ApsaraDB for RDS

User Guide

User Guide

Preface

Document overview

The ApsaraDB Relational Database Service (RDS) is a stable and reliable online database service with auto-scaling capabilities. Based on Apsara’ s distributed file system and high-performance storage technology, RDS supports MySQL, SQL Server and PostgreSQL, and provides a complete set of solutions for disaster recovery, backup, restoration, monitoring, migration and others to free you from the burden of operating and managing your own database.

This document describes how to configure RDS through the RDS console, and help you better understand RDS features and functions. You may also manage the RDS through APIs and SDKs.

If you need operators' help, you can click **Ticket Service > Submit Ticket** on the RDS console or Click Here to submit a ticket.

For more information about functions and pricing of the ApsaraDB, please log in to the Official Website of ApsaraDB.

Declaration

Some product features or services described in this document may not be available at certain regions. Refer to the relevant commercial contracts for specific Terms and Conditions.

This document serves as the user guidance. No content in this document shall constitute any express or implied warranty.

Due to product version upgrade or other reasons, the content of this document will be updated as needed. Ensure that the document version is consistent with the corresponding software version.

Note

RDS includes multiple types of databases. This document takes the MySQL database as an example to describe the features and usage of all RDS products. Some types of databases may not include certain features. The actual page shall prevail.

General description convention

Description	Note
Local database/Source database	Refers to the database deployed in the local equipment room or the database not on the ApsaraDB.In most cases, it refers to the source database to be migrated to the ApsaraDB in this document.
RDS for XX (XX is MySQL, SQL Server, PostgreSQL, or PPAS) indicates the RDS of a specific database type, for example, RDS for MySQL means the instance enabled on the RDS and whose database type is MySQL.	

Quick start

If you use ApsaraDB for the first time, refer to **ApsaraDB Quick Start** documents to familiarize RDS features and procedures on migrating your local database to the RDS.

ApsaraDB Quick Start (MySQL)

ApsaraDB Quick Start (SQL Server)

ApsaraDB Quick Start (PostgreSQL)

If you have questions beyond *Quick Start*, refer to this document.

Login and logout

RDS instances can be managed through the RDS console. This chapter describes how to login and log out to the RDS Console and access the specific instance management console interface to perform subsequent instance management and control operations.

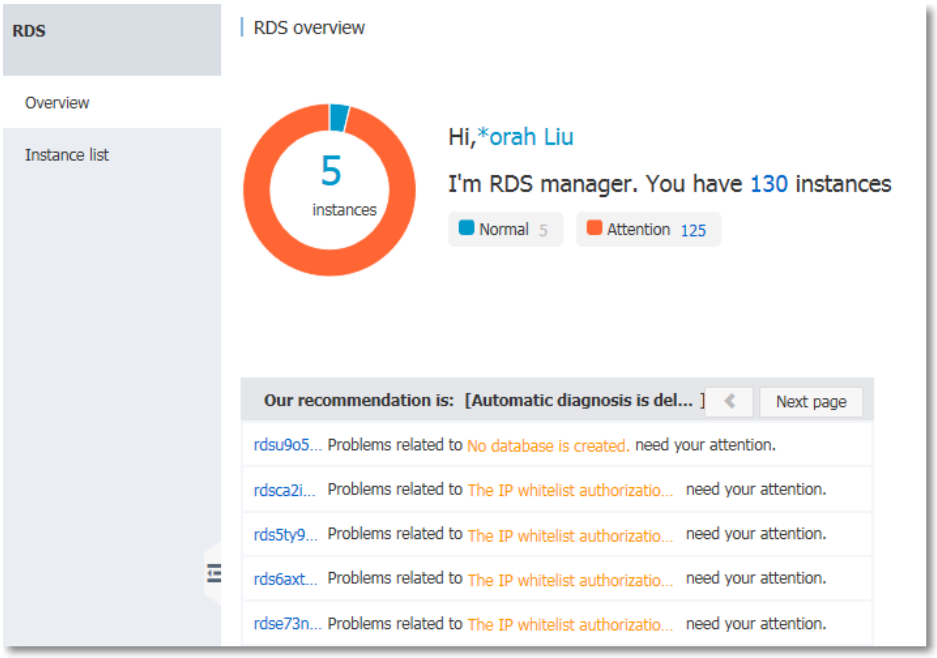
Prerequisites

Before logging on to the RDS Console, you need to purchase the RDS instance. For the method of buying a RDS instance, please refer to Purchase. For details on the pricing schema, refer to RDS Price.

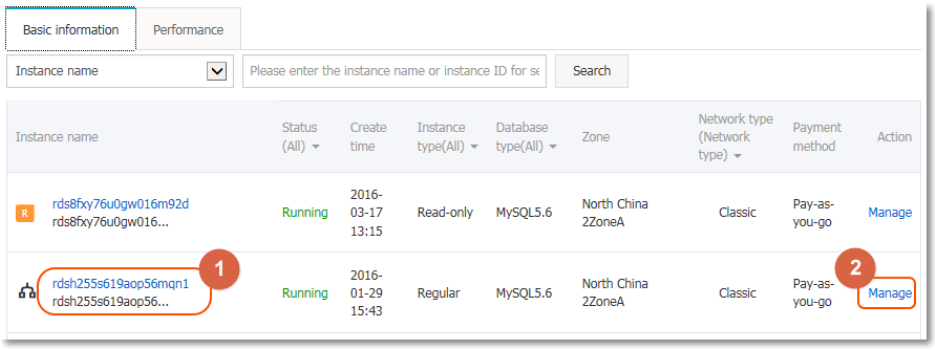
Log on to the RDS console

Log on to the RDS Console.

The system displays the **RDS Overview** interface, as shown in the figure below.



Select **Instance List** in the menu, and click **Instance Name** of the database or the corresponding **Manage** button to access the instance management interface, as shown in the figure below.



At the specific instance management console, you can manage the instance account and database, and configure instance parameters, etc.

Log out of the RDS console

You can log out of the RDS console in the following two methods:

Click user information in the upper right corner. In the displayed menu, click **Exit**.

Close the browser.

Instance management

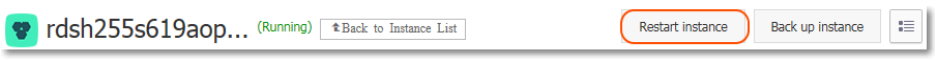
Restart an instance

Note: When restarting an instance, the connection will be interrupted during the process. Therefore, before restarting an instance, arrange your production first and restart with caution.

Operation procedure

Log on to the RDS console and select the target instance.

Click **Restart Instance** in the upper right corner on the instance management page. In the displayed dialog box, click **OK**, as shown in the figure as follows.



Set parameters

RDS allows you to define some instance parameters. For details about the parameters that can be configured, see *Parameter Settings* on the *RDS Console* .

Considerations

Configure parameters only within the permissible value ranges shown on the parameter settings page.

The instance must be restarted after modifying certain parameters. Refer to the parameter settings page to confirm if a restart is required. Before restarting, to avoid any interruption of production, ensure the appropriate business arrangements.

Background information

RDS is compatible with the native database service, with similar parameter setting methods. You can configure parameters through the RDS console by referring to this example, or through APIs to run related commands.

For a description of the database parameters, click the following link to review the official documents of different database versions.

- MySQL
  - MySQL 5.5
  - MySQL5.6
  - MySQL 5.7
- SQL Server
  - SQL Server 2008 R2

Operation procedure

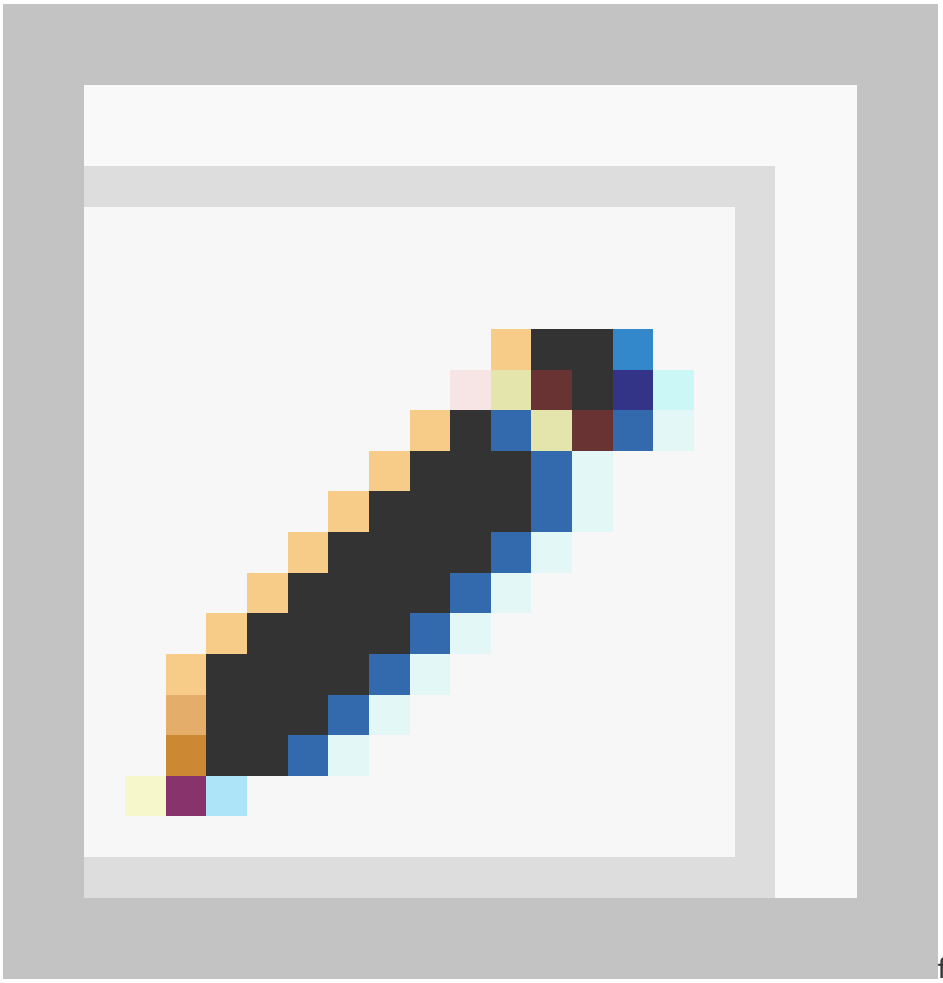
Log on to the RDS Console and select the target instance.

Select **Parameter Settings** in the instance menu.

Select the parameter modification method.

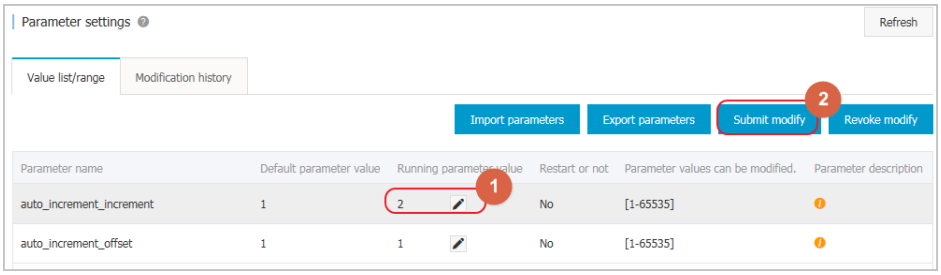
To modify a parameter

Click the



following th
target value and click **OK**, as indicated by 1 in the image below.

Click **Submit modify** to confirm the setting, as indicated by 2 in the image below.



To batch modify parameters

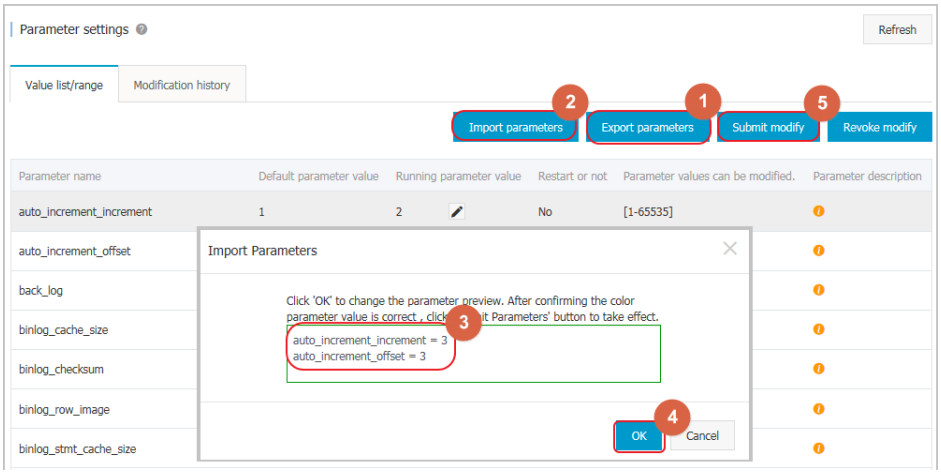
Click **Export Parameters** to export the parameter file (.txt) to your local device, as indicated by 1 in the image below.

Open the parameter file and batch modify the relevant parameters.

Click **Import Parameters**, as indicated by 2 in the image below.

In the **Import Parameters** window, paste the parameters to modify and the parameter values and click **OK**, as indicated by 3 and 4 in the image below.

Confirm the parameter modification results in the parameter list and click **Submit modify**, as indicated by 5 in the image below.



View the modification history

You can review the modification records on the *Modification History* tab page.

Configure the maintenance period

ApsaraDB needs to be regularly maintained to ensure overall instance health in production environment. You can set the maintenance period in the idle service hours based on service regularities to prevent the potential interruptions for production during maintenance. RDS will perform regular maintenance within the period you have configured.

Background information

To ensure the stability and efficiency of ApasaraDB RDS instances on Alibaba Cloud platform, the backend system perform a serial of maintenance tasks at irregular basis and as needed.

Before official maintenance, RDS sends text messages and emails to contacts configured by your Alibaba Cloud account.

To ensure stability during maintenance process, instances enter the **Instance being maintained** state before the preset maintenance period on the day of maintenance. When an instance is in this state, **normal data access to the database is not affected**. However, apart from account management, database management, adding IP addresses to the white list, and other services associated with changes (such as common operations including upgrade, degrade, and restart) are unavailable on the console of this instance. Query services such as performance monitoring are available.

When the maintenance period preset by an instance begins, transient disconnection will occur once or twice on the instance during this period. You must ensure that the application program supports the reconnection mechanism. After transient disconnection, the instance will restore to the normal state.

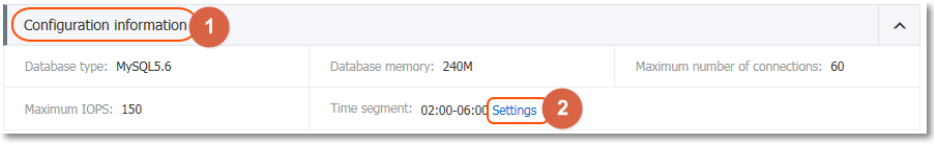
Operation procedure

Log on to the RDS Console and select the target instance.

Select **Basic information** in the menu.

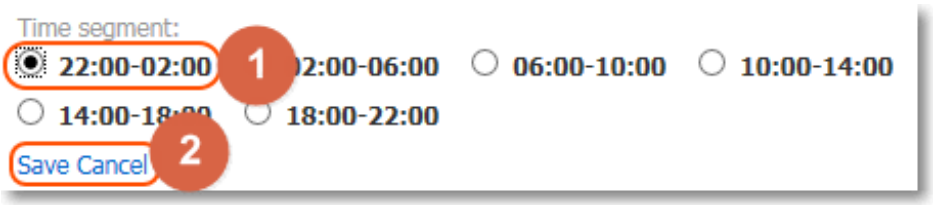
Click **Setting** behind *Time segment* in *Configuration information*, as shown in the figure below.

The default maintenance period of the RDS is from 02:00 to 06:00.



Select the maintenance period and click **Save**, as shown in the figure below.

Note: Time segment is the time in Beijing.



Instance migration across zones

If the zone of an instance is fully loaded or in other cases where the instance performance is affected, you can migrate the instance to another zone.

Background information

There will be 30 seconds of transient disconnection during migration across zones. Ensure that your application has a reconnection mechanism.

