RDS for SQL Server

User Guide

Issue 01

Date 2025-09-04

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 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.

2 Buying an RDS for SQL Server DB Instance

Scenarios

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

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

Prerequisites

- You have **created a Huawei ID and enabled Huawei Cloud services**.
- You can create an IAM user and user group on the IAM console and grant the user specific operation permissions, to perform refined management on Huawei Cloud. For details, see **Creating a User and Granting Permissions**.

Procedure

Step 1 Go to the **Buy DB Instance** page.

Step 2 On the displayed page, select a billing mode and configure information about your DB instance. Then, click **Next**.

- 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**.
- Basic Information

Table 2-1 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 and you cannot change the region of a DB instance after creating the instance. Therefore, exercise caution when selecting a region.
DB Engine	Set to **Microsoft SQL Server**.
DB Engine Version	For details, see **DB Engines and Versions**. Supported DB engine versions may vary by region. For the actual options, see them on the console. You are advised to select the latest available version because it is more stable, reliable, and secure.
DB Instance Type	– **Primary/Standby**: uses an HA architecture with a primary DB instance and a synchronous standby DB instance. It is suitable for production databases of large- and medium-sized enterprises in Internet, Internet of Things (IoT), retail e-commerce sales, logistics, gaming, and other sectors. The standby DB instance improves instance reliability and is invisible to you after being created. An AZ is a physical region where resources use independent power supply and networks. AZs are physically isolated but interconnected through an internal network. 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. – **Cloud SSD**: supports a maximum throughput of 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. – **Extreme SSD V2**: supports super-high IOPS and throughput as well as super-low latency for the most performance-demanding workloads. With the Extreme SSD V2 type, you can buy disks with the IOPS tailored to your workloads. The disk performance no longer changes with the disk capacity. **NOTE** – Extreme SSD V2 is now available only in CN South-Guangzhou. To use this storage type, **submit a service ticket** to apply for required permissions. – Extreme SSD V2 disks with a preconfigured IOPS higher than 128,000 can only reach the maximum performance on AC7 compute resources. To use such disks, **submit a service ticket**. – If you select **DSS** for **Resource Type**, only the storage type that you have selected when buying the DSS service is displayed by default. – The supported IOPS 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*.

- Instance Configuration

Table 2-2 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 different maximum IOPS. For details about instance classes, see **RDS for SQL Server Instance Classes**. After a DB instance is created, you can change its vCPU and memory. For details, see **Changing a DB Instance Class**. **NOTE** DB instances in a DCC only support the general-enhanced instance class.

Parameter	Description
Resource Type	– **EVS** – **DSS** **NOTE** This option is displayed only when you buy the DSS service.
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. You can enable storage autoscaling for your instance. If the available storage drops to a specified threshold, autoscaling is triggered. 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. – **Enable autoscaling**: If you select this option, autoscaling is enabled. To enable storage autoscaling, you need to **submit a service ticket** to apply for required permissions. – **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. 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 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. ▪ 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

Table 2-3 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 create multiple DB instances and read replicas at a time, the allowed length for each instance name will change. – If you buy multiple DB instances at a time, their names will include a four-digit suffix. For example, if you specify **instance** here, the names will be *instance-0001*, *instance-0002*, and so on. If existing instances' suffixes have already reached up to *0010*, the new instance names will start from *instance-0011*.

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 **rdsuser**.
Administrator Password	Must consist of 8 to 32 characters and contain at least three types of the following characters: uppercase letters, lowercase letters, digits, and special characters (**~!@#$ %^*-_=+?,**). Enter a strong password and periodically change it for security reasons. 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 section **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 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 the 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 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.
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, RDS allocates a security group to you by default. **NOTE** To configure the Active Directory (AD) domain for the DB instance, ensure that the DB instance and domain controller must be in the same security group.

Parameter	Description
Database Port	The default database port is **1433**. You can change it after a DB instance is created. If you need to set the port during purchase, **submit a service ticket** to request permissions. – 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 or 2100 to 9500 (excluding 5050, 5353, 5355, 5985, and 5986). – For other editions, the port number can be set to 1433 or 2100 to 9500 (excluding 5355 and 5985).

- Additional Options

Figure 2-1 Additional options

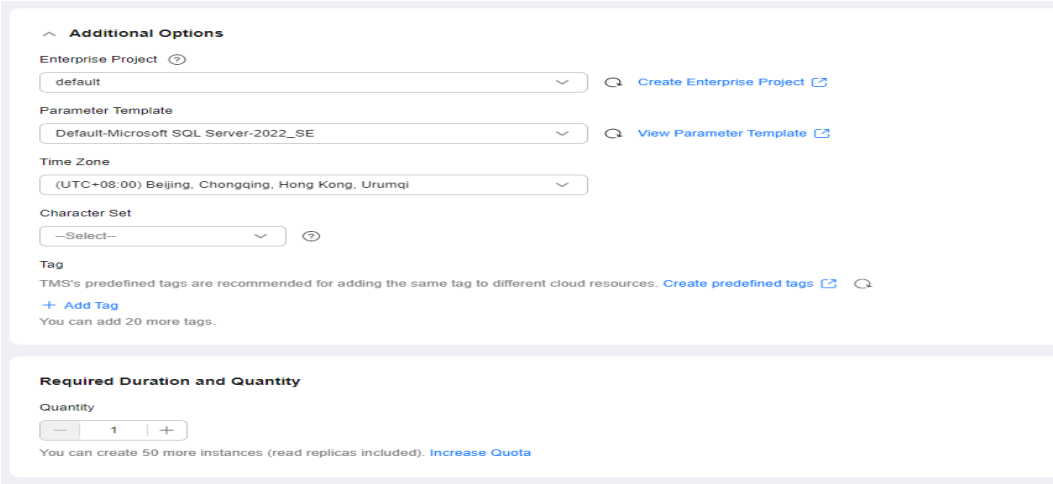


Table 2-4 Additional options

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. **NOTICE** If you use a custom parameter template when creating a DB instance, the specification-related parameter **max server memory (MB)** in the custom template is not delivered. Instead, the default value is used. You can modify the instance parameters as required after the DB instance is created. For details, see **Modifying RDS for SQL Server Instance Parameters**.
Time Zone	You need to select a time zone for your instance based on the region it is hosted in. After an instance is created, the time zone cannot be modified.
Character Set	Defines a collation of a database or table column, or a collation cast operation when applied to character string expression. It acts as the default character set for the DB instance.
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 section **RDS for SQL Server Tags**.

- AD Domain

Table 2-5 AD domain

Parameter	Description
AD Domain	A Windows domain controller directory service that allows domain users to connect to an RDS for SQL Server instance using Windows authentication. Active Directory (AD) is a directory service on Windows Standard Server, Windows Enterprise Server, and Windows Datacenter Server. Active Directory stores information about objects on the network and makes this information easy for administrators and users to find and use. Active Directory uses a structured data store as the basis for a logical, hierarchical organization of directory information. **NOTICE** 1. To use AD domain, **submit a service ticket** to apply for required permissions. 2. When you configure AD domain information during the DB instance creation, do not configure or disable Group Policy Object (GPO) for your domain controller server. Otherwise, the DB instance creation will fail. 3. If GPO is required, you need to buy an ECS and set up a new domain controller server with GPO disabled. Then, establish trust between your domain controller server and the new domain controller server. 4. The domain controller server time must be synchronized to an NTP server. Non-standard time or too large time difference may cause DB instance creation failures. – **Skip**: This option is selected by default. – **Configure**: To configure the AD domain, you must first prepare a domain controller on an ECS or on-premises database. Then, configure the directory address, domain name, directory administrator, and directory administrator password as required. **NOTE** If an RDS for SQL Server single-node instance is configured with the AD domain, it cannot be changed to primary/standby DB instances.
Directory Address	Enter the private IP address of the ECS that supports the AD domain. After the DB instance is created, you can view the directory address on the **Overview** page. Example value: 192.168.x.x.

Parameter	Description
Domain Name	A fully qualified domain name, such as DBStest.com, must: 1. Be the same as the ECS domain name. 2. Be no more than 48 characters long. 3. Only include letters, digits, dots (.), and hyphens (-). 4. Include a valid top-level domain name which contains at least 2 characters long and contains only dots (.) and letters, for example, **.com** 5. After the DB instance is created, you can view the domain name on the **Overview** page.
Directory Administrator	You are advised to set this parameter to the domain administrator which belongs to the Domain Admins group (because high permissions are required for a client to add a domain).
Directory Administrator Password	Password of the directory administrator. Keep this password secure. The system cannot retrieve it.

- Required Duration and Quantity

  Table 2-6 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.
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 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**.
- Otherwise, 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 the DB instance, go to the **Instances** page.

- During the creation process, the DB instance status is **Creating**. When the creation process is complete, the instance status will change to **Available**.
- The automated backup policy is enabled by default. An automated full backup is immediately triggered after a DB instance is created.
- After a DB instance is created, you can enter a description for it.
- The default database port is **1433**. After a DB instance is created, you can change its port. For details, see **Changing a Database Port**.

NOTE

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

----End

Related Operations

Creating a DB Instance Using an API

3 Instance Connection

3.1 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 3-1 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.

3.2 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

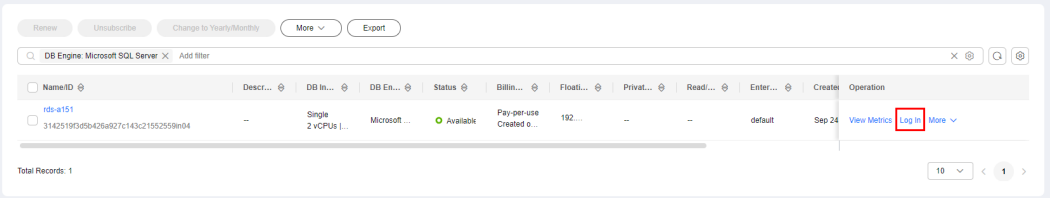
Step 1 Log in to the management console.

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**.

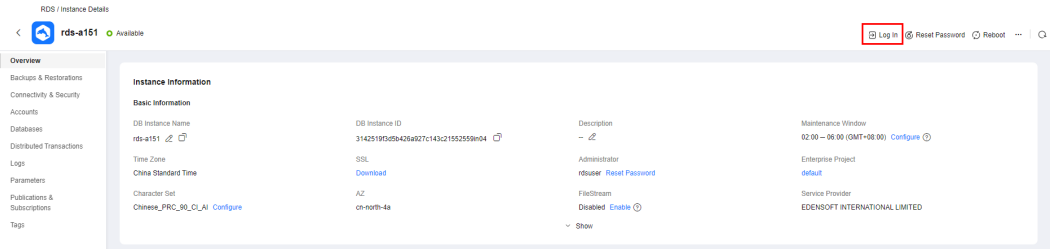
Step 4 On the **Instances** page, locate the DB instance and click **Log In** in the **Operation** column.

Figure 3-1 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 3-2 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**

3.3 Connecting to an RDS for SQL Server Instance Through the SQL Server Management Studio Client

3.3.1 Connecting to a DB Instance from a Windows ECS over a Private Network

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.

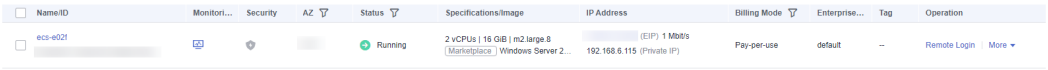
You can connect to an instance through a Secure Socket Layer (SSL) connection or a non-SSL connection using SQL Server Management Studio. The SSL connection encrypts data and is more secure.

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 3-3 ECS

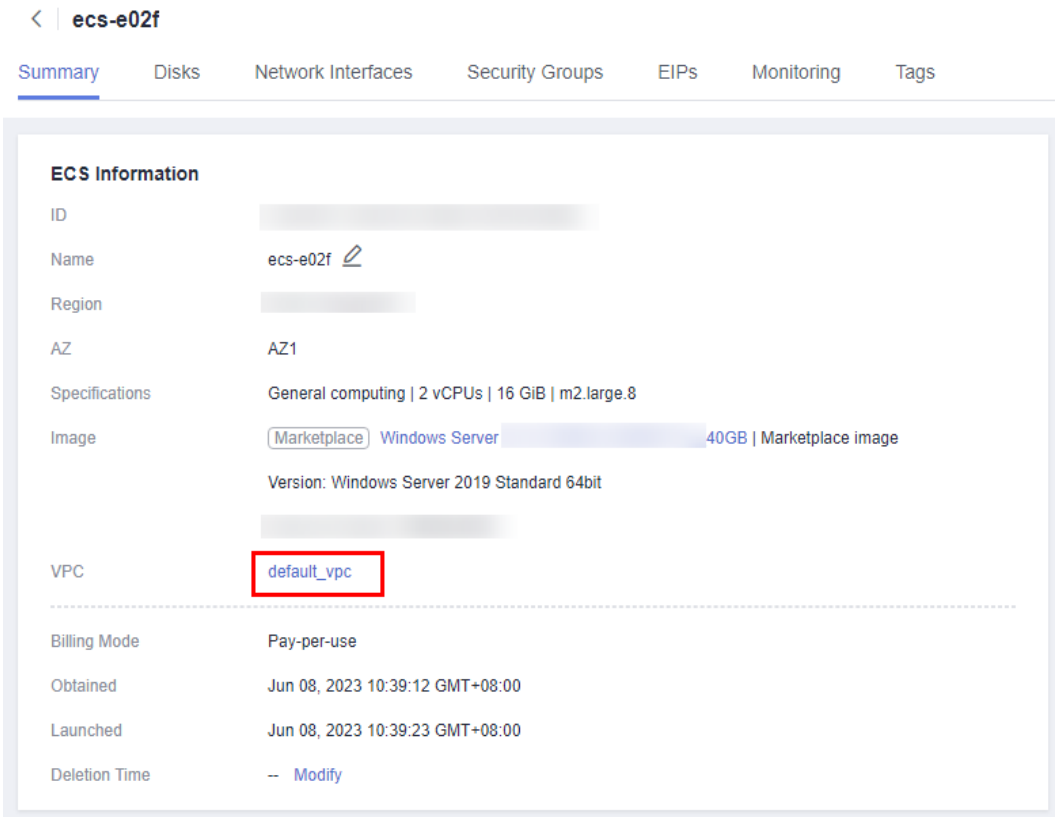
   

2. Buy an ECS and select Windows as its OS.

   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.

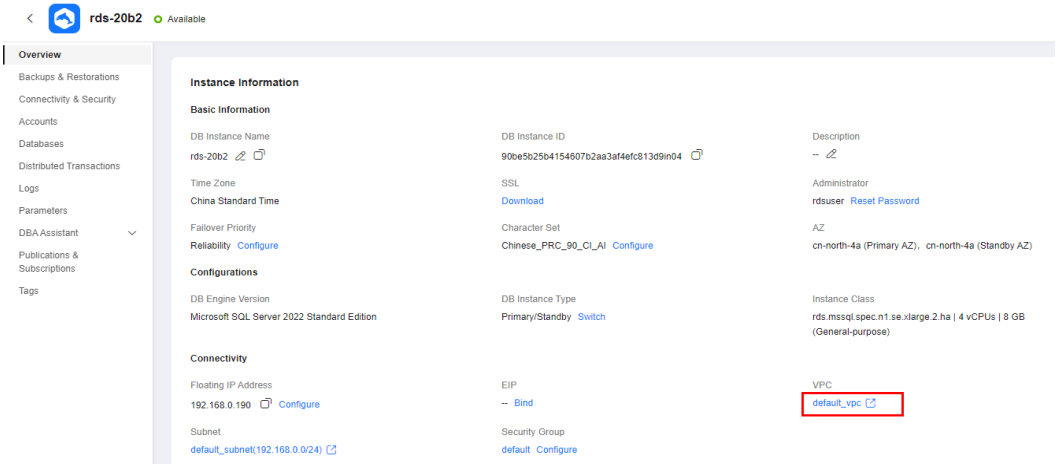
   For details about how to purchase a Windows ECS, see **Purchasing a Custom ECS** in *Elastic Cloud Server User Guide*.
3. On the **ECS Information** page, view the region and VPC of the ECS.

Figure 3-4 ECS information



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

Figure 3-5 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

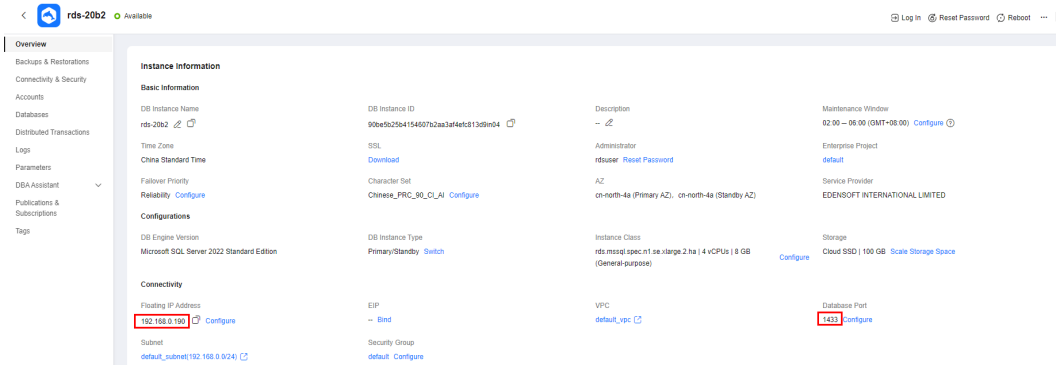
each other. To reduce network latency, deploy your DB instance in the region nearest to your workloads.

- 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 3-6 Connection information



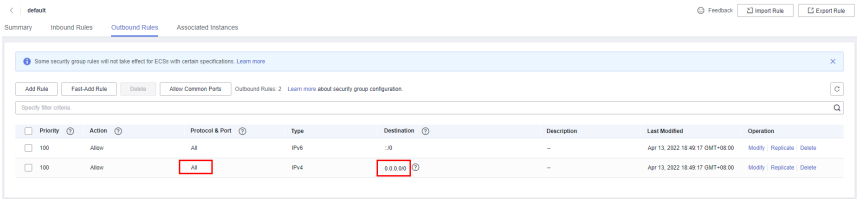
4. Open the cmd window on the ECS and check whether the floating IP address and database port of the DB instance can be connected.

   telnet *192.168.2.182 1433*

   - If yes, network connectivity is normal.
   - If no, check the security group rules.

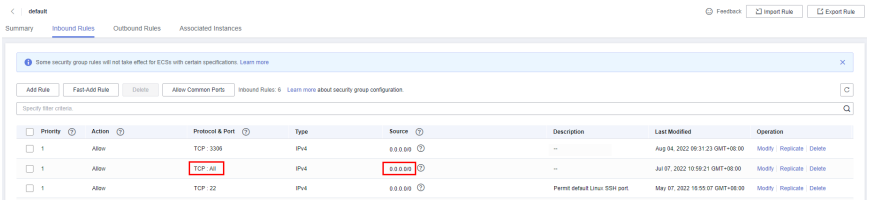
     - If in the security group of the ECS, there is no outbound rule with **Destination** set to **0.0.0.0/0** and **Protocol & Port** set to **All**, add an outbound rule for the floating IP address and port of the DB instance.

       Figure 3-7 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 3-8 DB instance security group

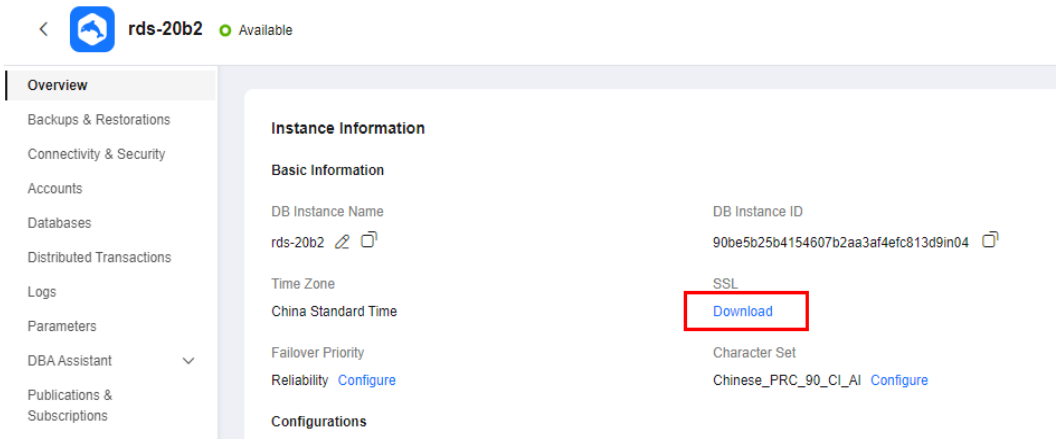


5. Open a browser on the ECS, visit the **Microsoft website**, and download the installation package, for example, SQL Server Management Studio 18.0.
6. Double-click the installation package and complete the installation as instructed.

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 3-9 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 Do I Import the SSL Certificate of an RDS Instance to a Windows or Linux Server?**
4. Start SQL Server Management Studio.
5. Choose **Connect** > **Database Engine**. In the displayed dialog box, enter login information.

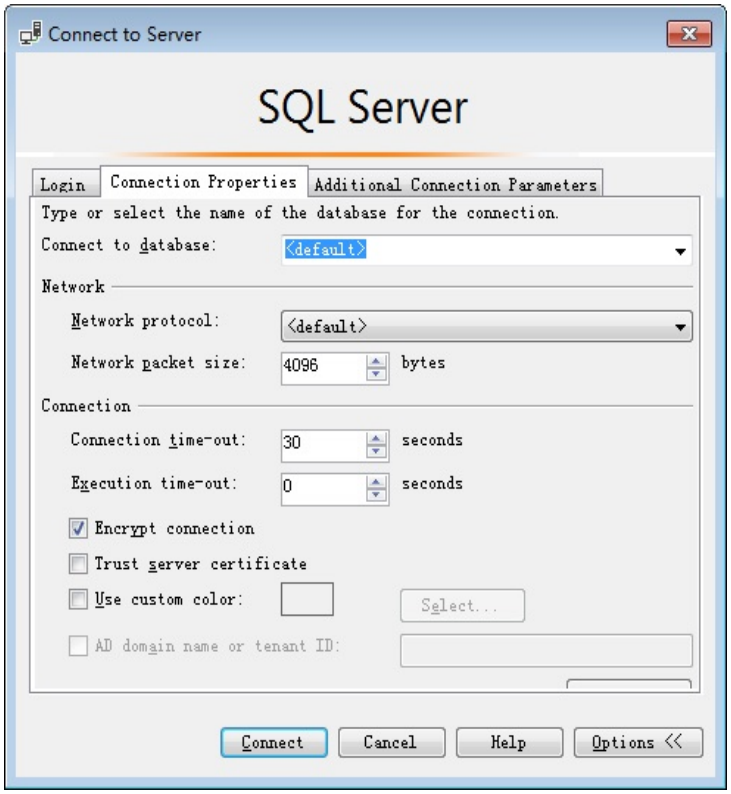
Figure 3-10 Connecting to the server

Table 3-2 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 3-11 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**

3.3.2 Connecting to a DB Instance from a Windows Server over a Public Network

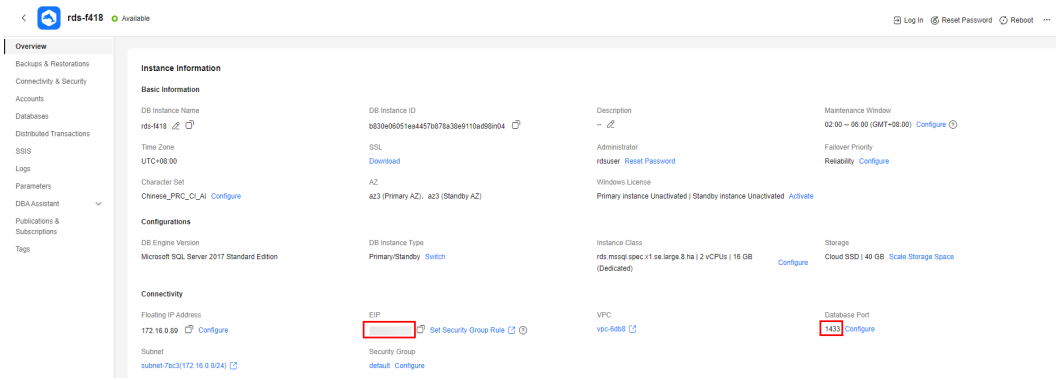
If you cannot access your DB instance through a floating IP address, bind an EIP to the DB instance and connect to it through the EIP.

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.

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 3-12 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.

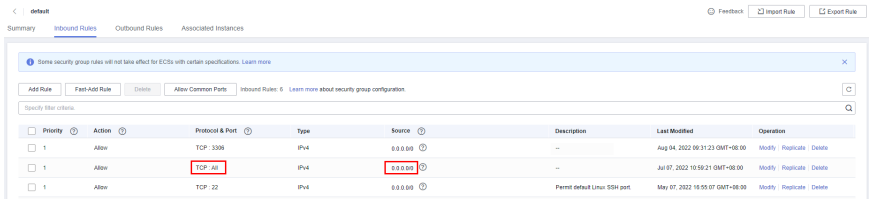
   telnet *EIP 1433*

   – 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 3-13 DB instance security group

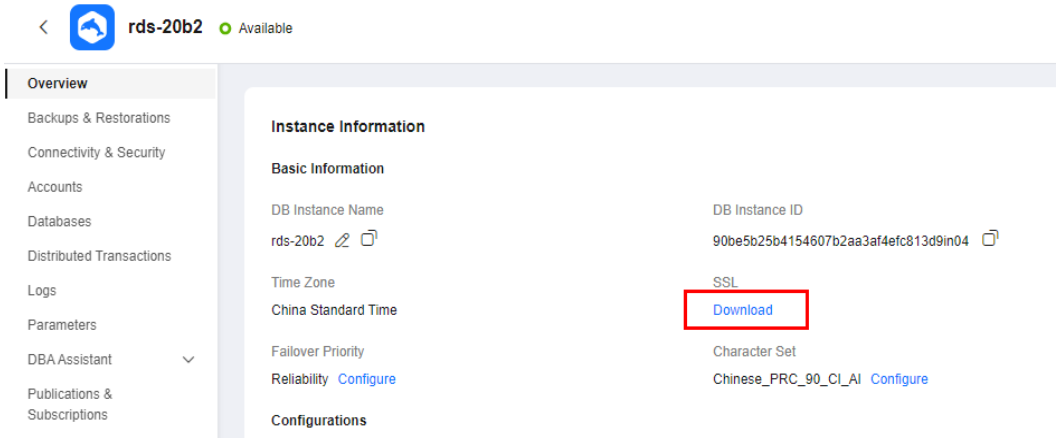
     

4. Open a browser on the local server, visit the **Microsoft website**, and download the installation package, for example, SQL Server Management Studio 18.0.
5. Double-click the installation package and complete the installation as instructed.

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 3-14 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 Do I Import the SSL Certificate of an RDS Instance to a Windows or Linux Server?**
4. Start SQL Server Management Studio.
5. Choose **Connect** > **Database Engine**. In the displayed dialog box, enter login information.

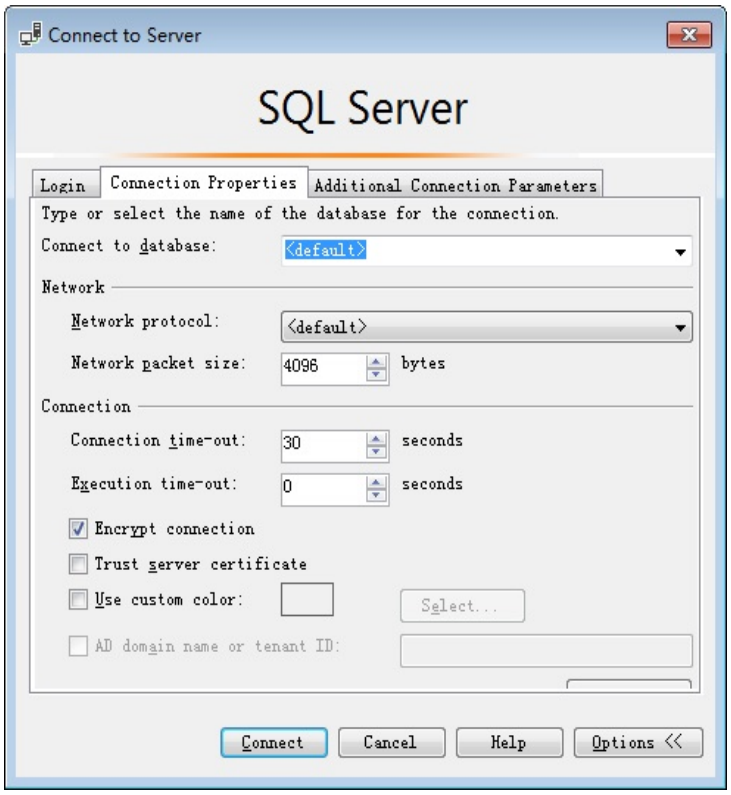
Figure 3-15 Connecting to the server

Table 3-3 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 3-16 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**

3.3.3 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 Database Migration

4.1 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 4-1 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**
	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**

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 SQL Server databases on other clouds to RDS for SQL Server.	From Microsoft SQL Server to RDS for SQL Server

4.2 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 Click in the upper left corner and select a region.

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 and click **Log In** in the **Operation** column.

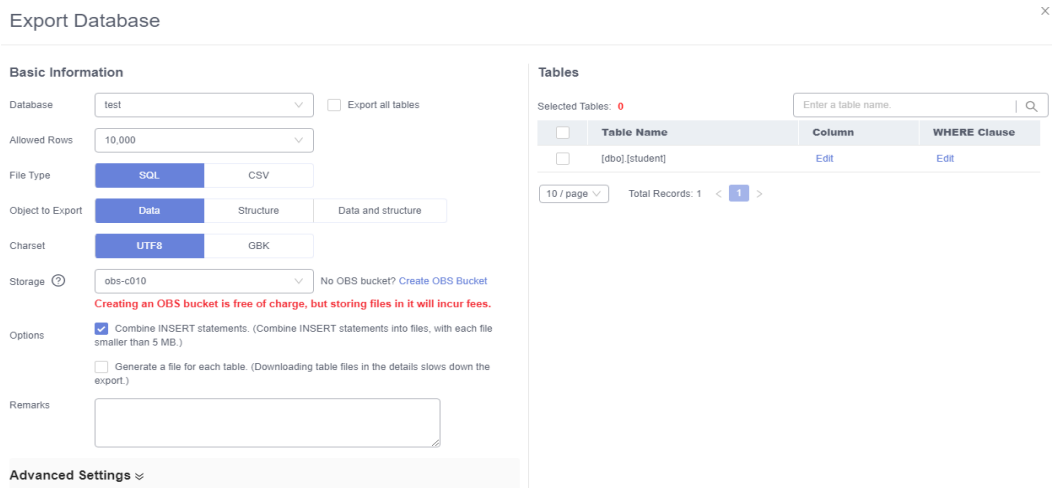
Step 4 On the displayed login page, enter the username and password and click **Log In**.

Step 5 On the top menu bar, choose **Import and Export** > **Export**.

Step 6 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.

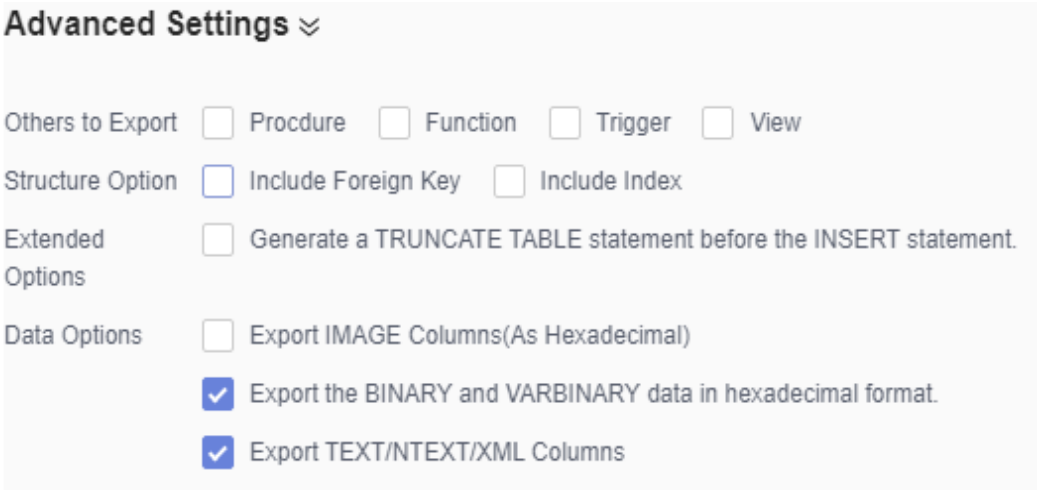
Alternatively, click **Quick Export** to export the specified database information quickly.

