backup data encryption, **submit a service ticket** to request required permissions.
- If disk encryption or backup data encryption is enabled, keep the key properly. Once the key is disabled, deleted, or frozen, the database will be unavailable and data may not be restored.
  - If disk encryption is enabled but backup data encryption is not enabled, you can **restore data to a new instance from backups**.
  - If both disk encryption and backup data encryption are enabled, data cannot be restored.
- If you scale up a DB instance with disks encrypted, the expanded storage space will also be encrypted using the original encryption key.

3.15.5 Using DBSS (Recommended)

Database Security Service (DBSS) is an intelligent database security service. Based on the machine learning mechanism and big data analytics technologies, it can audit your databases, detect SQL injection attacks, and identify high-risk operations.

You are advised to use DBSS to provide extended data security capabilities. For details, see **Database Security Service**.

Advantages

- DBSS can help you meet security compliance requirements.
  - DBSS can help you comply with DJCP (graded protection) standards for database audit.
  - DBSS can help you comply with security laws and regulations, and provide compliance reports that meet data security standards (such as Sarbanes-Oxley).

- DBSS can back up and restore database audit logs and meet the audit data retention requirements.
- DBSS can monitor risks, sessions, session distribution, and SQL distribution in real time.
- DBSS can report alarms for risky behavior and attacks and respond to database attacks in real time.
- DBSS can locate internal violations and improper operations and keep data assets secure.

Deployed in bypass pattern, database audit can perform flexible audits on the database without affecting user services.

- Database audit monitors database logins, operation types (data definition, operation, and control), and operation objects based on risky operations to effectively audit the database.
- Database audit analyzes risks and sessions, and detects SQL injection attempts so you can stay apprised of your database status.
- Database audit provides a report template library to generate daily, weekly, or monthly audit reports according to your configurations. It sends real-time alarm notifications to help you obtain audit reports in a timely manner.

3.16 Parameters

3.16.1 Modifying Parameters of an RDS for PostgreSQL Instance

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.

The following are the key points you should know when using parameters:

- Modifying instance parameters: When you modify dynamic parameters on the **Parameters** page of a target DB instance and save the modifications, the modifications take effect immediately regardless of the **Effective upon Reboot** setting. When you modify static parameters on the **Parameters** page of a target DB instance and save the modifications, the modifications take effect only after you manually reboot the target DB instance.
- Modifying parameter template parameters: When you modify parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications take effect only after you apply the parameter template to DB instances. When you modify static parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications take effect only after you apply the parameter template to DB instances and manually reboot the DB instances. For operation details, see **Applying a Parameter Template**.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **Instances** 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 **Viewing Parameter Change History**.

----End

Modifying a Custom Parameter Template and Applying It to DB Instances

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Parameter Templates** 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 **Change History** to view the modification details.

Step 7 Apply the parameter template to your DB instance. 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

FAQ

Q: Why did my changes to parameters fail to be applied to my DB instance after I rebooted the instance and the instance status remain **Parameter change. Pending reboot**?

A: If you change specification parameters, such as **work_mem**, **shared_buffers**, and **max_connections**, to large values, the instance may fail to be started. To ensure that the database runs properly, the system automatically rolls back the parameter change when the database startup fails. Check whether the new values you set are within the allowed ranges. If you do need to set specification parameters to values beyond those ranges, upgrade the instance class first. For details about how to change an instance class, see **Changing a DB Instance Class**.

Common Parameters

The modifications of some kernel parameters can be applied only after the instance is rebooted. After you modify the parameters on the console, the message "Parameter change. Pending reboot" is displayed.

Figure 3-151 Modifying parameters

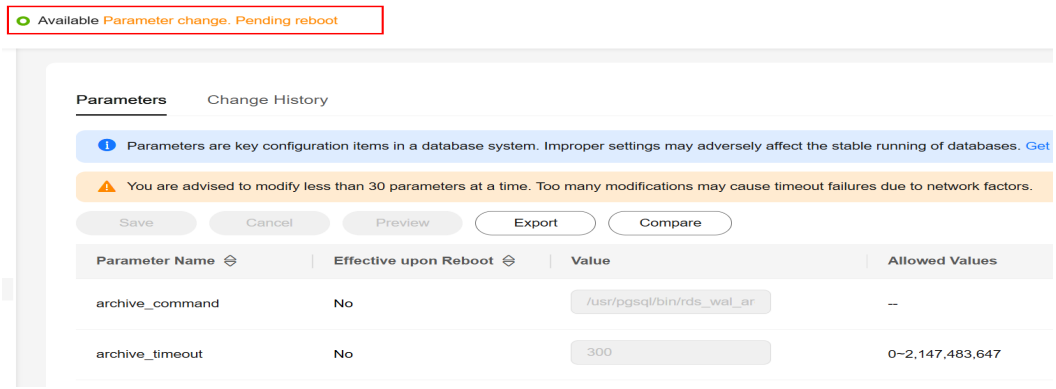


Table 3-56 Common parameters

Parameter	Description	Reference
timezone	The time zone for displaying and interpreting time stamps.	**How Can I Change the Time Zone?**
wal_level	The level of information written to the WAL. This parameter is always set to **logical** for read replicas.	**Does RDS for PostgreSQL Support the test_decoding Plugin?**
max_connections	The maximum number of concurrent connections.	**Instance Usage Suggestions**

3.16.2 Managing Parameter Templates

3.16.2.1 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.

If you create a DB instance without specifying a custom DB parameter template, a default parameter template is used. 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 **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 **Replicating a Parameter Template**.

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

- When you change a parameter value in a parameter template that has been applied to a DB instance and save the change, the change takes effect only to the DB instance and does not affect other DB instances.
- The changes to parameter values in a custom parameter template take effect only after you apply the template to DB instances. For details, see **Applying a Parameter Template**.
- When you change dynamic parameter values in parameter templates in batches and save the changes, the changes will take effect only after you apply the parameter templates to DB instances. When you change static parameter values in parameter templates in batches and save the changes, the changes will take effect for DB instances only after you apply the parameter templates to DB instances and manually reboot the DB instances.
- 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.

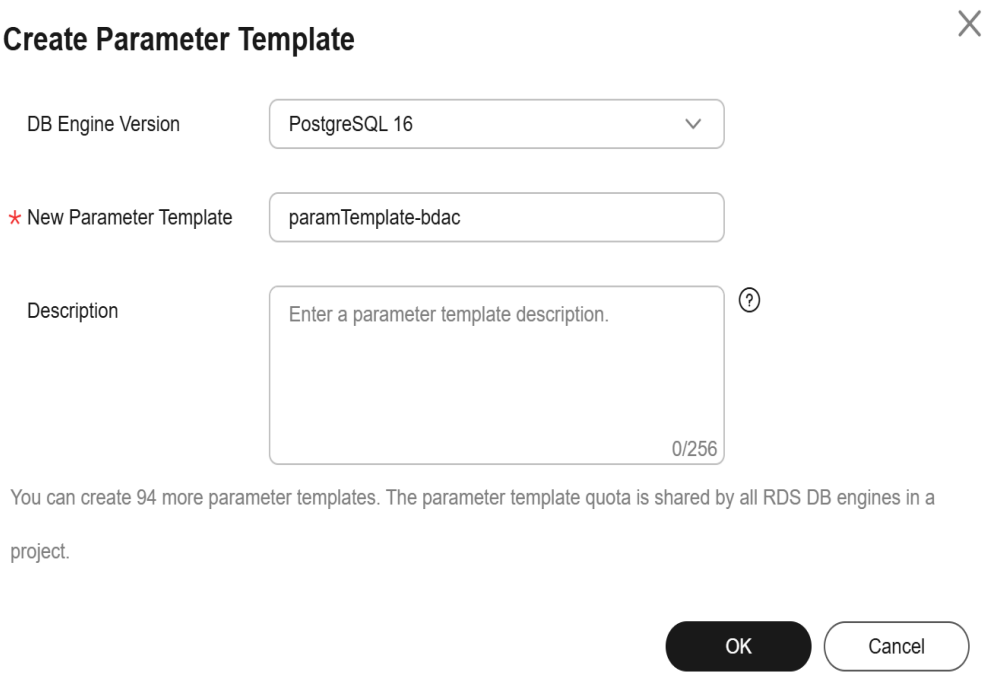
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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: >!<"&'=

Figure 3-152 Creating a parameter template



----End

3.16.2.2 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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 **Viewing Application Records of a Parameter Template**.

----End

3.16.2.3 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**.

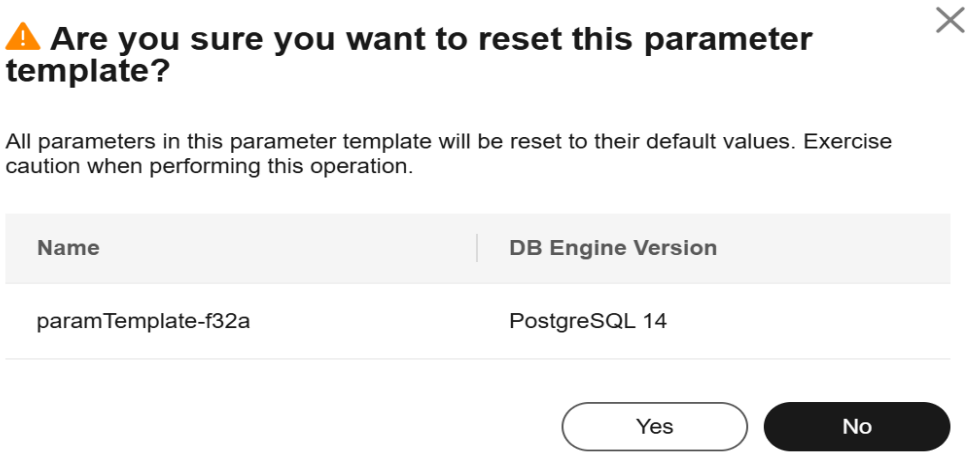
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

Figure 3-153 Confirming the reset



Step 6 Apply the parameter template to your DB instance. 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

3.16.2.4 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. You can also export the parameter template to generate a new parameter template for future use.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

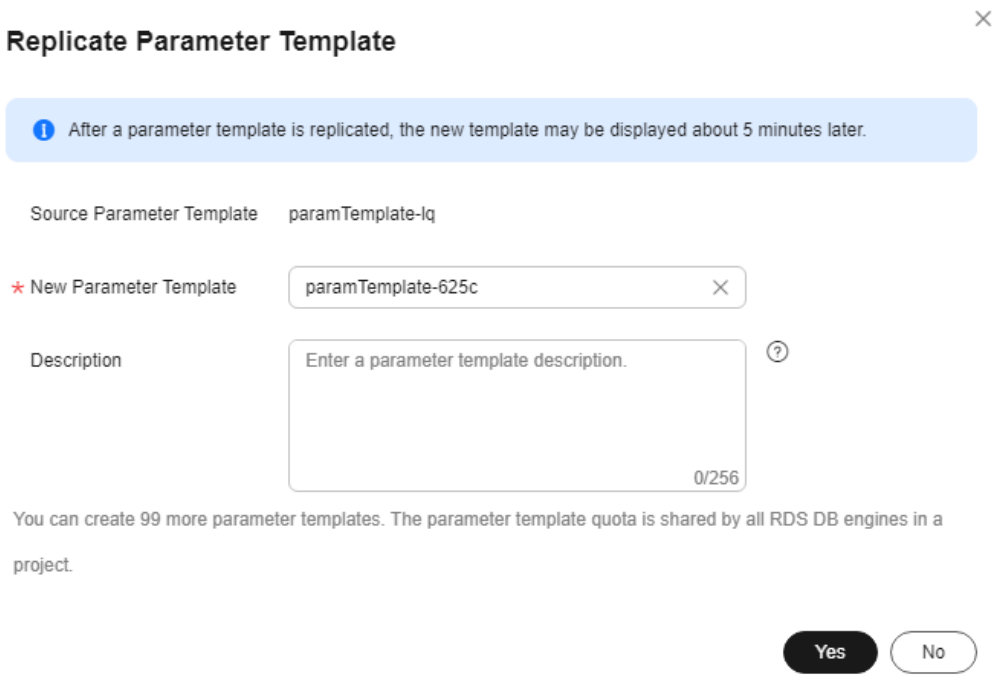
Alternatively, click the target DB instance on the **Instances** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

NOTE

To ensure that your parameter templates are applicable to all types of DB instances and databases can be started normally, the values of **innodb_flush_log_at_trx_commit** and **sync_binlog** exported from primary DB instances or read replicas are **1** by default.

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

Figure 3-154 Replicating a 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: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Templates** page.

----End

3.16.2.5 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.

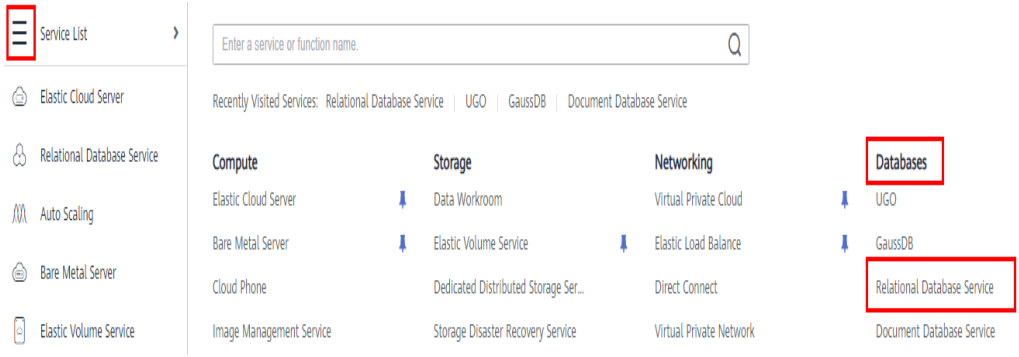
Comparing Instance Parameters with a Parameter Template

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

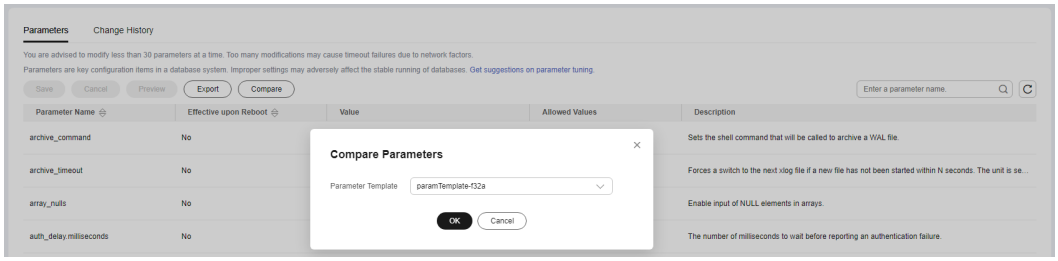
Figure 3-155 Relational Database Service



Step 4 On the **Instances** page, click the instance name.

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

Figure 3-156 Comparing instance parameters with those in a specified parameter template



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

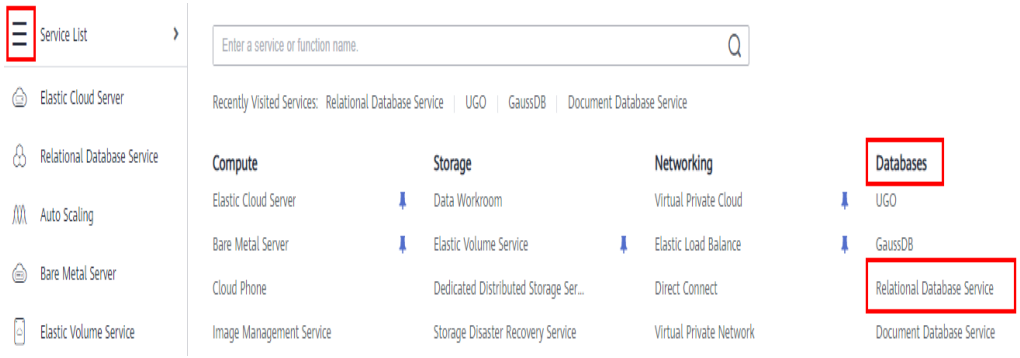
Comparing Parameter Templates

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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

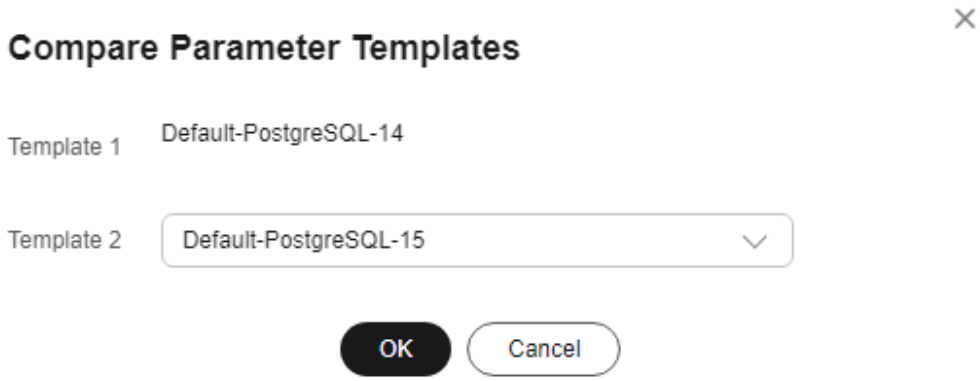
Figure 3-157 Relational Database Service



Step 4 On the **Parameter Templates** 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**.

Figure 3-158 Selecting a parameter template to be compared



- 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

3.16.2.6 Importing a Parameter Template

Scenarios

RDS allows you to import new parameter templates for future use. To apply an imported parameter template to new DB instances, see **Applying a Parameter Template**.

Constraints

- Any modification to read-only parameters in an imported parameter template does not take effect.
- Only parameter templates that were exported from the **Parameter Templates** page on the RDS console can be imported.
- If any modification to an exported parameter template causes a change in the file format, the template may not be able to be imported.
- The parameter template to be imported cannot contain parameters related to specifications. For details about such parameters, see **Constraints**.

Procedure

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

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

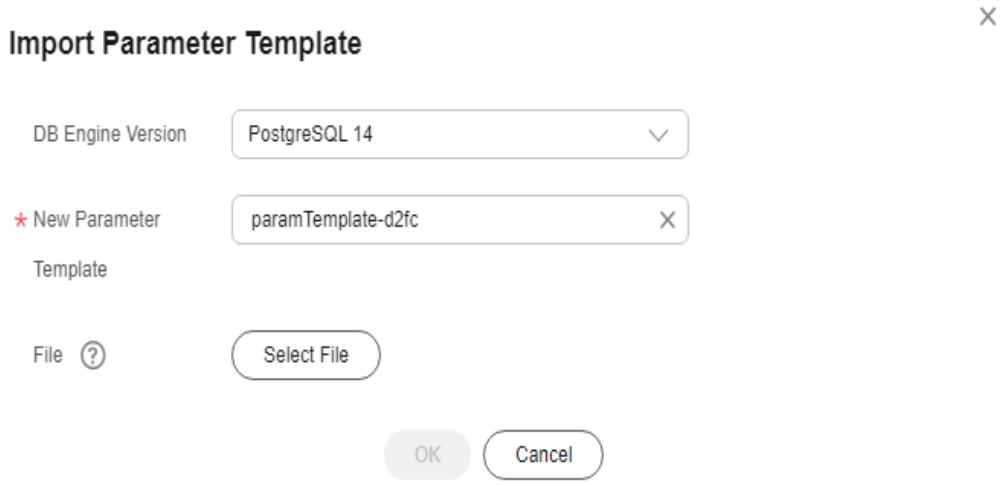
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Import Parameter Template**.

Step 5 In the displayed dialog box, click **Select File**, import the target parameter list (containing parameter names, values, and description), and click **OK**.

Only one file (CSV format) can be imported at a time. The file size cannot exceed 50 KB.

Figure 3-159 Importing a parameter template



----End

3.16.2.7 Exporting a Parameter Template

Scenarios

Exporting instance parameters

- You can export parameters of a DB instance as a new parameter template for future use. To apply the exported parameter template to new DB instances, see **Applying a Parameter Template**.
- You can also export the parameter information (including parameter names, values, and descriptions) of a DB instance to a CSV file for viewing and analyzing details.

Exporting a parameter template

- You can export an RDS for PostgreSQL parameter template (including parameter names, values, and descriptions) to a CSV file for viewing and analyzing details.

Exporting Instance Parameters

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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 in the **Custom Templates** tab on the **Parameter Templates** page.

- Exporting to a file

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

  NOTE

  The file name can contain 4 to 81 characters.

----End

Exporting a Parameter Template

Step 1 Log in to the management console.

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

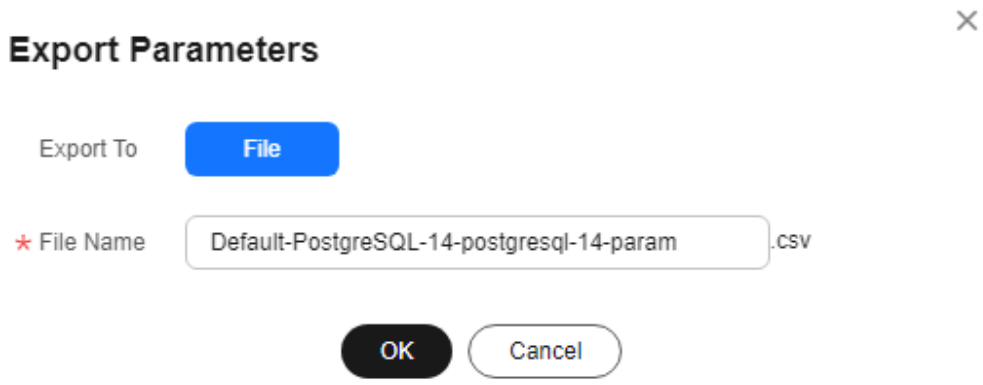
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Default Templates** or **Custom Templates** as needed. On the displayed page, locate the target template and choose **More** > **Export** in the **Operation** column.

Step 5 In the displayed dialog box, enter a file name and click **OK**.

The file name can contain 4 to 81 characters.

Figure 3-160 Exporting a parameter template



----End

3.16.2.8 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or 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

3.16.2.9 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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

3.16.2.10 Viewing Parameter Change History

Scenarios

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

NOTE

The change history for an exported or custom parameter template is initially blank.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

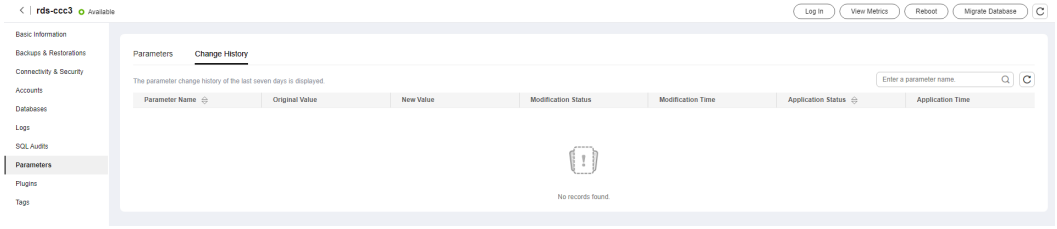
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

Figure 3-161 Viewing parameter change history



You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

----End

Viewing Change History of a Parameter Template

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

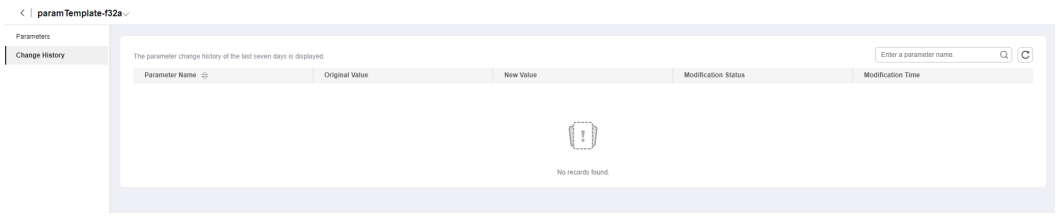
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Parameter Templates** 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.

Figure 3-162 Viewing parameter change history



You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

3.16.2.11 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.

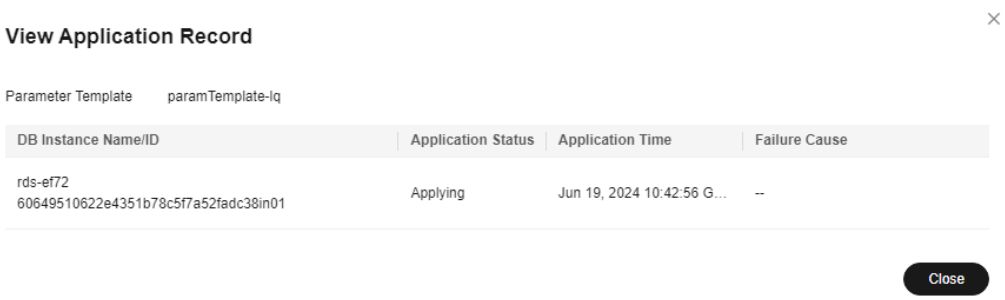
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and 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).

Figure 3-163 Viewing application records of a parameter template



----End

3.16.3 Suggestions on RDS for PostgreSQL Parameter Tuning

Parameters are key configuration items in a database system. Improper parameter settings may adversely affect database performance. This section describes some important parameters for your reference. For details, visit the **PostgreSQL official website**.

For details on how to modify RDS for PostgreSQL parameters on the console, see **Modifying Parameters of an RDS for PostgreSQL Instance**.

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. For details, see **Viewing and Downloading Slow Query Logs**.
- The **temp_file_limit** parameter limits the total size (in KB) of all temporary files when writing temporary files to the disk is triggered in a session. The value ranges from -1 to 2,147,483,647. The value **-1** indicates that the total size of the temporary files is not limited.
  - This parameter is only available to RDS for PostgreSQL 11, 12, 13, 14, and 15.
  - To prevent temporary files from occupying too much disk space and causing service exceptions, do not set this parameter to **-1**.
  - 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.
- The **max_pred_locks_per_transaction** and **max_locks_per_transaction** parameters need to be set based on the values of **max_connections** and **max_prepared_transactions**. Too large values may cause instance unavailability.

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.
- The **shared_buffers** parameter is recommended to be a value ranging from 25% to 40% of the system memory. The maximum value of this parameter cannot exceed 80% of the system memory to avoid affecting database performance.
- The **max_worker_processes** parameter should be set based on the values of **max_parallel_workers** and **max_parallel_workers_per_gather**. If the **max_worker_processes** value is too large, the database performance will be affected.

3.17 Log Management

3.17.1 Log Reporting

Scenarios

If you enable log reporting for your DB instance, new logs generated for the instance will be uploaded to Log Tank Service (LTS) for management.

Precautions

- You will be billed for enabling this function under LTS.
- Ensure that there are available LTS log groups and log streams in the same region as your instance.

  For more information about log groups and log streams, see **Log Management**.

Constraints

- Error logs and slow query logs cannot share the same log stream.
- If a structuring template (PostgreSQL slow log template or PostgreSQL error log template) has been bound to a log stream, ensure that the template type is the same as the log type when you select the log stream. For example, if a PostgreSQL error log template has been bound to a log stream, the log stream cannot be used for slow query logs.

  For details about how to bind a system template to a log stream, see **Log Structuring**.

Enabling Log Reporting in Batches

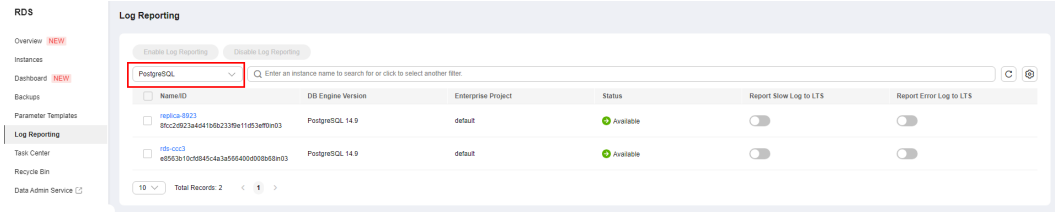
Step 1 **Log in to the management console**.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane, click **Log Reporting**. On the displayed page, select **PostgreSQL** from the drop-down list.

Figure 3-164 Log reporting



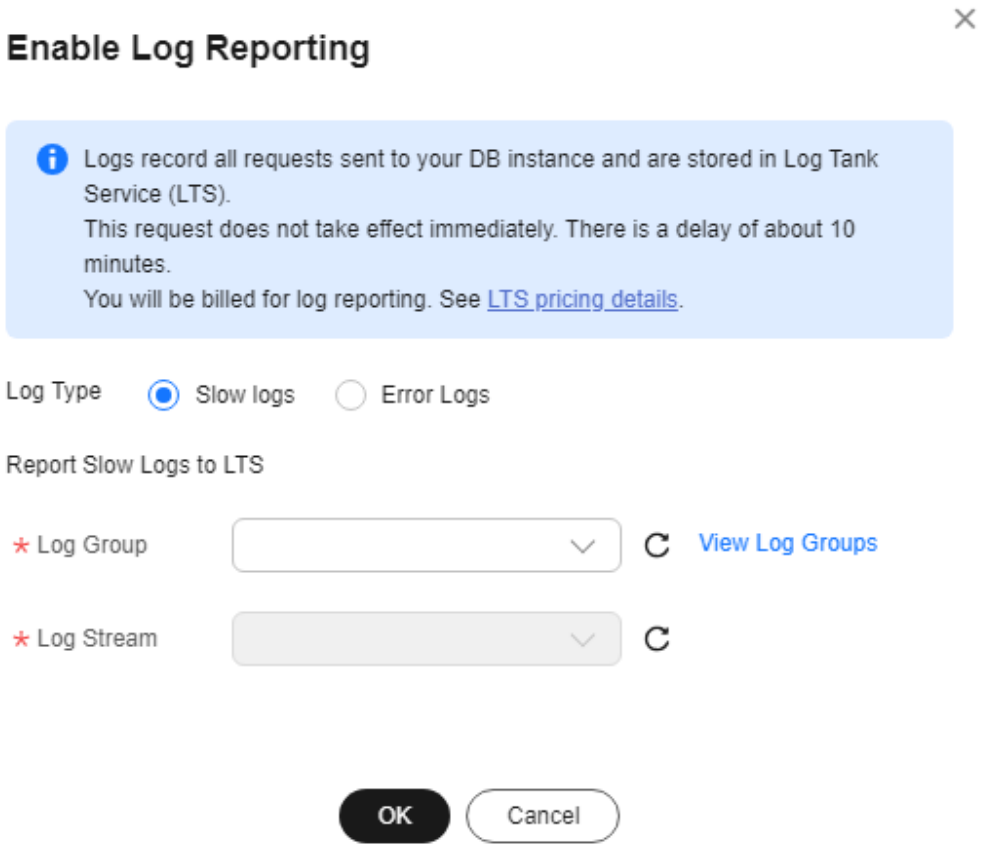
Step 5 Select one or more instances and click **Enable Log Reporting**.

Step 6 Select an LTS log group and log stream and click **OK**.

NOTE

- Error logs and slow query logs cannot share the same log stream.
- This request does not take effect immediately. There is a delay of about 10 minutes.

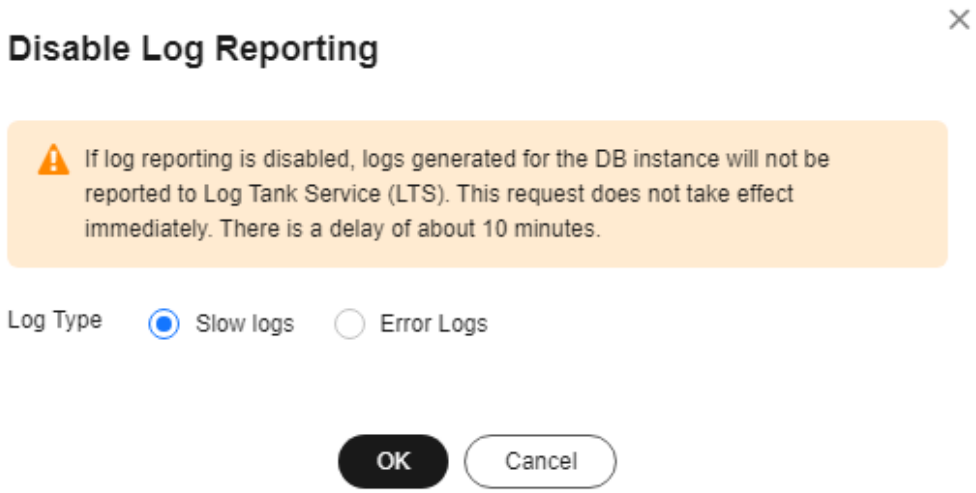
Figure 3-165 Enabling log reporting



Step 7 To disable log reporting, select one or more instances and click **Disable Log Reporting**.

Step 8 In the displayed dialog box, click **OK**.

Figure 3-166 Disabling log reporting



----End

3.17.2 Viewing and Downloading Error Logs

Scenarios

Error logs contain logs generated while the database is running. These can help you analyze problems with the database. You can also download error logs for service analysis.

Viewing Log Details

Step 1 Log in to the management console.

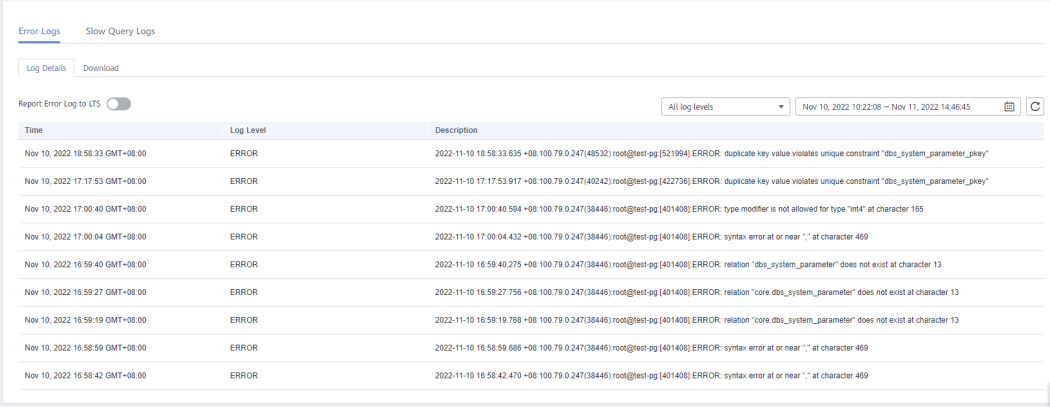
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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.

Figure 3-167 Error log details



- You can select a log level in the upper right corner of the log list.

  NOTE

  For RDS for PostgreSQL DB instances, the following levels of logs are displayed:
  - All log levels
  - ERROR
  - FATAL
  - PANIC

- 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.
- Currently, a maximum of 2,000 error log records can be displayed.

----End

Downloading a Log

Step 1 Log in to the management console.

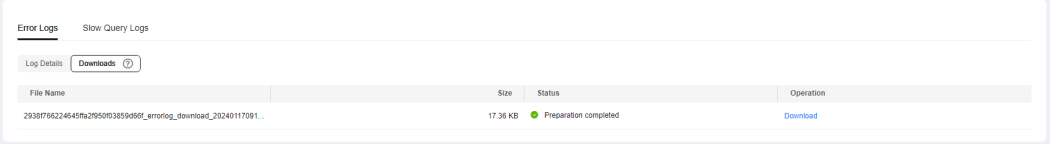
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Error Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

Figure 3-168 Downloading an error log



- It is recommended that a single log file to be downloaded contain a maximum of 10,000 lines and the file size be no more than 10 MB. Otherwise, the log information will be truncated.
- 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.
- Only the latest log file of 40 MB to 100 MB can 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 redownload the log, click **OK**.

----End

Enabling Error Log Reporting to LTS

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

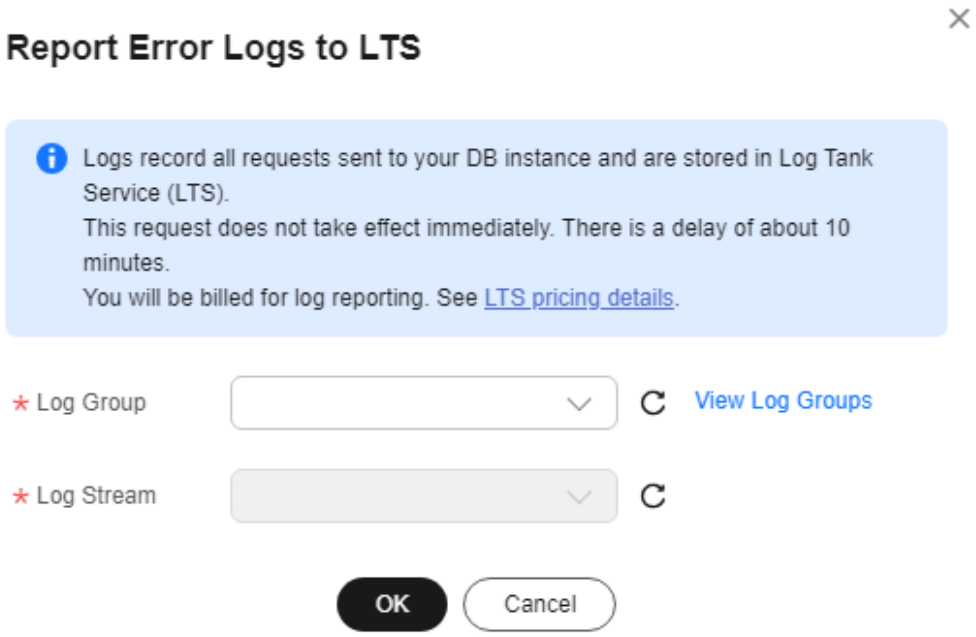
Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane, click **Logs**. On the **Error Logs** page, click **Log Details**.

Step 6 Click  next to **Report Error Logs to LTS**.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 3-169 Enabling error log reporting to LTS



----End

3.17.3 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed the **log_min_duration_statement** value. 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:

- All statement types
- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- ALTER
- DO
- CALL
- COPY

Parameter Description

Table 3-57 Parameters related to RDS for PostgreSQL slow queries

Parameter	Description
log_min_duration_statement	Specifies how many milliseconds a query has to run before it has to be logged. If this parameter is set to a smaller value, the number of log records increases, which increases the disk I/O and deteriorates the SQL performance.
log_statement	Specifies the statement type. The value can be **none**, **ddl**, **mod**, or **all**. The default value is **none**. If you change the value to **all**: • The database disk I/O increases, and the SQL performance deteriorates. • The log format changes, and you cannot view slow query logs on the console.
log_statement_stats	Specifies whether to generate performance statistics to server logs. The default value is **off**. If you change the value to **on**: • The database disk I/O increases, and the SQL performance deteriorates. • The log format changes, and you cannot view slow query logs on the console.

Showing Original Logs

NOTE

- To disable **Show Original Logs** on the console, contact customer service. If you want to disable **Show Original Logs** by calling an API, see **Showing Original Logs**.
- Original logs will be automatically deleted 30 days later. If the instance is deleted, its logs are also deleted.
- This function takes effect only for slow query logs generated after it is enabled. Historical slow query logs generated before the function is enabled are not displayed in plaintext.

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

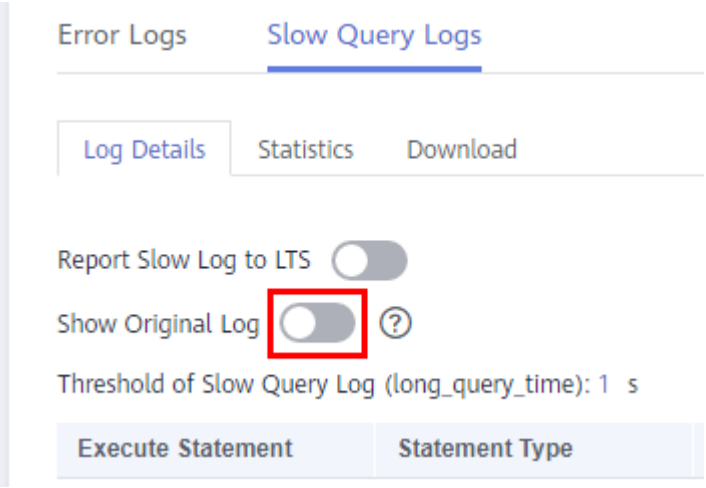
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Log Details** and then click on the right of **Show Original Logs**.

Figure 3-170 Enabling Show Original Logs



Step 6 In the displayed dialog box, click **Yes** to enable the display of original slow query logs.

----End

Viewing Log Details

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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 **log_min_duration_statement** parameter determines when a slow query log is recorded. However, changes to this parameter do not affect already recorded logs. If **log_min_duration_statement** is changed from 1,000 ms to 100 ms, RDS starts recording statements that meet the new threshold and still displays the previously recorded logs that do not meet the new threshold. For example, a 1,500 ms SQL statement that was recorded when the threshold was 1,000 ms will not be deleted now that the new threshold is 2,000 ms.
- Currently, a maximum of 2,000 slow log records can be displayed.

----End

Viewing Statistics

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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.

----End

Downloading a Log

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click **Downloads**. In the log list, locate a log whose status is **Preparation completed** and click **Download** in the **Operation** column.

- It is recommended that a single log file to be downloaded contain a maximum of 10,000 lines and the file size be no more than 10 MB. Otherwise, the log information will be truncated.
- 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.
- Only the latest log file of 40 MB to 100 MB can 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 redownload the log, click **OK**.

----End

Enabling Slow Log Reporting to LTS

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

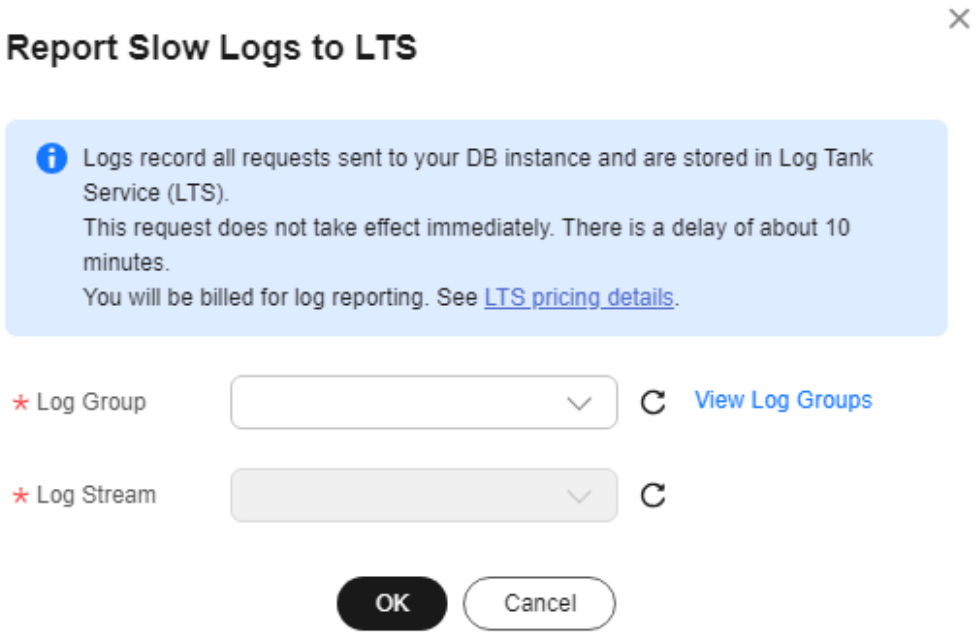
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, click **Logs**. On the **Slow Query Logs** page, click **Log Details**.

Step 6 Click next to **Report Slow Logs to LTS**.

Step 7 Select an LTS log group and log stream and click **OK**.

Figure 3-171 Enabling slow log reporting to LTS



----End

3.17.4 Enabling SQL Audit

Scenarios

After SQL audit is enabled for RDS for PostgreSQL DB instances, the system records SQL operations and uploads logs every half an hour or when the size of a single record reaches 100 MB. The generated audit logs are stored in OBS. If there is not enough free backup space available for generated audit logs, the additional space required is billed.

Precautions

- SQL audit is disabled for DB instances by default because enabling it increases database loads.
- To ensure good performance, SQL audit uses the Coordinated Universal Time (UTC) format and is not affected by the time zone configuration.
- To enable SQL audit, you need to install the pgAudit extension first. For details, see **pgAudit**.

Constraints

Only the following versions support SQL audit. To use this function, **submit a service ticket** to apply for required permissions. If your DB engine version is too early, upgrade it to the latest version by referring to **Upgrading a Minor Version**.

- Latest minor versions of RDS for PostgreSQL 12 and 13
- All versions of RDS for PostgreSQL 14 and above

Performance

The pgAudit impact on database performance depends on how many audit logs there are and how often they are generated. More audit logs mean more impact on the database performance. pgAudit can decrease the database performance by about 20%. You need to configure parameters for pgAudit based on your workloads to achieve a balance between audit requirements and database performance.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

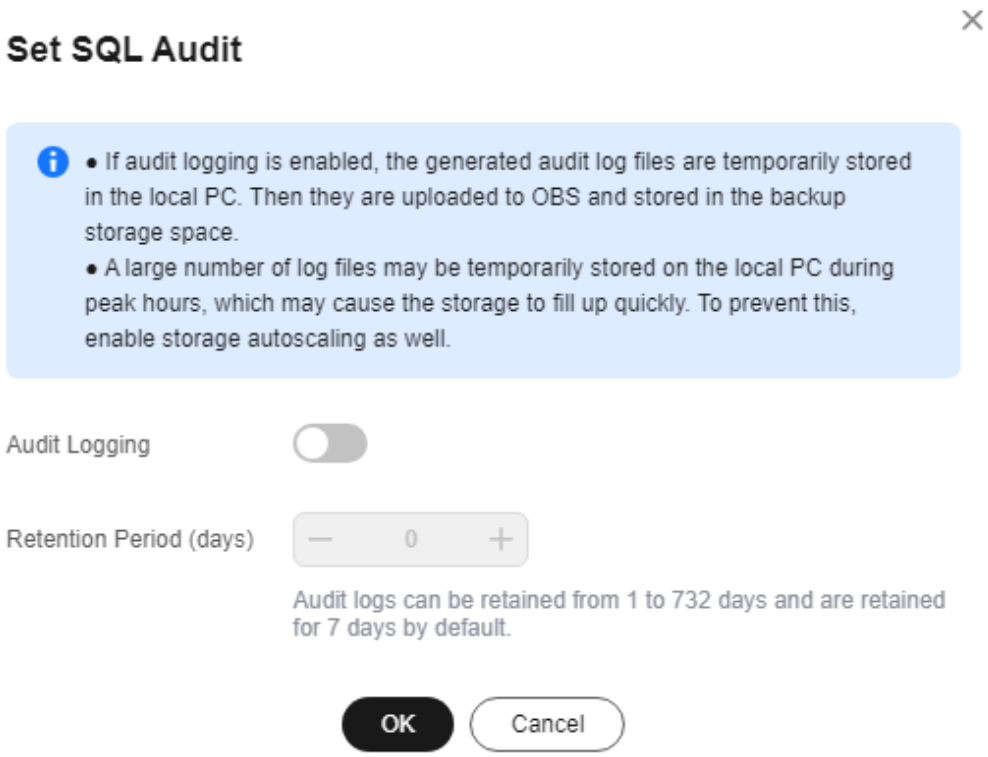
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **SQL Audits**. On the displayed page, click **Set SQL Audit**.

Step 6 In the displayed dialog box, set the number of days for storing SQL audit logs and click **OK**.

Audit logs can be retained from 1 to 732 days and are retained for 7 days by default.

Figure 3-172 Setting SQL audit

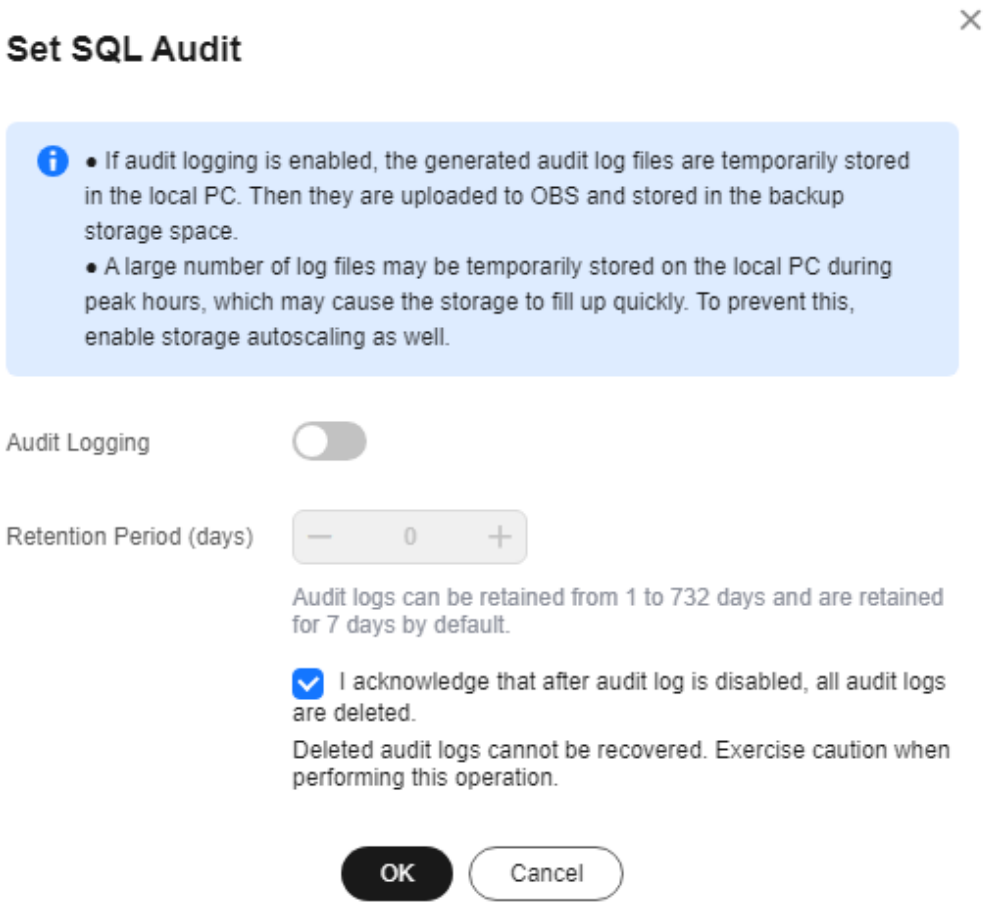


Step 7 To disable SQL audit, toggle off the **Audit Logging** switch, select the confirmation check box, and click **OK**.

NOTICE

After SQL audit is disabled, all audit logs will be deleted immediately and cannot be recovered. Exercise caution when performing this operation.

Figure 3-173 Disabling SQL audit



----End

3.17.5 Downloading SQL Audit Logs

If you **enable SQL audit**, all SQL operations will be logged, and you can download audit logs to view details. The minimum time unit of audit logs is second.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

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 3-58**.

Figure 3-174 RDS for PostgreSQL audit logs

```
AUDIT: CLIENT_AUTHENTICATION,SUCCESS,,,59160,1699934229671595,0,,root,postgres,192.168.57.136,,,,,
AUDIT: SESSION,1,1,DDL,59160,1699934246188832,0,psql,root,postgres,192.168.57.136,CREATE TABLE,TABLE,public.c,create table c (id int);,<not logged>
AUDIT: SESSION,2,1,DDL,59160,1699934285890419,0,psql,root,postgres,192.168.57.136,CREATE TABLE,TABLE,public.d,create table d (id int);,<not logged>
AUDIT: SESSION,3,1,READ,59160,1699934324751446,0,psql,root,postgres,192.168.57.136,SELECT,,,select * from a ;,<not logged>
AUDIT: SESSION,4,1,WRITE,59160,1699934342491524,0,psql,root,postgres,192.168.57.136,INSERT,,,insert into a values (1);,<not logged>
AUDIT: SESSION,5,1,WRITE,59160,1699934348652631,0,psql,root,postgres,192.168.57.136,INSERT,,,insert into a values (2);,<not logged>
AUDIT: SESSION,6,1,READ,59160,1699934357849825,0,psql,root,postgres,192.168.57.136,SELECT,,,select * from a ;,<not logged>
AUDIT: SESSION,7,1,READ,59160,1699934371993604,0,psql,root,postgres,192.168.57.136,SELECT,,,select * from a limit 1;,<not logged>
```

Table 3-58 Audit log field description

Field	Description
AUDIT:	Fixed prefix, which identifies an audit record.
AUDIT_TYPE	Audit type. The value can be **SESSION**, **OBJECT**, or **CLIENT_AUTHENTICATION**.
STATEMENT_ID	Unique statement ID for this session.
SUBSTATEMENT_ID	ID of each substatement in the main statement.
CLASS or AUTHENTICATION_RESULT	Operation type. • **CLASS**: The value depends on the **pgaudit.log** options, and can be **READ** or **ROLE**. • **AUTHENTICATION_RESULT**: The value can be **SUCCESS** or **FAIL**.
PID	Process ID.
STATEMENT_START_TIME	Statement start timestamp, in us.
connection_status	Session status, which is usually the returned error code of a statement. If the statement is successfully executed, the value **0** is returned.
APPLICATION_NAME	Application name, such as **PSQL** and **JDBC**.
USER_NAME	Username for logging in to the database.
DATABASE_NAME	Name of the database that was logged in to.
REMOTE_HOST	IP address of the host used for login.
COMMAND	Type of the SQL command, such as **ALTER TABLE** and **SELECT**.
OBJECT_TYPE	Object type, such as **TABLE**, **INDEX**, and **VIEW**.
OBJECT_NAME	Object name.
STATEMENT	Content of the SQL statement executed at the backend.

Field	Description
PARAMETER	Parameter value.

----End

3.18 Metrics and Alarms

3.18.1 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 3-59 lists the performance metrics of RDS for PostgreSQL DB instances.

Table 3-59 Performance metrics

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100	%	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0–100	%	N/A	RDS for PostgreSQL instance	1 minute
rds003_iops	IOPS	Average number of I/O requests processed by the system in a specified period	≥ 0	counts/s	N/A	RDS for PostgreSQL instance	1 minute
read_count_per_second	Read IOPS	Average number of read I/O requests processed by the system in a specified period	≥ 0	counts/s	N/A	RDS for PostgreSQL instance	1 minute
write_count_per_second	Write IOPS	Average number of write I/O requests processed by the system in a specified period	≥ 0	counts/s	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0	bytes/s	1024	RDS for PostgreSQL instance	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0	bytes/s	1024	RDS for PostgreSQL instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0–100	%	N/A	RDS for PostgreSQL instance	1 minute
rds040_transaction_logs_usage	Transaction Logs Usage	Storage space usage of transaction logs	≥ 0	MB	1024	RDS for PostgreSQL instance	1 minute
rds041_replication_slot_usage	Replication Slot Usage	Storage space usage of replication slot files	≥ 0	MB	1024	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds042_database_connections	Database Connections in Use	Number of database connections in use	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute
rds043_maximum_used_transaction_ids	Maximum Used Transaction IDs	Maximum number of transaction IDs that have been used	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute
rds044_transaction_logs_generations	Transaction Logs Generation	Size of transaction logs generated per second	≥ 0	MB/s	1024	RDS for 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	1024	RDS for PostgreSQL instance	1 minute
rds046_replication_lag	Replication Lag	Replication lag	≥ 0	ms	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4000 If you want to create a DB instance with storage up to 15,000 GB, **submit a service ticket** to apply for the required permission.	GB	1024	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4000 If you want to create a DB instance with storage up to 15,000 GB, contact customer service to apply for the required permission.	GB	1024	RDS for PostgreSQL instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0	bytes/s	1024	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0	bytes/s	1024	RDS for PostgreSQL instance	1 minute
rds082_tps	TPS	Execution times of submitted and rollback transactions per second	≥ 0	counts/s	N/A	RDS for PostgreSQL instance	1 minute
rds083_conn_usage	Connection Usage	Percent of used PostgreSQL connections to the total number of connections	0–100	%	N/A	RDS for PostgreSQL instance	1 minute
row_per_second	Operation Rows	Number of rows that are being inserted, deleted, updated, or queried	≥ 0	rows/s	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
active_connections	Active Connections	Number of active database connections	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute
idle_transaction_connections	Idle Transaction Connections	Number of idle transaction connections	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
oldest_transaction_duration	Oldest Active Transaction Duration	Length of time since the start of the transaction that has been active longer than any other current transaction	≥ 0	ms	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute
oldest_transaction_duration_2pc	Oldest Two-Phase Commit Transaction Duration	Length of time since the start of the transaction that has been prepared for two-phase commit longer than any other current transaction	≥ 0	ms	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
disk_io_usage	Disk I/O Usage	I/O usage of disks The disk I/O usage is the percentage of the time that the disk processes I/O requests to the total time.	0–100	%	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
		NOTE If the disk I/O usage reaches 100%, data is being written to the disk during the statistical period. The disk performance is determined by multiple metrics, such as IOPS, disk throughput, and read/write latency.					
lock_waiting_sessions	Sessions Waiting for Locks	Number of blocked sessions	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
swap_in_rate	Swap-In Rate	Volume of data written from the swap partition to the memory per second	≥ 0	KB/s	1024	RDS for PostgreSQL instance	1 minute
swap_out_rate	Swap-Out Rate	Volume of data written from the memory to the swap partition per second	≥ 0	KB/s	1024	RDS for PostgreSQL instance	1 minute
swap_total_size	Total Swap Size	Total size of the swap partition	≥ 0	MB	1024	RDS for PostgreSQL instance	1 minute
swap_usage	Swap Usage	Usage of the swap partition	0–100	%	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
db_max_age	Maximum Database Age	Maximum age of the current database, which is the value of **max(age(datfrozenxid))** in the **pg_database** table	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute
cpu_user_usage	User-mode CPU Time Percentage	Percentage of time that the CPU is in user mode	0–100	%	N/A	RDS for PostgreSQL instance	1 minute
cpu_sys_usage	Kernel-mode CPU Time Percentage	Percentage of time that the CPU is in kernel mode	0–100	%	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
cpu_wait_usage	Disk I/O Wait Time Percentage	Percentage of time that the CPU is waiting for disk I/O operations to complete	0–100	%	N/A	RDS for PostgreSQL instance	1 minute
io_read_delay	Read I/O Latency	Average latency (in milliseconds) of disks responding to read requests	≥ 0	ms	N/A	RDS for PostgreSQL instance	1 minute
io_write_delay	Write I/O Latency	Average latency (in milliseconds) of disks responding to write requests	≥ 0	ms	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
slow_sql_one_second	Number of SQL Statements Executed for More Than 1s	Number of slow SQL statements whose execution time is longer than 1s This metric shows an instantaneous value at the collection time instead of an accumulated value within 1 minute.	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
slow_sql_three_second	Number of SQL Statements Executed for More Than 3s	Number of slow SQL statements whose execution time is longer than 3s This metric shows an instantaneous value at the collection time instead of an accumulated value within 1 minute.	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
slow_sql_five_second	Number of SQL Statements Executed for More Than 5s	Number of slow SQL statements whose execution time is longer than 5s This metric shows an instantaneous value at the collection time instead of an accumulated value within 1 minute.	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
slow_sql_log_min_duration_statement	Number of SQL Statements Executed for More Than log_min_duration_statement	Number of slow SQL statements whose execution time is longer than the value of log_min_duration_statement. You can change the value of this metric as required. This metric shows an instantaneous value at the collection time instead of an accumulated value within 1 minute.	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 10 and later versions are supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
pg_dr_repl_stat	Replication Status Between Primary DB Instance and DR Instance	Replication status between the primary DB instance and DR instance. The value ranges from 0 to 5. • 0: abnormal • 1: startup • 2: catchup • 3: streaming • 4: backup • 5: stopping	≥ 0	counts	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
pg_dr_wal_delay	LSN Latency Between Primary DB Instance and DR Instance	Latency between the LSN of the primary DB instance and the replay LSN of the DR instance	≥ 0	bytes/s	1024	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute
round_trip_time	Network Latency Between Primary DB Instance and DR Instance	RTT between the primary DB instance and DR instance	≥ 0	ms	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
packet_loss_rate	Packet Loss Rate Between Primary DB Instance and DR Instance	Packet loss rate between the primary DB instance and DR instance	0–100	%	N/A	RDS for PostgreSQL instance **NOTE** Only RDS for PostgreSQL 12 is supported.	1 minute
inactive_logical_replication_slot	Inactive Logical Replication Slots	Number of inactive logical replication slots	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute
pgaudit_log_size	Audit Log Size	Size of audit logs	≥ 0	GB	1024	RDS for PostgreSQL instance	5 minutes

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
slave_replication_status	Stream Replication Status of Standby Node	Stream replication status of the standby node. The value **0** indicates abnormal stream replication; **1** indicates normal stream replication; and **2** means that this node is the primary node. For this metric, the standby node also includes read replicas.	• 0 • 1 • 2	counts	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
synchronous_replication_blocking_time	Synchronous Replication Blocking Time	Time during which synchronous replication between the primary and standby nodes is blocked	≥ 0	s	N/A	RDS for PostgreSQL instance	1 minute
temporary_files_generation_num	Temporary Files per Minute	Number of temporary files generated within 1 minute	≥ 0	counts/min	N/A	RDS for PostgreSQL instance	1 minute
temporary_files_generation_size	Temporary File Size per Minute	Size of temporary files generated within 1 minute	≥ 0	bytes/min	1024	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
sent_lsn_replication_latency_size	Size of Not Sent WAL	Size of WAL logs that have not been sent from the primary node to the standby node	≥ 0	bytes	1024	RDS for PostgreSQL read replica	1 minute
write_lsn_replication_latency_size	Size of Not Written WAL	Size of WAL logs that have not been written to the disk by the standby node	≥ 0	bytes	1024	RDS for PostgreSQL read replica	1 minute
flush_lsn_replication_latency_size	Size of Not Flushed WAL	Size of WAL logs that have not been flushed to the disk by the standby node	≥ 0	bytes	1024	RDS for PostgreSQL read replica	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
replay_lsn_replication_latency_size	Size of Not Replayed WAL	Size of WAL logs that have not been replayed by the standby node	≥ 0	bytes	1024	RDS for PostgreSQL read replica	1 minute
data_disk_inode_used	Inodes	Used data disk inodes	≥ 0	counts	N/A	RDS for PostgreSQL instance	5 minutes
user_current_connections	Connections in Use	Number of connections in use (excluding built-in connections used for monitoring and O&M)	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
user_active_connections	Active Connections of Users	Number of active connections used by users (excluding built-in active connections used for monitoring and O&M)	≥ 0	counts	N/A	RDS for PostgreSQL instance	1 minute
wal_size	WAL Size	Size of WAL logs	≥ 0	GB	1024	RDS for PostgreSQL instance	5 minutes
dbuser_passwd_deadline	The fastest expiration time for database users	Database user password expiration time minus current time **NOTE** If no password expiration time is set, this metric cannot be collected.	≥ 0	s	N/A	RDS for PostgreSQL instance	5 minutes

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
sys_memory_hit_rate	Memory Hit Rate	Memory hit rate	≥ 0	%	N/A	RDS for PostgreSQL instance	1 minute

Dimension

Key	Value
postgresql_cluster_id	RDS for PostgreSQL DB instance ID

3.18.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS metrics on the management console. For details, see **Viewing Metrics of DB Instances**.