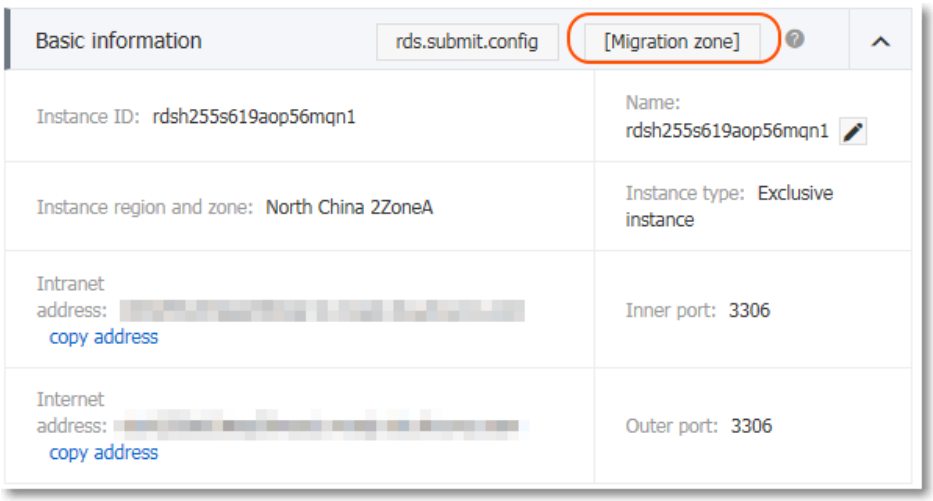
Migration across zones is possible only when the region of an instance has multiple zones.

Operation procedure

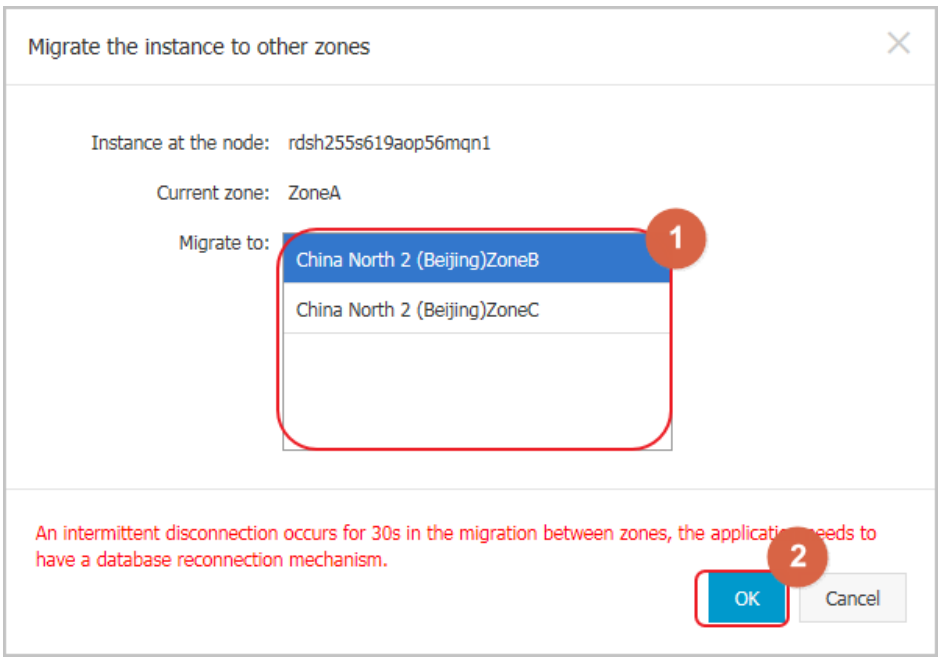
Log on to the RDS Console and select the target instance.

Select **Basic Information** in the menu.

Click **Migration zone** in *Basic Information*, as shown in the figure below.



Select a target zone on the *Migrate the instance to another zones* page and click **OK**, as shown in the figure below.



Switch master/slave instances

This section describes how to switch the master/slave instances.

Notice

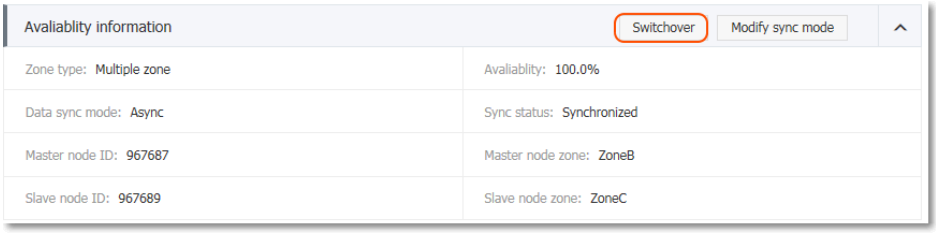
Switching the master/slave instances may result in transient disconnection. Ensure that your application has a reconnection mechanism.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Instance Availability** in the menu.

Click **Switchover** in *Instance Availability*, as shown in the figure below.



Click **Confirm** to switch the instance.

Modify the data replication method

You can select different data replication modes to improve RDS availability based on your business needs.

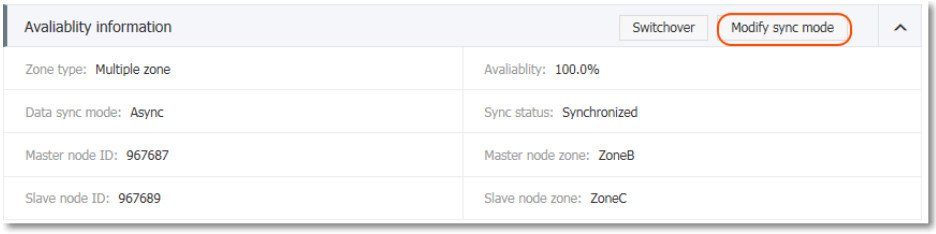
Note: This operation is only applicable to the database of MySQL 5.5/5.6.

Operation procedure

Log on to the **RDS Console** and select the target instance.

Select **Service Availability** in the menu.

Click **Modify Data Replication Mode** in *Instance Availability*, as shown in the figure below.

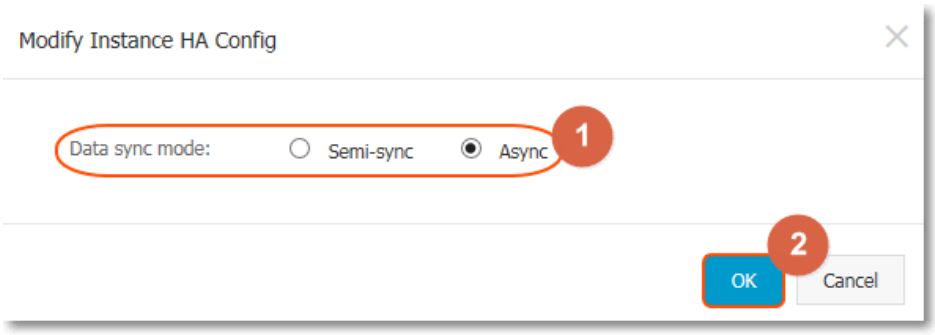


On the *Modify Data Replication Mode* page, select a data replication mode and click **OK**, as shown in the figure below.

Semi-synchronous replication: Normally data is replicated in the forced synchronous replication mode. If an exception (unavailability of the Slave node or network exception between the Master and Slave nodes) occurs when the Master node replicates data to the Slave node, the Master node will suspend response to

the application until the replication mode times out and degrades to asynchronous replication mode. If the application is allowed to update data for a given time, unavailability of the Master node will cause data inconsistency. When data replication between the two nodes resumes normally (the Slave node or network connection is recovered), asynchronous replication will be changed to forced synchronous replication mode. The length of time to restore to the forced synchronous replication mode depends on the implementation mode of semi-synchronous replication. ApsaraDB for MySQL5.5 is different from ApsaraDB for MySQL5.6 in this regard.

Asynchronous replication: The application initiates an update (including the Add, Delete, and Modify operations) request. After completing the corresponding operation, the Master node immediately responds to the application and then replicates data to the Slave node asynchronously. Therefore, in the asynchronous replication mode, unavailability of the Slave node does not affect the operation on the primary database, but there is a slight probability for unavailability of the Slave node to cause data inconsistency.



Release an instance

Background information

RDS instances support two payment methods: **Subscription** and **Pay-As-You-Go**.

Subscription: Instances paid by yearly or monthly subscription cannot be voluntarily deleted or released. When the service you purchased expires, the RDS instance will be locked and cannot be read or written. You must renew your subscription to continually use the instance. If you do not renew the subscription, the instance will continuously run for 15 days after the end of the service period and then the instance data will be kept for another 15 days. If you

still do not renew the subscription, the instance will be released without recovery. Therefore, you need to back up your data and migrate it from the RDS prior to the 6-day period to avoid any data loss. If there are read-only instances under the master instance, they will also be released.

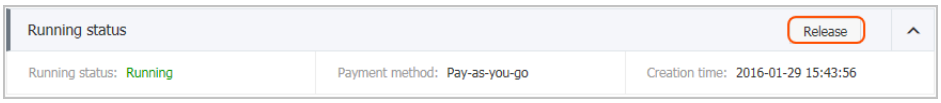
Pay-As-You-Go: These instances can be voluntarily released.

Operation procedure

Log on to the RDS Console and select the target instance.

Click the ID of the target instance to go to the “Basic information” page.

In *Running status* part, click **Release instance**, as shown below.



In the pop-up box, click **Confirm** to release the instance.

Upgrade database

Background Information

ApsaraDB for RDS allows users with lower database versions to upgrade to a higher version, but not downgrade to a lower version. Refer to the console interface for upgradeable versions.

Note: Currently this operation is only applicable to upgrading MySQL Database from version 5.5 to version 5.6.

Operation Procedure

Note:

We suggest you firstly purchase an instance with the database version you wish to upgrade to, in order to test its compatibility before upgrading.

During the database upgrading process, the RDS service may flash off for about 30 seconds. To avoid the impacts on your production, it is recommended that you upgrade the database at the low peak of the service, or make sure that your application has the automatic reconnection mechanism.

Log on to the RDS Console.

Select the region where the target instance is located.

Click the ID of the target instance to go to the “Basic information” page.

In the “Configuration information” part, click **Upgrade database** after “Database type”.

On the **Database version upgrade** page, select the database version to upgrade and click **Start upgrading**.

Account management

Create an account

Before using the database, you need to create an account in the RDS instance. This section describes how to create an account in RDS.

Background information

Databases under a single instance share all the resources of this instance.

MySQL and SQL Server instances support the creation of up to 500 accounts.

PostgreSQL and PPAS instances have no limit on the number of accounts.

You can create only an initial account on MySQL 5.7 and SQL Server 2012.

The account on MySQL 5.5/5.6 supports both of the traditional mode and the autonomous mode to

manage the instance. You can upgrade from the traditional mode to the autonomous mode, but the rollback is not supported. The mode will be changed to the autonomous mode automatically when the high-privilege account is created.

	Traditional mode	Autonomous mode (After creating the high-privilege account)
Supported engine and version	MySQL 5.5/5.6 SQL Server 2008R2	MySQL 5.5/5.6/5.7 SQL Server 2012 PostgreSQL 9.4 PPAS 9.3
Features	It is the first mode supported by RDS. All the accounts (User)/databases (DB) are created and managed through the Open API or RDS console.	It is a new mode. It is a totally autonomous management mode and provides higher privilege, making the account and privilege management more free and flexible. There are two types of accounts: the initial account (the high-privilege account) and the ordinary account.
Account number	Up to 500.	No limit.
Database number	MySQL: Up to 500. SQL Server: Up to 50.	No limit.
Methods to create accounts	OpenAPI/RDS console	Initial account: OpenAPI/RDS console Ordinary account: Execute SQL
Methods to create databases	OpenAPI/RDS console	Execute SQL
OpenAPI: account management	CreateAccount DeleteAccount DescribeAccounts GrantAccountPrivilege RevokeAccountPrivilege ModifyAccountDescription ResetAccountPassword	CreateAccount DeleteAccount DescribeAccounts ModifyAccountDescription ResetAccountPassword
OpenAPI: database management	CreateDatabase DeleteDatabase DescribeDatabases ModifyDBDescription	Not support.
Privilege management	Only including two kinds of the privileges: **read/write** and **read only**.	You can take full use of the privilege management advantages of the database engine, such as assigning the query permissions of different tables to different users.

Note:

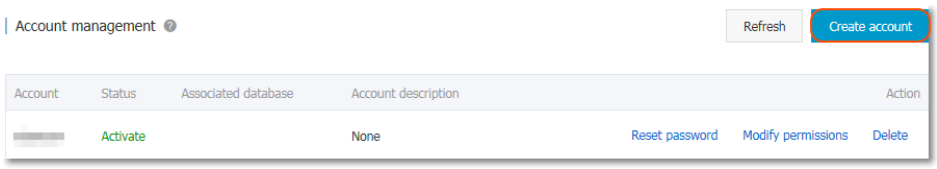
When assigning database account permissions, follow the minimum permission principle and service roles to create accounts and rationally assign Read-Only and Read/Write permissions. When necessary, you may split database accounts and databases into smaller units so that each database account can only access data for its own services. If you do not need to write data to a database, assign Read-Only permission.

Use strong passwords for database accounts and change the passwords on a regular basis.

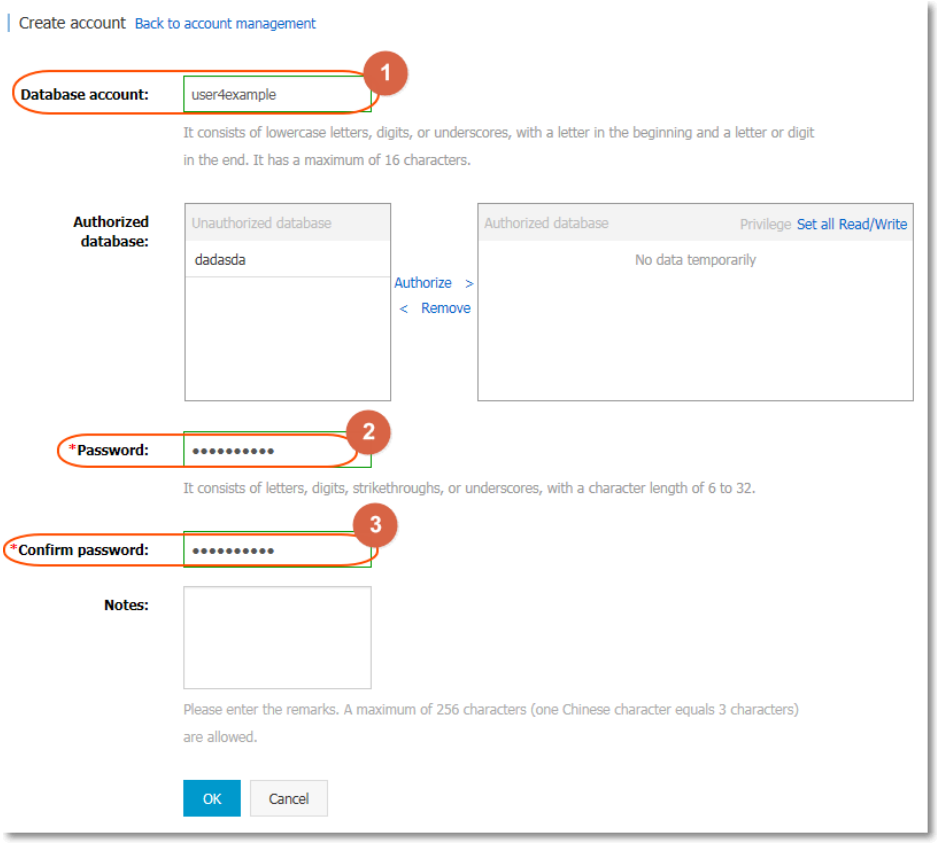
Operation procedure

Log on to the RDS Console and select the target instance.

Select **Account management** in the menu, and click **Create Account**, as shown in the figure below.



Enter the information of the account to create, and click **OK**, as shown in the figure below.



Database account: Consists of 2 to 16 characters (which can be lowercase letters, digits or underscores). It must begin with a letter and end with a letter or digit, for example, user4example*.

Authorized database: Refers to the authorized database of this account. Select **Unauthorized Database** on the left, and click **Authorize** to add the database to **Authorized Database**. This field can be blank if no database has been created.

You can click the permission setting button on the upper-right corner of *Authorized Database* to batch set the permissions of the databases under this account to **All Read/Write** or **All Read Only**.

Password: Refers to the password corresponding to this account. The password consists of 6 to 32 characters which may be letters, digits, hyphens or underscores, for example, *password4example*.