Figure 4-1 Creating an export task



Step 7 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 4-2 Advanced settings



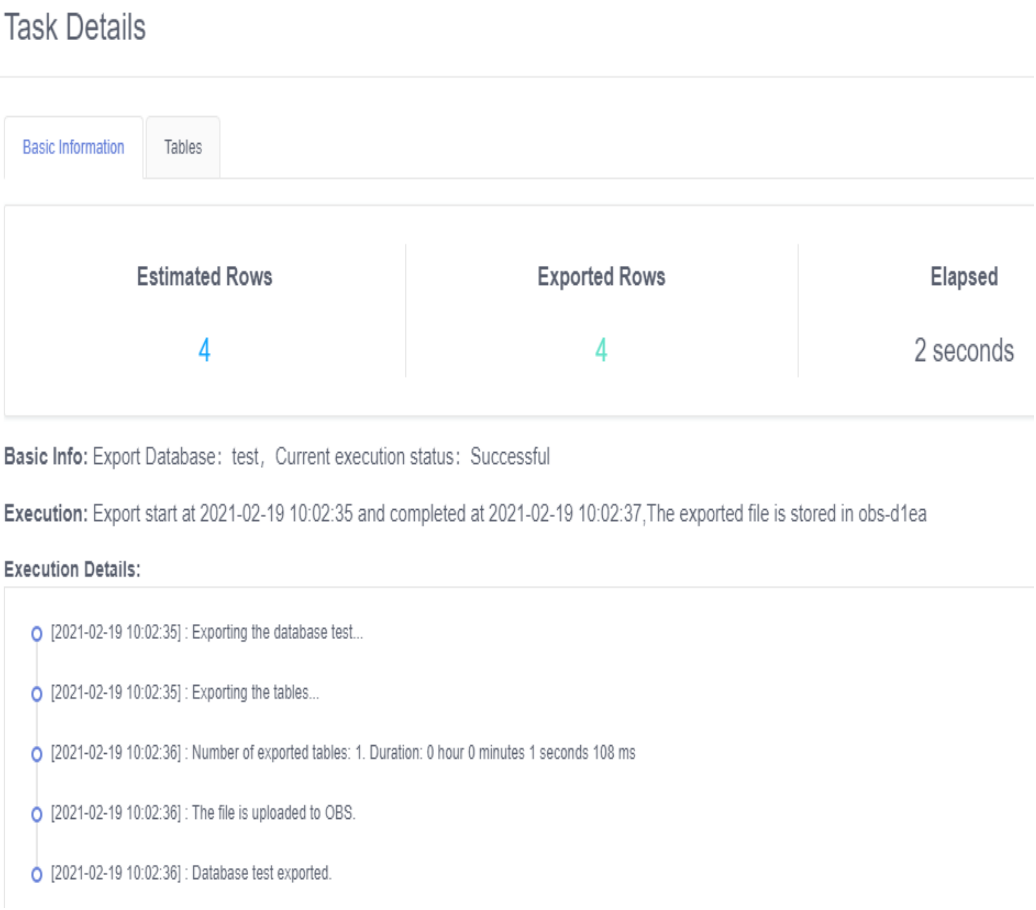
Step 8 After settings are complete, click **OK**.

Step 9 In the task list, view the task ID, type, status, and progress.

Figure 4-3 Task list

Step 10 Click **Details** in the **Operation** column to view task details.

Figure 4-4 Task details



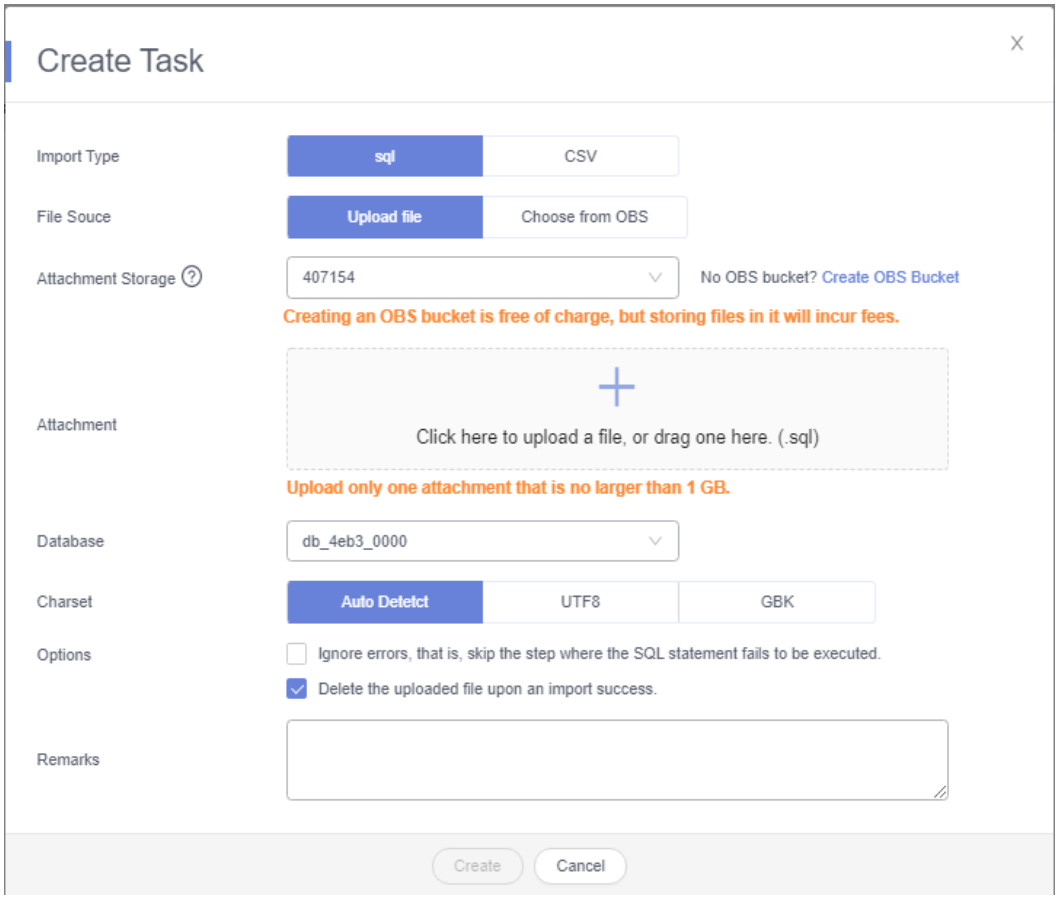
----End

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 4-5 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

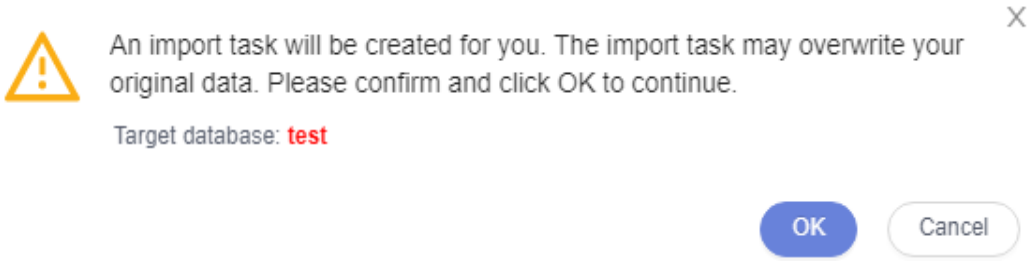
  In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

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 4-6 Confirmation page



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

----End

5 Performance Tuning

5.1 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.

5.2 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

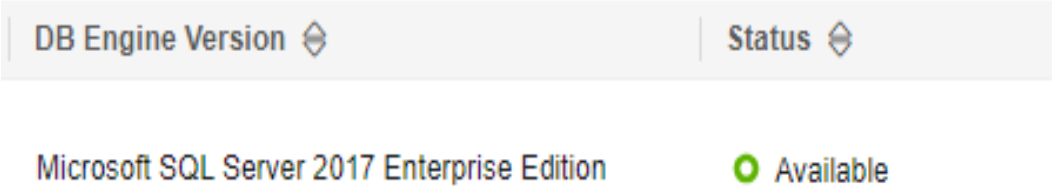
- **Scale up the storage.**
  a. In the instance list, choose **More** > **Scale Storage Space**.
  b. On the displayed page, the system determines the minimum space required.

  Figure 5-1 Scaling up storage space

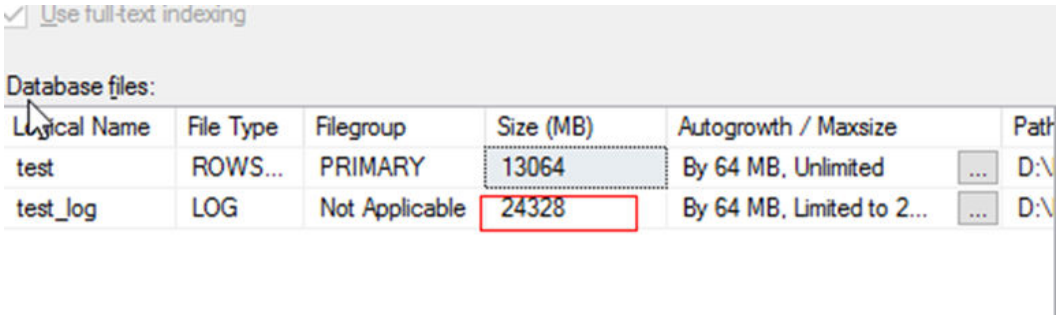
  c. After the scale-up is successful, check that the instance status becomes **Available**.

  Figure 5-2 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 5-3 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.

6 Using IAM to Grant Access to RDS

6.1 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.

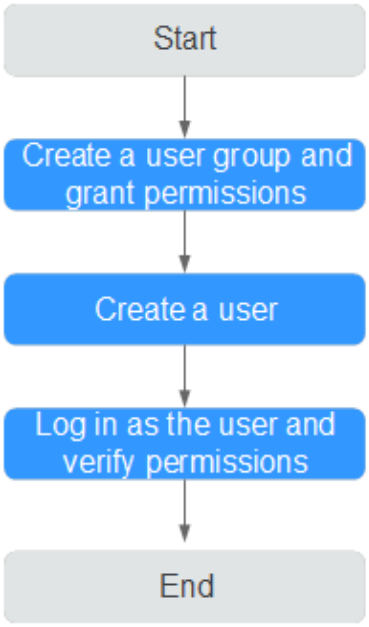
This section describes the procedure for granting permissions (see **Figure 6-1**).

Prerequisites

Before assigning permissions to user groups, you should learn about system-defined policies supported by RDS and select the policies based on service requirements. For details about the system-defined policies supported by RDS, see **System-defined permissions for RDS**. To grant permissions for other services, see **System-defined Permissions**.

Process Flow

Figure 6-1 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 console as the created user, switch to the region where it is authorized, and verify the permissions.

   - Choose **Relational Database Service** from the service list 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 from the service list. If a message appears indicating that you have insufficient permissions to access the service, the **RDS ReadOnlyAccess** policy has already taken effect.