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.

Viewing Metrics of DB Instances

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** 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 DB instance.

- On the Cloud Eye console, click **Select Metric** in the upper right corner. In the displayed dialog box, you can select the metrics to be displayed and sort them by dragging them to desired locations.
- You can sort graphs by dragging them based on service requirements.
- You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, 7 days, and 6 months.

----End

3.18.3 Setting Alarm Rules

Scenarios

You can set 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

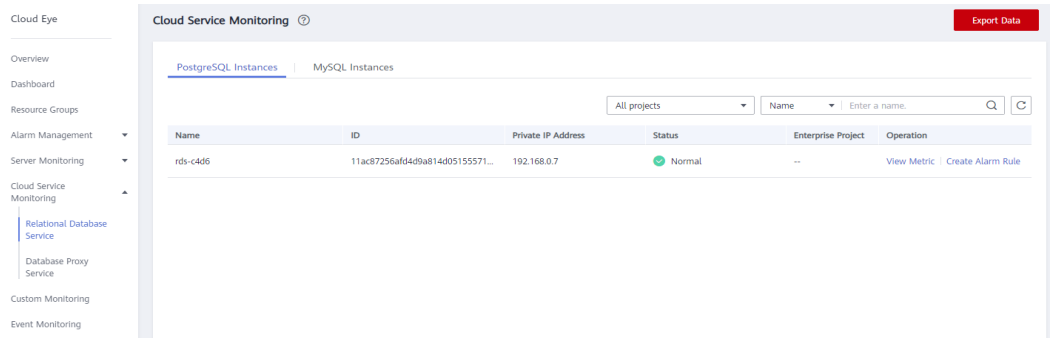
Step 1 Log in to the management console.

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

Step 3 Click **Service List**. Under **Management & Governance**, click **Cloud Eye**.

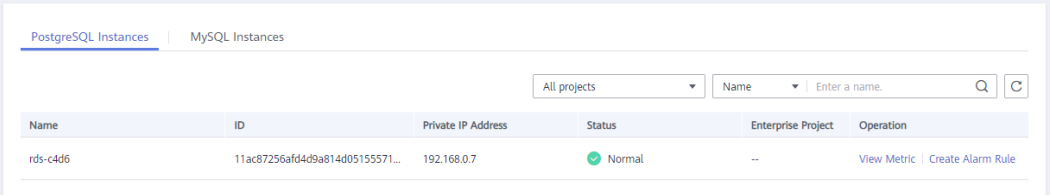
Step 4 In the navigation pane on the left, choose **Cloud Service Monitoring** > **Relational Database Service**.

Figure 3-175 Choosing a monitored object



Step 5 Locate the DB instance for which you want to create an alarm rule and click **Create Alarm Rule** in the **Operation** column.

Figure 3-176 Creating an alarm rule



Step 6 On the displayed page, set parameters as required.

Table 3-60 Alarm rule information

Parameter	Description
Name	Alarm rule name. The system generates a random name, which you can modify.
Description	Description about the rule.
Method	There are three options: **Associate template**, **Use existing template**, and **Configure manually**. **NOTE** If you select **Associate template**, after the associated template is modified, the policies contained in this alarm rule to be created will be modified accordingly. You are advised to select **Use existing template**. The existing templates already contain three common alarm metrics: CPU usage, memory usage, and storage space usage.
Template	Select the template to be used. You can select a default alarm template or create a custom template.

Parameter	Description
Alarm Policy	Policy for triggering an alarm. Whether to trigger an alarm depends on whether the metric data in consecutive periods reaches the threshold. For example, Cloud Eye triggers an alarm if the average CPU usage of the monitored object is 80% or more for three consecutive 5-minute periods. **NOTE** A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm is triggered.
Alarm Severity	The alarm severity can be **Critical**, **Major**, **Minor**, or **Informational**.

Figure 3-177 Configuring alarm notification

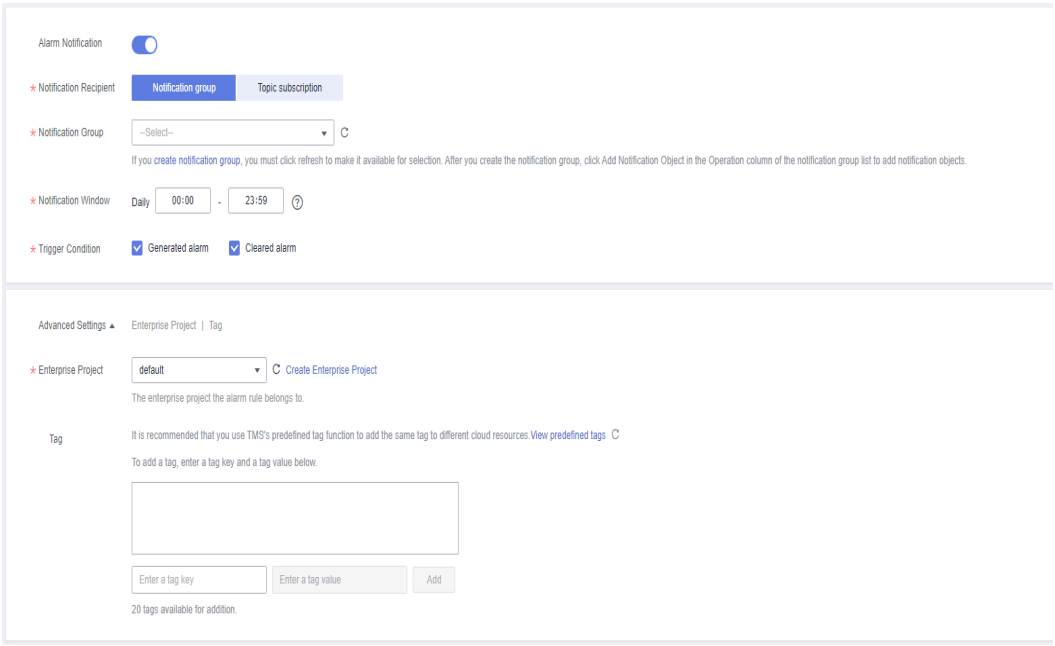


Table 3-61 Alarm notification

Parameter	Description
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Recipient	You can select a notification group or topic subscription as required.
Notification Group	Notification group the alarm notification is to be sent to.

Parameter	Description
Notification Object	Object the alarm notification is to be sent to. You can select the account contact or a topic. • The account contact is the mobile phone number and email address of the registered account. • A topic is used to publish messages and subscribe to notifications.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.
Tag	A tag is a key-value pair. Tags identify cloud resources so that you can easily categorize and search for your resources.

Step 7 Click **Create**. The alarm rule is created.

For details about how to create alarm rules, see **Creating an Alarm Rule** in the *Cloud Eye User Guide*.

----End

3.18.4 Event Monitoring

3.18.4.1 Introduction to Event Monitoring

Event monitoring provides event data reporting, query, and alarm reporting. You can create alarm rules for both system and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on RDS resources that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, for example, resetting the administrator password or modifying the backup policy.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see **Events Supported by Event Monitoring**.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by services to Cloud Eye.

For details about how to report custom events, see **Reporting Events**.

3.18.4.2 Viewing Event Monitoring Data

Scenarios

This section describes how to view the event monitoring data.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using the following method:

On the **Instances** page, click the DB instance name. On the displayed **Overview** page, click **View Metrics** in the upper right corner.

Step 5 Click to return to the main page of Cloud Eye.

Step 6 In the navigation pane on the left, choose **Event Monitoring**.

On the displayed **Event Monitoring** page, all system events generated in the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view the events generated in different periods.

Step 7 Click **View Graph**. On the details page, click **View Event** in the **Operation** column of a specific event to view details.

----End

3.18.4.3 Creating an Alarm Rule to Monitor an Event

Scenarios

This section describes how to create an alarm rule to monitor an event.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the **Create Alarm Rule** page, configure the parameters.

Table 3-62 Parameter description

Parameter	Description
Name	Specifies the alarm rule name. The system generates a random name, which you can modify.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create one.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Event Type	Specifies the event type of the metric corresponding to the alarm rule.
Event Source	Specifies the service the event is generated for. Select **Relational Database Service**.
Monitoring Scope	Specifies the monitoring scope for event monitoring. • **All resources**: If you select **All resources**, an alarm will be triggered when any RDS DB instance meets an alarm policy, and existing alarm rules will be automatically applied for newly purchased resources. • **Resource groups**: If you select **Resource groups**, an alarm will be triggered when any resource in the group meets the alarm policy. • **Specific resources**: Currently, RDS for PostgreSQL instance resources cannot be specified.
Method	Specifies the means you use to create the alarm rule.
Alarm Policy	**Event Name** indicates the instantaneous operations users performed on system resources, such as login and logout. For events supported by event monitoring, see **Events Supported by Event Monitoring**. You can select a trigger mode and alarm severity as needed.

Click  to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 3-63 Alarm notification

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Object	Specifies the object that receives alarm notifications. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Specifies the condition for triggering the alarm notification.

Step 6 Click **Create**.

----End

3.18.4.4 Events Supported by Event Monitoring

Table 3-64 Resource exception events

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
RDS	DB instance creation failure	createInstanceFailed	Major	A DB instance fails to create because the number of disks is insufficient, the quota is insufficient, or underlying resources are exhausted.	Check the number of disks and quota size. Release resources and create DB instances again.	DB instances cannot be created.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Full backup failure	fullBackupFailed	Major	A single full backup failure does not affect the files that have been successfully backed up, but prolongs the incremental backup restoration time during the point-in-time recovery (PITR).	Create a manual backup again.	Backup failed.
	Primary/standby switchover failure	activeStandBySwitchFailed	Major	The standby DB instance does not take over workloads from the primary DB instance due to network or server failures. The original primary DB instance continues to provide workloads within a short time.	Check whether the connection between your application and the database is re-established.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Replication status abnormal	abnormalReplicationStatus	Major	The possible causes are as follows: The replication delay between the primary and standby instances is too long, which usually occurs when a large amount of data is written to databases or a large transaction is processed. During peak hours, data may be blocked. The network between the primary and standby instances is disconnected.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write.
	Replication status recovered	replicationStatusRecovered	Major	The replication delay between the primary and standby instances is within the normal range, or the network connection between them has restored.	No action is required.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	DB instance faulty	faultyDBInstance	Major	A single or primary DB instance was faulty due to a disaster or a server failure.	Check whether an automated backup policy has been configured for the DB instance and submit a service ticket.	The database service may be unavailable.
	DB instance recovered	DBInstanceRecovered	Major	This event is reported after an RDS single-node instance is recovered from a disaster or its physical machine is recovered, or when a primary/standby instance completes a failover.	No action is required.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Failure of changing single DB instance to primary/ standby	singleToHaFailed	Major	A fault occurs when RDS is creating the standby DB instance or configuring replication between the primary and standby DB instances. The fault may occur because resources are insufficient in the data center where the standby DB instance is located.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write of the DB instance.
	Database process restarted	DatabaseProcessRestarted	Major	The database process is stopped due to insufficient memory or high load.	Log in to the Cloud Eye console. Check whether the memory usage increases sharply, the CPU usage is too high for a long time, or the storage space is insufficient. You can increase the CPU and memory specifications or optimize the service logic.	Downtime occurs. In this case, RDS automatically restarts the database process and attempts to recover the workloads.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Instance storage full	instanceDiskFull	Major	Generally, the cause is that the data space usage is too high.	Scale up the instance.	The DB instance becomes read-only because the storage space is full, and data cannot be written to the database.
	Instance storage full recovered	instanceDiskFullRecovered	Major	The instance disk is recovered.	No action is required.	The instance is restored and supports both read and write operations.
	Kafka connection failed	kafkaConnectionFailed	Major	The network is unstable or the Kafka server does not work properly.	Check your network connection and the Kafka server status.	Audit logs cannot be sent to the Kafka server.

Table 3-65 Operation events

Event Source	Event Name	Event ID	Event Severity	Description
RDS	Reset administrator password	resetPassword	Major	The password of the database administrator is reset.
	Operate DB instance	instanceAction	Major	The storage space is scaled or the instance class is changed.
	Delete DB instance	deleteInstance	Minor	The DB instance is deleted.
	Modify backup policy	setBackupPolicy	Minor	The backup policy is modified.
	Modify parameter group	updateParameterGroup	Minor	The parameter group is modified.
	Delete parameter group	deleteParameterGroup	Minor	The parameter group is deleted.
	Reset parameter group	resetParameterGroup	Minor	The parameter group is reset.
	Change database port	changeInstancePort	Major	Change database port

3.19 Billing Management

3.19.1 Renewing DB Instances

Scenarios

You can renew one or multiple yearly/monthly DB instances at a time.

NOTE

Pay-per-use DB instances cannot be renewed.

The statuses of yearly/monthly DB instances to be renewed must be **Available** or **Abnormal**.

Renewing One DB Instance

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the DB instance and click **Renew** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Overview** page. In the **Billing** area, click **Renew** under **Billing Mode**.

Step 5 Renew the DB instance.

----End

Renewing DB Instances in Batches

Step 1 Log in to the management console.

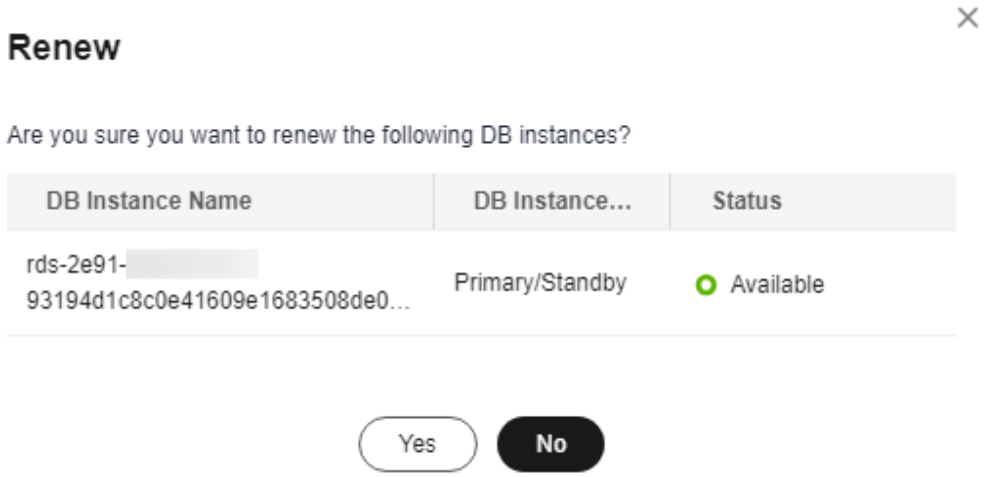
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the DB instances and click **Renew** above the DB instance list.

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

Figure 3-178 Renewing an instance



----End

3.19.2 Changing the Billing Mode from Pay-per-Use to Yearly/Monthly

Scenarios

If you use RDS for PostgreSQL for a long time, you can change the billing mode of one or multiple DB instances from pay-per-use to yearly/monthly at a time to save money.

NOTE

- Read replicas do not support changing the billing modes from pay-per-use to yearly/monthly.
- Pay-per-use DB instances in any of the following statuses cannot be changed to yearly/monthly: frozen, creation failed, changing instance class, and scaling up.

Changing the Billing Mode of One DB Instance

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and choose **More** > **Change to Yearly/Monthly** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Overview** page. In the **Billing** area, click **Configure** under **Billing Mode**.

Step 5 Select the renewal duration, in months. The minimum duration is one month.

- If you do not need to modify your settings, click **Pay** to go to the payment page.
- If you are not sure about the settings, the system will reserve your order. You can choose **Billing & Costs** > **Unpaid Orders** in the upper right corner and pay or cancel the order. The instance status is **Changing to Yearly/Monthly. Payment incomplete. Pay Now.**

Step 6 Select a payment method and click **Confirm**.

Step 7 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the instance list, click  to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Yearly/Monthly**.

----End

Changing the Billing Mode of DB Instances in Batches

NOTE

Only pay-per-use DB instances can be changed to yearly/monthly DB instances.

The statuses of pay-per-use DB instances must be **Available** or **Abnormal**.

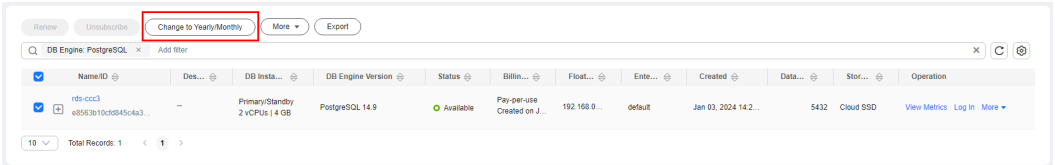
Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select multiple DB instances and click **Change to Yearly/Monthly** above the DB instance list.

Figure 3-179 Changing pay-per-use DB instances to yearly/monthly in batches



Step 5 Select the renewal duration, in months. The minimum duration is one month.

- If you do not need to modify your settings, click **Pay** to go to the payment page.
- If you are not sure about the settings, the system will reserve your order. You can choose **Billing & Costs** > **Unpaid Orders** in the upper right corner and pay or cancel the order. The instance status is **Changing to Yearly/Monthly. Payment incomplete. Pay Now.**

Step 6 Select a payment method and click **Confirm**.

Step 7 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the instance list, click  to refresh the list. After the change completes, the instance status will change to **Available** and the billing mode will change to **Yearly/Monthly**.

----End

3.19.3 Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Scenarios

You can change the billing mode of a DB instance from yearly/monthly to pay-per-use.

NOTICE

The pay-per-use billing mode is not applied until a yearly/monthly subscription expires, and only if auto-renew is not in effect.

Changing the Billing Mode from Yearly/Monthly to Pay-per-Use

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the yearly/monthly DB instance and choose **More** > **Change to Pay-per-use** in the **Operation** column.

Alternatively, click the DB instance name to go to the **Overview** page. In the **Billing** area, click **Change to Pay-per-use** under **Billing Mode**.

Step 5 On the displayed page, change the billing mode of the DB instance.

Step 6 Wait until the billing mode is successfully changed and view the instance on the **Instances** page.

In the upper right corner of the DB instance list, click  to refresh the list. After the DB instance billing mode is changed to pay-per-use, the instance status will change to **Available** and the billing mode will change to **Pay-per-use**.

----End

3.19.4 Unsubscribing from a Yearly/Monthly Instance

Scenarios

To delete a DB instance billed on the yearly/monthly basis, you need to unsubscribe the order. You can unsubscribe a single instance order by referring to **Unsubscribing a Single DB Instance (Method 1)** and **Unsubscribing a Single DB Instance (Method 2)** or unsubscribe multiple instance orders at a time by referring to **Unsubscribing DB Instances in Batches**. For unsubscription fees, see **Unsubscription Rules**.

If you unsubscribe from a DB instance, its read replicas (if any) will also be unsubscribed.

To release DB instances or read replicas billed on a pay-per-use basis, you need to locate the target DB instances or read replicas and click **Delete** on the **Instances** page. For details, see **Deleting a Pay-per-Use DB Instance or Read Replica**.

Constraints

- A DB instance cannot be unsubscribed when any operations are being performed on it. It can be unsubscribed only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be unsubscribed.

Unsubscribing a Single DB Instance (Method 1)

Unsubscribe a yearly/monthly DB instance or read replica on the **Instances** page.

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance or read replica and choose **More** > **Unsubscribe** in the **Operation** column.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription results. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing a Single DB Instance (Method 2)

Unsubscribe a yearly/monthly DB instance or read replica on the **Billing Center** page.

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

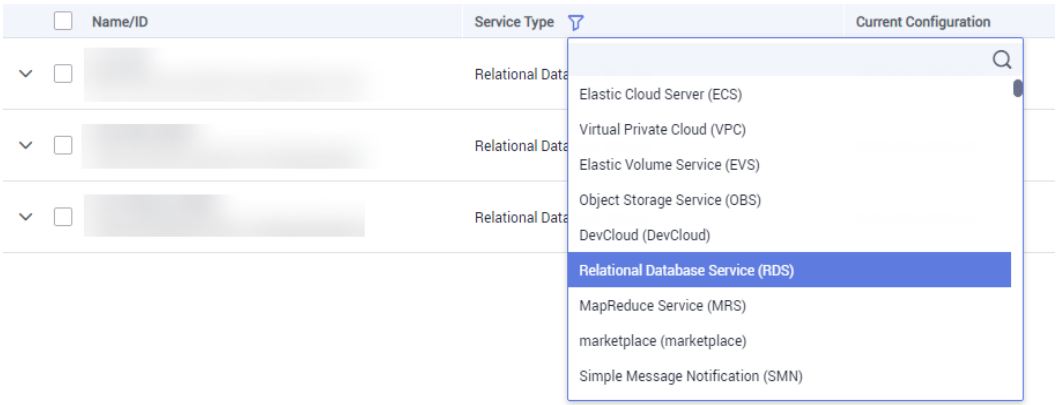
Step 4 In the upper right corner, click **Billing & Costs**.

Step 5 In the navigation pane, choose **Orders** > **Unsubscriptions**.

Step 6 On the displayed page, select the order to be unsubscribed and click **Unsubscribe** in the **Operation** column.

- You can select **Relational Database Service (RDS)** in the **Service Type** column to filter all RDS orders.

  Figure 3-180 Filtering all RDS orders



- Alternatively, search for target orders by name, order No., or ID in the search box.
- A maximum of 20 resources can be unsubscribed at a time.

Step 7 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 8 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 9 View the unsubscription result. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing DB Instances in Batches

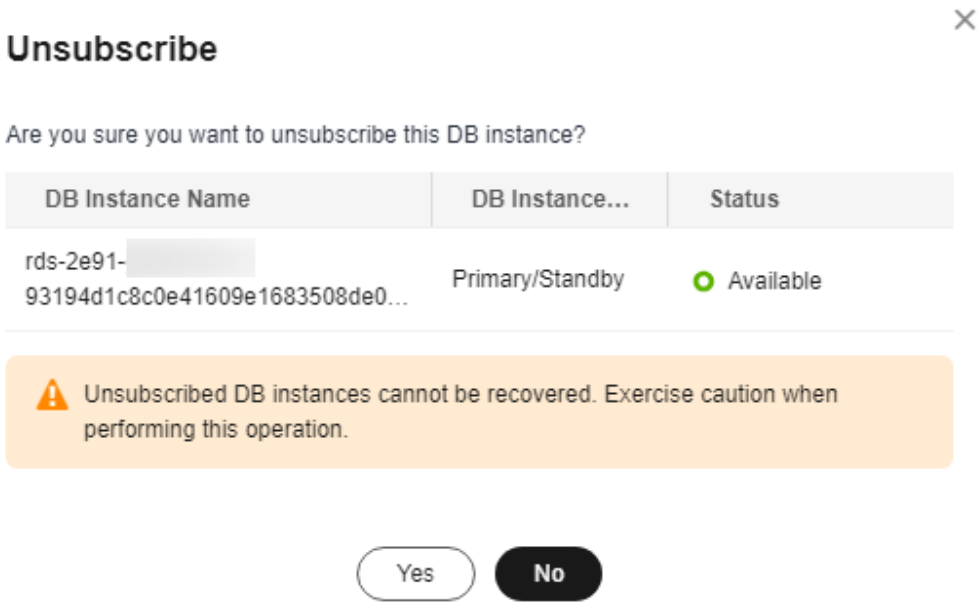
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the target DB instances to be unsubscribed and click **Unsubscribe** above the DB instance list. In the displayed dialog box, click **Yes**.

Figure 3-181 Unsubscribing yearly/monthly orders in batches



Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription results. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

3.20 Interconnection with CTS

3.20.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for further query, audit, and backtrack.

Table 3-66 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

Operation	Resource Type	Trace Name
Binding or unbinding an EIP	instance	setOrResetPublicIP
Modifying a security group	instance	modifySecurityGroup
Adding a tag	instance	setInstanceTag
Deleting a tag	instance	setInstanceTag
Editing a tag	instance	setInstanceTag
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	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 merged binlogs	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

Operation	Resource Type	Trace Name
Saving parameters in a parameter template	parameterGroup	saveParameterGroup
Deleting a frozen DB instance	all	rdsUnsubscribeInstance
Freezing a DB instance	all	rdsfreezeInstance
Changing the billing mode of a DB instance from pay-per-use to yearly/monthly or renewing a DB instance	all	bssUpdateMetadata
Creating a database account	instance	createDBUser
Resetting a password	instance	resetDBUserPassword
Changing account permissions	instance	grantDBUser
Modifying the host IP addresses of an account	instance	UpdateHostPrivilege
Deleting a database account	instance	deleteDBUser
Creating a database	instance	createDatabase
Authorizing a database	instance	grantDBUser
Deleting a database	instance	deleteDatabase

3.20.2 Viewing Tracing Events

For details about how to view audit logs, see **Querying Real-Time Traces**.

3.21 Task Center

3.21.1 Viewing a Task

You can view the progress and results of scheduled tasks on the **Task Center** page.

Supported Tasks

Table 3-67 Supported tasks

Task Type	Category	Task Name
Instant tasks	Instance creation	Creating a PostgreSQL DB instance, creating a PostgreSQL read replica

Task Type	Category	Task Name
	Instance lifecycle	Rebooting a PostgreSQL DB instance
	Instance modifications	Applying for a PostgreSQL private domain name, changing a PostgreSQL private domain name, migrating a standby PostgreSQL DB instance, changing the PostgreSQL instance type from single to primary/standby, scaling a PostgreSQL DB instance, binding an EIP to a PostgreSQL DB instance, unbinding an EIP from a PostgreSQL DB instance, switching PostgreSQL primary/standby instances, changing a PostgreSQL DB instance class
	Version upgrade	Upgrading a PostgreSQL major version, upgrading a PostgreSQL instance version
	Backup and restoration	Restoring to a new PostgreSQL DB instance, restoring to an existing PostgreSQL DB instance, restoring PostgreSQL databases, restoring PostgreSQL tables
	Parameter configuration	Modifying a PostgreSQL parameter template
	Instance DR	Creating DR relationship for PostgreSQL primary instance, creating DR relationship for PostgreSQL DR instance, promoting PostgreSQL DR instance to primary
Scheduled tasks	Instance lifecycle	Starting a PostgreSQL DB instance
	Instance modifications	Changing a PostgreSQL DB instance class

Viewing an Instant Task

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Task Center** page, locate the target task and view its details.

----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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

3.21.2 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. It 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 in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed **Instant Tasks** page, locate the task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

RDS allows you to delete the tasks in the following statuses:

- 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  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the 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. Click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

RDS allows you to delete the tasks in the following statuses:

- Completed
- Failed
- Canceled
- To be executed
- To be authorized

----End

3.22 RDS for PostgreSQL Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

- Log in to the management console. Click **Service List** and choose **Management & Governance** > **Tag Management Service**. Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Up to 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 Log in to the management console.

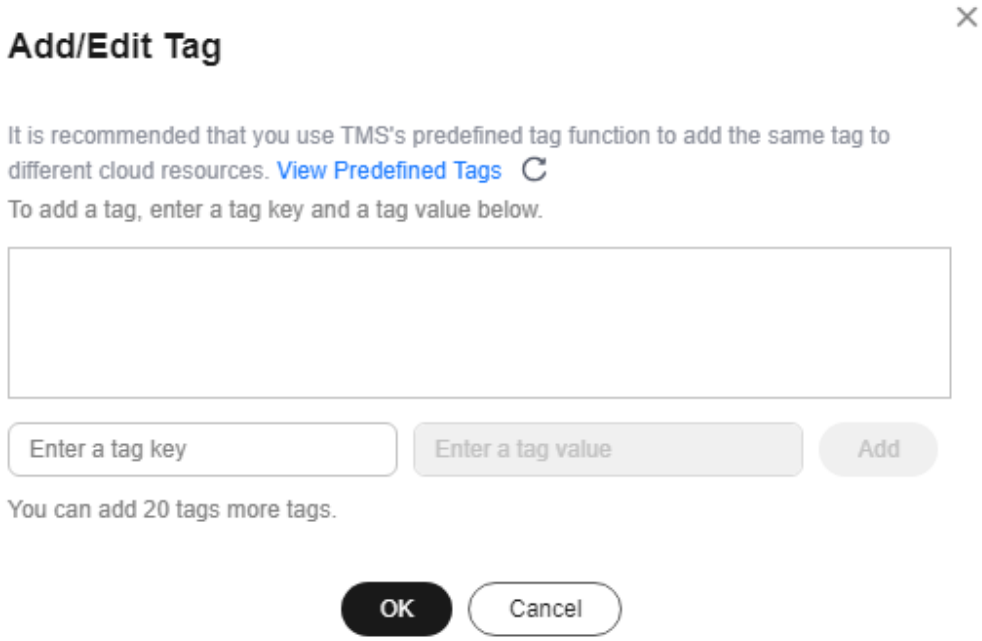
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Tags**. On the displayed page, click **Add/Edit Tag**. In the displayed dialog box, enter a tag key and value, click **Add**, and click **OK**.

Figure 3-182 Adding a tag



- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with your DB instances (except the current instance).
- The tag key must be unique. It must consist of 1 to 128 characters and can include letters, digits, spaces, and the following characters: _ **.** **:** **=** **+** **-** **@**. It cannot start or end with a space, or start with **_sys_**.
- The tag value (optional) can consist of up to 255 characters and can include letters, digits, spaces, and the following characters: _ **.** **:** **/** **=** **+** **-** **@**.

Step 6 After a tag has been added, view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane on the left, choose **Tags**. Select the tag to be deleted and click **Delete**. In the displayed dialog box, click **OK**.

After a tag has been deleted, it will no longer be displayed on the **Tags** page.

----End

3.23 RDS for PostgreSQL Quotas

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of RDS quotas include the maximum number of DB instances that you can create. Quotas are put in place to prevent excessive resource usage.

If a quota cannot meet your needs, apply for a higher quota.

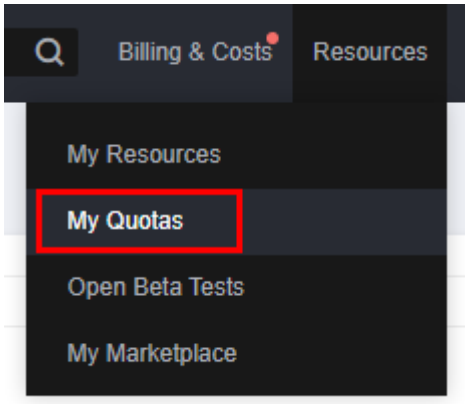
Viewing Quotas

Step 1 Log in to the management console.

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

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Figure 3-183 My quotas



Step 4 On the **Quotas** page, view the used and total quotas of each type of resources.

----End

Increasing Quotas

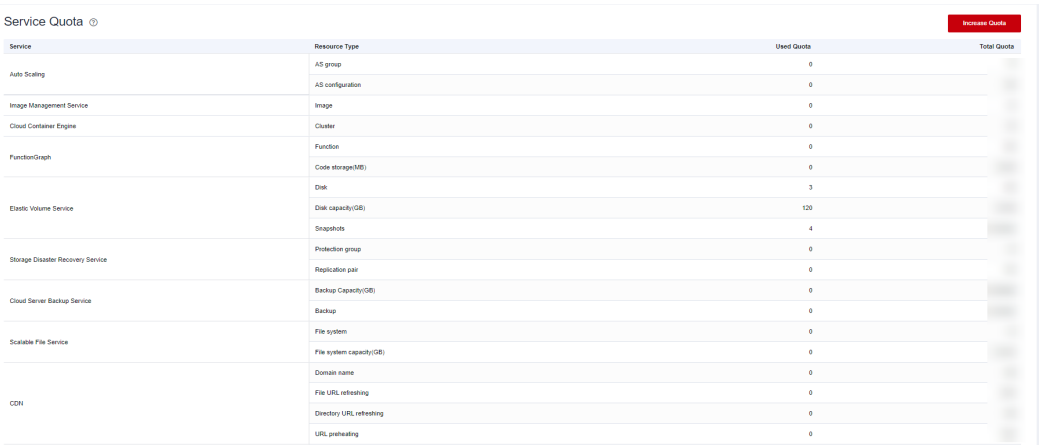
Step 1 **Log in to the management console**.

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

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 3-184 Increasing quotas



Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for adjustment.

Step 6 After all required parameters are configured, select the agreement and click **Submit**.

----End

3.24 RDS for PostgreSQL Enhanced Edition

3.24.1 Introduction to RDS for PostgreSQL Enhanced Edition

RDS for PostgreSQL has two editions: PostgreSQL Community Edition and PostgreSQL Enhanced Edition. RDS for PostgreSQL Enhanced Edition is no longer sold. If you have already purchased RDS for PostgreSQL Enhanced Edition, you can enjoy the same O&M assurance as before. If you are a new customer who wants to buy RDS for PostgreSQL Enhanced Edition, GaussDB is recommended instead.

3.24.2 Functions

This section describes the built-in functions and advanced function packages added to RDS for PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 3-68 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
nls_lower(char[, nlsparam])	Converts all alphabetic characters in the character string char to lowercase 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.
nvl(expr1, expr2)	Returns the first non-null value in expr1 and expr2.
rawtohex(raw)	Converts raw to a character value containing its hexadecimal representation.
regexp_count(source_char, pattern, position, match_param)	Returns the number of times a pattern occurs in a source string starting from the position that indicates the source_char character where the database begins the search. The match_param parameter is a text literal that lets you change the default matching behavior of the function. For example, match_param='i' specifies case-insensitive matching.
regexp_instr(source_char, pattern[, position[, occurrence[, return_opt[, match_param[, subexpr]]]]])	Extends the INSTR function and allows regular expression matching. The return type is INTEGER. ● position: indicates the start position of the search. ● occurrence: indicates the sequence number of the pattern in source_char. ● return_opt: – The value 0 indicates the start position of the return mode. – The value 1 indicates the end position of the return mode. ● match_param: indicates the control parameter of the regular expression, such as case sensitive. ● subexpr: indicates the group number of the regular expression group.
regexp_like(source_char, pattern[,match_param])	source_char is a character expression. Pattern is the regular expression. The match_param parameter is a text literal that lets you change the default matching behavior of the function.

Built-in Function	Description
regexp_substr(source_char, pattern[,position[,occurrence[, match_param[,subexpr]]]])	Matches the character string in the source_char string based on the regular expression. ● source_char 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. ● pattern is the text expression to search for. ● position is a nonzero integer indicating the character of source_char where the function begins the search. ● occurrence is an integer indicating which occurrence of pattern the function should search for. ● match_parameter is a text expression that lets you change the default matching behavior of the function. ● subexpr is a nonnegative integer from 0 to 9 indicating which subexpression in pattern is to be returned by the function.
raise_application_error(errnum , errmsg)	Sends the error code errnum and error message errmsg to the client.
remainder(n2, n1)	Returns the remainder of n2 divided by n1. The remainder function is similar to mod except that mod uses floor in its formula, whereas reminder uses ROUND. The return type is NUMERIC or double-precision floating-point number (determined by the input parameter type).
round(n,precision)	Returns n rounded to integer places to the right of the decimal point. The precision is the number of digits in a number.
scn_to_timestamp(number)	Returns the approximate timestamp associated with a system change number (SCN).
sinh(n)	Returns the hyperbolic sine of n. If n 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_yminterval('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 record rows of the table associated with correlation_variable in object table mode. The return type is the object table associated with correlation_variable.

Table 3-69 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, endianess)	Converts the integer n to the RAW type based on the memory alignment mode specified by endianess. The values of endianess are as follows: • 1: big_endian • 2: little_endian • 3: machine_endian

3.24.3 System Views

This section describes the system views added to RDS for PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 3-70 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 3-71 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.

3.24.4 Data Types

This section describes the data types added to RDS for PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 3-72 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

3.24.5 Implicit Type Conversion

This section describes the implicit type conversion added to RDS for 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

3.24.6 Predefined Parameters

This section describes the predefined parameters added to RDS for PostgreSQL Enhanced Edition on the basis of PostgreSQL 11 open-source edition.

Table 3-73 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.

3.24.7 Macro Variables

This section describes the macro variables added to RDS for 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.

3.24.8 Operators

This section describes the following operators added to RDS for 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 (<=).

3.24.9 Syntax

This section describes the syntax added to RDS for 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 (l) 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

4 Working with RDS for SQL Server

4.1 Suggestions on Using RDS for SQL Server

Instance Class

Do not use instances with 2 vCPUs and 4 GB memory for production workloads. Such instances are provided only for experience testing.

Use instances with at least 4 vCPUs and 8 GB memory for production workloads. Instances with 2 vCPUs and 4 GB memory are not suitable for production workloads because Microsoft SQL Server runs on Windows and both the engine and the OS require a large number of resources. Using instances with 2 vCPUs and 4 GB memory for a long time may result in memory exhaustion and system freezing.

Database Connection

- Use the form of "ip,port" (use a comma (,) between them) to connect to an RDS for SQL Server instance.
- Do not use the server name to connect to a database.
- Your application must be able to reconnect to the database if a disaster occurs in the database or the database is disconnected.

Database Migration

After the migration is complete, perform the following operations:

- Check the permissions integrity. Database migration only restores data. Other service-level permissions, such as those for database users and login names, must be recreated and re-associated with database accounts.
- Recreate indexes. After the migration is complete, the physical environment of data files changes, and the database indexes become invalid. You need to recreate the indexes to minimize the impact on the database performance.
- Compare parameter settings. After data is migrated to the cloud, RDS for SQL Server uses parameter groups provided on the cloud. You need to compare the parameter settings on the cloud with those of the original on-premises

database. Modify the parameter settings on the cloud to keep them the same as those for the original database.

Instance Usage

- Although RDS for SQL Server supports it, creating an instance configured with the AD domain is not recommended. This is because the domain controller server is deployed on the user side, and the user has overly lax permissions. If the user changes the group policy configurations of the domain controller server, the DB instance security can be affected.
- There can be no more than 100 databases in a single instance. The maximum number of databases that a single DB instance supports depends on the instance specifications. Too many databases occupy resources such as worker threads, which impacts instance performance.
- Do not use the sysadmin role to connect your application to a database. An account with the sysadmin role has the super administrator permission. Improper use of this account will threaten database security and stability. RDS does not grant the super administrator permission to any user.
- Do not create tables in the system database. Create a user-defined database to store user data. Do not create any tables in the system database to write data because storing data in the system database is insecure.
- Do not enable the AutoClose property for the database. Enabling AutoClose may result in failures to establish the replication relationship between primary and standby DB instances.
- Do not set the database to single-user mode. Single-user mode allows only one session to access the database at a time. If a fault occurs in the database, sessions initiated by O&M personnel will be unable to connect to the database. If you have set your database to single-user mode, change it to multi-user mode.
- Do not leave **Slow Query Log** enabled for a long time. Slow query logs help analyze slow SQL statements. However, if **Slow Query Log** is enabled for a long time, database performance will deteriorate. You are advised to disable **Slow Query Log** when you are not tracing or analyzing SQL problems.
- Schedule a time to automatically recreate indexes. When a database is used for a long time, a large number of index fragments may be generated. This slows down database access. To address this issue, create an SQL Agent job to recreate indexes once a month.
- Update statistics periodically. Database statistics need to be updated at regular intervals. You are advised to create an SQL Agent job to update statistics once a week.
- Pay attention to the database size and shrink the database as required. If a database has been used for a long time, some physical space may not be released in a timely manner. In this case, you need to shrink the database to release the physical space. Pay attention to the log file size and physical file size. If any file bloat is found, shrink the database during off-peak hours.
- The database name cannot exceed 64 characters. Only digits, uppercase letters, lowercase letters, hyphens (-), and underscores (_) are allowed.
- You are advised to change the default port. The default port of RDS for SQL Server is **1433**. Some insecure programs on the Internet may scan the default port.

- You are advised to use primary/standby instances. Primary/standby instances provide much better availability and reliability for production workloads.
- Deploy primary/standby instances across AZs for AZ-level DR.
- During off-peak hours, reboot instances that have been running for a long time. When an instance has been running for a long time, its performance may deteriorate. You are advised to reboot the instance every three months during off-peak hours.
- Configure the maximum degree of parallelism. This parameter affects the CPU usage of your workloads. Its default value is **0**, indicating that a session can use all CPUs. If you set it to the default value, the CPUs may not be allocated to other sessions due to a SQL problem. You are advised to configure this parameter based on instance specifications, for example, setting it to the value of the number of cores divided by 2.
- Create multiple NDF files for the tempdb database.
- If there is a permissions problem when you perform an operation, refer to **Usage of Stored Procedures** to find a proper stored procedure.
- To modify SQL Server parameters, instead of running SQL commands, modify them on the console.
- Back up and restore data on the console or by calling RDS APIs or SDK APIs. Do not use SQL Server Management Studio (SSMS) or SQL statements for backup and restoration. For details about how to migrate your data to RDS, see **Data Replication Service (DRS)**.
- Restoring data to an existing DB instance may cause existing data to be overwritten. Exercise caution when performing this operation. You are advised to restore data to a new DB instance.
- Set the recovery model of your database to FULL instead of SIMPLE.
  - In the SIMPLE recovery model, no incremental backup is performed for the database, so the database cannot be restored to a specified time point.
  - For primary/standby or cluster instances, if the recovery model is set to SIMPLE, no replication relationship will be established for the instances. As a result, a primary/standby switchover or instance class change cannot be performed.
- Avoid long-running transactions or transactions that stay uncommitted for a long time. Long-running transactions cause transaction logs to grow. The storage becomes full because the space cannot be reclaimed in phases. A large number of lock waits are generated, blocking the execution of other SQL statements. If you kill the long-running transactions, the rollback takes a longer time (at least 1.5 times that required for transaction execution). As a result, the primary/standby replication delay increases, the primary/standby switchover fails, and the instance class change fails.
- Do not create too many databases and login names (less than 100 each) for a DB instance. Too many databases and login names may cause slow permission synchronization after a primary/standby switchover or slow permission replay when the read-only state is removed after a scale-up. And also, it is difficult for you to manage the mapping between login names and database users.

4.2 Instance Connection

4.2.1 Connecting to an RDS for SQL Server Instance

You can connect to an RDS for SQL Server instance using the SQL Server Management Studio client or Data Admin Service (DAS).

Table 4-1 Connection methods

Connection Method	Description
Connecting to an RDS for SQL Server Instance Through DAS (Recommended)	DAS enables you to manage databases on a web-based console and provides you with database development, O&M, and intelligent diagnosis to make it easy to use and maintain your databases. The permissions required for connecting to DB instances through DAS are enabled by default.
Connecting to an RDS for SQL Server Instance Through the SQL Server Management Studio Client	You can use SQL Server Management Studio to connect to a DB instance through a non-SSL connection or an SSL connection. The SSL connection encrypts data and is more secure.

4.2.2 Connecting to an RDS for SQL Server Instance Through DAS (Recommended)

Scenarios

Data Admin Service (DAS) enables you to connect to and manage DB instances with ease on a web-based console. The permissions required for connecting to DB instances through DAS are enabled by default. Using DAS to connect to your DB instance is recommended, which is more secure and convenient.

Procedure

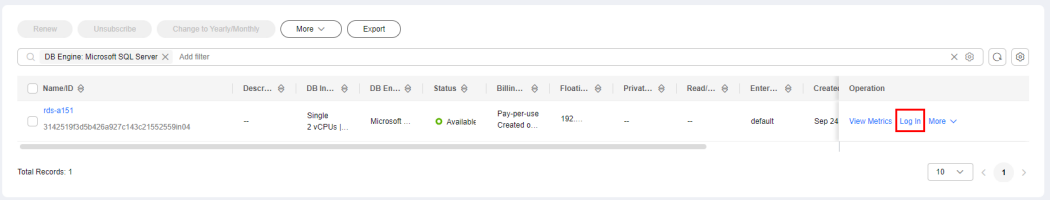
Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

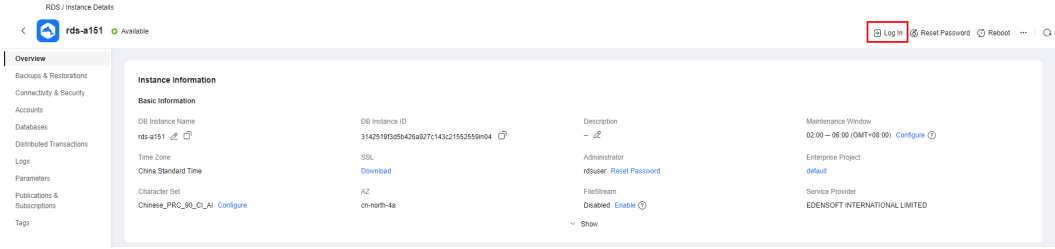
Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Figure 4-1 Logging in to an instance



Alternatively, click the DB instance name on the **Instances** page. On the displayed **Overview** page, click **Log In** in the upper right corner.

Figure 4-2 Logging in to an instance



Step 5 On the displayed login page, enter the username and password and click **Log In**.

----End

FAQs

Q: What can I do if the DAS console is not displayed after I click **Log In** in the **Operation** column of an instance on the **Instances** page?

A: Set your browser to allow pop-ups and try again.

- **What Should I Do If I Can't Connect to My DB Instance Due to Insufficient Permissions?**
- **What Should I Do If I Can't Connect to My RDS for SQL Server Instance?**

Follow-up Operations

After logging in to the DB instance, you can create or migrate your databases.

- **Managing RDS for SQL Server Databases Using DAS**
- **Migration Solution Overview**

4.2.3 Connecting to an RDS for SQL Server Instance Through the SQL Server Management Studio Client

4.2.3.1 Connecting to a DB Instance from a Windows ECS

When your applications are deployed on an ECS that is in the same region and VPC as your RDS for SQL Server DB instance, you are advised to use a floating IP address to connect to the DB instance through the ECS.

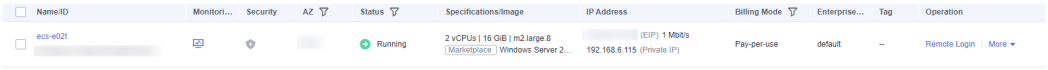
You can connect to an instance through a Secure Socket Layer (SSL) connection or a non-SSL connection using SQL Server Management Studio. The SSL connection encrypts data and is more secure.

For details about how to connect to a DB instance with SSL disabled, see **Connecting to a DB Instance from a Windows ECS**.

Step 1: Buy an ECS

1. **Log in to the management console** and check whether there is an ECS available.
   - If there is a Windows ECS, go to **3**.
   - If no Windows ECS is available, go to **2**.

   Figure 4-3 ECS

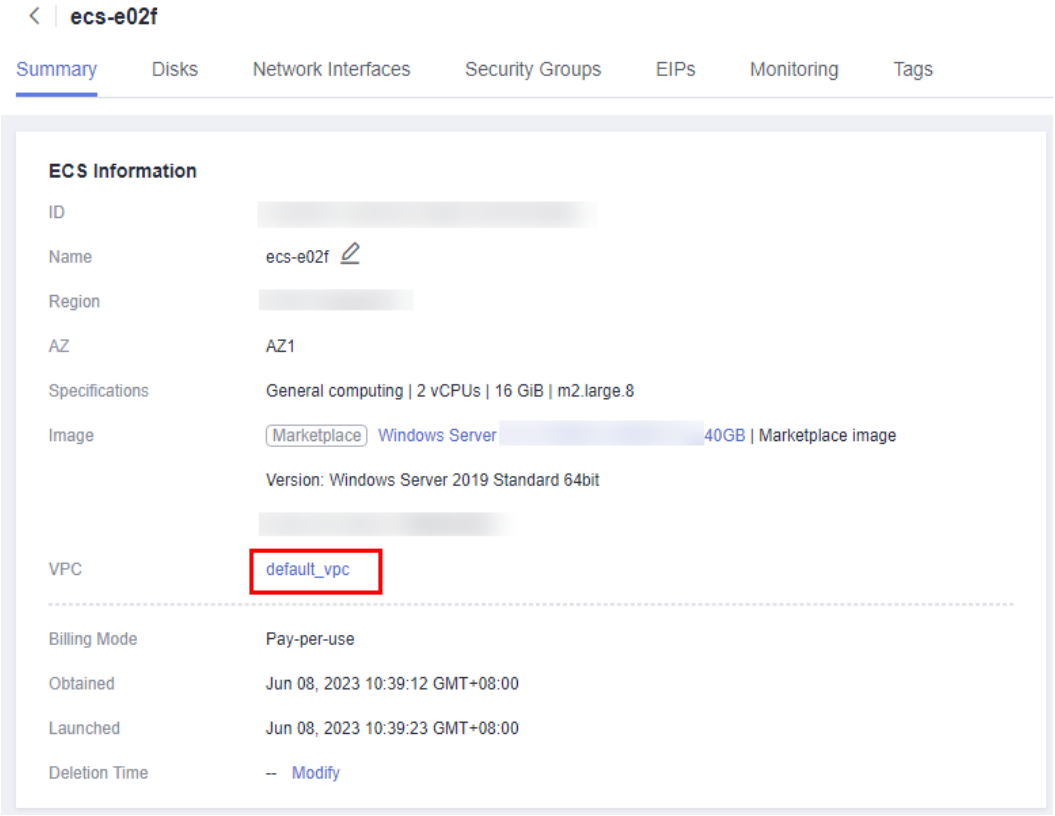
2. Buy an ECS and select Windows as its OS.

   To download SQL Server Management Studio to the ECS, bind an EIP to the ECS. The ECS must be in the same region, VPC, and security group as the RDS for SQL Server DB instance for mutual communications.

   For details about how to purchase a Windows ECS, see **Purchasing a Custom ECS** in *Elastic Cloud Server User Guide*.

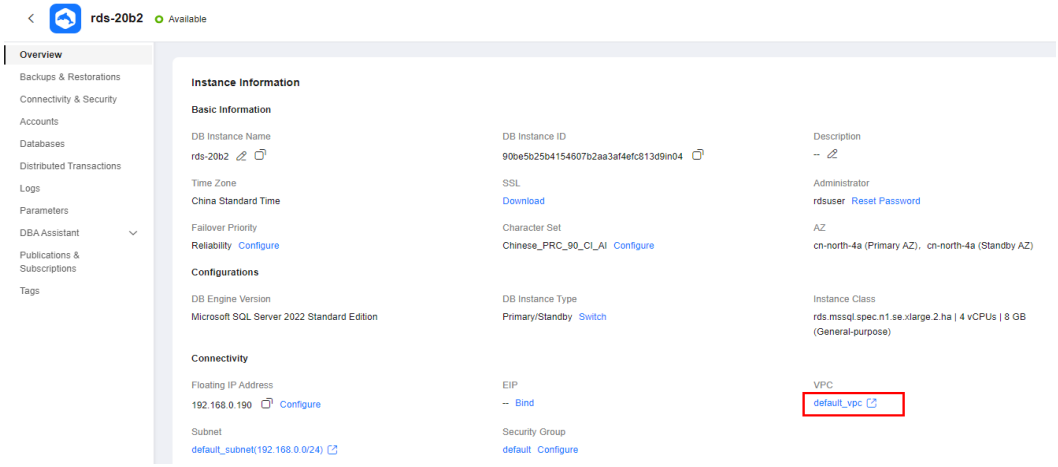
3. On the **ECS Information** page, view the region and VPC of the ECS.

   Figure 4-4 ECS information

4. On the **Overview** page of the RDS for SQL Server instance, view the region and VPC of the DB instance.

Figure 4-5 Overview

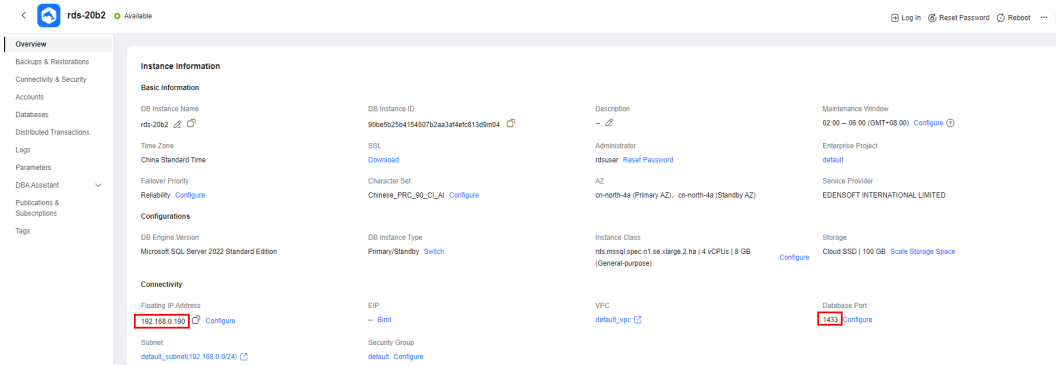


5. Check whether the ECS and RDS for SQL Server instance are in the same region and VPC.
   - If yes, go to **Step 2: Test Connectivity and Install SQL Server Management Studio**.
   - If they are not in the same region, purchase another ECS or DB instance. The ECS and DB instance in different regions cannot communicate with each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.
   - If the ECS and DB instance are in different VPCs, change the VPC of the ECS to that of the DB instance. For details, see **Changing a VPC**.

Step 2: Test Connectivity and Install SQL Server Management Studio

1. Log in to the ECS. For details, see **Login Using VNC** in the *Elastic Cloud Server User Guide*.
2. On the **Instances** page of the RDS console, click the DB instance name to go to the **Overview** page.
3. Obtain the floating IP address and database port of the DB instance.

Figure 4-6 Connection information



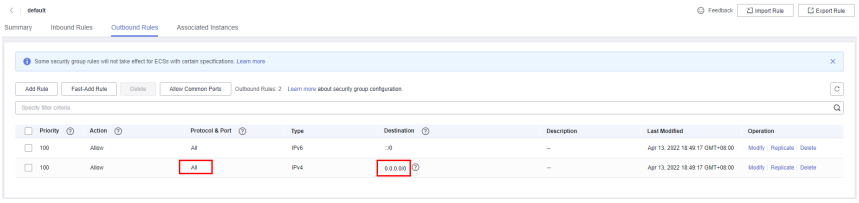
4. Open the cmd window on the ECS and check whether the floating IP address and database port of the DB instance can be connected.

   telnet *192.168.2.182 1433*

   - If yes, network connectivity is normal.
   - If no, check the security group rules.

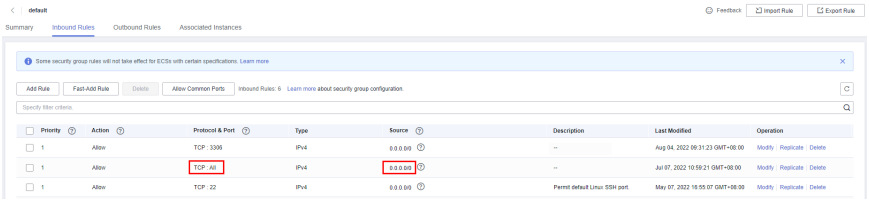
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the floating IP address and port of the DB instance.

       Figure 4-7 ECS security group

     - If in the security group of the DB instance, there is no inbound rule with **Source** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an inbound rule for the private IP address and port of the ECS. For details, see **Configuring Security Group Rules**.

       Figure 4-8 DB instance security group

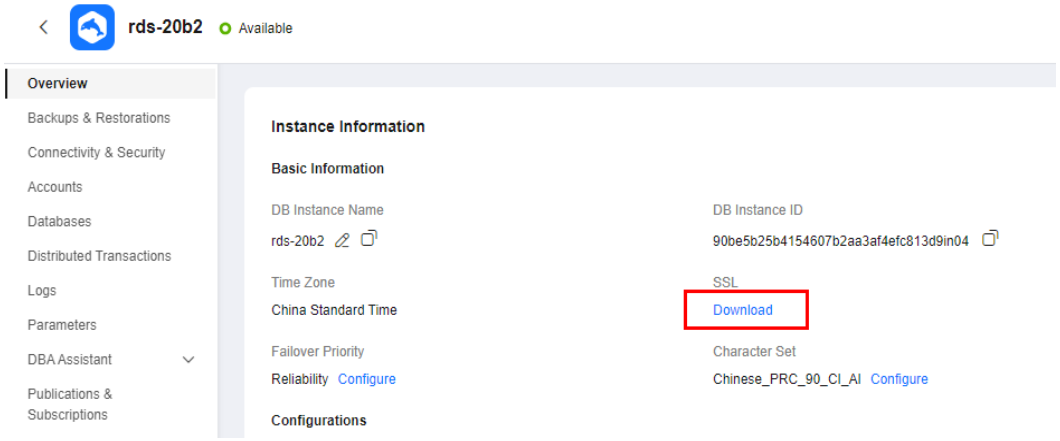
       

5. Open a browser on the ECS, visit the **Microsoft website**, and download the installation package, for example, SQL Server Management Studio 18.0.
6. Double-click the installation package and complete the installation as instructed.

Step 3: Connect to the DB Instance Using SQL Server Management Studio

1. On the **Instances** page of the RDS console, click the DB instance name to go to the **Overview** page.
2. Click **Download** under **SSL** to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Figure 4-9 Downloading a certificate



NOTE

- Replace the old certificate before it expires to improve system security.
- Replacing a certificate requires you to **submit a service ticket** to apply for permissions. After being granted the permissions, you can click **Replace Certificate** under **SSL** and then click **OK** in the displayed dialog box.
- After you bind an EIP to a DB instance, you must reboot the instance for the SSL connection to take effect.

3. Upload the root certificate **ca.pem** to the ECS. For details, see **How Can I Import the Root Certificate to a Windows or Linux OS?**
4. Start SQL Server Management Studio.
5. Choose **Connect** > **Database Engine**. In the displayed dialog box, enter login information.

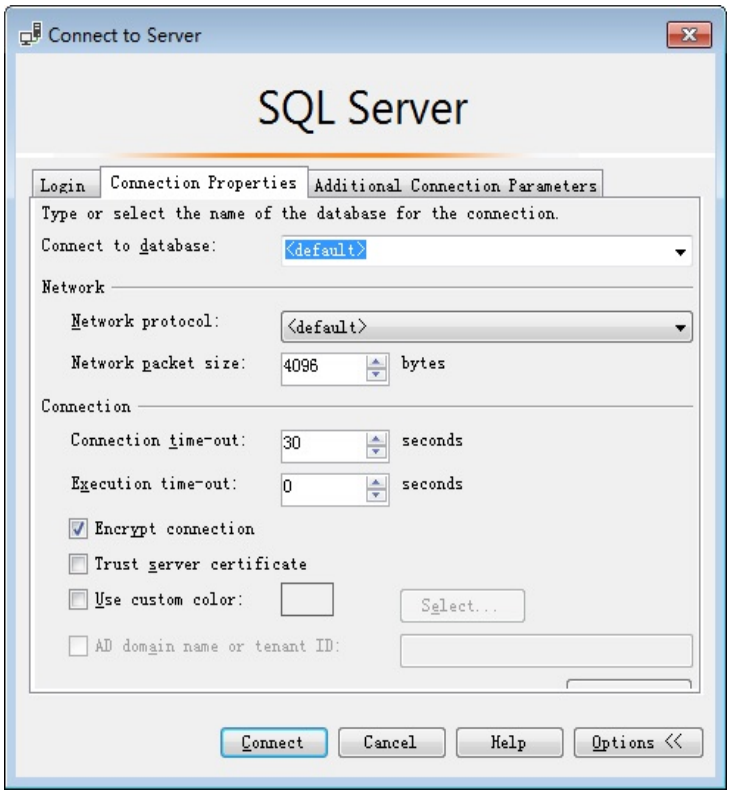
Figure 4-10 Connecting to the server

Table 4-2 Parameter description

Parameter	Description
Server name	Floating IP address and database port obtained in 3.
Authentication	Authentication mode. Select **SQL Server Authentication**.
Login	Name of the account used to access the DB instance. The default value is **rdsuser**.
Password	Password of the account.