Confirm password: Enters the password again, for example, *password4example* to ensure that a correct password is entered.

Remarks: This field can be used to store additional information relevant to the database to facilitate management. A maximum of 256 characters can be entered

(1 Chinese character is considered 3).

Reset instance password

When using RDS, if you forget the password of the database account, you can reset the password on the RDS Console.

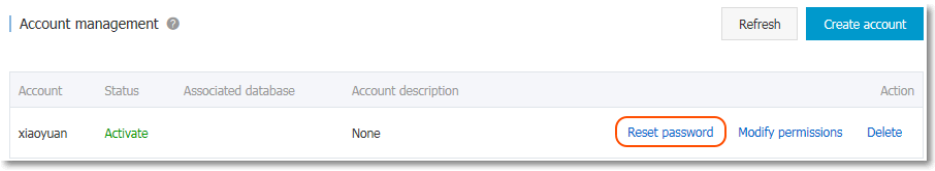
Note: For data security considerations, it is recommended to change the password periodically.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Account management** in the menu.

On the *Account list* tab page, click **Reset password** next to the account for which you want to reset password, as shown in the figure below.



On the *Reset Account Password* page, enter a *new password* and click **OK**.

Note: The password consists of 6 to 32 characters which must be letters, digits, hyphen or underscores. Old passwords are not recommended.

Change account permissions

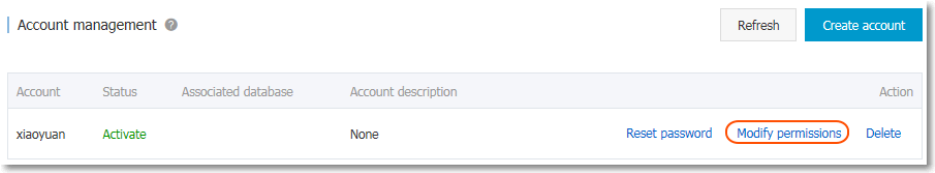
When using RDS, you can change permissions of the account in the instance based on actual requirements at any time.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Account Management** in the menu.

On the *Account list* page, click **Modify permissions** after the account of which the permissions will be changed, as shown in the figure below.

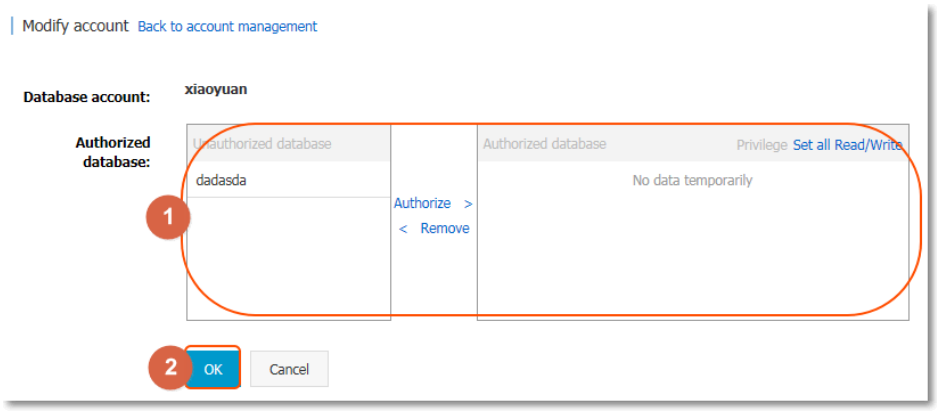


On the *Modify account* page, change the account permissions and click **OK**, as shown in the figure below.

Add an authorized database: Select a database in *Unauthorized database* and then click **Authorize >** to add it to *Authorized database*.

Delete an authorized database: Select a database in *Authorized database* and then click **< Remove** to add it to *Unauthorized database*.

Change permissions of *Authorized database*: Select a database in *Authorized database*, and then click the permission setting button in the upper-right corner of *Authorized database* to batch set the permissions of the databases under this account to **All Read/Write** or **All Read Only**.



Authorize a service account

If you are seeking for technical supports from Alibaba Cloud and if it is necessary to operate your database instance during technical support, you need to authorize a service account that will be used by the technical support staff to provide technical support services.

Background information

When you authorize the service account to view and modify configuration or view table structure, index and SQL, the system will generate a temporary service account and the corresponding permissions are given to this account according to your authorization information.

This temporary service account will be automatically deleted after the validity period of authorization expires.

Operation procedure

Log on to the RDS console and select the target instance.

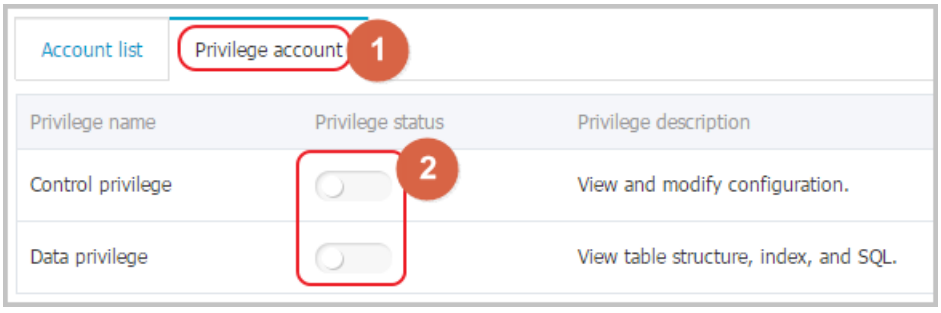
Select **Account management** in the left-side menu to go to the *Account Management* page.

Select the **Privilege account** tab page.

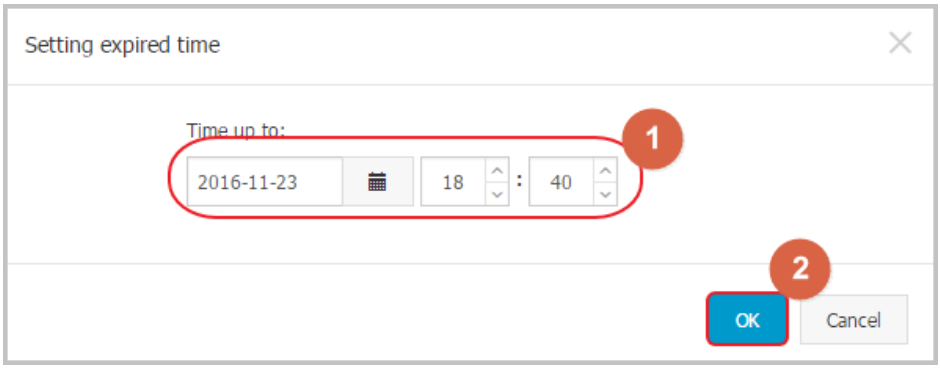
Select the permission to be authorized to the service account and click the button in the *Privilege status* column, as shown in the figure below.

For troubleshooting of the IP white lists, database parameters and other problems, you need to authorize **Control privilege** only.

For the database performance problems caused by your application, you need to authorize **Data privilege**.

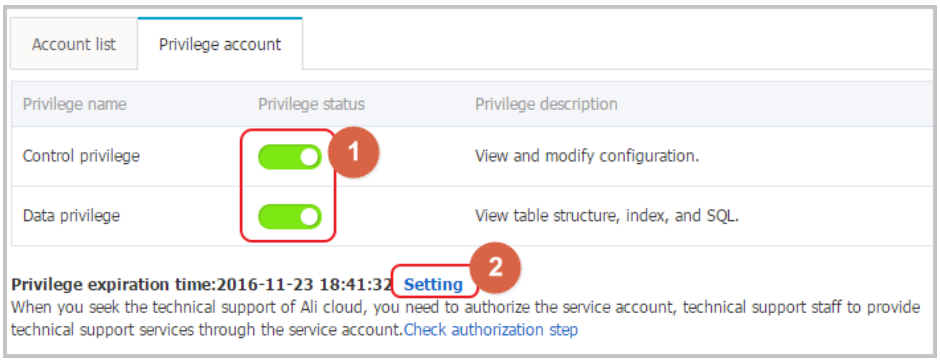


After setting the permission expiration time on the page of *Setting expired time*, click **OK** as shown in the figure below.



Subsequent operations

After a service account is authorized, you may cancel the authorization (as shown in Fig.1 below) or change the authorization validity period (as shown in Fig.2) on the *Privilege account* tab page.



Database management

Create a database

Users can create databases using the RDS Management Console. Database names are unique within an instance, but can be duplicate across instances.

> If you use MySQL 5.7, refer to Creating a database and an account (MySQL 5.7) to create a database using a client.

If you use SQL Server 2012, refer to Creating a database and an account (SQL Server 2012) to create a database using a client.

Background information

Databases under a single instance share all the resources of this instance.

MySQL instances support the creation of up to 500 databases.

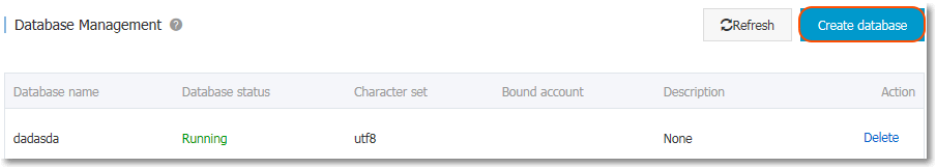
SQL Server 2008 R2 instances support the creation of up to 50 databases.

PostgreSQL and PPAS version instances have no limit on the number of databases.

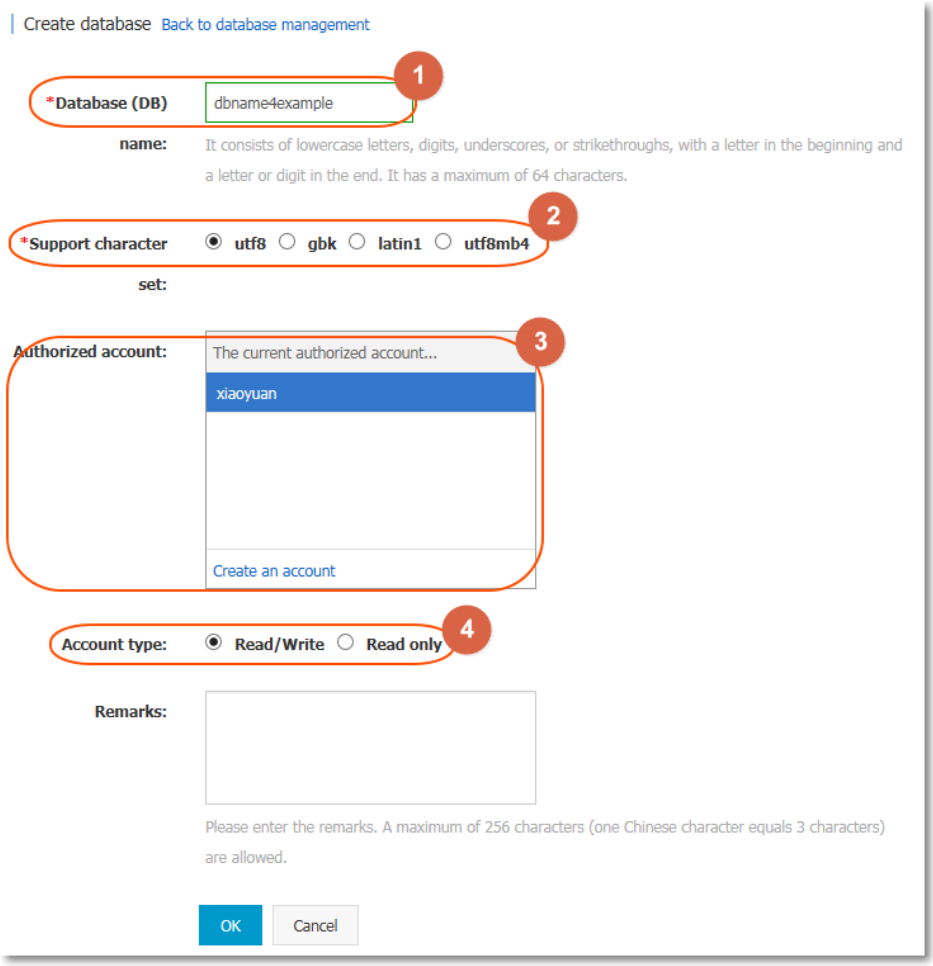
Operation procedure

Log on to the RDS Console and select the target instance.

Select **Database Management** in the menu, and click **Create database**, as shown in the figure below.



Enter the information of the database you want to create, and click **OK**, as shown in the figure below.



Database (DB) name: Contains 2 to 64 characters which consist of lowercase letters, digits, underscores, or hyphens. It must begin with a letter and end with a letter or digit, e.g., *dbname4example*.

Support character set: utf8, gbk, latin1 and utf8mb4.

Authorized account: Select an account authorized by this database. This field can be blank if no account has been created.

Account type: This option is visible after **Authorized Account** is selected. Set the permission authorized by this database to **Authorized Account**, which can be set to **Read/Write** or **Read only**.

Remarks: This field can be used to store additional information relevant to the database to facilitate management. A maximum of 256 characters can be entered (1 Chinese character is considered 3).

Delete a database

This section describes how to delete a database from an instance on RDS.

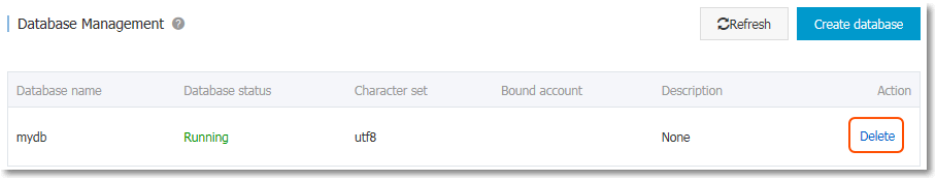
Note: If you use RDS for SQL Server 2012, delete the database using a client.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Database Management** in the menu, and the databases in this instance are listed on this page.

Select the database you want to delete, click **Delete**, and then click **OK** in the pop-up dialog box, as shown in the figure below.



Network management

Set access modes

RDS supports two access modes: *standard mode* and *safe connection mode*. This chapter describes the differences between the two access modes and the switching method.

Note:

The defaul access mode of MySQL 5.7 and SQL Server 2012 database is *standard mode*. The switch between *standard mode* and *safe connection mode* is not supported.

> The defaul access mode of PPAS and PostgreSQL database is *safe connection mode*. The switch between *standard mode* and *safe connection mode* is not supported.

The differences between the *standard mode* and *safe connection mode* are as follows:

Standard mode: RDS uses SLB to eliminate the impact of database engine HA switching on the application layer and shorten the response time, but that may slightly increase the probability of transient disconnections and disable the built-in SQL injection protection.

This mode supports only one connection address. When the instance has both the Intranet address and the Internet address, it is required to first release the Intranet address or the Internet address, and then switch to *Standard Mode*.

- Safe connection mode: This mode can prevent 90% of transient disconnections and provides SQL injection protection (attacks are prevented through semantic analysis), but the response time will be increased by 20% or more. This mode supports coexistence of the Intranet address and the Internet address.

Both the Intranet address and the Internet address are used in this example. When using the RDS, please configure the connection mode based on the system plan.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Database Connection** in the left-side menu.

In the *Connection information* part, click **Switch access mode**.

Click **Confirm** on the displayed confirmation interface to switch the access mode.

Set network types

ApsaraDB supports two network types: classic network and Virtual Private Cloud (VPC). This chapter describes the differences between the two network types and the method of configuration.

Background information

On Alibaba Cloud platform, a classic network and VPC have the following differences:

Classic network: The cloud service on the classic network is not isolated, and unauthorized access can be blocked only by the security group or white list policy of the cloud service.

VPC: It helps you build an isolated network environment on the Alibaba Cloud. You can customize the routing table, IP address range and gateway on the VPC. In addition, you can combine your own machine room and cloud resources in the VPC of Alibaba Cloud into a virtual machine room through a leased private line or VPN to migrate applications to the cloud seamlessly.

Operation procedure

By default, RDS uses the **classic network**. If you want to use **VPC**, ensure that RDS and VPC are in the same region. You may create VPC in either of the following scenarios:

If RDS is not created, first create VPC, and then create RDS under VPC. For details, refer to the section **New RDS Scenario** below.