6.2 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**. This 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"
   }]
}
```


7 Instance Lifecycle

7.1 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 Click in the upper left corner and select a region.

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** > **Buy Same DB Instance** in the **Operation** column.

Step 4 On the displayed page, the configurations are the same as those of the selected DB instance. You can change them as required. Then, click **Buy**.

Step 5 Confirm the instance specifications.

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

Step 6 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

7.2 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 a 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

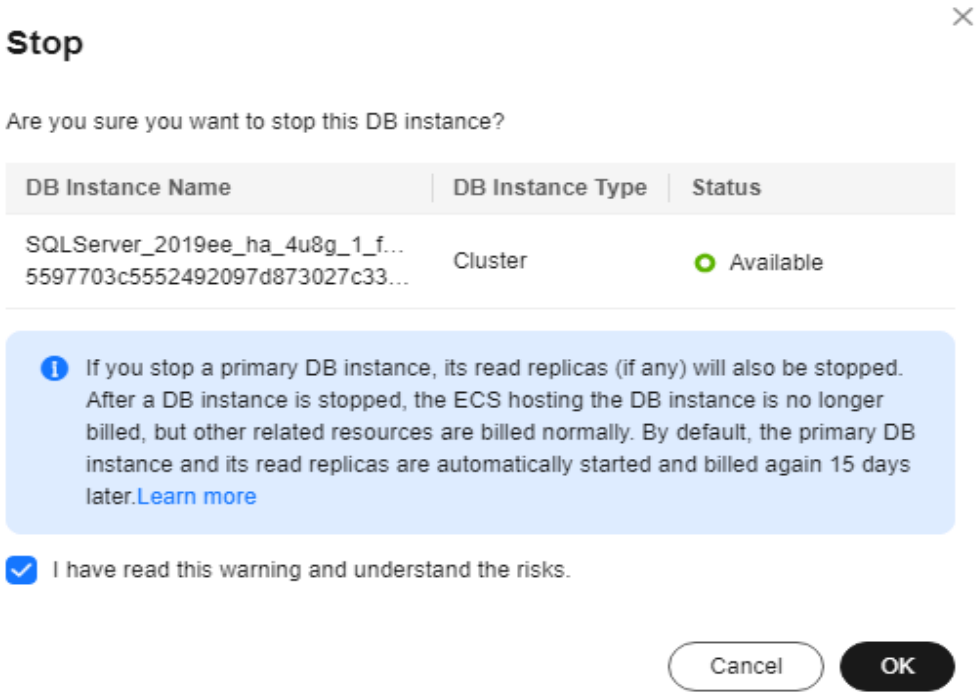
Step 1 Click  in the upper left corner and select a region.

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 instance that you want to stop and choose **More** > **Stop** in the **Operation** column.

Step 4 In the displayed dialog box, click **OK**.

Figure 7-1 Stopping an instance



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

----End

7.3 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

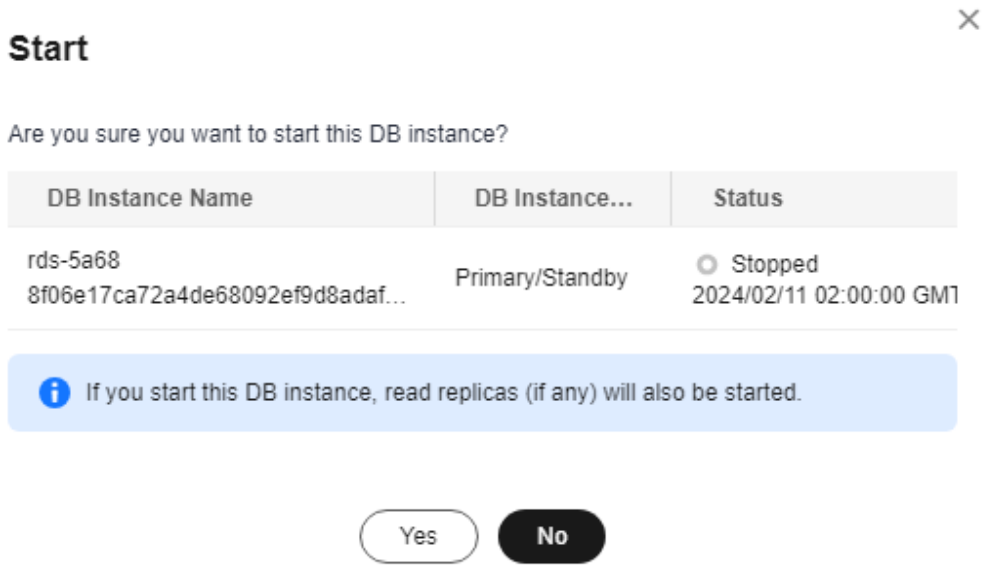
Step 1 Click in the upper left corner and select a region.

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 instance that you want to start and choose **More** > **Start** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

Figure 7-2 Starting an instance



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

----End

7.4 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 Click in the upper left corner and select a region.

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, 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**.

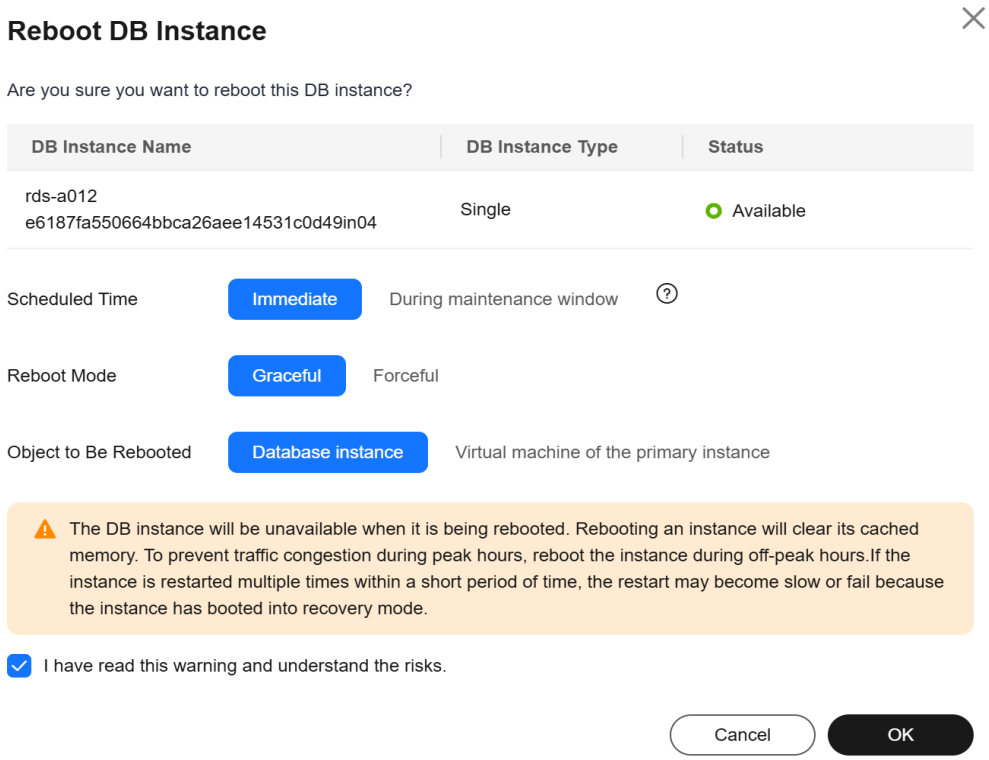
For primary/standby DB instances, if you reboot the primary DB instance, the standby DB instance is also rebooted automatically.

Step 4 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.

Figure 7-3 Rebooting a DB instance



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 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 Click in the upper left corner and select a region.

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 or read replicas (maximum: 50) to be rebooted and choose **More** > **Reboot** above the DB instance list.

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 and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

Step 6 Refresh the DB instance list and view the statuses of the DB instances. If their statuses are **Available**, they have rebooted successfully.

----End

7.5 Selecting Displayed Items

Scenarios

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

Procedure

Step 1 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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

7.6 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 Click  in the upper left corner and select a region.

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 is 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 Click  in the upper left corner and select a region.

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

7.7 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**.

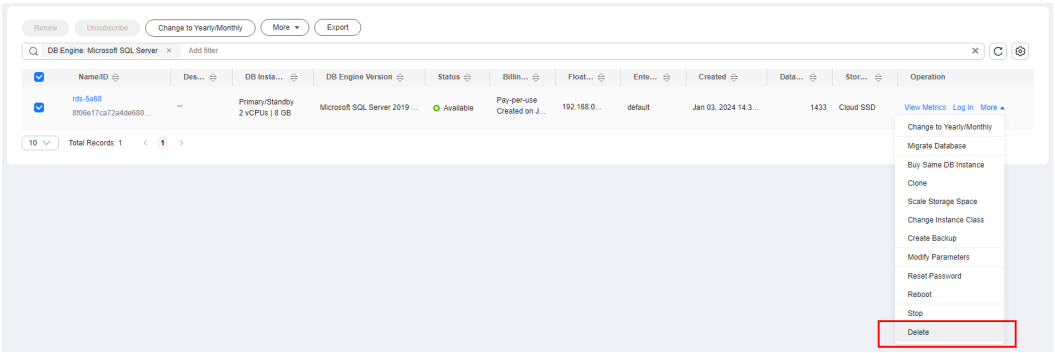
Deleting a Pay-per-Use DB Instance

Step 1 Click in the upper left corner and select a region.

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 7-4 Deleting a DB instance



Step 4 In the displayed dialog box, enter **DELETE** and click **OK**.

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

----End

Deleting a Pay-per-Use Read Replica

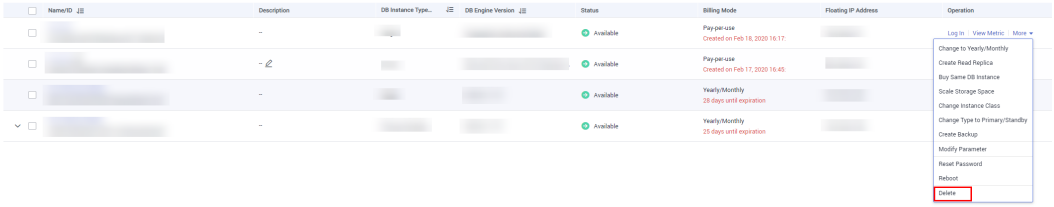
Step 1 Click in the upper left corner and select a region.

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 7-5 Deleting a read replica



Step 5 In the displayed dialog box, enter **DELETE** and 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***.

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

----End

7.8 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 Click in the upper left corner and select a region.

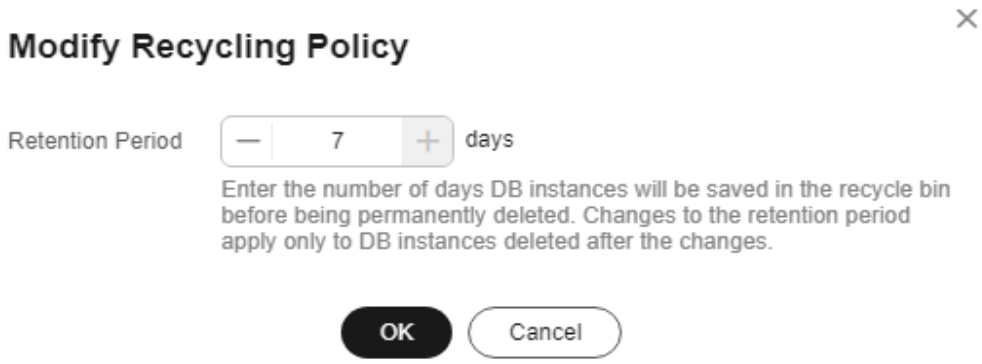
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 In the navigation pane on the left, choose **Recycle Bin**.

Step 4 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 5 Then, click **OK**.

Figure 7-6 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 Click in the upper left corner and select a region.

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

Step 5 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

8 Instance Modifications

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

8.2 Changing a DB Instance Description

Scenarios

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

Procedure

Step 1 Click in the upper left corner and select a region.

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 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 4 View the results on the **Overview** page.

----End

8.3 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 Always On 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

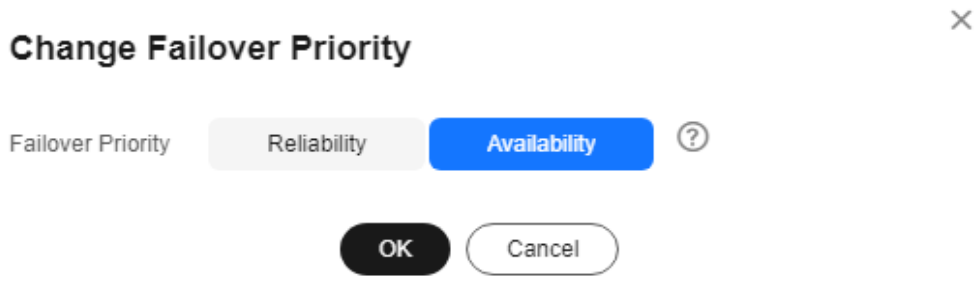
Step 1 Click in the upper left corner and select a region.

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 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 8-1 Changing the failover priority



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

----End

8.4 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 Click  in the upper left corner and select a region.

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 to be cloned and choose **More** > **Clone** in the **Operation** column.

Step 4 On the displayed page, keep the configurations the same as the original DB instance or change them as required. Then, click **Next**.

Step 5 Confirm the instance configurations.

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

Step 6 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

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.
- 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. 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 Click  in the upper left corner and select a region.

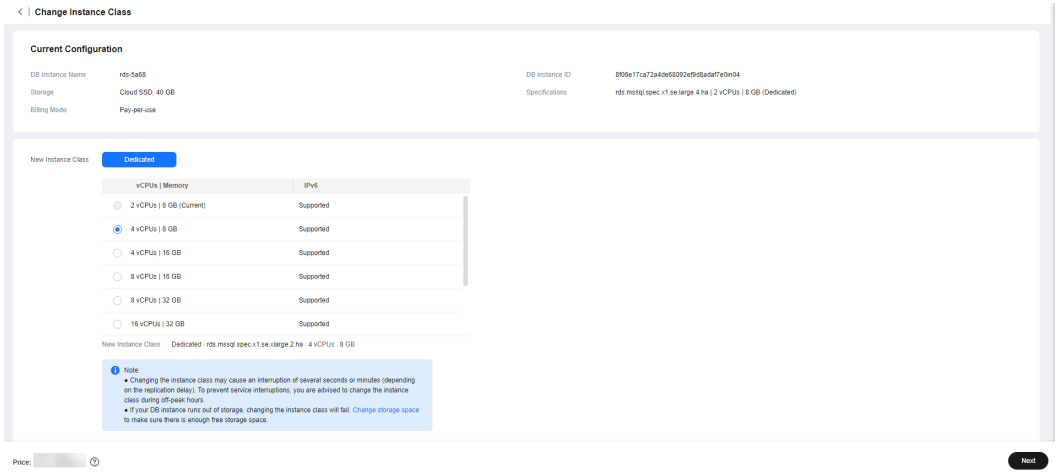
Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Change Instance Class** in the **Operation** column.

Alternatively, click the instance name to go to the **Overview** page. Under **Instance Class**, click **Configure**.

Step 4 On the displayed page, specify the new instance class and click **Next**.

Figure 8-2 Changing a DB instance class



To change the storage type to **Extreme SSD V2**, **submit a service ticket** to request required permissions. 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 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 Check the 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

8.6 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.

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 Click in the upper left corner and select a region.

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 scale up storage space in either of the following ways:

- Click the target instance name to enter 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 4 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 5 Confirm the information.

- 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 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

8.7 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

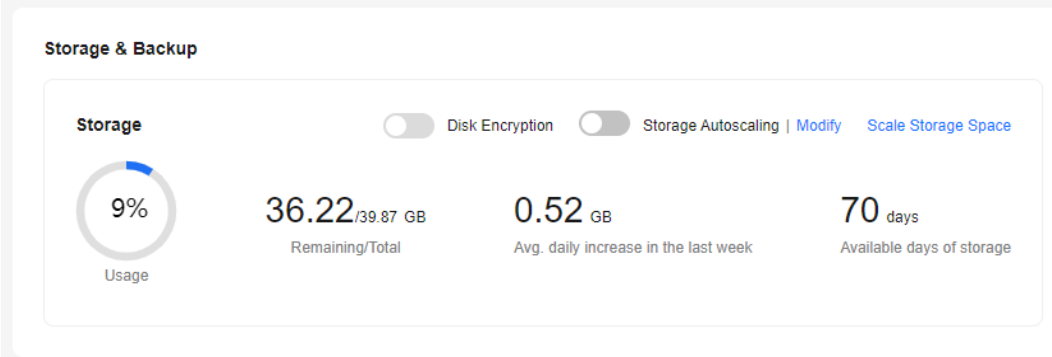
Step 1 Click  in the upper left corner and select a region.

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 name (click  in front of a DB instance to locate its read replica).

Step 4 In the **Storage & Backup** area, toggle on the **Storage Autoscaling** switch.

Figure 8-3 Storage space



Step 5 In the displayed dialog box, set the following parameters.

Figure 8-4 Configuring autoscaling

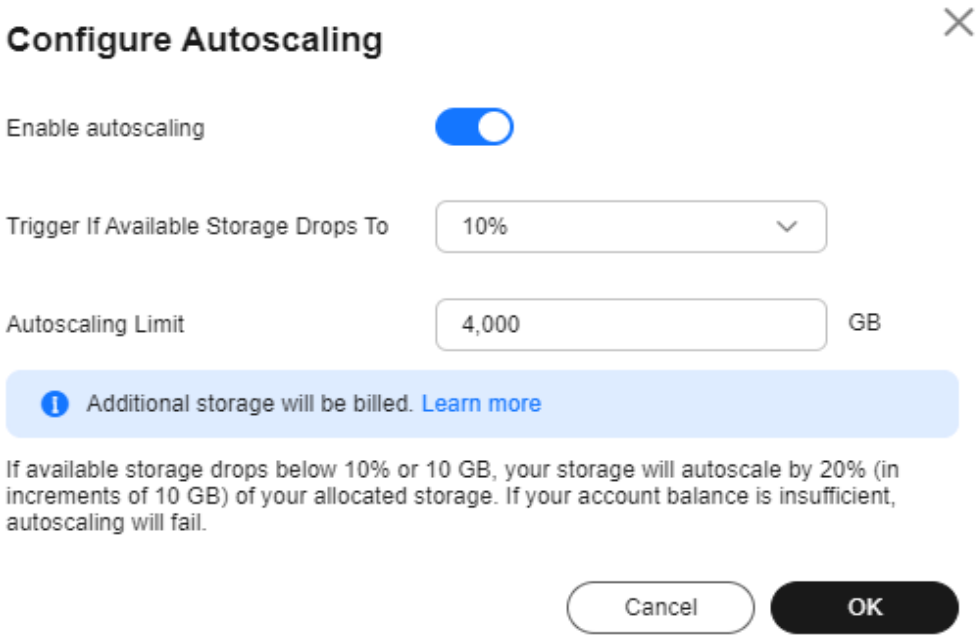


Table 8-1 Parameter description

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 to 10,000, in GB. The limit must be no less than the storage of the DB instance.

Step 6 Click **OK**.

----End

8.8 Changing a 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.

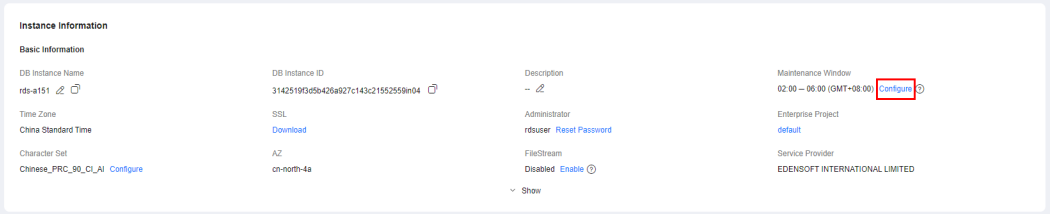
Procedure

Step 1 Click in the upper left corner and select a region.

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 instance name to go to the **Overview** page. Under **Maintenance Window**, click **Configure**.

Figure 8-5 Changing the maintenance window



Step 4 In the displayed dialog box, select a maintenance window and click **OK**.

NOTE

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

----End

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

Scenarios

- RDS enables you to change single-node 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-node 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-node DB instances.

Procedure

Step 1 Click  in the upper left corner and select a region.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, locate a single-node DB instance and choose **More** > **Change Type to Primary/Standby** in the **Operation** column.

Step 4 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 5 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

8.10 Manually Switching Between Primary and Standby DB Instances

Scenarios

If you choose to create a primary/standby DB instance, 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 Click  in the upper left corner and select a region.

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 instance name to go to the **Overview** page.

Step 4 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 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 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

8.11 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.

9 Read Replicas

9.1 Managing a Read Replica

Entering the Management Interface Through a Read Replica

Step 1 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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

10 Data Backups

10.1 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

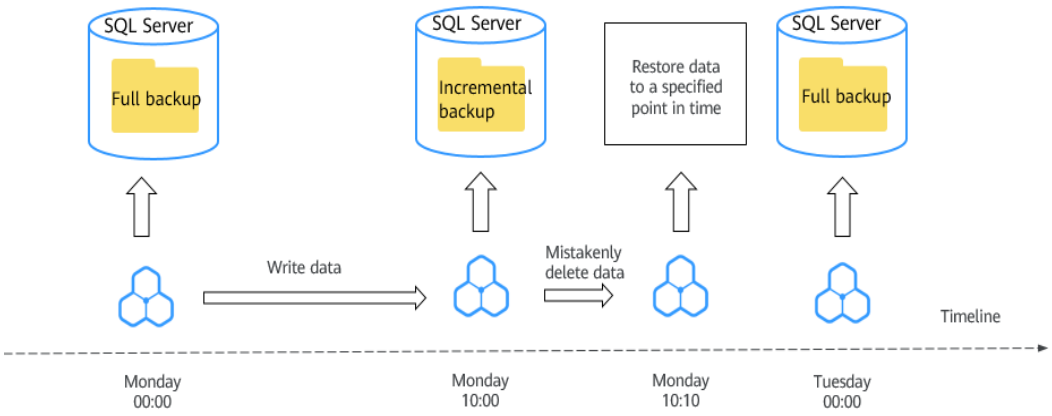
- Single-node instance

  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.

Figure 10-1 How RDS backs up data



Backup Solutions

Table 10-1 describes how to back up data and download backups.

Table 10-1 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.

10.2 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

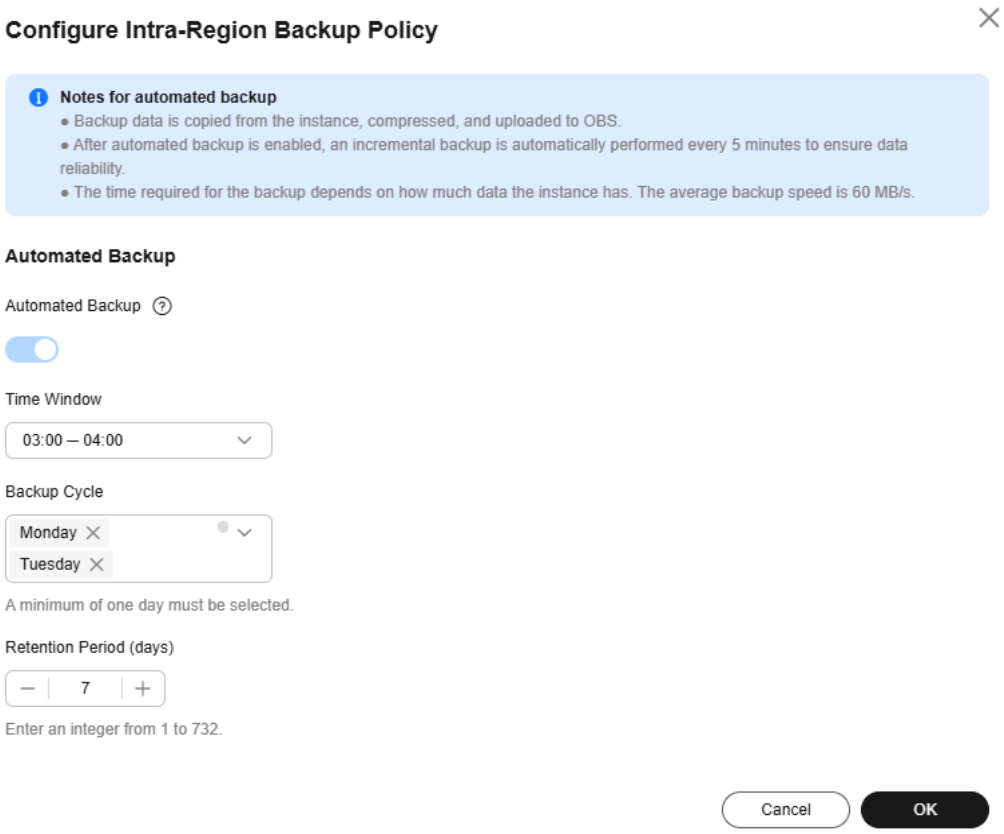
Step 1 Click  in the upper left corner and select a region.

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 **Intra-Region Backup Policies**. You can view the configured backup policy. To modify the backup policy, adjust the parameter values as needed.

Figure 10-2 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 5 Click **OK**.

----End

10.3 Creating a Manual Backup

Scenarios

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

NOTE

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

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 Click  in the upper left corner and select a region.

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, 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 5 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 Click in the upper left corner and select a region.

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, select a target database for which to create the backup, and enter a description. Then, 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 manual backup depends on the amount of data.

NOTE

System databases are backed up by default.

Step 5 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

10.4 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 Click  in the upper left corner and select a region.

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.

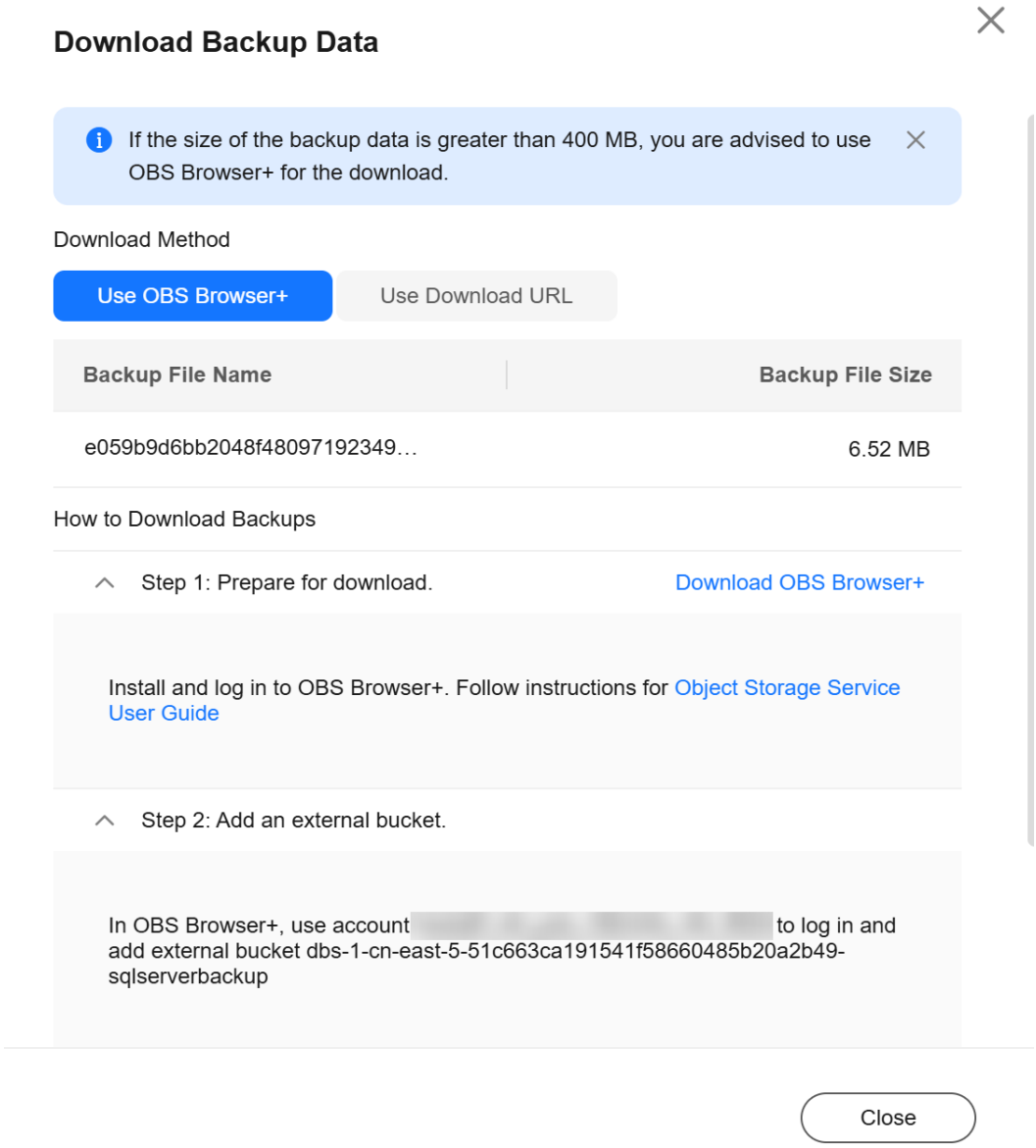
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

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***.

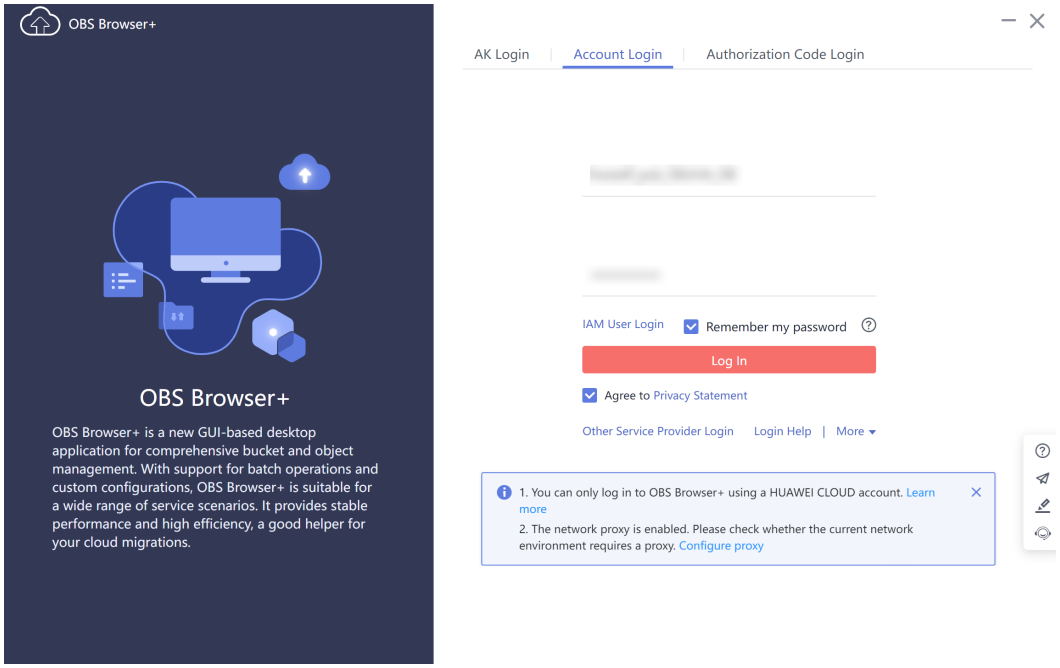
Step 5 In the displayed dialog box, select **Use OBS Browser+** for **Download Method** and download the backup as prompted.

Figure 10-3 Using OBS Browser+



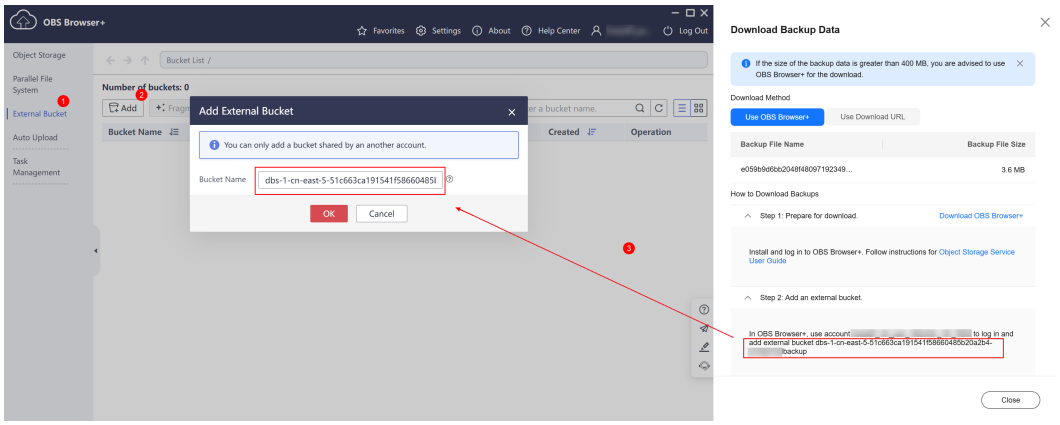
1. Download OBS Browser+ by clicking **Download OBS Browser+** in Step 1 on the download guide page.
2. Decompress and install OBS Browser+.
3. Log in to OBS Browser+ using the username provided in step 2 on the download guide page.

Figure 10-4 Logging in to OBS Browser+



4. Add an external bucket using the bucket name provided in step 2 on the download guide page.

Figure 10-5 Adding an external bucket



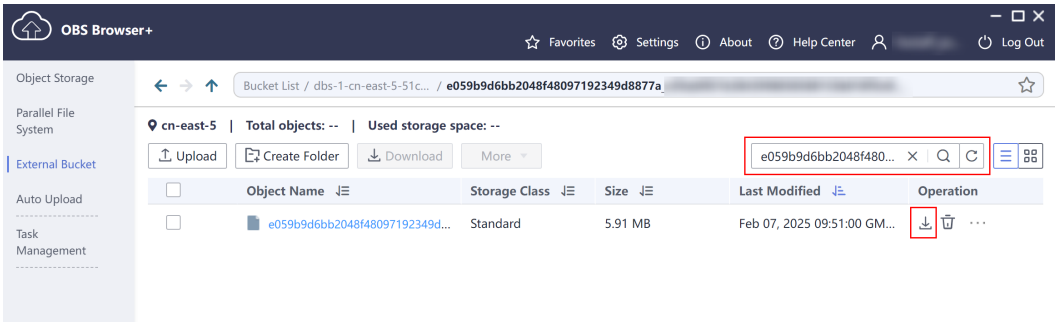
NOTE

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 10-6 Downloading a backup



Microsoft SQL Server allows you to download backup files of a specific database.

----End

Method 2: Using Download URL

Step 1 Click in the upper left corner and select a region.

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.

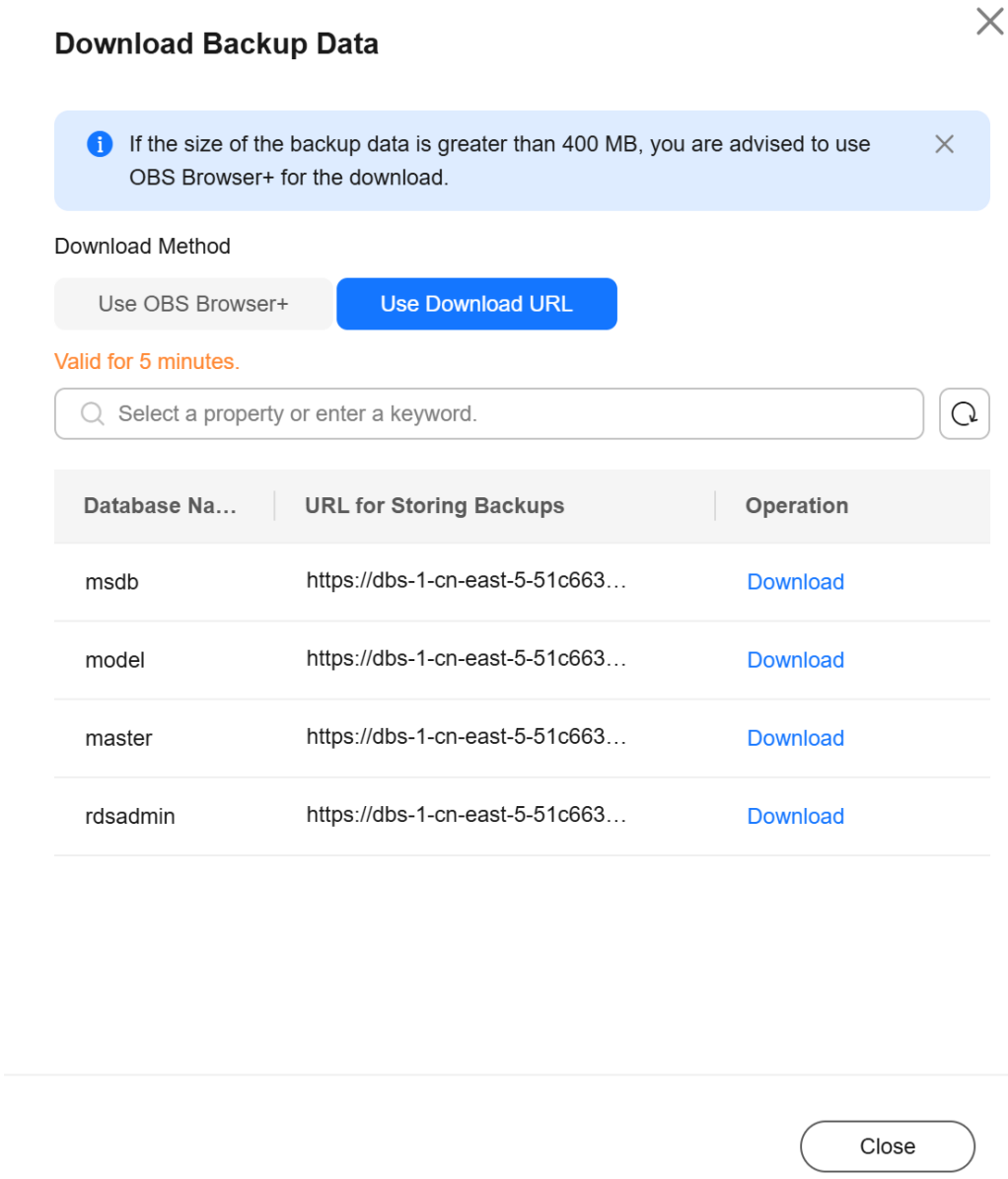
Alternatively, click the target DB instance. In the navigation pane on the left, choose **Backups & Restorations**. On the displayed page, locate the target backup to be downloaded and click **Download** in the **Operation** column.