6. Click **Options**. 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 4-11 Connection properties



7. Click **Connect** to connect to the DB instance.

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Managing RDS for SQL Server Databases Using DAS**
- **Migration Solution Overview**

4.2.3.2 Connecting to a DB Instance from a Windows Server

If you cannot access your DB instance through a floating IP address, bind an EIP to the DB instance and connect to it through the EIP.

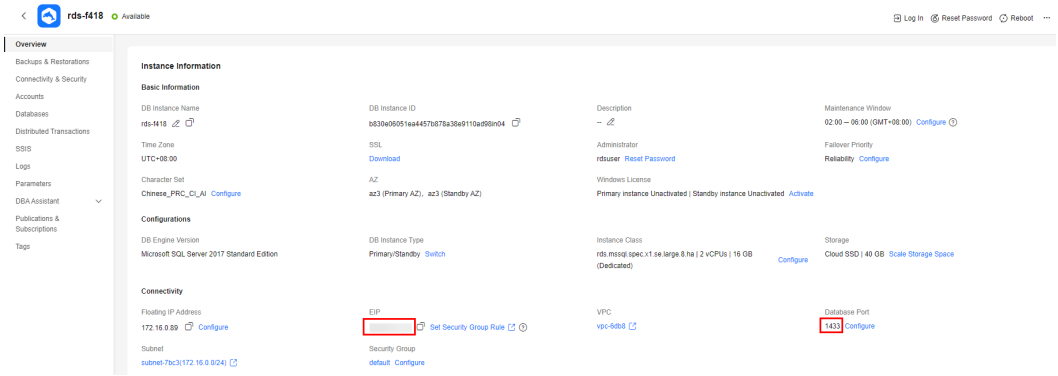
You can connect to an instance through a non-SSL connection or an SSL connection using SQL Server Management Studio. The SSL connection encrypts data and is more secure.

For details about how to connect to a DB instance with SSL disabled, see **Connecting to a DB Instance from a Windows Server**.

Step 1: Test Connectivity and Install SQL Server Management Studio

1. On the **Instances** page of the RDS console, click the DB instance name to go to the **Overview** page.
2. Obtain the EIP and database port of the DB instance.

   Figure 4-12 Connection information

   If no EIP has been bound to the DB instance, see **Binding and Unbinding an EIP**.

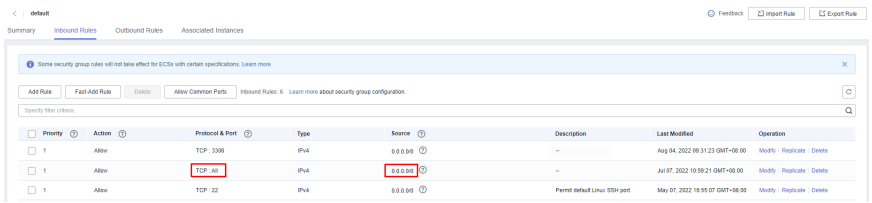
3. Open the cmd window on your local server and check whether the EIP and database port of the DB instance can be connected.

   telnet *EIP 1433*

   - If yes, network connectivity is normal.
   - If no, check the security group rules.

     If in the security group of the DB instance, there is no inbound rule with **Source** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an inbound rule for the EIP and port of the DB instance. For details, see **Configuring Security Group Rules**.

     Figure 4-13 DB instance security group

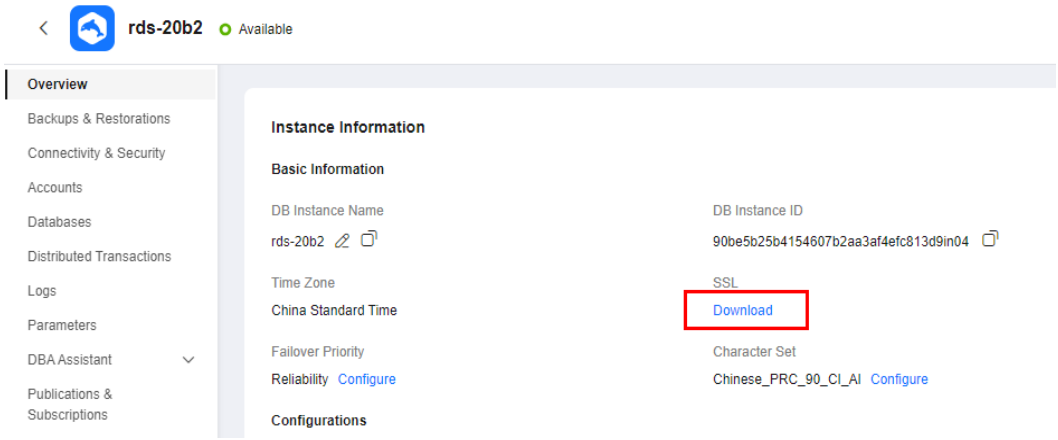
     

4. Open a browser on the local server, visit the **Microsoft website**, and download the installation package, for example, SQL Server Management Studio 18.0.
5. Double-click the installation package and complete the installation as instructed.

Step 2: Connect to the DB Instance Using SQL Server Management Studio

1. On the **Instances** page of the RDS console, click the DB instance name to go to the **Overview** page.
2. Click **Download** under **SSL** to download **Certificate Download.zip**, and extract the root certificate **ca.pem** and bundle **ca-bundle.pem** from the package.

Figure 4-14 Downloading a certificate



NOTE

- Replace the old certificate before it expires to improve system security.
- Replacing a certificate requires you to **submit a service ticket** to apply for permissions. After being granted the permissions, you can click **Replace Certificate** under **SSL** and then click **OK** in the displayed dialog box.
- After you bind an EIP to a DB instance, you must reboot the instance for the SSL connection to take effect.

3. Upload the root certificate **ca.pem** to the ECS. For details, see **How Can I Import the Root Certificate to a Windows or Linux OS?**
4. Start SQL Server Management Studio.
5. Choose **Connect** > **Database Engine**. In the displayed dialog box, enter login information.

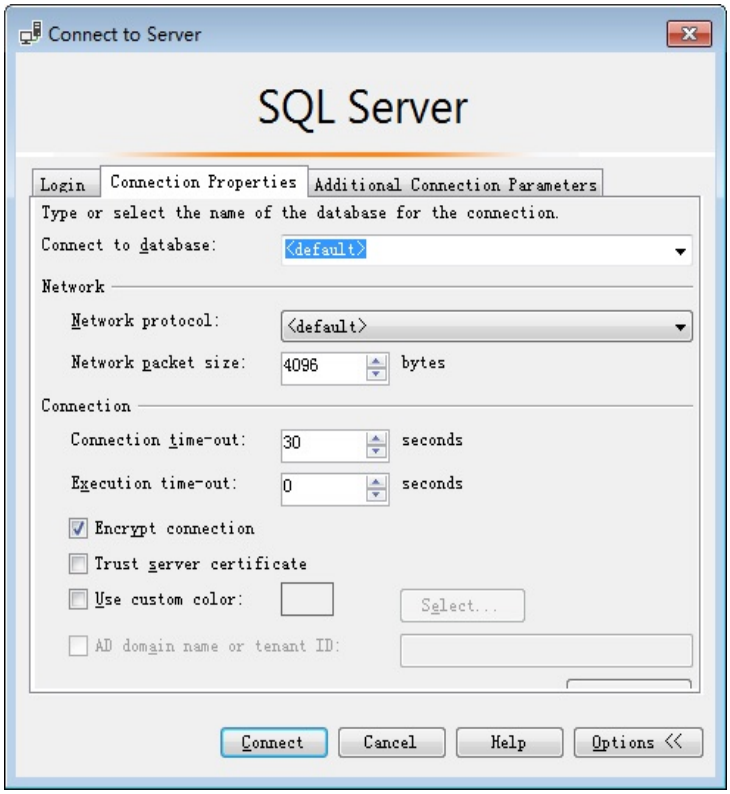
Figure 4-15 Connecting to the server

Table 4-3 Parameter description

Parameter	Description
Server name	EIP and database port obtained in 2.
Authentication	Authentication mode. Select **SQL Server Authentication**.
Login	Name of the account used to access the DB instance. The default value is **rdsuser**.
Password	Password of the account.

6. Click **Options**. 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 4-16 Connection properties



7. Click **Connect** to connect to the DB instance.

Follow-up Operations

After logging in to the DB instance, you can create or migrate databases.

- **Managing RDS for SQL Server Databases Using DAS**
- **Migration Solution Overview**

4.2.3.3 Installing 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 Windows 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 Double-click the installation package and complete the installation as instructed.

----End

4.3 Database Migration

4.3.1 Migration Solution Overview

You can migrate data from on-premises SQL Server databases or SQL Server databases built on other clouds to RDS for SQL Server, or from an RDS for SQL Server instance to another RDS for SQL Server instance.

Data migration tools include Data Replication Service (DRS) and Data Admin Service (DAS). You are advised to use DRS because it is easy to use and can complete a migration task in minutes. DRS facilitates data transfer between databases, helping you reduce DBA labor costs and hardware costs.

DRS provides backup migration and real-time synchronization.

- Backup migration: You can export data from the source database for backup and upload the backup files to OBS. Then, you can restore the backup files to the destination database to complete the migration. Using this method, data migration can be completed without exposing your source databases to the Internet.
- Real-time synchronization: Real-time synchronization refers to the real-time flow of key service data from sources to destinations through a synchronization instance while consistency of data can be ensured. It is different from migration. Migration means moving your overall database from one platform to another. Synchronization refers to the continuous flow of data between different services.

For more information, see **What Is DRS?**

Migration Solutions

Table 4-4 RDS for SQL Server migration solutions

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
RDS for SQL Server	Medium	One-time	Some time	Use DAS to export data from the source and then import the data to the destination RDS for SQL Server instance.	**Migrating Data to RDS for SQL Server Using the Export and Import Functions of DAS**
	Any	One-time or continuous	Minimal	Use DRS to migrate backup data from the source to the destination RDS for SQL Server instance.	**Creating an RDS Backup Migration Task**

Source Database	Data Size	One-Time or Continuous Migration	Application Downtime	Solution	Document
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from the source to the destination RDS for SQL Server instance.	From Microsoft SQL Server to RDS for SQL Server
On-premises SQL Server databases	Any	One-time or continuous	Minimal	Use DRS to migrate backup data of on-premises SQL Server databases to RDS for SQL Server.	Migrating Microsoft SQL Server Backup Data to RDS for SQL Server Instance
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from on-premises SQL Server databases to RDS for SQL Server.	From Microsoft SQL Server to RDS for SQL Server
SQL Server databases on other clouds	Any	One-time or continuous	Minimal	Use DRS to migrate backup data of SQL Server databases on other clouds to RDS for SQL Server.	Creating a Backup Using OBS Buckets
	Any	One-time or continuous	Minimal	Use DRS to synchronize data from SQL Server databases on other clouds to RDS for SQL Server.	From Microsoft SQL Server to RDS for SQL Server

4.3.2 Migrating Data to RDS for SQL Server Using the Export and Import Functions of DAS

Scenarios

To back up or migrate data, you can use Data Admin Service (DAS) to export data from the source database first and then import the data to from your local PC or OBS bucket to the destination database.

For more information, see **Import and Export**.

Constraints

- Take care when exporting or importing data. Improper operations can cause instance or workload exceptions.
- Only one file that is no larger than 1 GB can be imported at a time.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- If there are more than 10,000 tables in an RDS for SQL Server instance, an error will be reported when you export data using the **Export Database** function of DAS. In this case, use the **Export SQL Result** function instead.

Exporting Data

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

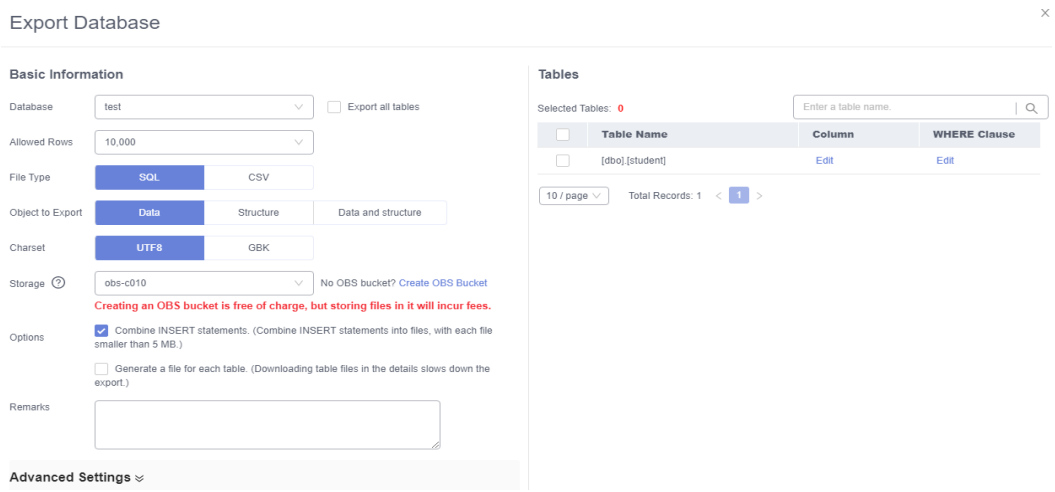
Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **Import and Export** > **Export**.

Step 7 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.

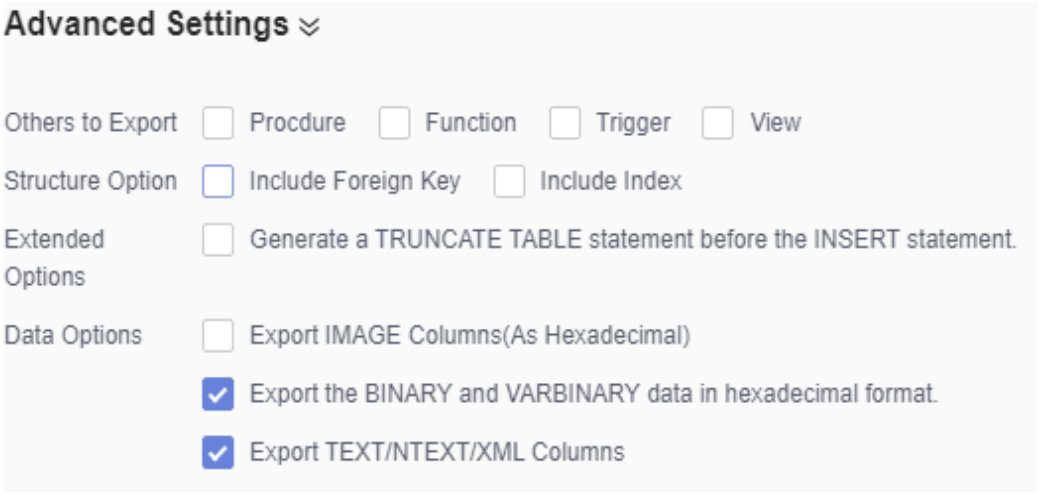
Alternatively, click **Quick Export** to export the specified database information quickly.

Figure 4-17 Creating an export task



Step 8 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

Figure 4-18 Advanced settings



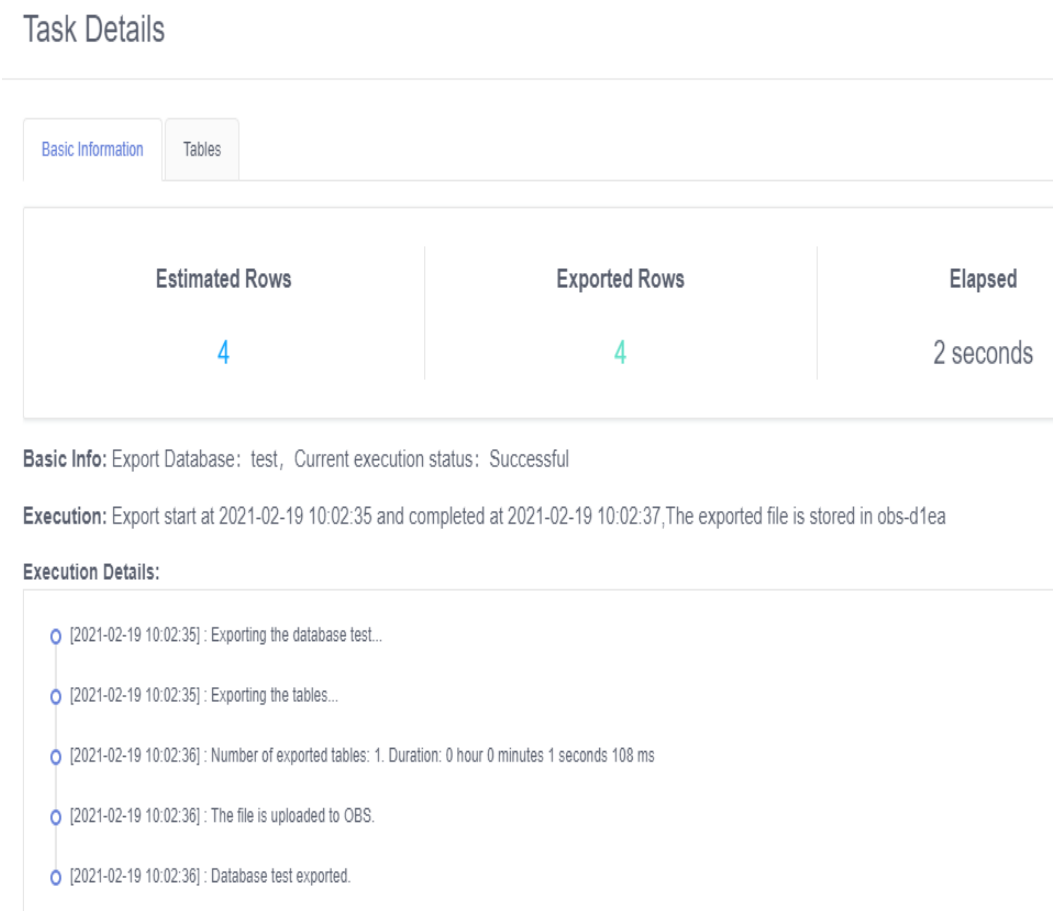
Step 9 After settings are complete, click **OK**.

Step 10 In the task list, view the task ID, type, status, and progress.

Figure 4-19 Task list

Step 11 Click **Details** in the **Operation** column to view task details.

Figure 4-20 Task details



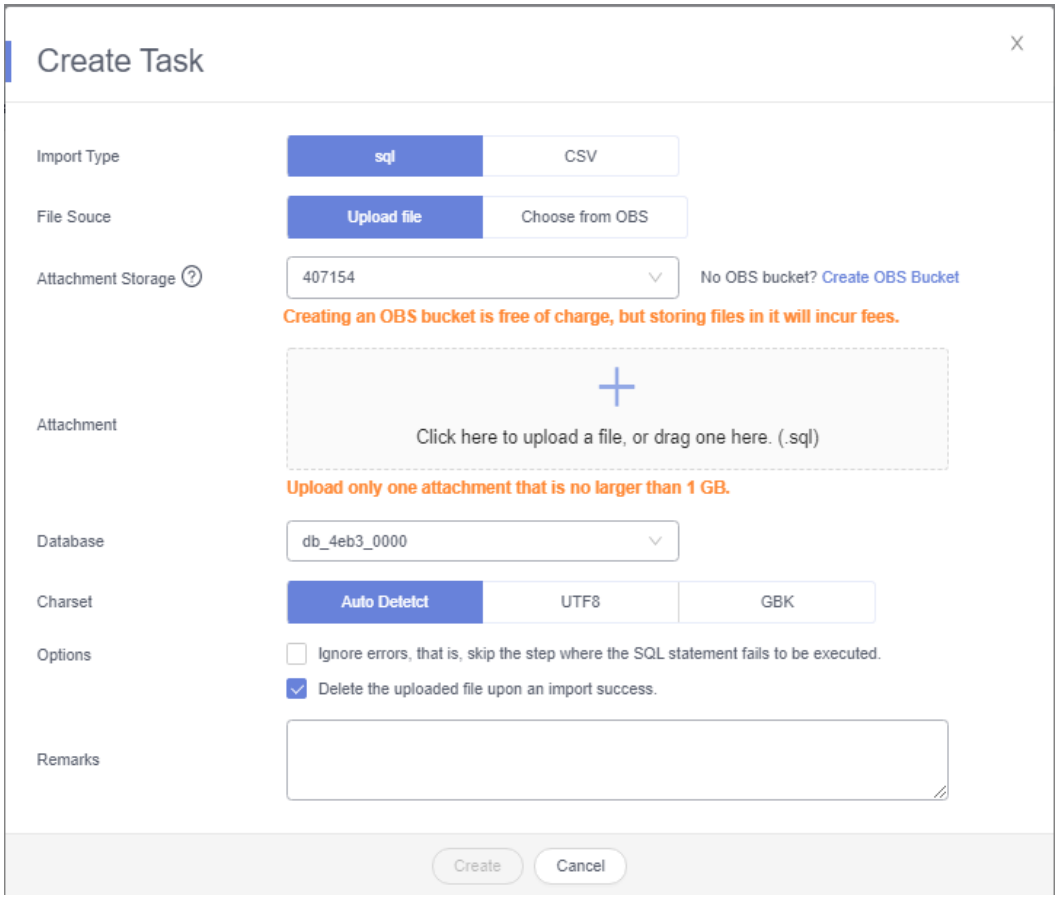
----End

Importing Data

Step 1 On the top menu bar, choose **Import and Export** > **Import**.

Step 2 Import a file from your local PC or an OBS bucket.

Figure 4-21 Creating an import task



- From your local PC

In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.

NOTE

- To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
- If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database.

- From an OBS bucket

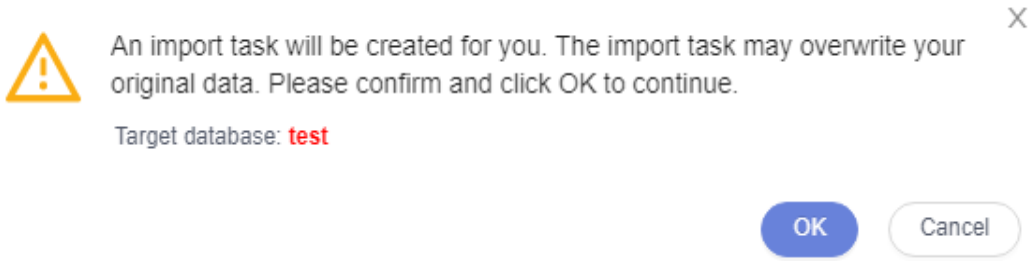
  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

NOTE

The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting the import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

Figure 4-22 Confirmation page



Step 4 View the import progress in the task list or check task details.

----End

4.4 Performance Tuning

4.4.1 High CPU Usage of RDS for SQL Server Instances

If the CPU usage is high or close to 100% when you use RDS for SQL Server, data read/write processing and network connection will slow down, and errors will be reported during deletion, affecting your services.

Solution

Analyze slow SQL logs and CPU usage to locate and optimize slow queries.

1. View the slow SQL logs to check for slowly executed SQL queries and view their performance characteristics (if any) to locate the cause.

   For details on viewing RDS for SQL Server logs, see **Viewing and Downloading Slow Query Logs**.
2. View the CPU usage of your RDS instance to facilitate problem locating.

   For details about supported monitoring metrics, see **Configuring Displayed Metrics**.
3. Create read replicas to offload read pressure from the primary DB instance.
4. Add indexes for associated fields in multi-table association queries.
5. Do not use the SELECT statement to scan all tables. You can specify fields or add the WHERE condition.

4.4.2 Full Storage of RDS for SQL Server Instances

When the storage usage of an instance reaches 97% or higher, RDS sets the instance to read-only to protect the disk from becoming abnormal. The instance status changes to **Storage full**.

Possible Causes

- Service volume increase
- Too large LDF files in some databases
- Temporary database (tempdb) occupying too much storage

Solution

- **Scale up the storage.**
  a. In the instance list, choose **More** > **Scale Storage Space**.
  b. On the displayed page, the system determines the minimum space required.

  Figure 4-23 Scaling up storage space

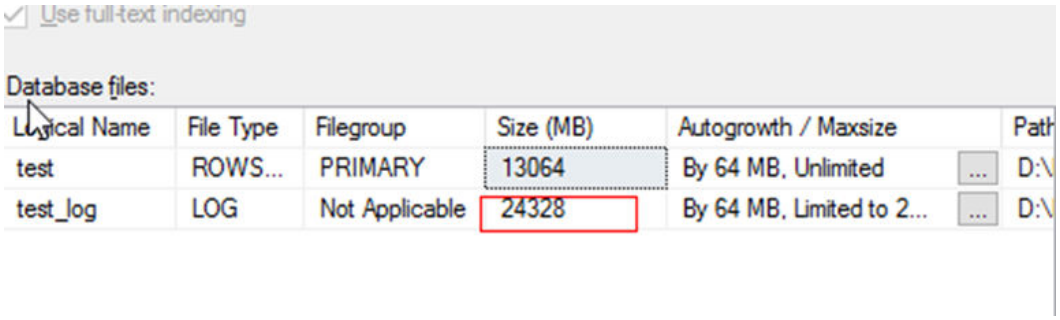
  c. After the scale-up is successful, check that the instance status becomes **Available**.

  Figure 4-24 Checking the instance status

- If the full storage is caused by too large LDF files in some databases, shrink the databases.
  For details, see **Shrinking Databases**.

  Figure 4-25 Checking the LDF file size

- If the tempdb database occupies too much space, perform the following operations:
  a. Query the tempdb database size.

```
SELECT name AS FileName,
size*1.0/128 AS FileSizeInMB,
CASE max_size
WHEN 0 THEN 'Autogrowth is off.'
WHEN -1 THEN 'Autogrowth is on.'
ELSE 'Log file grows to a maximum size of 2 TB.'
```



```
END,
growth AS 'GrowthValue',
'GrowthIncrement' =
CASE
WHEN growth = 0 THEN 'Size is fixed.'
WHEN growth > 0 AND is_percent_growth = 0
THEN 'Growth value is in 8-KB pages.'
ELSE 'Growth value is a percentage.'
END
FROM tempdb.sys.database_files;
```

b. Shrink the tempdb database by referring to **Shrinking Databases**. If the tempdb database is frequently used, the storage usage cannot be effectively reduced.

c. In the instance list, choose **More** > **Reboot** to reboot the instance. In this way, the free space of tempdb will be released.

   After the reboot, the instance status becomes **Available**.

NOTICE

An instance reboot can resolve the full storage problem caused by tempdb. But if the full storage is caused by large LDF files in some databases, a reboot does not work. It may cause databases with large transaction logs to enter the **inRecovery** state, in which the databases cannot be accessed for a long time. For higher security, you are advised to scale up storage.

4.5 Instance Lifecycle

4.5.1 Buying a Same DB Instance as an Existing DB Instance

Scenarios

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

NOTE

- You can buy 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Buy 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 how to buy a Microsoft SQL Server DB instance, see **Buying a DB Instance**.

Step 6 Confirm the instance specifications.

- For pay-per-use DB instances, click **Submit**.
- For yearly/monthly DB instances, click **Pay Now**.

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 **Instances** page.

----End

4.5.2 Stopping an Instance

Scenarios

If you use DB instances only for routine development, you can temporarily stop pay-per-use instances to save money. You can stop an instance for up to 15 days.

Billing

After a DB instance is stopped, the VM where the DB instance is located is no longer billed. Other resources, including EIPs, storage resources, and backups, are still billed.

Constraints

- Only pay-per-use instances using cloud SSDs can be stopped. RDS instances in a DCC cannot be stopped.
- If you stop a primary instance, read replicas (if there are any) will also be stopped. They are stopped for up to 15 days. You cannot stop a read replica without stopping the primary instance.
- A stopped instance cannot be deleted through the console.
- Stopping a DB instance will also stop its automated backups. After the DB instance is started, a full backup is automatically triggered.
- If you do not manually start your stopped instance after 15 days, your instance is automatically started during the next maintenance window. For details about the maintenance window, see **Changing the Maintenance Window**. To start an instance, see **Starting an Instance**.
- A stopped pay-per-use instance may fail to start due to insufficient ECS resources. In this case, try again later or **restore data to a new DB instance** using the latest backup. If you need assistance, **submit a service ticket**.

Procedure

Step 1 Log in to the management console.

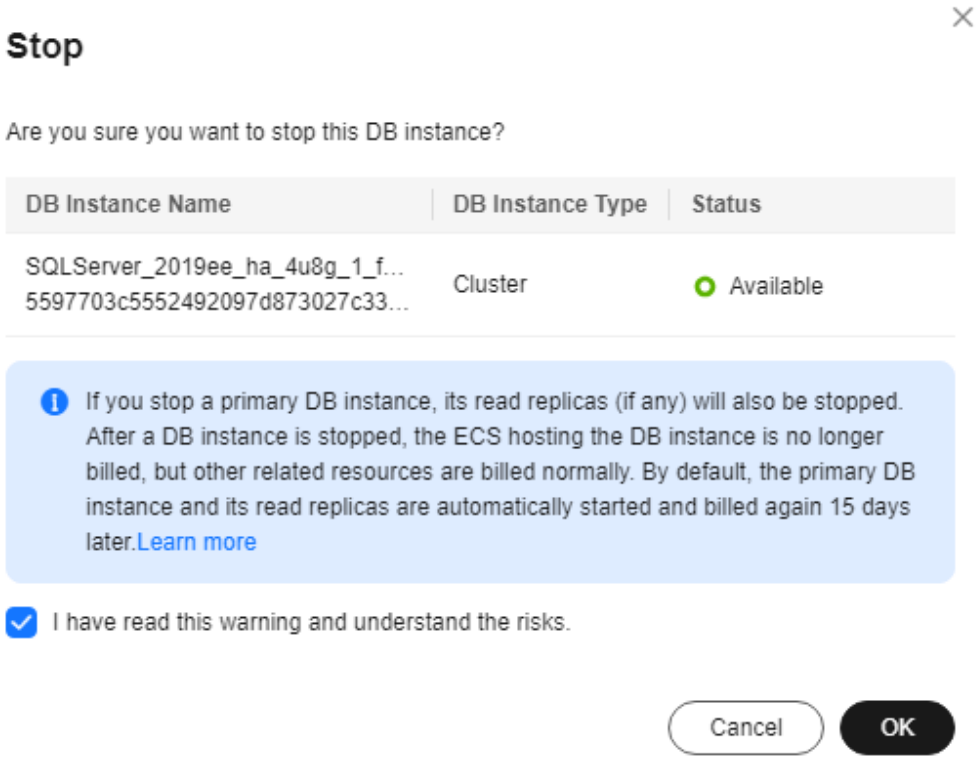
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary instance that you want to stop and choose **More** > **Stop** in the **Operation** column.

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

Figure 4-26 Stopping an instance



Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the instance list and view the status of the instance. If the status is **Stopped**, the instance is stopped successfully.

----End

4.5.3 Starting an Instance

Scenarios

You can stop your instance temporarily to save money. After stopping your instance, you can restart it to begin using it again.

Billing

After a DB instance is started, the VM where the DB instance is located is billed again.

Constraints

- If you start a primary instance, read replicas (if there are any) will also be started.
- When a stopped DB instance is started, a full backup is automatically triggered.
- Only instances in **Stopped** state can be started.
- A stopped pay-per-use instance may fail to start due to insufficient ECS resources. In this case, try again later or **restore data to a new DB instance** using the latest backup. If you need assistance, **submit a service ticket**.

Procedure

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

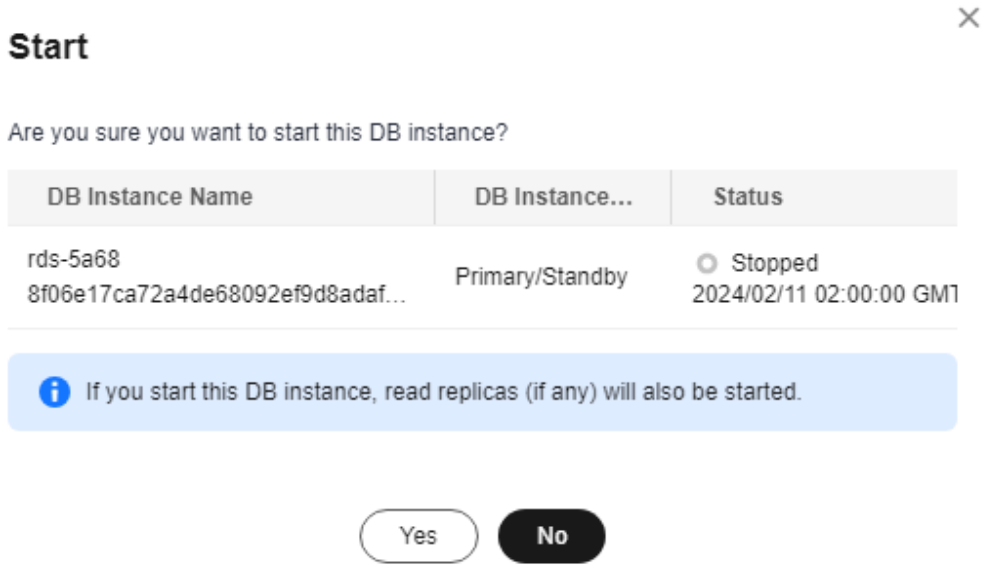
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary instance that you want to start and choose **More** > **Start** in the **Operation** column.

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

Figure 4-27 Starting an instance



Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the instance list and view the status of the instance. If the status is **Available**, the instance is started successfully.

----End

4.5.4 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. You can reboot a single DB instance or multiple DB instances at a time.

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.

Rebooting a DB Instance or Read Replica

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **Instances** page to go to the **Overview** page. In the upper right corner, click **Reboot**.

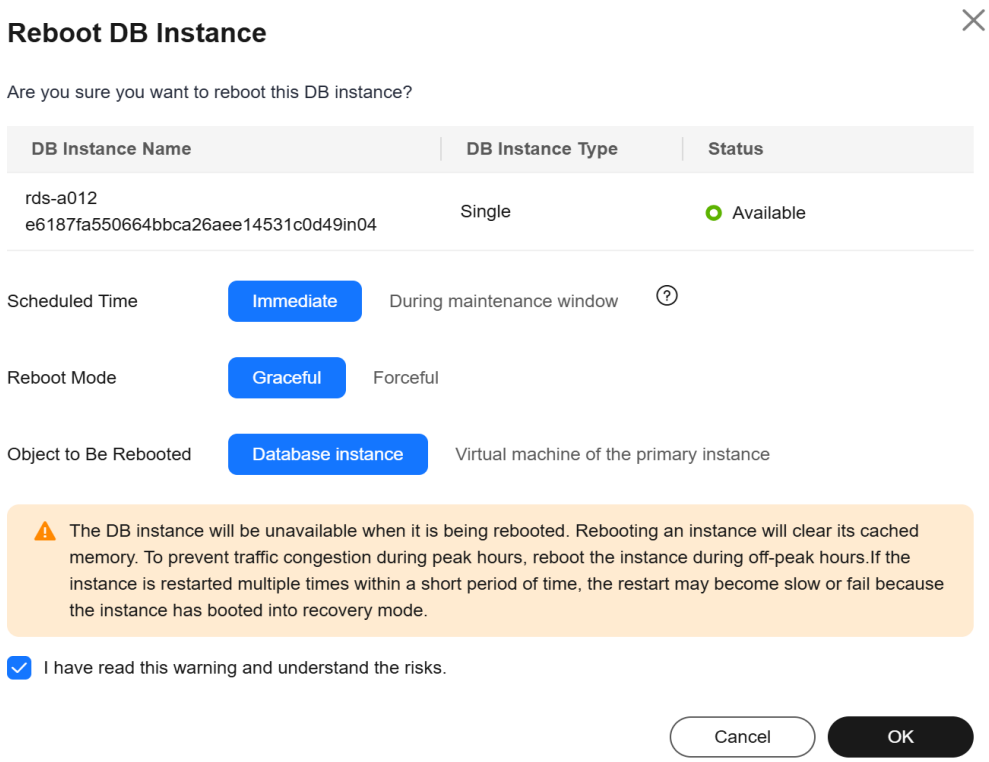
For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 5 In the displayed dialog box, select a scheduled time and click **OK**.

- **Immediate**: RDS reboots the instance immediately.
- **During maintenance window**: RDS will reboot the instance during the maintenance window you configured. To use this function, **submit a service ticket** to apply for required permissions.

  If you select **During maintenance window**, you can further click **Modify** under the option to change the maintenance window to a preferred time.
- **Virtual machine of the primary instance**: If the underlying VM where the DB instance is located has been running for a long time, the memory usage is high, and the paged pool is too large, you can select **Virtual machine of the primary instance** for **Object to Be Rebooted**. Then the underlying VM will be rebooted. Rebooting the VM can interrupt your workload. After the VM is rebooted, the memory is restored and the paged pool space is released. To use this function, **submit a service ticket** to apply for required permissions.

Figure 4-28 Rebooting a DB instance



Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

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

----End

Rebooting DB Instances or Read Replicas in Batches

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select one or more DB instances or read replicas (maximum: 50) to be rebooted and choose **More** > **Reboot** above the DB instance list.

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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list and view the statuses of the DB instances. If their statuses are **Available**, they have rebooted successfully.

----End

4.5.5 Selecting Displayed Items

Scenarios

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

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

- **Table Text Wrapping**: If you enable this function, excess text will move down to the next line.
- **Operation Column**: If you enable this function, the **Operation** column is always fixed at the rightmost position of the table.
- The following items can be displayed: **Name/ID**, **Description**, **DB Instance Type**, **DB Engine Version**, **Status**, **Disk Encryption** (**submit a service ticket** to apply for required permissions), **Billing Mode**, **Floating IP Address**, **Private Domain Name**, **IPv6 Address**, **Read/Write Splitting Address**, **Proxy ID**, **Enterprise Project**, **Created**, **Database Port**, **Storage Type**, **Tags**, and **Operation**.

----End

4.5.6 Exporting DB Instance Information

Scenarios

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

Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click **Export** above the DB instance list. By default, information about all DB instances is exported. In the displayed dialog box, you can select the items to be exported and click **OK**.

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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, filter DB instances by DB engine, DB instance name, DB instance ID, DB instance tag, enterprise project, or floating IP address, or select the DB instances to be exported, and click **Export** above the DB instance list. In the displayed dialog box, select the items to be exported and click **OK**.

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

----End

4.5.7 Deleting a Pay-per-Use DB Instance or Read Replica

Scenarios

To release resources, you can delete DB instances or read replicas billed on the pay-per-use basis as required on the **Instances** page. (To delete DB instances or read replicas billed on the yearly/monthly basis, you need to unsubscribe from the order. For details, see **Unsubscribing from a Yearly/Monthly Instance**.)

Billing

- You will not be billed for the instances that were not successfully created.
- If you delete a pay-per-use DB instance, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and generate additional costs.

Constraints

- DB instances cannot be deleted when operations are being performed on them. They can be deleted only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be deleted.
- 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 **rebuild a DB instance** that was deleted up to 7 days ago from the recycle bin.
- You can use a manual backup to restore a DB instance. For details, see **Restoring from Backup Files to RDS for SQL Server Instances**.

Deleting a Pay-per-Use DB Instance

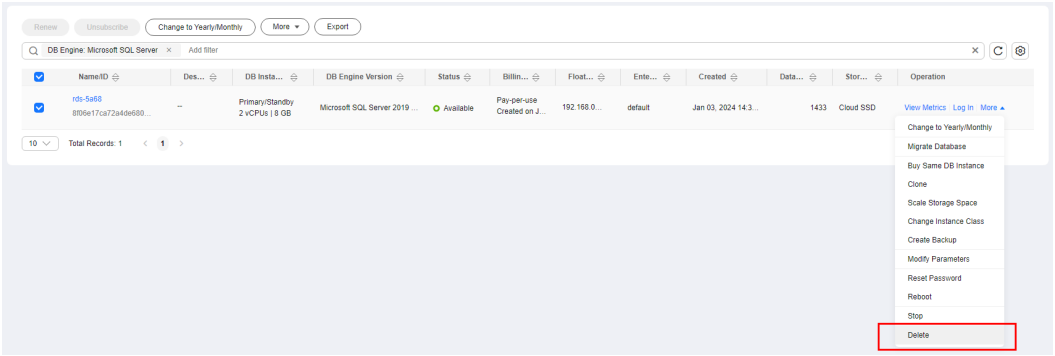
Step 1 **Log in to the management console**.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the primary DB instance to be deleted and click **More** > **Delete** in the **Operation** column.

Figure 4-29 Deleting a DB instance



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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 7 Refresh the DB instance list later to confirm that the deletion was successful.

----End

Deleting a Pay-per-Use Read Replica

Step 1 Log in to the management console.

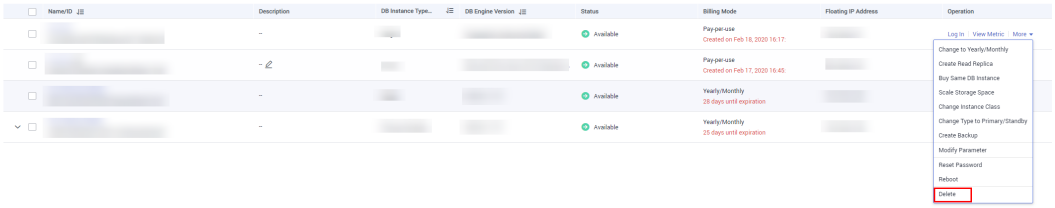
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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.

Figure 4-30 Deleting a read replica



Step 6 In the displayed dialog box, click **Yes**.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 8 Refresh the DB instance list later to check that the deletion is successful.

----End

4.5.8 Recycling a DB Instance

Scenarios

RDS allows you to move unsubscribed yearly/monthly DB instances and deleted pay-per-use DB instances to the recycle bin. You can rebuild a DB instance that was deleted up to 7 days ago from the recycle bin.

If resources are not renewed after expiration, you can rebuild DB instances from the recycle bin to restore data.

Constraints

- The recycle bin is free for use.
- Read replicas cannot be moved to the recycle bin.
- The recycle bin is enabled by default and cannot be disabled.
- After you submit a deletion request for your DB instance, a full backup will be performed. You can rebuild the DB instance only after the full backup is complete.

Modifying Recycling Policy

NOTICE

Instances in the recycle bin are retained for 7 days by default. 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.

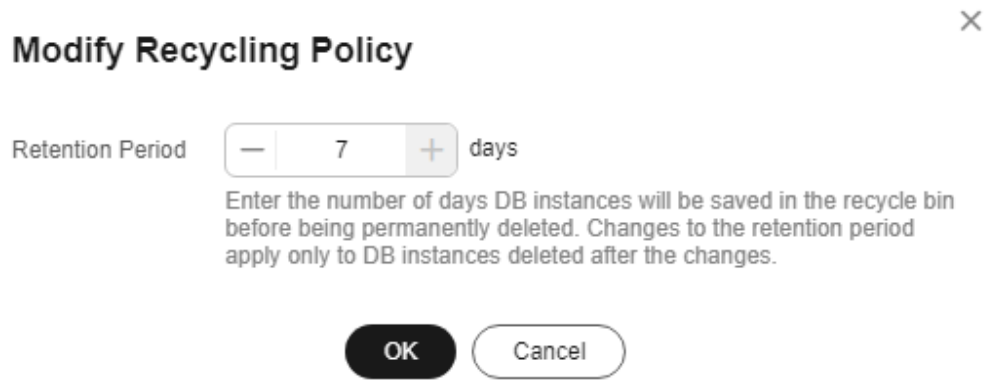
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 On the **Recycle Bin** page, click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances to 1 to 7 days.

Step 6 Then, click **OK**.

Figure 4-31 Modifying the recycling policy



----End

Rebuilding a DB Instance

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

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 On the **Recycle Bin** page, locate the target DB instance to be rebuilt and click **Rebuild** in the **Operation** column.

Step 6 On the **Rebuild DB Instance** page, configure required information and submit the rebuild task. For details, see **Restoring from Backup Files to RDS for SQL Server Instances**.

----End

4.6 Instance Modifications

4.6.1 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **DB Instance Name**, click to edit the instance name.

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

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

Step 5 View the results on the **Overview** page.

----End

4.6.2 Changing a DB Instance Description

Scenarios

After a DB instance is created, you can add a description.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance you wish to edit the description for and click in the **Description** column to make your modification. When you are finished, click **OK**.

Alternatively, click the instance name to go to the **Overview** page. Under **Description**, click to edit the instance description.

NOTE

The DB instance description can include up to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.).

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

Step 5 View the results on the **Overview** page.

----End

4.6.3 Changing the Failover Priority

Scenarios

RDS gives you control over the failover priority of your primary/standby DB instance. You can set it to **Reliability** or **Availability**.

- **Reliability** (default setting): Data consistency is preferentially ensured during a primary/standby failover. This is recommended for applications whose highest priority is data consistency.
- **Availability**: Database availability is preferentially ensured during a primary/standby failover. This is recommended for applications that require databases to provide uninterrupted online services.

Primary/Standby Replication Mode

- RDS for SQL Server uses synchronous replication between the primary and standby DB instances by default. SQL Server 2017 Enterprise Edition and 2019 Enterprise Edition use AlwaysOn availability groups (AGs). Other editions use database mirroring.
- RDS for SQL Server uses asynchronous replication between the primary DB instance and read replicas by default.

Procedure

Step 1 Log in to the management console.

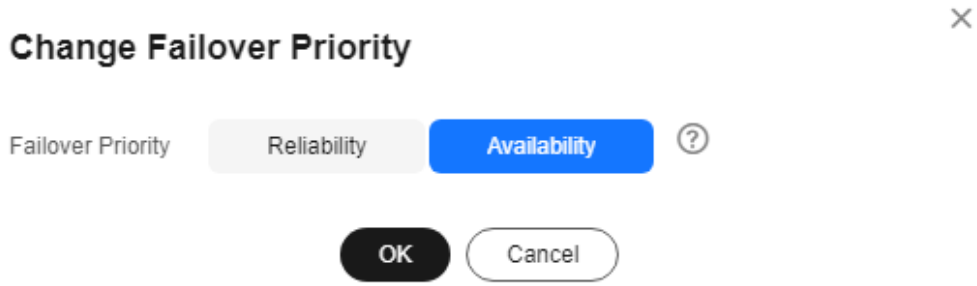
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the primary instance name.

Step 5 On the displayed **Overview** page, find **Failover Priority** and click **Configure** under it. In the displayed dialog box, select a priority and click **OK**.

Figure 4-32 Changing the failover priority



Step 6 View the results on the **Overview** page.

----End

4.6.4 Cloning a DB Instance

Scenarios

Cloning a DB instance means cloning the service data of a primary DB instance within about half an hour, so that you can analyze the data without interrupting services.

The primary DB instance is accessible while being cloned. Data inherited by the cloned instance is the same as that of the primary DB instance from when the cloning action began. If you want a cloned instance that has the same data as the original after the cloning completes, disable access to the primary DB instance before starting to clone the instance.

Constraints

- You can clone a DB instance only when your account balance is greater than or equal to $0 USD.
- To clone a DB instance, you need to **submit a service ticket** to apply for the required permissions.
- If the primary DB instance is kept accessible while being cloned, **data inherited by the cloned instance will match that of the source instance from when the cloning began, not from when it finished**.
- The storage type and space of the new instance must be the same as those of the primary DB instance.
- The AZ of the new instance must be the same as that of the primary DB instance.
- The parameter template, DB engine version, and DB instance type of the new instance must be the same as those of the primary DB instance.
- Microsoft SQL Server 2008 R2 Standard Edition and read replicas do not support instance cloning.
- The following operations cannot be performed on a cloning primary DB instance:
  - Changing the instance class
  - Enabling Transparent Data Encryption (TDE)
  - Enabling or disabling FileStream
  - Migrating data
  - Restoring data to the primary DB instance
  - Modifying MSDTC configurations
  - Deleting the primary DB instance
  - Switching over the primary and standby DB instances
  - Resetting a password
  - Changing DB instance type from single to primary/standby
  - Upgrading the version

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance to be cloned and choose **More** > **Clone** in the **Operation** column.

Step 5 On the displayed page, keep the configurations the same as the original DB instance or change them as required. Then, click **Next**.

Step 6 Confirm the instance configurations.

- For pay-per-use DB instances, click **Submit**.

- For yearly/monthly DB instances, click **Pay Now**.

Step 7 Refresh the DB instance list and check the clone result. If the status of the new instance is **Available**, the clone was successful.

You can manage the cloned instance on the **Instances** page.

----End

4.6.5 Changing a DB Instance Class

Scenarios

You can change the instance class (vCPUs and memory) of a DB instance as required.

Constraints

- You can change the DB instance class only when your account balance is greater than or equal to $0 USD.
- An instance cannot be deleted while its instance class is being changed.
- If the underlying ECS uses an architecture different from that of the target instance class, the instance class cannot be changed and a message will be displayed, indicating that instance class changes between the QingTian architecture and non-QingTian architecture are not allowed. For details about how to change ECS specifications, see **General Operations**.
- The dedicated instance class cannot be changed to any other instance class. For example, a general-purpose instance can be changed to a dedicated instance, but a dedicated instance cannot be changed to a general-purpose instance.
- To change the storage type to **Extreme SSD V2**, **submit a service ticket** to request required permissions.
- You can scale up or down the compute and memory capacity of RDS for SQL Server DB instances as needed.
- Only the instance classes of pay-per-use DB instances can be changed automatically during the maintenance window. To use this function, **submit a service ticket**.
- If you have selected **Maintenance Window** for **Scheduled Time**, the DB instance will be rebooted during the instance class change time and services will be interrupted. You are advised to set the maintenance window to off-peak hours.
- 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.

Instance Class or Storage Type Change

- You can change a general-purpose instance to a dedicated instance, but a dedicated instance cannot be changed to a general-purpose instance.
- You can change the storage type of an instance:
  - From high I/O to ultra-high I/O or cloud SSD.

- From ultra-high I/O to extreme SSD or extreme SSD V2.
- From cloud SSD to extreme SSD or extreme SSD V2.
- From extreme SSD to extreme SSD V2.
- To use the extreme SSD V2 storage type, **submit a service ticket** to apply for it.

Procedure

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

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

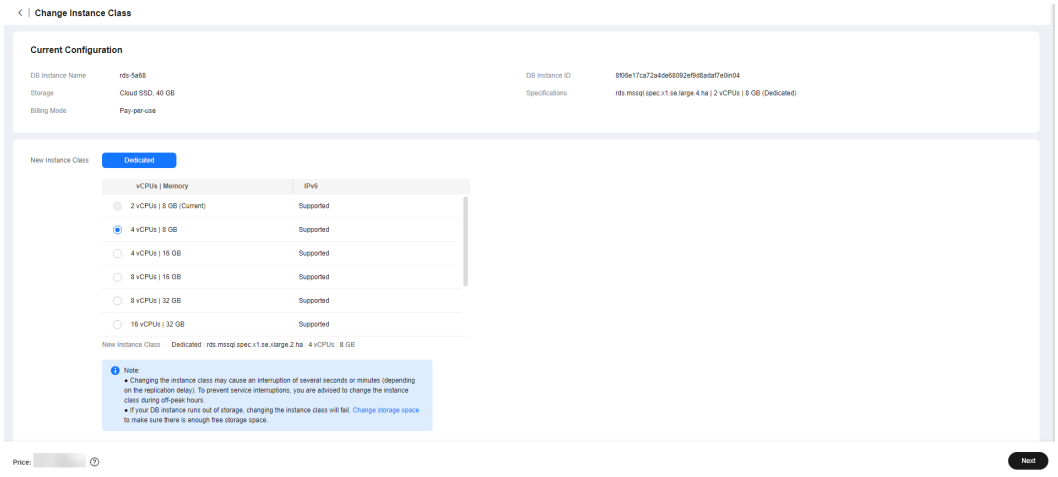
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Alternatively, click the instance name to go to the **Overview** page. Under **Instance Class**, click **Configure**.

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

Figure 4-33 Changing a DB instance class



To change the storage type to **Extreme SSD V2**, **submit a service ticket**. If the storage type is changed to **Extreme SSD V2**, you need to configure the IOPS. IOPS is separately billed on a pay-per-use basis.

If you select **Maintenance Window** for **Scheduled Time**, the DB instance will be rebooted during the instance class change time and services will be interrupted. You are advised to set the maintenance window to off-peak hours.

DB instances in a DCC only support the general-enhanced instance class.

Step 6 Confirm the specifications.

- If you need to modify your settings, click **Previous**.

- For pay-per-use DB instances, click **Submit**.

  To view the cost incurred by the DB instance class change, choose **Billing & Costs** > **Bills** in the upper right corner.
- For yearly/monthly DB instances:
  - If you intend to scale down the DB instance class, click **Submit**.

    The refund is automatically returned to your account. You can click **Billing** in the upper right corner and then choose **Orders** > **My Orders** in the navigation pane on the left to view the details.
  - If you intend to scale up the DB instance class, click **Pay Now**. The scaling starts only after the payment is successful.

Step 7 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 **Instances** page is **Changing instance class**. After a few minutes, click the instance name and view the instance class displayed on the **Overview** page to check that the change is successful.

NOTICE

After you change the instance class of an RDS for SQL Server instance, the value of **max server memory** will be changed accordingly. You are advised to set **max server memory** to Memory size (GB) x 1024 x 0.85 – 1.5 x 1024. For example, if your memory is 4 GB, set **max server memory** to 1946 MB (4 x 1024 x 0.85-1.5 x 1024).

----End

4.6.6 Scaling Up Storage Space

Scenarios

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

A 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**.

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

- You can scale up storage space only when your account balance is greater than or equal to $0 USD.
- 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. If you want to increase the storage upper limit, **submit a service ticket**

- The maximum allowed storage space for some Microsoft SQL Server DB instances is 2,000 GB because of constraints of the windows disk size. The actual maximum allowed storage space depends on the information displayed on the console.
- 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.
- If you scale up a DB instance with the disk encrypted, the expanded storage space will be encrypted using the original encryption key.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 instance name to enter the **Overview** page. In the **Storage & Backup** area, click **Scale Storage Space**.
- If the storage space is full, locate the target DB instance on the **Instances** page and click **Scale** in the **Status** column.

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.

The maximum allowed storage space for some Microsoft SQL Server DB instances is 2,000 GB because of constraints of the windows disk size. The actual maximum allowed storage space depends on the information displayed on the console.

Step 6 Confirm specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify the settings, click **Submit** for a pay-per-use instance or click **Pay Now** for a yearly/monthly instance.

Step 7 View the scale-up result.

Scaling up storage space takes 3-5 minutes. During this time period, the status of the DB instance on the **Instances** page will be **Scaling up**. After a while, click the instance name and view the new storage space on the displayed **Overview** page to verify that the scale-up is successful.

----End

4.6.7 Configuring Autoscaling

Scenarios

With storage autoscaling enabled, when RDS detects that you are running out of database space, it automatically scales up your storage.

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

Constraints

- You can enable storage autoscaling only when your account balance is greater than or equal to $0 USD.
- The storage space can be scaled up automatically only when your instance status is **Available** or **Storage full**.
- To enable storage autoscaling, you need to **submit a service ticket** to apply for required permissions.
- The maximum allowed storage is 10,000 GB. It varies depending on the storage type.
- For primary/standby DB instances, autoscaling the storage for the primary DB instance will also autoscale the storage for the standby DB instance.
- Storage autoscaling is unavailable when the DB instance is changing instance class or rebooting.
- If a yearly/monthly DB instance has pending orders, it will not autoscale.
- Storage of RDS for SQL Server instances cannot be scaled down. Exercise caution when enabling autoscaling.
- There is an upper limit on storage autoscaling. A maximum of two scale-ups are allowed within one hour and a maximum of five scale-ups within one day. If an instance scales up multiple times within a short period of time due to a sharp increase of temporary databases or log files, to reduce the storage usage, you can shrink the databases or log files.

Procedure

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

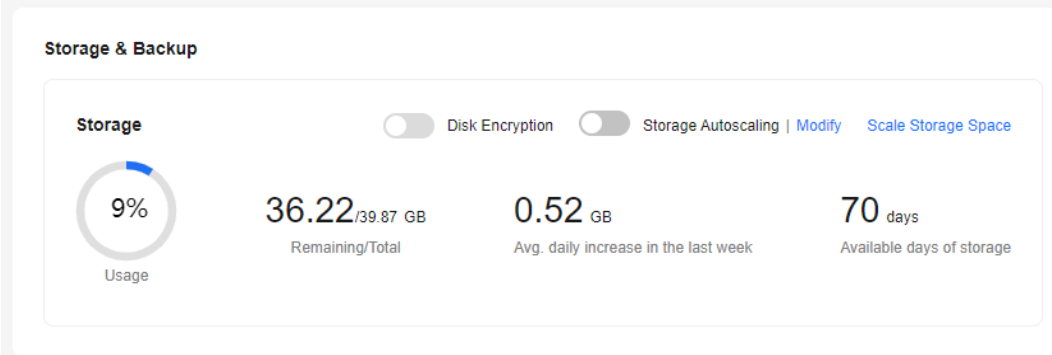
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance or read replica name (click in front of a DB instance to locate its read replica).

Step 5 In the **Storage & Backup** area, toggle on the **Storage Autoscaling** switch.

Figure 4-34 Storage space



Step 6 In the displayed dialog box, set the following parameters.

Figure 4-35 Configuring autoscaling

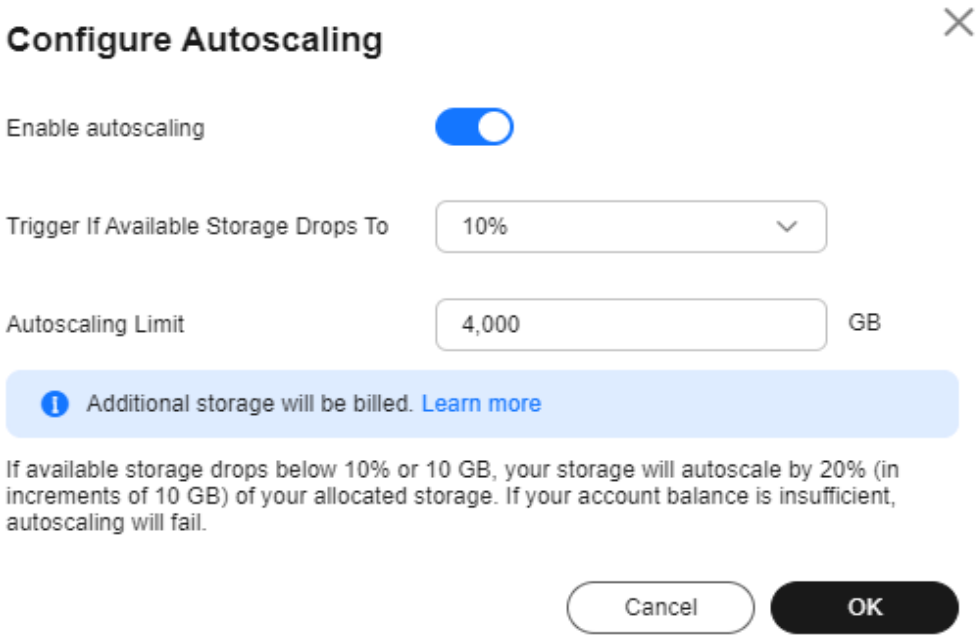


Table 4-5 Parameter description

Parameter	Description
Enable autoscaling	If you turn the toggle switch on, autoscaling is enabled.
Trigger If Available Storage Drops To	If the available storage drops to a specified threshold or 10 GB, autoscaling is triggered.
Autoscaling Limit	The value range is from 40 GB to 10,000 GB. The limit must be no less than the storage of the DB instance.

Step 7 Click **OK**.

----End

4.6.8 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 once or twice. Ensure that your applications support automatic reconnection.

Procedure

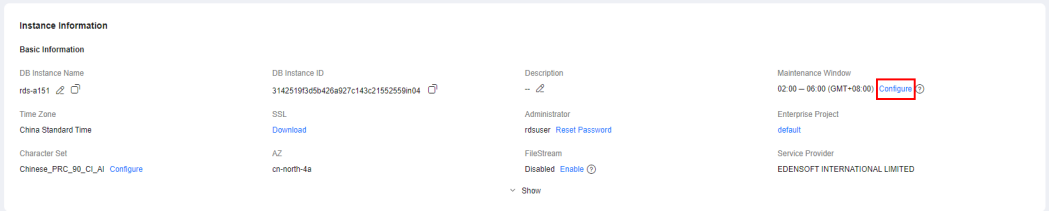
Step 1 **Log in to the management console**.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page. Under **Maintenance Window**, click **Configure**.

Figure 4-36 Changing the maintenance window



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

NOTE

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

----End

4.6.9 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.