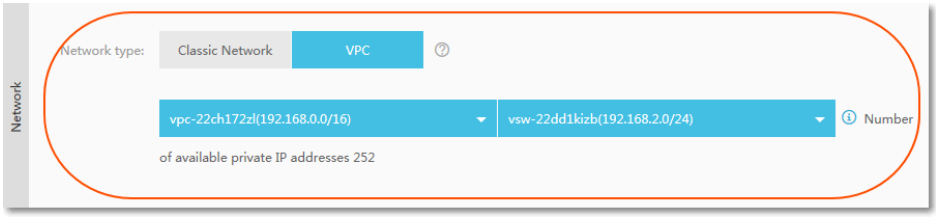
If RDS already exists, create VPC in the region where RDS is located and add RDS to VPC. For details, refer to the section **Existing RDS Scenario** below.

Scenario: For a new RDS

Create VPC. For details, refer to the VPC Quick Start.

Create an RDS instance in the region where VPC is located. For details, refer to Purchase.

During the purchase process, select **VPC** for **Network type** and select the created VPC, as shown in the figure below.



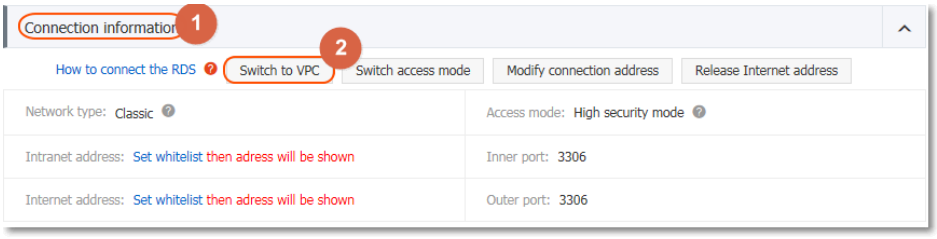
Scenario: For the existing RDS

Create VPC in the region where RDS is located. For details, refer to the **VPC Quick Start**.

Log on to the **RDS Console** and select the target instance.

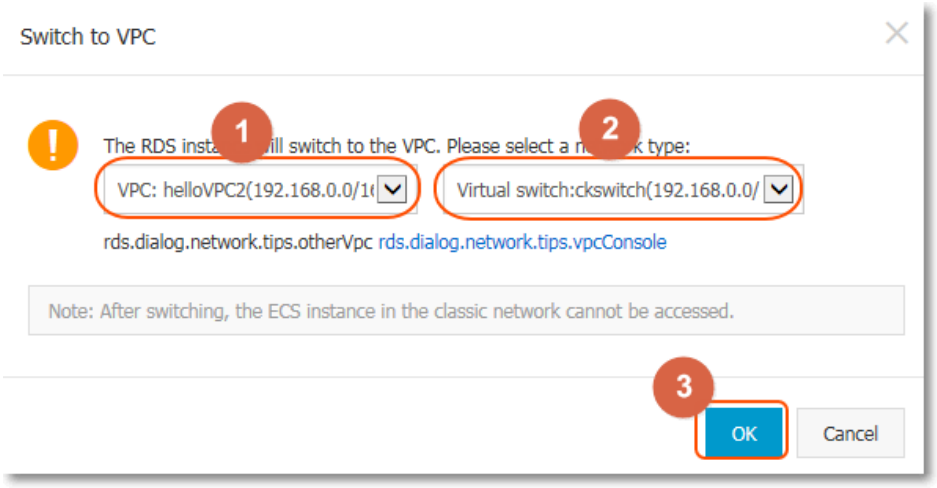
Select **Database Connection** in the instance menu.

Click **Switch to VPC** on the **Database Connection** page, as shown in the figure below.



On the **Switch to VPC** page, select *VPC* and *Virtual switch*, and click **OK**, as shown in the figure below.

NOTE: After switching to VPC, the original intranet address is changed from a classic network address to a VPC address whereas the original Internet address remains unchanged, and ECS outside of VPC cannot be accessed.



Set intranet and internet addresses

If your applications are deployed on the ECS in the same region, you do not need an Internet address. In this case, please skip this step. If your applications are deployed on the ECS in the other region or a

system other than Alibaba Cloud, you need to apply for an Internet address and use it for application interconnection.

Background information

RDS provides two kinds of connection addresses: Intranet address and Internet address.

The Intranet address or the Internet address can be used only when **Access Mode** is set to **Standard Mode**.

If your applications are deployed on the ECS in the same region, you can use the Intranet address. The system provides an Intranet address by default, and you can directly modify the connection address.

If your applications are deployed on the ECS in the other region or a system other than Alibaba Cloud, you need to use an Internet address. You can click the **Apply for an Internet Address** to release an Intranet address and generate an Internet address.

The Intranet address and the Internet address can be used at the same time only when **Access Mode** is **High Security Mode**. If your applications are deployed on the ECS in the same region and a system other than Alibaba Cloud at the same time, you must use both Intranet and Internet addresses.

Note:

Before accessing the database, you need to add the IP addresses or IP segments used to access the database to a white list. For details, refer to Set a white list.

The RDS will charge a fee for traffic over the Internet address. For detailed charges, please refer to RDS Price.

To obtain a higher transmission rate and a higher security level, you are recommended to migrate the applications to an Alibaba ECS in the same region as your RDS.

Operation procedure

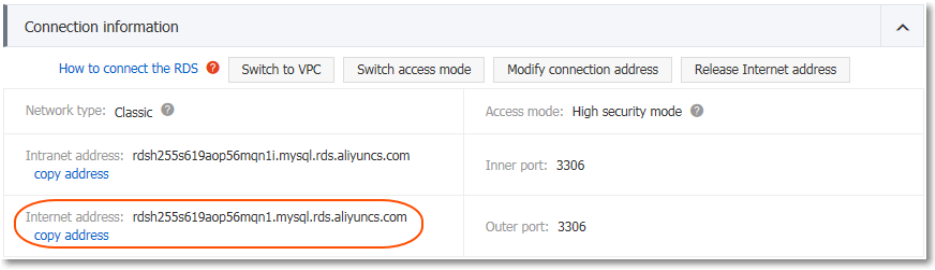
Both the intranet address and the internet address are used in this example. When using the RDS, configure the connection mode based on the system plan.

Log on to the RDS Console and select the target instance.

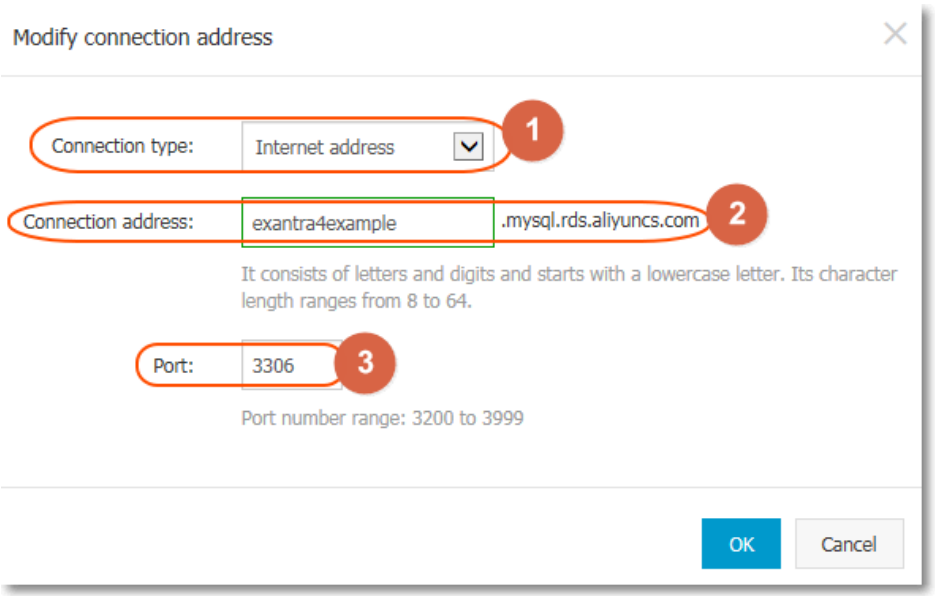
Select **Database Connection** in the menu.

Click **Apply for internet address** in *Connection information* part, and click **OK** on the displayed confirmation interface to generate an Internet address, as shown in the figure below.

Note: Be aware of that traffic at the Internet address may cause charges and reduce the instance security.



Click **Modify connection address**, set the Intranet and the Internet connection addresses and port numbers in the displayed window, and click **OK**, as shown in the figure below.



Connection type: Select **Intranet address** or **Internet address** according to the connection type to be modified.

Connection address: The address format is **xxx**.sqlserver.rds.aliyuncs.com. **xxx** is a

user-defined field consisting of 8 to 64 characters (only supporting letters and digits). It must start with a lowercase letter, for example, **extranet4example**.

Port: Indicates the number of the port through which the RDS provides external services, which can be an integer within the range of 3,200 to 3,999.

Security management

SQL audit

You can use the RDS Console or APIs to check SQL details and audit SQL periodically to detect problems in a timely manner.

Note: Currently, ApsaraDB for SQL Server does not support SQL audit.

Background information

You can use SQL Details and binlog to check incremental data of ApsaraDB for MySQL. Differences between the two audit methodologies are as follows.

SQL Audit: Similar to audit log of MySQL, SQL Audit collects statistics on all DML and DDL operation information. Some of the information is obtained through network protocol analysis. SQL Audit does not parse actual parameter values, and a small amount of records may be lost when the SQL query volume is large. Therefore, this method may result in inaccurate incremental data statistics.

binlog: It accurately records all ADD, DELETE and MODIFY operations performed on the database, and can accurately recover the user’ s incremental data. However, binlogs are first of all stored in an instance. The system periodically migrates full binlogs to OSS and stores the binlogs for seven days. Binlogs which are not full cannot be saved (it is why part of binlogs are not uploaded when you click **One-click Binlog Upload**). In this way, the incremental data of the database can be accurately recorded, but logs cannot be obtained in real time.

SQL audit records are retained for 30 days.

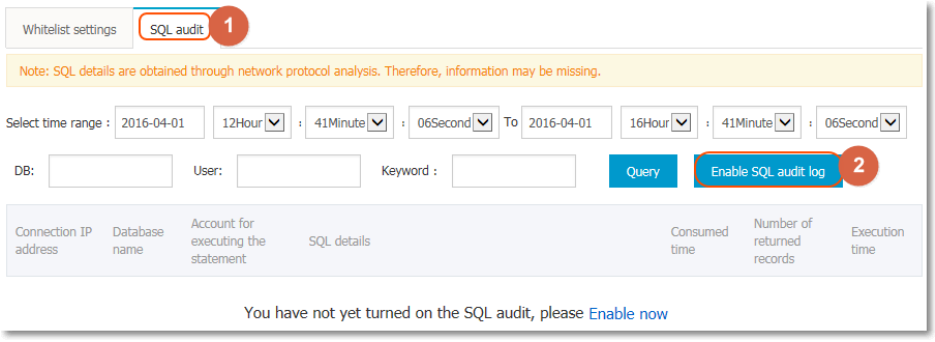
SQL Audit is disabled by default. If SQL Audit is enabled, additional charges are incurred. For detailed charges, refer to RDS Price.

Enable SQL aduit log

Log on to the RDS Console and select the target instance.

Select **Security control** in the instance menu.

On the **Security control** page, select the **SQL audit** tab page and then click **Enable SQL audit log**, as shown in the figure below.



After enabling SQL Audit, you can query SQL information based on criteria such as time, DB, user and key words.

Disable SQL audit log

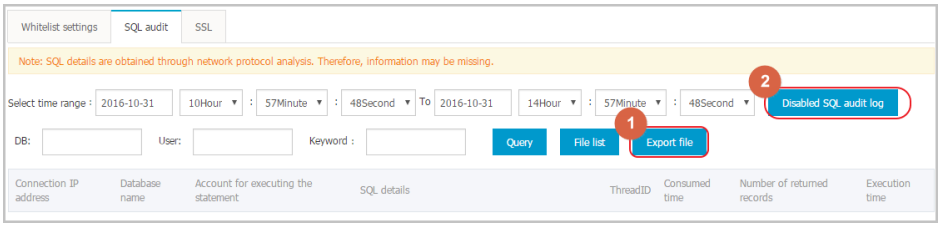
To disable SQL audit so as to avoid additional charges, you can disable SQL audit log on the **SQL audit** tab page.

Note: When the SQL audit log is disabled, all the audit contents (including the historical contents) will be cleared. Therefore, ensure that you export the audit contents and save them to a local file before disabling the SQL audit log.

Log on to the RDS Console and select the target instance.

Select **Security control** in the instance menu.

On the **Security control** page, select the **SQL audit** tab page and then click **Export file**, as shown in the figure below.



After the SQL audit contents have been exported and saved to a local file, click **Disable SQL audit log**.

Set a white list

To ensure the security and stability of a database, you need to add IP addresses or IP segments used to access the database to a white list. This section describes how to set a white list.

Before using the target instance, you need to modify the white list.

Context

You can access the database in three scenarios:

Access the RDS database through the Internet

Refer to Setting Intranet and Internet Addresses to apply for an Internet IP address.

Refer to this section to add the application service IP address to the white list.

If you cannot connect to the RDS after adding the application service IP address to the white list, refer to How to locate the local IP address using RDS for MySQL to obtain the actual IP address of the application service.

Access the RDS database through the Intranet:

Ensure that the network type is the same for RDS and ECS. For details about how to set the network type, refer to Setting Network Type.

Refer to Setting Intranet and Internet Addresses to apply for an Intranet IP address.

Refer to this section to add the ECS IP address to the white list.

Access the RDS database through the Internet and Intranet simultaneously:

Ensure that the network type is the same for RDS and ECS, and set the access mode to **High Security Mode**. For details about how to set the network type, refer to Setting Network Type.

Refer to Setting Intranet and Internet Addresses to apply for Internet and Intranet IP addresses.

Refer to this section to add the application service IP address and ECS IP address to the white list.

Operation procedure

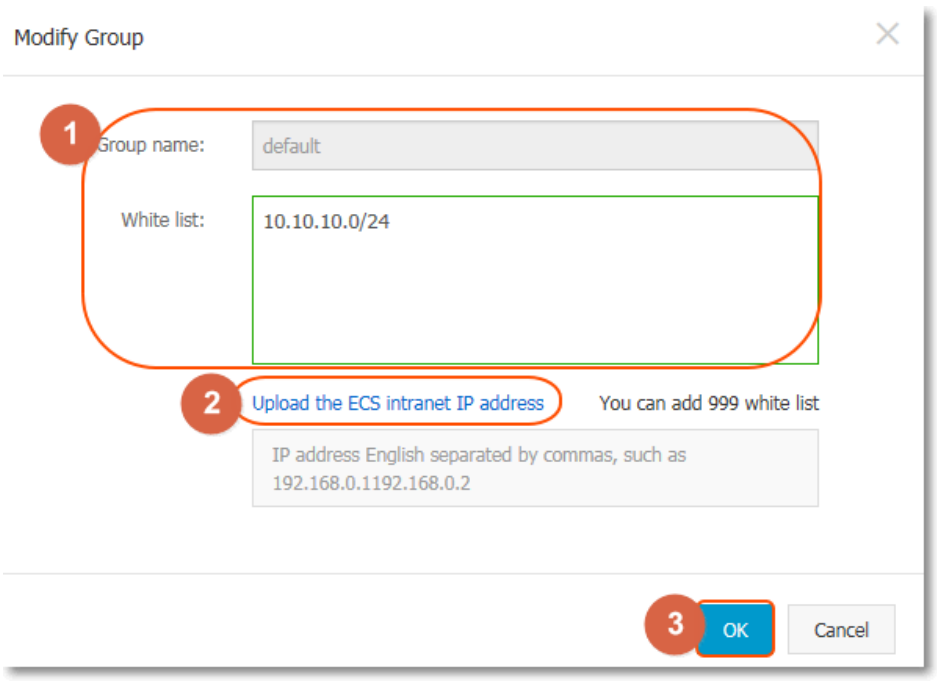
Log on to the RDS Console and select the target instance.

Select **Data Security** in the instance menu.

On the *Data Security* page, click **Modify** after the default group, as shown in the figure below.

You can also click **Clear** after the default group to delete the white list from the default group, and click **Add white list group** to create a custom group.

On the *Add white list group* page, delete the default white list *127.0.0.1*, enter a custom white list and then click **OK**, as shown in the figure below.



Parameters are described as follows:

Group name: The group name contains 2 to 32 characters which consist of lowercase letters, digits or underscores. The group name must start with a lowercase letter and end with a letter or digit. The default group cannot be modified or deleted.

White list: Enter IP addresses or IP segments which can access the database. IP addresses or IP segments are separated by commas.