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 a method to download backup data.

Figure 10-7 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

10.5 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 Click in the upper left corner and select a region.

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.

Figure 10-8 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 4 View the exported backup information.

----End

10.6 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 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**.

Procedure

Step 1 Click in the upper left corner and select a region.

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.

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

10.7 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 Click  in the upper left corner and select a region.

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 and cloud product. For details about how to enable operation protection, see ***Identity and Access Management User Guide***.

----End

11 Data Restorations

11.1 Restoration Solutions

If your database is damaged or mistakenly deleted, you can restore it from backup.

Table 11-1 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

11.2 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 Click  in the upper left corner and select a region.

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 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 4 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 5 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?

11.3 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.

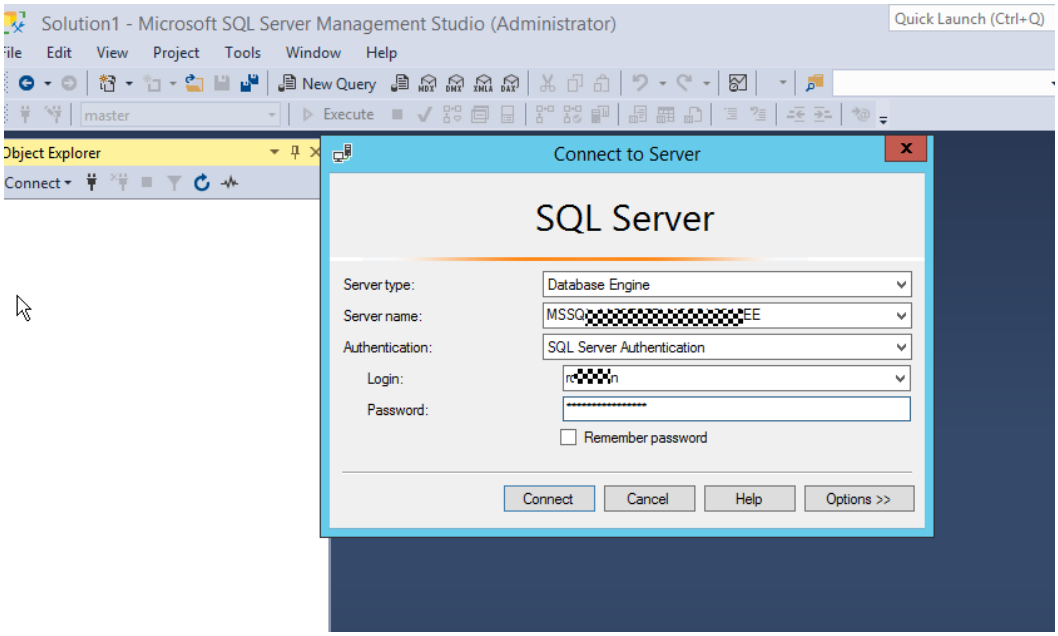
Prerequisites

You have downloaded the .bak files from the cloud to a local path of a self-managed database.

Restoring a Data Backup

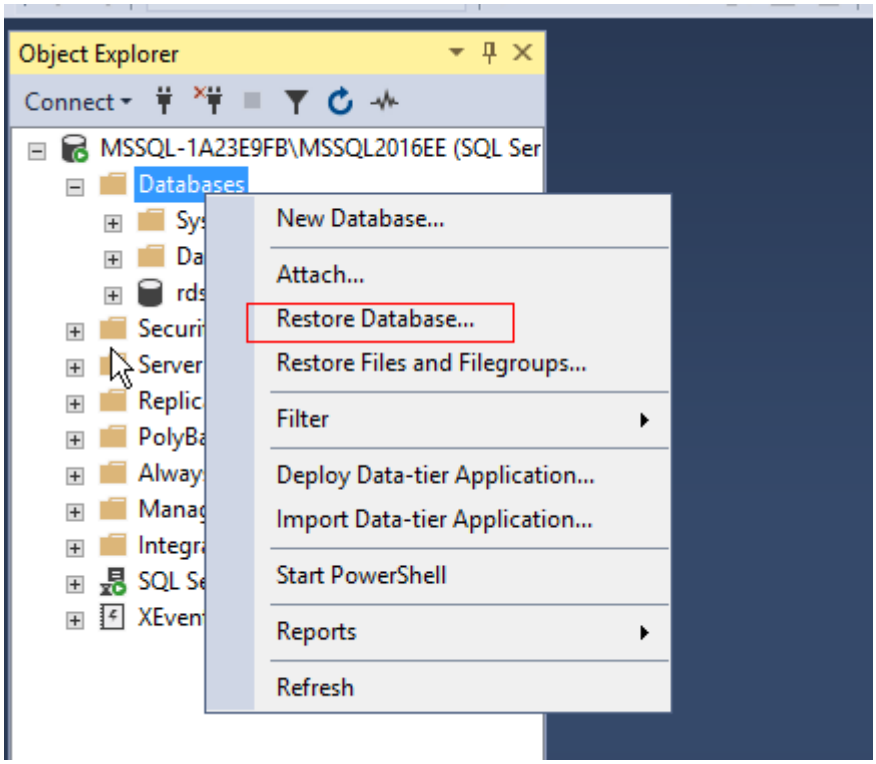
Step 1 Use the Microsoft official tool SQL Server Management Studio (SSMS) to log in to a self-managed database.

Figure 11-1 Logging in to a self-managed database



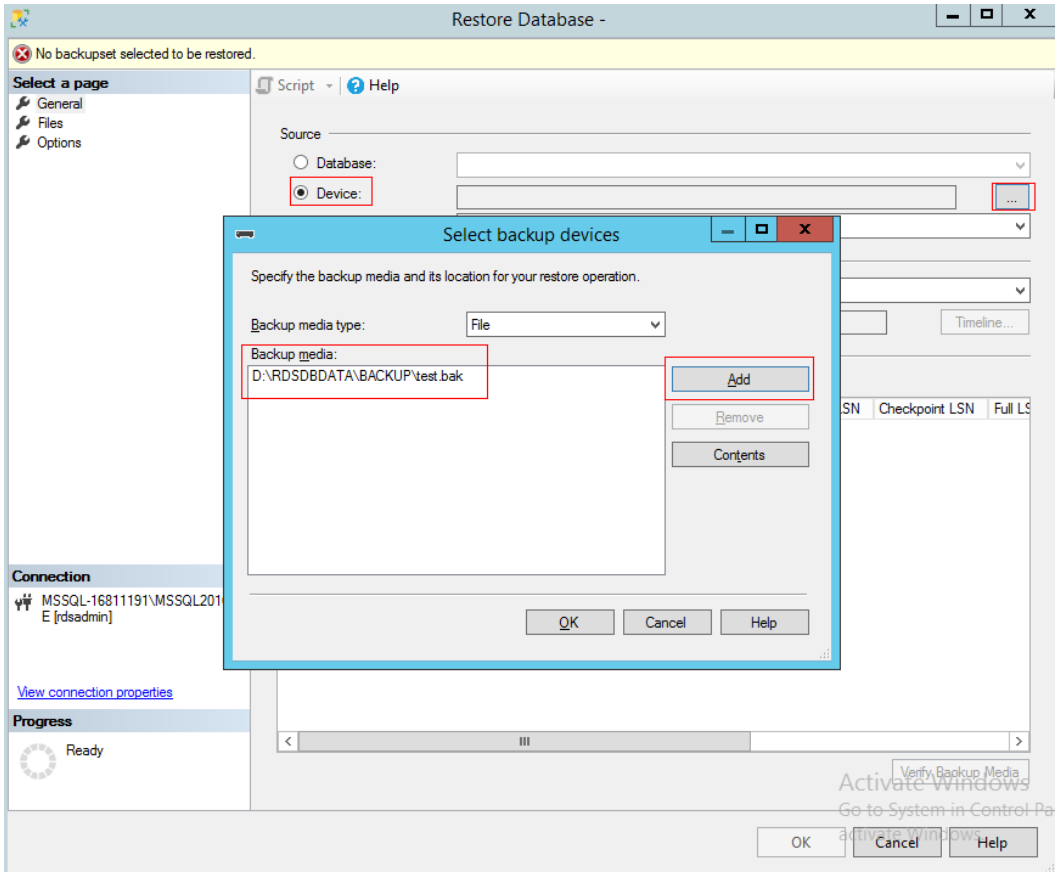
Step 2 Right-click **Databases**, and choose **Restore Database** from the shortcut menu.

Figure 11-2 Selecting a database



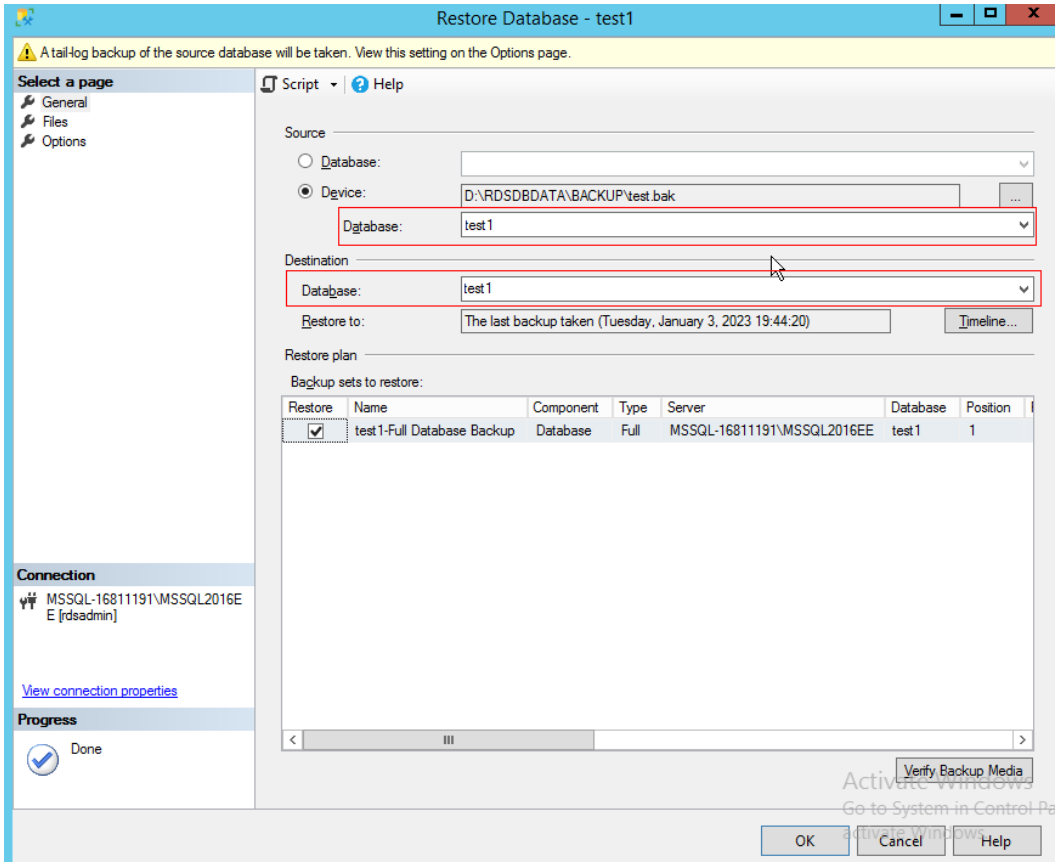
Step 3 Select **Device**, add the .bak backup file, and click **OK**.

Figure 11-3 Adding a backup file



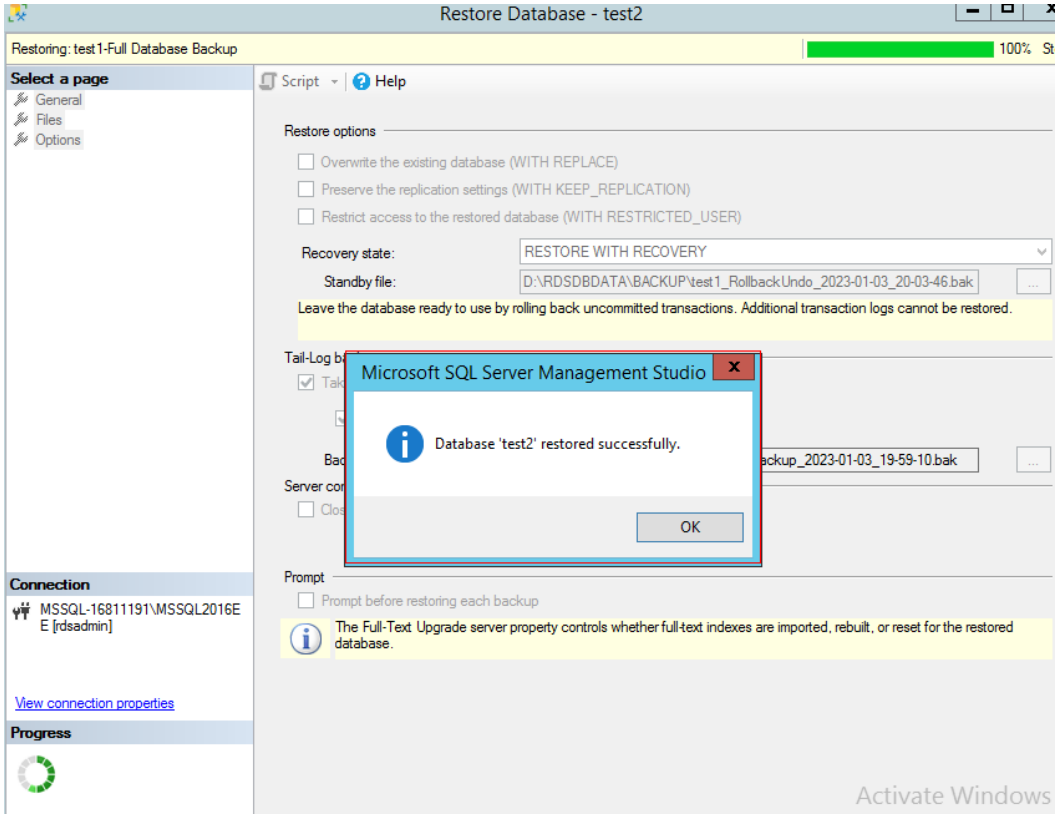
Step 4 Select the database to be restored. You can select the source database from the **Database** drop-down list box in the **Source** area and change the name of the destination database in the **Destination** area.

Figure 11-4 Selecting the source and destination databases



Step 5 Click **OK**.

Figure 11-5 Successful restoration



----End

Restoring Incremental Backups (Log Backups)

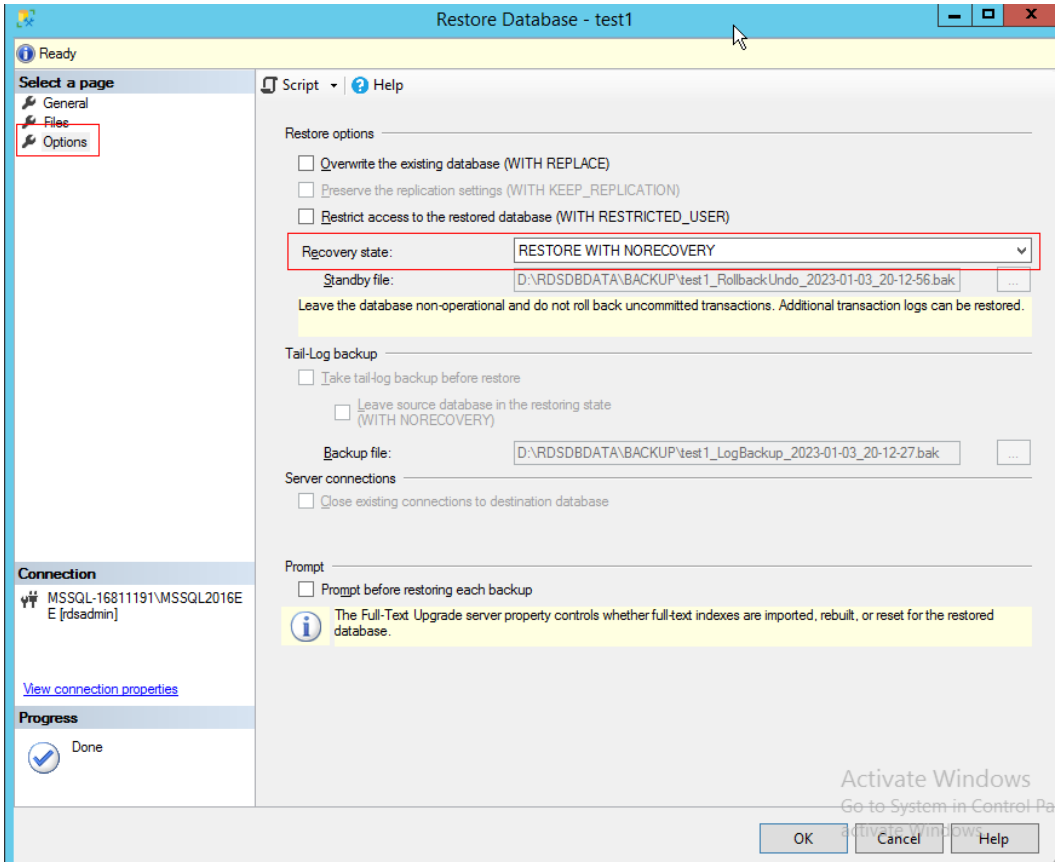
NOTE

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**.

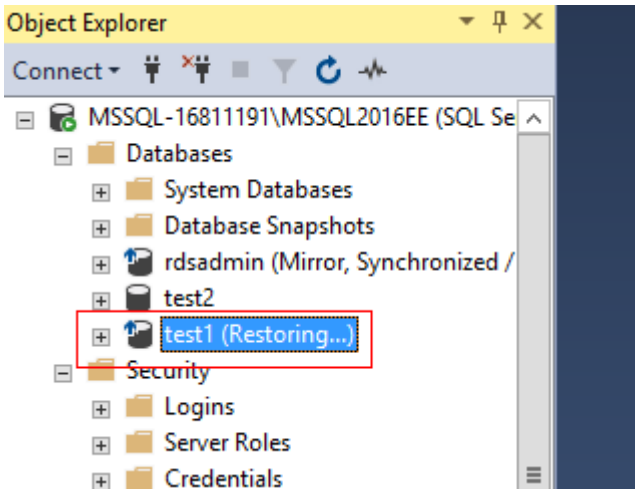
Step 2 Click **Option** and set **Recovery state** to **RESTORE WITH NORECOVERY**.

Figure 11-6 Setting Recovery state



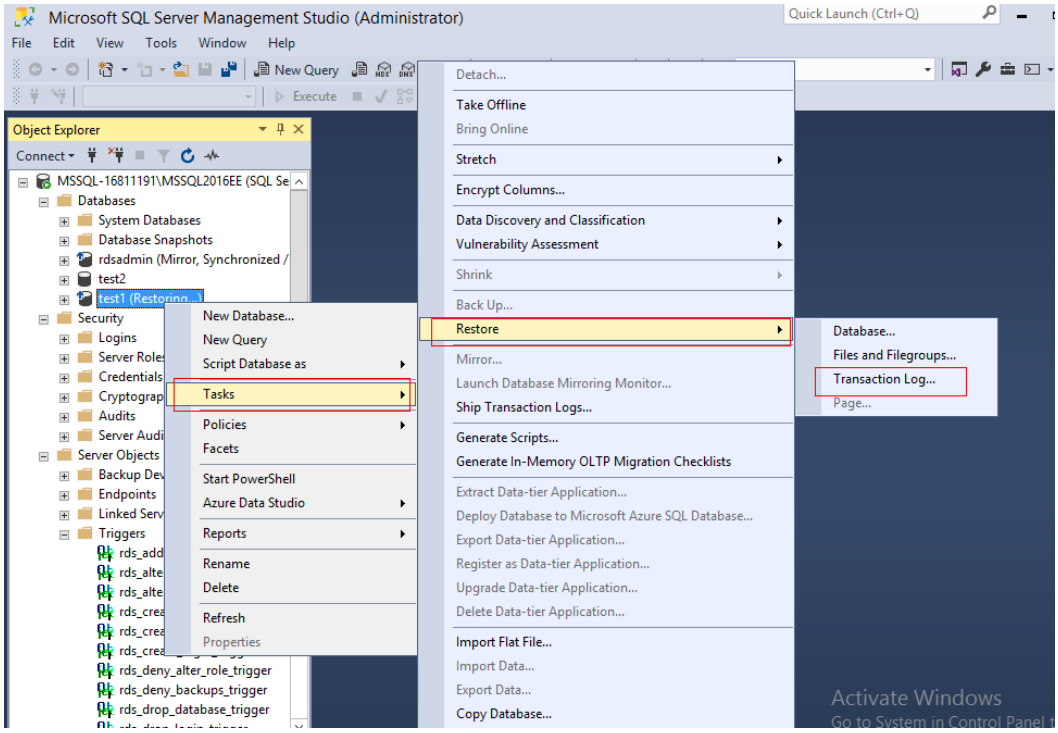
Step 3 Check that the database status is **Restoring**.

Figure 11-7 Checking the restoration status



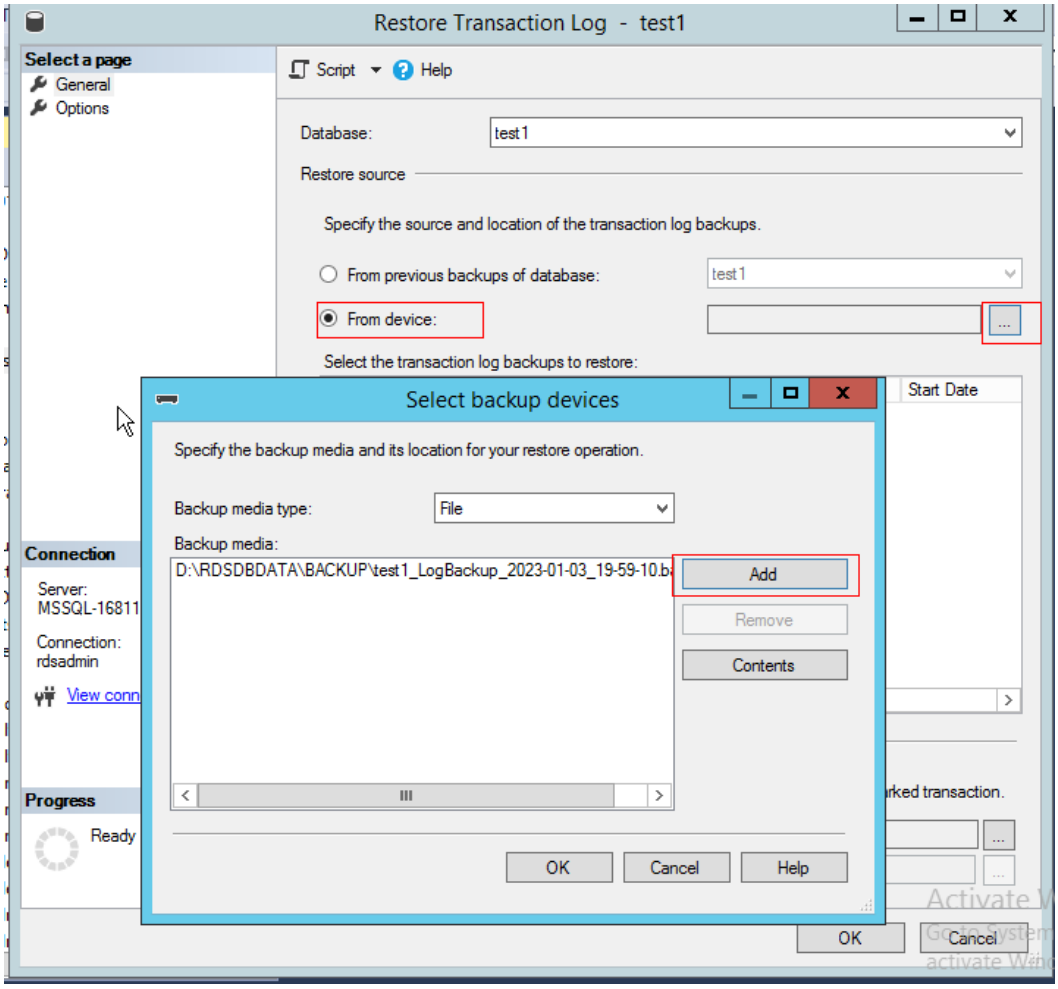
Step 4 Right-click the database and choose **Tasks** > **Restore** > **Transaction Log** from the shortcut menu.

Figure 11-8 Selecting a database



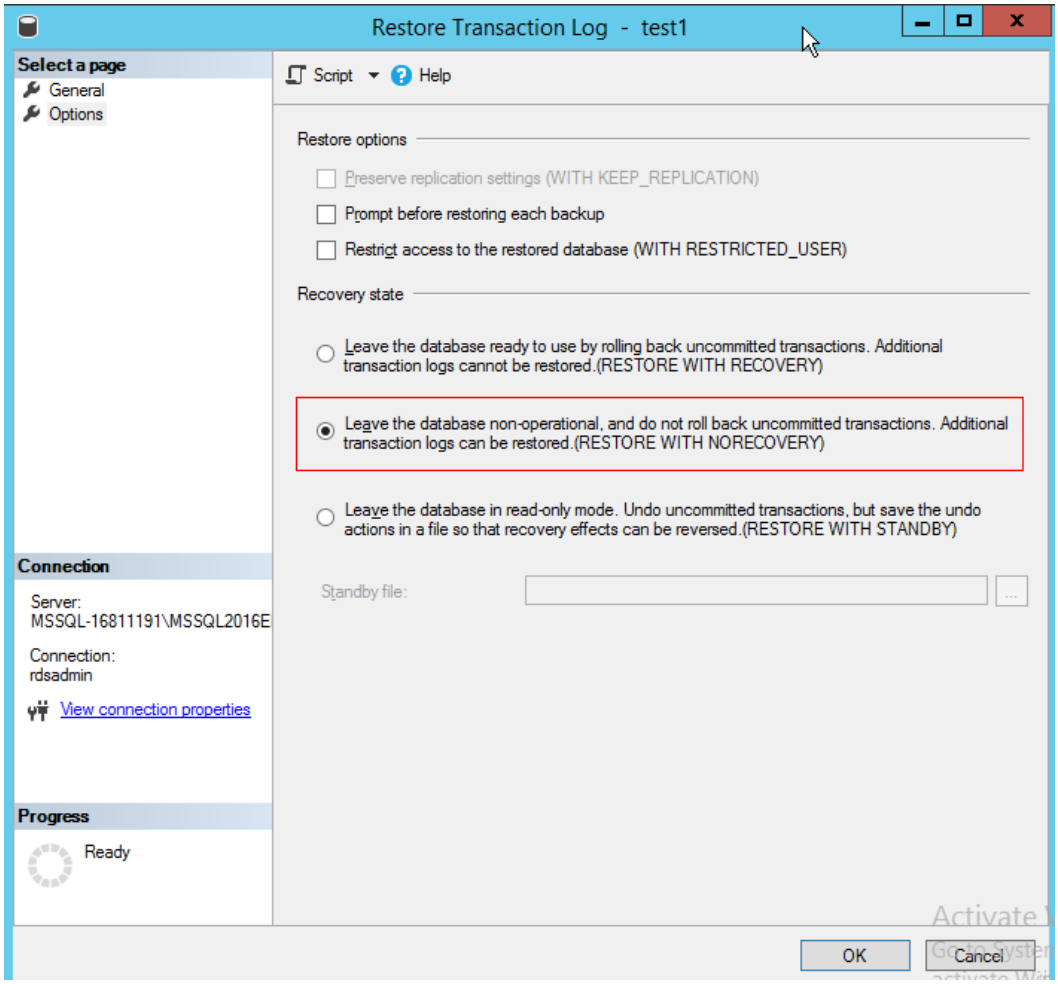
Step 5 Select **From device** and add the backup file to be restored.

Figure 11-9 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 11-10 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

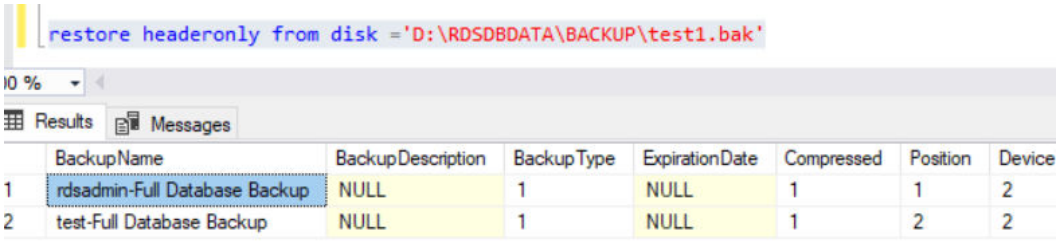
FAQ

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:

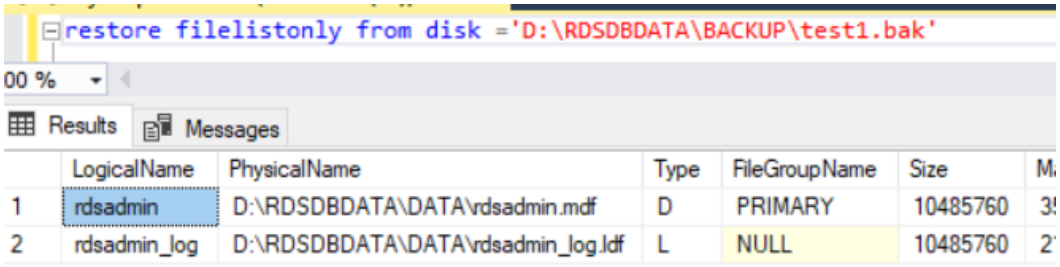
1. The downloaded backup file contains two databases. The first database is **rdsadmin**, and the second database is the target database, for example, **test**.
2. Query backup file header information.
   ```
   restore headeronly from disk='Local path of the .bak file'
   ```


Figure 11-11 Querying backup file information



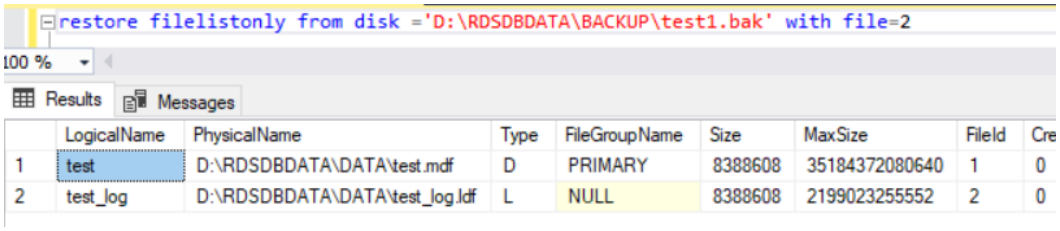
3. Query information about the databases that were backed up.
   restore filelistonly from disk='*Local path of the .bak file*'
   By default, only information about the first database is read.

Figure 11-12 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 11-13 Querying information about other databases



5. Restore data.

Figure 11-14 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.

11.4 PITR: 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 Click  in the upper left corner and select a region.

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 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 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.

FAQs

How Can I Restore Data If No Backup Is Available?

12 Parameters

12.1 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 Click  in the upper left corner and select a region.

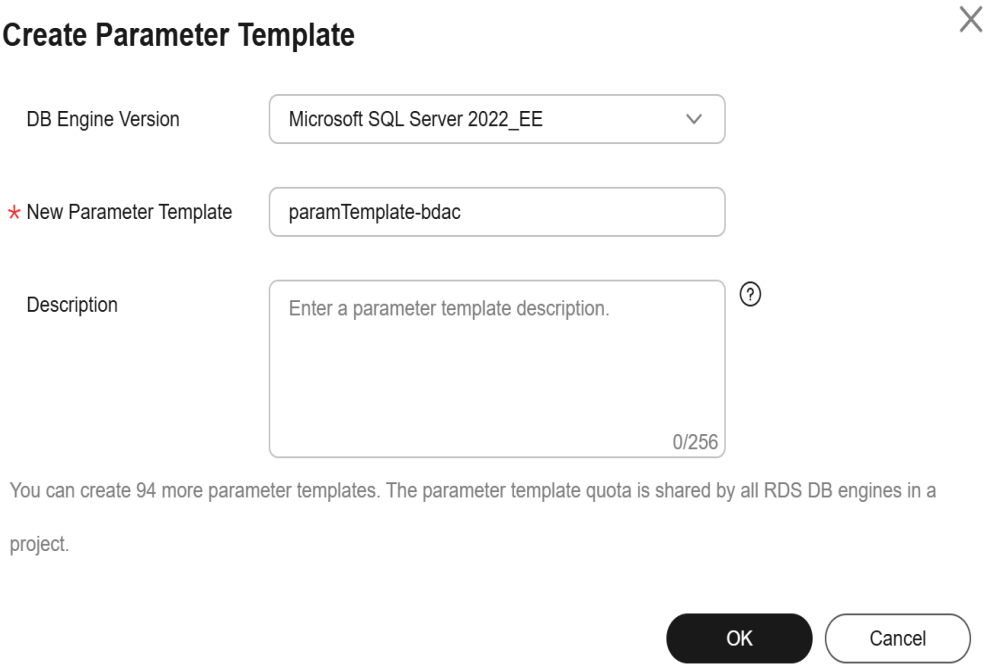
Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Parameter Templates** page, click **Create Parameter Template**.

Step 4 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 12-1 Creating a parameter template



----End

12.2 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the **Parameters** page, modify parameters as required.

Relevant parameters are as follows:

- Set **remote access** to **0** (default value) to prevent local stored procedures from running on a remote server and remote 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 5 After the parameter values are modified, you can click **Change History** to view the modification details.

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 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 Click  in the upper left corner and select a region.

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, modify parameters as required.

Relevant parameters are as follows:

- Set **remote access** to **0** (default value) to prevent local stored procedures from running on a remote server and remote 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 12-1 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 an RDS for SQL Server Instance Times Out?**

12.3 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 Click in the upper left corner and select a region.

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 instance name.

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**.

  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

12.4 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

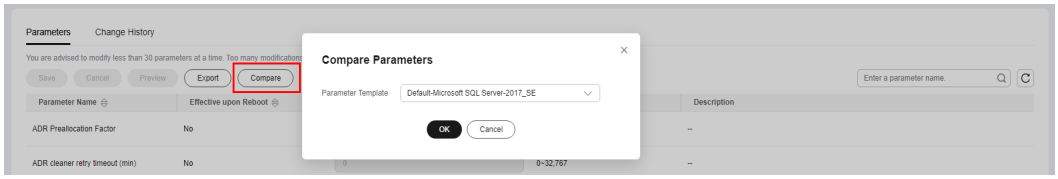
Step 1 Click in the upper left corner and select a region.

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 **Parameters**. On the displayed page, click **Compare** above the parameter list.

Figure 12-2 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

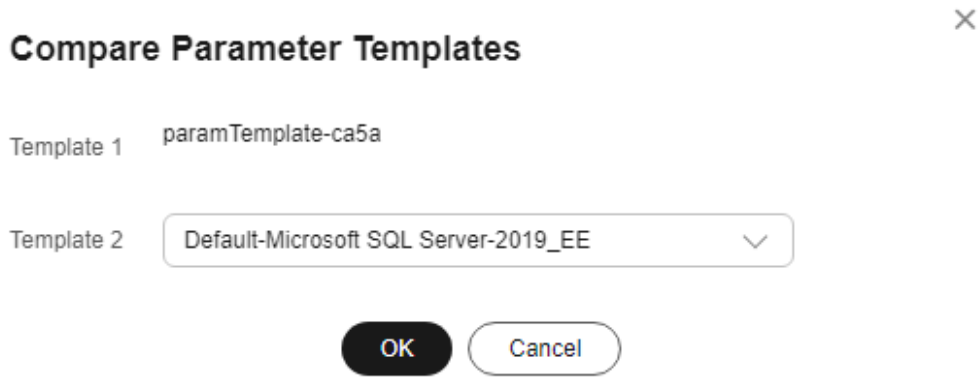
Step 1 Click in the upper left corner and select a region.

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 **Custom Templates**. Locate the target parameter template and click **Compare** in the **Operation** column.

Step 4 In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.

Figure 12-3 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

12.5 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

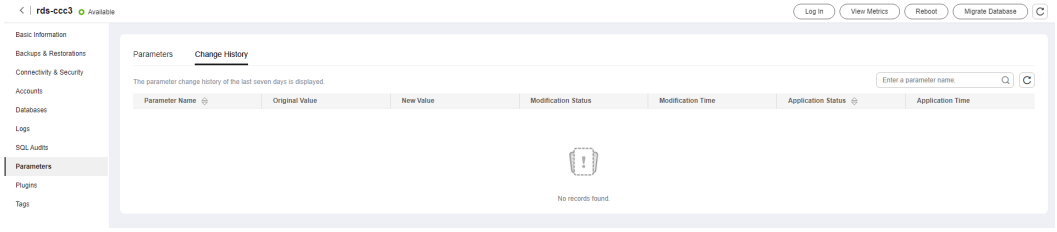
Step 1 Click in the upper left corner and select a region.

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 instance name.

Step 4 In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change History**.

Figure 12-4 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

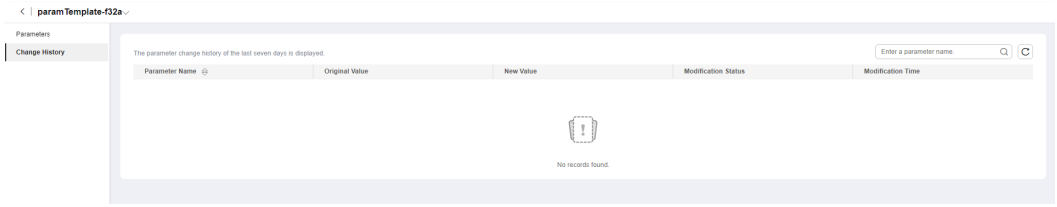
Step 1 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the displayed page, choose **Change History** in the navigation pane on the left.

Figure 12-5 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

12.6 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 Click  in the upper left corner and select a region.

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 **Custom Templates**. Locate the target parameter template and click **Replicate** in the **Operation** column.

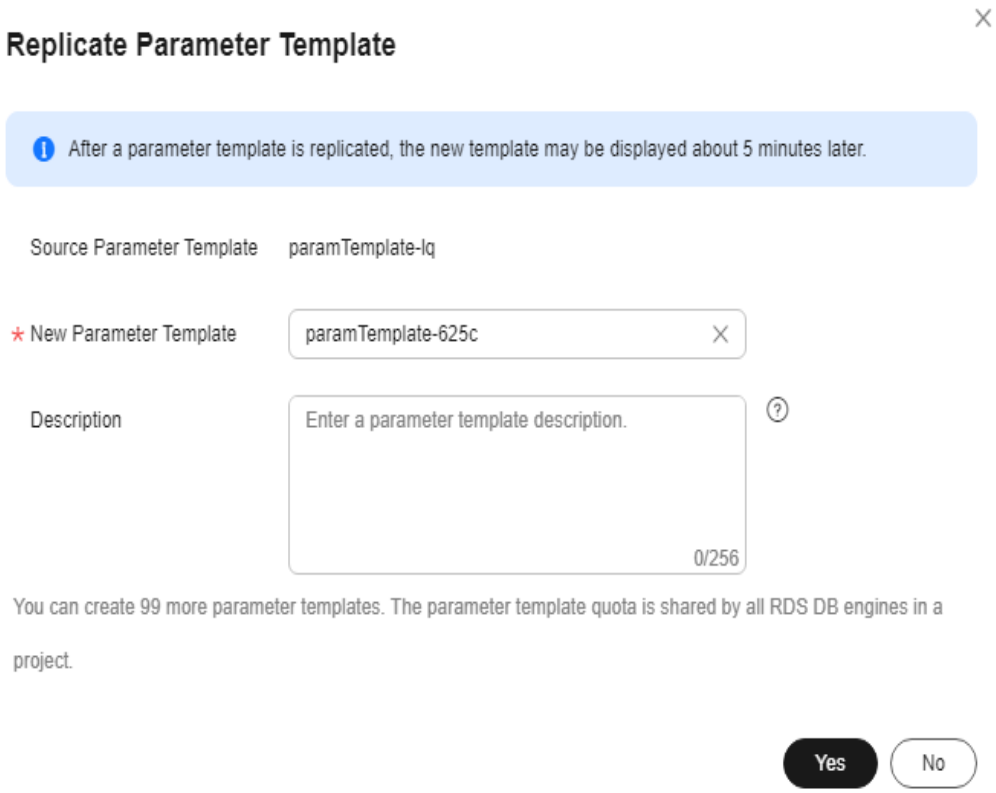
Alternatively, click the target DB instance on the **Instances** page. On the **Parameters** page, click **Export** to generate a new parameter template for future use.

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

Figure 12-6 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

12.7 Resetting a Parameter Template

Scenarios

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

Procedure

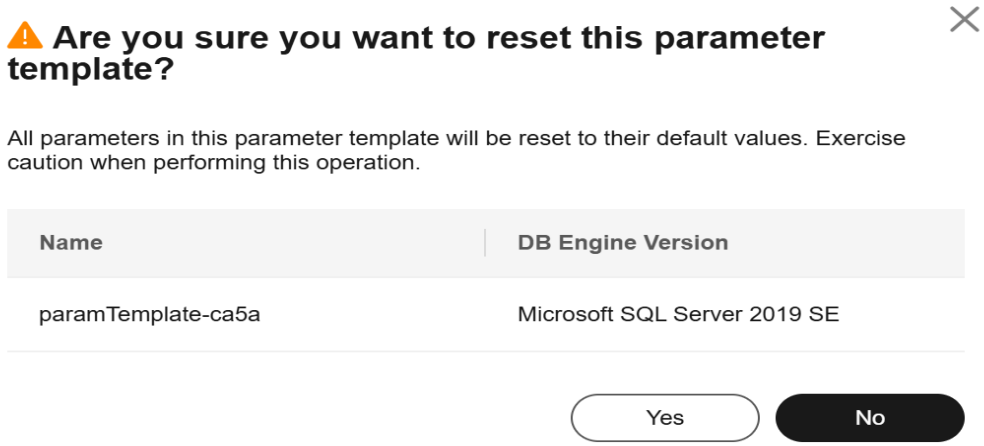
Step 1 Click in the upper left corner and select a region.

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 **Custom Templates**. Locate the target parameter template and choose **More** > **Reset** in the **Operation** column.

Step 4 Click **Yes**.

Figure 12-7 Confirming the reset



Step 5 Apply the parameter template to your DB instance. For details, see **Applying a Parameter Template**.

Step 6 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

12.8 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 4 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

12.9 Viewing Application Records of a Parameter Template

Scenarios

You can view the application records of a parameter template.

Procedure

Step 1 Click in the upper left corner and select a region.

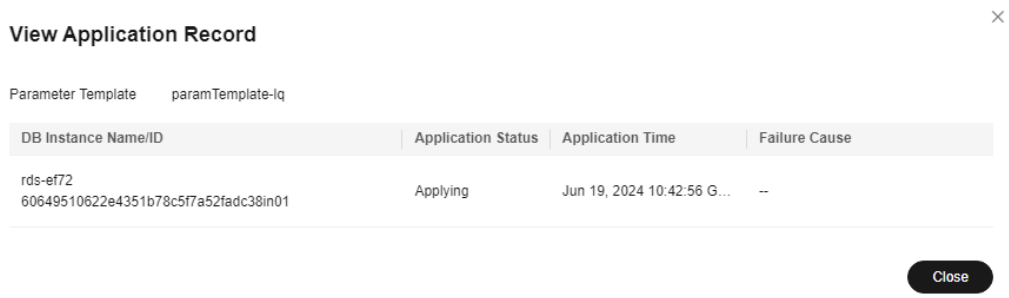
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left.

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 applies to, as well as the application status, application time, and failure cause (if failed).

Figure 12-8 Viewing application records of a parameter template



----End

12.10 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 Click in the upper left corner and select a region.

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 **Custom Templates**. Locate the target parameter template and click in the **Description** column.

Step 4 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

12.11 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 Click  in the upper left corner and select a region.

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 **Custom Templates**. Locate the target parameter template to be deleted and choose **More** > **Delete** in the **Operation** column.

Step 4 In the displayed dialog box, click **Yes**.

----End

13 Connection Management

13.1 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

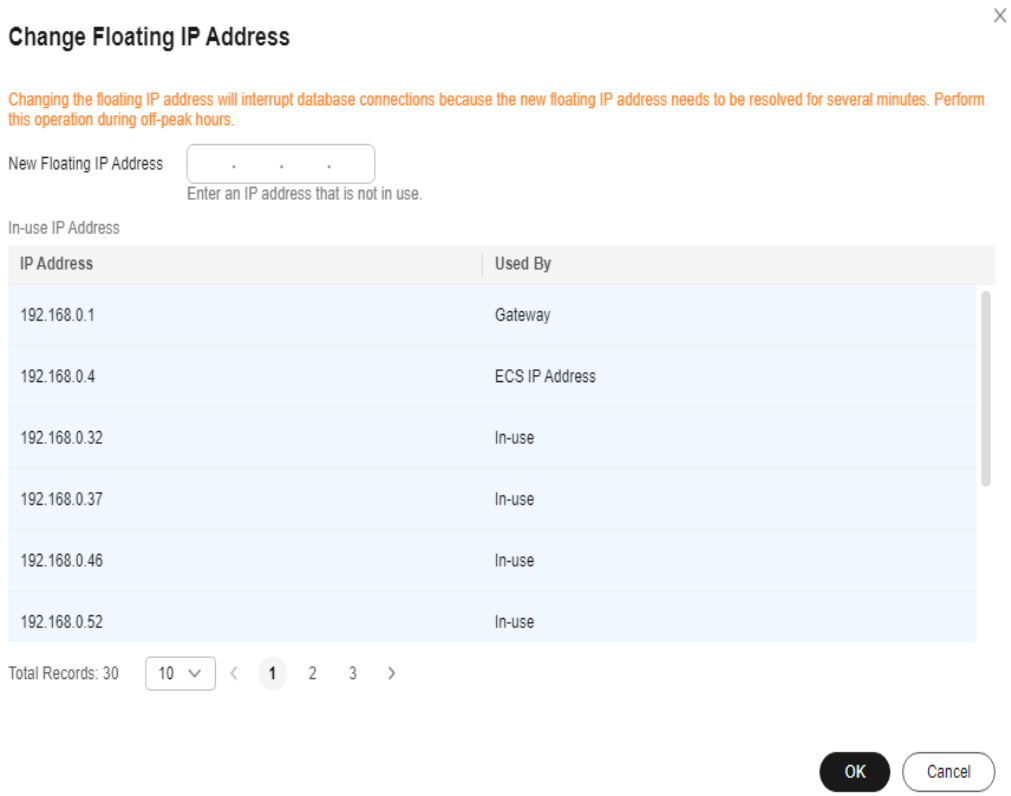
Step 3 On the **Instances** page, click the instance name to go to the **Overview** page.

Step 4 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 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 13-1 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 the *Identity and Access Management User Guide*.

----End

13.2 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 Click in the upper left corner and select a region.

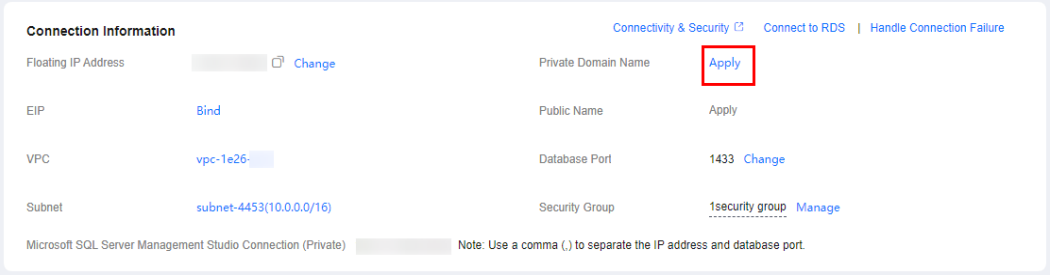
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name to go to the **Overview** page.

Step 4 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 13-2 Applying for a private domain name



Step 5 In the **Private Domain Name** field, view the generated private domain name.

----End

Changing a Private Domain Name

Step 1 Click in the upper left corner and select a region.

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 to go to the **Overview** page.

Step 4 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 5 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 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

13.3 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 Click in the upper left corner and select a region.

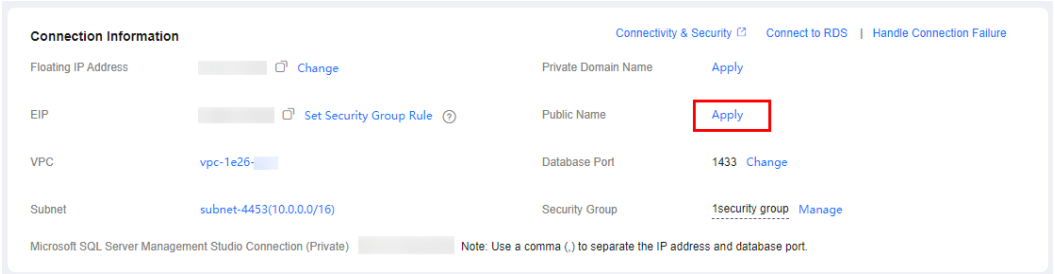
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name to go to the **Overview** page.

Step 4 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 13-3 Applying for a public domain name



Step 5 In the **Public Name** field, view the generated public domain name.

----End

Changing a Public Domain Name

Step 1 Click in the upper left corner and select a region.

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 to go to the **Overview** page.

Step 4 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 5 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 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

13.4 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 an 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

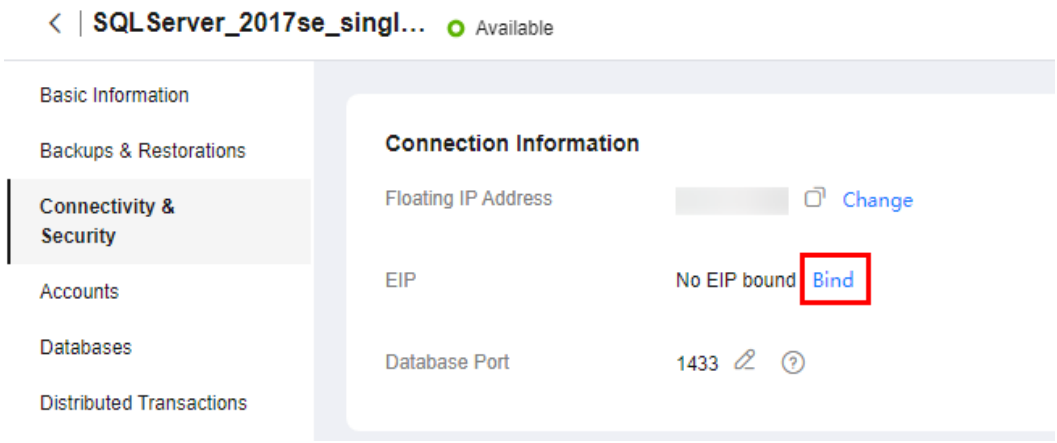
Step 1 Click in the upper left corner and select a region.

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 instance name.

Step 4 In the navigation pane on the left, choose **Connectivity & Security**. In the **Connection Information** area, click **Bind** next to the **EIP** field.

Figure 13-4 Binding an EIP



Step 5 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 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 Click in the upper left corner and select a region.

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**.

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 the *Identity and Access Management User Guide*.

Step 6 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