Precautions

RDS single DB instances can be changed to primary/standby DB instances, but not the other way around. You can use Data Replication Service (DRS) or the export and import tool of the client to migrate data from primary/standby DB instances to single DB instances.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

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 Check the instance status on the **Instances** 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

4.6.10 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

During a primary/standby switchover, the API for **querying databases** cannot be called.

You can switch the primary and standby instances only when all of the following conditions are met:

- The DB instance is running properly.
- The primary/standby replication is normal.
- The replication delay is less than 5 minutes, and the data on the primary and standby instances is consistent.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target primary/standby instance name to go to the **Overview** page.

Step 5 Under **DB Instance Type**, click **Switch**.

NOTICE

A primary/standby switchover may cause a brief interruption of several seconds or minutes (depending on the replication delay). If transaction logs are generated at a speed higher than 30 MB/s, services will probably be interrupted for several minutes. To prevent traffic congestion, perform a switchover during off-peak hours.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the ***Identity and Access Management User Guide***.

Step 7 In the displayed dialog box, click **OK**.

Step 8 After the switchover is successful, check the status of the DB instance on the **Instances** page.

- During the switchover, 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

4.6.11 Updating the DB Engine and OS of a DB Instance

The DB engine and OS of an RDS for SQL Server instance cannot be upgraded automatically during the maintenance window you specified. To upgrade them, **submit a service ticket**. Huawei Cloud engineers will help you upgrade the DB engine and OS if necessary.

Huawei Cloud installs hot patches as required to fix the vulnerabilities that may have major impacts on the DB engine or OS.

4.7 Read Replicas

4.7.1 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

----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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

4.8 Data Backups

4.8.1 Backup Solutions

RDS supports automated and manual backups. You can periodically back up databases. If a database is faulty or data is damaged, you can restore the database using backups to ensure data reliability.

RDS uses **sysbench** to import data models and a certain amount of data. After data is backed up, the compression ratio is about 80%. The more duplicate data there is, the higher the compression ratio is.

Compression ratio = Space occupied by backup files/Space occupied by data files x 100%

Backup Type

- Full backup: A full backup is to back up all data, even if no data has changed since the last backup.

  Full backups include automated backups and manual backups.
- Incremental backup: Incremental backups refer to transaction log backups. RDS automatically backs up data modifications made after the most recent full or incremental backup every five minutes.

How RDS Backs Up Data

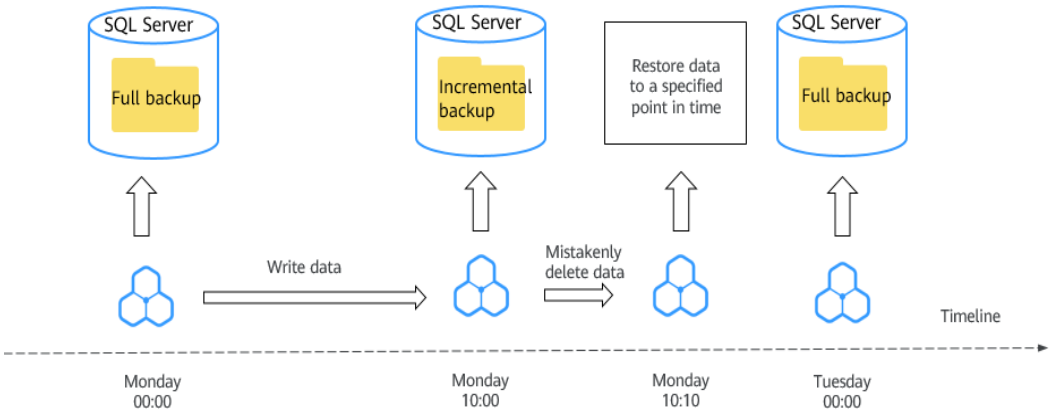
- Single instance

  A single-node architecture, which is more cost-effective than mainstream primary/standby DB instances. After a backup is triggered, data is backed up from the primary instance and stored as a package on OBS. The backup does not take up storage space of the instance.
- Primary/standby instance

  An HA architecture. In a primary/standby pair, each instance has the same instance class. After a backup is triggered, data is backed up from the primary instance and stored as a package on OBS. The backup does not take up storage space of the instance.

  If a database or table in the primary instance is maliciously or mistakenly deleted, the database or table in the standby instance will also be deleted. In this case, you can only use backups to restore the deleted data.

Figure 4-37 How RDS backs up data



Backup Solutions

Table 4-6 describes how to back up data and download backups.

Table 4-6 Backup solutions

Task	Backup Type	Description
Backing up data in the same region	**Automated backups**	RDS automatically creates full backups for your instance during a backup window you specified and saves the backups based on the configured retention period. If necessary, you can restore data to any point in time within the backup retention period. Once the automated backup policy is enabled, a full physical backup is triggered immediately. After that, full backups will be created according to the specified time window and backup cycle.
	Manual backups	Manual backups are user-initiated full backups of instances. The backup method is physical backup. Manual backups will not be deleted until you delete them manually.
	Incremental backups	Transaction logging is enabled for RDS for SQL Server instances by default. RDS automatically backs up data modifications made after the most recent automated or incremental backup every five minutes.
Downloading backups	**Downloading a backup**	You can use OBS Browser+, the browser, or the download URL to download a backup.

Billing

Backups are saved as packages in OBS buckets. Backups occupy backup space in OBS. If the free space RDS provides is used up, the additional space required will be billed. For details, see **How Is RDS Backup Data Billed?**

Deleting Backups

Manual backups and automated backups can be deleted in different ways:

- Manual backups can only be manually deleted.
- Automated backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your **automated backup policy**. Retained backups will be automatically deleted at the end of the retention period.

4.8.2 Configuring an Intra-Region 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 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.

After an automated backup policy is configured, full backups are created based on the time window and backup cycle specified in the policy. The time required for creating a backup depends on how much data there is in the instance. Backups are stored for as long as you specified in the backup policy.

You do not need to set an interval for incremental backup because RDS automatically backs up incremental data every 5 minutes. Incremental backups can be used to restore data to a specific point in time.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

Modifying an Automated Backup Policy

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

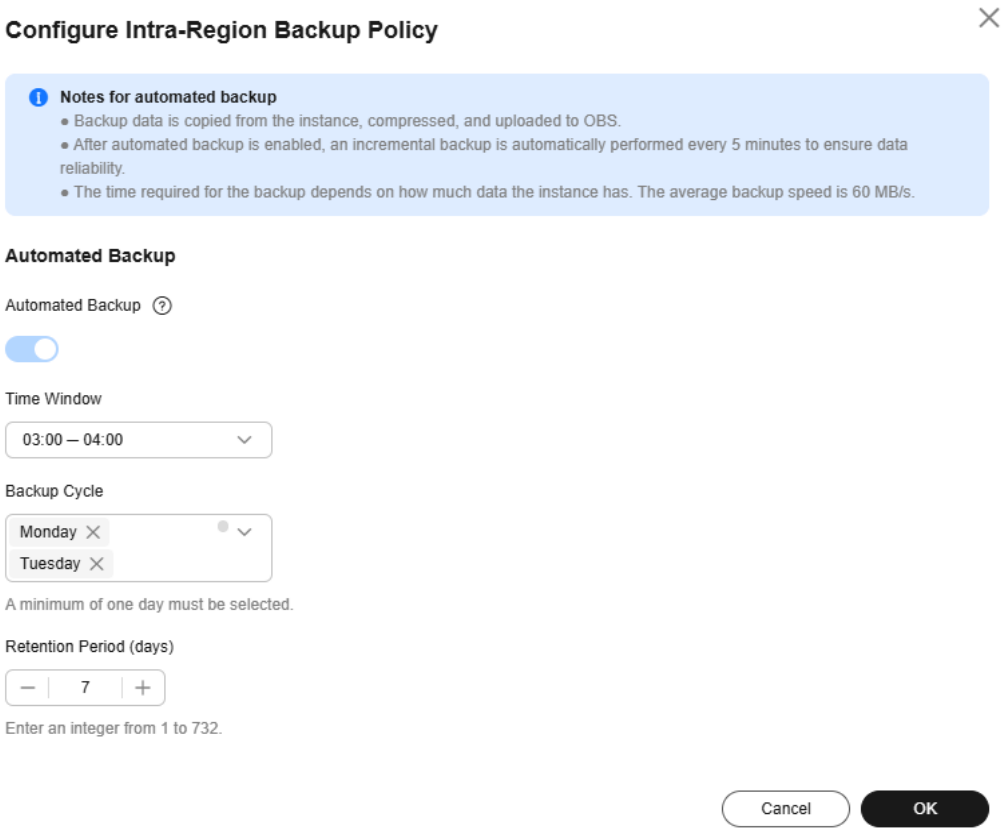
Step 2 Click  in the upper left corner and select a region.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, click **Intra-Region Backup Policies**. You can view the configured backup policy. To modify the backup policy, adjust the parameter values as needed.

Figure 4-38 Modifying an automated backup policy



- **Retention Period**: How many days your automated full backups and incremental backups can be retained. The retention period is from 1 to 732 days and the default value is **7**. To extend the retention period, **submit a service ticket** to request required permissions.
  - Extending the retention period improves data reliability.
  - Reducing the retention period takes effect for existing backups. Any backups (except manual backups) that have expired will be automatically deleted. Exercise caution when performing this operation.

  Policy for automatically deleting automated full backups:

  To ensure data integrity, even after the retention period expires, the most recent backup will be retained, for example, if **Backup Cycle** was set to **Monday** and **Tuesday** and **Retention Period** was set to **2**:

  - The full backup generated on Monday will be automatically deleted on Thursday because:

    The backup generated on Monday expires on Wednesday, but it is the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday.
  - The full backup generated on Tuesday will be automatically deleted on the following Wednesday because:

    The backup generated on Tuesday will expire on Thursday, but as it is the last backup, it will be retained until a new backup expires. The next backup will be generated on the following Monday and will expire on the following Wednesday, so the full backup generated on Tuesday will not be automatically deleted until the following Wednesday.

- **Manual Period**
  - **Permanent**: Manual backups are retained until you manually delete them.
  - **Custom**: You can customize the retention period for manual backups from 1 to 732 days. Manual backups that exceed the retention period will be automatically deleted.
- **Time Window**: A one-hour period the backup will be scheduled for each day, such as 01:00-02:00 or 12:00-13:00. The backup time window indicates when the backup starts. The backup duration depends on the data volume of your instance.

  NOTE

  To minimize the potential impact on services, set the time window to off-peak hours. The backup time is in UTC format. The backup time segment changes with the time zone during the switch between the DST and standard time.
- **Backup Cycle**: Daily backups are selected by default, but you can change it. At least one day must be selected.
- **Scheduled Backup Policy**: If this function is enabled, data is periodically backed up every month, which incurs certain fees. Up to 15 scheduled automated backups can be generated every month and retained for 90 to 732 days. If you want to extend the retention period, **submit a service ticket**.

  NOTE

  - To use the scheduled backup policy function, **submit a service ticket** to request required permissions.
  - Only one automated backup is generated for a DB instance every day. If you enable both **Automated Backup** and **Scheduled Backup Policy**, which one is triggered depends on the retention periods you configured for them. The backup with a longer retention period configured has a higher priority.
  - The time window for scheduled backup is the same every day.
  - Changing the time window and retention period for the scheduled backup policy affects only new backups. Expired backups will be automatically deleted.
  - If the scheduled backup policy is disabled, no new scheduled backups will be generated and expired backups will be automatically deleted.

Step 6 Click **OK**.

----End

4.8.3 Creating a Manual Backup

Scenarios

RDS allows you to create manual backups for an available 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.

You can create manual backups only when your account balance is no less than $0 USD.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see **How Is RDS Backup Data Billed?**

Method 1

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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.

To check whether the backup has been created, you can click in the upper right corner of the page to check the DB instance status. If the DB instance status becomes **Available** from **Backing up**, the backup has been created.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 **Backups** page.

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

----End

4.8.4 Downloading a Backup File

Scenarios

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

Constraints

- Unsynchronized backups are generated only for DB instances running Microsoft SQL Server 2017 Enterprise Edition. If a primary DB instance fails, the standby DB instance is promoted to the new primary instance. During the failover process, a small amount of data may not be synchronized and a differential backup is created for user-created databases on the original

primary DB instance. For more information, see **How Are Unsynchronized Backups Generated for RDS for SQL Server DB Instances?**

- If the size of the backup data is greater than 400 MB, you are advised to use OBS Browser+ to download the backup data.
- When you use OBS Browser+ to download backup data, there is no charge for the generated outbound traffic.

Method 1: Using OBS Browser+

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

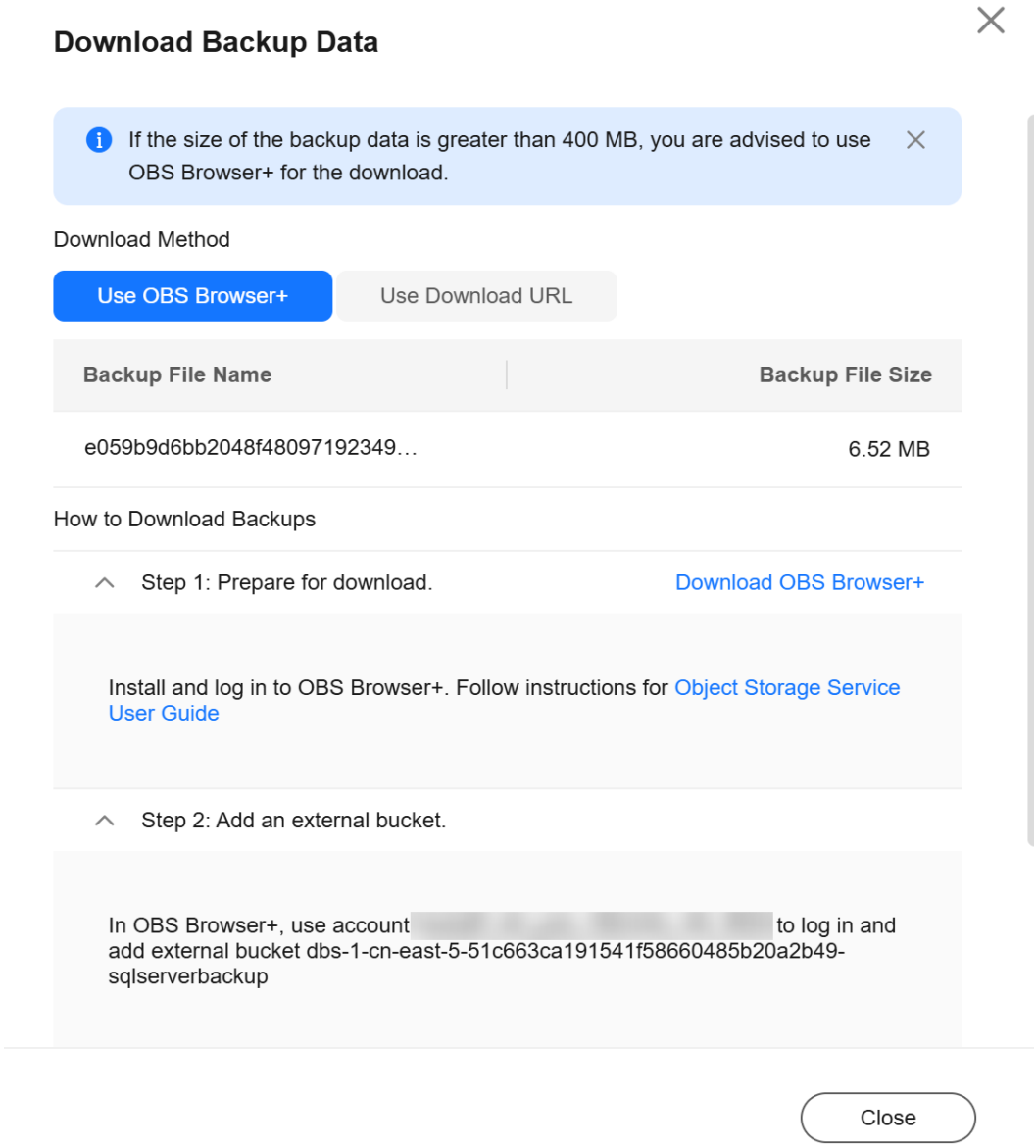
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

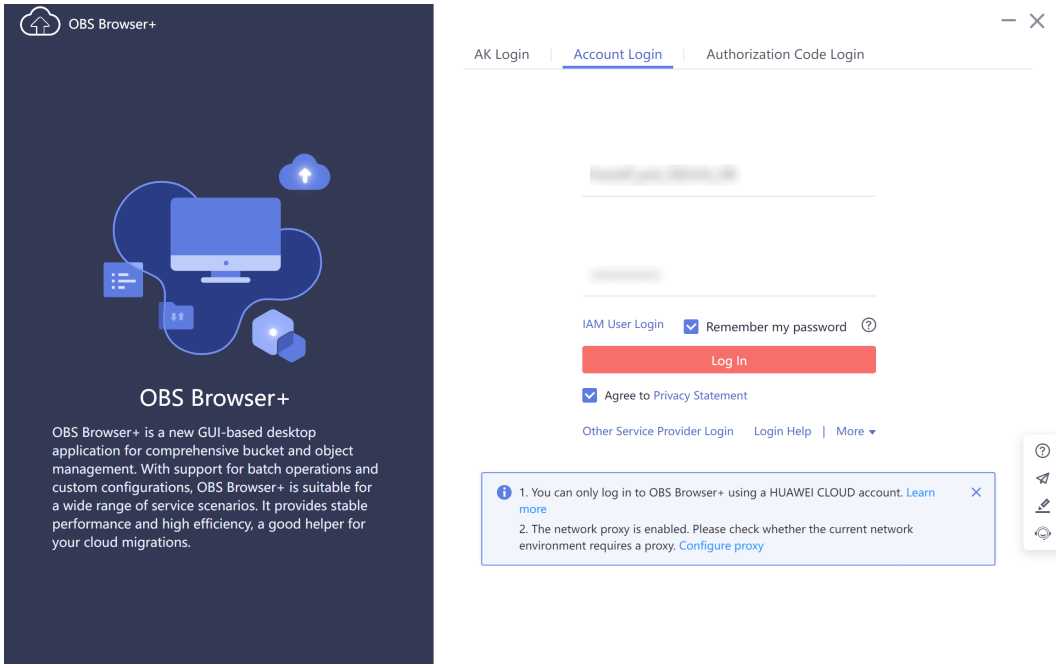
Step 6 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and download the backup as prompted.

Figure 4-39 Using OBS Browser+



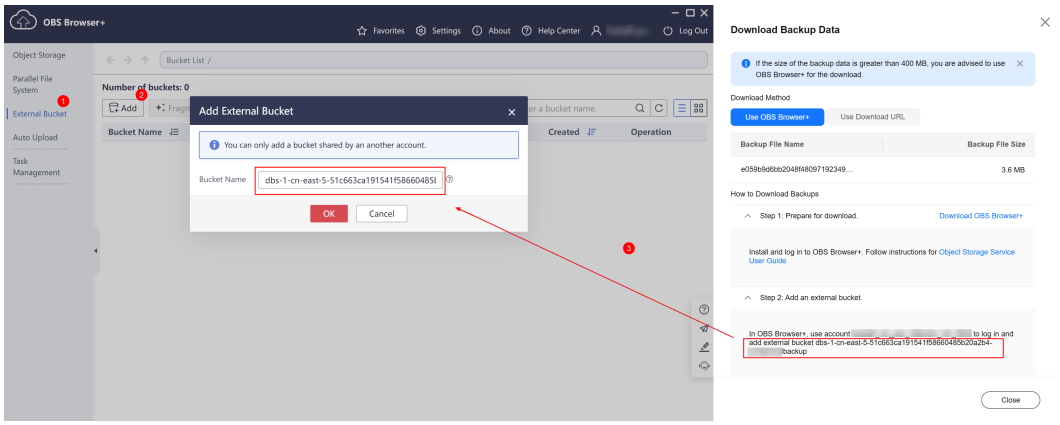
1. Download OBS Browser+ by clicking **Download OBS Browser+** in Step 1 on the download guide page.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

Figure 4-40 Logging in to OBS Browser+



4. Add an external bucket using the bucket name provided in step 2 on the download guide page.

Figure 4-41 Adding an external bucket



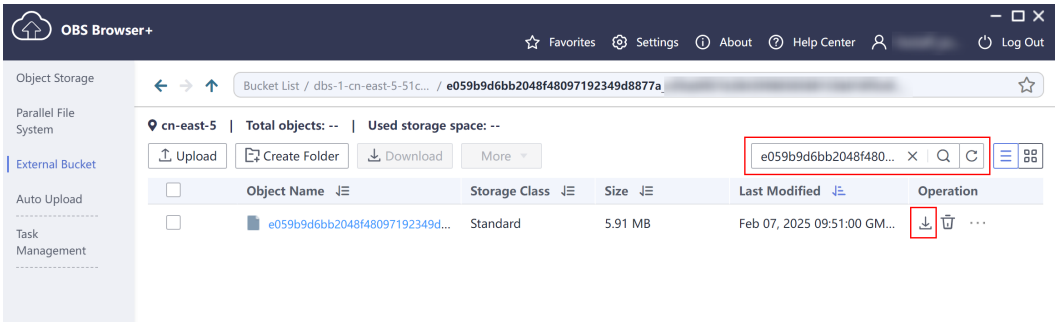
NOTE

If you want to access OBS external buckets across accounts, the access permission is required. For details, see **Granting IAM Users Under an Account the Access to a Bucket and Resources in the Bucket**.

5. Download the backup file.

   On the OBS Browser+ page, click the bucket that you added. In the search box on the right of the object list page, enter the backup file name and start a search. In the search result, locate the target backup and click in the **Operation** column.

Figure 4-42 Downloading a backup



Microsoft SQL Server allows you to download backup files of a specific database.

----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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the backup to be downloaded and click **Download** in the **Operation** column.

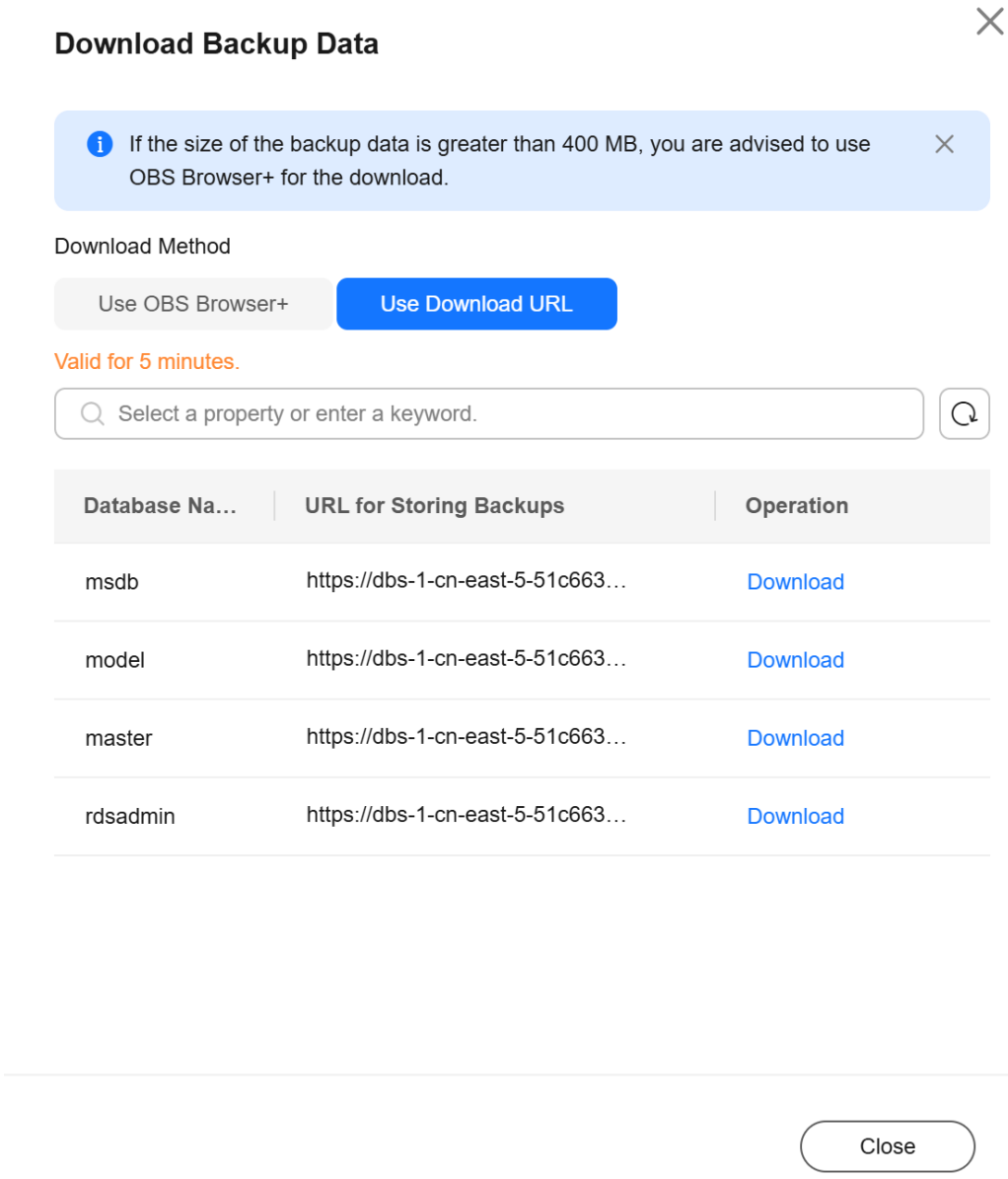
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 In the displayed dialog box, select a method to download backup data.

Figure 4-43 Using the download URL



In the displayed dialog box, select **Use Download URL** for **Download Method**, click to copy the URL, and enter the URL in your browser.

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

4.8.5 Checking and Exporting Backup Information

Scenarios

You can export backup information of RDS DB instances to an Excel file for further analysis. The exported information includes the DB instance name, backup start and end time, backup status, and backup size.

For details about how to export backup data, see **Downloading a Backup File**.

Procedure

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, select the backups you want to export and click to export backup information.

- Only the backup information displayed on the current page can be exported. The backup information displayed on other pages cannot be exported.
- The backup information is exported to an Excel file for your further analysis.

Figure 4-44 Backup information

	A	B	C	D	E	F	G	H	I	J	K	L
1	Backup ID	Backup Name	DB Instance Name	DB Instance ID	Key ID	DB Engine	Backup Type	Backup Method	Backup Time	Status	Size	Description
2	2a5621b105954f3892430969797 8c09bbr04	sqlserver-rds-5ad8-20231127040545837	rds-5ad8	57c90b7c4cd942d98d1344b60fafe338in04	--	Microsoft SQL Server 2019	Automated	Physical backup	Nov 27, 2023	Completed	6.01 MB	--

Step 5 View the exported backup information.

----End

4.8.6 Replicating a Backup

Scenarios

RDS supports replication of automated and manual backups.

Constraints

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

Snapshot-based backups, including CBR snapshot-based backups, cannot be replicated.

Backup Retention Policy

- If a DB instance is deleted, the automated backups created for it are also deleted.
- If an **automated backup policy** is enabled, the automated backups will be deleted after the backup retention period expires.
- 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.
- Replicating a backup does not interrupt your services.

Billing

Backups are saved as packages in OBS buckets. For details, see **How Is RDS Backup Data Billed?**

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required. For details, see **Product Pricing Details**.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name. On the **Backups & Restorations** page, locate the backup to be replicated and click **Replicate** in the **Operation** column.

Alternatively, choose **Backups** in the navigation pane on the left. On the displayed page, locate the 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 carriage return characters or the following special characters: >!<"&'=

Step 6 After the new backup has been created, you can view and manage it on the **Backups** page.

----End

4.8.7 Deleting a Manual Backup

Scenarios

You can delete manual backups to free up backup storage.

Constraints

- Deleted manual backups cannot be recovered.
- Manual backups that are being created cannot be deleted.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup to be deleted and choose **More** > **Delete** in the **Operation** column.

The following backups cannot be deleted:

- Automated backups
- Backups that are being restored
- Backups that are being replicated

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

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

4.9 Data Restorations

4.9.1 Restoration Solutions

If your database is damaged or mistakenly deleted, you can restore it from backup.

Table 4-7 Restoring a DB instance

When you...	Following Steps In...
Restore data to an RDS for SQL Server DB instance	**Restoring from Backup Files to RDS for SQL Server**
	Restoring a DB Instance to a Point in Time

4.9.2 Restoring from Backup Files to RDS for SQL Server 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.

Constraints

- Constraints on restoring data to an existing DB instance (other than the original instance):
  - Restoring to an existing DB instance will overwrite data on it and cause the existing DB instance to be unavailable during the restoration.
  - 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.
  - The storage space of the selected DB instance must be no less than that of the original DB instance. Otherwise, data will not be restored.
  - The time zone of the selected DB instance must be the same as that of the original DB instance. Otherwise, data inconsistency may occur.
  - DB instances with the TDE function enabled cannot be restored from backups to existing DB instances.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Backups** page, locate the target backup and click **Restore** in the **Operation** column.

Alternatively, click the target DB instance on the **Instances** page. In the navigation pane, choose **Backups & Restorations**. On the displayed page, locate the backup to be restored and click **Restore** in the **Operation** column.

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

1. Select a restoration method.
   - Restore to Existing

     Select an existing DB instance and click **Next**.

     If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

     Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.
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 (-), underscores (_), and periods (.) are allowed.

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

- Restore to Existing

  On the **Instances** 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

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

Backup data cannot be restored to original RDS for SQL Server DB instances. If you need to restore data to your original DB instance, restore backup data to a new or an existing DB instance and then **migrate the backup data to the original instance using DRS** or **change the floating IP address of the new DB instance to that of the original instance**.

FAQs

How Can I Restore Data If No Backup Is Available?

4.9.3 Restoring from Backup Files to a Self-Built SQL Server Database Using SSMS

RDS for SQL Server backups include data backups and incremental backups (log backups) in the .bak format. The .bak files can be used to restore data to a self-managed database.

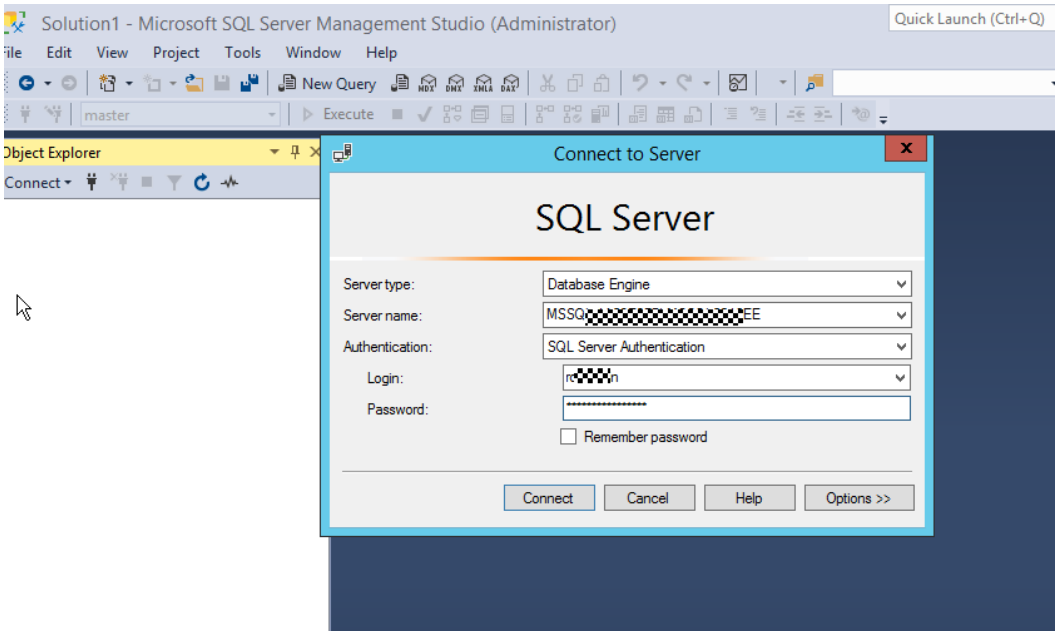
Prerequisites

You have downloaded the .bak files from the cloud to a local path of a self-managed database.

Restoring a Data Backup

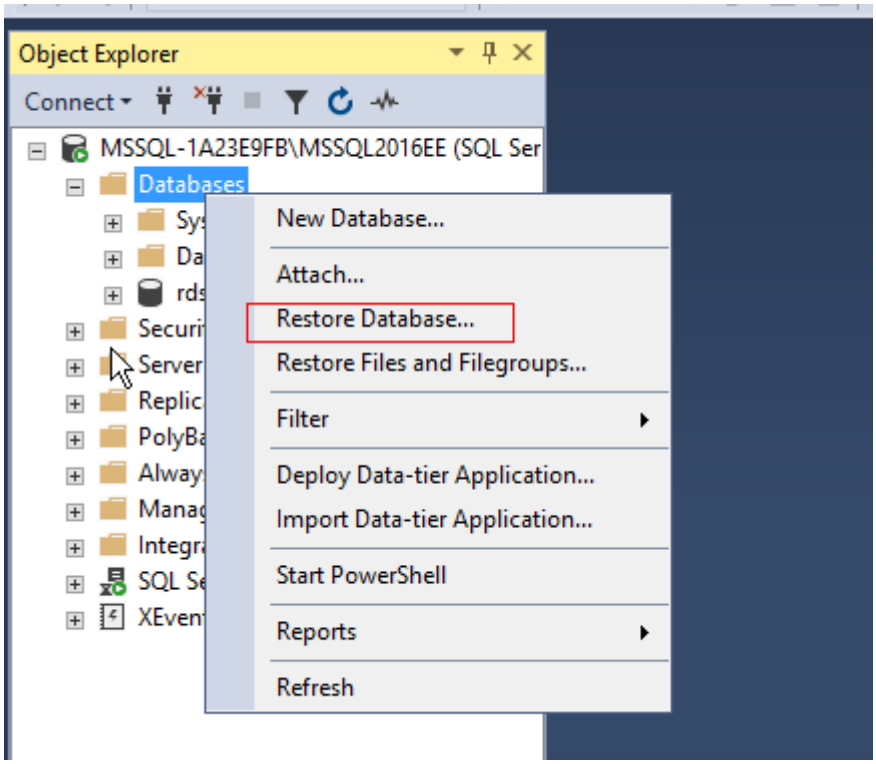
Step 1 Use the Microsoft official tool SQL Server Management Studio (SSMS) to log in to a self-managed database.

Figure 4-45 Logging in to a self-managed database



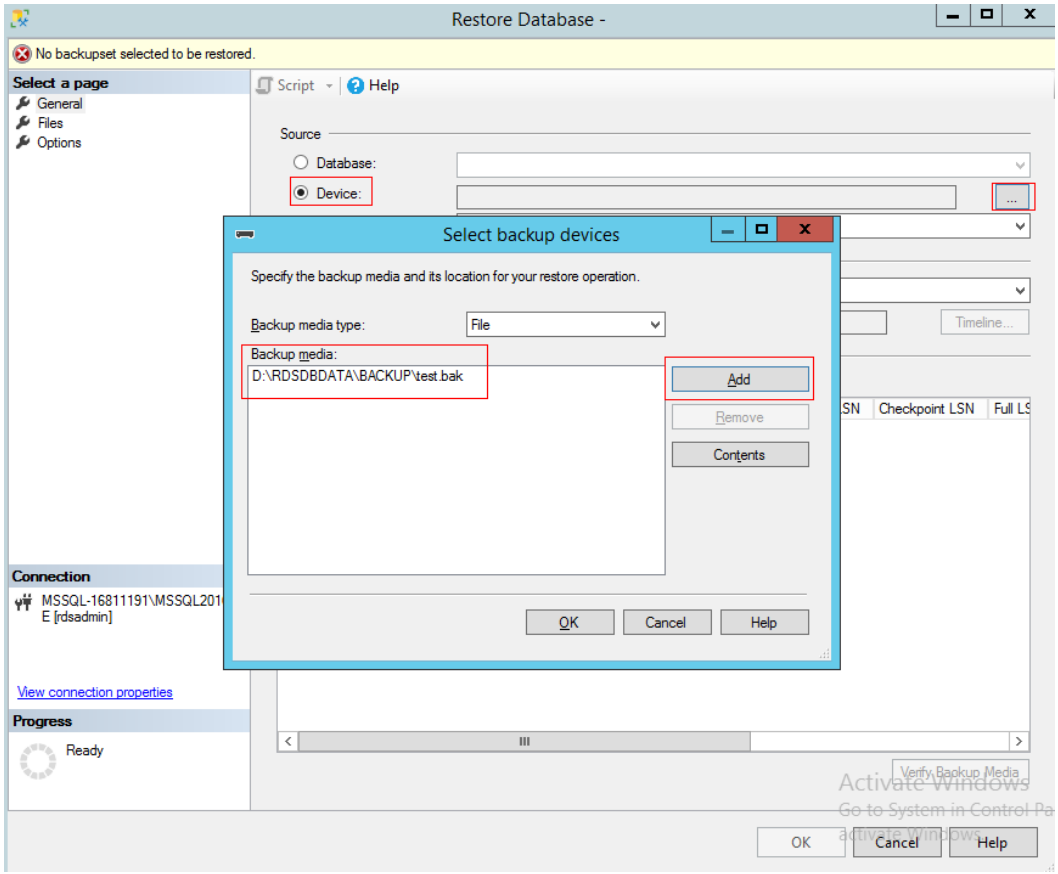
Step 2 Right-click **Databases**, and choose **Restore Database** from the shortcut menu.

Figure 4-46 Selecting a database



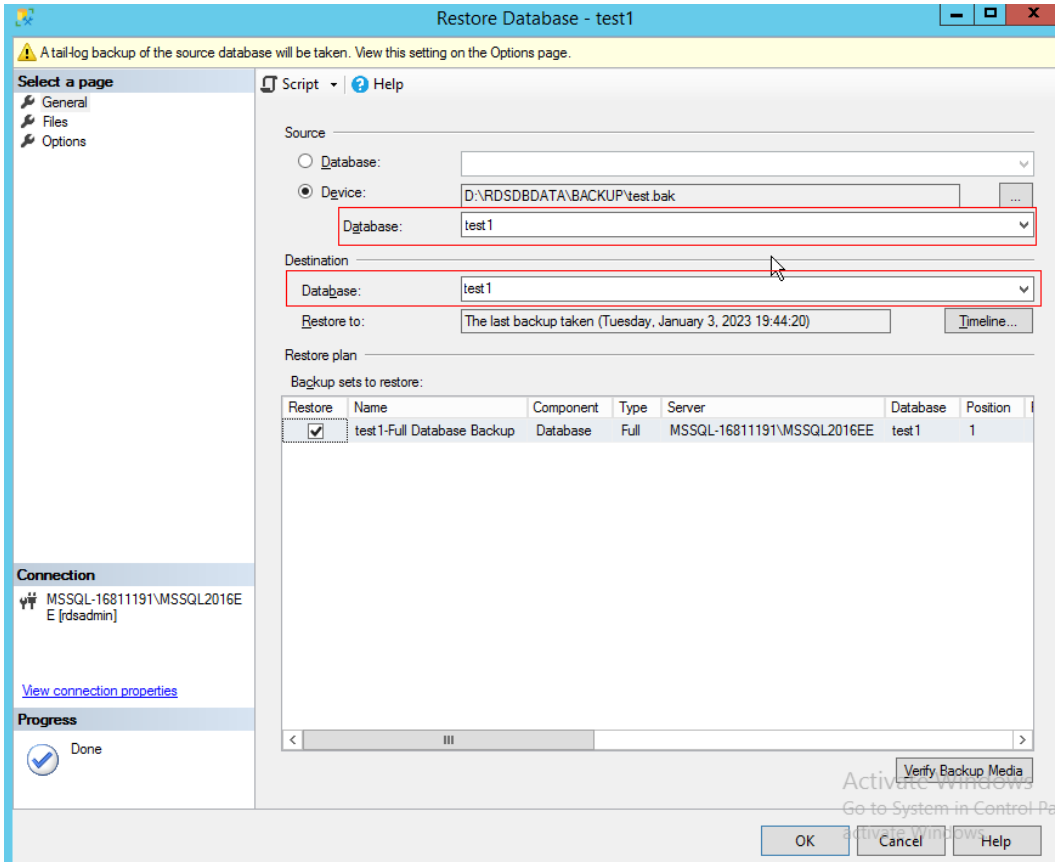
Step 3 Select **Device**, add the .bak backup file, and click **OK**.

Figure 4-47 Adding a backup file



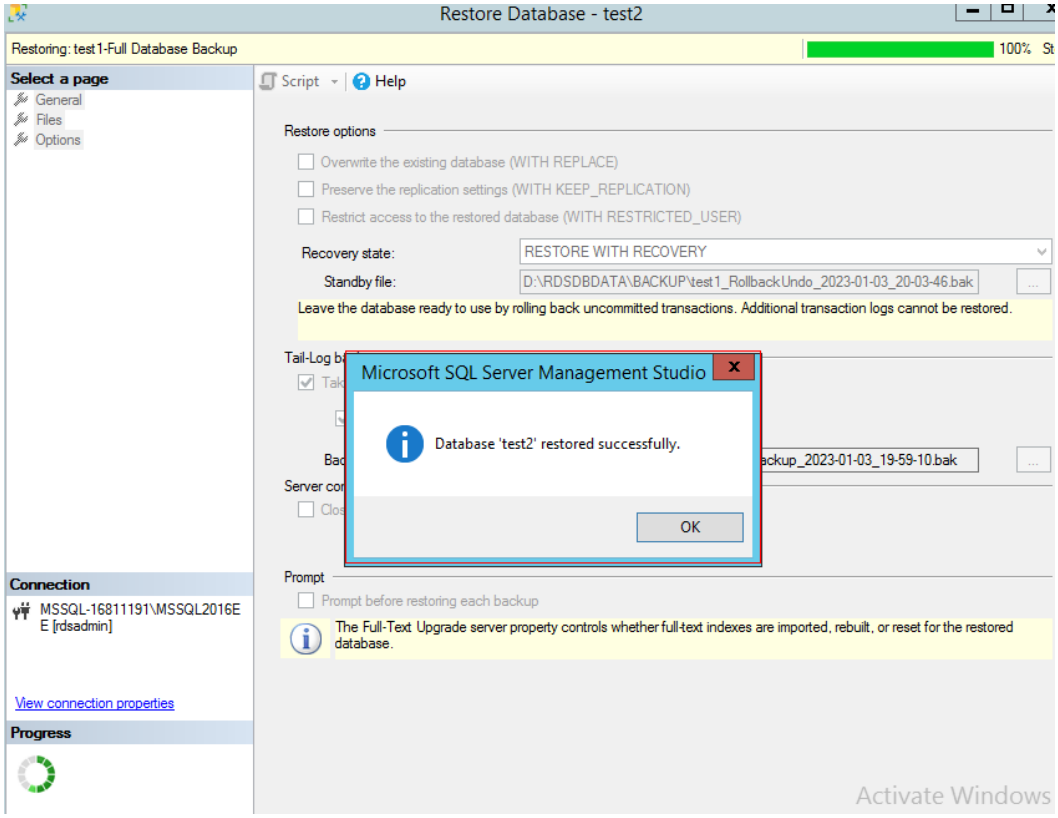
Step 4 Select the database to be restored. You can select the source database from the **Database** drop-down list box in the **Source** area and change the name of the destination database in the **Destination** area.

Figure 4-48 Selecting the source and destination databases



Step 5 Click **OK**.

Figure 4-49 Successful restoration



----End

Restoring Incremental Backups (Log Backups)

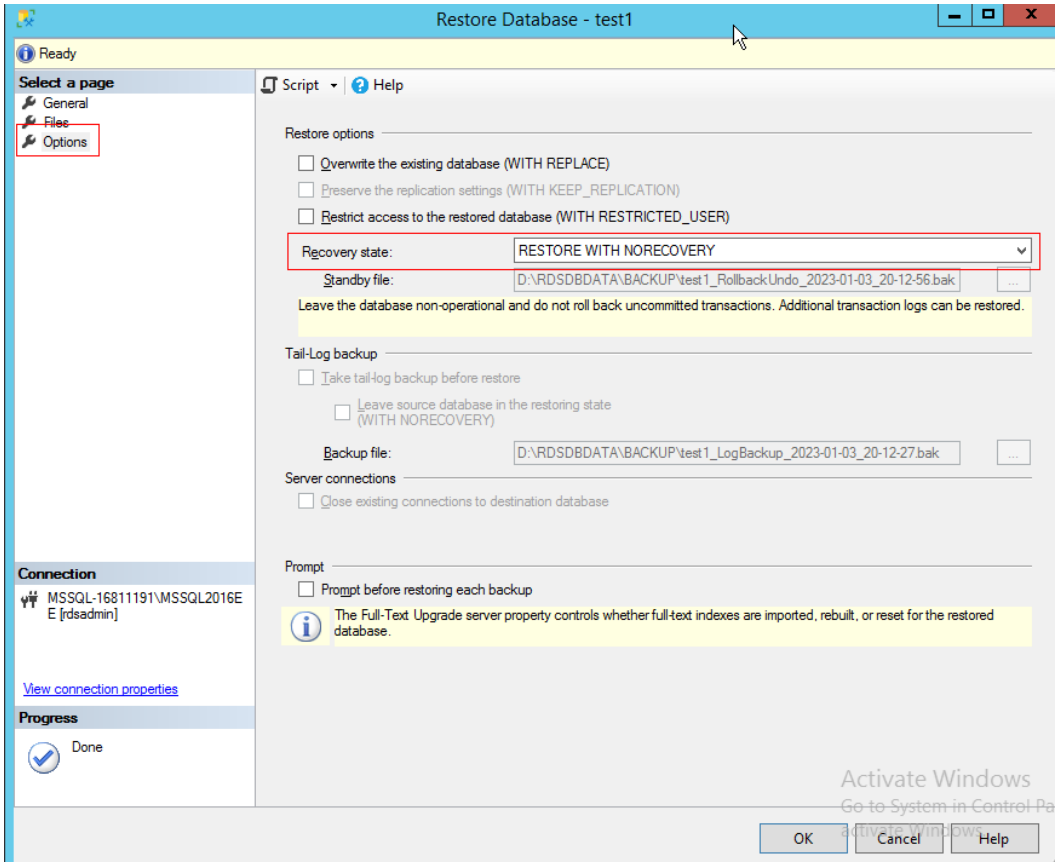
NOTE

Before restoring log backups, ensure that the data backup has been restored and the database is in the **Restoring** state. Log backups must be consecutive. You must restore a database according to its backup sequence. If any backup is missing, the restoration cannot be completed.

Step 1 Restore the data backup by referring to **Step 1** to **Step 4**.

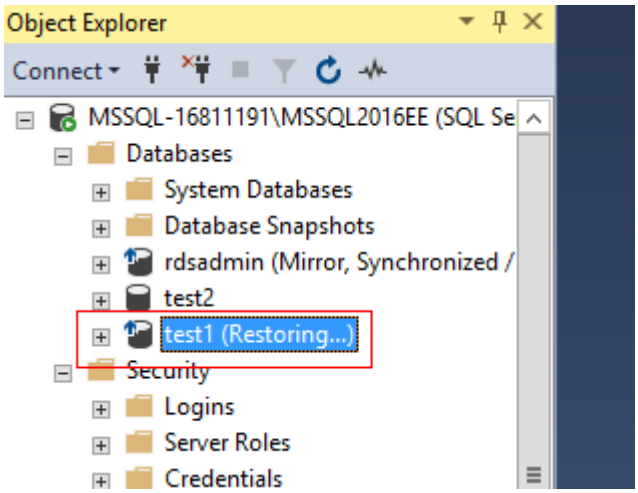
Step 2 Click **Option** and set **Recovery state** to **RESTORE WITH NORECOVERY**.

Figure 4-50 Setting Recovery state



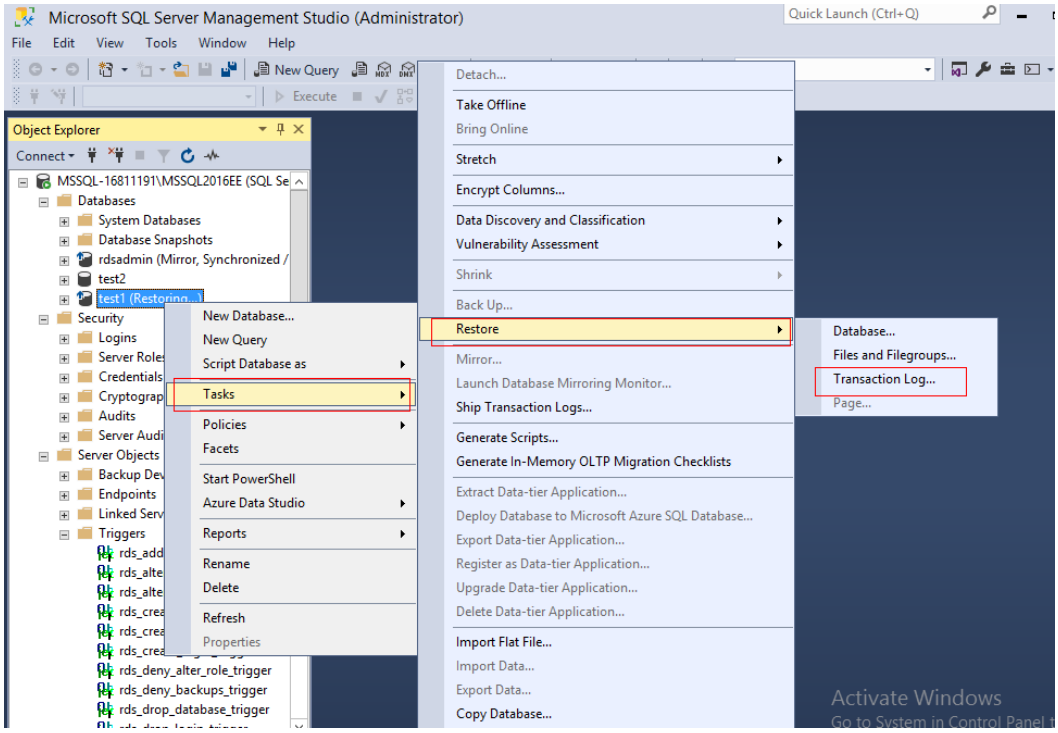
Step 3 Check that the database status is **Restoring**.

Figure 4-51 Checking the restoration status



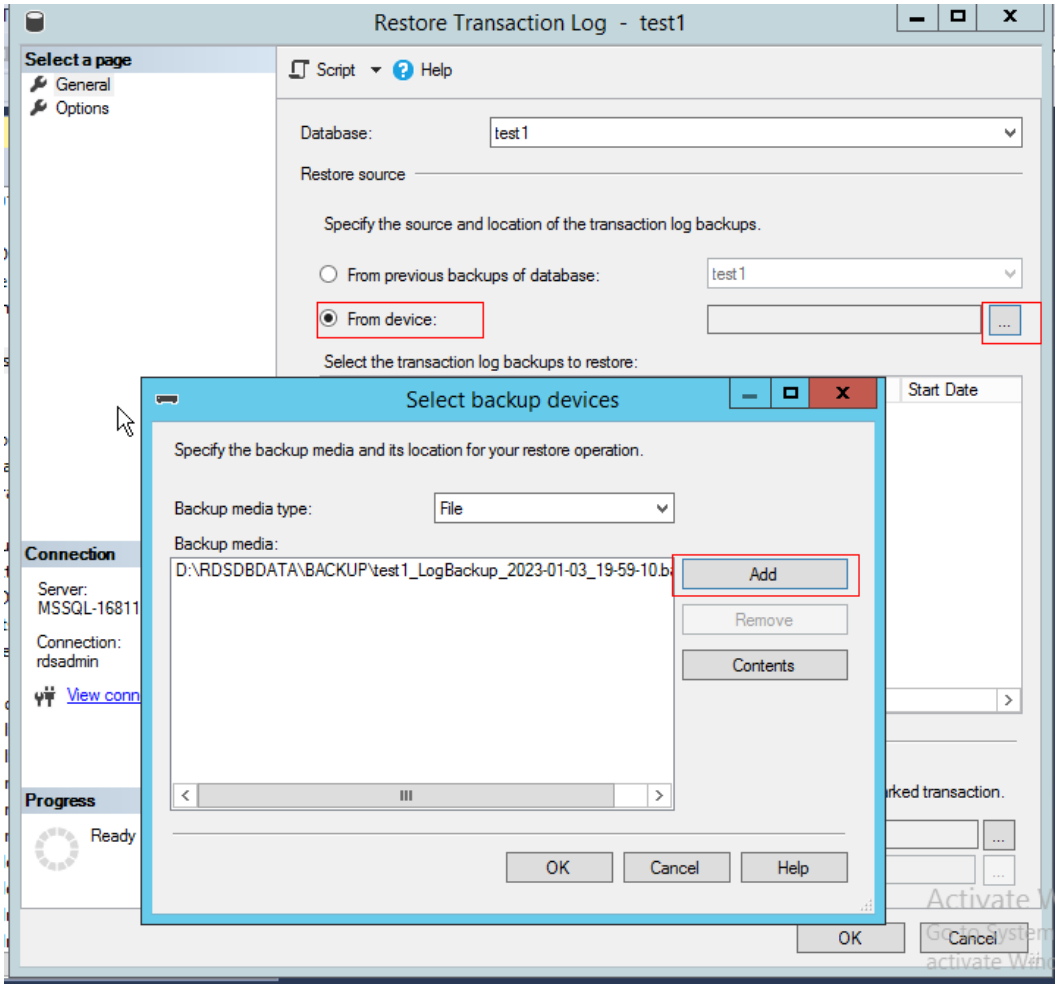
Step 4 Right-click the database and choose **Tasks** > **Restore** > **Transaction Log** from the shortcut menu.

Figure 4-52 Selecting a database



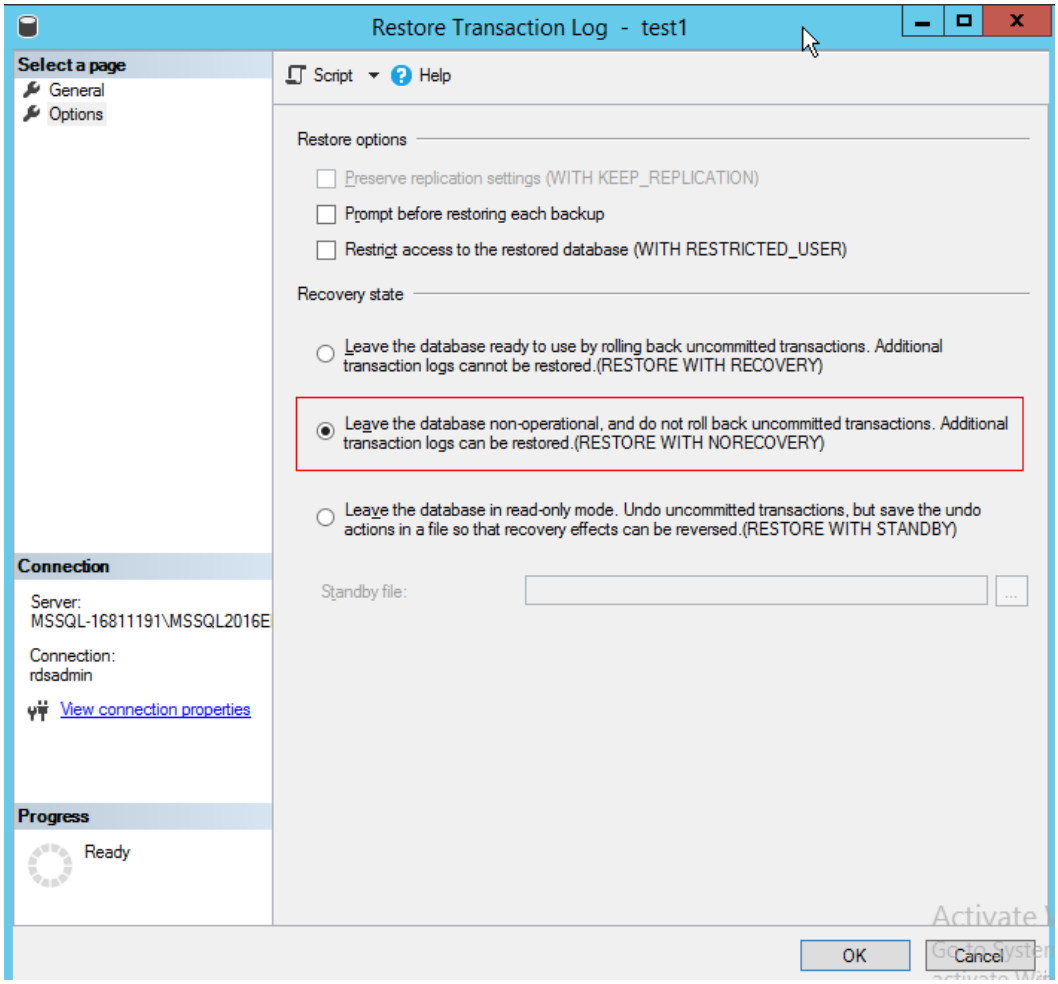
Step 5 Select **From device** and add the backup file to be restored.

Figure 4-53 Adding a backup file



Step 6 If the backup file is not the last incremental backup file and you need to restore other incremental backup files, change the value of **Recovery state** to **RESTORE WITH NORECOVERY**. Otherwise, select **RESTORE WITH RECOVERY** for **Recovery state** and click **OK**.

Figure 4-54 Restoring log backups



Step 7 If there are any other incremental backups that need to be restored, repeat **Step 4** to **Step 6** until the last log backup is restored.

----End

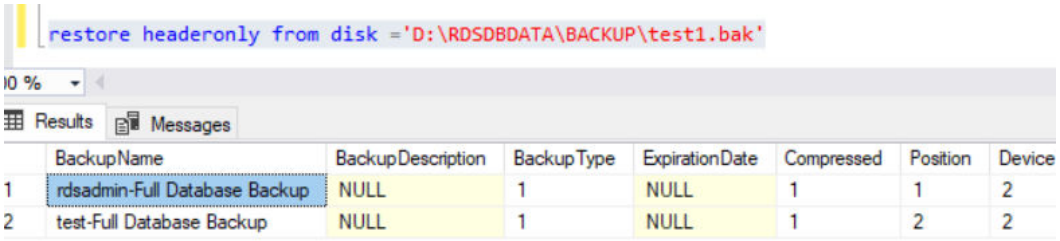
FAQ

Q: Can data be restored if there is only the **rdsadmin** database but no target database in the downloaded .bak file?

A: Yes. The solution is as follows:

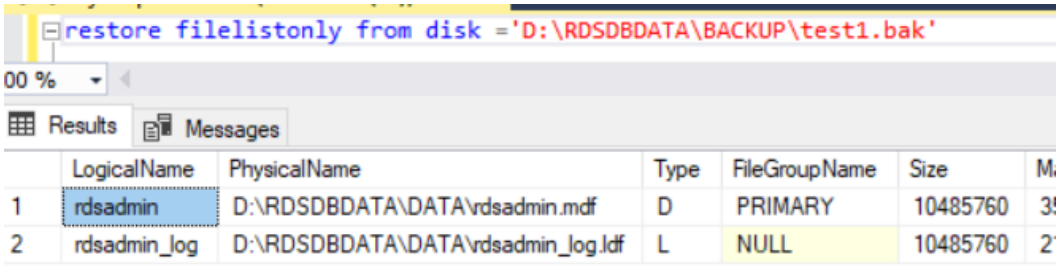
1. The downloaded backup file contains two databases. The first database is **rdsadmin**, and the second database is the target database, for example, **test**.
2. Query backup file header information.
   ```
   restore headeronly from disk='Local path of the .bak file'
   ```


Figure 4-55 Querying backup file information



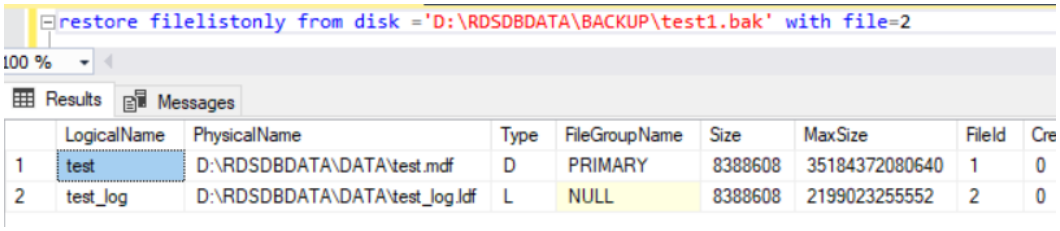
3. Query information about the databases that were backed up.
   restore filelistonly from disk='*Local path of the .bak file*'
   By default, only information about the first database is read.