1,000 white lists can be set for MySQL, PostgreSQL and PPAS, and 800 white lists can be set for SQL Server.

The white list can contain IP addresses (for example, 10.10.10.1) or IP segments (for example, 10.10.10.0/24, which indicates any IP address in the format of 10.10.10.X can access the database).

% or *0.0.0.0/0* indicates any IP address is allowed to access the database. **This configuration greatly reduces security of the database, and thus is not recommended unless necessary.**

After an instance is created, the local loopback IP address *127.0.0.1* is set as the default white list, and thus external IP addresses are prohibited to access this instance.

Upload the ECS intranet IP address: Click the IP address, and ECS of the same account is displayed. You can add the ECS to the white list.

Subsequent operations

Proper use of the white list provides improved access security protection for RDS, thus it is recommended to periodically maintain the white list.

During future operations, you can click **Modify** after the group name to modify an existing group, or click **Delete** to delete an existing group.

Set SSL encryption

To increase link security, you can enable SSL encryption and install SSL certificates on the necessary application services.

Background information

SSL (Secure Sockets Layer) is used on the transport layer to encrypt network connections. It increases the security and integrity of communication data, but also increases the network connection time.

Considerations

Due to the inherent drawbacks of SSL encryption, activating this function will significantly increase your CPU usage. We suggest only enable SSL encryption for Internet connections that require encryption. Intranet connection are relatively secure, and generally do not require link encryption.

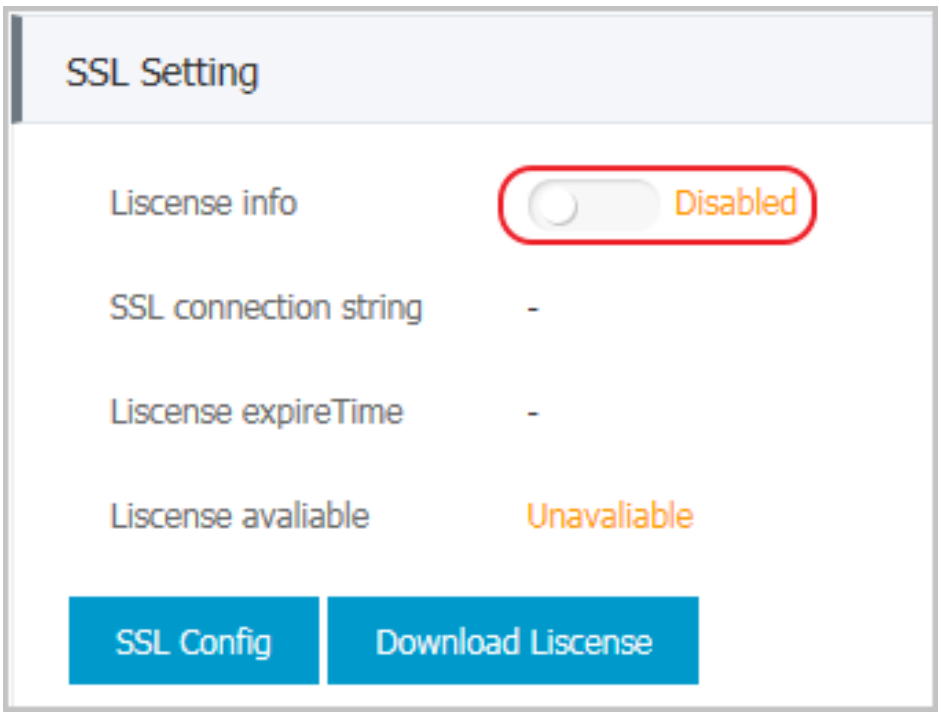
Operation procedure

Log on to the RDS Console and select the target instance.

In the left-side menu bar, select **Security control** to go to the **Security control** page.

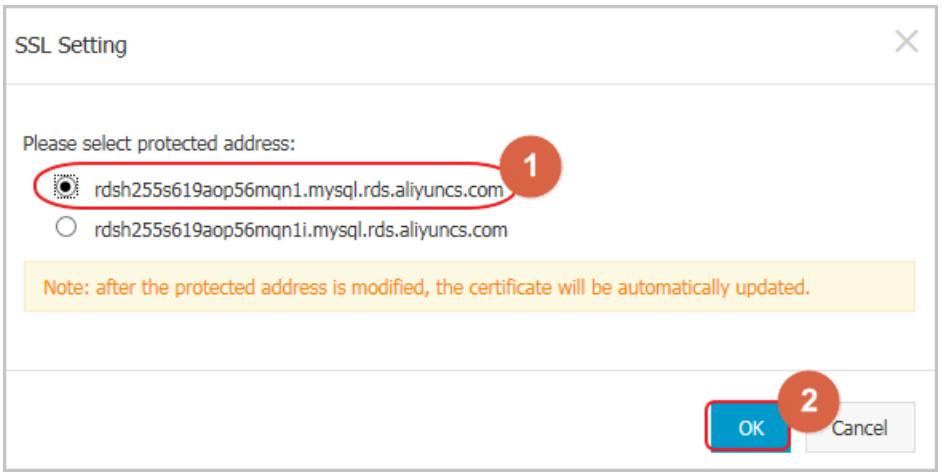
Select the **SSL** tab page.

Click the button next to **Disabled**, as shown below.



In the **SSL Setting** dialog box, select the link for which to activate SSL encryption and click **OK** to activate SSL encryption, as shown below.

Note: Users can choose to encrypt both Internet and intranet links as needed, but only one link can be encrypted.



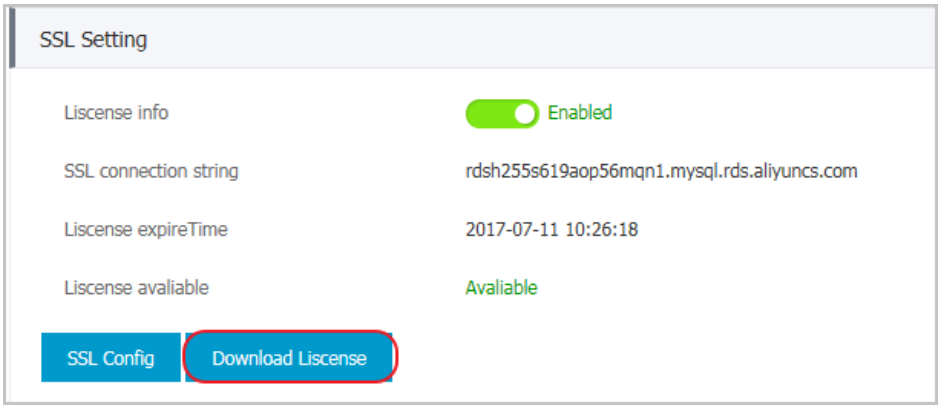
Click **Download License** to download the SSL certificate, as shown below.

Note: The downloaded SSL certificate is a package including the following two files:

p7b file: Used to import the CA certificate on Windows OS

PEM file: Used to import the CA certificate on other systems or for other

applications



Subsequent operations

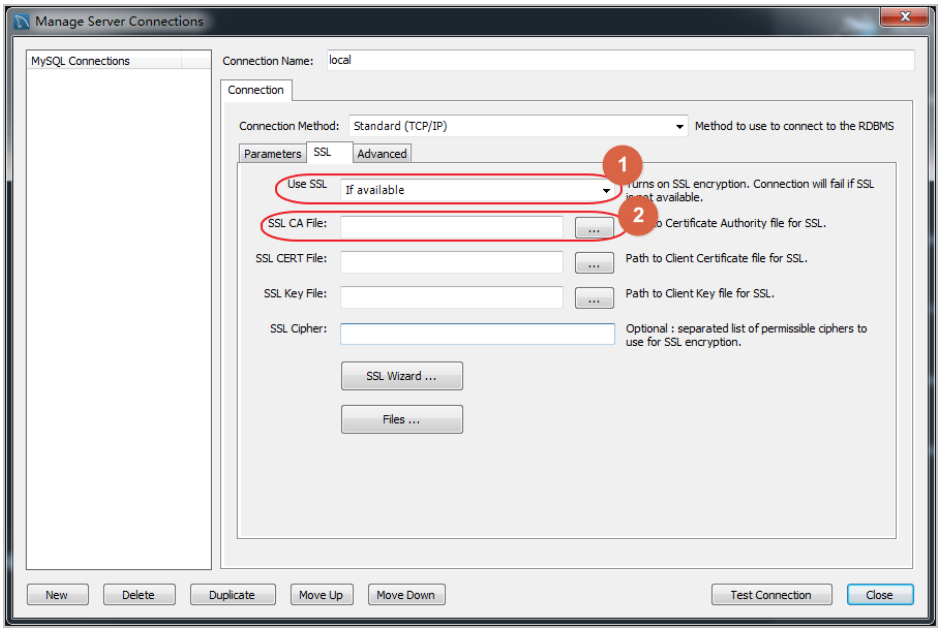
After activating SSL encryption, you need to configure the SSL certificate when you connect RDS to an application or a client.

This section uses MySQL Workbench as an example to describe how to install the SSL certificate. For details about how to install the SSL certificate on other applications or a client, refer to the corresponding product instructions.

Start MySQL Workbench.

Select **Database** > **Manage Connections**.

Enable **Use SSL** and import the SSL certificate, as shown in the figure below.



Set Transparent Data Encryption

Transparent Data Encryption (TDE) can be used to perform real-time I/O encryption and decryption on instance data files. To increase data security, you can enable TDE to encrypt instance data.

Note: Currently TDE is only applicable to the database of SQL Server 2008 R2.

Background information

TDE provides real-time I/O encryption and decryption on data files. The data are encrypted before being written to the disk and decrypted when read from the disk into the memory. TDE will not increase the size of data files. Developers will not have to modify any applications before using the TDE function.

Considerations

Once TDE is activated, it cannot be deactivated.

Encryption uses keys produced and managed by the Key Management Service (KMS). RDS does not provide the keys and certificates needed for encryption. After activating TDE, if the user wants to restore the data to the local device, he must use RDS to decrypt the data first.

After activating TDE, CPU usage will significantly increase.

Prerequisites

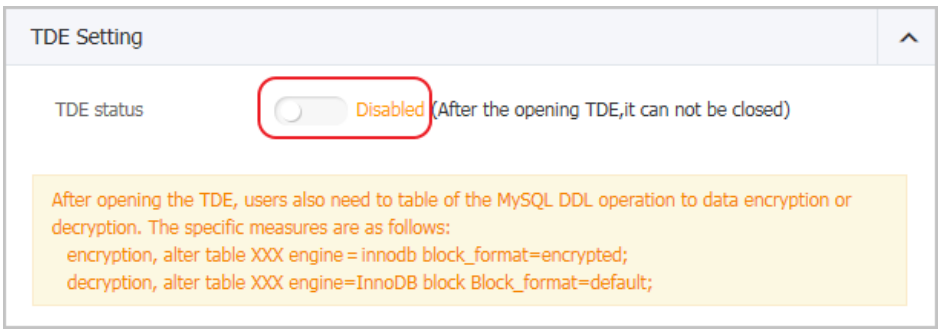
Key Management Service (KMS) is activated.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Data Security** in the left-side menu. Then, on the **Data Security** page, select the **TDE** tab.

Click **Not Activated**, as shown below.



Click **OK** to activate TDE.

Note: If you have not activated the Key Management Service, you will be prompted to do so when activating TDE. After activating the Key Management Service, click **Not Activated** to activate TDE.

Log on to the database and execute the following command to encrypt the relevant tables.

```
alter table <tablename> engine = innodb block_format=encrypted;
```

Subsequent operations

If you want to decrypt a table encrypted with TDE, execute the following command.

```
alter table <tablename> engine = innodb block_format=default;
```

Monitoring alarm

Set monitoring frequency

The RDS Console provides abundant performance metric items for users to conveniently view and know the running status of instances. You can use the RDS Console to set the monitoring frequency, view monitoring data of a specific instance, create monitoring views, and compare instances of the same type under the same account.

Background information

Improving the RDS performance monitoring frequency will incur additional charges. For detailed charges, refer to RDS Price.

Operation procedure

Log on to the RDS Console and select the target instance.

Select **Monitor and alarm** in the menu.

Different types of databases contain different metric items. For details, refer to the *Monitoring List* below.

Click **Monitoring frequency setting** on the **Monitor** tab page, as shown in the figure below.

On the specific metric item page, you can perform the following operations:

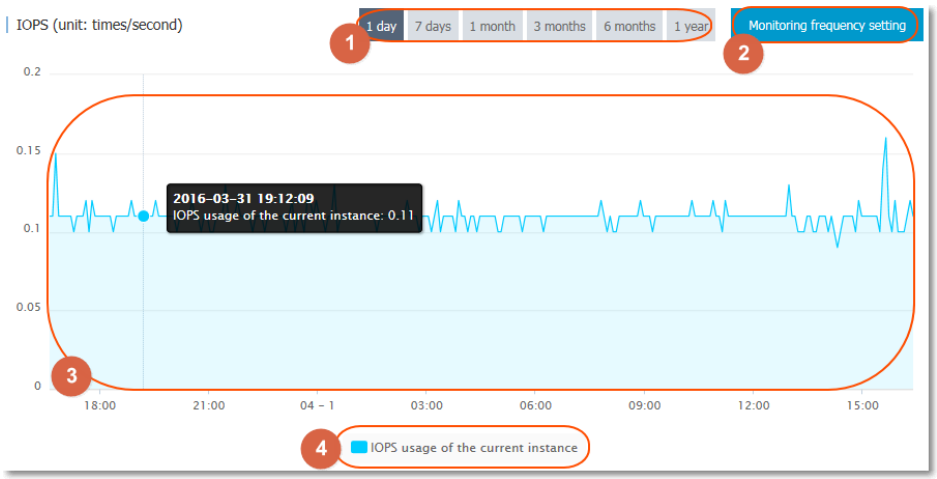
Select the monitoring type to be displayed (as shown in Figure 2)

Select the monitoring period to be displayed (as shown in Figure 3)

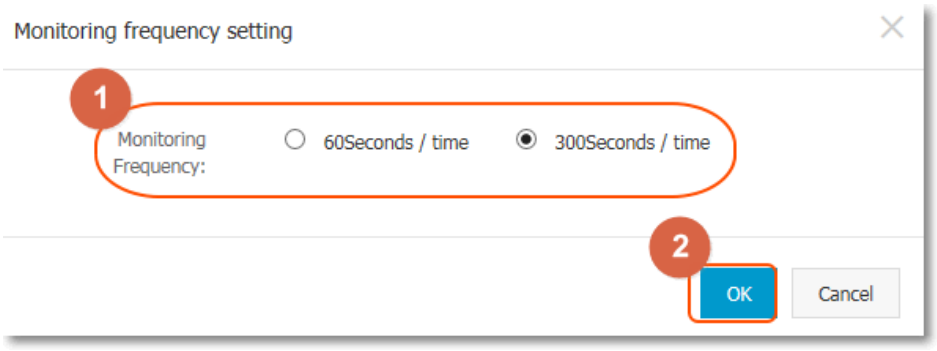
Display the monitoring data. Also you can frame select the monitoring period to be displayed (as shown in Figure 4), and click **Reset Zoon** to recover the monitoring period (as shown in Figure 6)

Select the monitoring data to be displayed (as shown in Figure 5)

Refresh the monitoring data to be displayed (as shown in Figure 8)



On the setting page displayed, click *Monitoring Frequency* and then **OK**, as shown in the figure below.