13.5 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 Click in the upper left corner and select a region.

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 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 5 Check the results on the **Overview** page.

----End

13.6 Configuring Security Group Rules

Scenarios

A security group is a collection of access control rules for ECSs and RDS DB instances that have the same security protection requirements and are mutually trusted within 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 a private network, check whether the ECS and DB instance are in the same security group.

- If they are in the same security group, they can communicate with each other by default. No security group rule needs to be configured. Go to **Connecting to a DB Instance from a Windows ECS over a Private Network**.
- If they 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 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.

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

NOTE

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 **1433**, 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 the **Adding a Security Group Rule** section in the *Virtual Private Cloud User Guide*.

Procedure

Step 1 Click  in the upper left corner and select a region.

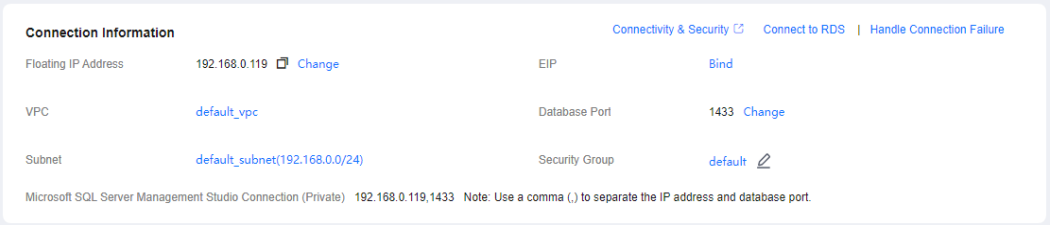
Step 2 Click ☰ in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name.

Step 4 Configure security group rules.

Under **Security Group**, click the security group name.

Figure 13-5 Connection information



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

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

Figure 13-6 Adding an inbound rule

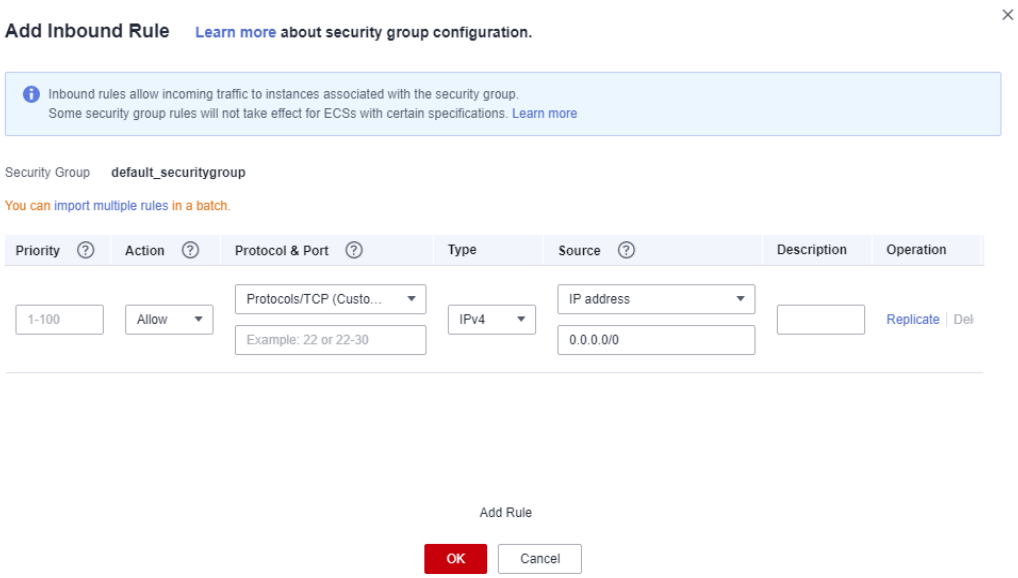


Table 13-1 Inbound rule parameter description

Parameter	Description	Example Value
Priority	Security group rule priority. Value range: 1 to 100. The default priority is 1 and has the highest priority. The security group rule with a smaller value has a higher priority.	1

Parameter	Description	Example Value
Action	Security group rule actions. A rule with a deny action overrides another with an allow action if the two rules have the same priority.	Allow
Protocol & Port	**Protocol**: network protocol. Available options: **All ports**, **Custom TCP**, **Custom UDP**, **ICMP**, and **GRE**.	TCP
	Port: the port over which the traffic can reach your DB instance. An RDS for SQL Server instance can use the default database port 1433 or any port from the range 2100-9500 (excluding 5355 and 5985). If your instance uses 2019 Enterprise Edition, 2019 Standard Edition, 2019 Web Edition, 2017 Enterprise Edition, 2017 Standard Edition, or 2017 Web Edition, ports 5050, 5353, and 5986 cannot be specified for it.	1433
Type	IP address type.	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 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

Parameter	Description	Example Value
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

14 Accounts (Non-Administrator)

14.1 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 the creation 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 Click in the upper left corner and select a region.

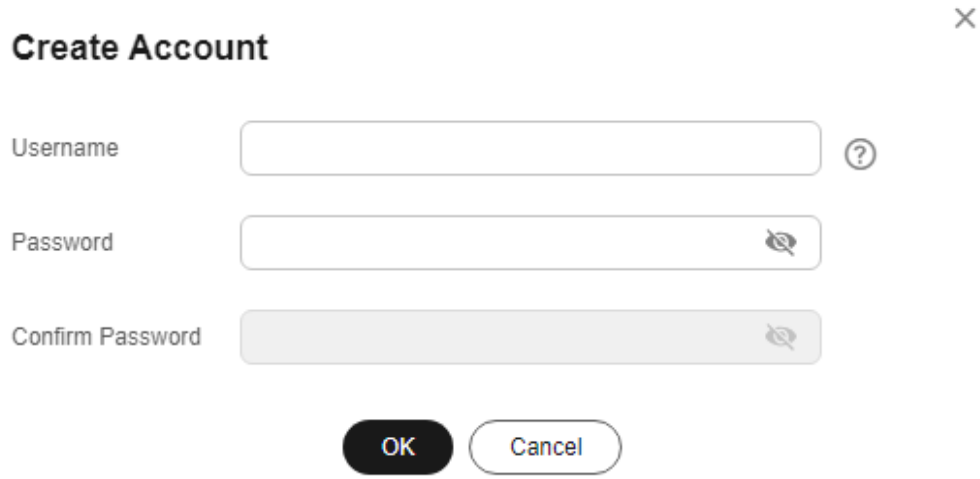
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 On the **Accounts** page, click **Create Account**.

Step 5 In the displayed dialog box, specify **Username**, **Password**, and **Confirm Password**, and click **OK**.

Figure 14-1 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 6 After the account is created, you can manage it on the **Accounts** page.

----End

Creating a Database Account Through DAS

Step 1 Click in the upper left corner and select a region.

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 **Log In** in the **Operation** column.

Step 4 On the displayed page, enter the username and password and click **Log In**.

Step 5 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 6 Run the following command to create an account.

create user *username*;

----End

14.2 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 Click in the upper left corner and select a region.

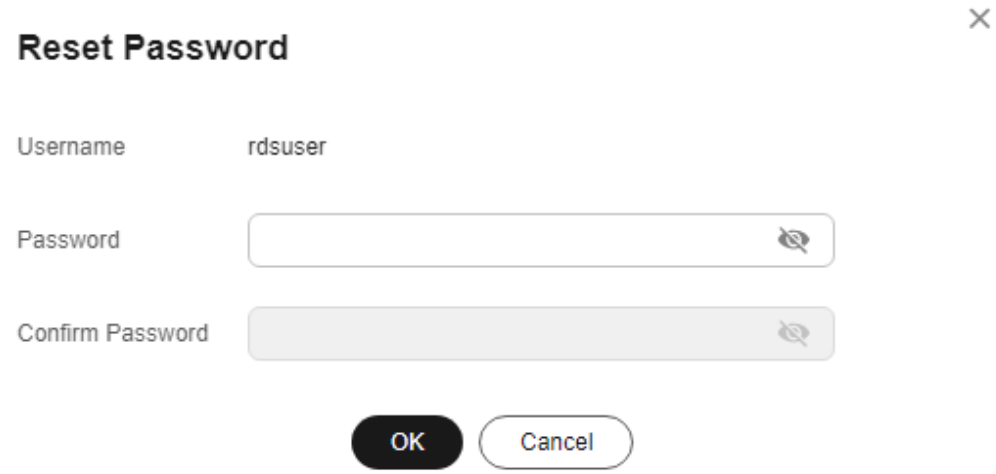
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane on the left, choose **Accounts**. On the **Accounts** page, locate the target username and click **Reset Password** in the **Operation** column.

Step 5 In the displayed dialog box, enter a new password, confirm the password, and click **OK**.

Figure 14-2 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

14.3 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 Click  in the upper left corner and select a region.

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 **Accounts**. On the displayed page, locate the target username and click **Delete** in the **Operation** column.

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

----End

15 Databases

15.1 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.
- Databases with HA relationships can be renamed only after the replication relationships are removed.

Creating a Database Through RDS

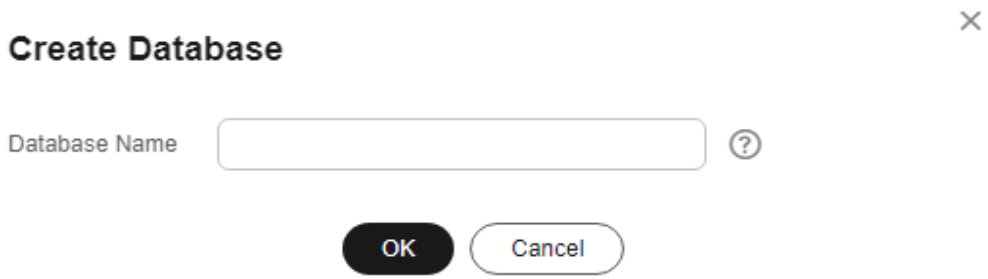
Step 1 Click in the upper left corner and select a region.

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 On the **Databases** page, click **Create Database**. In the displayed dialog box, enter a database name and click **OK**.

Figure 15-1 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 5 After the database is created, manage it on the **Databases** page.

----End

Creating a Database Through DAS

Step 1 Click in the upper left corner and select a region.

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 and click **Log In** in the **Operation** column.

Step 4 On the displayed page, enter the username and password and click **Log In**.

Step 5 On the top menu bar, choose **SQL Operations** > **SQL Query**.

Step 6 Run the following command to create a database.

create database *database_name*;

----End

15.2 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 Click in the upper left corner and select a region.

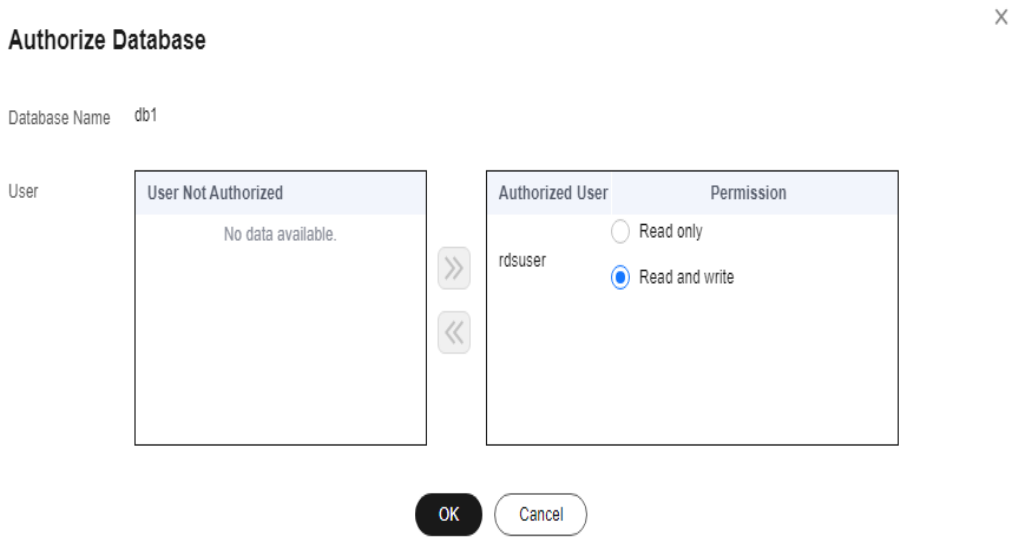
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 In the navigation pane on the left, choose **Databases**. On the displayed page, locate the target database and click **Authorize** in the **Operation** column.

Step 5 In the displayed dialog box, select unauthorized users and click to authorize them or select authorized users and click to revoke permissions.

Figure 15-2 Authorization



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

Step 6 Then, click **OK**.

----End

15.3 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 Click  in the upper left corner and select a region.

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

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 the *Identity and Access Management User Guide*.

----End

15.4 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 Click in the upper left corner and select a region.

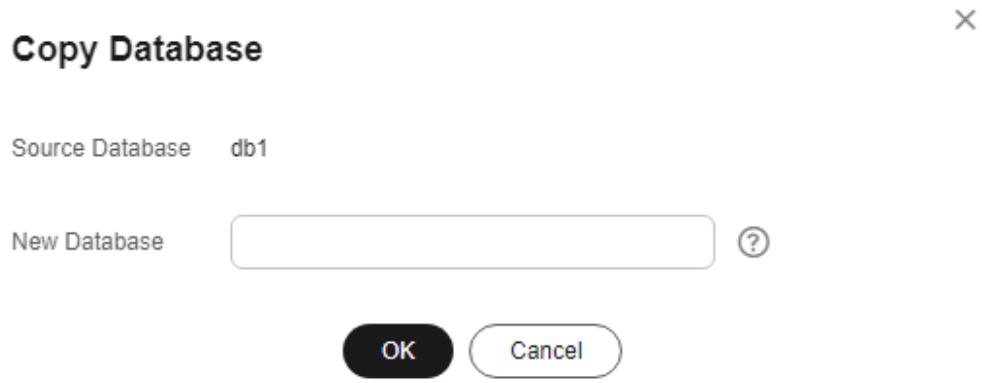
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 On the **Databases** page, locate the target database and choose **More** > **Copy** in the **Operation** column.

Step 5 In the displayed dialog box, enter the new database name and click **OK**.

Figure 15-3 Copying a database



----End

15.5 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 Click in the upper left corner and select a region.

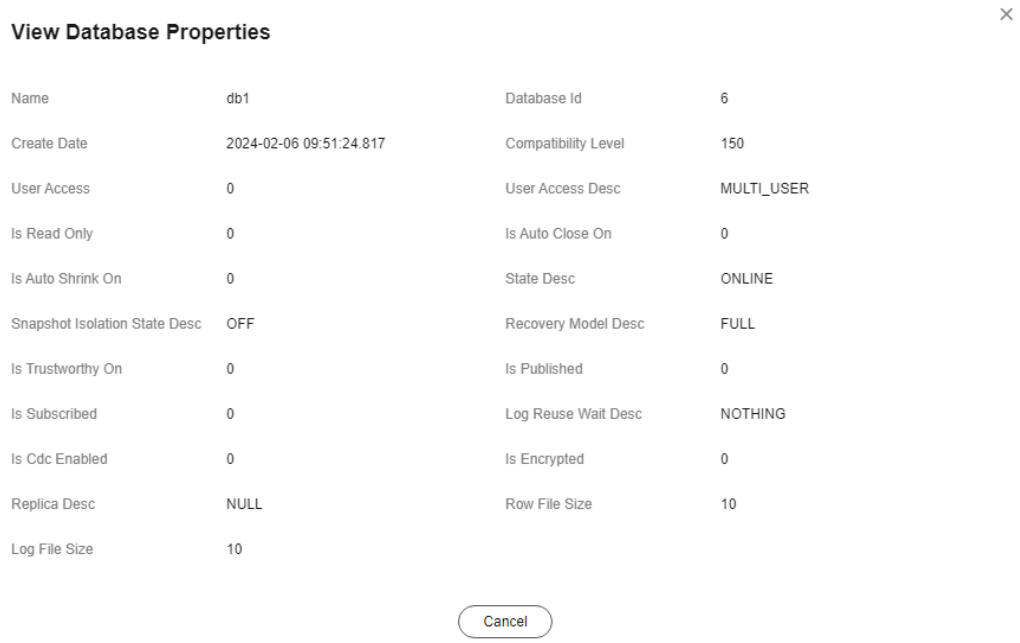
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the instance name.

Step 4 On the **Databases** page, locate the target database and choose **More** > **View Database Properties**.

Step 5 In the displayed dialog box, view the database properties.

Figure 15-4 Viewing database properties



----End

16 Security and Encryption

16.1 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, the new password strength must be at least as strong as the initial setting.

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.

16.2 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 Click  in the upper left corner and select a region.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

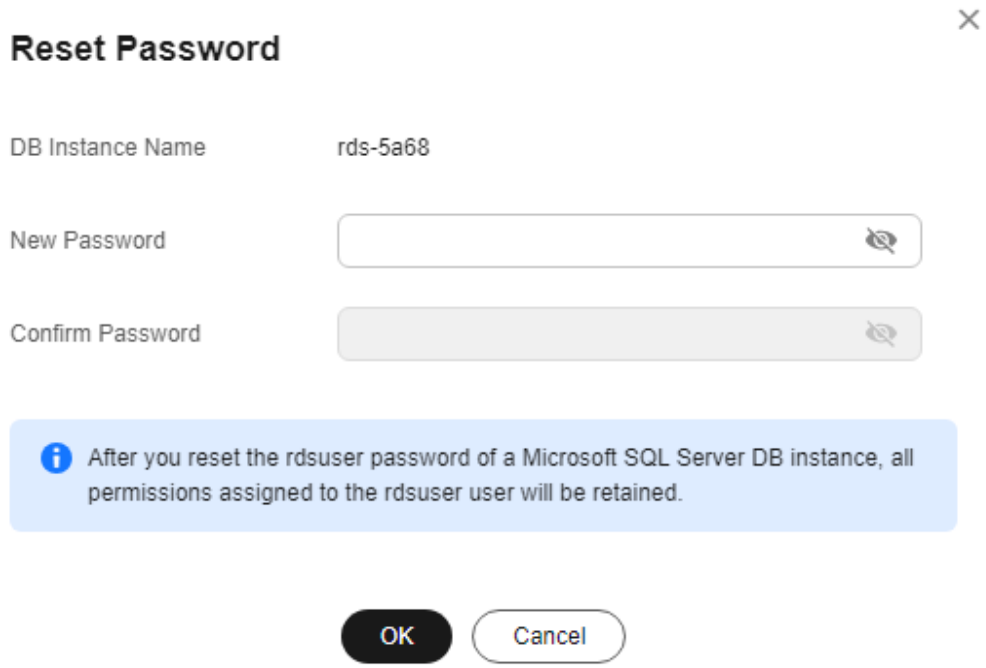
Step 3 On the **Instances** page, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

Step 4 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 16-1 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

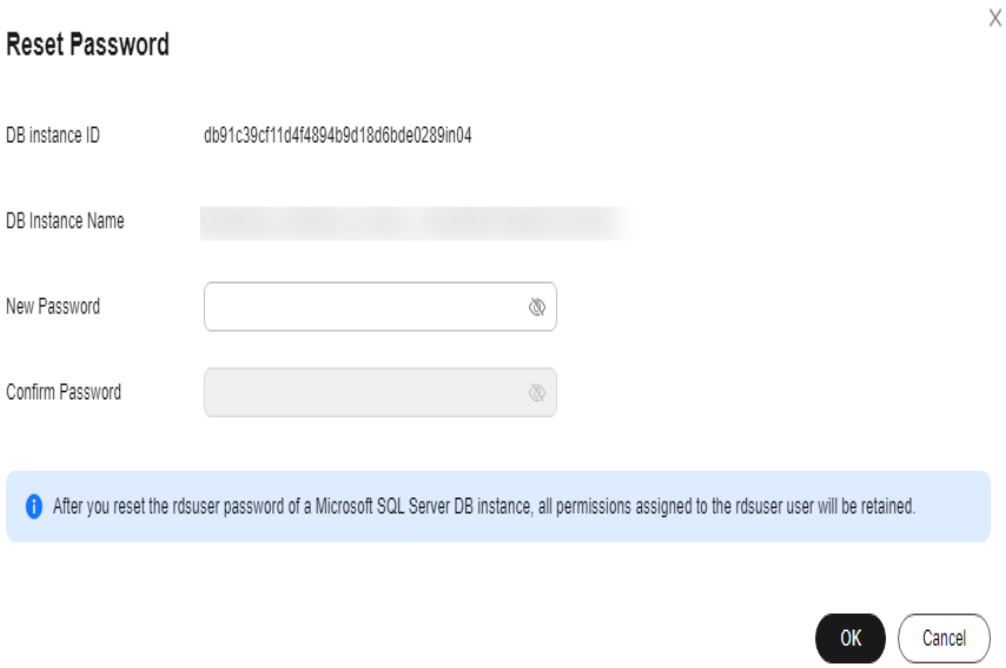
Step 1 Click in the upper left corner and select a region.

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 instance name.

Step 4 On the **Overview** page, find **Administrator** and click **Reset Password** under it. In the displayed dialog box, enter and confirm the new password.

Figure 16-2 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

16.3 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.

Managing Security Groups

Step 1 Click in the upper left corner and select a region.

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 or read replica.

Step 4 On the **Overview** page, click **Manage** under **Security Group**.

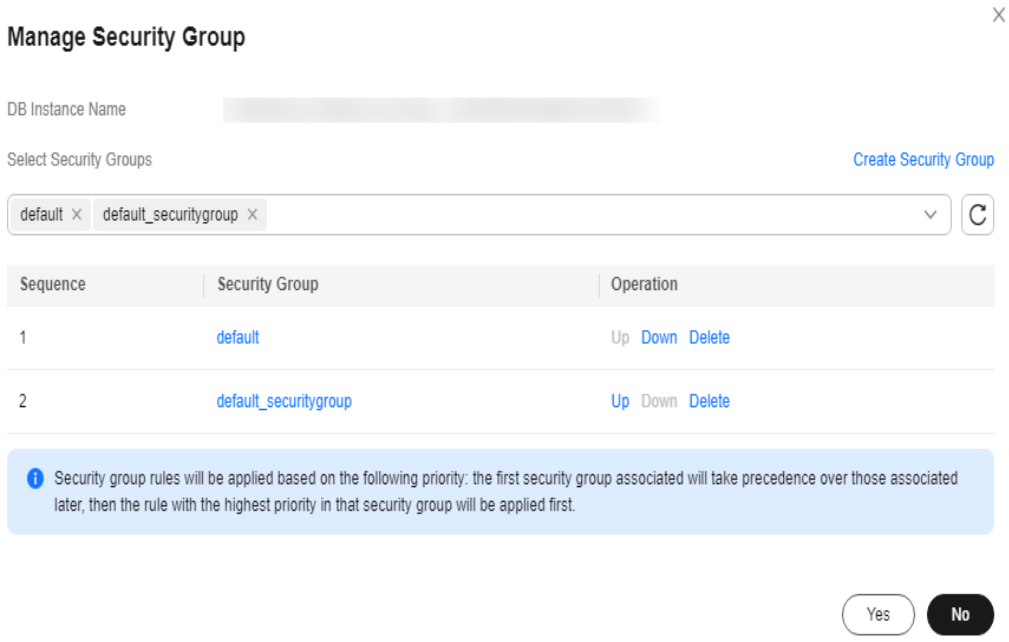
You can select multiple security groups at a time. The security group rules will be applied based on the following sequence: the first security group associated will take precedence over those associated later, then the rule with the highest priority in that security group will be applied first.

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 16-3 Managing security groups



Step 5 Click **Yes** to submit the modification.

----End

16.4 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. 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. 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 instance will be unavailable.
- After an RDS DB instance is created, do not disable or delete the key that is currently in use, 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.

16.5 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 16-1**.

Table 16-1 RDS for SQL Server editions that support TDE

DB Instance Type	Editions Support for TDE
Primary/Standby	• 2012 Enterprise Edition • 2014 Enterprise Edition • 2016 Enterprise Edition • 2019 Standard Edition • 2022 Standard Edition
Single-node	• 2012 Enterprise Edition • 2014 Enterprise Edition • 2016 Enterprise Edition • 2019 Standard Edition • 2022 Standard Edition
Cluster	• 2017 Enterprise Edition • 2019 Enterprise Edition • 2022 Enterprise Edition

Constraints

1. If TDE has been enabled for a single-node DB instance, the instance cannot be changed to a primary/standby DB instance.
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 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

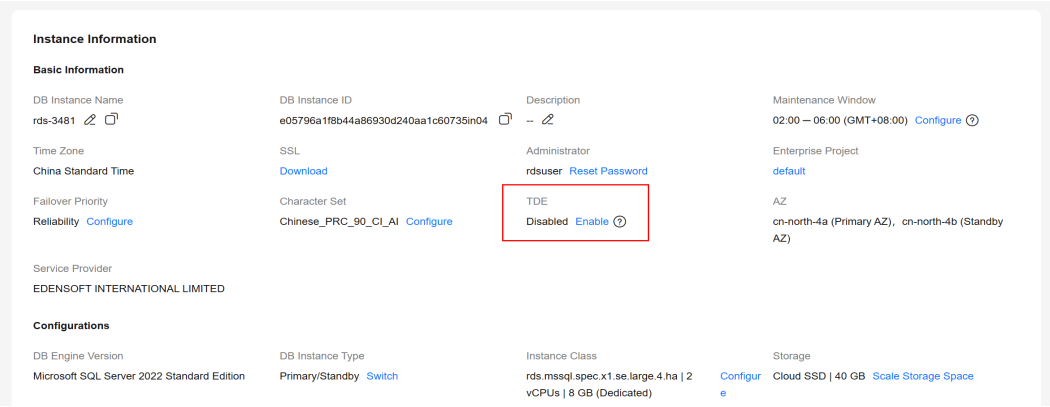
Step 1 Click in the upper left corner and select a region.

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 instance name to go to the **Overview** page.

Step 4 Under **TDE**, click **Enable**.

Figure 16-4 Enabling instance-level TDE



Step 5 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 an RDS for SQL Server Instance Through a Public Network**, **Connecting to an RDS for SQL Server Instance Through a Private Network**, and **Connecting to an RDS for SQL Server Instance Through DAS**.

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*

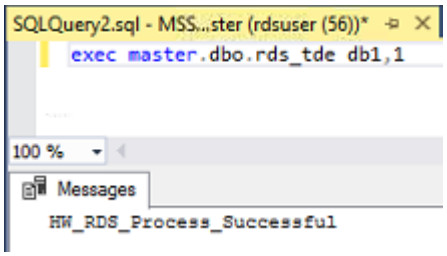
- ***DatabaseName***: indicates the target database name, which can be **null**.
- *TDE_Action*:
  - The value **-1** indicates that the database encryption status is queried.

    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

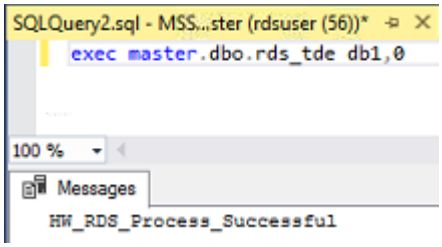
   Figure 16-5 Enabling TDE

2. Disable TDE for database db1.

   exec master.dbo.rds_tde db1, 0

   Figure 16-6 Disabling TDE

3. Query the TDE status of database db1.

   exec master.dbo.rds_tde db1, -1

   Figure 16-7 Querying the TDE status (Enabled)

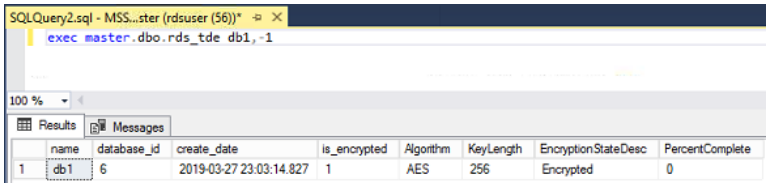
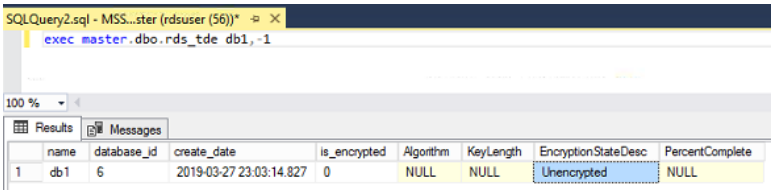
   

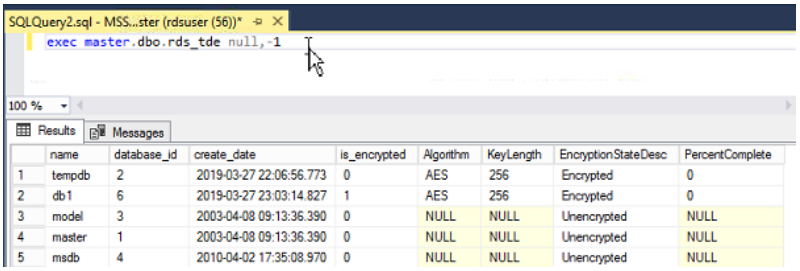
Figure 16-8 Querying the TDE status (Disabled)



4. Query the TDE status of all databases.

   exec master.dbo.rds_tde null, -1

Figure 16-9 Querying the TDE status of all databases



----End

16.6 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.

17 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 Click in the upper left corner and select a region.

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 **Distributed Transactions**. On the displayed page, click  in the **Distributed Transaction** field.

----End

Adding Hosts

Step 1 Click  in the upper left corner and select a region.

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 **Distributed Transactions**. On the displayed page, click **Add Host**.

Step 5 In the displayed dialog box, enter host names and IP addresses, and click **Test Connection**. After all host connection tests are successful, click **OK**.

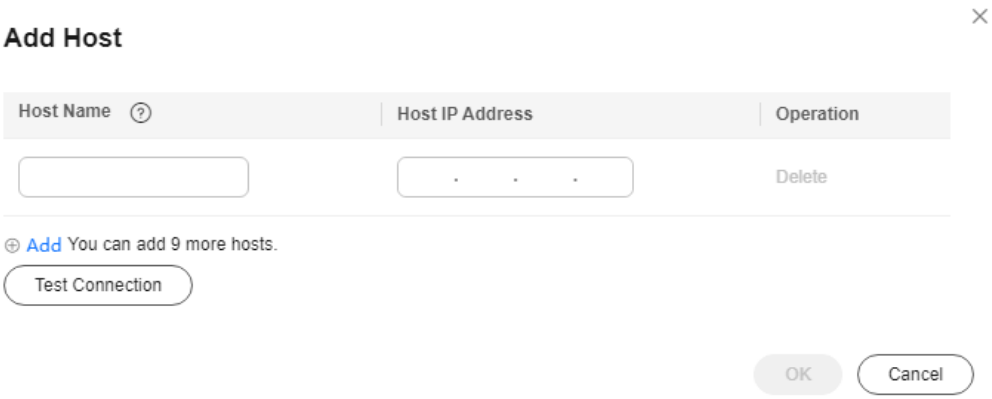
- Host IP name: Enter the names of the hosts for which you want to create distributed transactions with the RDS DB instance. Each host name must be unique and contains 1 to 64 characters, including only letters, digits, and hyphens (-).
- Host IP address: Enter the IP addresses of the hosts for which 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 17-1 Adding a host



----End

Resolving Names on Remote Servers (ECSs)

Step 1 Click in the upper left corner and select a region.