Figure 4-56 Querying information about the databases that were backed up



4. To read the second or third database, add **with file**. The value of **with file** is that of **position** in the command output of **restore headeronly**.
   restore filelistonly from disk='*Local path of the .bak file*' with file=2

Figure 4-57 Querying information about other databases



5. Restore data.

Figure 4-58 Restoring data

```
USE [master]
RESTORE DATABASE [test_new]
FROM  DISK = N'D:\RDSDBDATA\BACKUP\test1.bak'
WITH FILE = 2,
MOVE N'test' TO N'D:\RDSDBDATA\DATA\test.mdf',
MOVE N'test_log' TO N'D:\RDSDBDATA\DATA\test_log.ldf',
NOUNLOAD,  STATS = 5

GO
```

```
USE [master]
RESTORE DATABASE [@dbname]
```



```
FROM DISK='@path'
WITH FILE= @file
MOVE '@logicalname1' TO '@filepath1'
MOVE '@logicalname2' TO '@filepath2'
NOUNLOAD, STATS=5
GO
```

- *@dbname*: Database name.
- *@path*: Full backup file path.
- *@file*: Location of the database in the .bak file, that is, the value of **position** in the command output of **restore headeronly**.
- @logicalname1: Logical name in the backup file and the file path of the new database. Its value is that of **LogicalName** in the command output of **restore filelistonly**.
- *@filepath1*: Local path for storing physical files.
- *@logicalname2*: The same as *@logicalname1*.
- *@filepath2*: The same as *@filepath1*.

Run the SQL statements above based on the header information obtained in 2 to restore the data.

4.9.4 Restoring a DB Instance to a Point in Time

Scenarios

You can restore from automated backups to a specified point in time. The backup data can be restored to new or existing DB instances.

If you delete a database or modify some records in a database at a specified time, you only need to restore the database instead of restoring the whole DB instance. You can also restore databases to a point in time as required.

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.

Constraints

- Constraints on restoring data to an existing DB instance (other than the original instance):
  - 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.
  - The storage space of the selected DB instance must be no less than that of the original DB instance. Otherwise, data will not be restored.
  - The time zone of the selected DB instance must be the same as that of the original DB instance. Otherwise, data inconsistency may occur.
  - DB instances with the TDE function enabled cannot be restored from backups to existing DB instances.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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 In the displayed dialog box, specify required information and click **OK**.

1. Select the time range, select or enter a time point within the acceptable range.

   NOTE

   If your instance has been restored before by overwriting its data, the period from the time when the restoration started to the time when the first backup was created after the restoration will not be shown in the restorable time range.

2. Select a restoration method.
   - Restore to Existing

     If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

     Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see *Identity and Access Management User Guide*.

     Select an existing DB instance and click **Next**.
3. 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 (-), underscores (_), and periods (.) are allowed.

Step 7 View the restoration result. The result depends on which restoration method was selected:

- Restore to Existing

  On the **Instances** 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

Follow-up Operations

After the restoration is successful, you can **log in to the DB instance** for verification.

FAQs

How Can I Restore Data If No Backup Is Available?

4.10 Parameters

4.10.1 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.

If you create a DB instance without specifying a custom DB parameter template, a default parameter template is used. 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 **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 **Replicating a Parameter Template**.

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

- When you change a parameter value in a parameter template that has been applied to a DB instance and save the change, the change takes effect only to the DB instance and does not affect other DB instances.
- The changes to parameter values in a custom parameter template take effect only after you apply the template to DB instances. For details, see **Applying a Parameter Template**.
- When you change dynamic parameter values in parameter templates in batches and save the changes, the changes will take effect only after you apply the parameter templates to DB instances. When you change static parameter values in parameter templates in batches and save the changes,

the changes will take effect for DB instances only after you apply the parameter templates to DB instances and manually reboot the DB instances.

- 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.

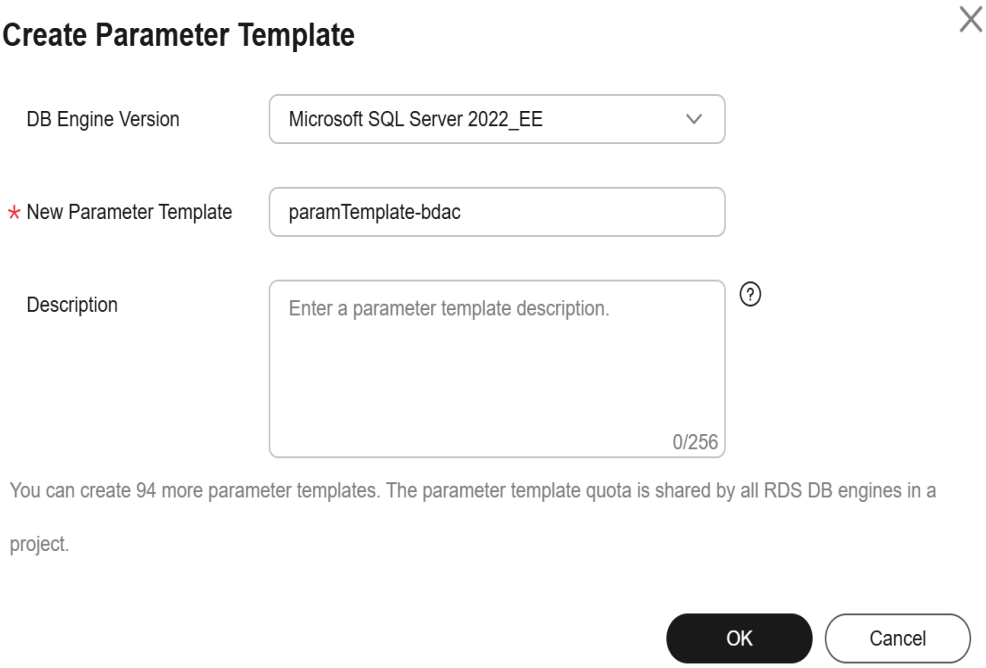
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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: >!<"&'=

Figure 4-59 Creating a parameter template



----End

4.10.2 Modifying RDS for SQL Server Instance 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.

The following are the key points you should know when using parameters:

- Modifying instance parameters: When you modify dynamic parameters on the **Parameters** page of a DB instance and save the modifications, the modifications take effect immediately regardless of the **Effective upon Reboot** setting. However, when you modify static parameters on the **Parameters** page of a DB instance and save the modifications, the modifications do not take effect until you manually reboot the DB instance.
- Modifying parameter template parameters: When you modify parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you have applied the template to your DB instances. For operation details, see **Applying a Parameter Template**. When you modify static parameters in a custom parameter template on the **Parameter Templates** page and save the modifications, the modifications do not take effect until you have applied the template to your DB instances and manually rebooted those DB instances.

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 a Custom Parameter Template and Applying It to DB Instances

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Parameter Templates** 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.

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 **Change History** to view the modification details.

Step 7 Apply the parameter template to your DB instance. 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** 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.

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 **Instances** 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

Common Parameters

Table 4-8 Common parameters

Parameter	Description	Reference
remote query timeout	The remote query timeout option, which specifies how long, in seconds, a remote operation can take before RDS for SQL Server times out.	**Will I Be Logged Out If the Connection to RDS for SQL Server Instances Times Out?**

4.10.3 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 analyzing details.

Exporting Instance Parameters

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

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 in the **Custom Templates** tab on the **Parameter Templates** page.

- Exporting to a file

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

NOTE

The file name can contain 4 to 81 characters.

----End

4.10.4 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.

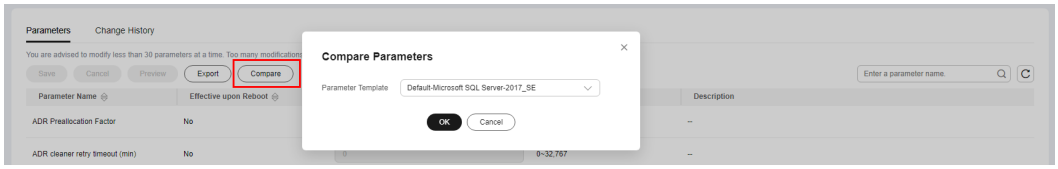
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

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

Figure 4-60 Comparing instance parameters with those in a specified parameter template



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.

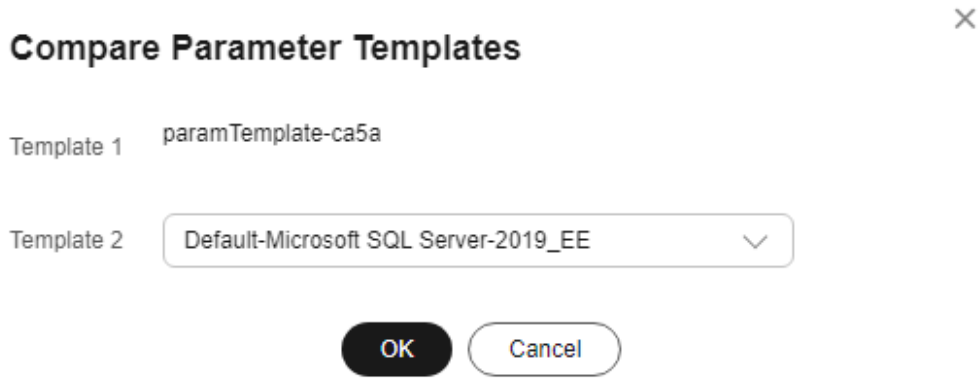
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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**.

Figure 4-61 Selecting a parameter template to be compared



- 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

4.10.5 Viewing Parameter Change History

Scenarios

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

NOTE

The change history for an exported or custom parameter template is initially blank.

Viewing Change History of a DB Instance

Step 1 Log in to the management console.

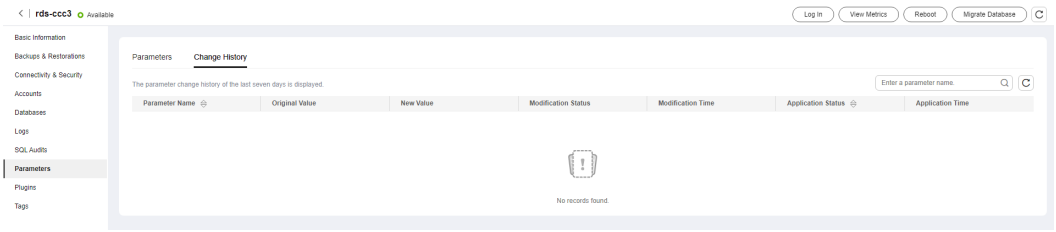
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

Figure 4-62 Viewing parameter change history



You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

----End

Viewing Change History of a Parameter Template

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

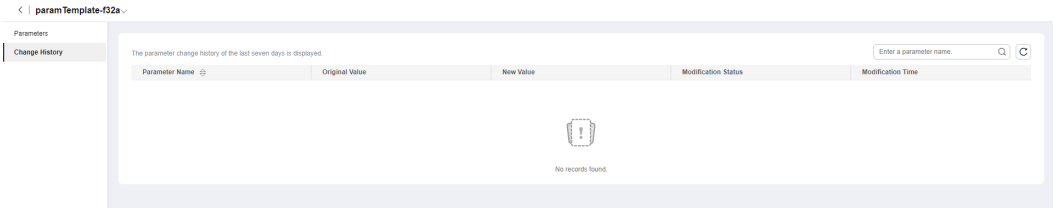
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Parameter Templates** 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.

Figure 4-63 Viewing parameter change history



You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

You can apply the parameter template to DB instances as required by referring to **Applying a Parameter Template**.

----End

4.10.6 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. You can also export the parameter template to generate a new parameter template for future use.

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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

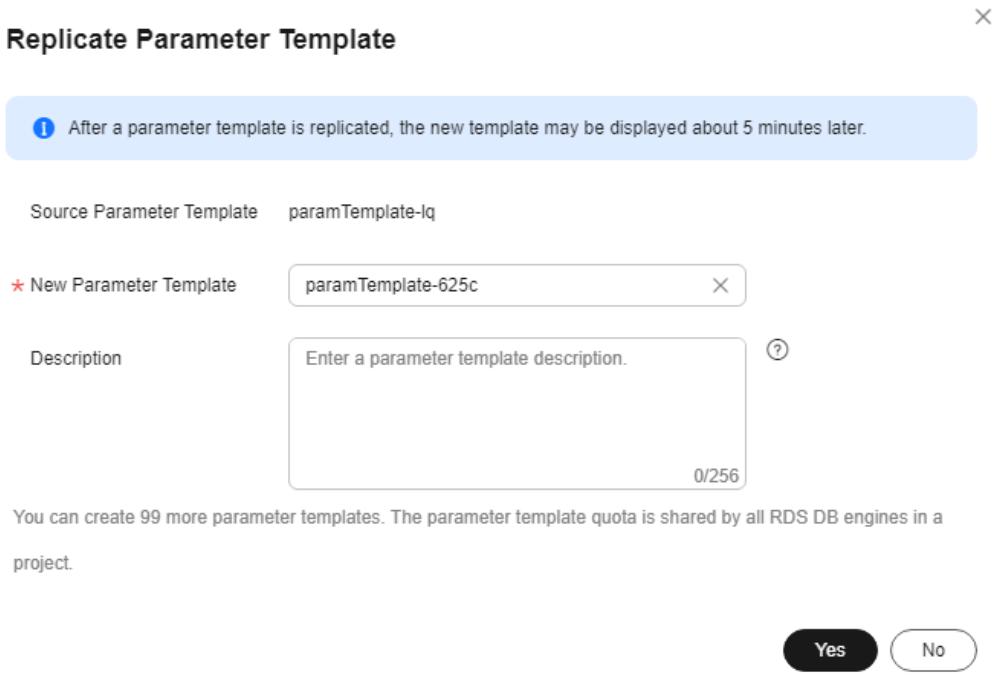
Alternatively, click the target DB instance on the **Instances** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

NOTE

To ensure that your parameter templates are applicable to all types of DB instances and databases can be started normally, the values of **innodb_flush_log_at_trx_commit** and **sync_binlog** exported from primary DB instances or read replicas are **1** by default.

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

Figure 4-64 Replicating a 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: >!<"&'=

After the parameter template is replicated, a new template is generated in the list on the **Parameter Templates** page.

----End

4.10.7 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.

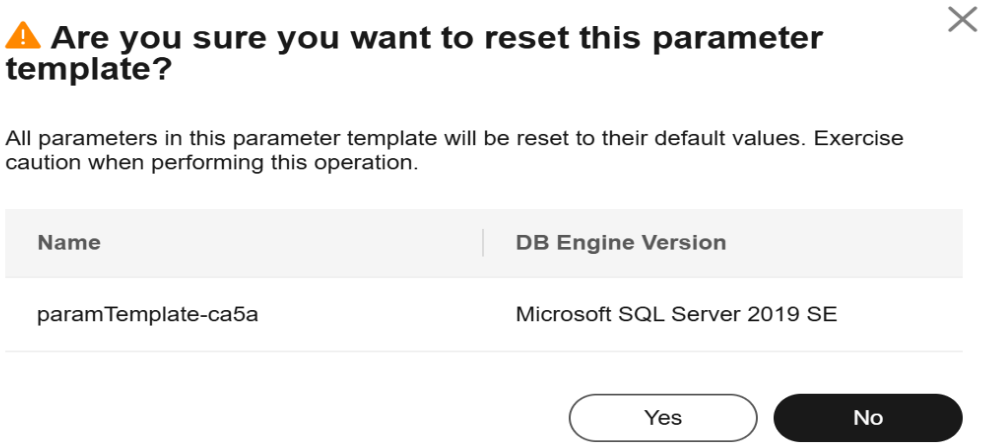
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** page, click **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 5 Click **Yes**.

Figure 4-65 Confirming the reset



Step 6 Apply the parameter template to your DB instance. 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

4.10.8 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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 **Viewing Application Records of a Parameter Template**.

----End

4.10.9 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.

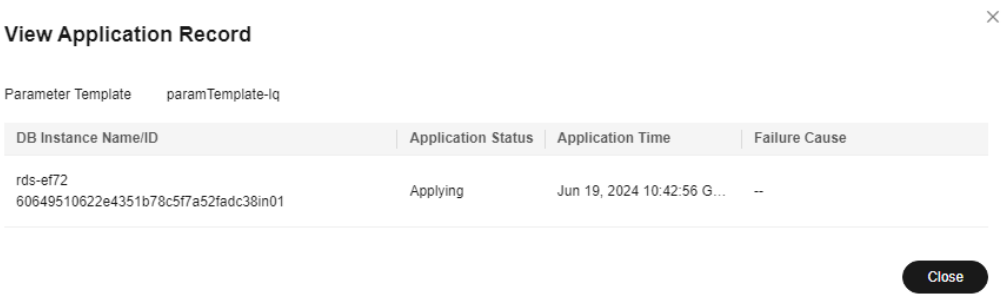
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Default Templates** or **Custom Templates** page, locate the target parameter template and 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).

Figure 4-66 Viewing application records of a parameter template



----End

4.10.10 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 Enter a new description and click **OK** to submit the modification or click **Cancel** to cancel the modification.

- The description consists of a maximum of 256 characters and cannot contain carriage return characters or 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

4.10.11 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Parameter Templates** 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

4.11 Connection Management

4.11.1 Viewing 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.

Only floating IPv4 addresses can be changed.

Procedure

You can use a self-configured floating IP address when creating a DB instance.

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.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

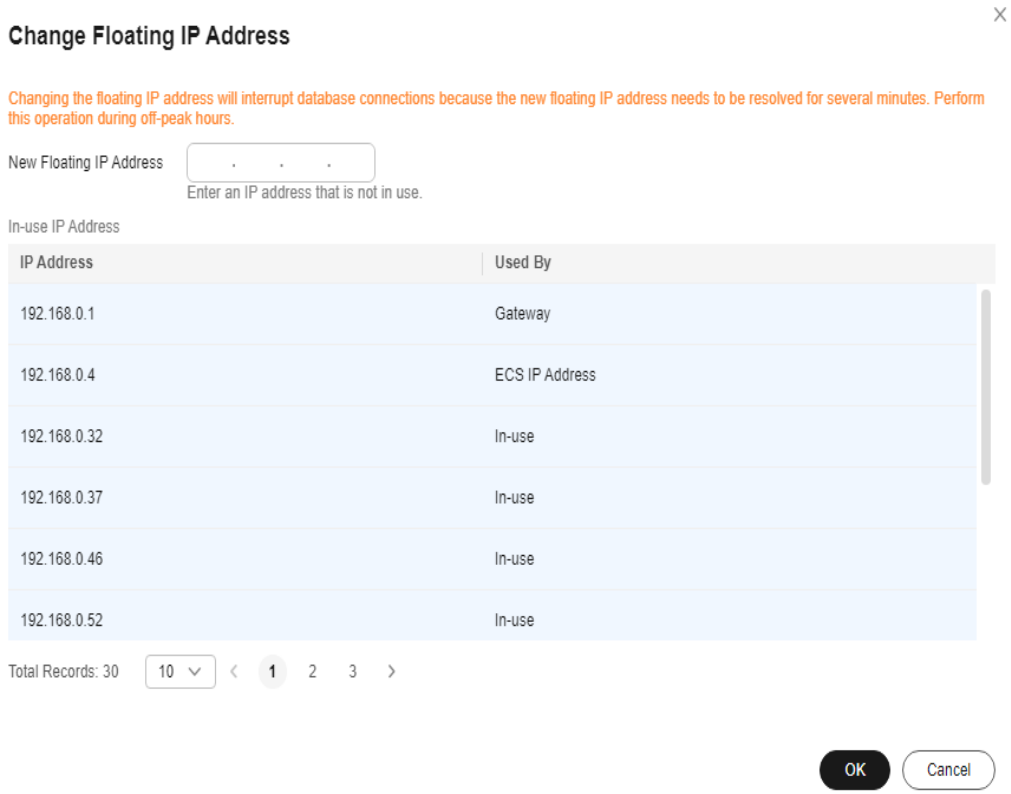
Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **Floating IP Address**, click **Configure**.

Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Change** next to the **Floating IP Address** field.

Step 6 In the displayed dialog box, check the number of in-use IP addresses. If the in-use IP addresses are less than 254, there are unused floating IP addresses.

Figure 4-67 Changing a floating IP address



Step 7 Enter an available IP address and click **OK**.

An in-use IP address cannot be used as the new floating IP address of the DB instance.

Step 8 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

4.11.2 Applying for and Changing a Private Domain Name

You can apply for a private domain name and connect to your RDS DB instance through the private domain name.

Constraints

- To apply for or change a private domain name, you need to **submit a service ticket** to apply for required permissions.
- After a private domain name is generated, changing the floating IP address will interrupt database connections. Exercise caution when performing this operation.

Applying for a Private Domain Name

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

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

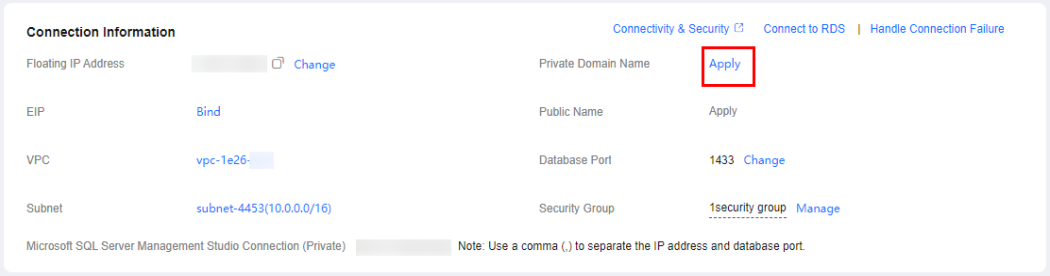
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name to go to the **Overview** page.

Step 5 Under **Private Domain Name**, click **Apply**.

Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Apply** next to the **Private Domain Name** field.

Figure 4-68 Applying for a private domain name



Step 6 In the **Private Domain Name** field, view the generated private domain name.

----End

Changing a Private Domain Name

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **Private Domain Name**, click **Configure**.

Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Change** next to the **Private Domain Name** field.

Step 6 In the displayed dialog box, enter a new private domain name. Click **OK**.

NOTE

- Only the prefix of a private domain name can be modified.
- The prefix of a private domain name can contain 8 to 63 characters, and can include only letters and digits.
- The new private domain name must be different from the existing ones.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

4.11.3 Applying for and Changing a Public Domain Name

You can apply for a public domain name and connect to your RDS DB instance through the public domain name.

Constraints

- To apply for or change a public domain name, you need to **submit a service ticket** to apply for required permissions.
- Before applying for a public domain name, you need to bind an EIP to your instance.
- After a public domain name is generated, changing the EIP will interrupt database connections. Exercise caution when performing this operation.

Applying for a Public Domain Name

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

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

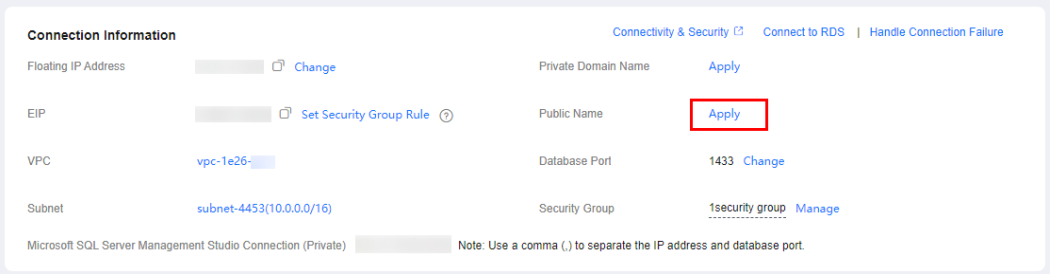
Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name to go to the **Overview** page.

Step 5 Under **Public Name**, click **Apply**.

Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Apply** next to the **Public Name** field.

Figure 4-69 Applying for a public domain name



Step 6 In the **Public Name** field, view the generated public domain name.

----End

Changing a Public Domain Name

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 5 Under **Public Name**, click **Configure**.

Alternatively, in the navigation pane on the left, choose **Connectivity & Security**. On the displayed page, click **Change** next to the **Public Name** field.

Step 6 In the displayed dialog box, enter a new public domain name. Click **OK**.

NOTE

- Only the prefix of a public domain name can be modified.
- The prefix of a public domain name can contain 8 to 63 characters, and can include only letters and digits.
- The new public domain name must be different from existing ones.

Step 7 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

4.11.4 Binding and Unbinding an EIP

Scenarios

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.

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, add an individual IP address or an IP address range that will access the DB instance to the inbound rule. For details, see **Configuring Security Group Rules**.
- Traffic generated by the public network is charged. You can unbind the EIP from your DB instance when the EIP is no longer used.

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**.

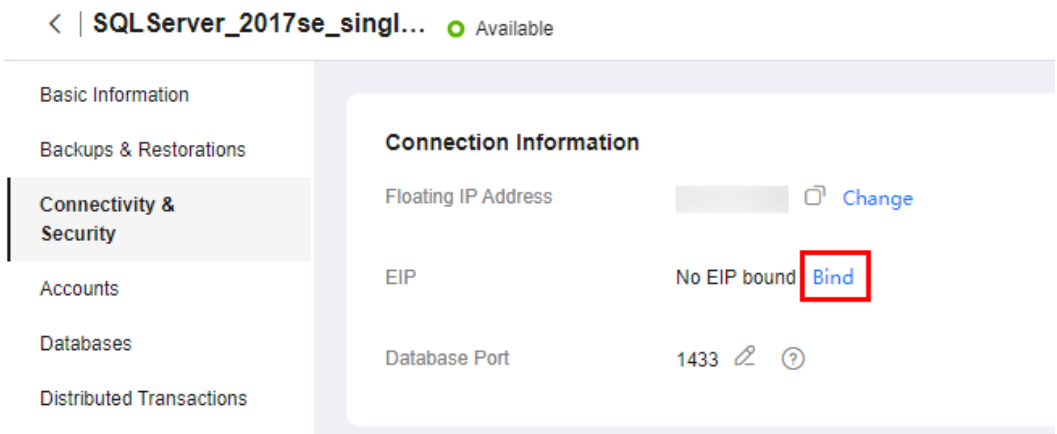
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Figure 4-70 Binding an EIP



Step 6 In the displayed dialog box, all unbound EIPs are listed. Select the EIP to be bound and click **Yes**. If no available EIPs are displayed, click **View EIP** and obtain an EIP.

Step 7 On the **Connectivity & Security** 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.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance that has an EIP bound.

Step 5 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Unbind** next to the **EIP** field. In the displayed dialog box, click **Yes**.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

Step 7 On the **Connectivity & Security** page, view the results.

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

4.11.5 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.

If specific security group rules have been configured for a DB instance, you need to change the inbound rules of the security group to which the DB instance belongs after changing the database port.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

Step 5 On the **Overview** page, find **Database Port** and click **Configure** under it.

Alternatively, choose **Connectivity & Security** in the navigation pane on the left. On the displayed page, click **Change** next to the **Database Port** field.

NOTE

- For RDS for SQL Server 2022 Enterprise Edition, 2022 Standard Edition, 2022 Web Edition, 2019 Enterprise Edition, 2019 Standard Edition, 2019 Web Edition, 2017 Enterprise Edition, 2017 Standard Edition, and 2017 Web Edition, the port number can be set to 1433 (default) or 2100 to 9500 (excluding 5050, 5353, 5355, 5985, and 5986).
- For other editions, the port number can be set to 1433 (default) or 2100 to 9500 (excluding 5355 and 5985).

- To submit the change, click .
  - In the dialog box, click **OK**.

    If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

    Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

    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 Microsoft SQL Server, only 2017 Enterprise Edition supports read replicas.

    iii. This process takes 1-5 minutes.
  - In the dialog box, click **Cancel** to cancel the modification.
- To cancel the change, click .

Step 6 View the result on the **Overview** page.

----End

4.12 Accounts (Non-Administrator)

4.12.1 Creating a Database Account

Scenarios

When you create an RDS for SQL Server instance, account **rdsuser** is created at the same time by default. You can create other database accounts as needed.

You can create a database account using RDS or DAS:

- **RDS**: RDS is easy to use. There are no commands to remember.
- **DAS**: DAS is a powerful platform that offers more flexibility, but you need to be familiar with various commands. The process requires a bit more expertise.

Constraints

Database accounts cannot be created for DB instances that are 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.

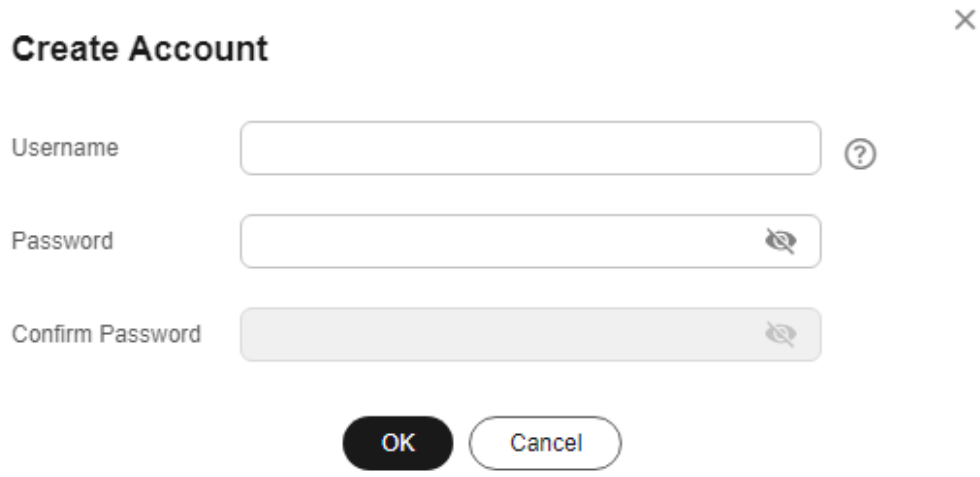
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Accounts** page, click **Create Account**.

Step 6 In the displayed dialog box, specify **Username**, **Password**, and **Confirm Password**, and click **OK**.

Figure 4-71 Creating an account



- The username can contain 1 to 128 characters. It can include letters, digits, hyphens (-), and underscores (_), and it must be different from system accounts. System accounts include **rdsadmin**, **rdsuser**, **rdsbackup**, and **rdsmirror**.
- 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 must differ from the account name or the account name in reverse order.
- Enter a strong password to improve security, preventing security risks such as brute force cracking.
- If you require fine-grained permissions control, log in to the database through the DAS console.

Step 7 After the account is created, you can manage it on the **Accounts** page.

----End

Creating a Database Account Through DAS

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click **Log In** in the **Operation** column.

Step 5 On the displayed page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Run the following command to create an account.

create user *username*;

----End

4.12.2 Resetting a Password for a Database Account

Scenarios

You can reset passwords for the accounts you have created. To protect your instance against brute force cracking, change your password periodically, such as every three or six months.

Constraints

Passwords cannot be reset for DB instances that are being restored.

Procedure

Step 1 Log in to the management console.

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

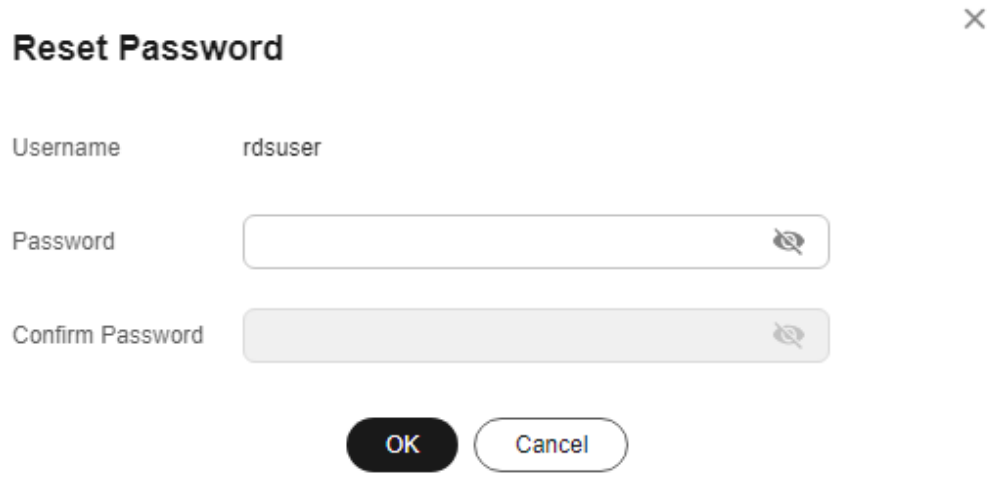
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

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 dialog box, enter a new password, confirm the password, and click **OK**.

Figure 4-72 Resetting a password



- 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 must differ from the account name or the account name in reverse order.
- Enter a strong password to improve security, preventing security risks such as brute force cracking.
- After the password is reset, the DB instance will not be rebooted and your permissions will not be changed.

----End

4.12.3 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

- This operation is not allowed for DB instances that are being restored.
- Account deletions on the primary instance are not synchronized to the read replicas.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 In the navigation pane on the left, choose **Accounts**. On the displayed page, locate the target username and click **Delete** in the **Operation** column.

Step 6 In the displayed dialog box, click **OK**.

----End

4.13 Databases

4.13.1 Creating a Database

Scenarios

After a DB instance is created, you can create databases on it.

Constraints

- A maximum of 1,000 databases can be created for each DB instance.
- Databases cannot be created when the DB instance is being restored or its instance class is being changed.
- Database names must be unique.
- After a database is created, the database name cannot be changed.

Creating a Database Through RDS

Step 1 Log in to the management console.

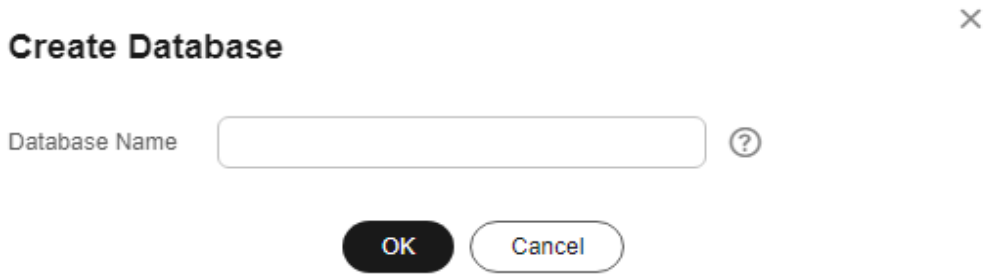
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Databases** page, click **Create Database**. In the displayed dialog box, enter a database name and click **OK**.

Figure 4-73 Creating a database



- The database name can contain 1 to 64 characters, and can include letters, digits, hyphens (-), underscores (_), and periods (.). It cannot start or end with an RDS for SQL Server system database name. RDS for SQL Server system databases include **master**, **msdb**, **model**, **tempdb**, **resource**, and **rdsadmin**.
- The character set of the DB instance is used by default.

Step 6 After the database is created, manage it on the **Databases** page.

----End

Creating a Database Through DAS

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Step 5 On the displayed page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 7 Run the following command to create a database.

create database *database_name*;

----End

4.13.2 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 being restored.

Procedure

Step 1 Log in to the management console.

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

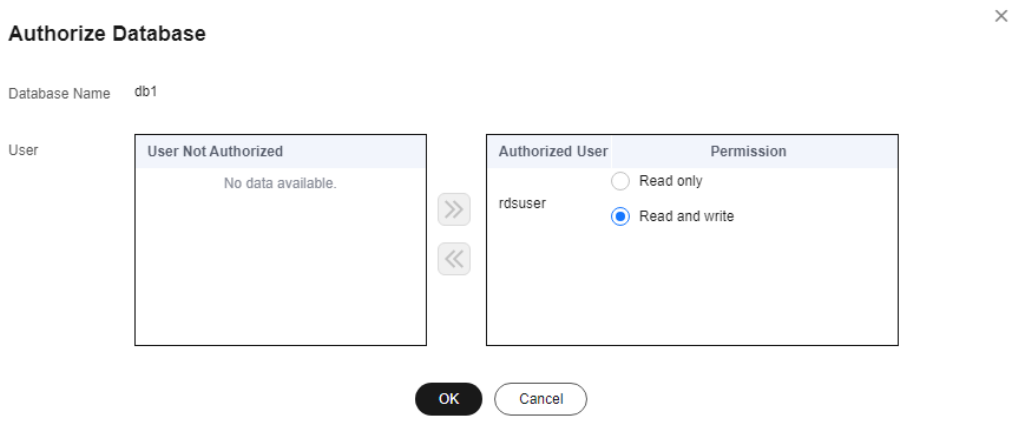
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

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.

Figure 4-74 Authorization



If no users are available, you can create one by referring to **Creating a Database Account**.

Step 7 Then, click **OK**.

----End

4.13.3 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

Databases cannot be deleted from DB instances that are being restored.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Databases** page, locate the target database and choose **More** > **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see the *Identity and Access Management User Guide*.

----End

4.13.4 Copying a Database

Scenarios

You can copy a database on a DB instance.

Constraints

- Copying a database with a large amount of data takes an extended period of time.
- System databases **master**, **tempdb**, **model**, **msdb**, and **rdsadmin** cannot be copied.

Procedure

Step 1 Log in to the management console.

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

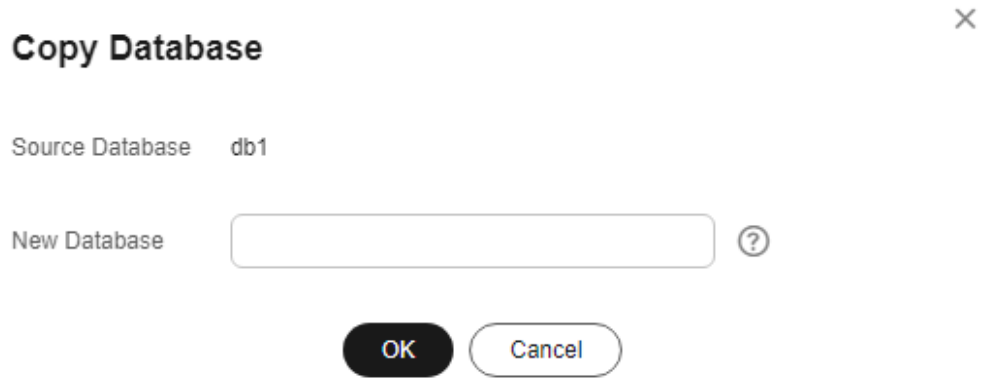
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Databases** page, locate the target database and choose **More** > **Copy** in the **Operation** column.

Step 6 In the displayed dialog box, enter the new database name and click **OK**.

Figure 4-75 Copying a database



----End

4.13.5 Viewing Database Properties

Scenarios

You can view properties of a database, including the database creation time, user connection, whether the database is read-only, and file size.

Procedure

Step 1 Log in to the management console.

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

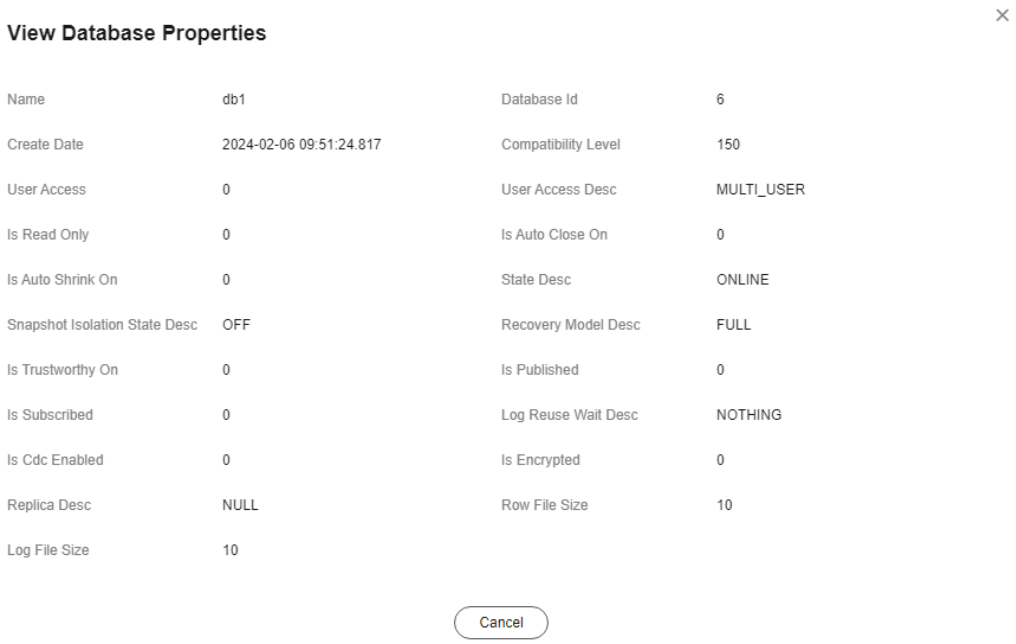
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the instance name.

Step 5 On the **Databases** page, locate the target database and choose **More** > **View Database Properties**.

Step 6 In the displayed dialog box, view the database properties.

Figure 4-76 Viewing database properties



----End

4.14 Security and Encryption

4.14.1 Database Account Security

Password Strength Requirements

NOTICE

SQL Server supports disabling of the database password complexity check. However, to ensure database security, you are advised not to disable it.

- RDS has a password security policy for user-created database accounts. You are advised to enable this policy. Passwords must:
  - Consist of 8 to 128 characters.
  - Contain at least three types of the following: uppercase letters, lowercase letters, digits, and special characters.
  - Not contain the username.

When you are creating a DB instance, the password strength is checked. You can modify the password strength as user **rdsuser**. For security reasons, you are advised to use a password that is at least as strong as the default password.

Account Description

To provide O&M services, the system automatically creates system accounts when you create RDS for SQL Server DB instances. These system accounts are unavailable to you.

NOTICE

Attempting to delete, rename, change passwords for, or change privileges for these accounts will result in an error.

- rdsadmin: has the sysadmin service role and is used to query DB instance information, monitor instance status, rectify faults, migrate data, and restore data.
- rdsmirror: indicates the primary/standby replication account, which is used to create mirroring endpoints.
- rdsbackup: indicates the backup account, which is used for backend backup.
- Mike: indicates the Windows system account of RDS for SQL Server. It is used to initialize SQL statements during the DB instance initialization, including creating the rdsadmin database and related accounts.

4.14.2 Resetting the Administrator Password

Scenarios

You can reset the administrator password only through the primary DB instance.

If you forget the password of the administrator account **rdsuser**, you can reset the password.

If an error occurs on the rdsuser account, for example, the rdsuser account is lost or deleted, you can restore the rdsuser account rights through resetting the password.

Precautions

- If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password.
- 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.
- 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.
- After you reset the administrator password of an RDS for SQL Server instance, all permissions assigned to the administrator will be retained.

Method 1

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

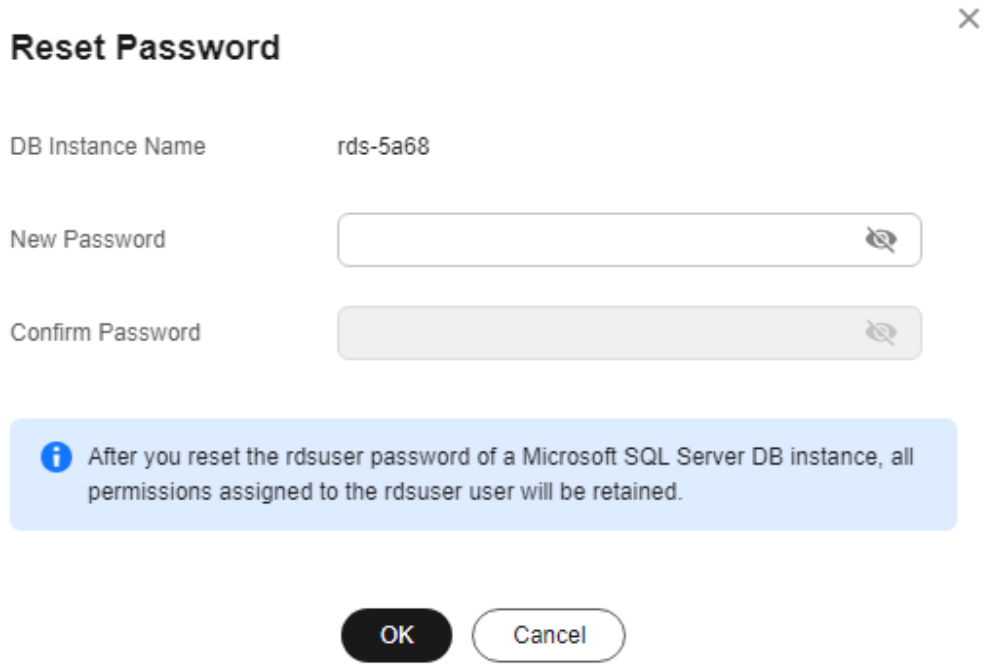
Step 4 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 5 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 Enter and confirm the new password.

Figure 4-77 Resetting the administrator 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 **Yes**.
- To cancel the reset operation, click **No**.

----End

Method 2

Step 1 Log in to the management console.

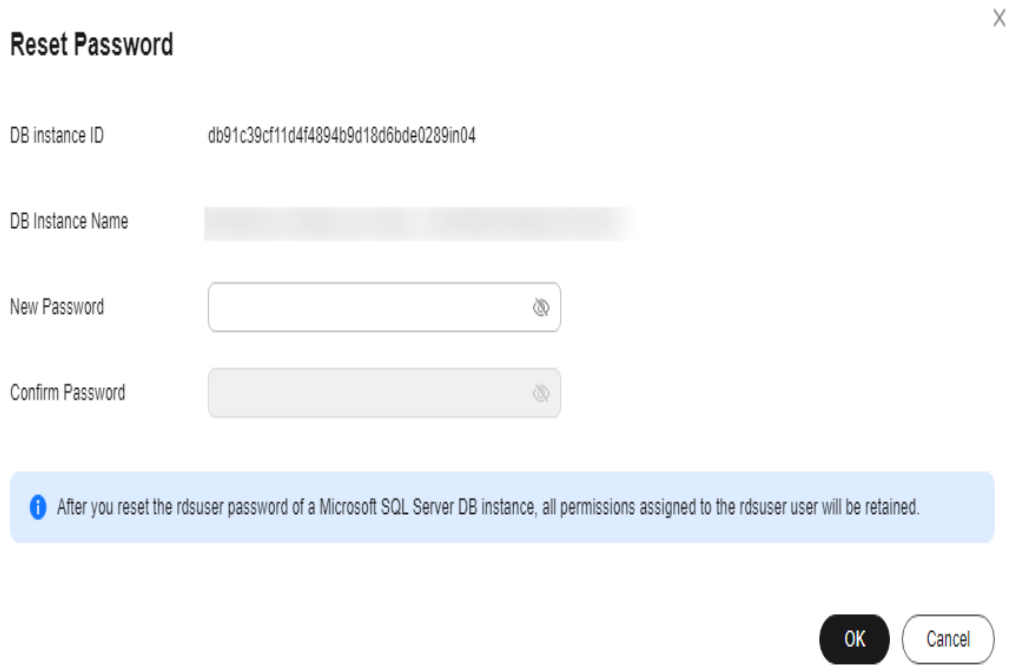
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 On the **Overview** page, find **Administrator** and click **Reset Password** under it. In the displayed dialog box, enter and confirm the new password.

Figure 4-78 Resetting the administrator 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 **Yes**.
- To cancel the reset operation, click **No**.

----End

4.14.3 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 be changed as well.

Precautions

You can add or modify rules for the security group associated with your RDS instance, but you cannot disassociate or delete the security group.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target primary DB instance or read replica.

Step 5 On the **Overview** page, click **Configure** under **Security Group** and select a new security group.

- 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 **Overview** page to view the results.

----End

Managing Security Groups

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance or read replica.

Step 5 On the **Overview** page, click **Manage** under **Security Group**.

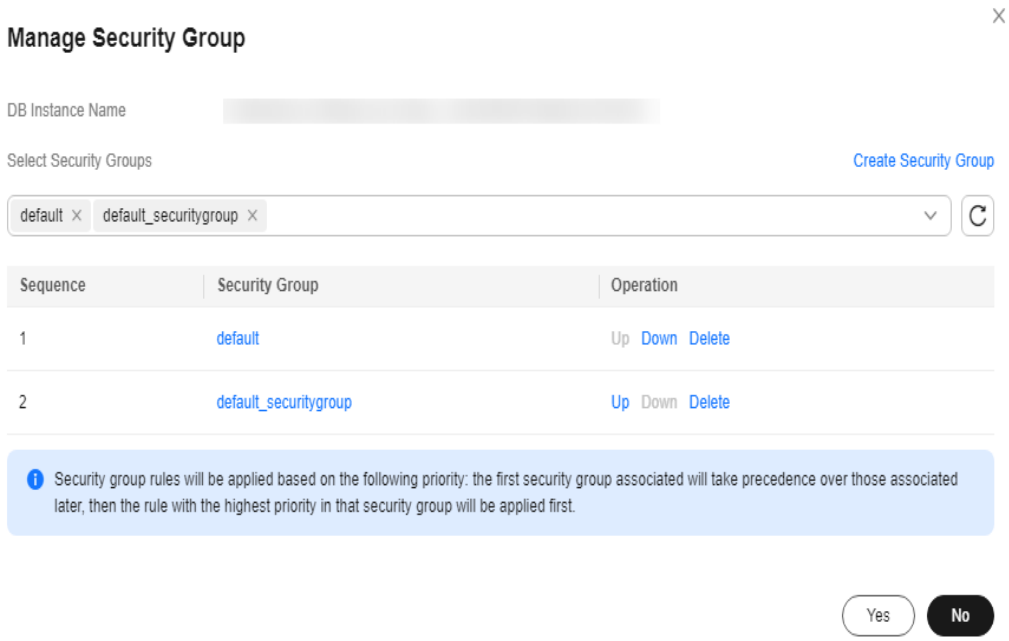
You can select multiple security groups at a time. The security group rules will be applied based on the following sequence: the first security group associated will take precedence over those associated later, then the rule with the highest priority in that security group will be applied first.

To create a new security group, click **Create Security Group**.

NOTE

Using multiple security groups may deteriorate the network performance. You are suggested to select no more than five security groups.

Figure 4-79 Managing security groups



Step 6 Click **Yes** to submit the modification.

----End

4.14.4 Performing a Server-Side Encryption

Introduction

The RDS console provides server-side encryption with Data Encryption Workshop (DEW)-managed keys.

DEW uses a third-party hardware security module (HSM) to protect keys, enabling you to easily create and control encryption keys. For security reasons, keys are not displayed in plaintext outside of HSMs. With DEW, all operations on keys are controlled and logged, and usage records of all keys can be provided to meet regulatory compliance requirements.

If server-side encryption is enabled, disk data will be encrypted and stored on the server when you create a DB instance or expand disk capacity. When downloading encrypted objects, the encrypted data will be decrypted on the server and displayed to you in plaintext.

Encrypting Disks Using Server-Side Encryption

For server-side encryption, you need to first create a key using DEW or use the default key that DEW comes with. When creating a DB instance, select **Enable** for **Disk Encryption** and select or create a key. This key is the end tenant key and will be used for server-side encryption. For details, see **Getting Started with RDS for SQL Server**.

- You will need the KMS administrator permission for the region where RDS is deployed. This permission can be granted using Identity and Access Management (IAM). On the IAM console, add permission policies to user groups. For details, see **Creating a User Group and Assigning Permissions**.
- If you want to use a user-defined key to encrypt objects to be uploaded, create a key using DEW. For details, see **Creating a CMK**.
- If you enable disk encryption during instance creation, the disk encryption status and the key cannot be changed later. Disk encryption will not encrypt backup data stored in OBS.
- If disk encryption is enabled, keep the key properly. Once the key is disabled, deleted, or frozen, the database will be unavailable.
- After an RDS DB instance is created, you cannot disable or delete the key for that instance, or the DB instance will become unusable and the data cannot be restored.
- If you scale up a DB instance with disks encrypted, the expanded storage space will also be encrypted using the original encryption key.

4.14.5 Configuring the TDE Function

Transparent Data Encryption (TDE) encrypts data files and backup files using certificates to implement real-time I/O encryption and decryption. This function effectively protects the security of databases and data files.

TDE is only available for certain RDS for SQL Server editions. For details, see **Table 4-9**.

Table 4-9 RDS for SQL Server editions that support TDE

DB Instance Type	Editions Support for TDE
Primary/Standby	• 2008 R2 Enterprise Edition • 2012 Enterprise Edition • 2014 Enterprise Edition • 2016 Enterprise Edition • 2019 Standard Edition. To enable TDE in this edition, **submit a service ticket** to request required permissions.

DB Instance Type	Editions Support for TDE
Single	• 2016 Enterprise Edition • 2017 Enterprise Edition • 2019 Standard Edition
Cluster	• 2017 Enterprise Edition

Constraints

1. If TDE has been enabled for a single DB instance, the instance cannot be changed to primary/standby DB instances.
2. RDS for SQL Server currently does not support TDE certificate download. To restore data offline using the encrypted .bak file, perform the following operations:
   a. Disable TDE for the database. For details, see **Configuring Database-Level TDE**.
   b. Create a manual backup for the database.
   c. Restore data from the manual backup.
   d. Enable TDE for the database as required.
3. Enabling TDE improves data security but affects read and write performance of encrypted databases. Exercise caution when enabling TDE.
4. To migrate on-premises encrypted databases to RDS for SQL Server DB instances, you need to disable database-level TDE first.
5. DB instances with the instance-level TDE function enabled cannot be restored from backups to existing DB instances.
6. When enabling the instance-level TDE function or using the stored procedure rds_tde to enable or disable database-level TDE, you are advised not to perform the following operations:
   - Delete files from file groups in databases.
   - Delete databases.
   - Take databases offline
   - Split databases.
   - Convert databases or file groups to the READ ONLY state.
   - Run the ALTER DATABASE command.
   - Create backups.
   - Start backup for databases or database files.
   - Start restoration for databases or database files.

Enabling Instance-Level TDE

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

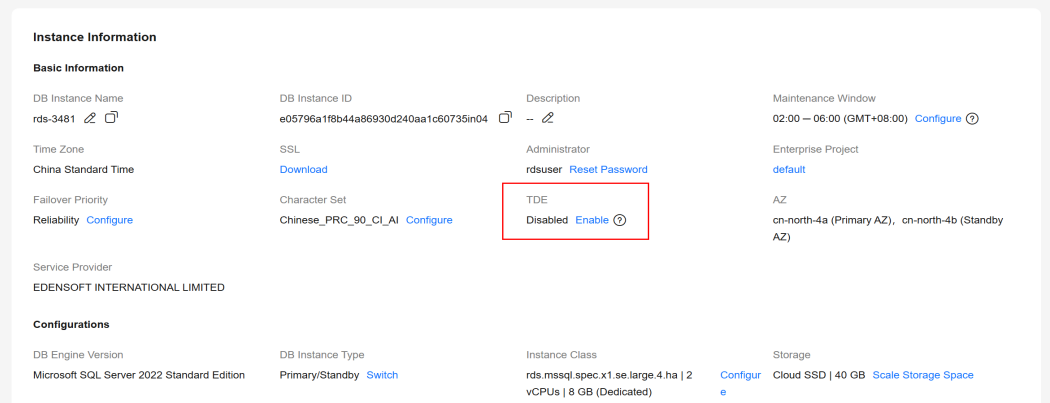
Step 2 Click in the upper left corner and select a region.

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name to go to the **Overview** page.

Step 5 Under **TDE**, click **Enable**.

Figure 4-80 Enabling instance-level TDE



Step 6 In the displayed dialog box, click **Yes**.

NOTE

Once enabled, the instance-level TDE function cannot be disabled. Exercise caution when deciding to enable instance-level TDE.

----End

Configuring Database-Level TDE

NOTE

Before enabling the database-level TDE function, ensure that the instance-level TDE function has been enabled.

Step 1 Connect to the target DB instance.

For details, see **Connecting to a DB Instance Through a Public Network**, **Connecting to a DB Instance Through a Private Network**, and **Connecting to a DB Instance Through DAS (Recommended)**.

Step 2 Use the stored procedure rds_tde to enable, disable, or query the database-level TDE status.

exec master.dbo.rds_tde *DatabaseName*,*TDE_Action*

- ***DatabaseName***: indicates the target database name, which can be **null**.
- *TDE_Action*:
  - The value **-1** indicates that the database encryption status is queried.

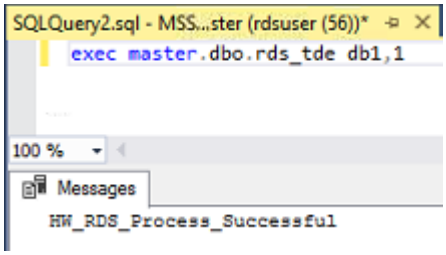
    If ***DatabaseName*** is **null**, the encryption status of all databases is returned.

- The value **0** indicates that the TDE function is disabled.
- The value **1** indicates that the TDE function is enabled.

1. Enable TDE for database db1.

   exec master.dbo.rds_tde db1, 1

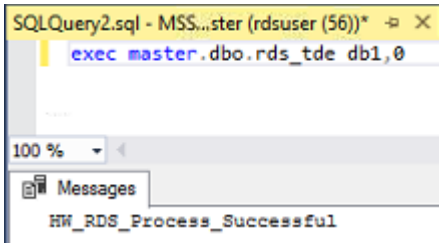
   Figure 4-81 Enabling TDE

2. Disable TDE for database db1.

   exec master.dbo.rds_tde db1, 0

   Figure 4-82 Disabling TDE

3. Query the TDE status of database db1.

   exec master.dbo.rds_tde db1, -1

   Figure 4-83 Querying the TDE status (Enabled)

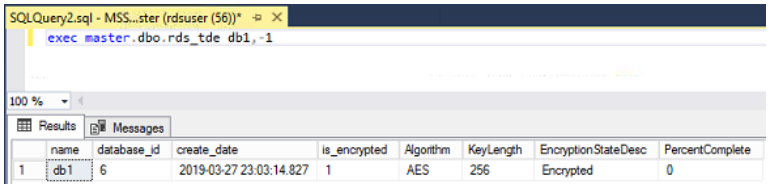
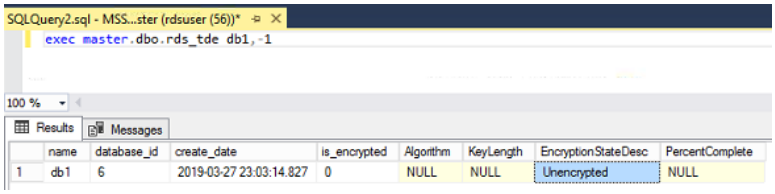
   

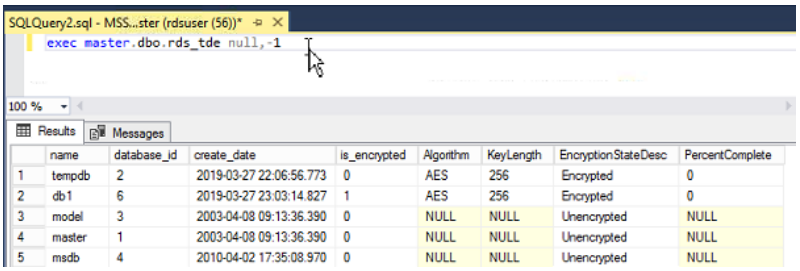
   Figure 4-84 Querying the TDE status (Disabled)

4. Query the TDE status of all databases.

   exec master.dbo.rds_tde null, -1

Figure 4-85 Querying the TDE status of all databases



	name	database_id	create_date	is_encrypted	Algorithm	KeyLength	EncryptionStateDesc	PercentComplete
1	tempdb	2	2019-03-27 22:06:56.773	0	AES	256	Encrypted	0
2	db1	6	2019-03-27 23:03:14.827	1	AES	256	Encrypted	0
3	model	3	2003-04-08 09:13:36.390	0	NULL	NULL	Unencrypted	NULL
4	master	1	2003-04-08 09:13:36.390	0	NULL	NULL	Unencrypted	NULL
5	msdb	4	2010-04-02 17:35:08.970	0	NULL	NULL	Unencrypted	NULL

----End

4.14.6 Using DBSS (Recommended)

Database Security Service (DBSS) is an intelligent database security service. Based on the machine learning mechanism and big data analytics technologies, it can audit your databases, detect SQL injection attacks, and identify high-risk operations.

You are advised to use DBSS to provide extended data security capabilities. For details, see **Database Security Service**.

Advantages

- DBSS can help you meet security compliance requirements.
  - DBSS can help you comply with DJCP (graded protection) standards for database audit.
  - DBSS can help you comply with security laws and regulations, and provide compliance reports that meet data security standards (such as Sarbanes-Oxley).
- DBSS can back up and restore database audit logs and meet the audit data retention requirements.
- DBSS can monitor risks, sessions, session distribution, and SQL distribution in real time.
- DBSS can report alarms for risky behavior and attacks and respond to database attacks in real time.
- DBSS can locate internal violations and improper operations and keep data assets secure.

Deployed in bypass pattern, database audit can perform flexible audits on the database without affecting user services.

- Database audit monitors database logins, operation types (data definition, operation, and control), and operation objects based on risky operations to effectively audit the database.
- Database audit analyzes risks and sessions, and detects SQL injection attempts so you can stay apprised of your database status.
- Database audit provides a report template library to generate daily, weekly, or monthly audit reports according to your configurations. It sends real-time alarm notifications to help you obtain audit reports in a timely manner.

4.15 Distributed Transactions

Scenarios

The participator, transaction-supported server, resource server, and transaction manager of a distributed transaction are deployed on different nodes in a distributed system. Operations contained in a transaction are considered as a logical unit, and they either succeed completely, or fail completely. Distributed transactions are used to ensure data consistency among different databases.