List of metric items

RDS for MySQL

Metric Item	Description	Monitoring Frequency	Monitoring Period
Disk Space	Disk space usage of the instance, including the overall usage of the disk space, data space, log space, temporary file space and system file space Unit: MByte	60s/time 300s/time	30 days
IOPS	I/O request times of the instance per second Unit: time/s	60s/time 300s/time	30 days
Connections	Total number of current connections, including the number of active connections and total connections	60s/time 300s/time	30 days
CPU memory usage	Usage of CPU and memory of the instance (not including memory used by the operating system)	60s/time 300s/time	30 days
Network Traffic	Incoming/outgoing traffic of the	60s/time 300s/time	30 days

	instance per second Unit: KByte		
QPS/TPS	The number of SQL statements executed and transactions processed per second	60s/time 300s/time	30 days
InnoDB buffer pool	InnoDB buffer pool read hit rate, utilization rate, and percentage of dirty data blocks	60s/time 300s/time	30 days
InnoDB read/write volume	Average InnoDB data reads and writes per second Unit: KByte	60s/time 300s/time	30 days
InnoDB reads/writes	The number of InnoDB reads and writes per second	60s/time 300s/time	30 days
InnoDB log	The number of InnoDB physical writes to the log file, log write requests, and FSYNC writes to the log file	60s/time 300s/time	30 days
Temporary tables	The number of temporary tables created automatically on the hard disk when the database executes the SQL statement	60s/time 300s/time	30 days
MyISAM Key Buffer	Average per-second Key Buffer read hit rate, write hit rate and usage of MyISAM	60s/time 300s/time	30 days
MyISAM read and write times	Times of MyISAM read and write from/to the buffer pool and from/to the hard disk per second	60s/time 300s/time	30 days
COMDML	The number of statements executed for the database per second. The statements include Insert, Delete, Insert_Select, Replace,	60s/time 300s/time	30 days

	Replace_Select, Select and Update.		
ROWDML	The number of operations performed on InnoDB, including the number of physical writes to the log file per second, the number of rows read, updated, deleted and inserted to InnoDB table per second.	60s/time 300s/time	30 days

RDS for SQL Server

Metric Item	Description	Monitoring Frequency	Monitoring Period
Disk Space	Disk space usage of the instance, including the overall usage of the disk space, data space, log space, temporary file space and system file space Unit: MByte	60s/time 300s/time	30 days
IOPS	I/O request times of the instance per second Unit: time/s	60s/time 300s/time	30 days
Connections	Total number of current connections, including the number of active connections and total connections	60s/time 300s/time	30 days
CPU usage	CPU usage (including CPU used by the operating system) of the instance	60s/time 300s/time	30 days
Network Traffic	Incoming/outgoing traffic of the instance per second Unit: KByte	60s/time 300s/time	30 days
TPS	The number of transactions	60s/time 300s/time	30 days

	processed per second		
QPS	The number of SQL statements executed per second	60s/time 300s/time	30 days
Cache hit rate	Read hit rate of the buffer pool	60s/time 300s/time	30 days
Average full table scans per second	Average number of full table scans per second	60s/time 300s/time	30 days
SQL compilations per second	The number of compiled SQL statements per second	60s/time 300s/time	30 days
Page writes of the checking point per second	The number of page writes of the checking point in the instance per second	60s/time 300s/time	30 days
Logins per second	The number of logins per second	60s/time 300s/time	30 days
Lock timeouts per second	The number of lock timeouts per second	60s/time 300s/time	30 days
Deadlocks per second	The number of deadlocks in the instance per second	60s/time 300s/time	30 days
Lock waits per second	The number of lock waits per second	60s/time 300s/time	30 days

RDS for PostgreSQL

Metric Item	Description	Monitoring Frequency	Monitoring Period
Disk space	Usage of the instance disk space Unit: MByte	60s/time 300s/time	30 days
IOPS	The number of I/O requests of the data disk and log disk in the instance per second Unit: time/s	60s/time 300s/time	30 days

RDS for PPAS

Metric Item	Description	Monitoring Frequency	Monitoring Period

Disk space	Usage of the instance disk space Unit: MByte	60s/time 300s/time	30 days
IOPS	The number of I/O requests of the data disk and log disk in the instance per second Unit: time/s	60s/time 300s/time	30 days

Set monitoring rules

The RDS instance provides the instance monitoring function, and sends messages to users after detecting an exception in the instance. Besides, when the instance is locked due to the insufficient disk space, the system also sends a message to notify the users.

Background information

Monitoring and alarm are implemented through Alibaba Cloud Monitor. Alibaba Cloud Monitor can be used to set monitored items, and notify all contacts in the alarm contact group when the alarm rules of the monitored items are triggered. You can maintain an alarm contact group corresponding to an alarm monitored item, so as to promptly notify the relevant contacts when an alarm occurs.

Operation procedures

Log on to the RDS Console.

Select the region where the target instance is located.

Click the name of the target instance to go to the “Basic information” page.

Select **Monitor And Alarm** in the left menu.

Select the **Alarm** tab.

Click **Alarm Rules** to open the Cloud Monitoring console.

Note: You can click **Refresh** to manually refresh the current status of the alarm monitored item.

Select **Alarm service** > **Alarm Contact** in the left menu.

Note: When alarm rules are set for the first time, if the alarm notification object is not a contact of the Alibaba Cloud account of RDS, the alarm contact and alarm contact group must be created first. If you have already set the alarm contact and the alarm contact group, go to Step 10.

Click **New Contact**.

Enter the alarm contact information, click **Send verification code**, enter the verification code sent to your mailbox in the **Verification code** field, and then click **Save**.

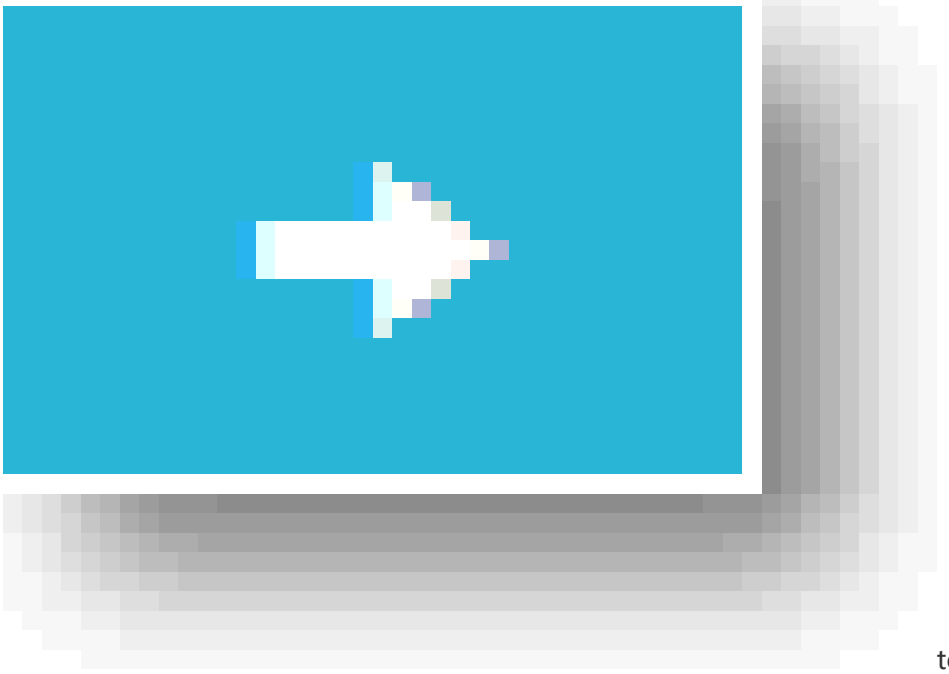
Note:

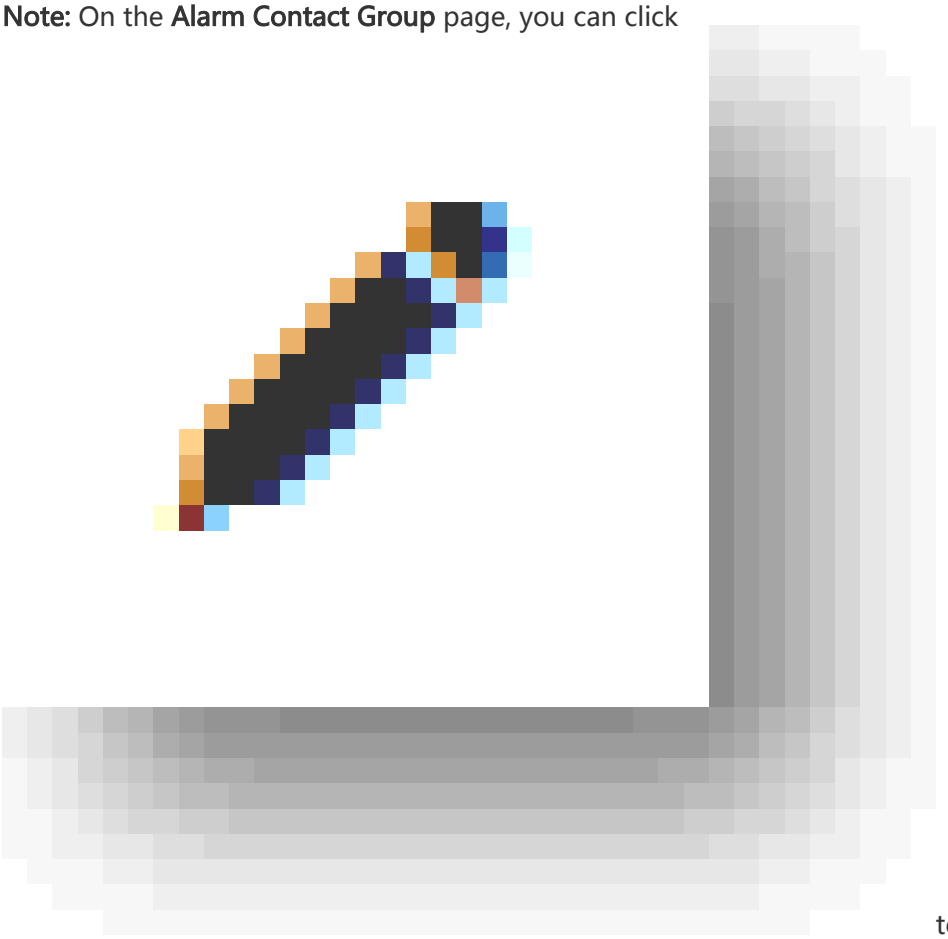
It is recommended to perform the next step to create the alarm contact group after you add all alarm notification objects.

You can click **Edit** on the "Alarm Contact" page to modify the corresponding contact information, or click **Delete** to delete a corresponding contact.

On the "Alarm contact management" page, Select the **Alarm contact group** tab.

Click **New contact group**.

Enter the **Group name** and **Remarks**, select a contact from the **Existing contact**, click  to add the contact to the **Selected contact**, and then click **OK**.

Note: On the **Alarm Contact Group** page, you can click  to modify the corresponding contact group, click **X** to delete the corresponding contact group, or click **Delete** next to a member in the contact group to quickly delete the member.

After creating the alarm contact group, select **Cloud Service Monitoring** > **RDS** in the left menu.

Select the region of RDS for which the alarm rule is to be set.

Find the target instance and click **Alarm rules** in the Actions column.

The system displays the monitored items of the current alarm. The IOPS usage, Connections usage, CPU usage and Disk usage monitored items are enabled by default.

Click **New alarm rule** to create or add new alarm rules.

Note: You can also click **Modify** after a monitored item to modify this item, click **Suspend** to stop the monitored item, or click **Delete** to delete the monitored item.

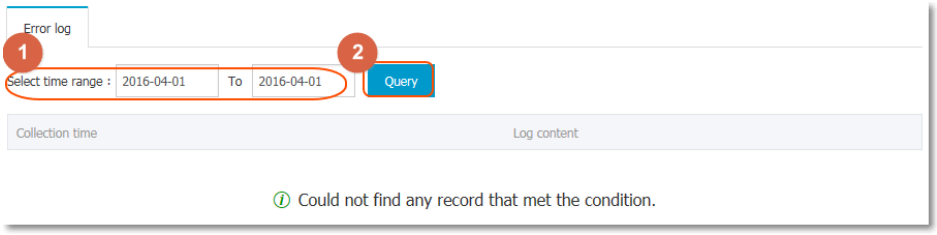
Log management

The RDS Console can be used to query error logs of an instance to locate and analyze failures.

Operation procedure

1. Log in to the RDS Console and select the target instance.
2. Select **Log Management** in the instance menu.

   On the *Error Log* page, select the time range of the error logs to be queried, and click **Query**, as shown in the figure below.



Backup and recovery

Back up RDS data

You can configure a backup policy to adjust the cycles of RDS data backup and log backup and thereby realize automatic backup. You can also manually back up RDS data.

Instance backup files occupy backup space. Charges will be incurred if the used space exceeds the free quota. Please design a backup cycle properly to meet service requirements based on backup space capacity. For detailed charges, refer to RDS Price.

Background information

ApsaraDB supports data backup and log backup. Enable log backup if you want to recover data by time. The following table lists the backup policies applicable to different database types:

Database Type	Data Backup	Log Backup
MySQL	- Automatic backup	After being generated,

	supports full physical backup - Manual backup supports full physical backup, full logical backup, and single-database logical backup - >MySQL 5.7 backup files retained for a maximum period of 7 days, and does not support a logical backup	binglogs (500 MB per log) are compressed and uploaded immediately. Local files are deleted within 24 hours. Binlog files occupy instance disk capacity. Using **One-key Binlog Upload**, users can upload Binlog files to the OSS. This does not affect the data recovery function and will stop the Binlog files from occupying instance disk space.
SQL Server	- Supports full physical backup and incremental physical backup - Automatic backup uses the cycle **Full Backup-Incremental Backup-Incremental Backup**. For example, if a full backup is performed on Monday, incremental backups will be performed on Tuesday and Wednesday, another full backup will be performed on Thursday, then incremental backups on Friday and Saturday, and so on. If a full backup is manually performed at any time in the backup cycle, the next two backups will be incremental	Included in data backup; individual transaction logs are not provided for download

	backups. - The SQL Server always compresses transaction logs during the backup process - On the *Backup and Recovery* page of the target instance's management console, you can click **Compress Transaction Log** to manually compress the transaction log.	
PostgreSQL	Supports full physical backup	After being generated, write-ahead logs (WALs) (16 MB per log) are compressed and uploaded immediately. Local files are deleted within 24 hours
PPAS	Supports full physical backup	After being generated, WALs (16 MB per log) are compressed and uploaded immediately. Local files are deleted within 24 hours

Automatic backup (Backup policy setting)

After you configure a backup policy, ApsaraDB will automatically back up databases based on the policy.

Log on to the RDS Console and select the target instance.

Select **Backup and Recovery** in the left-side menu.

On the "Backup and recovery" page, select **Backup cycle** and click **Edit**.

On the "Backup cycle" page, set backup specifications and click **OK**. The parameters are explained as follows:

Retention Days: Specifies the number of days when backup files are retained. The default value is 7 days. The value range is 7-730 days.

Backup cycle: You can set it to one or multiple days in a week. SQL Server, PostgreSQL, and PPAS instances are backed up daily by default and this cannot be modified.

Backup time: This value can be set to any time; units: hours.

Log Backup: Specifies whether to enable log backup. This is enabled by default for SQL Server instances and cannot be modified.

Log Retention Days: Specifies the number of days when the log backup files are retained. The default value is 7 days. The value range is 7-730 days and it must be less than or equal to the value of the retention days.

Manual backup

Log on to the RDS Console.

Select the region where the target instance is located.

Click the ID of the target instance to go to the “Basic information” page.

Click **Backup instance** in the upper right corner.

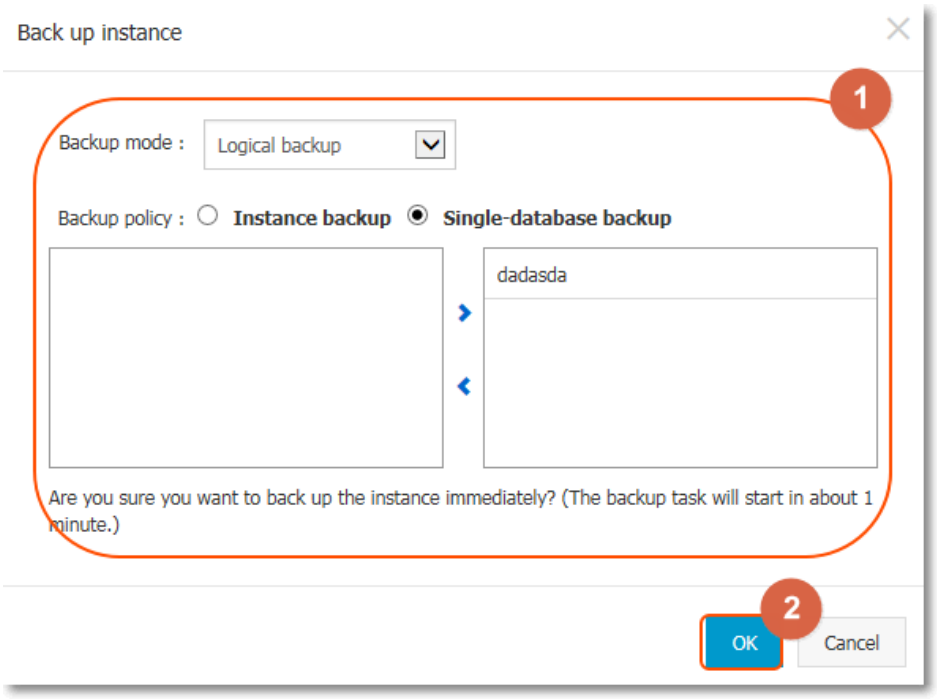
On the *Backup instance* page, select a backup mode and policy, and click **OK**, as shown in the figure below.