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 **Distributed Transactions**. On the displayed page, obtain information about the RDS DB instance.

Step 4 Add the RDS DB instance information to the hosts file in **C:\Windows\System32\drivers\etc\hosts**.

----End

Configuring MSDTC on a Remote Server

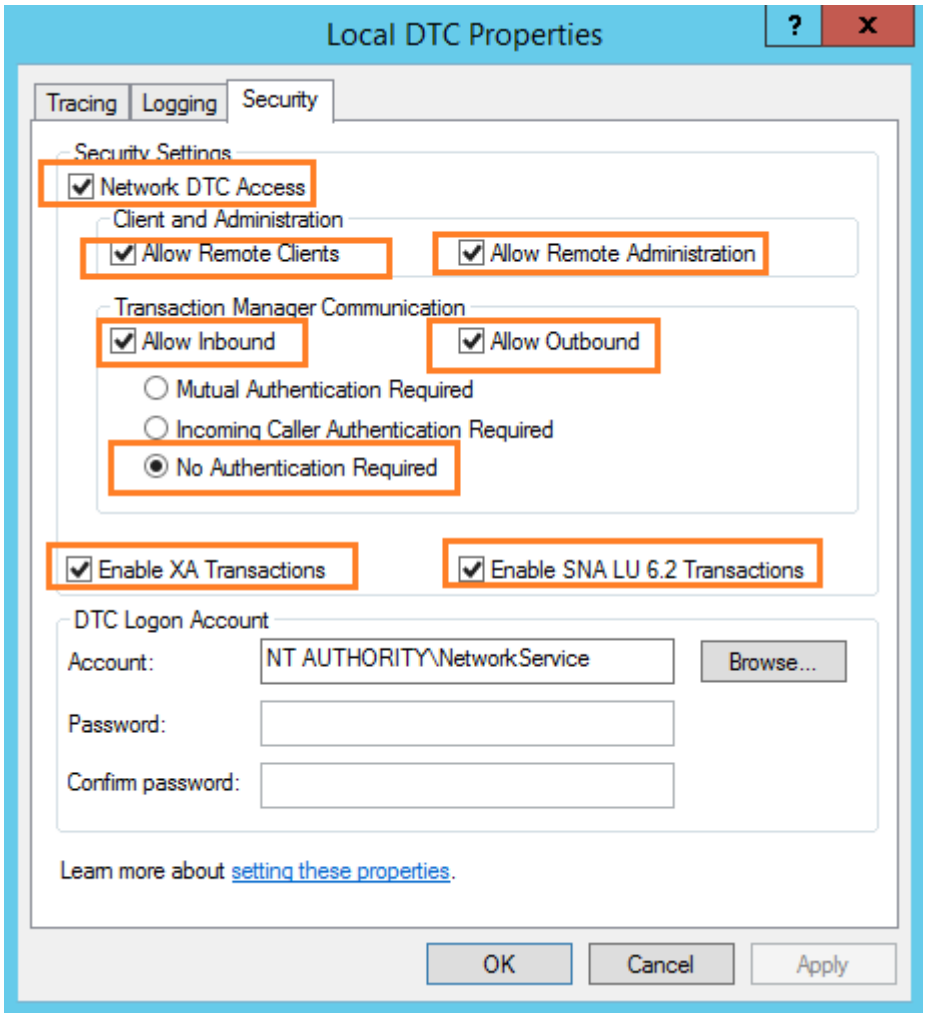
Step 1 Click **Start** and choose **Control Panel** > **Administrative Tools** > **Component Services**.

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 17-2** and click **OK**.

Figure 17-2 Local DTC properties



----End

Deleting Hosts

Step 1 Click in the upper left corner and select a region.

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

----End

18 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 Click in the upper left corner and select a region.

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 instance name.

Step 4 In the navigation pane on the left, click **SSIS**. On the displayed page, click next to **Enable SSIS**.

Step 5 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 Click in the upper left corner and select a region.

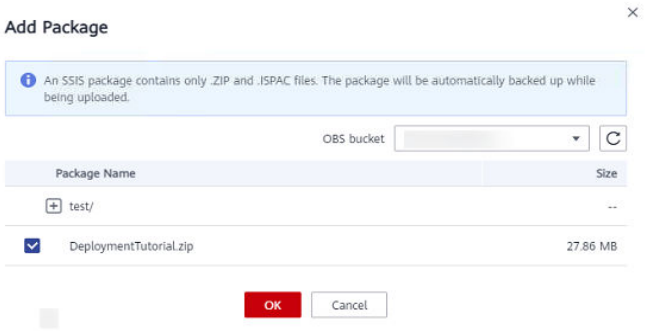
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name.

Step 4 In the navigation pane on the left, choose **SSIS** and click **Add Package**.

Step 5 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 18-1 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**.

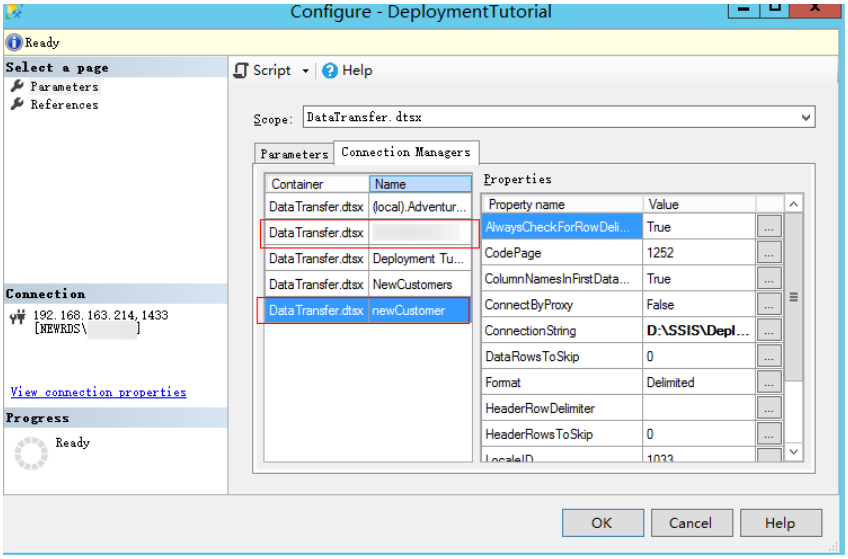
Step 3 Choose **Integration Service Catalogs** > **SSISDB**, right-click to create an SSIS folder, and enter a name for the folder. Then, two subfolders **Projects** and **Environments** are automatically created.

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.

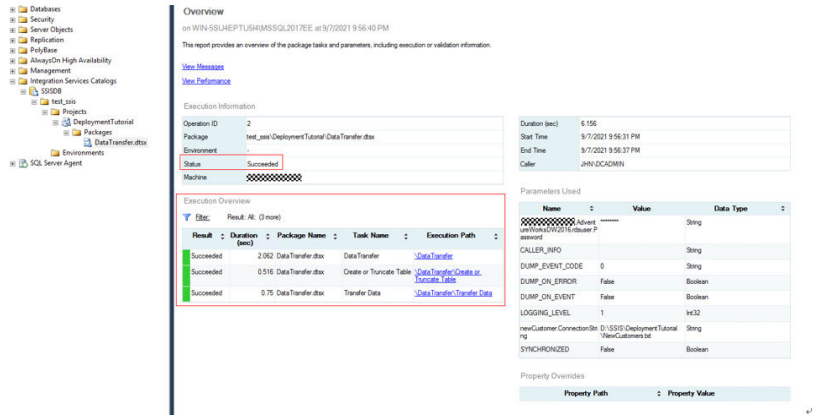
1. Right-click the package name, choose **Configure** from the shortcut menu, and configure parameters.

Figure 18-2 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 18-3 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

19 Metrics and Alarms

19.1 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 19-1 lists the performance metrics of RDS for SQL Server DB instances.

Table 19-1 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

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
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
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

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	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
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

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
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
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

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
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
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

Metric ID	Name	Description	Value Range	Unit	Conversion Rule	Monitored Object (Dimension)	Monitoring Interval (Raw Data)
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
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-node 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

19.2 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 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 Click in the upper left corner and select a region.

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 **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 4 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

19.3 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

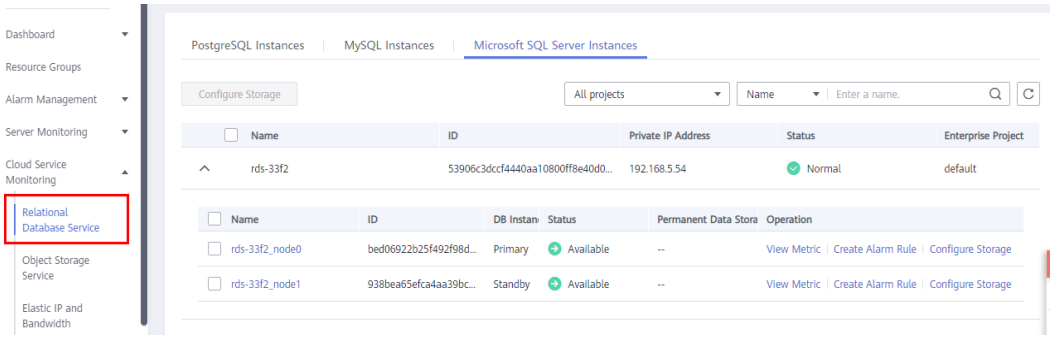
Step 1 Log in to the management console.

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**.

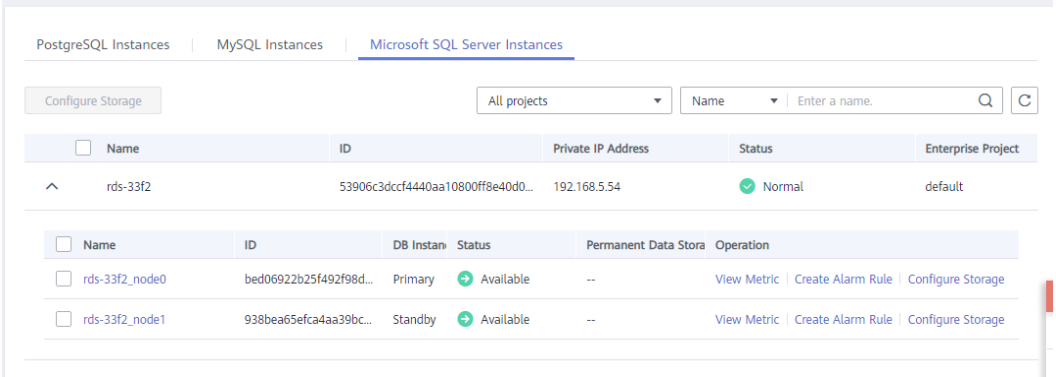
Step 4 In the navigation pane on the left, choose **Cloud Service Monitoring** > **Relational Database Service**.

Figure 19-1 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 19-2 Creating an alarm rule



Step 6 On the displayed page, set parameters as required.

Figure 19-3 Configuring alarm information

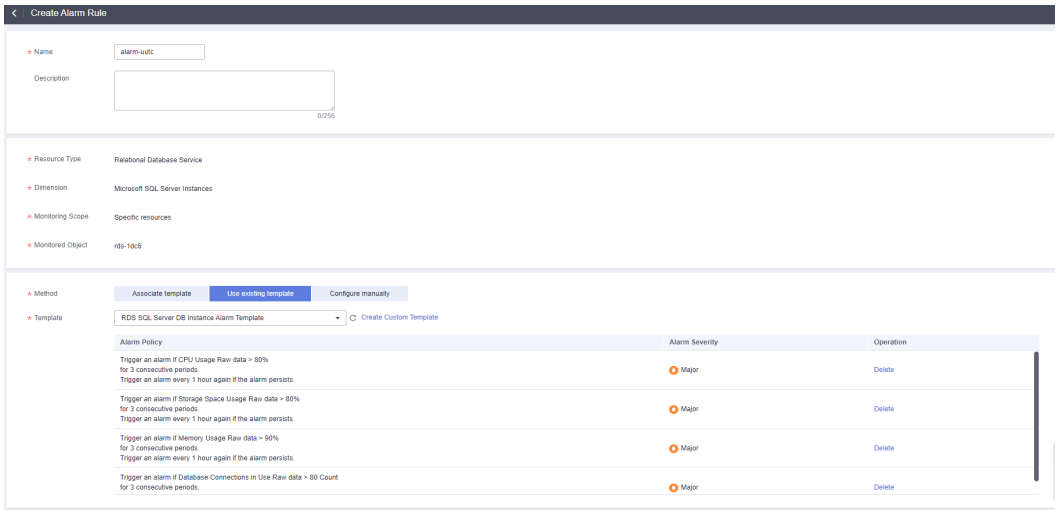


Table 19-2 Alarm rule information

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.

Parameter	Description
Alarm Severity	The alarm severity can be **Critical**, **Major**, **Minor**, or **Informational**.

Figure 19-4 Configuring alarm notification

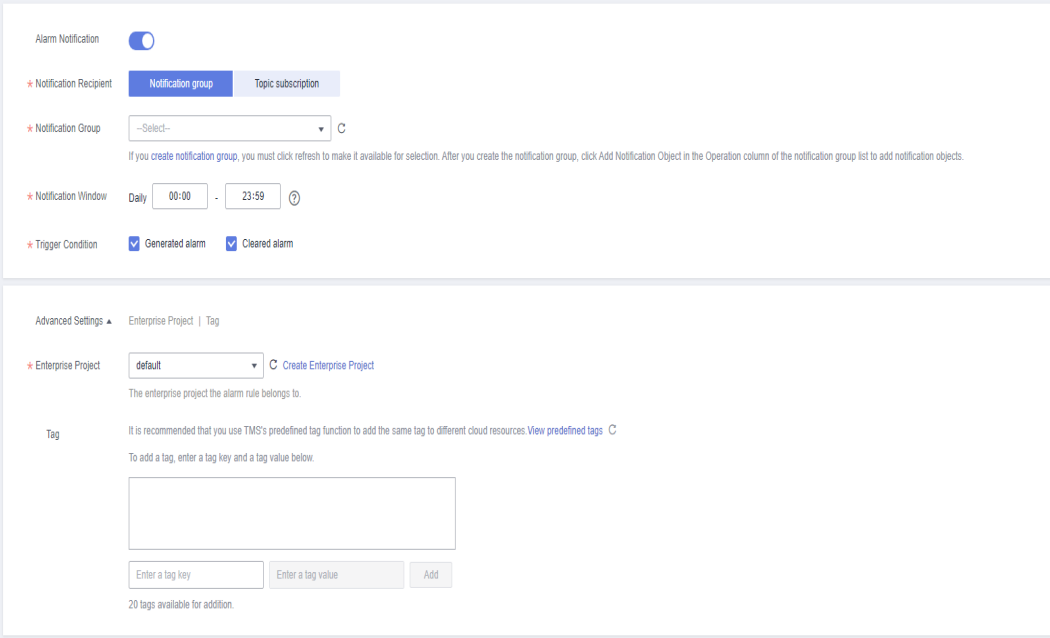


Table 19-3 Alarm notification

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.

Parameter	Description
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

19.4 Event Monitoring

19.4.1 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**.

19.4.2 Viewing Event Monitoring Data

Scenarios

This section describes how to view the event monitoring data.

Procedure

Step 1 Click  in the upper left corner and select a region.

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 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 4 Click  to return to the main page of Cloud Eye.

Step 5 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 6 Click **View Graph**. On the details page, click **View Event** in the **Operation** column of a specific event to view details.

----End

19.4.3 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 19-4 Parameter description

Parameter	Description
Name	Specifies the alarm rule name. The system generates a random name, which you can modify.

Parameter	Description
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 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 19-5 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

19.4.4 Events Supported by Event Monitoring

Description

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.

Namespace

SYS.RDS

Events That Can Be Monitored

Table 19-6 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.
	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.
	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 19-7 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
	Primary/standby switchover or failover	PrimaryStandbySwitched	Major	A switchover or failover is performed.

20 Interconnection with CTS

20.1 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 20-1 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

Operation	Resource Type	Trace Name
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
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

Operation	Resource Type	Trace Name
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	createParameterGroup
Modifying parameters in a parameter template (using the console or Trove APIs)	parameterGroup	updateParameterGroup
Deleting a parameter template (using the console or Trove APIs)	parameterGroup	deleteParameterGroup
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	compareParameterGroup
Applying a parameter template (using the console)	parameterGroup	applyParameterGroup
Saving parameters in a parameter template (using the console)	parameterGroup	saveParameterGroup
Deleting a frozen DB instance (using the console)	all	deleteInstance
Freezing a DB instance (using the console)	all	rdsfreezeInstance
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

20.2 Viewing Tracing Events

For details about how to view audit logs, see **Querying Real-Time Traces**.

21 Log Management

21.1 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

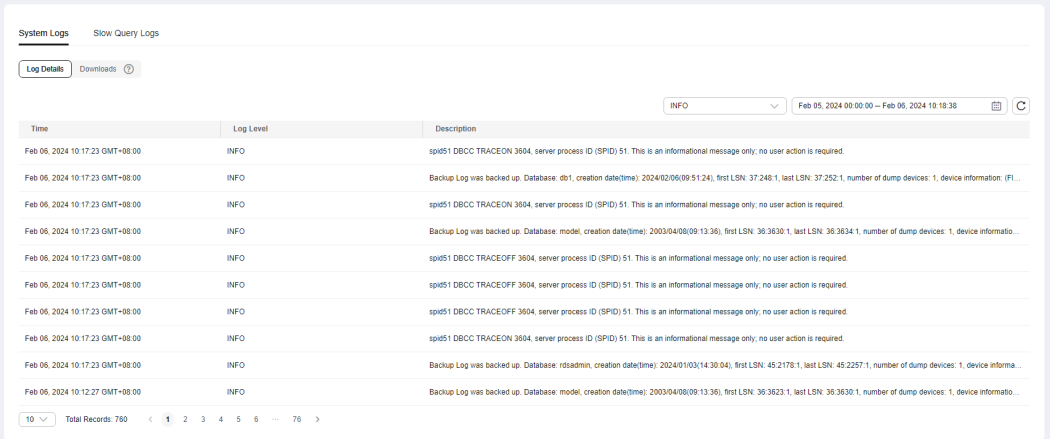
Step 1 Click in the upper left corner and select a region.

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 instance name.

Step 4 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Log Details** to view details about system logs.

Figure 21-1 System log details



- You can select a log level in the upper right corner to view logs of the selected level. Currently, only INFO logs can be viewed.
- You can click in the upper right corner to view logs generated in different time segments.
- A maximum of 2,000 system log records can be queried.
- 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

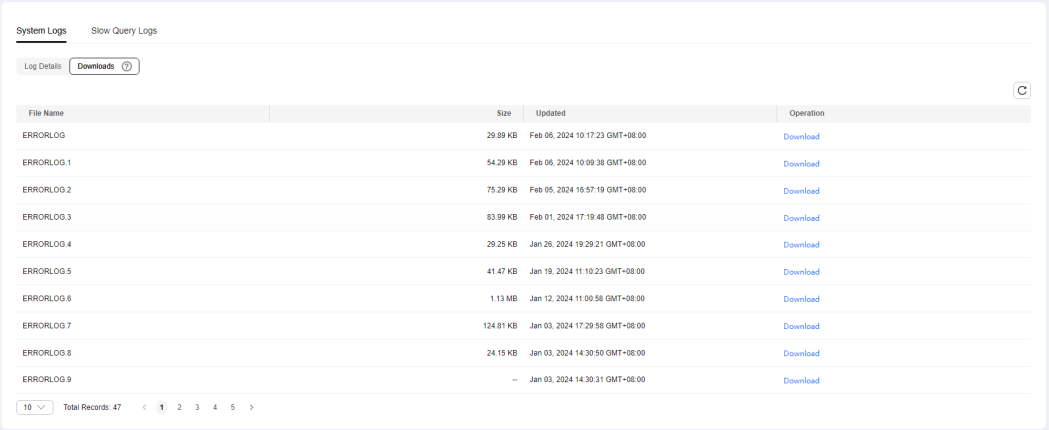
Step 1 Click in the upper left corner and select a region.

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 instance name.

Step 4 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Downloads**.

Figure 21-2 Downloading system logs



NOTE

- **ERRORLOG** refers to error logs. You can download a maximum of 30 rotated error log files at a time, with each not exceeding 20 MB.
- *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 4.1** to try to download a log again.

----End

21.2 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 21-1** 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 21-1 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.
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.

Action Group Name	Description
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.
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 Click  in the upper left corner and select a region.

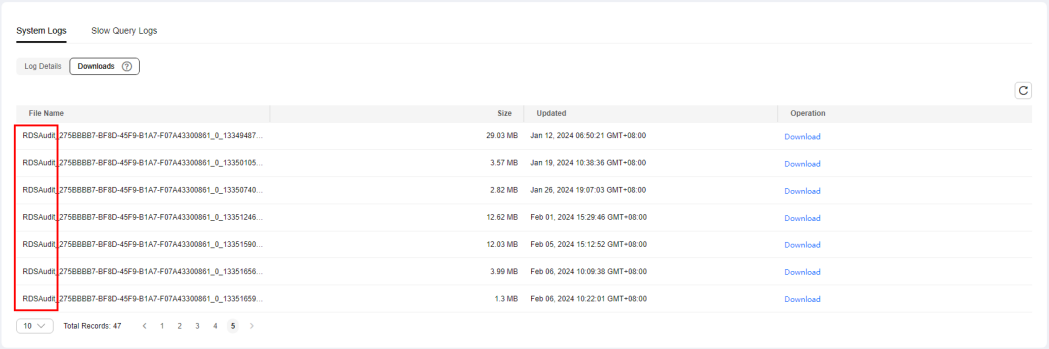
Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name.

Step 4 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Downloads**.

Step 5 On the **Downloads** page, record the names of audit logs.

Figure 21-3 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 6 Connect to the DB instance through a SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Step 7 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*)**

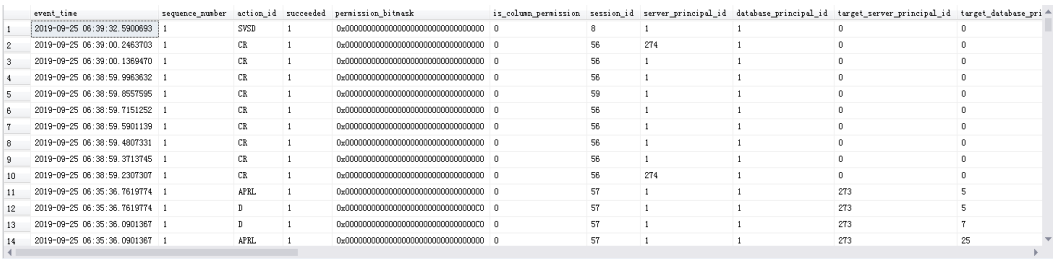
Table 21-2 Parameter description

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 21-4 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 Click in the upper left corner and select a region.

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 instance name.

Step 4 In the navigation pane on the left, choose **Logs**. On the **System Logs** page, click **Downloads**.

Step 5 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 5** to try to download a log again.

----End

21.3 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 21-3 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 Click in the upper left corner and select a region.

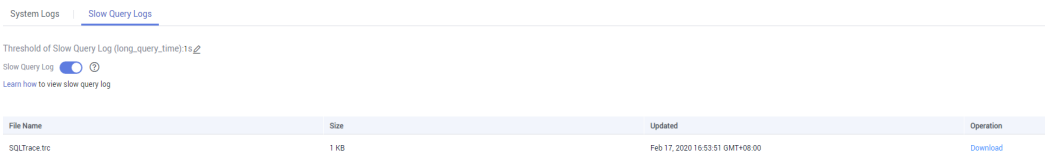
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the target instance name.

Step 4 In the navigation pane on the left, choose **Logs**. On the **Slow Query Logs** page, click to enable the slow query log function.

Step 5 The generated slow query logs are displayed.

Figure 21-5 Slow query logs



NOTE

Enabling slow query log slightly affects DB instance performance.

Step 6 Connect to the DB instance through a SQL Server client. For details, see **Connecting to a DB Instance Through a Public Network**.

Step 7 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

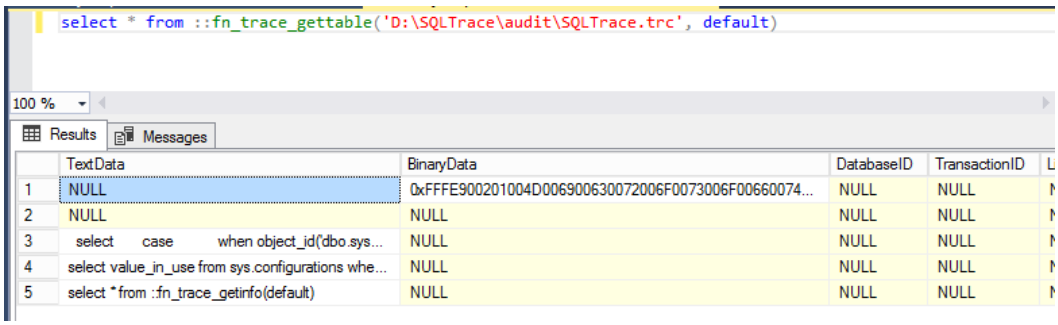
XXX indicates the name of the slow query log recorded in **Step 5**.

Example:

select * from ::fn_trace_gettable('D:\SQLTrace\audit\SQLTrace.trc', default)

The result is shown in **Figure 21-6**.

Figure 21-6 Slow query log details



----End

Downloading a Log

Step 1 Click in the upper left corner and select a region.

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 instance name.

Step 4 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 5 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 5** 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

22 DBA Assistant

22.1 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**.

22.2 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

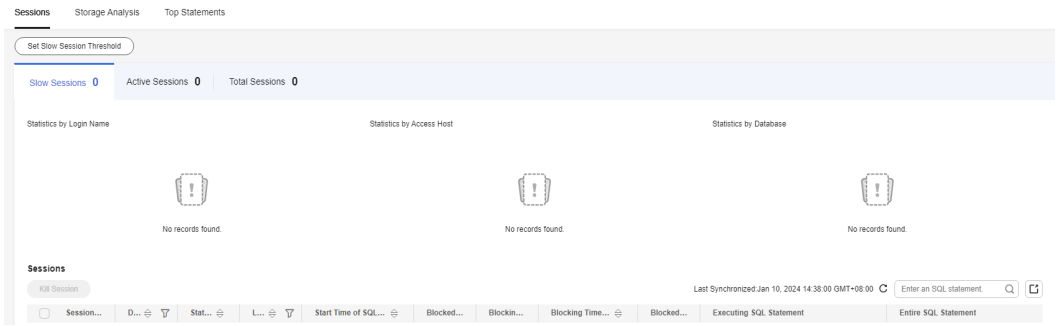
Step 1 Click in the upper left corner and select a region.

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, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 Click the **Sessions** tab to view current session statistics by login name, access host, and database.



Step 6 In the session list, select the abnormal session you want to end and click **Kill Session** to recover the database.

----End

22.3 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 Click in the upper left corner and select a region.

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 name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 On the **Storage Analysis** tab page, view storage usage. If your storage is insufficient, scale it up.

Figure 22-1 Overview

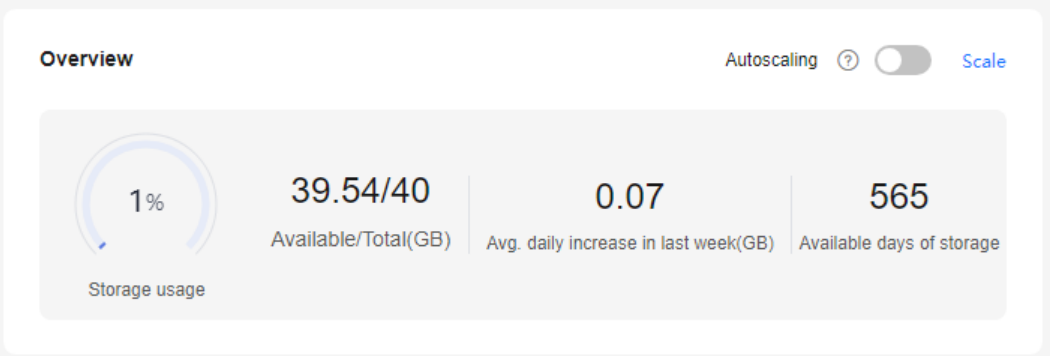


Table 22-1 Parameter description

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.

Parameter	Description
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 22-2 Disk space distribution

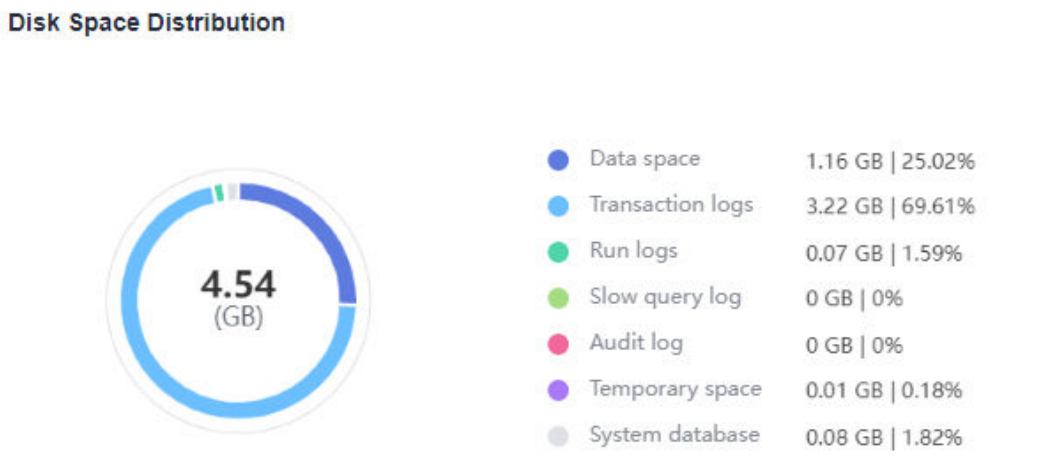


Table 22-2 Disk space distribution parameters

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.

Parameter	Description
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 22-3 Database list parameters

Parameter	Description
Database	Database name.
Status	Database status.
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 22-4 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.

Parameter	Description
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 22-5 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.

22.4 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 Click in the upper left corner and select a region.

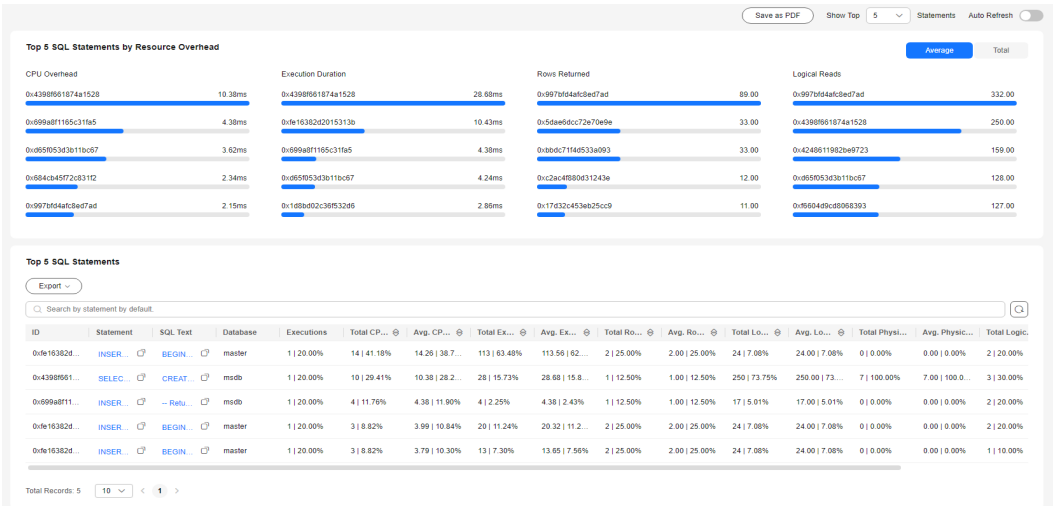
Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **DBA Assistant** > **Real-Time Diagnosis**.

Step 5 On the **Real-Time Top SQL** tab page, view the top SQL statements by CPU overhead, execution duration, rows returned, and logical reads.