The Microsoft Distributed Transaction Coordinator (MSDTC) service is a component of modern versions of Microsoft Windows that is responsible for coordinating transactions that span multiple resource managers. To use distributed transactions on databases, you must enable MSDTC on each participating server. MSDTC has been enabled when you enable distributed transactions on RDS for SQL Server databases. To enable MSDTC on remote servers, see **Configuring MSDTC on a Remote Server**.

For more information, see **MS DTC Distributed Transactions**.

Constraints

- Distributed transactions are enabled for newly created DB instances by default.
- Read replicas do not support distributed transactions.
- Once enabled, distributed transactions cannot be disabled.
- Enabling distributed transactions will cause DB instance to reboot. Exercise caution when you perform this operation.
- After a database link is created for an RDS for SQL Server DB instance, if a primary/standby switchover or failover occurs, the database link will not be automatically synchronized to the new primary DB instance. You need to create a database link on the new primary DB instance again.

Enabling Distributed Transactions

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Distributed Transactions**. On the displayed page, click  in the **Distributed Transaction** field.

----End

Adding Hosts

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Distributed Transactions**. On the displayed page, click **Add Host**.

Step 6 In the displayed dialog box, enter host names and IP addresses, and click **Test Connection**. After all host connection tests are successful, click **OK**.

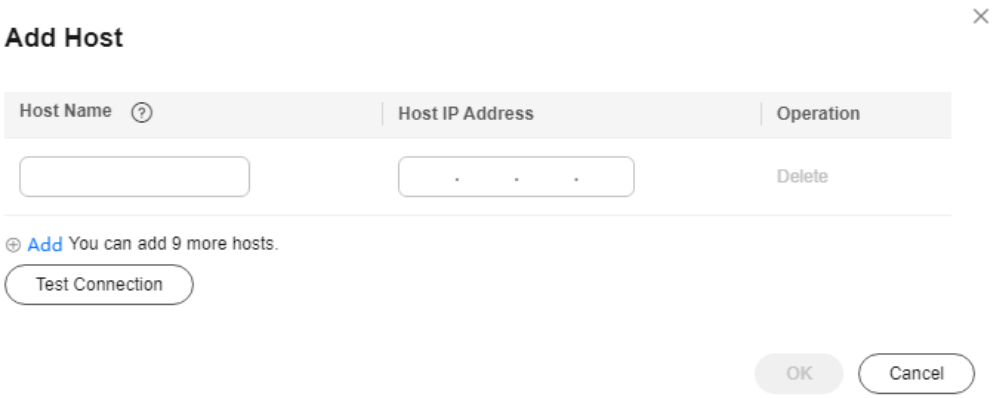
- Host IP name: Enter the names of the hosts for which you want to create distributed transactions with the RDS DB instance. Each host name must be unique and contains 1 to 64 characters, including only letters, digits, and hyphens (-).
- Host IP address: Enter the IP addresses of the hosts for which you want to create distributed transactions with the RDS DB instance. You need to configure the inbound and outbound rules in the security group for the host IP addresses first.

  For details about the requirements of security group rules, see the **Adding a Security Group Rule** section in the *Virtual Private Cloud User Guide*.

  NOTE

  - If the hosts to be added are ECSs that are in the same VPC as your RDS DB instance, enter the private IP address of the ECS. You can obtain the ECS's private IP address on the ECS details page.
  - If the hosts to be added are ECSs that are in different VPCs from the RDS DB instance, enter the public IP addresses of the ECSs. You need to bind an EIP to the RDS DB instance by referring to **Binding and Unbinding an EIP**.

Figure 4-86 Adding a host



----End

Resolving Names on Remote Servers (ECSs)

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 In the navigation pane on the left, choose **Distributed Transactions**. On the displayed page, obtain information about the RDS DB instance.

Step 5 Add the RDS DB instance information to the hosts file in **C:\Windows\System32\drivers\etc\hosts**.

----End

Configuring MSDTC on a Remote Server

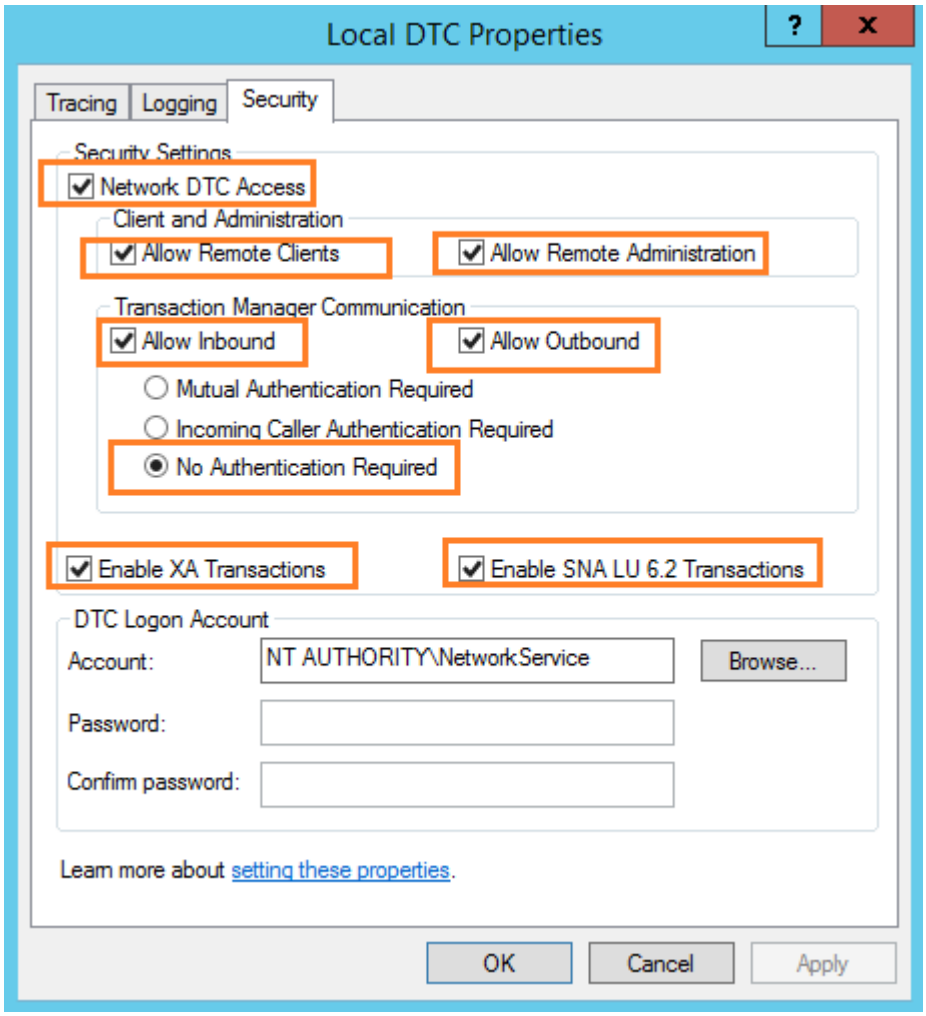
Step 1 Click **Start** and choose **Control Panel** > **Administrative Tools** > **Component Services**.

Step 2 Expand the nodes in the **Console** pane Choose **Computers** > **My Computer** > **Distributed Transaction Coordinator**.

Step 3 Right-click **Local DTC** and choose **Properties**.

Step 4 In the displayed dialog box, click the **Security** tab. Configure information as required as shown in **Figure 4-87** and click **OK**.

Figure 4-87 Local DTC properties



----End

Deleting Hosts

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane on the left, choose **Distributed Transactions**. In the host list, locate the host to be deleted and click **Delete** in the **Operation** column.

Alternatively, select one or more hosts to be deleted and click **Delete** above the list to delete hosts in batches.

Step 6 In the displayed dialog box, click **Yes**.

----End

4.16 SQL Server Integration Services (SSIS)

Scenarios

SSIS provides enterprises with data integration solutions and workflows to create business intelligence (BI). It can be used to extract, transform, and load (ETL) data from various sources. RDS for SQL Server provides the SSIS feature. You can enable SSIS, synchronize project files, authorize and deploy projects, and configure jobs to execute projects.

Constraints

- To enable this feature for your instance, you need to add the instance to an AD domain and use a domain account to log in to the instance. The AD domain name and directory address are displayed on the **Overview** page.
- SSIS cannot be enabled for read replicas.
- The parameter **clr enabled** of your instance is set to **1**.
- Only project deployment is supported.
- SQL Server Agent can be used to run SSIS packages.
- SSIS is provided only in the following editions: 2014 Standard Edition, 2014 Enterprise Edition, 2016 Standard Edition, 2016 Enterprise Edition, 2017 Standard Edition, 2017 Enterprise Edition
- The path used for building an SSIS package must start with **D:\SSIS**. After the SSIS package is deployed on the ECS, the package is automatically stored in the **D:\SSIS\{projectName}\{projectFile}** directory. Ensure that all files, parameter variables, and expressions used in the project are stored in the path starting with **D:\SSIS**.
- To deploy the SSIS package, you must run the **msdb.dbo.rds_ssis_task** stored procedure. For details, see **Deploying an SSIS Project**. Deployment of the project directly into an RDS instance is not supported.
- Use the **DontSaveSensitive** protection level to build SSIS project files (.ispac) for deployment.
- Do not create or restore the database with the name SSISDB. Otherwise, SSIS of your instance may be unavailable.
- The SSIS project files must be uploaded to an OBS bucket. The **.zip** and **.ispac** files are supported. The file name must be the same as the project name. The ZIP package must contain the **.ispac** project files. The file name can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).

Enabling SSIS

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, click **SSIS**. On the displayed page, click next to **Enable SSIS**.

Step 6 In the displayed dialog box, click **Yes** to enable SSIS.

- This function cannot be disabled after being enabled.
- After SSIS is enabled, the parameter **clr enabled** is set to **1** by default. Do not disable this parameter. Otherwise, SSIS cannot work properly.

----End

After SSIS is enabled, you can add an SSIS package.

- After this function is enabled, the instance enters the data synchronization state. After the SSISDB synchronization is complete, the instance becomes available.
- You need to add the RDS host information to the ECS or local device added to the AD domain so that you can access the database from the ECS or local device.

Adding an SSIS package

NOTE

Before performing the following steps, upload SSIS project files to the OBS bucket.

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

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

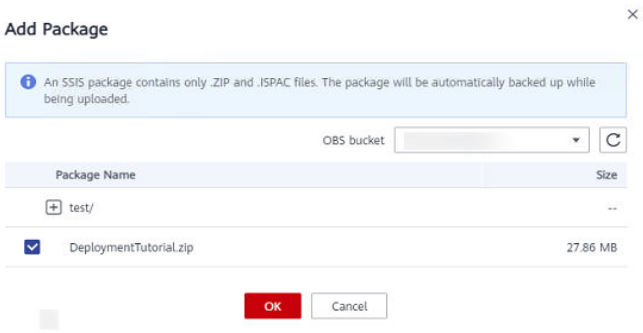
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **SSIS** and click **Add Package**.

Step 6 In the displayed dialog box, select a package name and click **OK**. After the package is added, information about the package is displayed on the **SSIS** page.

Figure 4-88 Adding a package



----End

Deploying an SSIS Project

Step 1 Use SQL Server Management Studio to connect to the database.

Step 2 Run the stored procedure **master.dbo.rds_grant_ssis_to_login** to grant SSIS-related permissions to the domain account. For details, see **Granting SSIS Permissions to a Domain Account**.

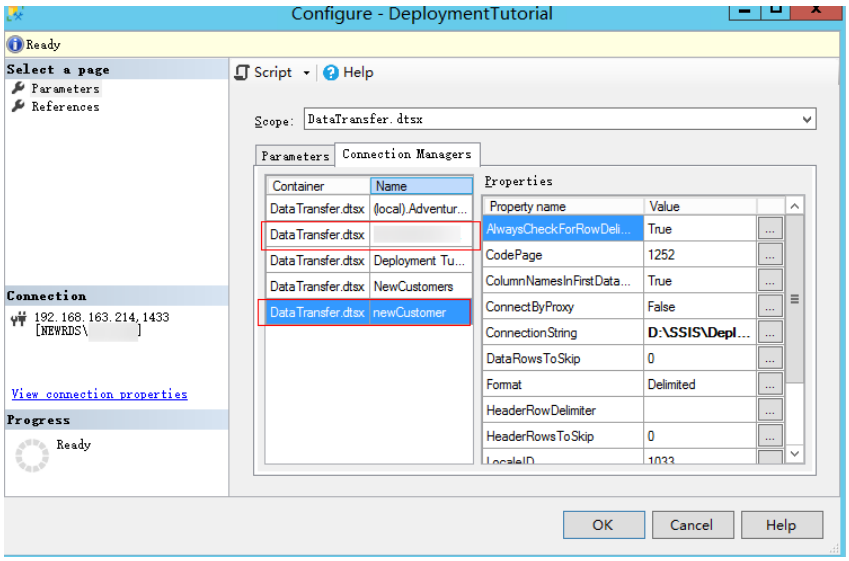
Step 3 Choose **Integration Service Catalogs** > **SSISDB**, right-click to create an SSIS folder, and enter a name for the folder. Then, two subfolders **Projects** and **Environments** are automatically created.

Step 4 Run the stored procedure **msdb.dbo.rds_ssis_task** using the domain account to deploy the SSIS package. For details, see **Deploying an SSIS Project**.

Step 5 Configure and execute the package. Before executing the package, configure password information for the connection manager and package parameters because **DontSaveSensitive** has been selected as the project protection level for building the package.

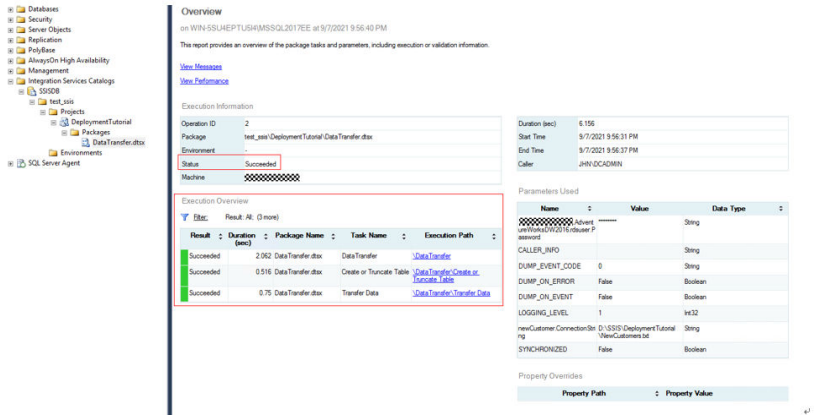
1. Right-click the package name, choose **Configure** from the shortcut menu, and configure parameters.

   Figure 4-89 Configuring parameters



2. After the configuration is complete, click **Execute** to run the SSIS project. If execution information similar to what is shown in the following figure is displayed, the SSIS project has been executed.

Figure 4-90 Execution information



Step 6 Create a credential that you use to execute the SSIS package. Specifically, choose **Security** > **Credentials**, right-click and choose **New Credentials** from the shortcut menu. On the displayed page, enter the domain account information.

Step 7 Run the following SQL statements to create an SSIS proxy:

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy @proxy_name=N'test_proxy', @credential_name=N'ssis_credential',
@enabled=1
go
exec msdb.dbo.rds_grant_proxy_subsystem 'test_proxy', 'SSIS'

USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy @proxy_name=N'test_proxy', @login_name=N'JHN\dcadmin'
GO
```

- **sp_add_proxy**: a system stored procedure for creating a proxy (**@proxy_name**) and accessing the proxy credential (**@credential_name**)
- **sp_grant_login_to_proxy**: a system stored procedure for granting the account (**@login_name**) the permission to access the proxy (**@proxy_name**)
- **rds_grant_proxy_subsystem**: a stored procedure provided by RDS for granting subsystem permissions to the proxy

The parameters in the stored procedures are explained as follows:

@proxy_name: the name of the proxy to create.

@proxy_subsystem: the subsystem name. To grant SSIS subsystem permissions to the proxy, set this parameter to **SSIS**.

Step 8 To create an SQL Server Agent job, choose **SQL Server Agent** > **Jobs** and enter a job name. Then, add a step to execute the SSIS package. You can view the job details once the job is created.

Step 9 Right-click the job name, choose **start job** from the shortcut menu, and wait for the execution result.

Step 10 Check the SSIS project and operation records.

----End

4.17 Metrics and Alarms

4.17.1 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 4-10 lists the performance metrics of RDS for SQL Server DB instances.

Table 4-10 Performance metrics

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	0–100	%	N/A	RDS for 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	N/A	RDS for SQL Server instance	1 minute
rds039_disk_util	Storage Space Usage	Storage space usage of the monitored object	0–100	%	N/A	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds002_mem_util	Memory Usage	Memory usage of the monitored object	0–100	%	N/A	RDS for SQL Server instance	1 minute
rds004_bytes_in	Network Input Throughput	Incoming traffic in bytes per second	≥ 0	bytes/s	1024	RDS for SQL Server instance	1 minute
rds005_bytes_out	Network Output Throughput	Outgoing traffic in bytes per second	≥ 0	bytes/s	1024	RDS for SQL Server instance	1 minute
rds049_disk_read_throughput	Disk Read Throughput	Number of bytes read from the disk per second	≥ 0	bytes/s	1024	RDS for SQL Server instance	1 minute
rds050_disk_write_throughput	Disk Write Throughput	Number of bytes written into the disk per second	≥ 0	bytes/s	1024	RDS for SQL Server instance	1 minute
rds047_disk_total_size	Total Storage Space	Total storage space of the monitored object	40–4000	GB	1024	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds048_disk_used_size	Used Storage Space	Used storage space of the monitored object	0–4000	GB	1024	RDS for 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	counts	N/A	RDS for SQL Server instance	1 minute
rds054_db_connections_in_use	Database Connections in Use	Number of database connections in use	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds055_transactions_per_sec	Transactions per Second	Number of transactions started for the database per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds056_batch_per_sec	Batches per Second	Number of Transact-SQL command batches received per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds057_logins_per_sec	Logins per Second	Total number of logins started per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds058_logouts_per_sec	Logouts per Second	Total number of logouts started per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds059_cache_hit_ratio	Cache Hit Ratio	Ratio of pages found in the buffer cache without having to read from the disk to total pages	0–100	%	N/A	RDS for SQL Server instance	1 minute
rds060_sql_compilations_per_sec	SQL Compilations per Second	Number of SQL compilations per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds061_sql_recompilations_per_sec	SQL Recompilations per Second	Number of SQL recompilations per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds062_full_scans_per_sec	Full Scans per Second	Number of unrestricted full scans per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds063_errors_per_sec	Errors per Second	Number of errors per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds064_latch_waits_per_sec	Latch Waits per Second	Number of latch requests that have not been granted immediately	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds065_lock_waits_per_sec	Lock Waits per Second	Number of lock wait requests per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds066_lock_requests_per_sec	Lock Requests per Second	Number of new locks and lock conversions per second requested from the lock manager	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds067_timeouts_per_sec	Lock Timeouts per Second	Number of lock timeouts per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds068_avg_lock_wait_time	Average Lock Wait Time	Average wait time (ms) of lock requests	≥ 0	ms	N/A	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds069_deadlocks_per_sec	Deadlocks per Second	Number of deadlocks per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
rds070_checkpoint_pages_per_sec	Checkpoint Pages per Second	Number of pages flushed to the disk per second by a checkpoint or other operations that require all dirty pages to be flushed	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
rds077_replication_delay	Replication Delay	Delay for replication between primary and standby DB instances. The replication delay of RDS for SQL Server DB instances is at the database level because data is synchronized on each database. The instance-level replication delay refers to the maximum replication delay of the databases (the delay 0s for single DB instances).	≥ 0	s	N/A	RDS for SQL Server instance	1 minute
mssql_mem_grant_pending	Memory Grants Pending	Total number of processes waiting for a workspace memory grant	≥ 0	counts	N/A	RDS for SQL Server instance	1 minute

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
mssql_lazy_write_per_sec	Lazy Writes per Second	Number of lazy writes per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
mssql_page_life_expectancy	Page Life Expectancy	Number of seconds a page will stay in the buffer pool without references	≥ 0	s	N/A	RDS for SQL Server instance	1 minute
mssql_page_reads_per_sec	Page Reads per Second	Number of page reads per second	≥ 0	counts/s	N/A	RDS for SQL Server instance	1 minute
mssql_tempdb_disk_size	Temporary Tablespace Size	Disk space occupied by the current temporary tablespace.	≥ 0	MB	1024	RDS for SQL Server instance	1 minute
mssql_worker_threads_usage_rate	Usage of Worker Threads	Ratio of the total worker threads to the value of **Max Worker Threads**.	0–100	%	N/A	RDS for SQL Server instance	1 minute

Dimension

Key	Value
rds_cluster_sqlserver_id	RDS for SQL Server instance ID

4.17.2 Viewing Monitoring Metrics

Scenarios

Cloud Eye monitors the statuses of RDS DB instances. You can view RDS monitoring metrics on the management console. For details, see **Viewing Metrics of Primary DB Instances**.

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.

Viewing Metrics of Primary DB Instances

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, click the target DB instance. On the displayed page, click **View Metrics** 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.

- On the Cloud Eye console, click **Select Metric** in the upper right corner. In the displayed dialog box, you can select the metrics to be displayed and sort them by dragging them at desired locations.
- You can sort graphs by dragging them based on service requirements.
- You can view the performance metrics in the last 1 hour, 3 hours, 12 hours, 1 day, 6 months, and 7 days.

----End

4.17.3 Setting Alarm Rules

Scenarios

You can set 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

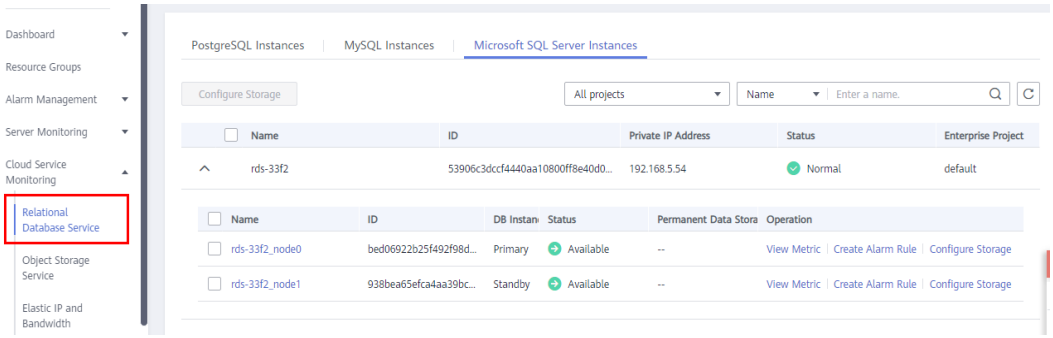
Step 1 **Log in to the management console**.

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

Step 3 Click **Service List**. Under **Management & Governance**, click **Cloud Eye**.

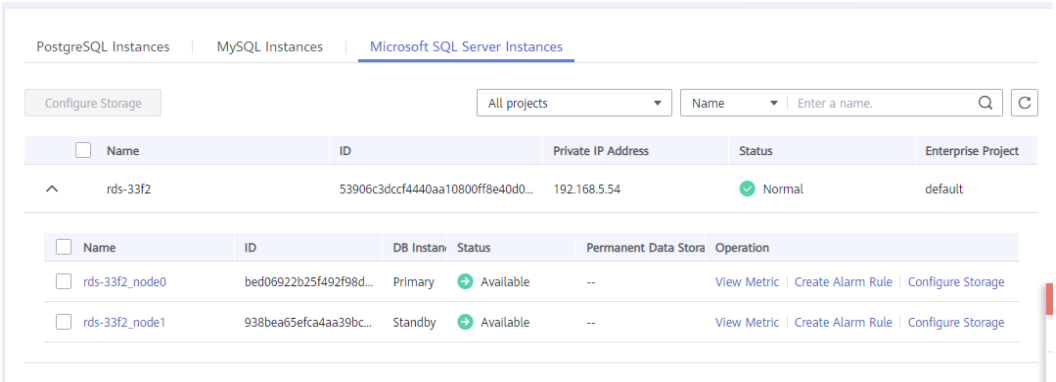
Step 4 In the navigation pane on the left, choose **Cloud Service Monitoring** > **Relational Database Service**.

Figure 4-91 Choosing a monitored object



Step 5 Locate the DB instance for which you want to create an alarm rule and click **Create Alarm Rule** in the **Operation** column.

Figure 4-92 Creating an alarm rule



Step 6 On the displayed page, set parameters as required.

Figure 4-93 Configuring alarm information

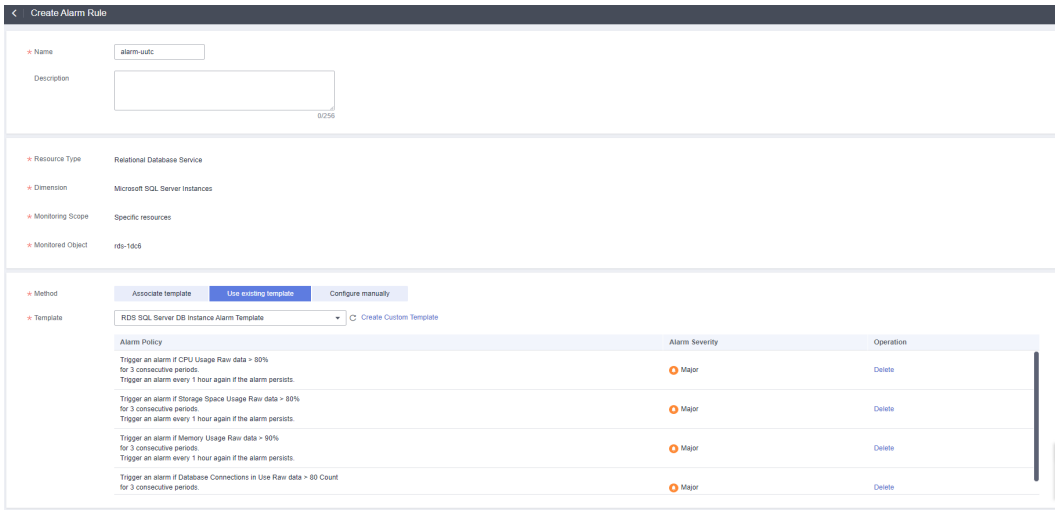


Table 4-11 Alarm rule information

Parameter	Description
Name	Alarm rule name. The system generates a random name, which you can modify.
Description	Description about the rule.
Method	There are three options: **Associate template**, **Use existing template**, and **Configure manually**. **NOTE** If you select **Associate template**, after the associated template is modified, the policies contained in this alarm rule to be created will be modified accordingly. You are advised to select **Use existing template**. The existing templates already contain four common alarm metrics: CPU usage, storage space usage, memory usage, and database connections in use.
Template	Select the template to be used. You can select a default alarm template or create a custom template.
Alarm Policy	Policy for triggering an alarm. Whether to trigger an alarm depends on whether the metric data in consecutive periods reaches the threshold. For example, Cloud Eye triggers an alarm if the average CPU usage of the monitored object is 80% or more for three consecutive 5-minute periods. **NOTE** A maximum of 50 alarm policies can be added to an alarm rule. If any one of these alarm policies is met, an alarm is triggered.
Alarm Severity	The alarm severity can be **Critical**, **Major**, **Minor**, or **Informational**.

Figure 4-94 Configuring alarm notification

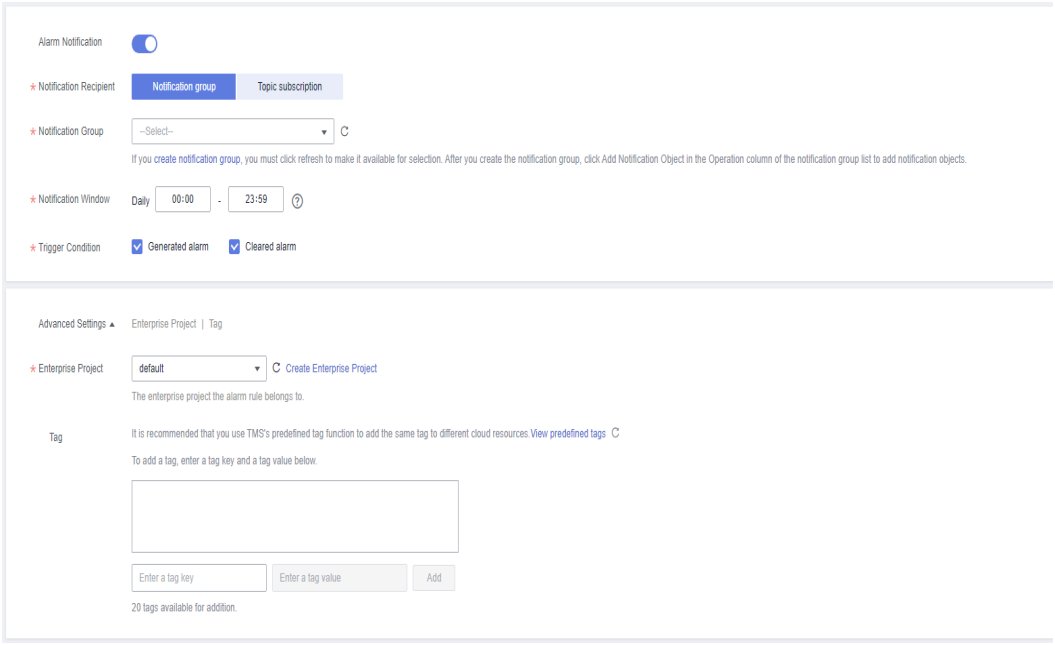


Table 4-12 Alarm notification

Parameter	Description
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Recipient	You can select a notification group or topic subscription as required.
Notification Group	Notification group the alarm notification is to be sent to.
Notification Object	Object the alarm notification is to be sent to. You can select the account contact or a topic. • The account contact is the mobile phone number and email address of the registered account. • A topic is used to publish messages and subscribe to notifications.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Condition for triggering an alarm notification. You can select **Generated alarm** (when an alarm is generated), **Cleared alarm** (when an alarm is cleared), or both.
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users with the enterprise project permissions can view and manage the alarm rule.

Parameter	Description
Tag	A tag is a key-value pair. Tags identify cloud resources so that you can easily categorize and search for your resources.

Step 7 Click **Create**. The alarm rule is created.

For details about how to create alarm rules, see **Creating an Alarm Rule** in the *Cloud Eye User Guide*.

----End

4.17.4 Event Monitoring

4.17.4.1 Introduction to Event Monitoring

Event monitoring provides event data reporting, query, and alarm reporting. You can create alarm rules for both system and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on RDS resources that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, for example, resetting the administrator password or modifying the backup policy.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see **Events Supported by Event Monitoring**.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by services to Cloud Eye.

For details about how to report custom events, see **Reporting Events**.

4.17.4.2 Viewing Event Monitoring Data

Scenarios

This section describes how to view the event monitoring data.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the DB instance and click **View Metrics** in the **Operation** column to go to the Cloud Eye console.

Alternatively, go to the Cloud Eye console using the following method:

On the **Instances** page, click the DB instance name. On the displayed **Overview** page, click **View Metrics** in the upper right corner.

Step 5 Click < to return to the main page of Cloud Eye.

Step 6 In the navigation pane on the left, choose **Event Monitoring**.

On the displayed **Event Monitoring** page, all system events generated in the last 24 hours are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view the events generated in different periods.

Step 7 Click **View Graph**. On the details page, click **View Event** in the **Operation** column of a specific event to view details.

----End

4.17.4.3 Creating an Alarm Rule to Monitor an Event

Scenarios

This section describes how to create an alarm rule to monitor an event.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

Step 3 In the navigation pane on the left, choose **Event Monitoring**.

Step 4 On the event list page, click **Create Alarm Rule** in the upper right corner.

Step 5 On the **Create Alarm Rule** page, configure the parameters.

Table 4-13 Parameter description

Parameter	Description
Name	Specifies the alarm rule name. The system generates a random name, which you can modify.
Description	(Optional) Provides supplementary information about the alarm rule.
Enterprise Project	You can select an existing enterprise project or click **Create Enterprise Project** to create one.
Alarm Type	Specifies the alarm type corresponding to the alarm rule.
Event Type	Specifies the event type of the metric corresponding to the alarm rule.

Parameter	Description
Event Source	Specifies the service the event is generated for. Select **Relational Database Service**.
Monitoring Scope	Specifies the monitoring scope for event monitoring. • **All resources**: If you select **All resources**, an alarm will be triggered when any RDS DB instance meets an alarm policy, and existing alarm rules will be automatically applied for newly purchased resources. • **Resource groups**: If you select **Resource groups**, an alarm will be triggered when any resource in the group meets the alarm policy. • **Specific resources**: Currently, RDS for SQL Server instance resources cannot be specified.
Method	Specifies the means you use to create the alarm rule.
Alarm Policy	**Event Name** indicates the instantaneous operations users performed on system resources, such as login and logout. For events supported by event monitoring, see **Events Supported by Event Monitoring**. You can select a trigger mode and alarm severity as needed.

Click  to enable alarm notification. The validity period is 24 hours by default. If the topics you require are not displayed in the drop-down list, click **Create an SMN topic**.

Table 4-14 Alarm notification

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email, text message, or HTTP/HTTPS message.
Notification Object	Specifies the object that receives alarm notifications. You can select the account contact or a topic. • **Account contact** is the mobile phone number and email address of the registered account. • **Topic** is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see **Creating a Topic** and **Adding Subscriptions**.
Notification Window	Cloud Eye sends notifications only within the notification window specified in the alarm rule. If **Notification Window** is set to **08:00-20:00**, Cloud Eye sends notifications only within 08:00-20:00.

Parameter	Description
Trigger Condition	Specifies the condition for triggering the alarm notification.

Step 6 Click **Create**.

----End

4.17.4.4 Events Supported by Event Monitoring

Table 4-15 Resource exception events

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
RDS	DB instance creation failure	createInstanceFailed	Major	A DB instance fails to create because the number of disks is insufficient, the quota is insufficient, or underlying resources are exhausted.	Check the number of disks and quota size. Release resources and create DB instances again.	DB instances cannot be created.
	Full backup failure	fullBackupFailed	Major	A single full backup failure does not affect the files that have been successfully backed up, but prolongs the incremental backup restoration time during the point-in-time recovery (PITR).	Create a manual backup again.	Backup failed.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Primary/standby switchover failure	activeStandBySwitchFailed	Major	The standby DB instance does not take over workloads from the primary DB instance due to network or server failures. The original primary DB instance continues to provide workloads within a short time.	Check whether the connection between your application and the database is re-established.	None
	Replication status abnormal	abnormalReplicationStatus	Major	The possible causes are as follows: The replication delay between the primary and standby instances is too long, which usually occurs when a large amount of data is written to databases or a large transaction is processed. During peak hours, data may be blocked. The network between the primary and standby instances is disconnected.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Replication status recovered	replicationStatusRecovered	Major	The replication delay between the primary and standby instances is within the normal range, or the network connection between them has restored.	No action is required.	None
	DB instance faulty	faultyDBInstance	Major	A single or primary DB instance was faulty due to a disaster or a server failure.	Check whether an automated backup policy has been configured for the DB instance and submit a service ticket.	The database service may be unavailable.
	DB instance recovered	DBInstanceRecovered	Major	RDS rebuilds the standby DB instance with its high availability. After the instance is rebuilt, this event will be reported.	No action is required.	None

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Failure of changing single DB instance to primary/ standby	singleToHaFailed	Major	A fault occurs when RDS is creating the standby DB instance or configuring replication between the primary and standby DB instances. The fault may occur because resources are insufficient in the data center where the standby DB instance is located.	Submit a service ticket.	Your applications are not affected because this event does not interrupt data read and write of the DB instance.
	Database process restarted	DatabaseProcessRestarted	Major	The database process is stopped due to insufficient memory or high load.	Log in to the Cloud Eye console. Check whether the memory usage increases sharply, the CPU usage is too high for a long time, or the storage space is insufficient. You can increase the CPU and memory specifications or optimize the service logic.	Downtime occurs. In this case, RDS automatically restarts the database process and attempts to recover the workloads.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	Instance storage full	instanceDiskFull	Major	Generally, the cause is that the data space usage is too high.	Scale up the instance.	The DB instance becomes read-only because the storage space is full, and data cannot be written to the database.
	Instance storage full recovered	instanceDiskFullRecovered	Major	The instance disk is recovered.	No action is required.	The instance is restored and supports both read and write operations.

Event Source	Event Name	Event ID	Event Severity	Description	Solution	Impact
	RDS for SQL Server publication/ subscription error	mssqlReplicationError	Major	An error is reported for RDS for SQL Server publication and subscription.	Rectify the fault based on the provided details.	Data synchronization from the RDS for SQL Server publisher to the subscriber is affected.
	Kafka connection failed	kafkaConnectionFailed	Major	The network is unstable or the Kafka server does not work properly.	Check your network connection and the Kafka server status.	Audit logs cannot be sent to the Kafka server.

Table 4-16 Operation events

Event Source	Event Name	Event ID	Event Severity	Description
RDS	Reset administrator password	resetPassword	Major	The password of the database administrator is reset.
	Operate DB instance	instanceAction	Major	The storage space is scaled or the instance class is changed.

Event Source	Event Name	Event ID	Event Severity	Description
	Delete DB instance	deleteInstance	Minor	The DB instance is deleted.
	Modify backup policy	setBackupPolicy	Minor	The backup policy is modified.
	Modify parameter group	updateParameterGroup	Minor	The parameter group is modified.
	Delete parameter group	deleteParameterGroup	Minor	The parameter group is deleted.
	Reset parameter group	resetParameterGroup	Minor	The parameter group is reset.
	Change database port	changeInstancePort	Major	Change database port
	Primary/standby switchover or failover	PrimaryStandbySwitched	Major	A switchover or failover is performed.

4.18 Interconnection with CTS

4.18.1 Key Operations Supported by CTS

Cloud Trace Service (CTS) records operations related to RDS for SQL Server instances for further query, audit, and backtrack.

NOTE

This section lists only common key operations.

Table 4-17 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
Deleting a tag (using the console or open APIs)	instance	deleteTag
Modifying a tag (using the console or open APIs)	instance	modifyTag

Operation	Resource Type	Trace Name
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
Comparing parameter templates (using the console)	parameterGroup	compareParameterGr oup
Applying a parameter template (using the console)	parameterGroup	applyParameterGrou p

Operation	Resource Type	Trace Name
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
Creating a database account	instance	createDBUser
Resetting a password	instance	resetDBUserPassword
Changing account permissions	instance	grantDBUser
Deleting a database account	instance	deleteDBUser
Creating a database	instance	createDatabase
Authorizing a database	instance	grantDBUser
Deleting a database	instance	deleteDatabase

4.18.2 Viewing Tracing Events

For details about how to view audit logs, see **Querying Real-Time Traces**.

4.19 Log Management

4.19.1 Viewing and Downloading System Logs

Scenarios

System logs contain logs generated during the database running. These can help you analyze problems with the database. You can also download system logs for service analysis.

Viewing Log Details

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

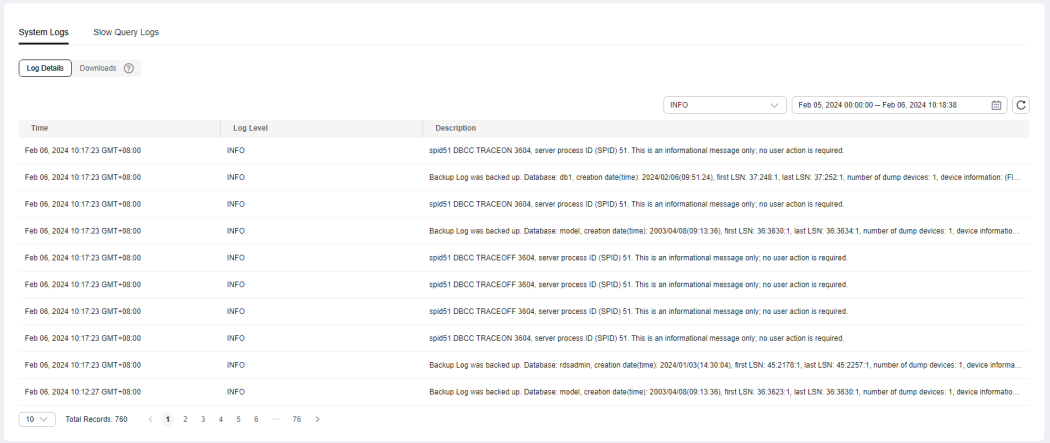
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Log Details** to view details about system logs.

Figure 4-95 System log details



- You can select a log level in the upper right corner to view logs of the selected level.

  NOTE

  For RDS for SQL Server DB instances, only info-level logs are displayed currently.

- You can click in the upper right corner to view 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

Download a Log

Step 1 Log in to the management console.

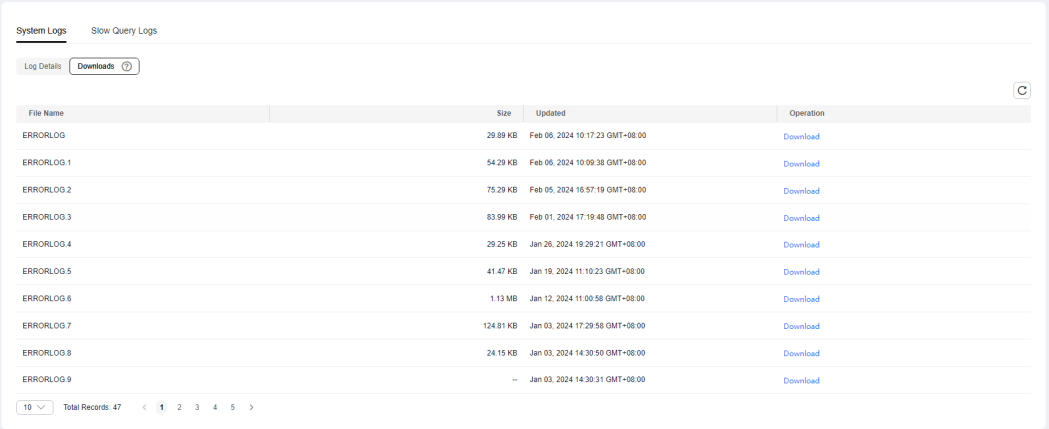
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Downloads**.

Figure 4-96 Downloading system logs



NOTE

- **ERRORLOG** refers to error logs.
- *xxxx*.**xel** refers to extended event logs.
- *xxxx*.**trc** refers to default trace logs.
- Logs whose names start with **RDSAudit** are audit logs. A GUID and timestamp are automatically added to an audit log name. For details, see **Viewing and Downloading Audit Logs**.

1. Locate a log to be downloaded 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**.

2. In the displayed dialog box, click **OK** to download the log whose status is **Preparation completed**. If you click **Cancel**, the system does not download the log and returns to the **Downloads** page.

   If the size of a log to be downloaded is greater than 40 MB, you need to use OBS Browser+ to download it. For details, see **Method 1: Using OBS Browser +**.

   The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. You can close the window and repeat the procedure **Step 5.1** to try to download a log again.

----End

4.19.2 Viewing and Downloading Audit Logs

The SQL audit function is enabled by default for Microsoft SQL Server DB instances and cannot be disabled. Major change operations on services, databases, and tables are recorded in audit log files for future query and download.

RDS for SQL Server Audit enables you to audit server-level and database-level groups of events and individual events. RDS for SQL Server audits consist of zero or more audit action items. **Table 4-18** shows the server-level audit action groups and provides the equivalent RDS for SQL Server Event Class where applicable. For more information, see **SQL Server Audit Action Groups and Actions**.

NOTE

- The maximum size of an audit log file is 50 MB. Up to 50 audit log files can be displayed.
- RDS for SQL Server 2008 Web and Standard Editions do not support the SQL audit function.
- No audit is performed for job creation and modifications on parameters, server attribute parameters, SQL agent attribute parameters, and database extended attribute parameters.
- The **succeeded** parameter displayed in the audit log indicates whether the event is triggered successfully. Its value cannot be **null**. For all events except login events, only the success or failure of the permission check (not the operation) is reported.
- For details about the audit of table-level and column-level architecture changes, see the audit result of the SQL Server engine.
- To read audit logs, you can obtain the audit log file name from the console and then run the following statement:

  select * from msdb.dbo.rds_fn_get_audit_file('D:\ServerAudit\audit\RDSAudit_test.sqlaudit', default, default)

  If you have already downloaded the audit log file to a local directory, log in to your local SQL Server database and then run the following statement (the local account must have the CONTROL SERVER permission):

  select * from sys.fn_get_audit_file('\\path\RDSAudit_test.sqlaudit', default, default)

Table 4-18 Audit action groups

Action Group Name	Description
APPLICATION_ROLE_CHANGE_PASSWORD_GROUP	This event is raised whenever a password is changed for an application role.
DATABASE_CHANGE_GROUP	This event is raised when a database is created, altered, or dropped.
DATABASE_OBJECT_CHANGE_GROUP	This event is raised when a CREATE, ALTER, or DROP statement is executed on database objects, such as schemas.
DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP	This event is raised when a change of owner for objects within database scope.
DATABASE_OBJECT_PERMISSION_CHANGE_GROUP	This event is raised when a GRANT, REVOKE, or DENY has been issued for database objects, such as assemblies and schemas.

Action Group Name	Description
DATABASE_OWNERSHIP_CHANGE_GROUP	This event is raised when you use the ALTER AUTHORIZATION statement to change the owner of a database.
DATABASE_PERMISSION_CHANGE_GROUP	This event is raised whenever a GRANT, REVOKE, or DENY is issued for a statement permission by any user in SQL Server for database-only events such as granting permissions on a database.
DATABASE_PRINCIPAL_CHANGE_GROUP	This event is raised when principals, such as users, are created, altered, or dropped from a database.
DATABASE_ROLE_MEMBER_CHANGE_GROUP	This event is raised whenever a login is added to or removed from a database role.
FAILED_LOGIN_GROUP	Indicates that a principal tried to log on to a SQL Server database and failed. Events in this class are raised by new connections or by connections that are reused from a connection pool.
LOGIN_CHANGE_PASSWORD_GROUP	This event is raised whenever a login password is changed by way of ALTER LOGIN statement or sp_password stored procedure.
SCHEMA_OBJECT_CHANGE_GROUP	This event is raised when a CREATE, ALTER, or DROP operation is performed on a schema.
SCHEMA_OBJECT_OWNERSHIP_CHANGE_GROUP	This event is raised when the permissions to change the owner of schema object (such as a table, procedure, or function) is checked.
SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP	This event is raised whenever a grant, deny, revoke is performed against a schema object.
SERVER_OBJECT_CHANGE_GROUP	This event is raised for CREATE, ALTER, or DROP operations on server objects.
SERVER_OBJECT_OWNERSHIP_CHANGE_GROUP	This event is raised when the owner is changed for objects in server scope.
SERVER_OBJECT_PERMISSION_CHANGE_GROUP	This event is raised whenever a GRANT, REVOKE, or DENY is issued for a server object permission by any principal in SQL Server.

Action Group Name	Description
SERVER_PERMISSION_CHANGE_GROUP	This event is raised when a GRANT, REVOKE, or DENY is issued for permissions in the server scope.
SERVER_PRINCIPAL_CHANGE_GROUP	This event is raised when server principals are created, altered, or dropped.
SERVER_ROLE_MEMBER_CHANGE_GROUP	This event is raised whenever a login is added or removed from a fixed server role.
SERVER_STATE_CHANGE_GROUP	This event is raised when the SQL Server service state is modified.
USER_CHANGE_PASSWORD_GROUP	This event is raised whenever the password of a contained database user is changed by using the ALTER USER statement (SQL Server 2008 is not involved).

Querying Audit Logs

Step 1 Log in to the management console.

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

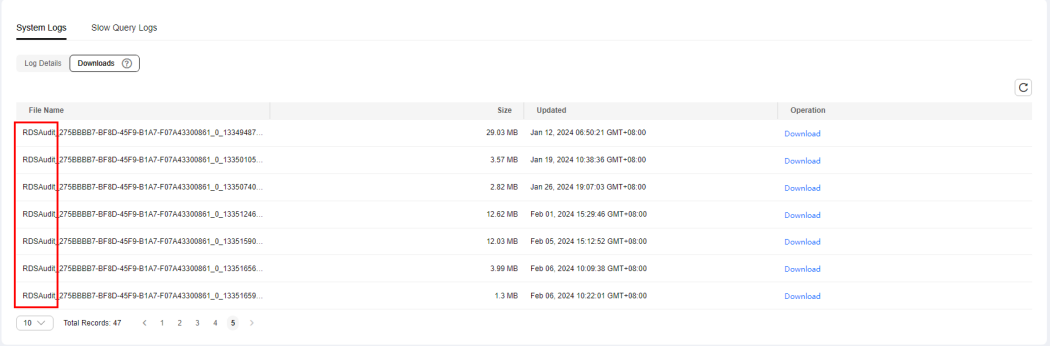
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Downloads**.

Step 6 On the **Downloads** page, record the names of audit logs.

Figure 4-97 RDS for SQL Server audit logs



NOTE

The audit log name starts with RDSAudit. The system automatically adds the GUID and timestamp to the file name as a suffix.

Step 7 Connect to the DB instance through the Microsoft SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Step 8 After the DB instance is connected, run the following command to view details about SQL audit logs:

select * from msdb.dbo.rds_fn_get_audit_file(*file_pattern***, *initial_file_name*, *audit_record_offset*)**

Table 4-19 Parameter description

Parameter	Description
file_pattern	Specifies the directory or path and file name for the audit file set to be read.
initial_file_name	Specifies the path and name of a specific file in the audit file set to start reading audit records from.
audit_record_offset	Specifies a known location with the file specified for the **initial_file_name**.

Example:

select * from msdb.dbo.rds_fn_get_audit_file('D:\ServerAudit\audit*.sqlaudit', default, default)

Figure 4-98 Audit log details

	event_time	sequence_number	action_id	succeeded	permission_bitmask	is_column_permission	session_id	server_principal_id	database_principal_id	target_server_principal_id	target_database_pri
1	2019-09-25 06:39:32.5900693	1	SVSD	1	0x00000000000000000000000000000000	0	8	1	1	0	0
2	2019-09-25 06:39:00.2463703	1	CR	1	0x00000000000000000000000000000000	0	56	274	1	0	0
3	2019-09-25 06:39:00.1369470	1	CR	1	0x00000000000000000000000000000000	0	56	1	1	0	0
4	2019-09-25 06:38:59.9963632	1	CR	1	0x00000000000000000000000000000000	0	56	1	1	0	0
5	2019-09-25 06:38:59.8557595	1	CR	1	0x00000000000000000000000000000000	0	59	1	1	0	0
6	2019-09-25 06:38:59.7151252	1	CR	1	0x00000000000000000000000000000000	0	56	1	1	0	0
7	2019-09-25 06:38:59.5901139	1	CR	1	0x00000000000000000000000000000000	0	56	1	1	0	0
8	2019-09-25 06:38:59.4807331	1	CR	1	0x00000000000000000000000000000000	0	56	1	1	0	0
9	2019-09-25 06:38:59.3713745	1	CR	1	0x00000000000000000000000000000000	0	56	1	1	0	0
10	2019-09-25 06:38:59.2307307	1	CR	1	0x00000000000000000000000000000000	0	56	274	1	0	0
11	2019-09-25 06:35:36.7619774	1	APRL	1	0x00000000000000000000000000000000	0	57	1	1	273	5
12	2019-09-25 06:35:36.7619774	1	D	1	0x00000000000000000000000000000000	0	57	1	1	273	5
13	2019-09-25 06:35:36.0901367	1	D	1	0x00000000000000000000000000000000	0	57	1	1	273	7
14	2019-09-25 06:35:36.0901367	1	APRL	1	0x00000000000000000000000000000000	0	57	1	1	273	25

----End

Downloading SQL Audit Logs

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Downloads**.

Step 6 Locate a log to be downloaded and click **Download** in the **Operation** column.

1. 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**.
2. In the displayed dialog box, click **OK** to download the log whose status is **Preparation completed**. If you click **Cancel**, the system does not download the log.

   The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. You can close the window and repeat the procedure **Step 6** to try to download a log again.

----End

4.19.3 Viewing and Downloading Slow Query Logs

Scenarios

Slow query logs record statements that exceed the **long_query_time** value (1 second by default). You can view log details to identify statements that are executing slowly and optimize the statements. You can also download slow query logs for service analysis.

Parameter Description

Table 4-20 Parameters related to RDS for SQL Server slow queries

Parameter	Description
long_query_time	Specifies how many microseconds a SQL query has to take to be defined as a slow query log. The default value is 1s. When the execution time of an SQL statement exceeds the value of this parameter, the SQL statement is recorded in slow query logs. You can modify the slow log threshold as required. 1. **Log in to the management console**. 2. Click in the upper left corner and select a region. 3. Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**. 4. On the **Instances** page, click the target instance name. 5. In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click next to the **Threshold of Slow Query Log (long_query_time)** field to change the threshold. • To submit the change, click . • To cancel the change, click . **NOTE** The recommended value is **1s**. The lock wait time is not calculated into the query time.

Viewing Slow Query Logs

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

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

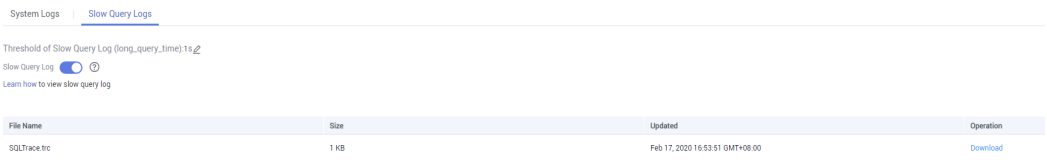
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click to enable the slow query log function.

Step 6 The generated slow query logs are displayed.

Figure 4-99 Slow query logs



NOTE

Enabling slow query log slightly affects DB instance performance.

Step 7 Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Step 8 After the DB instance is connected, run the following command to view slow query log details:

select * from ::fn_trace_gettable('D:\SQLTrace\audit*XXX*', default)

NOTE

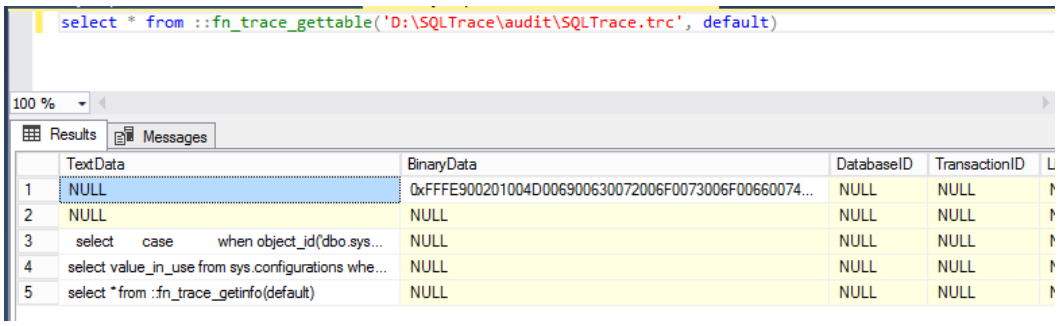
XXX indicates the name of the slow query log recorded in **Step 6**.

Example:

select * from ::fn_trace_gettable('D:\SQLTrace\audit\SQLTrace.trc', default)

The result is shown in **Figure 4-100**.

Figure 4-100 Slow query log details



----End

Downloading a Log

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click to enable the slow query log function.

NOTE

Enabling slow query log slightly affects DB instance performance.

Step 6 Locate a log to be downloaded and click **Download** in the **Operation** column.

1. 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**.

2. You can determine how to download a log file depending on the file size.
   - Only logs no more than 40 MB can be downloaded directly from this page. The time range is calculated from the time you download the logs back to the time when the accumulated file size reaches 40 MB.
   - It is impossible to generate a log file much larger than 40 MB, like 100 MB or 200 MB. If a log file that is a little larger than 40 MB is required, use OBS Browser+ to download it by referring to **Method 1: Using OBS Browser+**.
3. In the displayed dialog box, click **OK** to download the log whose status is **Preparation completed**. If you click **Cancel**, the system does not download the log.

   The download link is valid for 5 minutes. After the download link expires, a message is displayed indicating that the download link has expired. You can close the window and repeat the procedure **Step 6** to try to download a log again.

   NOTE

   After downloading slow query logs to a local PC, you can use SSMS to connect to the local database and run the following SQL statement to view the slow query log details:

   select * from ::fn_trace_gettable('*XXX*', default)

   In the preceding command, *XXX* indicates the local path for storing slow query logs.

----End

4.20 DBA Assistant

4.20.1 Function Overview

DBA Assistant provides you with a range of database O&M functions, making it easy to diagnose database problems, locate faults, analyze and optimize database performance. The functions include Dashboard, Sessions, Performance, Storage Analysis, Locks & Transactions, Slow Query Log, SQL Explorer, Concurrency Control, Auto Flow Control, Daily Reports, and Anomaly Snapshots.

NOTE

To use DBA Assistant, **submit a service ticket** to apply for required permissions.

Sessions

The **Sessions** page displays slow sessions, active sessions, and total sessions. You can quickly filter slow sessions or active sessions by user, host IP address, or database name. **Kill Session** can be used for urgent instance recovery to ensure database availability. For details, see **Sessions**.

Storage Analysis

Storage occupied by data and logs and historical changes of storage usage are important for database performance. The **Storage Analysis** page displays storage overview and disk space distribution of your instance. In addition, DBA Assistant can estimate the available days of your storage based on historical data and intelligent algorithms, so that you can scale up storage in a timely manner. **Overview**, **Abnormal Tables**, **Top 20 Databases**, and **Top 20 Tables** are also available on this page. For details, see **Storage Analysis**.

Real-Time Top SQL

The **Real-Time Top SQL** page displays top 5, top 10, and top 15 SQL statements by resource overhead based on real-time SQL data analysis, helping you quickly locate exception causes. For details, see **Real-Time Top SQL**.

Slow Query Log

The **Slow Query Log** page displays slow queries within a specified time period. You can view top 5 slow query logs by user or client IP address, sort statistics, and identify sources of slow SQL statements. For details, see **Slow Query Log**.

4.20.2 Sessions

Scenarios

You can view current session statistics of your instance, identify abnormal sessions, and kill the sessions.

Constraints

To use this function, **submit a service ticket** to request required permissions.

When the instance load is high, the session statistics cannot be obtained due to system traffic limiting.

Killing a session may cause the application to disconnect from the instance. Your application should be able to reconnect to the instance.

Procedure

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

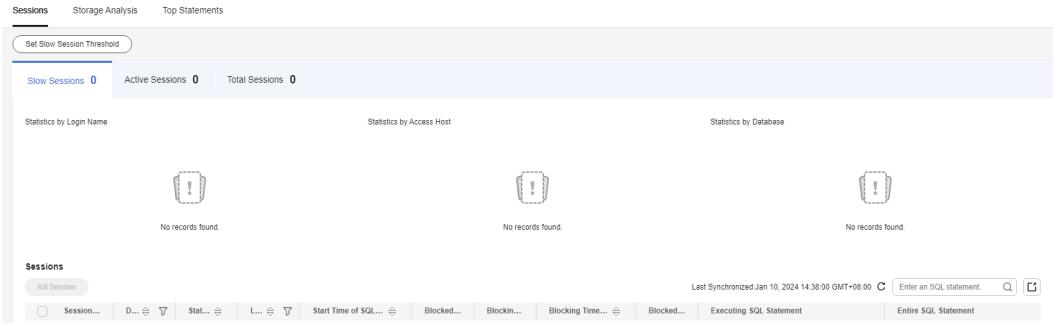
Step 2 Click  in the upper left corner and select a region.

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 Click the **Sessions** tab to view current session statistics by login name, access host, and database.



Step 7 In the session list, select the abnormal session you want to end and click **Kill Session** to recover the database.

----End

4.20.3 Storage Analysis

Scenarios

RDS for SQL Server provides space monitoring and analysis by instance, database, and even table, helping you quickly learn about space information and identify space problems.

Constraints

To use the storage analysis function, **submit a service ticket** to request required permissions.

Overview

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 On the **Storage Analysis** tab page, view storage usage. If your storage is insufficient, scale it up.

Figure 4-101 Overview

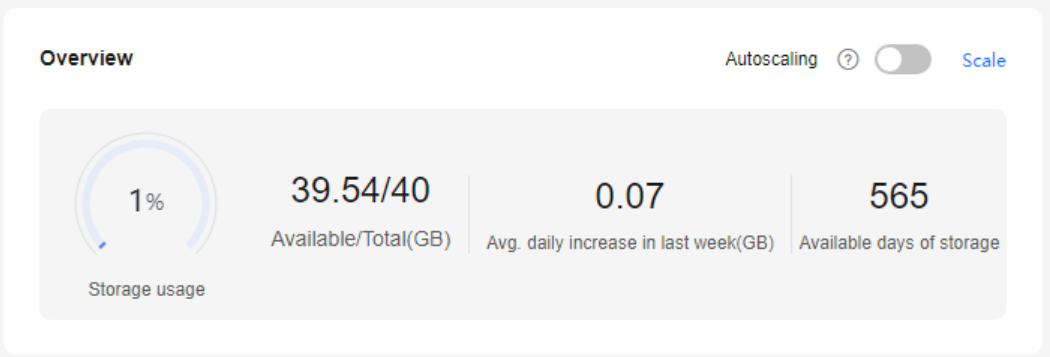


Table 4-21 Parameter description

Parameter	Description
Storage usage	Used storage space of the DB instance.
Total	Total storage space of the DB instance.
Available	Available storage space of the DB instance.
Avg. daily increase in last week(GB)	Average daily increase in storage usage in the last seven days.
Available days of storage	Estimated number of days that the remaining storage space can be used.

----End

Disk Space Distribution

You can view the distribution and changes of the storage space.

Figure 4-102 Disk space distribution

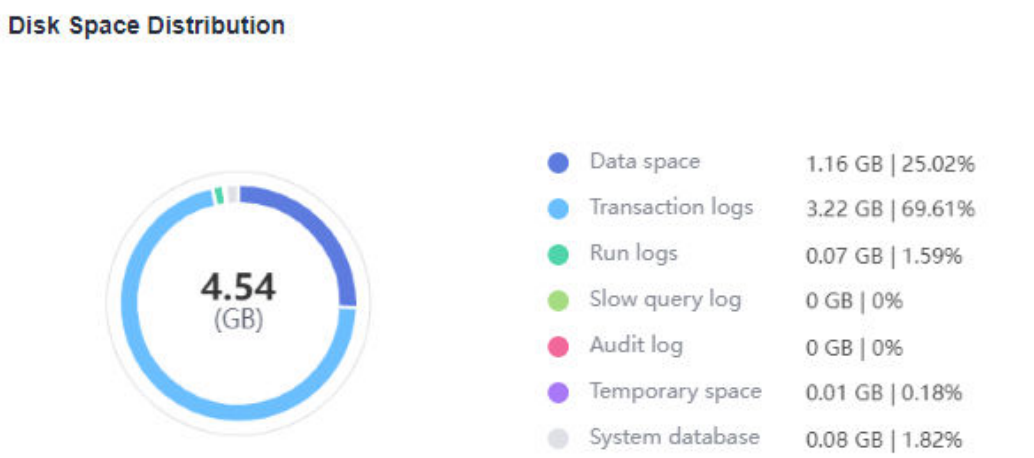


Table 4-22 Disk space distribution parameters

Parameter	Description
Data space	Total space occupied by data files.
Transaction logs	Total space occupied by transaction logs.
Run logs	Total space occupied by run logs.
Slow query log	Total space occupied by slow query logs.
Audit log	Total space occupied by audit logs.
Temporary space	Total space of the **tempdb** database.
System database	Total space of system database **msdb**.

Top 20 Databases

You can view details about the top 20 databases by physical file size, including file information.

Table 4-23 Database list parameters

Parameter	Description
Database	Database name.
Status	Database status.

Parameter	Description
Total(MB)	Total space of the database, in MB.
Used(MB)	Used space of the database, in MB.
Available(MB)	Available space of the database, in MB.
Used by Logs(MB)	Space used by transaction logs in the database, in MB.
Available to Logs(MB)	Space available to transaction logs in the database, in MB.

- You can click **View Chart** in the database list to view database space changes in the last 7 days, last 30 days, or a custom time period.
- You can click ^ in front of a database to expand the list of files contained in the database.

  Table 4-24 File list parameters

Parameter	Description
File Group	Name of the file group where the file is located. The file group of log files is **NULL**.
File Type	Type of the file, which can be **Data**, **Log**, or **Filestream**.
File Name	Name of the file.
Total(MB)	Total space of the file, in MB.
Used space(MB)	Used space of the file, in MB.
Available(MB)	Available space of the file, in MB.
Max. File Size(MB)	Maximum file space, in MB. The value **-1** indicates that the file space is not limited.
Automatic File Growth	Automatic growth of the file, in MB or percentage.

  In the file list, you can select one or more files and click **Shrink Files** to shrink the files. (This operation is not allowed for the **master**, **msdb**, **model**, and **rdsadmin** databases.)

Top 20 Tables

You can view details about the top 20 tables by physical file size. Tables whose names contain non-English character sets cannot be displayed.

Table 4-25 Table parameters

Parameter	Description
Table Name	Name of the table.
Reserved(MB)	Total space reserved for the table.
Data Space(MB)	Total space occupied by table data.
Index Space(MB)	Total space occupied by table indexes.
Available(MB)	Available space of the table.
Rows	Total number of rows in the table.
Indexes	Number of indexes created in the table.
Created	Time when the table is created. The format is affected by the character set of the instance.

You can click **View Chart** in the table list to view tablespace changes in the last 7 days, last 30 days, or a custom time period.

4.20.4 Real-Time Top SQL

Scenarios

Real-Time Top SQL shows the top SQL statements by resource overhead, helping you identify performance problems and optimize SQL statements.

To use this function, **submit a service ticket** to request required permissions.

NOTE

The statistics come from the data in the kernel cache after the instance was started last time. When new SQL statements are executed, the data in the cache is updated synchronously. You can refresh the top SQL statement list to view the latest data.

If your instance is rebooted, the data in the cache will be lost, and the top SQL statements will be recalculated.

Procedure

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

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

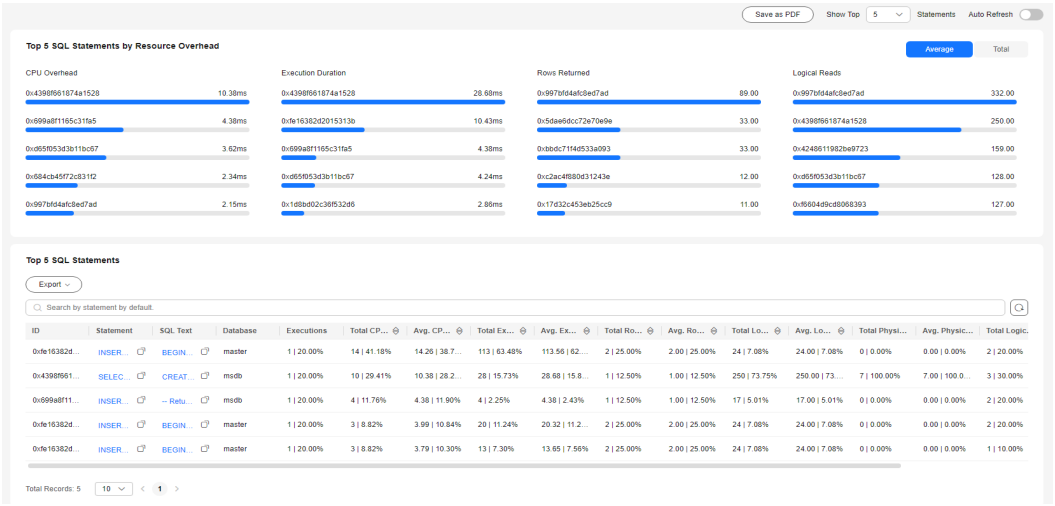
Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 6 On the **Real-Time Top SQL** tab page, view the top SQL statements by CPU overhead, execution duration, rows returned, and logical reads.

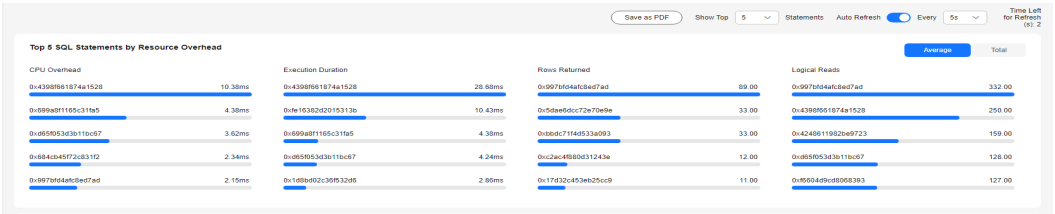
Figure 4-103 Real-Time Top SQL



- To export details about top SQL statements, click  .
- To sort parameter values in the top SQL details, click  in the table header.

Step 7 To enable auto-refresh for the **Real-Time Top SQL** page, click **Auto Refresh**. You can choose to auto-refresh the page every 5s, 10s, or 15s.

Figure 4-104 Auto Refresh



----End

Parameter Description

- Top SQL statements by resource overhead: Top SQL statements are displayed by CPU overhead, execution duration, rows returned, and logical reads. You can also check the top SQL statements by average overhead and total overhead.

Figure 4-105 Average overhead

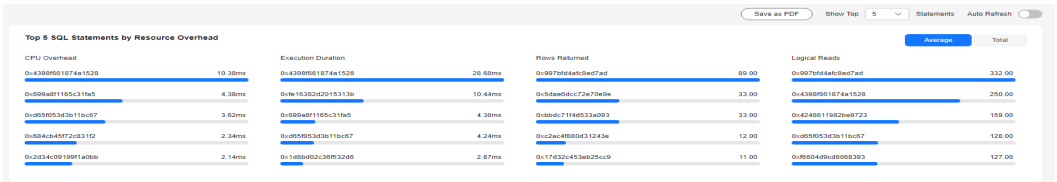


Figure 4-106 Total overhead

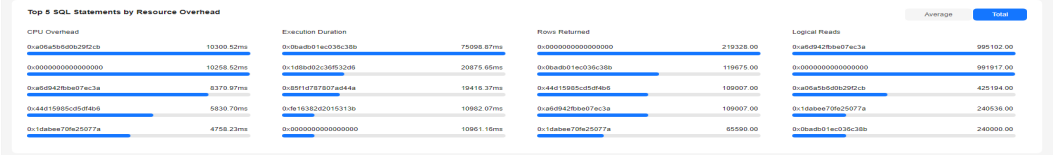


Table 4-26 Parameter description for top SQL statements by resource overhead

Parameter	Description
CPU Overhead	CPU time consumed for executing a SQL statement, in milliseconds.
Execution Duration	Execution duration of a SQL statement, in milliseconds.
Rows Returned	Number of rows returned after a SQL statement is executed.
Logical Reads	Logical reads executed by a SQL statement.

- Top SQL statement list

Figure 4-107 Top SQL statement list

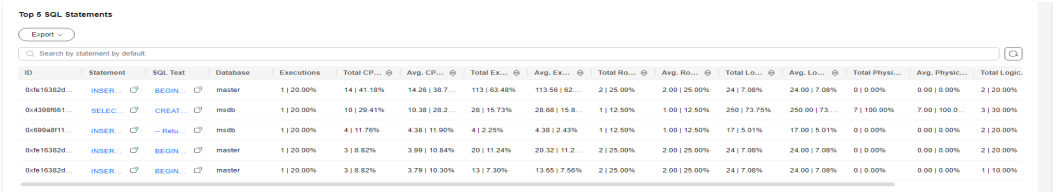


Table 4-27 Parameter description for top SQL statement list

Parameter	Description
ID	A binary hash value calculated for the query. IDs are used to identify queries with similar logic.
Statement	SQL statement. To view details, click the statement name.
SQL Text	Text of the SQL statement block. To view details, click the text name.
Database	Database where the SQL statement was executed.
Executions	Total executions of the SQL statement.
Total CPU Time	Total CPU overhead.

Parameter	Description
Average CPU Time	Average CPU overhead.
Total Execution Duration	Total execution duration.
Avg. Execution Duration	Average execution duration.
Total Rows Returned	Total number of returned rows.
Avg. Rows Returned	Average number of returned rows.
Total Logical Reads	Total logical read overhead.
Avg. Logical Reads	Average logical read overhead.

4.20.5 Slow Query Log

Scenarios

Slow Query Log displays a chart of SQL statements that are taking too long to execute and allows you to sort slow SQL statements by multiple dimensions, such as by user, client IP address, or SQL template. It helps you quickly identify bottlenecks and improve instance performance.

NOTE

If Intelligent O&M is not subscribed, records of a maximum of 1 hour can be retained.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

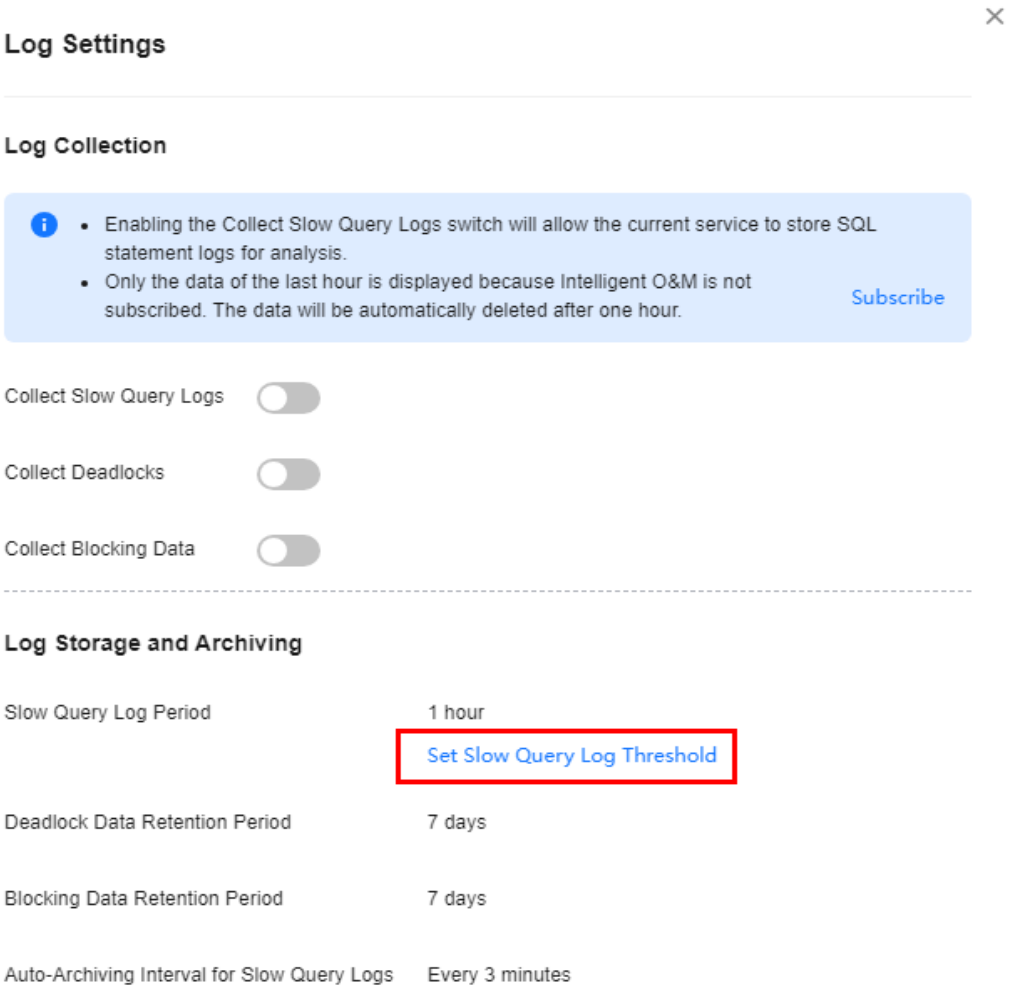
Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Slow Query Log** tab.

Step 7 Click next to the **Collect Slow Query Logs** field.

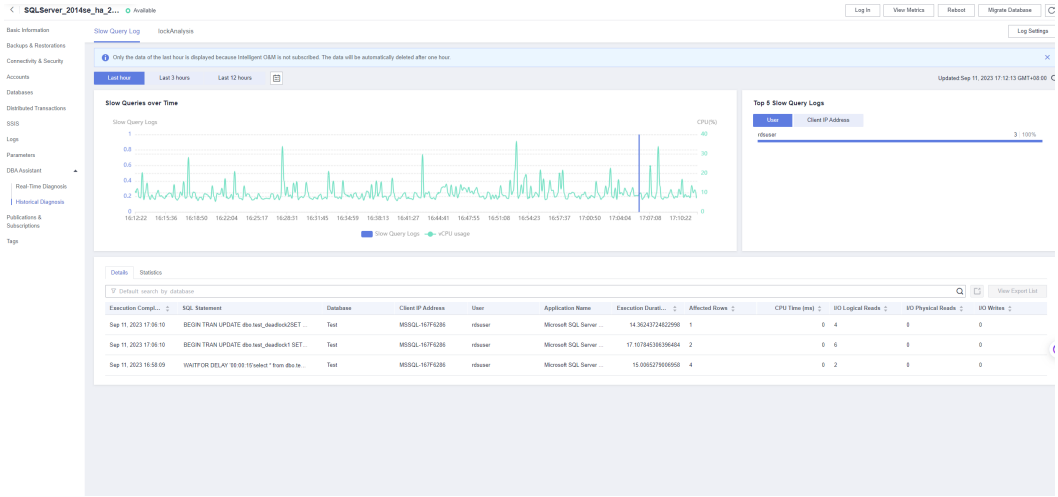
Step 8 Click **Log Settings** in the upper right corner of the page to adjust the slow query log threshold.

Figure 4-108 Setting the slow query log threshold



Step 9 View slow queries over time for the last 1 hour, last 3 hours, last 12 hours, or a custom time period (spanning no more than one day), slow log details, and template statistics.

Figure 4-109 Slow query log



- The collection of slow query logs is delayed for 1 to 3 minutes.
- You can search for slow query log details by database, client IP address, user, or execution duration.
- You can search for template information by database.
- To export slow query log information, click  .
- To view log export history, click **View Export List**.

----End

4.20.6 Deadlocks

Scenarios

RDS for SQL Server provides powerful deadlock detection. If there are two or more processes accessing the same resource at the same time, a deadlock may occur because the processes are waiting for each other to release the resource and cannot continue running. In this case, RDS for SQL Server kills one of the processes so that the other processes can complete their transactions.

To solve this problem, the **Deadlocks** page is provided. On this page, you can quickly locate various types of deadlocks in your instance. The **Details** area displays information such as transaction start time, session ID, locked resource details, and deadlock mode, helping you locate and optimize problematic SQL statements and other exceptions.

NOTE

If Intelligent O&M is not subscribed, records of a maximum of seven days can be retained.

Procedure

Step 1 Log in to the management console.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 6 Click the **Deadlocks** tab.

- **Deadlocks**

  You can view the number of deadlocks in the last day, last week, last two weeks, last month, or in a custom time interval.

- **Deadlocks over Time**

  You can view deadlocks over time within a specified period.

Table 4-28 Parameter description

Parameter	Description
Total	Total number of deadlocks.
Key Lock	Number of index-related deadlocks.
Object Lock	Number of object-related deadlocks.
Rid Lock	Number of row deadlocks.
Page Lock	Number of page deadlocks.
Compile Lock	Number of compilation deadlocks.

- **Details**

  You can view details of deadlocks within a specified period.

  To view the deadlock relationship diagram, click **Deadlock Diagram** in the **Operation** column. In the displayed dialog box, you can click **Download** to download the diagram.

  NOTE

  The downloaded deadlock diagram is an XDL file. You can open and check it using the **SQL Server Management Studio** (SSMS) client.

  Table 4-29 Parameter description

Parameter	Description
LastTranStarted	Start time of the transaction.
SPID	ID of the session that starts the transaction.
isVictim	Whether the session is killed.
Database	Name of the database where the transaction is executed.
LogUsed	Size of logs that have been generated for the session, in bytes.
LockMode	Lock mode.
WaitResourceDesc	Details of the resource that the transaction is waiting for.
ObjectOwned	Locked object.
ObjectRequested	Object that the transaction requests to lock.
WaitResource	Resource that the transaction is waiting for.
HostName	Name of the host on which the transaction is run.

Parameter	Description
LoginName	Username of the account that is used to run the transaction.
Status	Transaction status.
ClientApp	Name of the client that initiates the transaction.
SQL	SQL statement details.
Operation	You can view the deadlock diagram.

----End

4.21 Publications and Subscriptions

4.21.1 Creating a Publication

What Is Publication and Subscription?

RDS for SQL Server provides publications and subscriptions. This function uses the replication technology to split data reads and writes as well as synchronize data between cloud databases and between cloud databases and on-premises databases.

Scenarios

To synchronize data from your instance to another one, you can use your instance as the publisher instance, configure a distributor for it, create a publication, and then add a subscriber for the created publication.

Figure 4-110 Topology

Constraints

- RDS for SQL Server does not support cross-region publications or subscriptions.
- Only one distributor can be configured for an instance. All publications of the instance use this distributor. Deleting a distributor will also delete the publications using this distributor.

- RDS for SQL Server Web Edition instances cannot be used as distributors or publisher instances, but can be used as subscribers.
- When you create a publication, the database name and publication name must be different from those of existing publications.
- RDS for SQL Server supports only transactional publications.
- If you add a subscription server other than RDS, the account used for logging in to the server must have the **sysadmin** permission.
- If you add an RDS subscription server, you can select a maximum of 10 destination databases at a time.
- The floating IP address and port number of an instance with a publication or subscription created cannot be changed.
- Chinese characters are not allowed in table names or field names on the publisher or subscriber.
- If the primary node of the distributor fails, the publication and subscription link cannot be restored through primary/standby switchover.

Configuring a Distributor

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

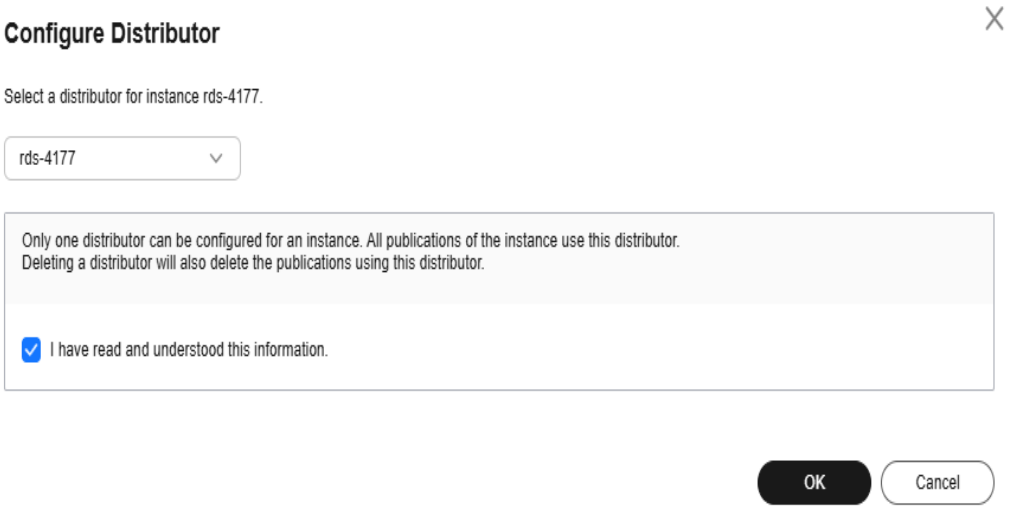
Step 4 On the **Instances** page, click the DB instance name.

Step 5 In the navigation pane, choose **Publications & Subscriptions**.

Step 6 On the **Publications** page, click **Configure Distributor**.

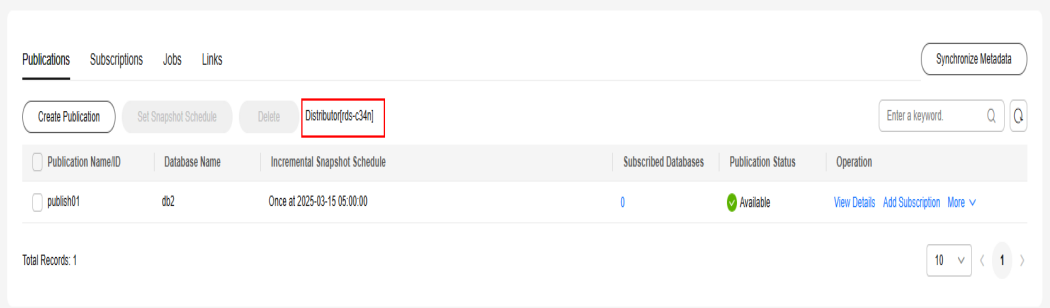
Step 7 In the displayed dialog box, select the current instance or another instance as the distributor, select **I have read and understood this information**, and click **OK**.

Figure 4-111 Configuring a distributor



Step 8 View the configured distributor.

Figure 4-112 Viewing a distributor



----End

Creating a Publication

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **Publications & Subscriptions**.

Step 3 On the **Publications** page, click **Create Publication**.

Step 4 On the displayed page, configure parameters and click **OK**.

- Enter a publication name, select a publication database, and specify publication objects.
- To set a filter for publishing tables/fields, click **Set Filter**.
- To set project properties, click **Set Project Properties**.
- You can customize an incremental snapshot schedule by day, week, or month to generate incremental snapshots on the distributor.

Figure 4-113 Creating a publication

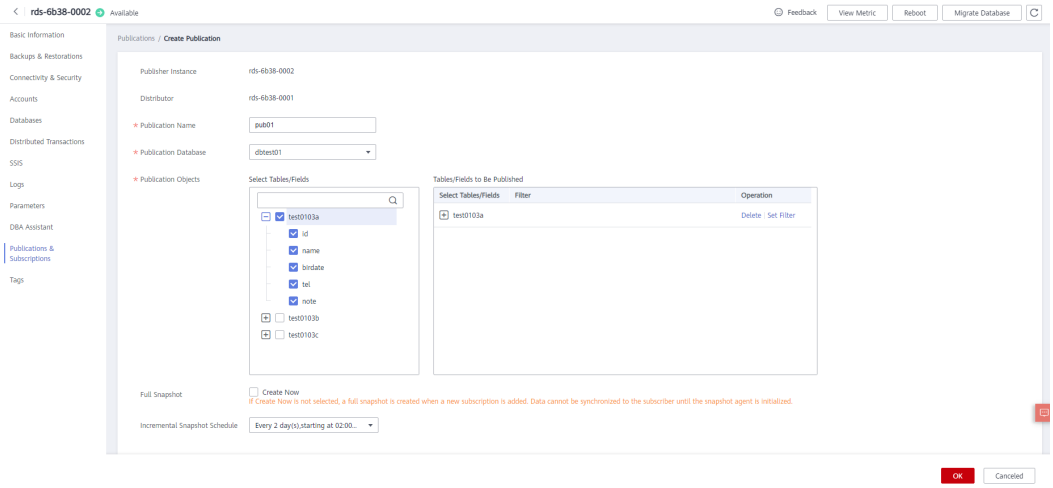
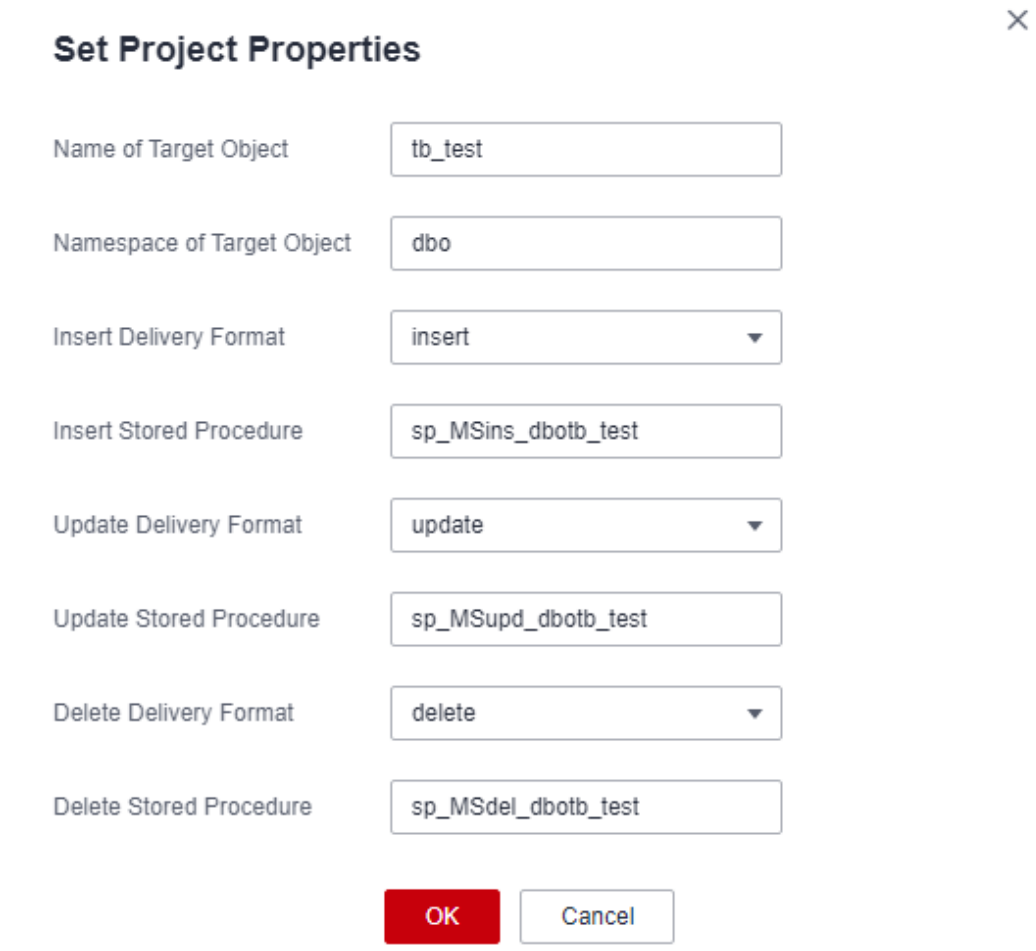
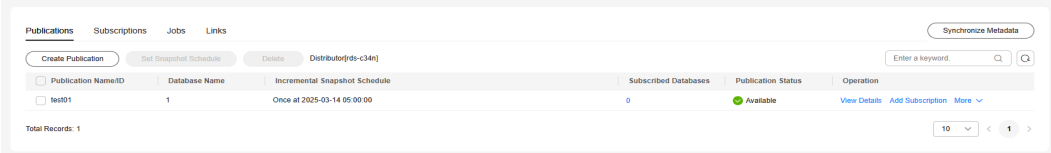


Figure 4-114 Setting project properties



Step 5 View the created publication.

Figure 4-115 Viewing a publication



- To add a subscription for the publication, follow the instructions in **Adding a Subscriber**.
- To view **Latency for Data Changes** and **Transactions**, click **Monitor**.
- To modify tables or fields to be published and the incremental snapshot schedule, choose **More** > **Modify Publication**.
- To delete the publication, choose **More** > **Delete**.
- To regenerate an incremental snapshot on the distributor, choose **More** > **Regenerate**.

----End

Adding a Subscriber

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **Publications & Subscriptions**.

Step 3 On the **Publications** page, locate the created publication and click **Add Subscription** in the **Operation** column.

Step 4 Click **Add Subscriber**.

Step 5 On the displayed page, configure parameters and click **OK**.

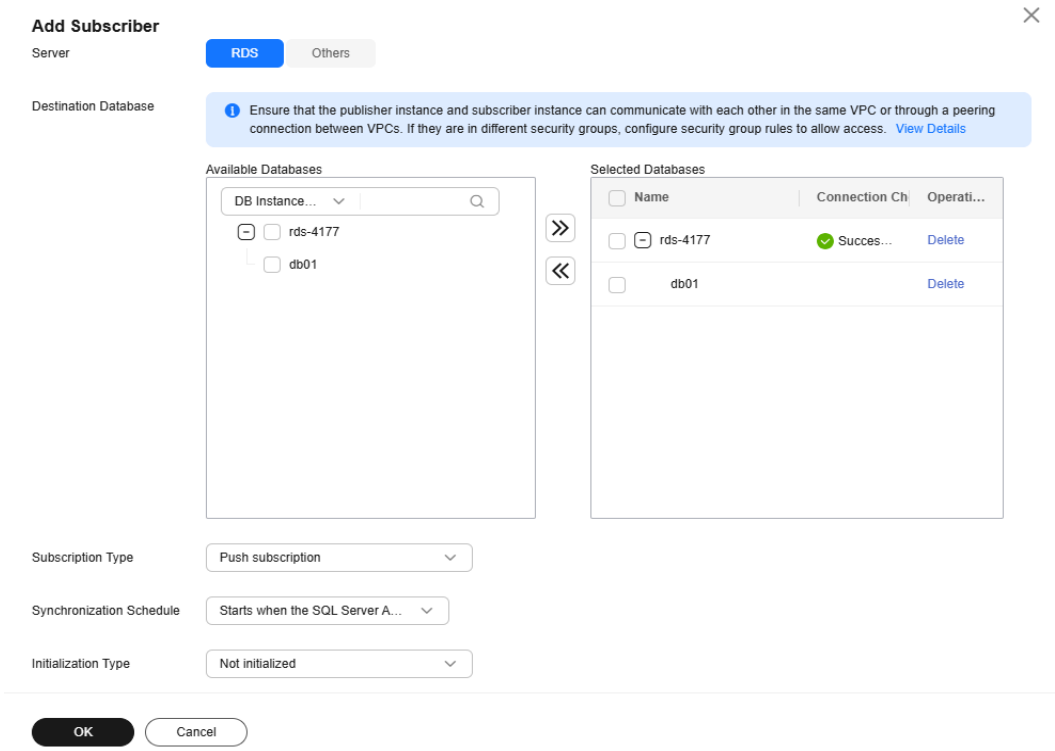
For details about the publisher and subscriber types supported by RDS for SQL Server, see **Compatibility Between Publishers and Subscribers**.

If you select **RDS** for **Server**:

- Select one or more RDS for SQL Server subscriber instances and destination databases, and click  to synchronize the selected databases to the right box.

  Ensure that the publisher instance and subscriber instance can communicate with each other in the same VPC or through a peering connection between VPCs. If they are in different security groups, configure security group rules to allow access.
- Select **Push subscription** for **Subscription Type**.
- Select a synchronization schedule for data subscription. You can customize a schedule by day, week, or month.

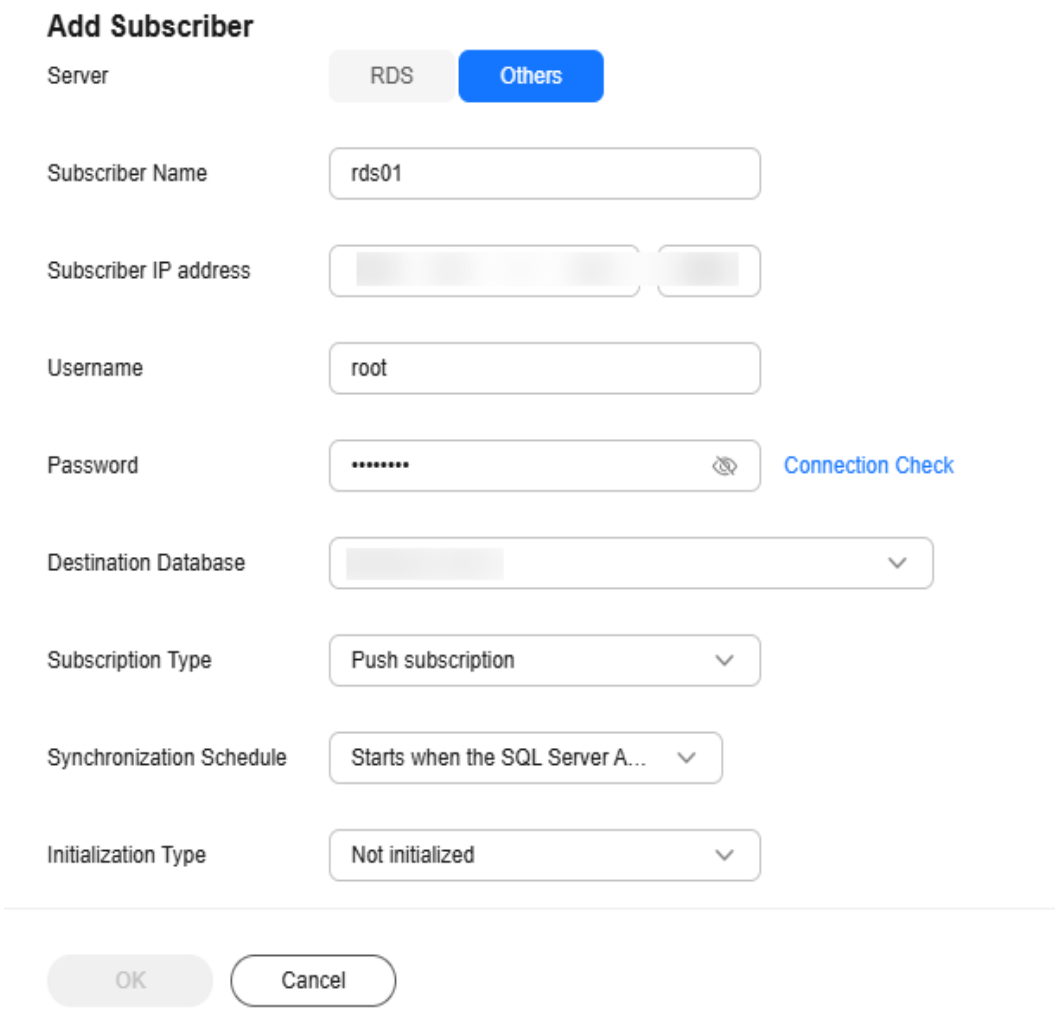
Figure 4-116 Adding an RDS subscriber



If you select **Others** for **Server**:

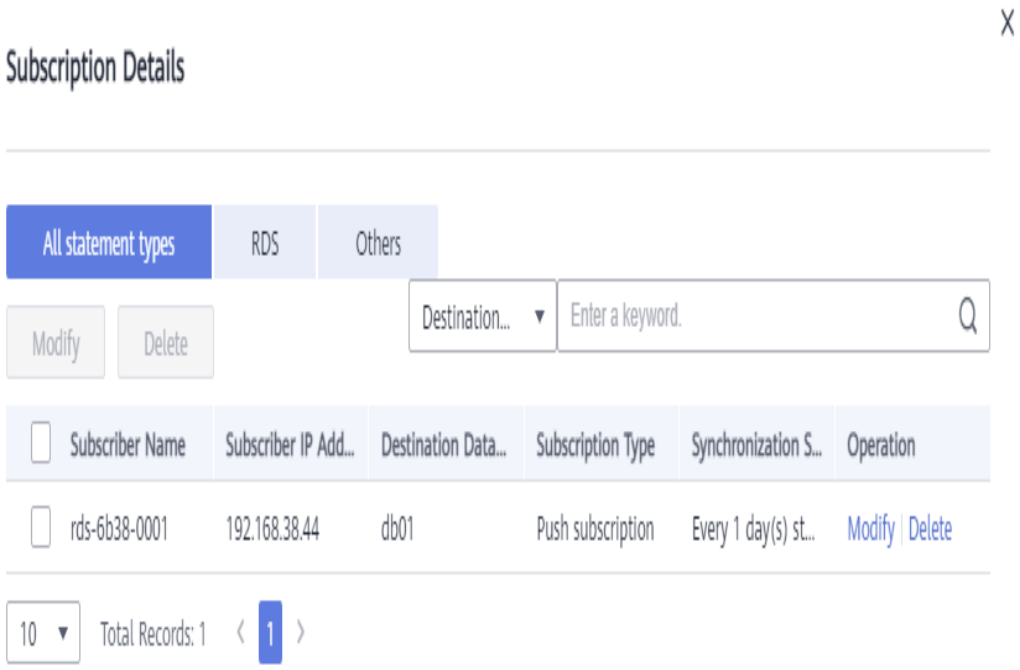
- Enter the subscriber name, subscriber IP address, port number, login username, and password, and specify destination databases.
- Select **Push subscription** for **Subscription Type**.
- Select a synchronization schedule for data subscription. You can customize a schedule by day, week, or month.

Figure 4-117 Adding other subscribers



Step 6 Locate the created publication, click the number in the **Subscribed Databases** column to view the subscription details.

Figure 4-118 Subscription details



----End

4.21.2 Creating a Subscription

Scenarios

You can add **a created publication** for your RDS for SQL Server instance to synchronize data from the instance to the subscriber through a distributor.

Constraints

- RDS for SQL Server does not support cross-region publications or subscriptions.
- You can add multiple publications for one instance.
- RDS for SQL Server Web Edition instances cannot be used as distributors or publisher instances, but can be used as subscribers.
- Only one publication can be added to a database.
- The floating IP address and port number of an instance with a publication or subscription created cannot be changed.

Compatibility Between Publishers and Subscribers

Table 4-30 Compatibility between publishers and subscribers

Publisher (Except Web Edition)	Distributor (Except Web Edition)	Subscriber
SQL Server 2022	SQL Server 2022	All RDS for SQL Server versions
RDS for SQL Server 2019	SQL Server 2022 RDS for SQL Server 2019	
RDS for SQL Server 2017	SQL Server 2022 RDS for SQL Server 2019 RDS for SQL Server 2017	
RDS for SQL Server 2016	SQL Server 2022 RDS for SQL Server 2019 RDS for SQL Server 2017 RDS for SQL Server 2016	
RDS for SQL Server 2014	SQL Server 2022 RDS for SQL Server 2019 RDS for SQL Server 2017 RDS for SQL Server 2016 RDS for SQL Server 2014	
RDS for SQL Server 2012	SQL Server 2022 RDS for SQL Server 2019 RDS for SQL Server 2017 RDS for SQL Server 2016 RDS for SQL Server 2014 RDS for SQL Server 2012	
RDS for SQL Server 2008 R2	SQL Server 2022 RDS for SQL Server 2019 RDS for SQL Server 2017 RDS for SQL Server 2016 RDS for SQL Server 2014 RDS for SQL Server 2012 RDS for SQL Server 2008 R2	

Creating a Subscription

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **Publications & Subscriptions**.

Step 3 On the **Subscriptions** page, click **Create Subscription**.

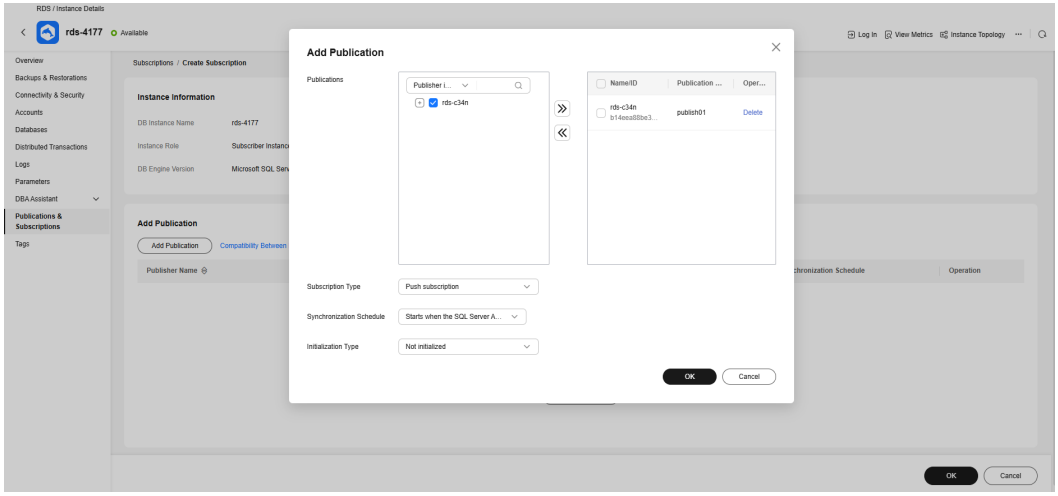
Step 4 Click **Add Publication**.

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

- Select distributors and publications, and click  to synchronize the selected publications to the right box.

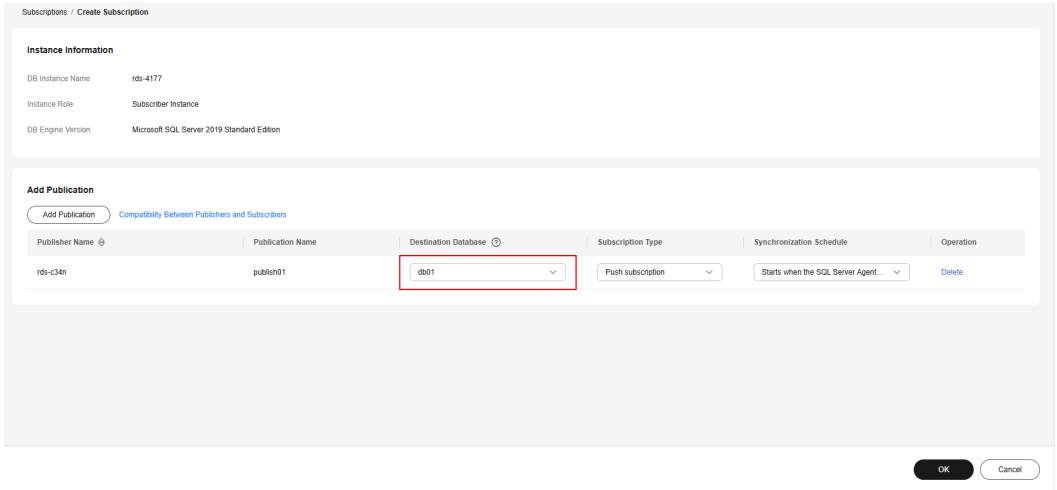
  For details about the publisher and subscriber types supported by RDS for SQL Server, see **Compatibility Between Publishers and Subscribers**.
- Select **Push subscription** for **Subscription Type**.
- Select a synchronization schedule for data subscription. You can customize a schedule by day, week, or month.

Figure 4-119 Adding a publication



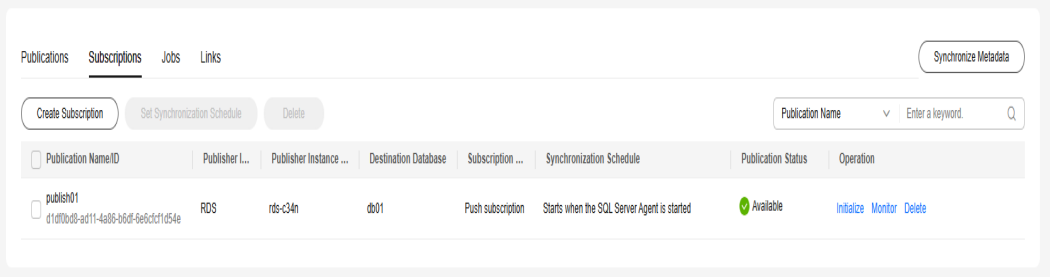
Step 6 Select a destination database that the publication is to be subscribed, and click **OK**.

Figure 4-120 Selecting a destination database



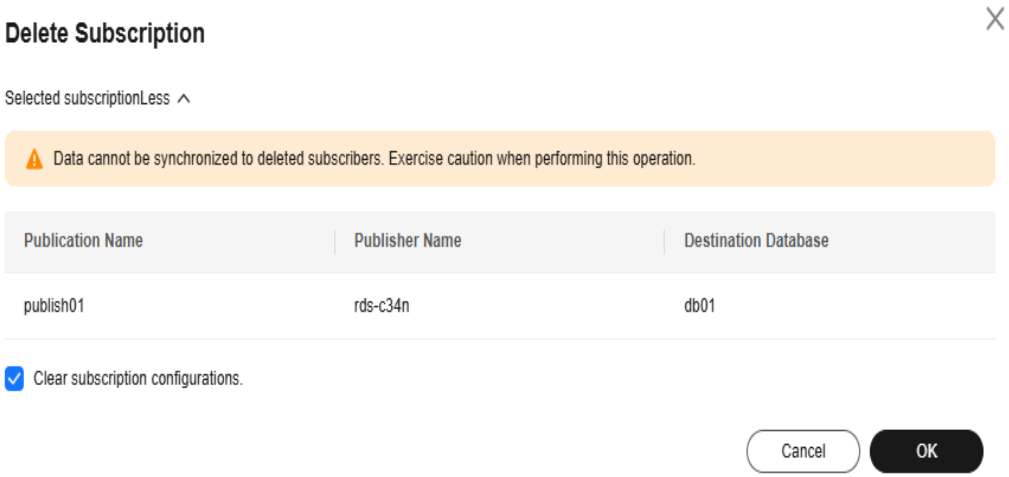
Step 7 View the created subscription.

Figure 4-121 Subscription



To delete the subscription, click **Delete**. When a subscription is deleted, its configurations can be deleted synchronously.

Figure 4-122 Deleting a subscription



----End

4.21.3 Checking Jobs and Links

Scenarios

RDS for SQL Server provides job monitoring and link monitoring. Job monitoring allows you to view publication and subscription jobs and their execution history. You can also modify profiles and restart jobs. Link monitoring allows you to check or download information about the publisher instance, subscriber instance, and distributor.

Constraints

Jobs can only be displayed if the current instance is used as a distributor.

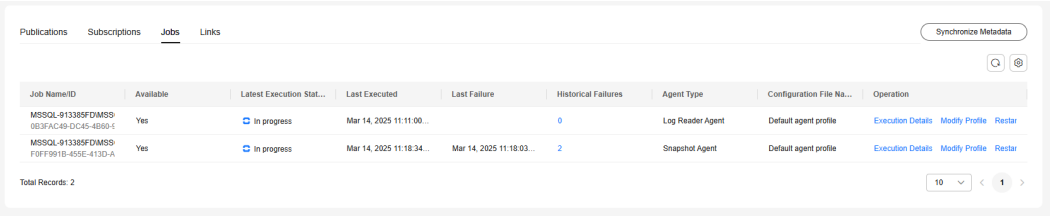
Checking Jobs

Step 1 On the **Instances** page, click the DB instance name.

Step 2 In the navigation pane, choose **Publications & Subscriptions**.

Step 3 Click the **Jobs** tab to check the jobs.

Figure 4-123 Jobs



- If a job fails to be executed, click the number in the **Historical Failures** column or **Execution Details** in the **Operation** column to check the execution process in the displayed dialog box on the right and locate the failure cause.
- To rectify an error, click **Modify Profile** and select a new profile. For details about profiles, see **Replication Agent Profiles**.
- To re-execute a job, click **Restart**. The ongoing job will be interrupted.

----End

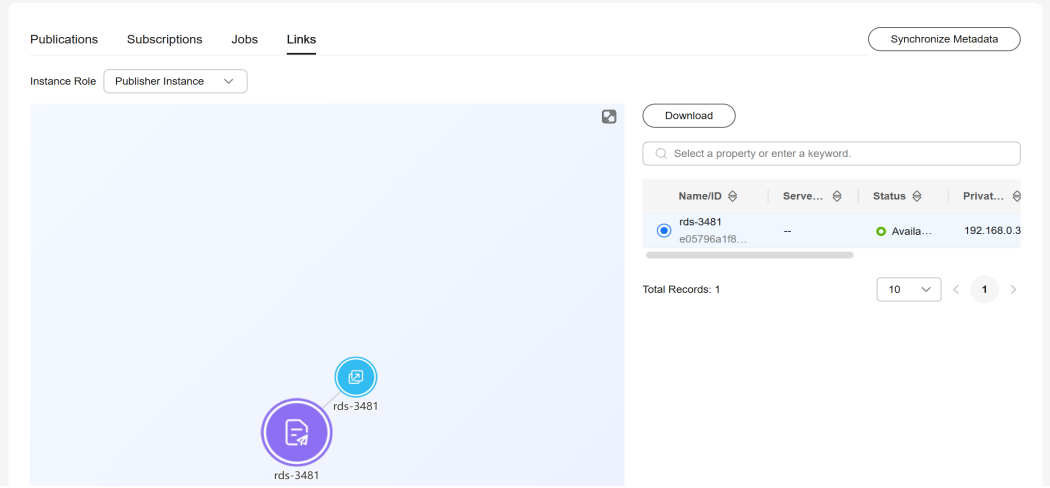
Checking Links

Step 1 On the **Instances** page, click the DB instance name to go to the **Overview** page.

Step 2 In the navigation pane, choose **Publications & Subscriptions**.

Step 3 Click the **Links** tab to check or download information about the publisher instance, subscriber instance, and distributor.

Figure 4-124 Links



----End

4.22 Task Center

4.22.1 Viewing a Task

You can view the progress and results of scheduled and instant tasks on the **Task Center** page.

Supported Tasks

Table 4-31 Supported tasks

Task Type	Category	Task Name
Instant tasks	Instance creation	Creating a Microsoft SQL Server DB instance, creating a Microsoft SQL Server read replica
	Instance lifecycle	Rebooting a Microsoft SQL Server DB instance, stopping a Microsoft SQL Server DB instance, starting a Microsoft SQL Server DB instance, deleting a Microsoft SQL Server DB instance
	Instance modifications	Scaling a Microsoft SQL Server DB instance, switching Microsoft SQL Server primary/standby DB instances, cloning a Microsoft SQL Server DB instance, changing the Microsoft SQL Server instance type from single to primary/standby, changing a Microsoft SQL Server instance class, changing a Microsoft SQL Server storage type, migrating a standby Microsoft SQL Server DB instance, changing a Microsoft SQL Server character set
	Connection management	Creating a public domain name for Microsoft SQL Server, changing a public domain name for Microsoft SQL Server, creating a private domain name for Microsoft SQL Server, changing a private domain name for Microsoft SQL Server, binding an EIP to a Microsoft SQL Server instance, unbinding an EIP from a Microsoft SQL Server instance, updating an SSL certificate for Microsoft SQL Server
	Backup and restoration	Restoring to a new Microsoft SQL Server DB instance, restoring to an existing Microsoft SQL Server DB instance
	FileStream	Enabling FileStream for Microsoft SQL Server

Task Type	Category	Task Name
	Security and encryption	Enabling TDE for Microsoft SQL Server, rotating TDE certificate for Microsoft SQL Server
Scheduled tasks	Instance lifecycle	Starting Microsoft SQL Server instance, upgrading Microsoft SQL Server instance system

Viewing an Instant Task

Step 1 Log in to the management console.

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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 view the progress and status of tasks in a specific period. The default period is seven days.

  The task list can only show up to 30 days of past tasks.
- 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

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

4.22.2 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed **Instant Tasks** page, locate the 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:

- 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.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the 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 **OK** to cancel the task. Then, click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

Step 6 Click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

You can delete the records of scheduled tasks in any of the following statuses:

- Completed
- Failed
- Canceled
- To be executed
- To be authorized

----End

4.22.3 Authorizing a Task

You can authorize tasks on the **Task Center** page so that they can be executed as scheduled within the maintenance window.

Currently, minor version upgrades of Microsoft SQL Server DB instances require authorization.

Procedure

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Task Center** in the navigation pane on the left. On the **Scheduled Tasks** page, locate the target task to be authorized and check whether the task status is **To be authorized**.

- If yes, go to **Step 5**.
- If no, no further action is required.

Step 5 Click **Authorize** in the **Operation** column. In the displayed dialog box, select a scheduled date and the check box before the authorization notice, and click **Yes**.

Step 6 After the task is authorized successfully, it will be executed as scheduled within the maintenance window.

NOTE

The DB instance will be rebooted during the task execution, which causes service interruptions. To prevent service interruptions, you are advised to set the maintenance window to off-peak hours. For details about how to change the maintenance window, see **Changing the Maintenance Window**.

----End

4.23 Billing Management

4.23.1 Unsubscribing from a Yearly/Monthly Instance

Scenarios

To delete a DB instance billed on the yearly/monthly basis, you need to unsubscribe the order. You can unsubscribe a single instance order by referring to **Unsubscribing a Single DB Instance (Method 1)** and **Unsubscribing a Single DB Instance (Method 2)** or unsubscribe multiple instance orders at a time by referring to **Unsubscribing DB Instances in Batches**. For unsubscription fees, see **Unsubscription Rules**.

If you unsubscribe from a DB instance, its read replicas (if any) will also be unsubscribed.

To release DB instances or read replicas billed on a pay-per-use basis, you need to locate the target DB instances or read replicas and click **Delete** on the **Instances** page. For details, see **Deleting a Pay-per-Use DB Instance or Read Replica**.

Constraints

- A DB instance cannot be unsubscribed when any operations are being performed on it. It can be unsubscribed only after the operations are complete.
- If a backup of a DB instance is being restored, the instance cannot be unsubscribed.

Unsubscribing a Single DB Instance (Method 1)

Unsubscribe a yearly/monthly DB instance or read replica on the **Instances** page.

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, locate the target DB instance or read replica and choose **More** > **Unsubscribe** in the **Operation** column.

Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription results. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing a Single DB Instance (Method 2)

Unsubscribe a yearly/monthly DB instance or read replica on the **Billing Center** page.

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

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

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

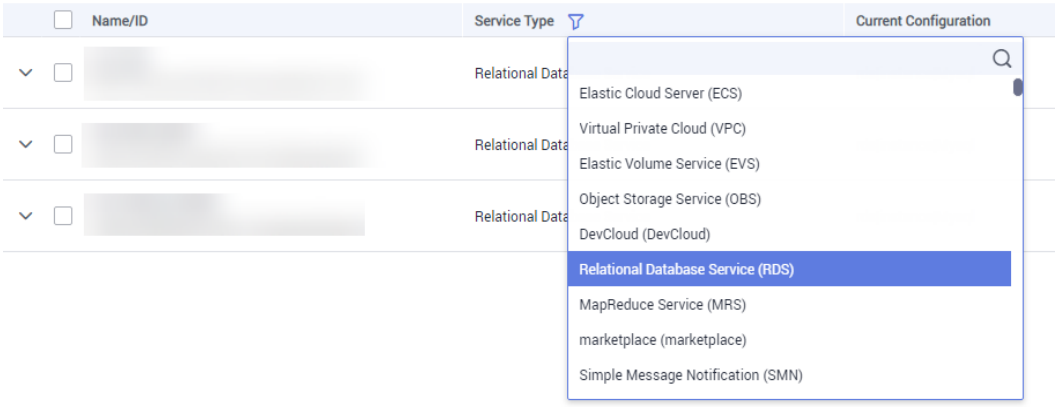
Step 4 In the upper right corner, click **Billing & Costs**.

Step 5 In the navigation pane, choose **Orders** > **Unsubscriptions**.

Step 6 On the displayed page, select the order to be unsubscribed and click **Unsubscribe** in the **Operation** column.

- You can select **Relational Database Service (RDS)** in the **Service Type** column to filter all RDS orders.

Figure 4-125 Filtering all RDS orders



- Alternatively, search for target orders by name, order No., or ID in the search box.
- A maximum of 20 resources can be unsubscribed at a time.

Step 7 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 8 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 9 View the unsubscription result. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing DB Instances in Batches

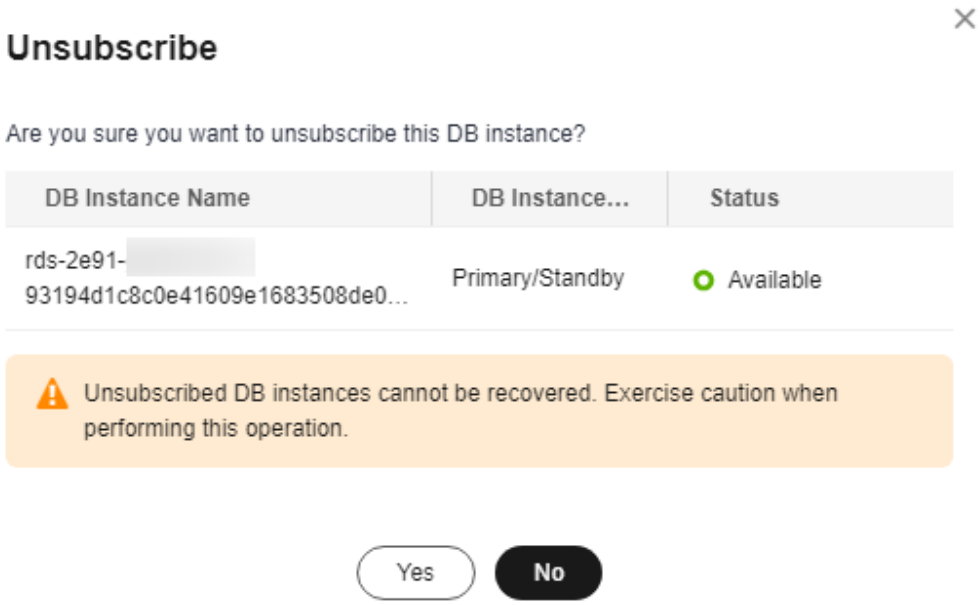
Step 1 **Log in to the management console**.

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, select the target DB instances to be unsubscribed and click **Unsubscribe** above the DB instance list. In the displayed dialog box, click **Yes**.

Figure 4-126 Unsubscribing yearly/monthly orders in batches



Step 5 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details about unsubscribing resources, see **Unsubscription Rules**.

Step 6 In the displayed dialog box, click **Yes**.

NOTICE

1. After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
2. If you want to retain data, complete a manual backup before submitting the unsubscription request.

Step 7 View the unsubscription results. After the DB instance order is successfully unsubscribed, the DB instance and read replicas are no longer displayed in the instance list on the **Instances** page.

----End

4.24 Enabling or Disabling FileStream

Scenarios

FileStream of RDS for SQL Server stores unstructured data, such as documents and images, in file systems, effectively improving the database performance.

Constraints

- Enabling or disabling FileStream for a DB instance will reboot the DB instance. A DB instance reboot takes about 3 to 5 minutes. Services are unavailable when a DB instance is rebooting.
- This feature is supported only for single instances and 2017 Enterprise Edition cluster instances.
- FileStream cannot be disabled for a DB instance when databases of the DB instance have been created with this feature enabled. If you need to disable this feature, delete the databases first.
- Databases of a DB instance with FileStream enabled cannot be created using v3 APIs. For details about how to create a database using APIs, see **Creating a Database**.
- After you enable FileStream, the DB instance type cannot be changed from single to primary/standby.
- The backups generated after you enable FileStream can only be restored to an existing instance (not the original one) or a new instance, and the instance must be a single instance or 2017 Enterprise Edition cluster instance.
  - If you want to restore to an existing DB instance (not the original one), ensure that FileStream has been enabled for this DB instance.
  - If you want to restore to a new DB instance, FileStream will be automatically enabled for the new DB instance.

Enabling FileStream

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name. On the displayed **Overview** page, click **Enable** under **FileStream**.

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

NOTICE

Enabling FileStream will reboot your DB instance, and services will be unavailable during the reboot. Exercise caution when performing this operation.

----End

Disabling FileStream

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

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

Step 3 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name. On the displayed **Overview** page, click **Disable** under **FileStream**.

Step 5 In the displayed dialog box, click **OK** to disable FileStream.

NOTICE

Disabling FileStream will cause your DB instance to reboot, and services will be unavailable during the reboot. Exercise caution when performing this operation.

----End

Parameters

filestream access level

Default value: **0**

Function: Use this parameter to change the FileStream access level for your RDS for SQL Server DB instance.

Value range:

- **0**: disables this function.
- **1**: enables this function for Transact-SQL access.
- **2**: enables this function for all streaming access.

  To change the parameter value, see **Modifying RDS for SQL Server Instance Parameters**.

Impact:

- If FileStream is enabled, set this parameter to **1** or **2**.
- If FileStream is disabled, you can set this parameter only to **0**.

4.25 CLR Integration

Scenarios

The common language runtime (CLR) is the core of .NET Framework and provides the execution environment for all .NET Framework code. Code that runs within the CLR is referred to as managed code. The CLR provides various functions and services required for program execution, including just-in-time (JIT) compilation, allocating and managing memory, enforcing type safety, exception handling, thread management, and security.

SQL CLR is a new function of SQL Server 2005. It injects the CLR service of .NET Framework into SQL Server, so that some database objects of SQL Server can be developed using the .NET Framework programming language (currently, only VB.NET and C# are supported). These database objects include stored procedures,

triggers, user-defined functions, user-defined types, and user-defined aggregates. To execute the CLR code, you need to enable CLR integration first.

For more information about CLR integration, see **Common Language Runtime (CLR) Integration Programming Concepts**.

For more information about CLR integration security, see **CLR Integration Security**.

Prerequisites

RDS for SQL Server can deploy SAFE assemblies only.

Enabling CLR

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

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

Step 3 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, enter **clr enabled** and **clr strict security** in the search box in the upper right corner.

NOTE

- **clr enabled** determines whether the CLR integration is enabled.
- **clr strict security** is a specific parameter to RDS for SQL Server 2017. This parameter explains the SAFE, EXTERNAL ACCESS, and UNSAFE permissions of RDS for SQL Server. The value **1** causes the DB engine to ignore the PERMISSION_SET information on the assemblies, and always interprets them as UNSAFE. For more information, see **CLR strict security** at the Microsoft site.

Step 6 Set the **clr enabled** value.

Set **clr enabled** to **1** and click **Save**. In the displayed dialog box, click **Yes** to enable the CLR function.

NOTE

- **clr enabled**: The value **1** indicates that the CLR function is enabled. The value **0** indicates that the CLR function is disabled. Only **clr enabled** needs to be set to enable the CLR function.
- **clr strict security**: The default value is **1** and no configuration is required.

Step 7 On the **Change History** tab, check that the value of **clr enabled** has been changed to **1**.

----End

Creating a SAFE CLR Assembly

The following factors should be considered when you design assemblies:

- Packaging assemblies
- Management assembly security
- Restrictions on assemblies

For more information, see **Designing Assemblies**.

Example: Creating a C# CLR Assembly

RDS for SQL Server provides assemblies to make database operations simple and convenient.

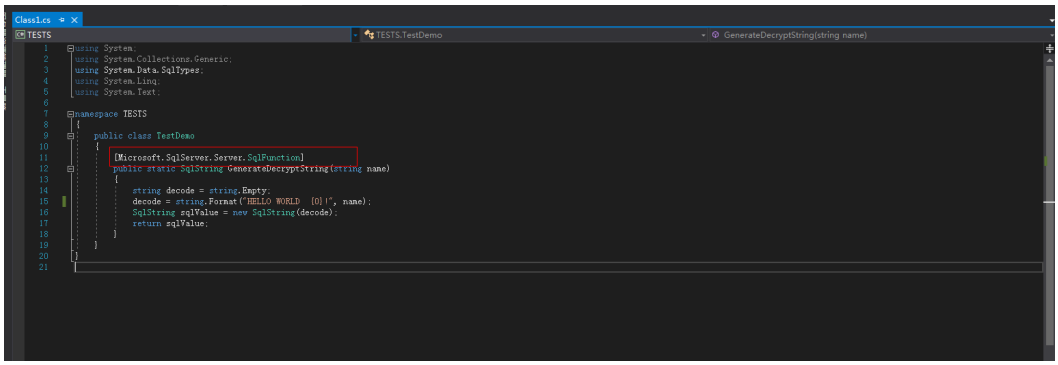
NOTE

When you restore data to a new or an existing DB instance, the **clr enabled** parameter is disabled by default. To use the CLR integration function, you need to enable **clr enabled** first.

Procedure

Step 1 Create a C# function to compile an RDS for SQL Server DLL.

Figure 4-127 C# function code

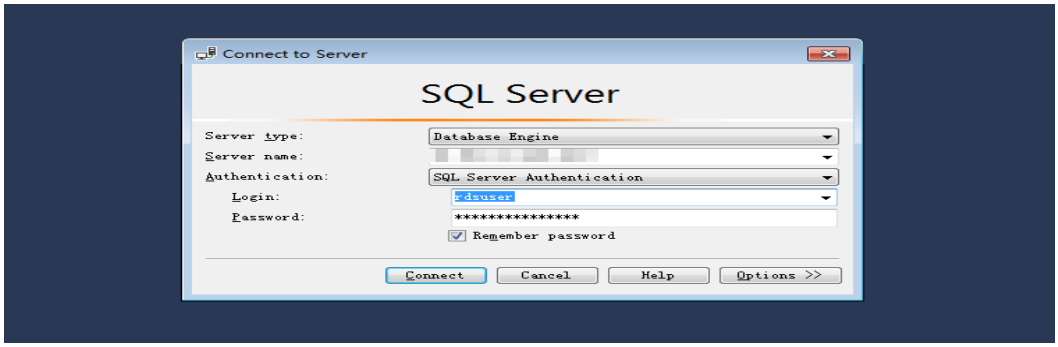


NOTICE

For more information about user-defined functions, see **CLR User-Defined Functions**.

Step 2 Use SQL Server Management Studio to connect to the database.

Figure 4-128 Connecting to the server



Step 3 Select the target database and create the corresponding assembly.

NOTE

- Only the SAFE assembly (**Permission** set is **Safe**) can be created.
- The DLL file is saved in the hexadecimal format, as shown in **Figure 4-130**.

Figure 4-129 Creating an assembly

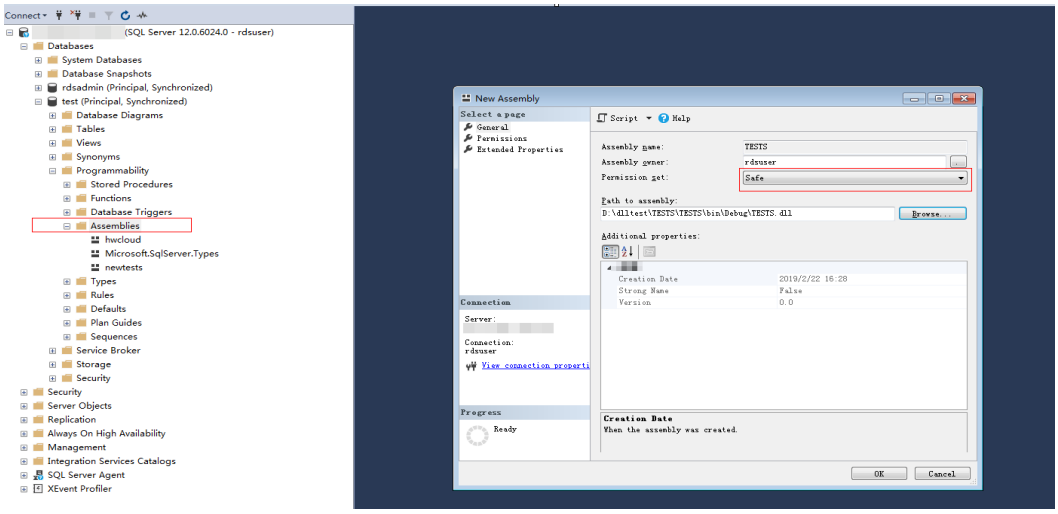
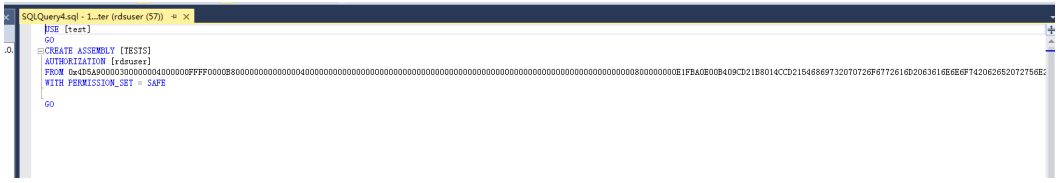


Figure 4-130 DLL file



Step 4 Execute the program. If the execution result is shown as **Figure 4-131**, the execution is successful. The TESTS assembly is added, as shown in **Figure 4-132**.

Figure 4-131 Execution result

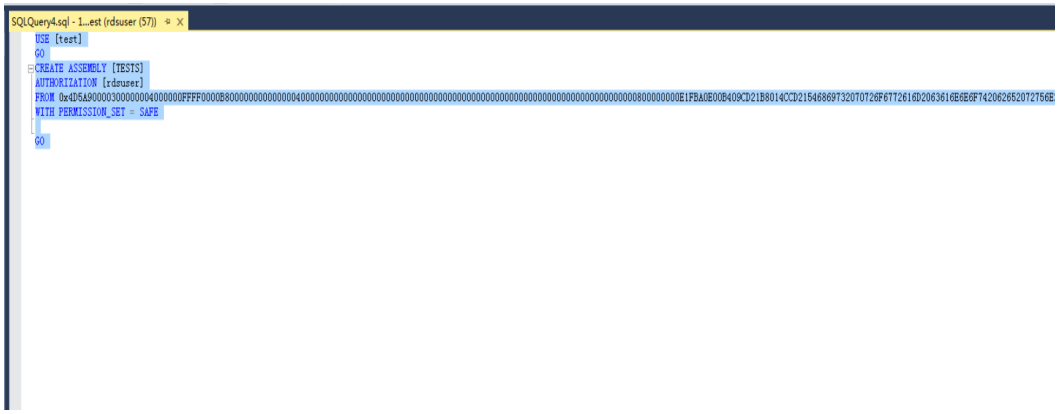
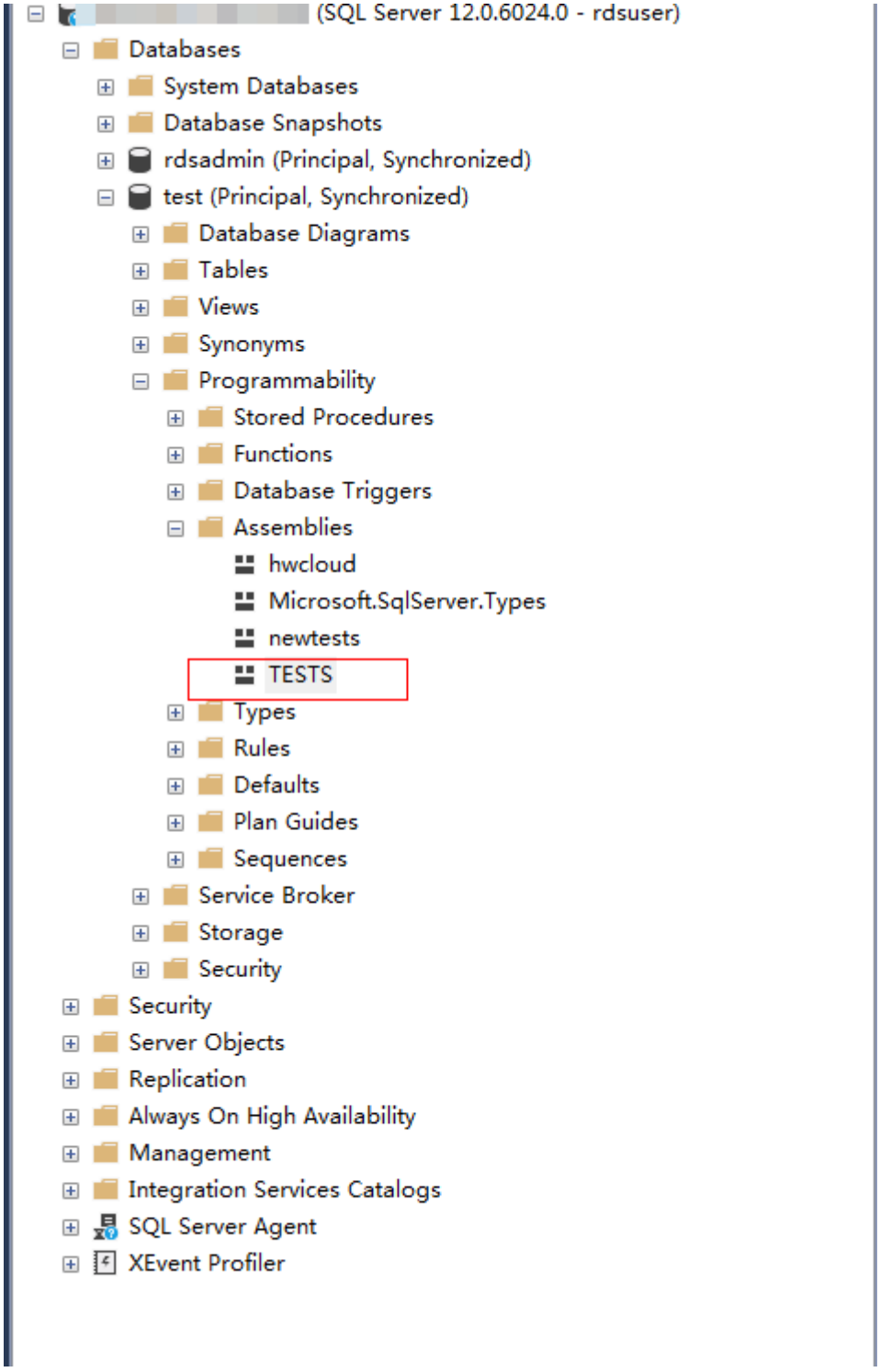


Figure 4-132 TESTS assembly



----End

4.26 Default Language Setting for RDS for SQL Server

Scenarios

The **default language** option specifies the default language for logins. To set the default language, specify the **langid** value of the language you want. The **langid** value can be obtained by querying the **sys.syslanguages** compatibility view. For introduction to sys.syslanguages (Transact-SQL), see **sys.syslanguages (Transact-SQL)**.

For more information, see **Configure the default language Server Configuration Option**.

For RDS for SQL Server DB instances, you can set the default language by modifying the **default language** parameter in a DB instance or custom parameter template.

Modifying the Instance Parameter

Step 1 **Log in to the management console**.

Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Parameters**. On the displayed page, enter **default language** in the search box.

NOTE

- The **default language** parameter specifies the default language for all newly created logins.
- The default value of **default language** is **0**, indicating that the default language is English.

Step 6 Set the **default language** value.

Set **default language** to **30** and click **Save**. In the displayed dialog box, click **Yes** to set the default language to simplified Chinese.

Step 7 On the **Change History** tab, check that the value of **default language** has been successfully changed to **30**.

NOTE

The change to **default language** takes effect immediately. You do not need to reboot the DB instance.

----End

Modifying the Parameter in a Custom Parameter Template

Step 1 Log in to the management console.

Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 5 On the displayed page, enter **default language** in the search box in the upper right corner.

Step 6 Set the **default language** value.

Set **default language** to **30** and click **Save**. In the displayed dialog box, click **Yes** to set the default language to simplified Chinese.

Step 7 On the **Change History** tab, check that the value of **default language** has been successfully changed to **30**.

Step 8 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.

Step 9 In the displayed dialog box, select one or more DB instances to which the parameter template will be applied and click **OK**.

NOTE

- A parameter template can be applied to one or more DB instances.
- After you reset the parameter template, view the status of the DB instance to which the parameter template is applied in the DB instance list. If the status is **Parameter change. Pending reboot**, a reboot is required for the reset 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.

Step 10 After the application is successful, locate the target parameter template on the **Custom Templates** page and choose **More** > **View Application Record** in the **Operation** column to view the application records.

----End

4.27 Usage of Stored Procedures

4.27.1 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 RDS for 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 instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance Through a Public Network**.

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='******';

After the account is created successfully, information similar to the following is displayed:

```
RDS_Process_Successful
```

4.27.2 Granting SSIS Permissions to a Domain Account

Scenarios

Run a stored procedure to grant SSIS permissions to a specified domain account.

Prerequisites

You have connected to an RDS for SQL Server instance. For details about how to connect to an instance, see **Connecting to an Instance**.

Procedure

Run the following command to authorize a domain account:

EXEC master.dbo.rds_grant_ssis_to_login [*login*];

login: indicates the name of the domain account to be authorized.

Example

Run the following command to authorize the domain account **JHA\DCADMIN**:

EXEC master.dbo.rds_grant_ssis_to_login [JHA\DCADMIN];

If the authorization is successful, the following information is displayed:

```
RDS_Process_Successful
```

4.27.3 Deploying an SSIS Project

Scenarios

Run a stored procedure to deploy an SSIS project.

Prerequisites

You have connected to an RDS for SQL Server instance. For details about how to connect to an instance, see **Connecting to an Instance**.

Procedure

Run the following command to deploy an SSIS project:

EXEC msdb.dbo.rds_ssis_task '@task_type', '@folder_name', '@project_name', '@file_name';

- **@task_type**: task type. Set this parameter to **DEPLOY_PROJECT**.
- **@folder_name**: SSIS folder name.
- **@project_name**: SSIS project name.
- **@file_name**: name of the **.ispac** file generated by the SSIS project.

Example

Run the following command to deploy the SSIS project **DeploymentTutorial**:

EXEC msdb.dbo.rds_ssis_task 'DEPLOY_PROJECT', 'test_ssis', 'DeploymentTutorial', 'DeploymentTutorial.ispac';

After the deployment is successful, the following information is displayed:

```
RDS_Process_Successful
```


4.27.4 Changing Custom Database Names

Scenarios

You can use a stored procedure to change a custom database name.

Prerequisites

- An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance**.
- For primary/standby DB instances, you need to run the following command to remove database mirroring between them:

  alter database [*dbname*] **set partner off**
- After the primary database name is changed, the system will automatically establish mirroring relationship.

  If you do not remove database mirroring for primary/standby DB instances and attempt to change the primary database name, the system displays the following information:

  ```
  Database database name is on mirroring_state.
  ```
- For a DB instance whose DB engine version is 2017 Enterprise Edition, if the database to be renamed is added to the [AG-RDS-YUN] availability group, you must remove the database from the availability group before renaming it. For details, see **Removing a Custom Database from an Availability Group**.

Constraints

- System database names cannot be changed. If you attempt to change the name of a system database, the system displays the following information:

  ```
  Error DBName_Source or DBName_Target. Please can not include in ('msdb','master','model','tempdb','rdsadmin','resource') .
  ```
- The new database name must be unique. If the new database name already exists, the system displays the following information:

  ```
  Database database name already exists. Cannot rename database with the same name.
  ```

Procedure

Run the following command to change a custom database name:

exec msdb.dbo.rds_rename_database N'*oldname***',N'***newname***';**

- ***oldname*** indicates the original database name.
- ***newname*** indicates the new database name.

For example, to change a database name from **ABC** to **XYZ**, run the following command:

exec msdb.dbo.rds_rename_database N'ABC**',N'**XYZ**';**

If the database name is changed, the system displays the following information:

```
The database name 'XYZ' has been set.
```

After the database name is changed, the system will automatically perform a full backup.

4.27.5 Viewing Error Logs

Scenarios

You can use a stored procedure to query specific error logs.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance**.

Procedure

Run the following command to view specific error logs:

EXEC master.dbo.rds_read_errorlog @FileID, @LogType, '@FilterText', '@FilterBeginTime', '@FilterEndTime';

- @FileID: specifies the serial number of a log. The value can be **0**, **1**, **2**, etc.
- @LogType: specifies the log type. **1** indicates error logs and **2** indicates Agent logs.
- @FilterText: specifies the filtering character string.
- @FilterBeginTime: specifies the start time of the specified logs.
- @FilterEndTime: specifies the end time of the specified logs.

For example, to obtain the Agent logs from 09/25/2018 to 09/30/2018 and set the filtering character string to **recovery**, run the following command:

EXEC master.dbo.rds_read_errorlog 0, 1, 'recovery', '2018-09-25', '2018-09-30';

4.27.6 Tracing Flags

Scenarios

You can use a stored procedure to trace flags in the following scenarios:

- Obtain in-depth RDS for SQL Server information, such as Lock Manager lock operations.
- Change some preset RDS for SQL Server behaviors such as stopping the query optimizer to find the timeout time for the execution plan.
- Change the current behavior of certain commands, such as terminating the use of a query prompt.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance Through a Public Network**.

Constraints

- The stored procedure must be executed by a user who has the [CREATE ANY DATABASE] permission. If a user who does not have this permission attempts

to execute the stored procedure, the system displays the following information:

```
Database restores can only be performed by database logins with [CREATE ANY DATABASE] permissions.
```

- The current version only supports the tracing flags 1117, 1118, 1204, 1211, 1222, 1224, and 3604. If you perform operations on other flags, the system displays the following information:

```
Current version just open 1117, 1118, 1204, 1211, 1222, 1224, 3604 permission.
```

- The tracing flag can only contain 1, 0, and -1. If other operations are performed, the system displays the following information:

```
Just support Open:1 Close:0 Check:-1
```

Procedure

To trace a flag, run the following command:

EXEC msdb.dbo.rds_dbcc_trace @Trace_Flag, @Trace_Action;

- @Trace_Flag: specifies the sequence number of a trace flag. Currently, only trace flags 1117, 1118, 1204, 1211, 1222, 1224, and 3604 are supported.
- @Trace_Action: specifies the trace flag operation. The value **1** means to enable the trace flag. The value **0** means to disable the trace flag. The value **-1** means to view the trace flag.

For example, to enable trace flag 1117, run the following command:

EXEC msdb.dbo.rds_dbcc_trace 1117, 1;

4.27.7 Capturing Change Data

Scenarios

You can use a stored procedure to enable or disable the change data capture (CDC) function for a specified database. Change data capture can record the insertion, update, and deletion activities of an enabled table, and provide detailed change information using an easy-to-use relational format.

NOTE

Only RDS for SQL Server enterprise editions, RDS for SQL Server 2016 Standard Edition, and later standard editions support change data capture.

For more information about change data capture, see the official documents.

Prerequisites

- An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance Through a Public Network**.
- The stored procedure must be executed by a user who has the [CREATE ANY DATABASE] permission. If a user who does not have this permission attempts to execute the stored procedure, the system displays the following information:

```
Database restores can only be performed by database logins with [CREATE ANY DATABASE] permissions.
```


Constraints

- The change data capture function cannot be enabled or disabled for system databases. If you attempt to enable or disable change data capture for a system database, the system displays the following information:
  ```
  CDC can not open on system database and [rdsadmin].
  ```
- The change data capture operation can only be 1 or 0. If other operations are performed, the system displays the following information:
  ```
  @dbAction just support 1:open 0:close
  ```

Procedure

- Enable or disable database-level CDC.
  ```
  EXEC msdb.dbo.rds_cdc_db '@DBName', @DBAction;
  ```
  - *@DBName*: Name of the target database.
  - *@DBAction*: Operation type. Entering the value **1** indicates enabling CDC, and entering the value **0** indicates disabling CDC.

  For example, to enable CDC for testDB_1, run the following command:
  ```
  EXEC msdb.dbo.rds_cdc_db 'testDB_1', 1;
  ```
- Enable table-level CDC.
  ```
  EXEC sys.sp_cdc_enable_table
      @source_schema = 'dbo',
      @source_name = 'testtable',
      @role_name = NULL
  ```
  - *@source_schema*: Schema name.
  - *@source_name*: Table name.
  - *@role_name*: used to restrict the access permission on the modified data. If you enter **NULL**, the access permission is not restricted.
- Disable table-level CDC.
  ```
  EXEC sys.sp_cdc_disable_table
      @source_schema = 'dbo',
      @source_name = 'testtable',
      @capture_instance ='all'
  ```
  - *@source_schema*: Schema name.
  - *@source_name*: Table name.
  - *@capture_instance*: Name of the instance for which CDC is disabled. If you enter **all**, CDC will be disabled for all instances.
- Check whether table-level CDC is enabled.
  ```
  use [testdb]
  SELECT is_tracked_by_cdc FROM sys.tables WHERE name='table_name'
  ```
  - If the value **1** is returned, table-level CDC is enabled.
  - If the value **0** is returned, table-level CDC is disabled.

4.27.8 Removing a Custom Database from an Availability Group

Scenarios

You can use a stored procedure to remove a custom database from the availability group [AG-RDS-YUN].

NOTE

The stored procedure supports RDS for SQL Server 2017 Enterprise Edition only.

Prerequisites

- An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance**.
- The custom database to be removed must have been added to the [AG-RDS-YUN] availability group. If you remove a database that has not been added to the availability group, the system displays the following information:
  Database *Database name* is not joined to AG-RDS-YUN.

Constraints

You cannot remove system databases. If you attempt to remove a system database, the system displays the following information:
Error DBName can not in ('msdb','master','model','tempdb','rdsadmin','resource') .

Procedure

To remove a custom database from an availability group, run the following command:

EXEC rdsadmin.dbo.rds_remove_database_from_ag '@DBName';

@DBName: specifies the custom database to be removed.

For example, to remove database testDB_1 from the availability group [AG-RDS-YUN], run the following command:

EXEC rdsadmin.dbo.rds_remove_database_from_ag 'testDB_1';

4.27.9 Replicating Databases

Scenarios

You can use a stored procedure to back up a database and restore it to a new database.

Prerequisites

- An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance Through a Public Network**.
- The stored procedure must be executed by a user who has the [CREATE ANY DATABASE] permission. If a user who does not have this permission attempts to execute the stored procedure, the system displays the following information:
  Database restores can only be performed by database logins with [CREATE ANY DATABASE] permissions.
- To back up a custom database, the execution account must be a member of the db_owner or db_backupoperator role group in the database. If a user who does not have the corresponding permission attempts to execute the stored procedure, the system displays the following information:


```
Database backups can only be performed by members of db_owner or db_backupoperator roles in the source database
```

Constraints

- You cannot replicate the system databases. If you attempt to replicate a system database, the system displays the following information:
  ```
  Error DBName_Source or DBName_Target. Please can not include in ('msdb','master','model','tempdb','rdsadmin','resource') .
  ```
- The target database to be restored to cannot have the same database name as the source database. Otherwise, the system displays the following information:
  Database *database name* already exists. Cannot restore database with the same name.

Procedure

Run the following command to replicate a database:

EXEC msdb.dbo.rds_copy_database '@DBName_Source', '@DBName_Target';

- @DBName_Source: indicates the source database to be backed up.
- @DBName_Target: indicates the target database to be restored to.

For example, to replicate database **testDB_1** to obtain a new database **testDB_2**, run the following command:

EXEC msdb.dbo.rds_copy_database 'testDB_1', 'testDB_2';

NOTE

- If the database version is RDS for SQL Server 2012 (Standard Edition, Enterprise Edition, or Web Edition), use the stored procedure **msdb.dbo.rds_copy_database_2012** to back up the database.
- If the database version is RDS for SQL Server 2016 (Standard Edition, Enterprise Edition, or Web Edition), use the stored procedure **msdb.dbo.rds_copy_database_2016** to back up the database.
- If the database version is RDS for SQL Server 2017 Enterprise Edition, use the stored procedure **msdb.dbo.rds_copy_database_2017** to back up the database.

4.27.10 Granting Database Permissions to Subaccounts

Scenarios

You can use a stored procedure to grant permissions of a custom database to a specified subaccount created by the **rdsuser** user to make the database visible to the subaccount. If the database permissions are not granted to the subaccount, the subaccount cannot see or perform operations on the database.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance Through a Public Network**.

Constraints

- You cannot use the stored procedure to grant system database permissions to subaccounts. If you attempt to grant system database permissions to a subaccount, the system displays the following information:
  ```
  Error DatabaseName. Please can not include in ('msdb','master','model','tempdb','rdsadmin') .
  ```
- You cannot use the stored procedure to grant database permissions to system administrators. If you attempt to grant any database permissions to a system administrator, the system displays the following information:
  ```
  Error Login. Please can not include in ('rdsadmin','rdsmirror','rdsbackup','rdsuser') .
  ```
- If an account is already specified in a database, you cannot use the stored procedure to grant the database permissions to the account. Otherwise, the system displays the following information:
  ```
  The proposed new database owner is already a user or aliased in the database.
  ```
  In this case, you can delete the subaccount from the database as the **rdsuser** user first and then execute the stored procedure to grant permissions.
- If an account has the Create Any Database permission, the stored procedure does not take effect for this account.

Procedure

Run the following command to grant database permissions to a subaccount:

EXEC rdsadmin.dbo.rds_AUTHORIZATION_DatabaseForLogin '@DBName', '@Login';

- @DBName: indicates the database for which the permissions are to be granted.
- @Login: indicates the account for which the permissions are to be granted.

For example, to grant permissions of database testDB_1 to account user_1, run the following command:

EXEC rdsadmin.dbo.rds_AUTHORIZATION_DatabaseForLogin 'testDB_1', 'user_1';

After the permissions are granted, the **user_1** user can see and perform operations on the testDB_1 database. For databases whose permissions are not granted, the **user_1** user cannot see or perform operations on them.

4.27.11 Deleting Custom Databases

Scenarios

You can use a stored procedure to delete a custom database.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance**.

Constraints

- This stored procedure cannot be used to delete system databases. If you attempt to delete a system database, the system displays the following information:
  ```
  Error DBName can not  in ('msdb','master','model','tempdb','rdsadmin','resource') .
  ```
- The stored procedure cannot be used to delete a database that does not exist. If you attempt to delete a database that does not exist, the system displays the following information:
  ```
  Cannot find database XXX.
  ```

Procedure

Run the following command to delete a custom database:

EXEC rdsadmin.dbo.rds_drop_database '*@DBName*';

In the preceding command, ***@DBName*** indicates the name of the database to be deleted.

For example, to delete custom database testDB_1, run the following command:

EXEC rdsadmin.dbo.rds_drop_database 'testDB_1';

4.27.12 Updating Database Statistics

Scenarios

You can use a stored procedure to update statistics to improve query performance.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details about how to connect to a DB instance, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following command to update statistics on all databases by default:

EXEC rdsadmin.dbo.rds_updatestats ;

Run the following command to update statistics on a specified database:

EXEC rdsadmin.dbo.rds_updatestats *'@DBname'*;

The ***@DBname*** parameter indicates the name of the database whose statistics are to be updated.

Example:

EXEC rdsadmin.dbo.rds_updatestats *'MyTestDb'*;

After the database statistics are updated, the system displays the following information:

```
Statistics for all tables have been updated.
```


4.27.13 Cycling SQL Server Agent Error Logs

Scenarios

You can use a stored procedure to close the current RDS for SQL Server Agent error log file and cycle the Agent error log extension numbers (just like a server restart). The new Agent error log contains a line indicating that the new log has been created.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the Microsoft SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following command to cycle the RDS for SQL Server Agent error log:

EXEC [msdb].[dbo].[rds_cycle_agent_errorlog]

After the command is executed, the system displays the following information.

```
HW_RDS_Process_Successful_Completed
```

4.27.14 Cycling SQL Server Error Logs

Scenarios

You can use a stored procedure to close the current SQL Server error log file and cycle the SQL Server error log extension numbers (just like a server restart). The new error log contains version and copyright information and a line indicating that the new log has been created.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following command to cycle the RDS for SQL Server error log:

EXEC [msdb].[dbo].[rds_cycle_errorlog]

After the command is executed, the system displays the following information.

```
DBCC execution completed. If DBCC printed error messages, contact your system administrator.
RDS_Process_Successful_Completed
```


4.27.15 Creating Alerts

Scenarios

You can use a stored procedure to create an alert.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to create an alert:

EXEC [msdb].[dbo].[rds_add_alert]

@name='name',

@message_id=message_id,

@severity=severity,

@enabled=enabled,

@delay_between_responses= delay_between_responses,

@notification_message='notification_message',

@include_event_description_in=include_event_description_in,

@database_name='database',

@event_description_keyword='event_description_keyword_pattern',

@job_id=job_id,

@job_name='job_name',

@raise_snmp_trap=raise_snmp_trap,

@performance_condition='performance_condition',

@category_name='category',

@wmi_namespace='wmi_namespace',

@wmi_query='wmi_query';

Table 4-32 Parameter description

Parameter	Description
'name'	The name of the alert. The name appears in the e-mail or pager message sent in response to the alert. It must be unique and can contain the percent (%) character. **name** is **sysname**, with no default.

Parameter	Description
message_id	The message error number that defines the alert. (It usually corresponds to an error number in the **sysmessages** table.) **message_id** is **int**, with a default of **0**. If **severity** is used to define the alert, **message_id** must be **0** or **NULL**.
severity	The severity level (from 1 through 25) that defines the alert. Any SQL Server message stored in the **sysmessages** table sent to the Windows application log with the indicated severity causes the alert to be sent. **severity** is **int**, with a default of **0**. If **message_id** is used to define the alert, **severity** must be **0**.
enabled	The current status of the alert. **enabled** is **tinyint**, with a default of **1** (enabled). If the value is **0**, the alert is not enabled and does not fire.
delay_between_responses	The wait period, in seconds, between responses to the alert. **delay_between_responses** is **int**, with a default of **0**, which means there is no waiting between responses (each occurrence of the alert generates a response). The response can be in either or both of these forms: ● One or more notifications sent through e-mail or pager. ● A job to execute. By setting this value, it is possible to prevent, for example, unwanted e-mail messages from being sent when an alert repeatedly occurs in a short period of time.
'notification_message'	An optional additional message sent to the operator as part of the e-mail, net send, or pager notification. **notification_message** is **nvarchar(512)**, with a default of **NULL**. Specifying **notification_message** is useful for adding special notes such as remedial procedures.
include_event_description_in	Whether the description of the SQL Server error should be included as part of the notification message. **include_event_description_in** is **tinyint**, with a default of **5** (e-mail and net send), and can have one or more of these values combined with an **OR** logical operator.
'database'	The database in which the error must occur for the alert to fire. If **database** is not supplied, the alert fires regardless of where the error occurred. **database** is **sysname**. Names that are enclosed in brackets ([ ]) are not allowed. The default value is **NULL**.
'event_description_keyword_pattern'	The sequence of characters that the description of the SQL Server error must be like. Transact-SQL LIKE expression pattern-matching characters can be used. **event_description_keyword_pattern** is **nvarchar(100)**, with a default of **NULL**. This parameter is useful for filtering object names (for example, **%customer_table%**).

Parameter	Description
job_id	The job identification number of the job to run in response to this alert. **job_id** is **uniqueidentifier**, with a default of **NULL**.
'job_name'	The name of the job to be executed in response to this alert. **job_name** is **sysname**, with a default of **NULL**.
raise_snmp_trap	Not implemented in SQL Server version 7.0. **raise_snmp_trap** is **tinyint**, with a default of **0**.
'performance_condition'	A value expressed in the format "itemcomparatorvalue." **performance_condition** is **nvarchar(512)** with a default of **NULL**, and consists of these elements. • Item: A performance object, performance counter, or named instance of the counter • Comparator: One of these operators: >, <, or = • Value: Numeric value of the counter
'category'	The name of the alert category. **category** is **sysname**, with a default of **NULL**.
'wmi_namespace'	The WMI namespace to query for events. **wmi_namespace** is **sysname**, with a default of **NULL**. Only namespaces on the local server are supported.
'wmi_query'	The query that specifies the WMI event for the alert. **wmi_query** is **nvarchar(512)**, with a default of **NULL**.

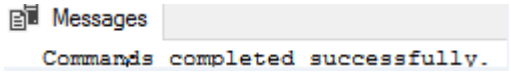
After the command is executed, the system displays the following information.

```
Commands completed successfully.
```

Example

```
EXEC [msdb].[dbo].[rds_add_alert]
        @name = N'test',
        @message_id = 1001,
        @severity = 0,
        @notification_message = N'notification_message',
        @job name=N'jobname';
```

The command output is as follows.



4.27.16 Setting Up Notifications for Alert

Scenarios

You can use a stored procedure to set up a notification for an alert.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to set up notifications for the alert:

EXEC [msdb].[dbo].[rds_add_notification]

@alert_name='alert',

@operator_name='operator',

@notification_method= notification_method;

Table 4-33 Parameter description

Parameter	Description
'alert'	The alert for this notification. **alert** is **sysname**, with no default.
'operator'	The operator to be notified when the alert occurs. **operator** is **sysname**, with no default.
notification_method	The method by which the operator is notified. **notification_method** is **tinyint**, with no default. **notification_method** can be one or more of these values combined with an **OR** logical operator. • **1**: E-mail • **2**: Pager • **4**: net send

After the command is executed, the system displays the following information.

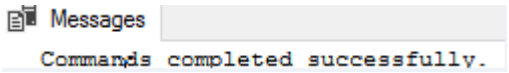
```
Commands completed successfully.
```

Example

```
EXEC [msdb].[dbo].[rds_add_notification]

        @alert_name = N'test',
        @operator_name = N'TestOperator',
        @notification_method = 1;
```

The following figure shows an example command output.



4.27.17 Creating Operators for Alerts and Jobs

Scenarios

You can use a stored procedure to create an operator (notification recipient) for use with alerts and jobs.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to create an operator for alerts and jobs:

EXEC [msdb].[dbo].[rds_add_operator]

@name ='name',

@enabled=enabled,

@email_address='email_address',

@pager_address='pager_address',

@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 4-34 Parameter description

Parameter	Description
'name'	The name of an operator (notification recipient). This name must be unique and cannot contain the percent (%) character. **name** is **sysname**, with no default.
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.

Parameter	Description
'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_address'	The pager address of the operator. This string is passed directly to the e-mail system. **pager_address** 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 the SQL Server Agent 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.
saturday_pager_start_time	The time after which the SQL Server Agent 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 the SQL Server Agent 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 the SQL Server Agent 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 the SQL Server Agent 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.

Parameter	Description
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**.
'netsend_address' '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.
```

Example

```
EXEC    [msdb].[dbo].[rds_add_operator]
        @name = N'HWTest01',
        @enabled = 1,
        @email_address = N'hw',
        @pager_address = N'test01@      .com',
        @weekday_pager_start_time = 080000,
        @weekday_pager_end_time = 170000,
        @pager_days = 62;
```

The command output is as follows.

```
Messages
  Commands completed successfully.
```

4.27.18 Updating Alert Settings

Scenarios

You can use a stored procedure to update the settings of an existing alert.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**. An RDS for SQL Server DB instance has been connected.

Procedure

Run the following commands to update the settings of an existing alert:

EXEC [msdb].[dbo].[rds_update_alert]

@name='name',

@new_name = 'new_name',

@message_id=message_id,

@severity=severity,

@enabled=enabled,

@delay_between_responses= delay_between_responses,

@notification_message='notification_message',

@include_event_description_in=include_event_description_in,

@database_name='database',

@event_description_keyword= 'event_description_keyword',

@job_id=job_id | @job_name='job_name',

@occurrence_count= occurrence_count,

@count_reset_date= count_reset_date,

@count_reset_time= count_reset_time,

@last_occurrence_date= last_occurrence_date,

@ last_occurrence_time= last_occurrence_time,

@ last_response_date= last_response_date,

@ last_response_time= last_response_time,

@ raise_snmp_trap= raise_snmp_trap,

@ performance_condition= 'performance_condition',

@category_name='category',

@wmi_namespace='wmi_namespace',

@wmi_query='wmi_query';

Table 4-35 Parameter description

Parameter	Description
'name'	The name of the alert that is to be updated. **name** is **sysname**, with no default.
'new_name'	A new name for the alert. The name must be unique. **new_name** is **sysname**, with a default of **NULL**.
message_id	A new message or error number for the alert definition. Typically, **message_id** corresponds to an error number in the **sysmessages** table. **message_id** is **int**, with a default of **NULL**. A message ID can be used only if the severity level setting for the alert is **0**.
severity	A new severity level (from 1 through 25) for the alert definition. Any SQL Server message sent to the Windows application log with the specified severity will activate the alert. **severity** is **int**, with a default of **NULL**. A severity level can be used only if the message ID setting for the alert is **0**.
enabled	Whether the alert is enabled (**1**) or not enabled (**0**). **enabled** is **tinyint**, with a default of **1** (enabled). If the value is **0**, the alert is not enabled and does not fire.
delay_between_responses	The new waiting period, in seconds, between responses to the alert. **delay_between_responses** is **int**, with a default of **0**, which means there is no waiting between responses (each occurrence of the alert generates a response). The response can be in either or both of these forms: • One or more notifications sent through e-mail or pager. • A job to execute. By setting this value, it is possible to prevent, for example, unwanted e-mail messages from being sent when an alert repeatedly occurs in a short period of time.
'notification_message'	The revised text of an additional message sent to the operator as part of the e-mail, net send, or pager notification. **notification_message** is **nvarchar(512)**, with a default of **NULL**. Specifying **notification_message** is useful for adding special notes such as remedial procedures.
include_event_description_in	Whether the description of the SQL Server error from the Windows application log should be included in the notification message. **include_event_description_in** is **tinyint**, with a default of **NULL**, and can be one or more of these values. • **0**: None • **1**: E-mail • **2**: Pager • **4**: net send • **7**: All

Parameter	Description
'database'	The name of the database in which the error must occur for the alert to fire. If **database** is not supplied, the alert fires regardless of where the error occurred. **database** is **sysname**. Names that are enclosed in brackets ([ ]) are not allowed. The default value is **NULL**.
'event_description_keyword'	A sequence of characters that must be found in the description of the error in the error message log. Transact-SQL LIKE expression pattern-matching characters can be used. **event_description_keyword** is **nvarchar(100)**, with a default of **NULL**. This parameter is useful for filtering object names (for example, **%customer_table%**).
job_id	The job identification number. **job_id** is **uniqueidentifier**, with a default of **NULL**. If **job_id** is specified, **job_name** must be omitted.
'job_name'	The name of the job that executes in response to this alert. **job_name** is **sysname**, with a default of **NULL**. If **job_name** is specified, **job_id** must be omitted.
occurrence_count	Resets the number of times the alert has occurred. **occurrence_count** is **int**, with a default of **NULL**, and can be set only to **0**.
count_reset_date	Resets the date the occurrence count was last reset. **count_reset_date** is **int**, with a default of **NULL**.
count_reset_time	Resets the time the occurrence count was last reset. **count_reset_time** is **int**, with a default of **NULL**.
last_occurrence_date	Resets the date the alert last occurred. **last_occurrence_date** is **int**, with a default of **NULL**, and can be set only to **0**.
last_occurrence_time	Resets the time the alert last occurred. **last_occurrence_time** is **int**, with a default of **NULL**, and can be set only to **0**.
last_response_date	Resets the date the alert was last responded to by the SQL Server Agent service. **last_response_date** is **int**, with a default of **NULL**, and can be set only to **0**.
last_response_time	Resets the time the alert was last responded to by the SQL Server Agent service. **last_response_time** is **int**, with a default of **NULL**, and can be set only to **0**.
raise_snmp_trap	Reserved.

Parameter	Description
'performance_condition'	A value expressed in the format "itemcomparatorvalue." **performance_condition** is **nvarchar(512)**, with a default of **NULL**, and consists of the following elements: • Item: A performance object, performance counter, or named instance of the counter • Comparator: One of these operators: >, <, or = • ReplTest1: Numeric value of the counter
'category'	The name of the alert category. **category** is **sysname**, with a default of **NULL**.
'wmi_namespace'	The WMI namespace to query for events. **wmi_namespace** is **sysname**, with a default of **NULL**.
'wmi_query'	The query that specifies the WMI event for the alert. **wmi_query** is **nvarchar(512)**, with a default of **NULL**.

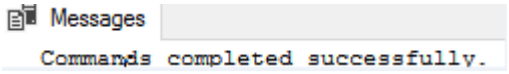
After the command is executed, the system displays the following information.

```
Commands completed successfully.
```

Example

```
EXEC    [msdb].[dbo].[rds_update_alert]
        @name='testAlert',
        @new_name='newName',
        @enabled=0;
```

The command output is as follows.



4.27.19 Updating Alert Notification Methods

Scenarios

You can use a stored procedure to update the notification method of an alert notification.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to update the notification method of an alert notification:

EXEC [msdb].[dbo].[rds_update_notification]

@alert_name = 'alert',

@operator_name ='operator',

@notification_method =notification;

Table 4-36 Parameter description

Parameter	Description
'alert'	The name of the alert associated with this notification. **alert** is **sysname**, with no default.
'operator'	The operator who will be notified when the alert occurs. **operator** is **sysname**, with no default.
notification	The method by which the operator is notified. **notification** is **tinyint** with no default, and can be one or more of the following values: • **1**: E-mail • **2**: Pager • **4**: net send • **7**: All methods

After the command is executed, the system displays the following information.

```
Commands completed successfully.
```

Example

```
EXEC      [msdb].[dbo].[rds_update_notification]
        @alert_name='testAler',
        @operator_name='operator',
        @notification_method=7;
```

The command output is as follows.

```
Messages
  Commands completed successfully.
```


4.27.20 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 for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

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 4-37 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.

Parameter	Description
'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 the SQL Server Agent 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.
saturday_pager_start_time	The time after which the SQL Server Agent 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 the SQL Server Agent 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 the SQL Server Agent 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 the SQL Server Agent 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.

Parameter	Description
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.
```

4.27.21 Removing Alerts

Scenarios

You can use a stored procedure to remove an alert.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to remove an alert:

```
EXEC [msdb].[dbo].[rds_delete_alert]
@name='name';
```


Table 4-38 Parameter description

Parameter	Description
'name'	The name of the alert. This parameter is of **sysname** data type, with no default value.

After the command is executed, the system displays the following information.

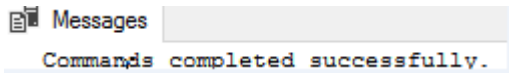
```
Commands completed successfully.
```

Example

EXEC [msdb].[dbo].[rds_delete_alert]

@name='test';

The command output is as follows.



4.27.22 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 for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to remove the RDS for SQL Server Agent notification definition for a specific alert and operator:

EXEC [msdb].[dbo].[rds_delete_notification]

@alert_name = 'alert',

@operator_name ='operator';

Table 4-39 Parameter description

Parameter	Description
'alert'	The name of the alert. This parameter is of **sysname** data type, with no default value.

Parameter	Description
'operator'	The name of the operator. This parameter is of **sysname** data type, with no default value.

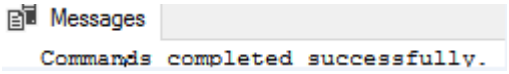
After the command is executed, the system displays the following information.

```
Commands completed successfully.
```

Example

```
EXEC [msdb].[dbo].[rds_delete_notification]
@alert_name = 'alert',
@operator_name = N'TestOperator';
```

The command output is as follows.



4.27.23 Removing Operators

Scenarios

You can use a stored procedure to remove an operator.

Prerequisites

An RDS for SQL Server DB instance has been connected. Connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following commands to remove an operator:

```
EXEC [msdb].[dbo].[rds_delete_operator]
@name='name',
@reassign_to_operator = 'reassign_operator';
```

Table 4-40 Parameter description

Parameter	Description
'name'	The name of the operator to delete. This parameter is of **sysname** data type, with no default value.

Parameter	Description
'reassign_operator'	The name of an operator to whom the specified operator's alerts can be reassigned. This parameter is of **sysname** data type, with a default value of **NULL**.

After the command is executed, the system displays the following information.

```
Commands completed successfully.
```

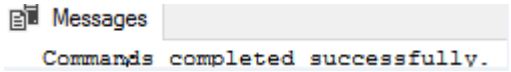
Example

EXEC [msdb].[dbo].[rds_delete_operator]

@name = N'Test01',

@reassign_to_operator = NULL;

The command output is as follows.



4.27.24 Shrinking Databases

Scenarios

You can use stored procedures to shrink the size of the data and log files in a specified database.

- **rds_shrink_database**: shrinks all files of a specified database.
- **rds_shrink_database_log**: shrinks log files of a specified database.

For more operations, see **Shrinking an RDS for SQL Server Database**.

Prerequisites

- Before shrinking a database, ensure that your instance has sufficient storage space.
- Your RDS for SQL Server DB instance has been connected. You can connect to the DB instance through the SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Shrinking a Database

Step 1 Run the following command to shrink the database:

EXEC [master].[dbo].[rds_shrink_database] @DBName='*myDbName***';**

Table 4-41 Parameter description

Parameter	Description
myDbName	Name of the database to be shrunk. If this parameter is not specified, all databases are shrunk by default.

Step 2 After the command is successfully executed, the following information is displayed:

```
RDS_Process_Successful: Shrink Database Done.
```

----End

Shrinking Database Log Files

Run the following command to shrink log files of a specified database:

EXEC [master].[dbo].[rds_shrink_database_log] @dbname;

@dbname: indicates the name of the database whose log files are to be shrunk.

Example

1. Run the following command to shrink the **dbtest2** database:

   EXEC [master].[dbo].[rds_shrink_database] @DBName = 'test';

   The command output is as follows.

   ```
   [Shrink Start] Date and time: 2024-02-19 19:58:06

   Start to shrink files in database [test], current file id is 1...
   Cannot shrink file '1' in database 'test' to 6400 pages as it only contains 1024 pages.
   DBCC execution completed. If DBCC printed error messages, contact your system administrator.
   Shrink file (id: 1) in database [test] done!

   Start to shrink files in database [test], current file id is 2...
   Cannot shrink file '2' in database 'test' to 6400 pages as it only contains 1024 pages.
   DBCC execution completed. If DBCC printed error messages, contact your system administrator.
   Shrink file (id: 2) in database [test] done!

   [Shrink End] Date and time: 2024-02-19 19:58:06

   RDS_Process_Successful: Shrink Database Done.
   ```

2. Run the following command to shrink all databases:

   EXEC [master].[dbo].[rds_shrink_database];

3. Run the following command to shrink the log files of the **testdb** database:

   EXEC [master].[dbo].[rds_shrink_database_log]@dbname='dbtest3';

FAQs

1. If an error message indicating that the log file is in use is displayed during the execution of the stored procedure, run the stored procedure later.
2. If the log file size is not changed after the stored procedure is executed, run the following SQL statement in the database to check whether there is enough available space in the log file:

   ```
   SELECT name, size/128.0 - CAST(FILEPROPERTY(name, 'SpaceUsed') AS int)/128.0 AS AvailableSpaceInMB FROM sys.database_files WHERE type_desc='LOG';
   ```

3. If the log file size does not change after the stored procedure for log shrinking has been executed multiple times, the log file is in use. Run the following SQL statement to check whether the log file is being used:


```
SELECT name, log_reuse_wait_desc FROM sys.databases where name='test';
```

If the log file is being used, wait for a period of time and then shrink it again.

Table 4-42 log_reuse_wait_desc description

log_reuse_wait_desc Value	Description
NOTHING	There are one or more reusable virtual log files (VLFs).
CHECKPOINT	Checkpoints have not been generated since the last log truncation, or the log header has not been moved across VLFs (all recovery models).
LOG_BACKUP	Before the transaction log is truncated, it needs to be backed up.
ACTIVE_BACKUP_OR_RESTORE	Data is being backed up or restored.
ACTIVE_TRANSACTION	The transaction is active.
DATABASE_MIRRORING	Database mirroring is suspended, or the mirror database lags behind the principal database in high-performance mode.
REPLICATION	During transaction replication, the transaction related to the publication is still not delivered to the distribution database.
DATABASE_SNAPSHOT_CREATION	A database snapshot is being created.
LOG_SCAN	Log scanning is in progress.
AVAILABILITY_REPLICA	The secondary replica of an availability group is applying the transaction log records of this database to the corresponding secondary database.

4.27.25 Changing the Permission to View All Databases

Scenarios

You can use a stored procedure to grant the permission to view all databases for a specified account. If this permission is revoked, only the master and tempdb databases can be viewed.

Precautions

- The stored procedure can only be executed by the **rdsuser** user or the database login account. The login account has all the permissions of the **rdsuser** user on RDS for SQL Server instances. For details about the stored

procedure for creating a database login account, see **Creating a Database Account**.

- By default, all users are assigned the public role and can view all databases in the instance. However, they cannot access or edit the databases that they do not have permissions for.
- The database viewing permissions of **rdsuser** and other built-in accounts cannot be changed. For details about the built-in accounts, see **Database Account Security**.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details about how to connect to an instance through the SQL Server client, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following command to configure the permission to view all databases (excluding the master and tempdb databases) for a user:

EXEC master.dbo.rds_view_any_database @user, @action ;

- *@user*: Name of the user.
- *@action*: Operation to be performed.
  - **deny**: Do not allow the user to view all databases.
  - **revoke**: Allow the user to view all databases.

Example

- Do not allow the **testuser** user to view all databases:

  EXEC master.dbo.rds_view_any_database 'testuser','deny' ;
- Allow the **testuser** user to view all databases:

  EXEC master.dbo.rds_view_any_database 'testuser','revoke' ;

4.27.26 Granting Permissions of Database-Level db_owner Role

Scenarios

You can use a stored procedure to grant the **db_owner** role permissions of a database to a specified user.

Precautions

- The stored procedure can be executed only by the **rdsuser** user or the database login account. The login account has all the permissions of the **rdsuser** user on RDS for SQL Server instances. For details about the stored procedure for creating a database login account, see **Creating a Database Account**.

- The database you will grant the permissions for cannot be any of the following system databases: msdb, master, model, tempdb, rdsadmin, and resource.
- Permissions of the **db_owner** role can be granted to **rdsuser**.

Prerequisites

An RDS for SQL Server DB instance has been connected. For details, see **Connecting to a DB Instance Through a Public Network**.

Procedure

Run the following command to grant permissions of the **db_owner** role to a specified user:

EXEC master.dbo.rds_add_db_owner @dbname, @user;

- *@dbname*: name of the database
- *@user*: name of the user

Example

Grant the **db_owner** role permissions of the database **testdb** to **testuser**:

EXEC master.dbo.rds_add_db_owner @dbname='testdb',@user='testuser';

4.28 RDS for SQL Server Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

- Log in to the management console. Click **Service List** and choose **Management & Governance** > **Tag Management Service**. Set predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- Up to 20 tags can be added for each DB instance.

Adding or Editing a Tag

Step 1 **Log in to the management console**.

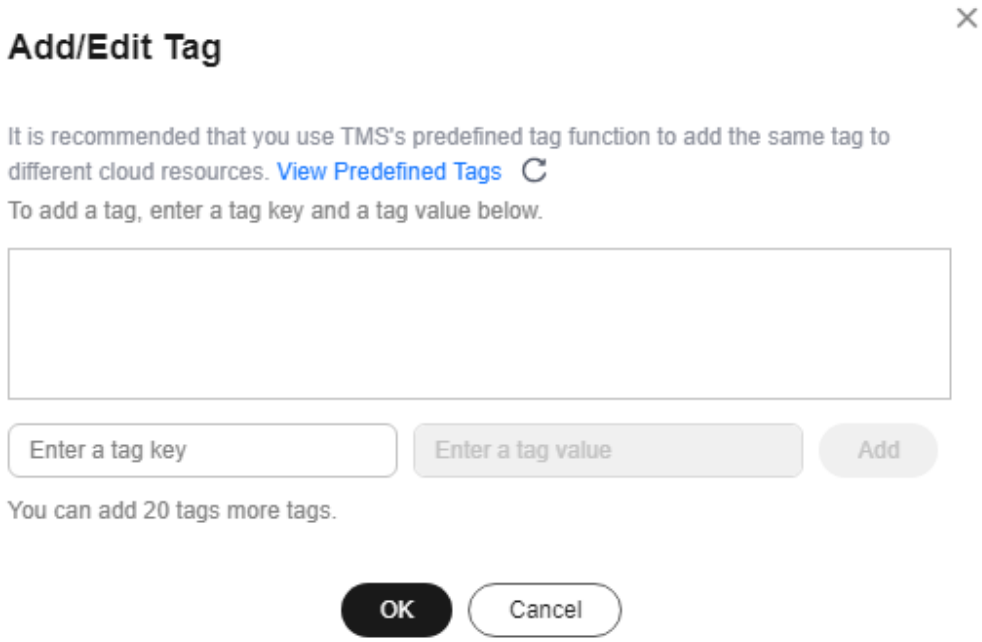
Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target instance name.

Step 5 In the navigation pane on the left, choose **Tags**. On the displayed page, click **Add/Edit Tag**. In the displayed dialog box, enter a tag key and value, click **Add**, and click **OK**.

Figure 4-133 Adding a tag



- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with your DB instances (except the current instance).
- The tag key must be unique. It must consist of 1 to 128 characters and can include letters, digits, spaces, and the following characters: _ **.** **:** **=** **+** **-** **@**. It cannot start or end with a space, or start with **_sys_**.
- The tag value (optional) can consist of up to 255 characters and can include letters, digits, spaces, and the following characters: _ **.** **:** **/** **=** **+** **-** **@**.

Step 6 After a tag has been added, view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 **Log in to the management console**.

Step 2 Click in the upper left corner and select a region.

Step 3 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 4 On the **Instances** page, click the target DB instance.

Step 5 In the navigation pane on the left, choose **Tags**. Select the tag to be deleted and click **Delete**. In the displayed dialog box, click **OK**.

After a tag has been deleted, it will no longer be displayed on the **Tags** page.

----End

4.29 RDS for SQL Server Quotas

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of RDS quotas include the maximum number of DB instances that you can create. Quotas are put in place to prevent excessive resource usage.

If a quota cannot meet your needs, apply for a higher quota.

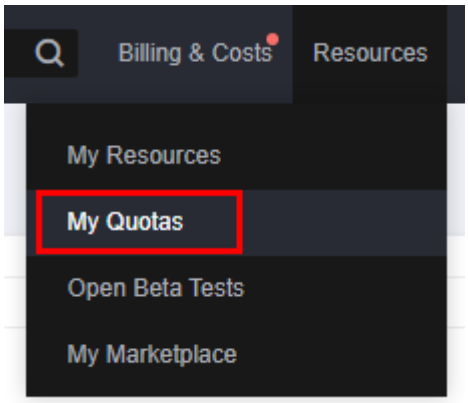
Viewing Quotas

Step 1 Log in to the management console.

Step 2 Click in the upper left corner and select a region.

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Figure 4-134 My quotas



Step 4 On the **Quotas** page, view the used and total quotas of each type of resources.

----End

Increasing Quotas

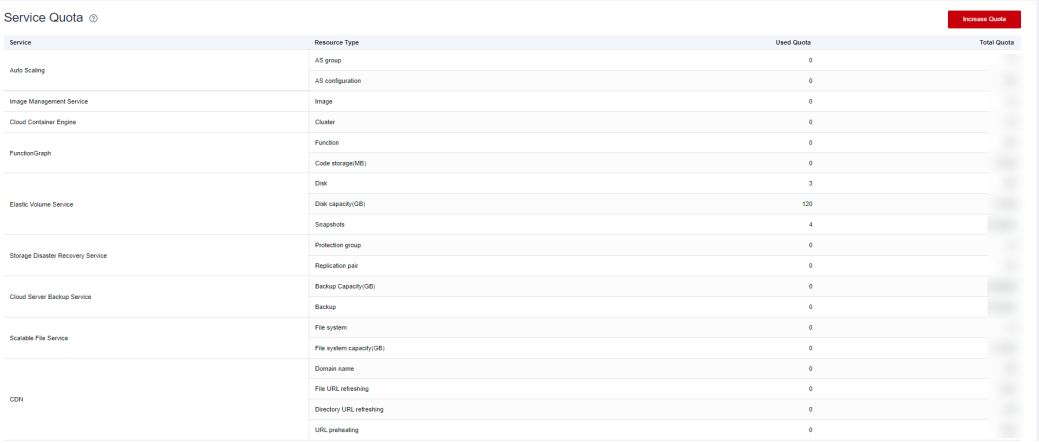
Step 1 Log in to the management console.

Step 2 Click in the upper left corner and select a region.

Step 3 In the upper right corner of the RDS console, choose **Resources** > **My Quotas**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 4-135 Increasing quotas



Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for adjustment.

Step 6 After all required parameters are configured, select the agreement and click **Submit**.

----End