NOTE: MySQL Single-database logical backup is used as an example.

Backup mode: For MySQL instances, the values are *Physical backup* or *Logical backup*. For SQL Server, PostgreSQL, and PPAS instances, the values are *Automatic backup* and *Full backup*. When *Automatic backup* is selected for an SQL Server instance, incremental backup will be performed if either the current or the last two backups are full. If not, a full backup will be performed.

Backup Policy: If you select *Logical backup* or *full Backup*, you can select *Instance backup* or *Single-Database backup*. Single-database backup is not supported for

PostgreSQL and PPAS instances. If you select *Single-Database backup*, select the database you want to back up on the left and click > to add the selected database to the list on the right.



Recover RDS for MySQL data

The data recovery function can minimize the damage caused by database misoperations.

Recover data to the master instance through a temporary instance (recommended)

Creating a temporary instance does not affect the current production instance but will provide a temporary instance for data access. It is recommended that you recover data to a temporary instance and verify the data before migrating the data to the master instance. This avoids the impact of data recovery on services.

Note:

A temporary instance will inherit the account and password of the backup file but keep the network type of the current instance.

A temporary instance uses its instance name as the password.

Only one temporary instance can be generated at the same time. Before you create a temporary instance, delete any existing temporary instance.

A temporary instance is valid for 48 hours.

Operation procedures

Log on to the RDS Console and select the region where the target instance is located.

Click the ID of the target instance to go to the “Basic information” page.

Select **Backup And Recovery** in the left menu to go to the “Backup and recovery” page, and then select the **Temporary instance** tab.

Select the recoery time point，and click **Create temporary instance**.

Click **OK** in the pop-up dialog box to create a temporary instance.

After the temporary instance is created, go to the “Instance list” page.

Click the ID of the target instance to go to the “Basic information” page.

Click **Create a data migration task** in the upper right corner to go to Data Transmission console.

Select **Data migration** in the left-side menu to go to the “Migration task list” page.

Click **Create migration task**.

Fill in the task name, source database and target database information, and click **Authorize whitelist and enter into next step**.

Parameters description:

Task name: custom task name or default value.

Source database information:

Instance type: Specifies the instance type of a database. Select **RDS Instance**.

Instance region: The region where the master instance is located.

RDS instance ID: Click the drop-down list and select the temporary instance ID.

Database account: It is consistent with the account name of the master instance. Ensure that this account has the read/write privilege to all the data to be migrated.

Database password: It is consistent with the password of the master instance account.

Target database information:

Instance type: **RDS Instance** by default.

Instance region: The region where the master instance is located.

RDS instance ID: ID of the target RDS instance. Click the drop-down list and select the master instance mapped to the temporary instance

Database account: The master instance account name. Ensure that this account has the read/write privilege to all the data to be migrated.

Database password: The password of the master instance account.

Select the migration type, choose the migration object from the *Migration objects* column, and click **>** to add the migration object to the *Selected* column. Then click **Pre-check and start**, as shown in the figure below.

Note:

During data migration, the data (structure) of the source database is copied to the target database without affecting the data (structure) of the source database.

DDL operations are not supported during data migration. DDL operations may lead to a migration failure.

MySQL DTS incremental migration can only be performed for MySQL. The duration of a DTS incremental migration task is 15 days at most. If the task is not terminated after 15 days, the system resources may be recovered.

To modify the migration object name in the target database, you can click **Edit** on the right side of the *Selected* list to modify the name.

Note: Pre-check failure is described below. If pre-check is passed, go to step 10.

If the system displays the pre-check failure result, click **!** next to the check item with *Check Result* as *Failed* to check the detailed failure information, and perform troubleshooting accordingly.

After troubleshooting, select the current migration task on the **Migration task list** page and click **Start**.

After pre-check is passed, click **OK** to automatically execute the migration task.

Recover data directly to the master instance

During the direct data recovery, the specified backup data will overwrite the data of the master instance, and the data generated after creation of the specified backup data will be lost. It is recommended that you create a temporary instance for data recovery and migration to ensure higher security.

Note:

This method is only applicable to the database of SQL Server 2008 R2 and MySQL.

If a read-only instance exists, the specified backup data cannot directly overwrite the original data of the master instance.

For details, refer to the section **Recovering data to the master instance through a temporary instance (recommended)** above.

Operation procedures

Log on to the RDS Console.

Select the region where the target instance is located.

Click the name of the target instance to go to the “Basic information” page.

Select **Backup And Recovery** in the left menu.

Select the **Backup list** tab.

Select the time range for recovery and click **Query**.

Select the target backup file and click **Coverage restoration**.

Click **Confirm** in the popup dialog box to recover data to the master instance.

Download RDS data and log backup

To protect users’ rights, RDS allows users to download data backup files and log backup files that are not encrypted.

Background information

RDS is based on the master-slave instance architecture. Each instance has a unique ID. RDS performs data backup on the backup instance and log backup on the master instance and slave instance.

If you want to download data to the local device and recover the data to the local database, you need to download the data file and log file under the same instance ID.

RDS supports different backup policies for different types of databases. Accordingly, downloadable data backups and log backups are also different. For details, refer to Back Up RDS Data.

Operation procedure

Log on to the RDS Console and select the target instance.

Click **Backup and Recovery** in the menu.

Select the **Backup list** tap page on the *Backup and recovery* page.

Select the latest data backup file prior to the data recovery time, and click **Download**.

Click the desired download method on the *Download the instance backup file* page. The download methods are described as follows:

NOTE: Traffic fees will be incurred when you download backup files over the Internet. For detailed charges, see RDS Price.

Download: The backup file will be downloaded using an Internet address.

Copy intranet address: When ECS and RDS are in the same region, you can use an intranet address on ECS to download the backup file at a faster speed and with higher security

Copy internet address: An Internet address is copied and used to download the backup file via other tools

Select the **Binlog list** tab page, select the log backup file generated after the data backup time but before the recovery time, and click **Download**. Note that the selected log backup file must be under the same instance ID as that for the data backup file.

Download the log backup file. For details, refer to Step 5.

Tag management

Create tags

If you have a large number of instances, you can bind tags to facilitate classification and management. Each tag composes a key-value pair, enabling you implement two-level classification for the instances you created.

Constraints

Up to 10 tags can be bound to a single instance and they must have unique TagKeys. Tags with the same TagKeys will be overwritten.

You cannot bind/unbind more than 5 tags at once.

Tag information is independent in different regions.

After unbinding a tag, if it is not bound to any other instances, it will be deleted.

Operation procedure

Log on to the RDS Console to go to the *Instance list* page.

Select the region where the target instance is located.

Add tags to the instances. There are two methods:

Add tags to a single instance

Click **More** > **Edit tags** in the Action column of the target instance.

If you want use the existing tags, click **Available Tags**. Select the tag key and tag value, and click **Confirm**.

If you want to add a new tag, click **Create**. Fill in the tag *Key* and *Value*, and click **Confirm**.

When the tags you need have been added to the tag table, click **Confirm**.

Add tags to multiple instances

Click the checkbox next to the target instance; if you want select all the instances, click the checkbox before *Edit tags*.

Click **Edit tags**.

If you want use the existing tags, click **Available Tags**. Select the tag key and tag value, and click **Confirm**.

If you want to add a new tag, click **Create**. Fill in the tag *Key* and *Value*, and click **Confirm**.

When the tags you need have been added to the tag table, click **Confirm**.

Delete tags

If you have changed your instance or no longer need to use tags, you can delete the tags from the instance.

Constraints

You cannot bind/unbind more than 5 tags at once.

After unbinding a tag, if it is not bound to any other instances, it will be deleted.

Operation procedure

Log on to the RDS Console to go to the *Instance list* page.

Select the region where the target instance is located.

Click **More** > **Edit tags** in the Action column of the target instance.

Click **X** after the tag to delete it, as shown below.

Click **Confirm** to complete the operation.

Filter instances by tags

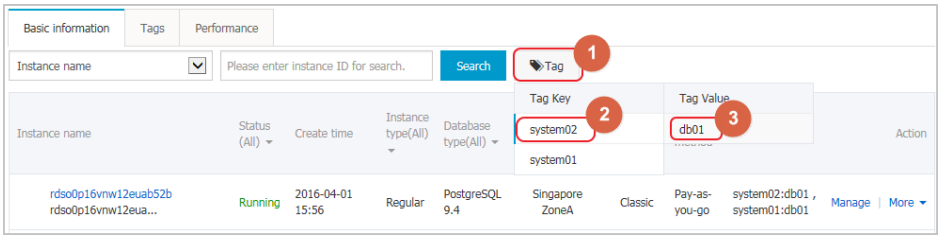
Operation procedure

Log on to the RDS Console to go to the *Instance list*.

Click **Tag** next to the *Search* button.

Select the *Tag Key* and *Tag Value* to filter instances, as shown below.

Note: After filtering instances by tags, if you need to remove a filter, you can remove the filtered tag next to the *Tag* button.



Data migration

Migrate local data to RDS

You can import the data of the local database to the ApsaraDB to realize smooth service migration. The data import method varies with different types of RDS. Select the proper data migration cases based on the actual scenarios.

Migrate data from self-built databases on ECS to RDS

Data migration from self-built databases on ECS to RDS/MongoDB/PetaData/OceanBase

Migrate data from local database to RDS for MySQL

Migrate data from local MySQL to RDS for MySQL

Migrate data from local Oracle to RDS for MySQL

Migrate data from local database to RDS for SQL Server

Migrate data from local SQL Server to RDS for SQL Server

Migrate data from SQL server without disabling services

Migrate data from local database to RDS for PostgreSQL

Migrate data from local PostgreSQL to RDS for PostgreSQL

Migrate data from local database to RDS for PPAS

Migrate data from Oracle to PPAS without disabling services

- For details on how to import data to RDS for MySQL, refer to Quick Start (MySQL).
- For details on how to import data to RDS for SQL Server, refer to Quick Start (SQL Server).
- For details on how to import data to RDS for PostgreSQL, refer to Quick Start (PostgreSQL).
- For details on how to import data to RDS for PPAS, refer to Quick Start (PPAS).

Migrating RDS data to the local database

Migrate RDS for PPAS to local Oracle

Constraints

At present, only files and normal types of data can be exported. BLOB and other binary types are not

supported.

Prerequisites

An Oracle database must be installed on the server.

The IP address of the Oracle server must be added to the white list of the RDS for PPAS database instance. For specific instructions, see Setting a White List.

You must create a table structure in Oracle that corresponds to the RDS for PPAS database table structure.

The PostgreSQL client must be fetched and uploaded to the Oracle database server.

Operation procedures

Note: This document uses the migration of data from RDS for PPAS to an Oracle database installed on an ECS instance as an example. In this example, the ECS instance OS is CentOS 6.5.

Install the PostgreSQL client on the Oracle database server.

```
[root@oraclexe ~]# yum install postgresql.x86_64
[root@oraclexe ~]# /usr/bin/psql --version
psql (PostgreSQL) 8.4.20
```

On the ECS instance, configure password-free login for RDS for PPAS.

```
[root@oraclexe ~]# vim ~/.pgpass
[root@oraclexe ~]# cat ~/.pgpass
rm-2ze466l5u1k657yyn.ppas.rds.aliyuncs.com:3433:ora:myadmin:xxxxxxx
//Parameter format: HOSTNAME:PORT:DATABASE:USERNAME:PASSWORD
[root@oraclexe ~]# chmod 0600 ~/.pgpass
```

Note: The configuration file .pgpass is located in the HOME directory.

Test the connection between ECS and RDS for PPAS.

```
[root@oraclexe ~]# psql -h rm-2ze466l5u1k657yyn.ppas.rds.aliyuncs.com -p 3433 -U myadmin ora
psql.bin (9.3.1.3, server 9.3.13.37)
Input "help" to obtain help information.
```

```
ora=>
```

If you can log on to RDS for PPAS as ora, it means that the connection has been established. After a successful test, return to the root user.

```
ora=> \q
[root@oraclexe ~]#
```

Create a data export script in the ECS instance.

Create a file **ppas_exp_all_tables_to_csv.sh**.

```
vi ppas_exp_all_tables_to_csv.sh
```

Insert the following text into the **ppas_exp_all_tables_to_csv.sh** script.

```
# ppas_exp_all_tables_to_csv.sh <hostname> <port> <username> <database>
# Author: Xiao Shaocong (Scott Siu)
# E-Mail: shaocong.xsc@alibaba-inc.com

TMP_PATH="/tmp/ppas_tables_$1_$2_$3_$4"
mkdir $TMP_PATH
if [ $? -ne 0 ]
then
exit 1;
fi

echo "select '$1 $2 $3 $4 ' || tablename || ' $TMP_PATH ' || tablename from pg_tables where
tableowner='$3' and (schemaname='$3' or schemaname='public');" > /tmp/ppas_tables_$1_$2_$3_$4.sql

psql -h $1 -p $2 -U $3 $4 -f /tmp/ppas_tables_$1_$2_$3_$4.sql | head -n -2 | tail -n +3 | awk -F " " '{printf
("psql -h %s -p %s -U %s %s -c \"\\copy %s TO '\''%s/%s'\'' CSV HEADER\"\n",$1,$2,$3,$4,$5,$6,$7)}' | sh
```

Grant the execution permission to the **ppas_exp_all_tables_to_csv.sh** script.

```
[root@oraclexe ~]# chmod 0755 ppas_exp_all_tables_to_csv.sh
```

Execute the data export script in the ECS instance.

```
[root@oraclexe ~]# ./ppas_exp_all_tables_to_csv.sh rm-2ze466l5u1k657yyn.ppas.rds.aliyuncs.com 3433
myadmin ora
```

Verify the data in the exported CSV file.

```
[root@oraclexe ~]# cat /tmp/ppas_tables_rm-
2ze466l5u1k657yyn.ppas.rds.aliyuncs.com_3433_myadmin_ora/*

deptno,dname,loc
10,ACCOUNTING,NEW YORK
20,RESEARCH,DALLAS
30,SALES,CHICAGO
40,OPERATIONS,BOSTON

empno,ename,job,mgr,hiredate,sal,comm,deptno
7369,SMITH,CLERK,7902,17-DEC-80 00:00:00,800.00,,20
7499,ALLEN,SALESMAN,7698,20-FEB-81 00:00:00,1600.00,300.00,30
7521,WARD,SALESMAN,7698,22-FEB-81 00:00:00,1250.00,500.00,30
7566,JONES,MANAGER,7839,02-APR-81 00:00:00,2975.00,,20
7654,MARTIN,SALESMAN,7698,28-SEP-81 00:00:00,1250.00,1400.00,30
7698,BLAKE,MANAGER,7839,01-MAY-81 00:00:00,2850.00,,30
7782,CLARK,MANAGER,7839,09-JUN-81 00:00:00,2450.00,,10
7788,SCOTT,ANALYST,7566,19-APR-87 00:00:00,3000.00,,20
7839,KING,PRESIDENT,,17-NOV-81 00:00:00,5000.00,,10
7844,TURNER,SALESMAN,7698,08-SEP-81 00:00:00,1500.00,0.00,30
7876,ADAMS,CLERK,7788,23-MAY-87 00:00:00,1100.00,,20
7900,JAMES,CLERK,7698,03-DEC-81 00:00:00,950.00,,30
7902,FORD,ANALYST,7566,03-DEC-81 00:00:00,3000.00,,20
7934,MILLER,CLERK,7782,23-JAN-82 00:00:00,1300.00,,10

empno,startdate,enddate,job,sal,comm,deptno,chgdesc
7369,17-DEC-80 00:00:00,,CLERK,800.00,,20,New Hire
7499,20-FEB-81 00:00:00,,SALESMAN,1600.00,300.00,30,New Hire
7521,22-FEB-81 00:00:00,,SALESMAN,1250.00,500.00,30,New Hire
7566,02-APR-81 00:00:00,,MANAGER,2975.00,,20,New Hire
7654,28-SEP-81 00:00:00,,SALESMAN,1250.00,1400.00,30,New Hire
7698,01-MAY-81 00:00:00,,MANAGER,2850.00,,30,New Hire
7782,09-JUN-81 00:00:00,,MANAGER,2450.00,,10,New Hire
7788,19-APR-87 00:00:00,12-APR-88 00:00:00,CLERK,1000.00,,20,New Hire
7788,13-APR-88 00:00:00,04-MAY-89 00:00:00,CLERK,1040.00,,20,Raise
7788,05-MAY-90 00:00:00,,ANALYST,3000.00,,20,Promoted to Analyst
7839,17-NOV-81 00:00:00,,PRESIDENT,5000.00,,10,New Hire
7844,08-SEP-81 00:00:00,,SALESMAN,1500.00,0.00,30,New Hire
7876,23-MAY-87 00:00:00,,CLERK,1100.00,,20,New Hire
7900,03-DEC-81 00:00:00,14-JAN-83 00:00:00,CLERK,950.00,,10,New Hire
7900,15-JAN-83 00:00:00,,CLERK,950.00,,30,Changed to Dept 30
7902,03-DEC-81 00:00:00,,ANALYST,3000.00,,20,New Hire
7934,23-JAN-82 00:00:00,,CLERK,1300.00,,10,New Hire
```

Import the CSV file into Oracle.