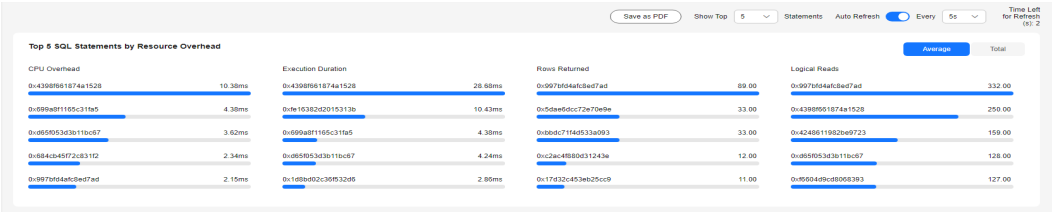
Figure 22-3 Real-Time Top SQL



- To export details about top SQL statements, click .
- To sort parameter values in the top SQL details, click in the table header.

Step 6 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 22-4 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 22-5 Average overhead

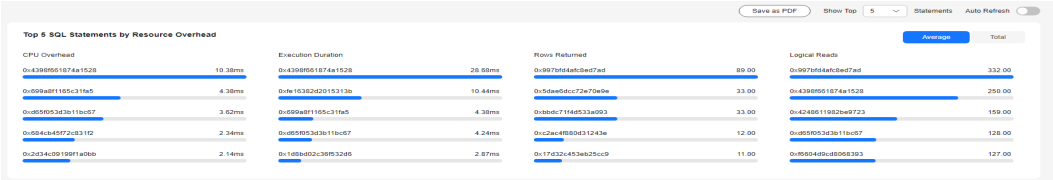
  

  Figure 22-6 Total overhead

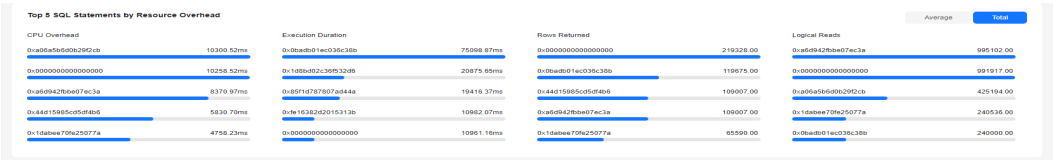
  

  Table 22-6 Parameter description for top SQL statements by resource overhead

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 22-7 Top SQL statement list

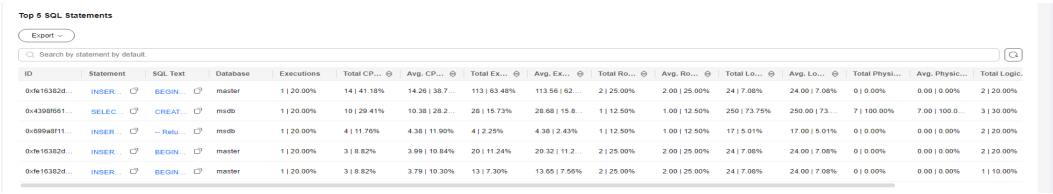


Table 22-7 Parameter description for top SQL statement list

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.
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.

22.5 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 Click in the upper left corner and select a region.

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.

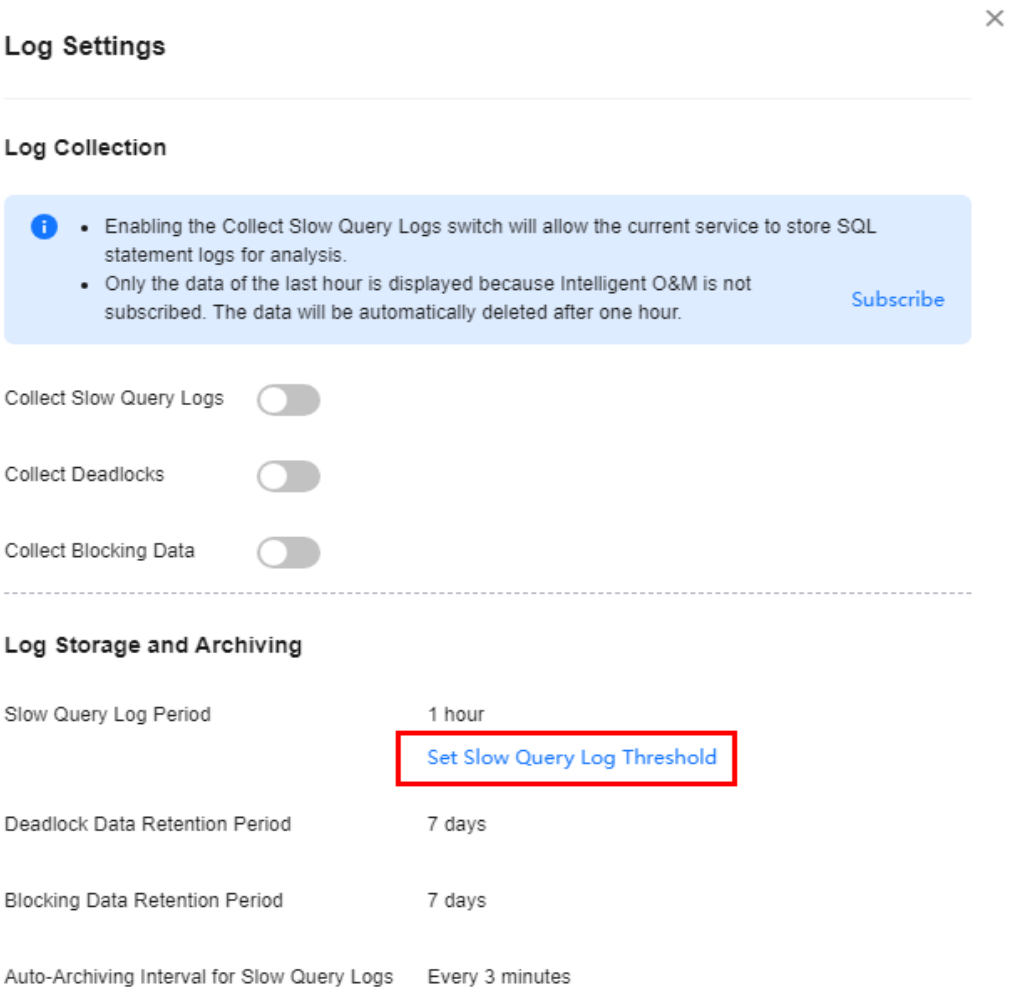
Step 4 In the navigation pane, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 Click the **Slow Query Log** tab.

Step 6 Click next to the **Collect Slow Query Logs** field.

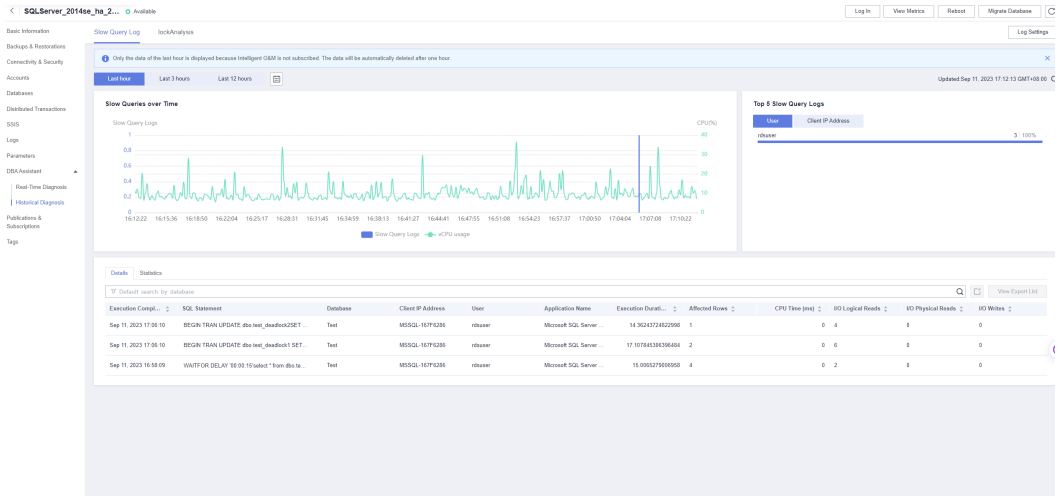
Step 7 Click **Log Settings** in the upper right corner of the page to adjust the slow query log threshold.

Figure 22-8 Setting the slow query log threshold



Step 8 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 22-9 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

22.6 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 Click  in the upper left corner and select a region.

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, choose **DBA Assistant** > **Historical Diagnosis**.

Step 5 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 22-8 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 22-9 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.

Parameter	Description
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.
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

23 Publications and Subscriptions

23.1 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 23-1 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

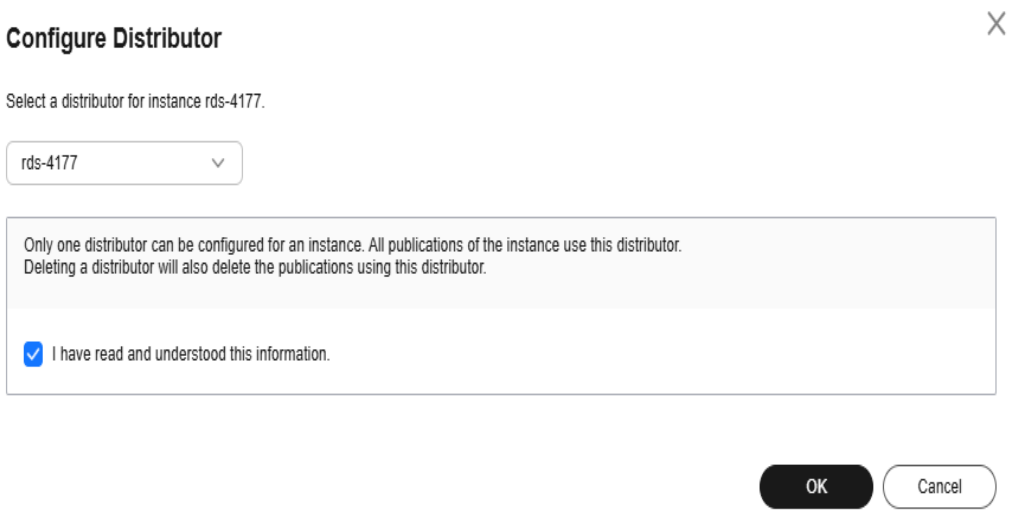
Step 3 On the **Instances** page, click the DB instance name.

Step 4 In the navigation pane, choose **Publications & Subscriptions**.

Step 5 On the **Publications** page, click **Configure Distributor**.

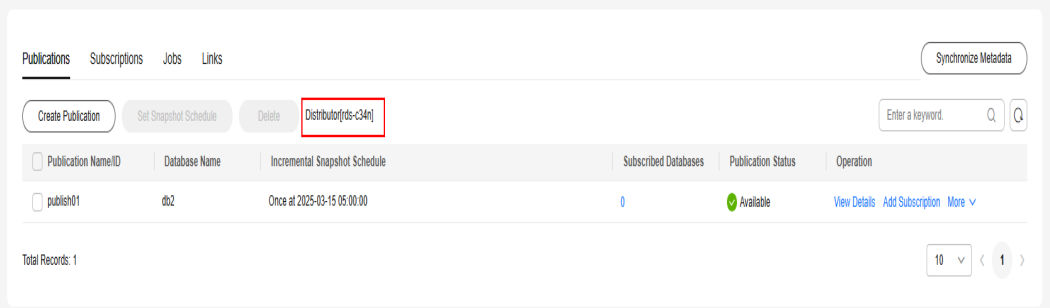
Step 6 In the displayed dialog box, select the current instance or another instance as the distributor, select **I have read and understood this information**, and click **OK**.

Figure 23-2 Configuring a distributor



Step 7 View the configured distributor.

Figure 23-3 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 23-4 Creating a publication

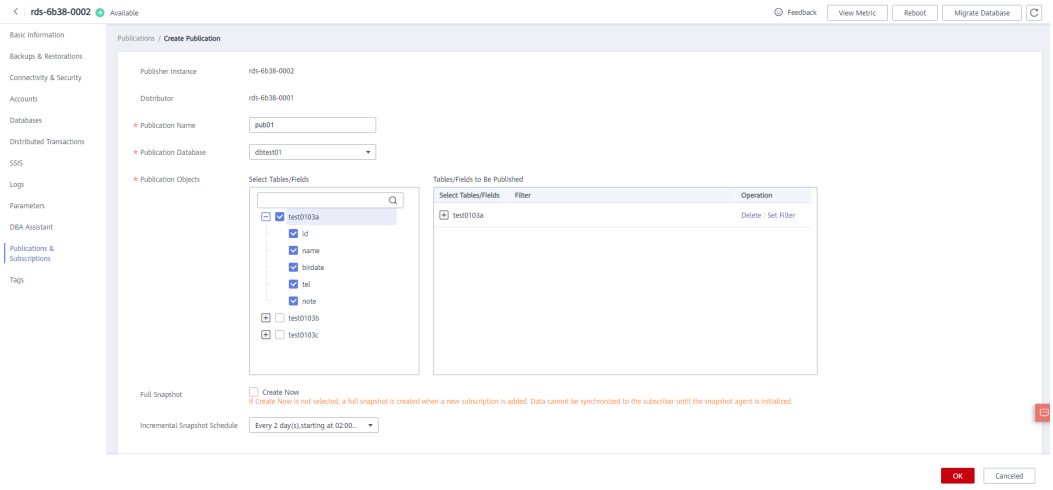
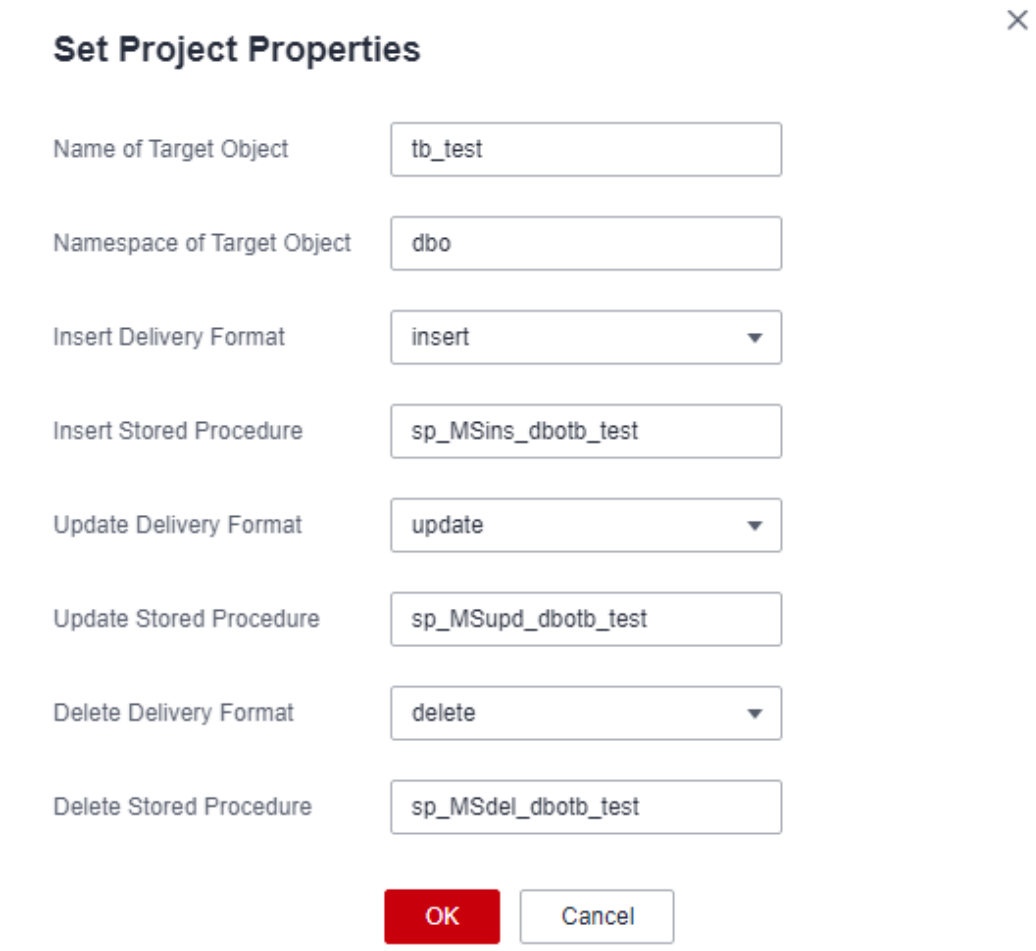
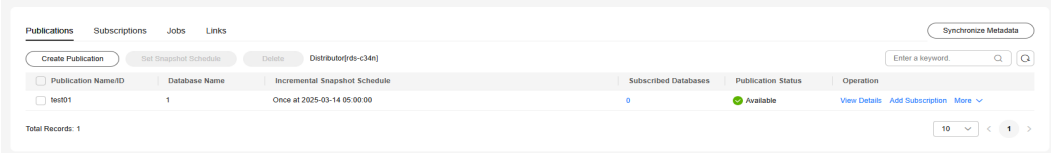


Figure 23-5 Setting project properties



Step 5 View the created publication.

Figure 23-6 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**.

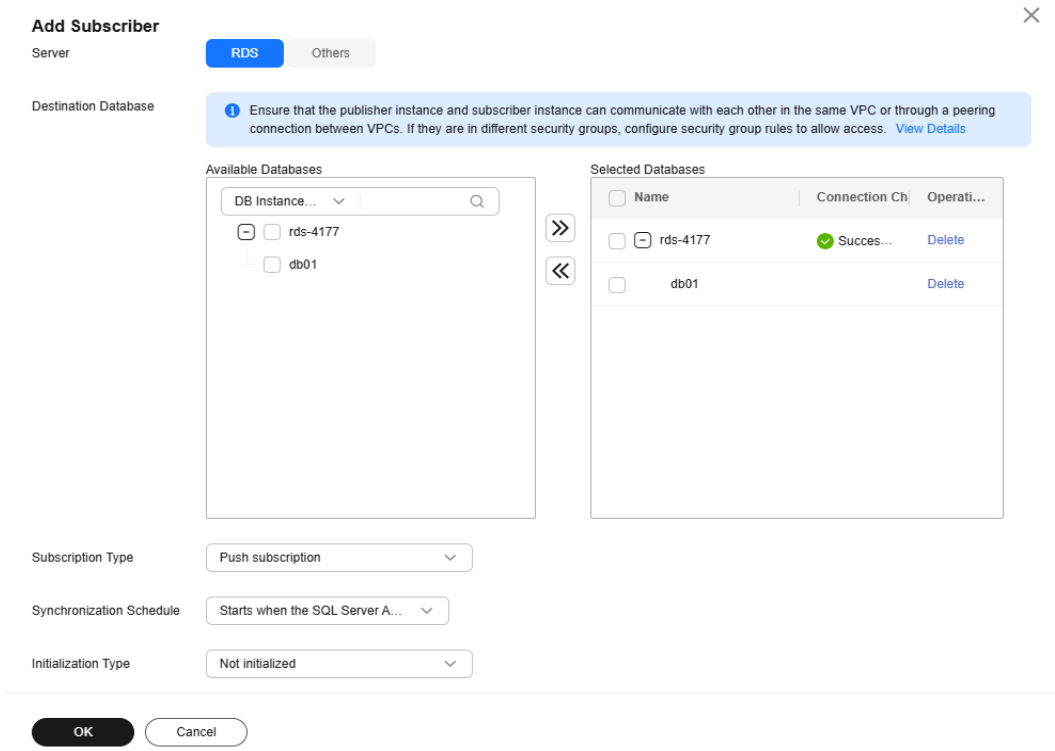
For details about the publisher and subscriber types supported by RDS for SQL Server, see **Compatibility Between Publishers and Subscribers**.

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.

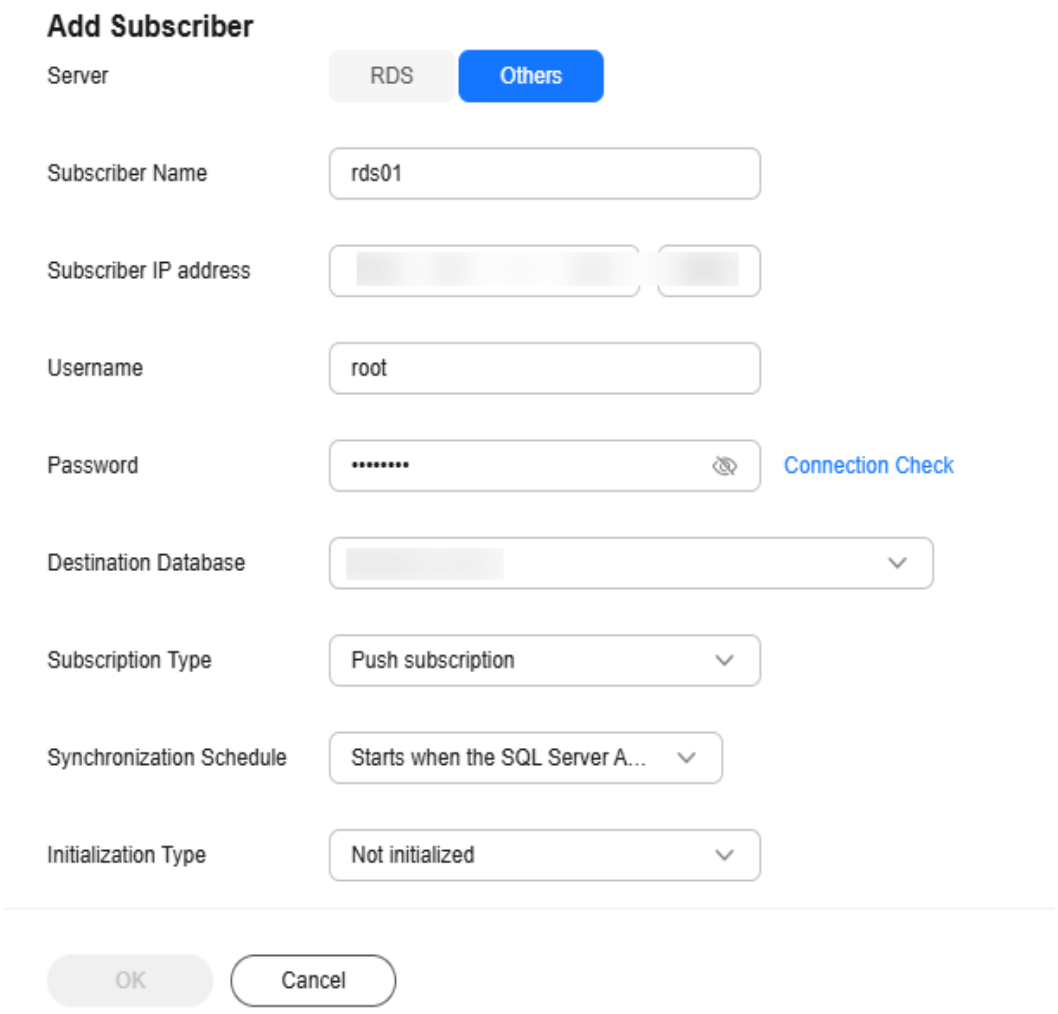
Figure 23-7 Adding an RDS subscriber



If you select **Others** for **Server**:

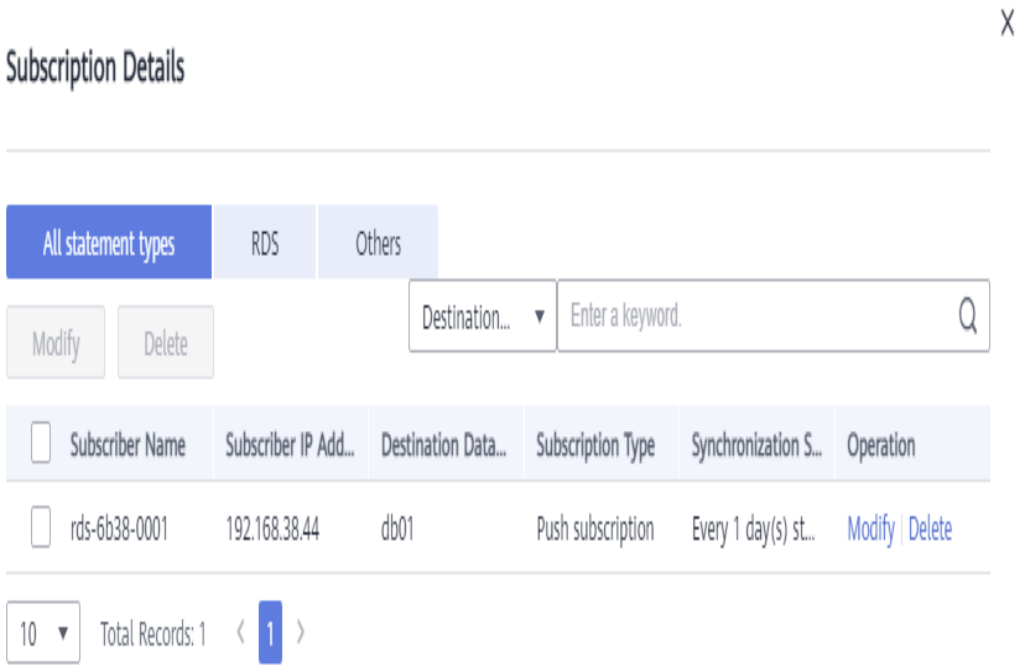
- Enter the subscriber name, subscriber IP address, port number, login username, and password, and specify destination databases.
- Select **Push subscription** for **Subscription Type**.
- Select a synchronization schedule for data subscription. You can customize a schedule by day, week, or month.

Figure 23-8 Adding other subscribers



Step 6 Locate the created publication, click the number in the **Subscribed Databases** column to view the subscription details.

Figure 23-9 Subscription details



----End

23.2 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 23-1 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**.

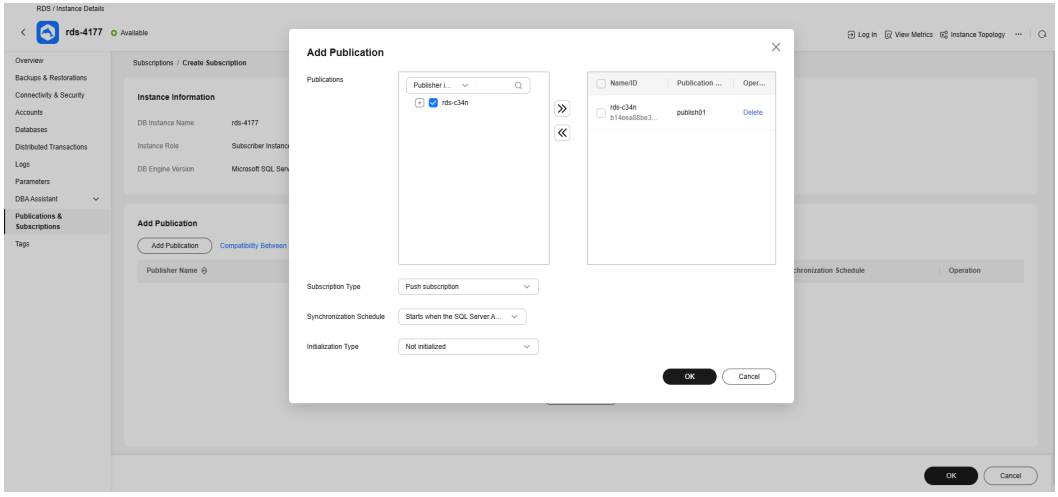
Step 4 Click **Add Publication**.

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.

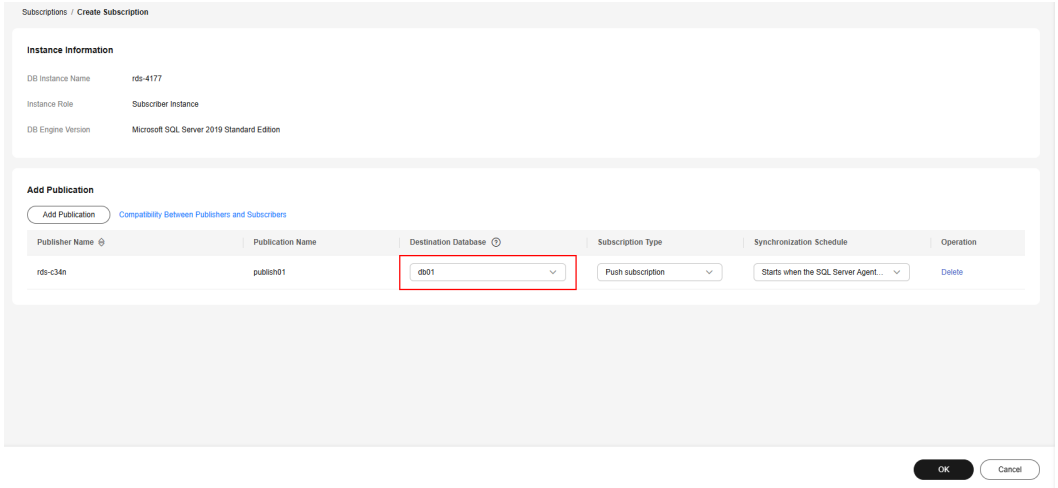
  For details about the publisher and subscriber types supported by RDS for SQL Server, see **Compatibility Between Publishers and Subscribers**.
- Select **Push subscription** for **Subscription Type**.
- Select a synchronization schedule for data subscription. You can customize a schedule by day, week, or month.

Figure 23-10 Adding a publication



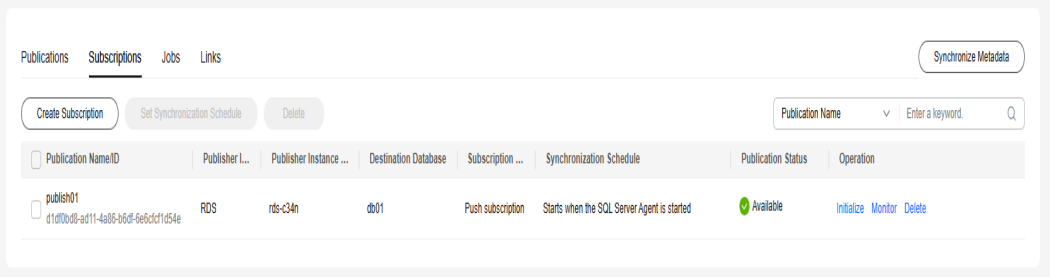
Step 6 Select a destination database that the publication is to be subscribed, and click **OK**.

Figure 23-11 Selecting a destination database



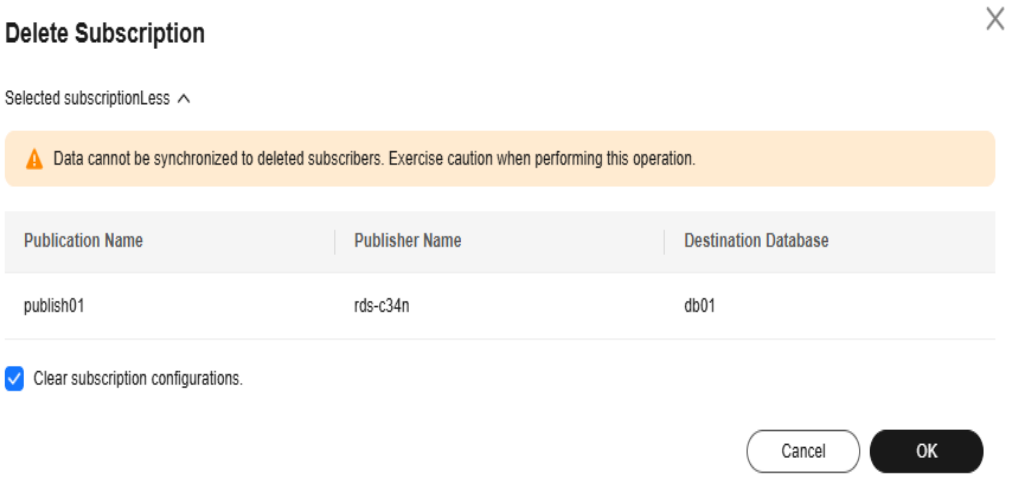
Step 7 View the created subscription.

Figure 23-12 Subscription



To delete the subscription, click **Delete**. When a subscription is deleted, its configurations can be deleted synchronously.

Figure 23-13 Deleting a subscription



----End

23.3 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.

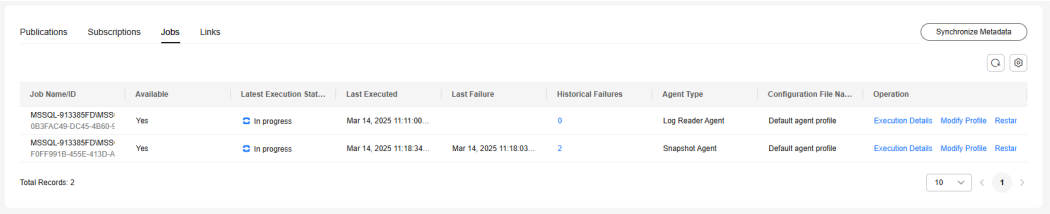
Checking Jobs

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 23-14 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

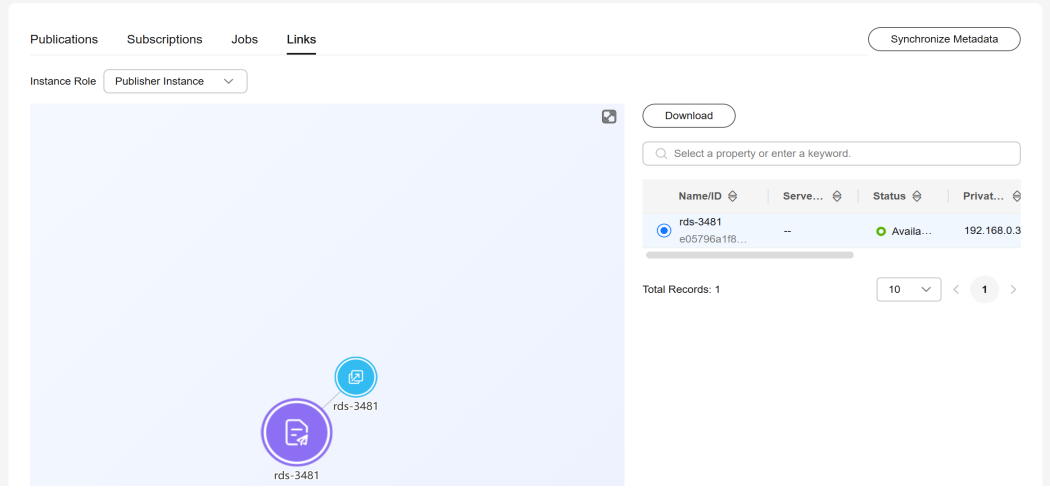
Checking Links

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 23-15 Links



----End

24 Task Center

24.1 Viewing a Task

You can view the progress and results of scheduled and instant tasks on the **Task Center** page.

Supported Tasks

Table 24-1 Supported tasks

Task Type	Category	Task Name
Instant tasks	Instance creation	Creating a SQL Server instance, creating a SQL Server read replica
	Instance lifecycle	Rebooting a SQL Server instance, stopping a SQL Server instance, starting a SQL Server instance, deleting a SQL Server instance
	Instance modifications	Scaling up a SQL Server instance, switching SQL Server primary and standby instances, cloning a SQL Server instance, changing the SQL Server instance type from single to primary/standby, changing a SQL Server instance class, changing a SQL Server storage type, migrating a standby SQL Server instance to another AZ, changing a SQL Server character set

Task Type	Category	Task Name
	Connection management	Creating a public domain name for a SQL Server instance, changing a public domain name for a SQL Server instance, creating a private domain name for a SQL Server instance, changing a private domain name for a SQL Server instance, binding an EIP to a SQL Server instance, unbinding an EIP from a SQL Server instance, updating an SSL certificate for a SQL Server instance
	Backup and restoration	Restoring to a new SQL Server instance, restoring to an existing SQL Server instance
	FileStream	Enabling FileStream for a SQL Server instance
	Security and encryption	Enabling TDE for a SQL Server instance, rotating TDE certificate for a SQL Server instance
Scheduled tasks	Instance lifecycle	Starting a SQL Server instance, upgrading a SQL Server instance system

Viewing an Instant Task

Step 1 Click  in the upper left corner and select a region.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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

24.2 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 Click  in the upper left corner and select a region.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 4**.
- If no, go to **Step 5**.

Step 4 Click **Cancel** in the **Operation** column. In the displayed dialog box, click **OK** to cancel the task. Click **Delete** in the **Operation** column. In the displayed dialog box, click **OK** to delete the task record.

Step 5 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

24.3 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 Click  in the upper left corner and select a region.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 4**.
- If no, no further action is required.

Step 4 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 5 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 a Maintenance Window**.

----End

25 Billing Management

25.1 Unsubscribing from a Yearly/Monthly Instance

Scenarios

To delete a DB instance billed on the yearly/monthly basis, you need to unsubscribe from the order. You can unsubscribe from a single instance by referring to **Unsubscribing from a Single DB Instance (Method 1)** and **Unsubscribing from a Single DB Instance (Method 2)** or unsubscribe from instances in batches by referring to **Unsubscribing from 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 from a Single DB Instance (Method 1)

Unsubscribe from a yearly/monthly DB instance or read replica on the **Instances** page.

Step 1 Click in the upper left corner and select a region.

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 or read replica and choose **More** > **Unsubscribe** in the **Operation** column.

Step 4 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

Step 5 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 6 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 from a Single DB Instance (Method 2)

Unsubscribe from a yearly/monthly DB instance or read replica on the **Billing Center** page.

Step 1 Click  in the upper left corner and select a region.

Step 2 Click  in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

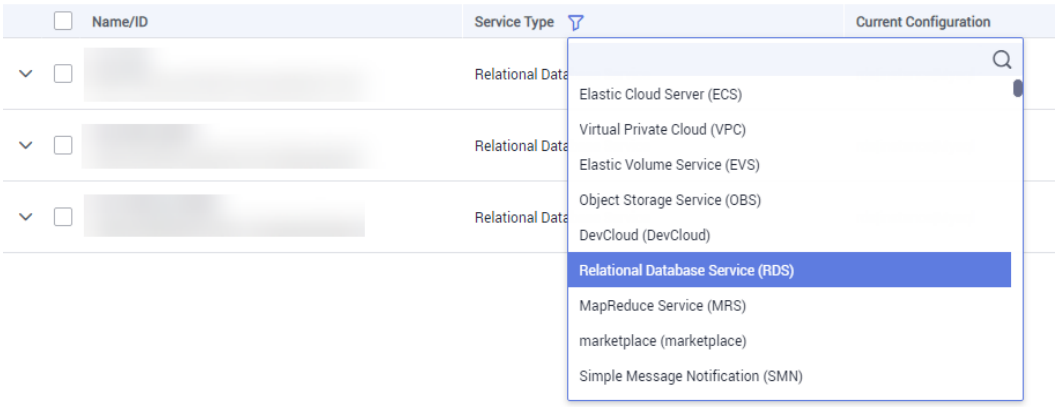
Step 3 In the upper right corner, click **Billing & Costs**.

Step 4 In the navigation pane, choose **Orders** > **Unsubscriptions**.

Step 5 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 25-1 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 6 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

Step 7 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 8 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

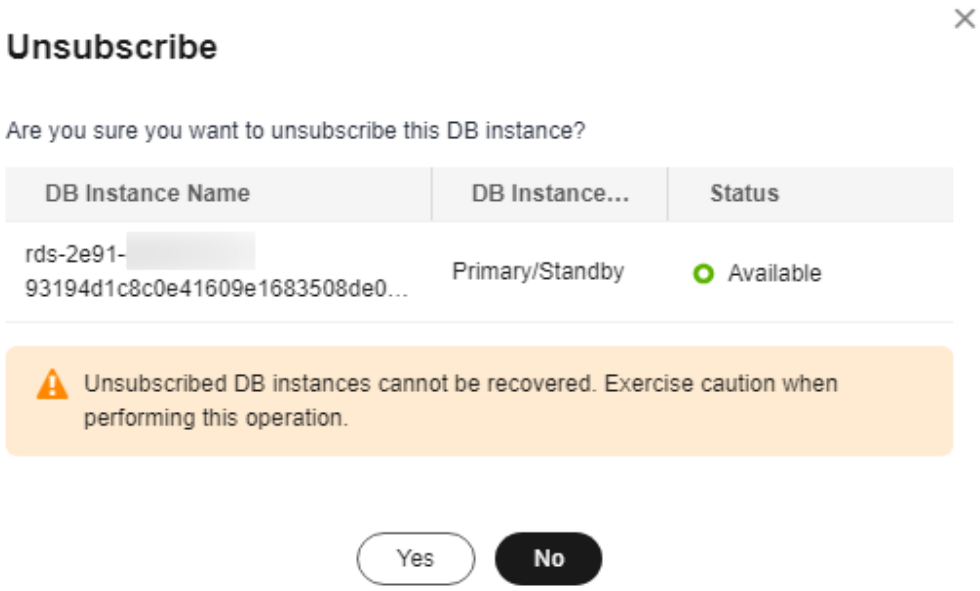
Unsubscribing from DB Instances in Batches

Step 1 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 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 25-2 Unsubscribing from yearly/monthly instances in batches



Step 4 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For details, see **Unsubscription Rules**.

Step 5 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 6 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

26 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-node 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-node 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 Click  in the upper left corner and select a region.

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 instance name. On the displayed **Overview** page, click **Enable** under **FileStream**.

Step 4 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 Click  in the upper left corner and select a region.

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 instance name. On the displayed **Overview** page, click **Disable** under **FileStream**.

Step 4 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**.

27 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 Click in the upper left corner and select a region.

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 instance name.

Step 4 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 5 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 6 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.

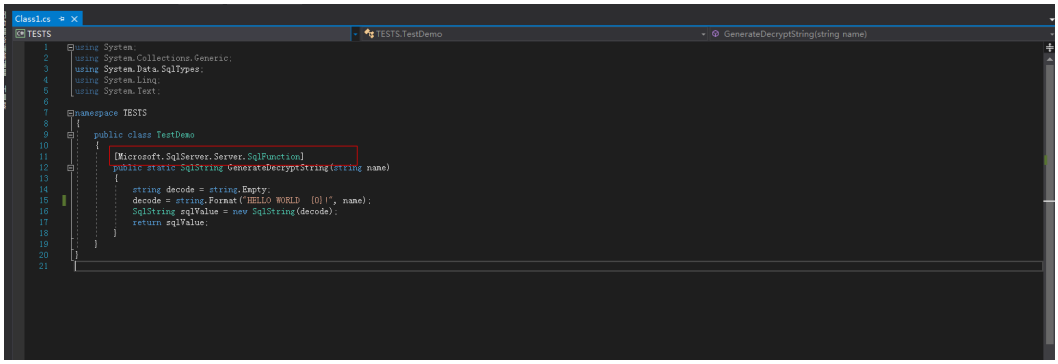
NOTE

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 27-1 C# function code

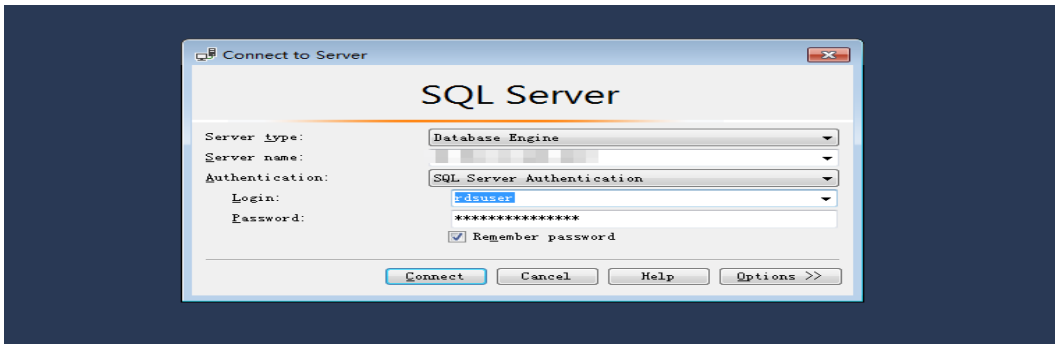


NOTICE

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 27-2 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 27-4**.

Figure 27-3 Creating an assembly

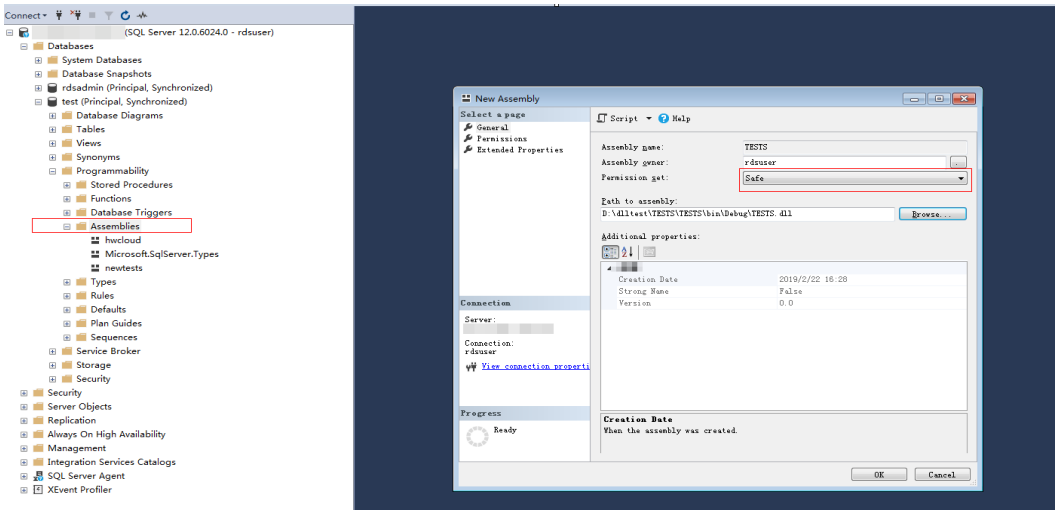
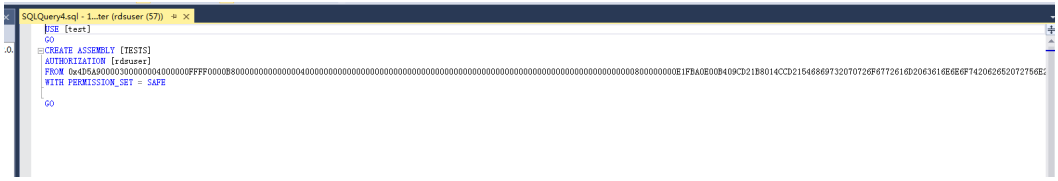


Figure 27-4 DLL file



Step 4 Execute the program. If the execution result is shown as **Figure 27-5**, the execution is successful. The TESTS assembly is added, as shown in **Figure 27-6**.

Figure 27-5 Execution result

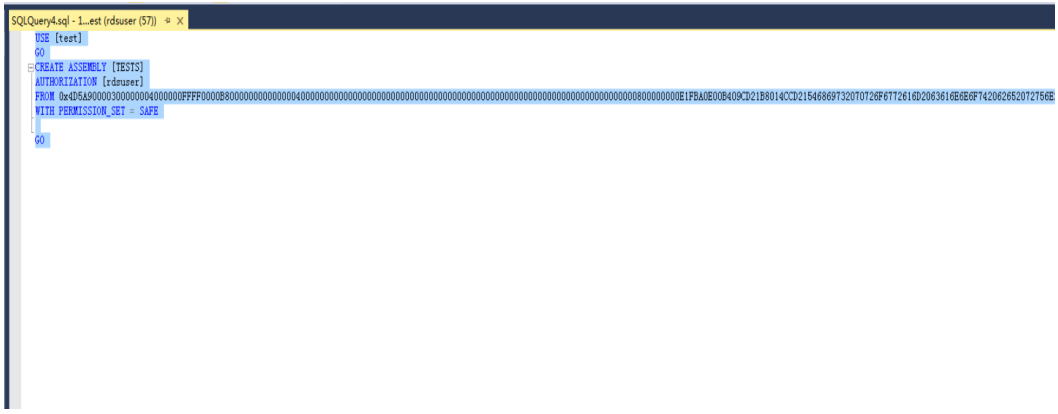
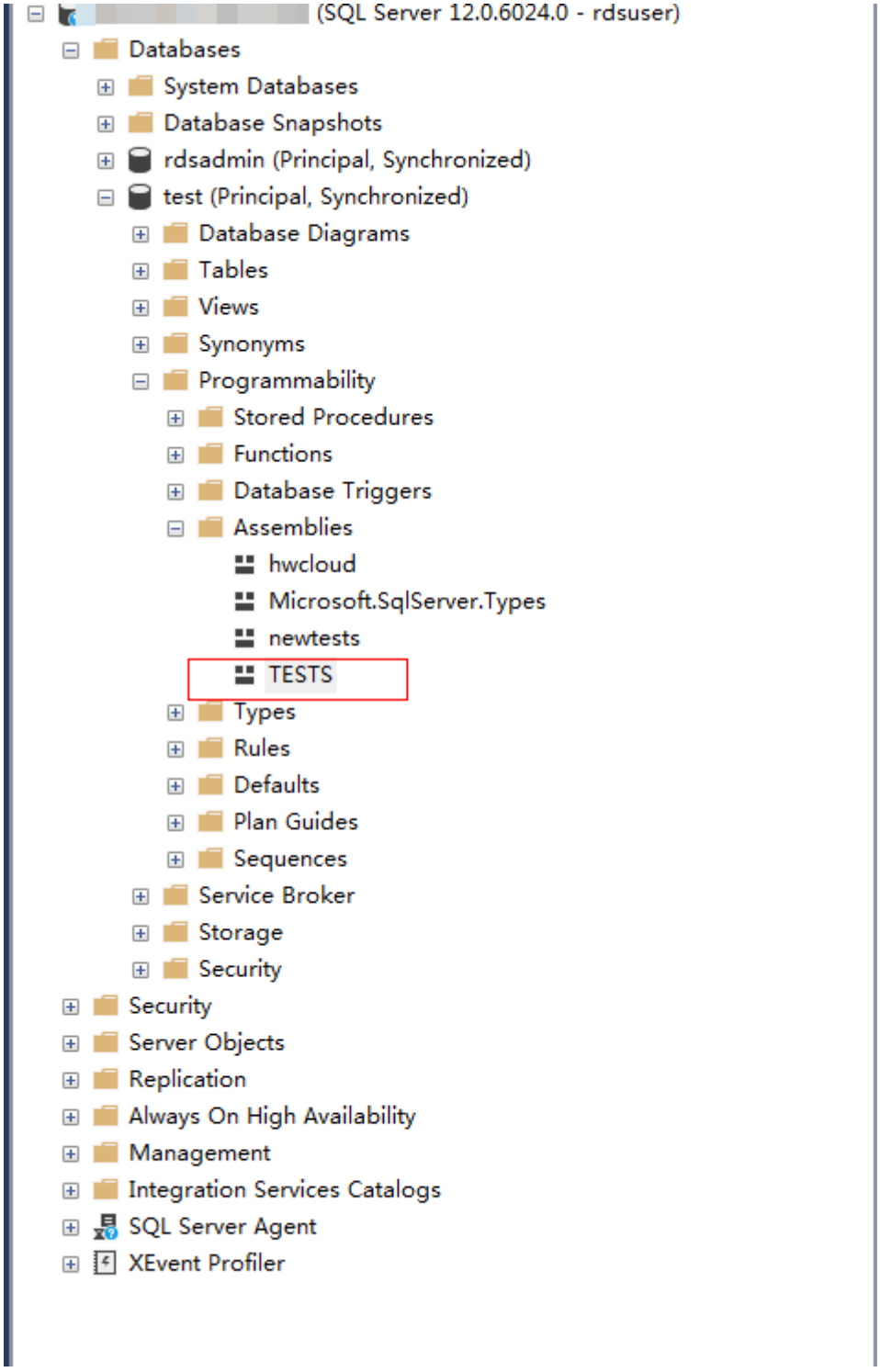


Figure 27-6 TESTS assembly



----End

28 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 Click in the upper left corner and select a region.

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 instance name.

Step 4 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 5 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 6 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 Click in the upper left corner and select a region.

Step 2 Click in the upper left corner of the page and choose **Databases** > **Relational Database Service**.

Step 3 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the target parameter template.

Step 4 On the displayed page, enter **default language** in the search box in the upper right corner.

Step 5 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 6 On the **Change History** tab, check that the value of **default language** has been successfully changed to **30**.

Step 7 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 8 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 9 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

29 Usage of Stored Procedures

29.1 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**.

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
```

29.2 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 a DB 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
```

29.3 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 a DB 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
```

29.4 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 an 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.

29.5 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 an 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';

29.6 Managing Trace Flags

Scenarios

You can use a stored procedure to manage trace flags. Trace flags can be used to:

- Obtain in-depth RDS for SQL Server information, such as Lock Manager lock operations.
- Change some preset RDS for SQL Server behaviors, such as disabling the query optimizer from determining the timeout period 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 from a Windows Server**.

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 trace 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.
  ```
- Trace flag operations only include **1**, **0**, and **-1**. If any other operation is performed, the system displays the following information:
  ```
  Just support Open:1 Close:0 Check:-1
  ```

Procedure

To manage a trace 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 a trace flag operation. The value **1** means enabling the trace flag, **0** means disabling the trace flag, and **-1** means viewing the trace flag.

For example, to enable trace flag 1117, run the following command:

EXEC msdb.dbo.rds_dbcc_trace 1117, 1;

29.7 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**.
- 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.

29.8 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 an 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';

29.9 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**.
- 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.

29.10 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**.

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.

29.11 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 an 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';

29.12 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**.

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.
```

29.13 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. You can connect to the DB instance through a 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
```


29.14 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. You can connect to the DB instance through a 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
```

29.15 Creating Alerts

Scenarios

You can use a stored procedure to create an alert.

Prerequisites

An RDS for SQL Server DB instance has been connected. You can connect to the DB instance through a 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 29-1 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.
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.

Parameter	Description
'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%**).
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**.

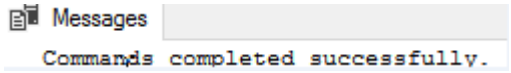
After the command is executed, the system displays the following information.

```
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.



29.16 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. You can connect to the DB instance through a 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 29-2 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.

Parameter	Description
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.

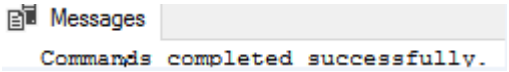
```
Commands completed successfully.
```

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.



29.17 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. You can connect to the DB instance through a 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 29-3 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.
'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.

Parameter	Description
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.
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**.

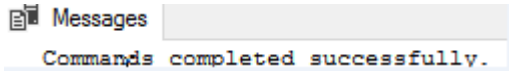
After the command is executed, the system displays the following information.

```
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.



29.18 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. You can connect to the DB instance through a 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 29-4 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.

Parameter	Description
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
'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.

Parameter	Description
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.
'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.

```
Messages
  Commands completed successfully.
```

29.19 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. You can connect to the DB instance through a 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 29-5 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.

Parameter	Description
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.
```

29.20 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. You can connect to the DB instance through a 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 29-6 Parameter description

Parameter	Description
'name'	The name of the operator to modify. This name must be unique and cannot contain the percent (%) character. **name** is **sysname**, with no default.
'new_name'	The new name for the operator. This name must be unique. **new_name** is **sysname**, with a default of **NULL**.
enabled	The current status of the operator. **enabled** is **tinyint**, with a default of **1** (enabled). If the value is **0**, the operator is not enabled and does not receive notifications.
'email_address'	The e-mail address of the operator. This string is passed directly to the e-mail system. **email_address** is **nvarchar(100)**, with a default of **NULL**.
'pager_number'	The pager address of the operator. This string is passed directly to the e-mail system. **pager_number** is **nvarchar(100)**, with a default of **NULL**.
weekday_pager_start_time	The time after which SQL Server Agent sends pager notification to the specified operator on the weekdays, from Monday through Friday. **weekday_pager_start_time** is **int**, with a default of **090000**, which indicates 9:00 A.M. on a 24-hour clock, and must be entered using the form HHMMSS.

Parameter	Description
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.
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**.

Parameter	Description
'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.
```

29.21 Removing Alerts

Scenarios

You can use a stored procedure to remove an alert.

Prerequisites

An RDS for SQL Server DB instance has been connected. You can connect to the DB instance through a 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 29-7 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.

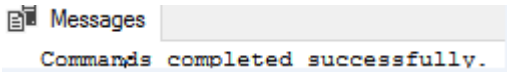
```
Commands completed successfully.
```

Example

EXEC [msdb].[dbo].[rds_delete_alert]

@name='test';

The command output is as follows.



29.22 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. You can connect to the DB instance through a 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 29-8 Parameter description

Parameter	Description
'alert'	The name of the alert. This parameter is of **sysname** data type, with no default value.
'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.
```

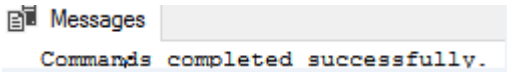
Example

EXEC [msdb].[dbo].[rds_delete_notification]

@alert_name = 'alert',

@operator_name = N'TestOperator';

The command output is as follows.



29.23 Removing Operators

Scenarios

You can use a stored procedure to remove an operator.

Prerequisites

An RDS for SQL Server DB instance has been connected. You can connect to the DB instance through a 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 29-9 Parameter description

Parameter	Description
'name'	The name of the operator to delete. This parameter is of **sysname** data type, with no default value.
'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.
```

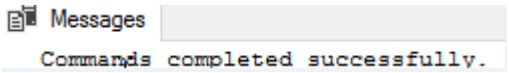
Example

EXEC [msdb].[dbo].[rds_delete_operator]

@name = N'Test01',

@reassign_to_operator = NULL;

The command output is as follows.



29.24 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 a 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 29-10 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 29-11 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.

log_reuse_wait_desc Value	Description
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.

29.25 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. You can connect to the DB instance through a SQL Server client. For details, 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' ;

29.26 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. You can connect to the DB instance through a SQL Server client. 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';

30 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

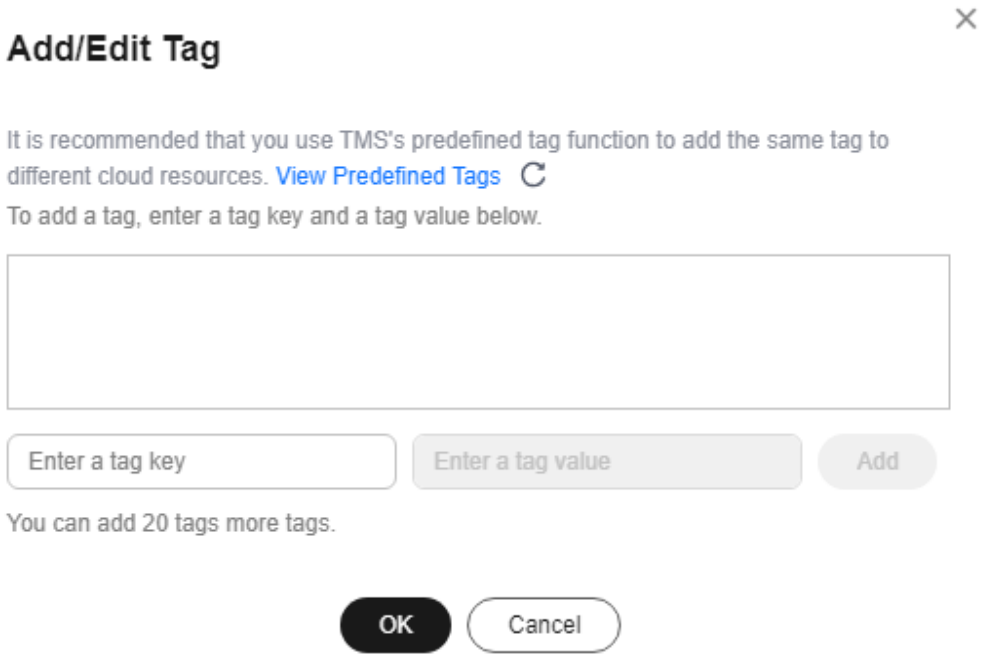
Step 1 Click  in the upper left corner and select a region.

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 instance name.

Step 4 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 30-1 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 5 After a tag has been added, view and manage it on the **Tags** page.

----End

Deleting a Tag

Step 1 Click in the upper left corner and select a region.

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 **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

31 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.

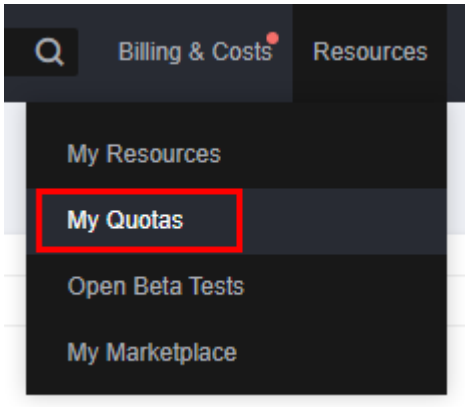
Viewing Quotas

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, choose **Resources** > **My Quotas**.

Figure 31-1 My quotas



Step 4 On the **Quotas** page, view the used and total quotas of each type of resources.

----End

Increasing Quotas

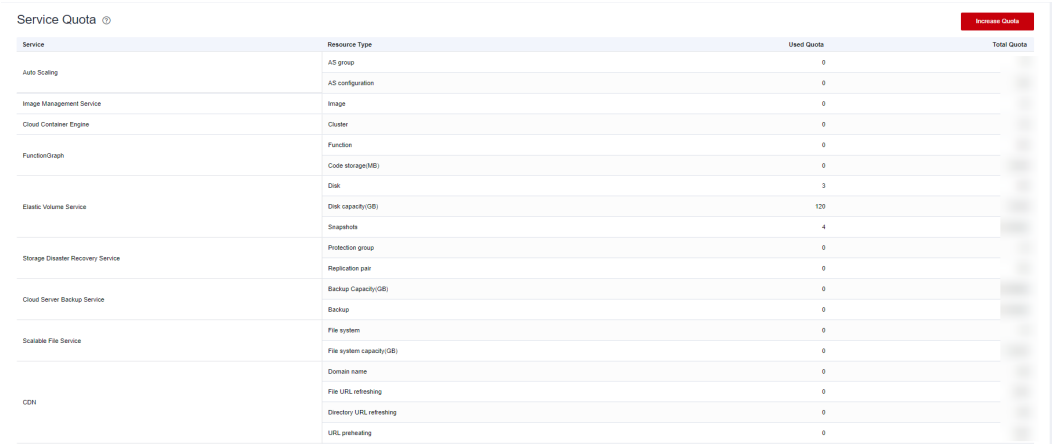
Step 1 Log in to the management console.

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

Step 3 In the upper right corner, choose **Resources** > **My Quotas**.

Step 4 In the upper right corner of the page, click **Increase Quota**.

Figure 31-2 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