Method 1: Use Oracle’ s SQL*Loader to import data. For details, refer to [Oracle SQL*Loader Overview](http://www.oracle.com/technetwork/database/enterprise-edition/sql-loader-overview-095816.html “Oracle SQL*Loader Overview”).

Method 2: Use Oracle SQL Developer to import data. For details, refer to SQL Developer Concepts and Usage.

Troubleshooting

Problem

During the execution of data export script, the system displays a message that a directory cannot be created, as shown below.

```
[root@oraclexe ~]# ./ppas_exp_all_tables_to_csv.sh rm-2ze466l5u1k657yyn.ppas.rds.aliyuncs.com 3433 myadmin
ora
mkdir: Cannot create directory: "/tmp/ppas_tables_rm-
2ze466l5u1k657yyn.ppas.rds.aliyuncs.com_3433_myadmin_ora": file already exists
```

Handling process

Delete the existing directory.

```
[root@oraclexe ~]# rm -rf /tmp/ppas_tables_rm-2ze466l5u1k657yyn.ppas.rds.aliyuncs.com_3433_myadmin_ora
```

Migrate RDS for MySQL data to the local MySQL database

RDS for MySQL supports the migration of cloud data to the local database by using physical and logical backup files.

Export using a physical backup file

Background information

Due to software restrictions, data recovery is supported only in Linux currently. If you want to recover data to Windows, you need first of all recover data to Linux and then migrate the data to Windows.

Prerequisites

RDS adopts the open source software Percona XtraBackup 2.0.6 to perform full physical backup on

the MySQL database. You need to download the software for data recovery.

The official website for Percona XtraBackup is http://www.percona.com/. Download the version compatible with your operating system.

For example:

Download the RHEL6/x86_64 version and run the rpm command to install it.

```
sudo rpm -ivh percona-xtrabackup-2.0.6-521.rhel6.x86_64.rpm
```

Operation procedure

This example assumes that the local server runs the RHEL6/x64 system and the path to the backup file is */home/mysql/*.

Download the RDS **physical backup file** and upload the file to the target server.

For details about how to obtain the backup file, refer to Downloading Backup Data.

If the target server can access the source instance, you can use wegt "url" to download the backup file. *url* indicates the backup file download address.

Switch to the backup file path.

```
cd /home/mysql/
```

Decompress the backup file.

```
tar vizxf filename.tar.gz
```

filename.tar.gz indicates the name of the backup file.

Check whether the databases contained in the decompressed file are correct.

```
cd filename/
ll
```

The system displays the following information, in which *db0dz1rv11f44yg2*, *mysql*, and *test* are the databases in RDS:

```
-rw-r--r-- 1 root root       269 Aug 19 18:15 backup-my.cnf
drwxr-xr-x 2 root root 4096 Aug 21 10:31 db0dz1rv11f44yg2
```

```
-rw-rw---- 1 root root 209715200 Aug 7 10:44 ibdata1
drwxr-xr-x 2 root root 4096 Aug 21 10:31 mysql
drwxr-xr-x 2 root root 4096 Aug 21 10:31 test
-rw-r--r-- 1 root root 10 Aug 19 18:15 xtrabackup_binary
-rw-r--r-- 1 root root 23 Aug 19 18:15 xtrabackup_binlog_info
-rw-r--r-- 1 root root 77 Aug 19 18:15 xtrabackup_checkpoints
-rw-r--r-- 1 root root 2560 Aug 19 18:15 xtrabackup_logfile
-rw-r--r-- 1 root root 72 Aug 19 18:15 xtrabackup_slave_info
```

Recover the data file.

```
innobackupex --defaults-file=./backup-my.cnf --apply-log ./
```

Data is successfully recovered when the system displays innobackupex: completed OK!.

Modify the configuration file.

Comment out *innodb_fast_checksum*, *innodb_page_size*, and *innodb_log_block_size* in the decompressed file *backup-my.cnf*, and add *datadir=/home/mysql*, as shown in the figure below.

```
# This MySQL options file was generated by innobackupex-1.5.1.

# The MySQL Server
[mysqld]
innodb_data_file_path=ibdata1:200M:autoextend
innodb_log_files_in_group=2
innodb_log_file_size=524288000
#innodb_fast_checksum=0
#innodb_page_size=16364
#innodb_log_block_size=512
datadir=/home/mysql/
```

Reinstall MySQL and obtain the root permission of the database.

```
rm -rf mysql
mysql_install_db --user=mysql --datadir=/home/mysql/
```

MySQL is successfully reinstalled when the system displays the following information:

```
Installing MySQL system table...
OK
Filling help table...
OK
```

Modifv the file owner.

```
chown -R mysql:mysql /home/mysql/
```

Start the MySQL process.

```
mysqld_safe --defaults-file=/home/mysql/backup-my.cnf &
```

Log on to the database from a client.

```
mysql –u root –p
```

Verify database integrity.

```
show databases;
```

The database is successfully recovered when the system displays the following information:

```
+--------------------+
| Database |
+--------------------+
| information_schema |
| db0dz1rv11f44yg2 |
| mysql |
| performance_schema |
| test |
+--------------------+
```

Export using a logical backup file

This example assumes that the local server runs the RHEL6/x64 system and the path to the backup file is */home/mysql/*.

Operation procedure

Download the RDS **logical backup file** and upload the file to the target server.

For details about how to obtain the backup file, refer to Downloading Backup Data.

If the target server can access the source instance, you can use wegt "url" to download the backup file.*url* indicates the backup file download address.

Switch to the backup file path.

```
cd /home/mysql/
```

Decompress the backup file.

```
tar vizxf filename.tar.gz
```

filename.tar.gz indicates the name of the backup file.

Decompress the SQL file.

```
gunzip filename.sql.gz
```

filename.sql.gz indicates the name of the compressed SQL file.

Perform logical import to import data to the target database.

```
mysql -u userName -p -h hostName -P port dbName < filename.sql
```

filename.sql indicates the name of the decompressed SQL file.

Migrate RDS for SQL Server data to the local SQL Server database

RDS for MySQL supports the migration of cloud data to the local database by using physical backup files.

Operation procedure

Download the full and incremental **physical backup files** of RDS and upload the files to the target server.

For details about how to obtain the backup file, refer to Downloading Backup Data.

If the target server can access the source instance, you can use wegt "url" to download the

backup file.*url* indicates the backup file download address.

After download, decompress the full physical backup file and incremental physical backup file.

A backup file is named in the format of *database name+backup type+date and time+task ID.bak*, of which *backup type* may be one of the following:

datafull: Specifies full backup, such as *rdsumu2myfzbeai1_datafull_201402250050_2250050.bak*.

datadiff: Specifies incremental backup, such as *rdsumu2myfzbeai1_datadiff_201402260050_2260050.bak*.

log: Specifies log backup, such as *rdsumu2myfzbeai1_log_201402260050_2260050.bak*.

Obtain the decompressed full backup file and incremental backup file. This example assumes that the backup files are stored in the following paths:

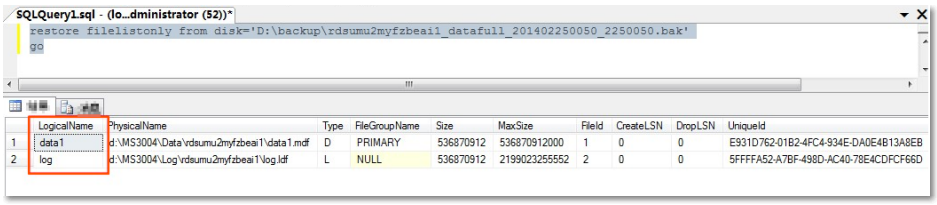
Path to the full backup file:
d:\backup\rdsumu2myfzbeai1_datafull_201402250050_2250050.bak

Path to the incremental backup file:
d:\backup\rdsumu2myfzbeai1_datadiff_201402260050_2260050.bak

Log on to the local SQL Server Console and query the logical names of the RDS files based on the backup files.

```
restore filelistonly from
disk=&apos;d:\backup\rdsumu2myfzbeai1_datafull_201402250050_2250050.bak&apos;
go
```

The system displays the following information, where the logical name of the data file is *data1* and that of the log file is *log*:



Load the full backup file.

```
restore database rdsumu2myfzbeai1 from
disk=&apos;d:\backup\rdsumu2myfzbeai1_datafull_201402250050_2250050.bak&apos; with
replace,norecovery,stats=10,
move &apos;data1&apos; to &apos;d:\database\rdsumu2myfzbeai1\data\data1.mdf&apos;,
move &apos;log&apos; to &apos;d:\database\rdsumu2myfzbeai1\log\log.ldf&apos;
go
```

Parameters description:

d:\database\rdsumu2myfzbeai1\data is the data address, and *data1.mdf* is the logical name of the data file

d:\database\rdsumu2myfzbeai1\log is the log address, and *log.ldf* is the logical name of the log file

After the script is executed, the database *rdsumu2myfzbeai1* will be in *Recovering* state.

NOTE: If you only want to recover full backup data, skip Step 6 and proceed to Step 7. If you also want to recover incremental backup data, perform Step 6.

Load the incremental backup file.

```
restore database rdsumu2myfzbeai1 from
disk=&apos;D:\backup\rdsumu2myfzbeai1_datadiff_201402260050_2260050.bak&apos; with
replace,norecovery,stats=10,
move &apos;data1&apos; to &apos;d:\database\rdsumu2myfzbeai1\data\data1.mdf&apos;,
move &apos;log&apos; to &apos;d:\database\rdsumu2myfzbeai1\log\log.ldf&apos;
go
```

After the script is executed, the database *rdsumu2myfzbeai1* will be in *Recovering* state.

Recover the database.

```
restore database rdsumu2myfzbeai1 with recovery
go
```

After the script is executed, the database *rdsumu2myfzbeai1* will be available.

Migrate RDS for PostgreSQL data to the local PostgreSQL database

RDS for PostgreSQL supports the migration of cloud data to the local database by using logical backup files.

Operation procedure

Connect the PostgreSQL client to RDS.

Run the following command to back up the data.

```
pg_dump -U username -h hostname -p port databasename -f filename
```

Parameters description:

username: Indicates the username used for database login

hostname: Indicates the host name of the database

port: Indicates the database port number

databasename: Indicates the name of the database you want to back up

filename: Name of the backup file to be generated

For example:

```
pg_dump -U myuser -h rds2z2tp80v3752wb455.pg.rds.aliyuncs.com -p 3433 pg001 -f pg001.sql
```

Save the *pg001.sql* backup file to the target server.

Run the following command to recover data to the local database:

```
psql -U username -h hostname -d desintationdb -p port -f dumpfilename.sql
```

Parameters description:

username: Indicates the username used for database login

hostname: Indicates the database address

port: Indicates the database port number

databasename: Indicates the database name

filename: Indicates the backup file name

For example:

```
psql -U myuser -h localhost -d pg001 -p 5432 -f pg001.sql
```

Since the permission configuration of the RDS database is inconsistent with that of the local database, some permission related warnings or errors may occur during data import. They can be ignored, for example:

```
WARNING:  no privileges could be revoked for "xxxxx"
ERROR: role "xxxxx" does not exist
```

Migrate RDS for PPAS to local PPAS

ApsaraDB for PPAS supports the migration of cloud data to the local database by using logical backup files.

Operation procedures

Connect the PostgreSQL client to RDS.

Run the following command to back up the data.

```
pg_dump -U username -h hostname -p port databasename -f filename
```

Parameters are described as follows:

username: Indicates the username used for database login

hostname: Indicates the host name of the database

port: Indicates the database port number

databasename: Indicates the name of the database you want to back up

filename: Name of the backup file to be generated

For example:

```
pg_dump -U ppas_user -h rdsv07z563m7o25cj550public.ppas.rds.aliyuncs.com -p 3433 edb -f ppas.sql
```

Save the *ppas.sql* backup file to the target server.

Run the following command to recover data to the local database:

```
psql -U username -h hostname -d desintationdb -p port -f dumpfilename.sql
```

Parameters are described as follows:

username: Indicates the username used for database login

hostname: Indicates the database address

port: Indicates the database port number

databasename: Indicates the database name

filename: Indicates the backup file name

For example:

```
psql -U ppas_user -h localhost -d edb -p 5444 -f ppas.sql
```

As the permission settings of the RDS database are different from those of the local database, some permission related warnings or errors may occur during data import. They can be ignored, for example:

```
WARNING:  no privileges could be revoked for "xxxxx"
```

```
ERROR: role "xxxxx" does not exist
```

Typical application

Cached data persistence

RDS can be used together with AliCloudDB for Memcached and AliCloudDB for Redis to form a storage solution with high throughput and low delay. The following section describes the cached data persistence solution based on the combined use of RDS and AliCloudDB for Memcached.

Background information

Compared with the RDS, the RDS cache product has two features:

- High response speed: The request delay of the RDS for Memcached and the RDS for Redis is usually within several milliseconds
- The cache area can support a higher QPS (Requests Per Second) than the RDS

System requirements

Bmemcached (with support of SASL extension) has been installed in the local environment or ECS.

The bmemcached download address is https://github.com/jaysonsantos/python-binary-memcached.

The bmemcached installation command is as follows:

```
pip install python-binary-memcached
```

- Python is used as an example. Python and pip must be installed in the local environment or ECS.

Sample code

The following sample code realizes the combined use of RDS and AliCloudDB for Memcached:

```
#!/usr/bin/env python
import bmemcached
Memcached_client = bmemcached.Client((‘ip:port’), ‘user’, ‘passwd’)
#Search for a value in AliCloudDB for Memcached
res = os.client.get(‘test’)
if res is not None:
return res #Return the searched value
else:
#Query RDS if the value is not found
res = mysql_client.fetchone(sql)
Memcached_client.put(‘test’, res) #Write cached data to AliCloudDB for Memcached
return res
```

Multi-structure data storage

The OSS is a cloud storage service provided by Alibaba Cloud, featuring massive capacity, security, low cost, and high reliability.The RDS can work with the OSS to form multiple types of data storage solutions.

For example, when the business application is a forum and the RDS works with the OSS, resources such as registered users' images and post content images can be stored in the OSS to reduce the storage pressure of the RDS.

Sample code

Example on the combined use of OSS and RDS.

Initialize OssAPI.

```
from oss.oss_api import *
endpoint=” oss-cn-hangzhou.aliyuncs.com”
accessKeyId, accessKeySecret=” your id” ,” your secret”
oss = OssAPI(endpoint, accessKeyId, accessKeySecret)
```

Create a bucket.

```
#Set the bucket to private-read-write
res = oss.create_bucket(bucket,"private")
print "%s\n%s" % (res.status, res.read())
```

Upload an object.

```
res = oss.put_object_from_file(bucket, object, &amp;quot;test.txt")
print "%s\n%s" % (res.status, res.getheaders())
```

Obtain the corresponding object.

```
res = oss.get_object_to_file(bucket, object, "/filepath/test.txt")
print "%s\n%s" % (res.status, res.getheaders())
```

In the ECS application code, RDS stores the ID of each user, and OSS stores the avatar resource of the user. The Python code is as follows:

```
```
#!/usr/bin/env python
from oss.oss_api import *
endpoint=” oss-cn-hangzhou.aliyuncs.com”
accessKeyId, accessKeySecret=” your id” ,” your secret”
oss = OssAPI(endpoint, accessKeyId, accessKeySecret)
user_id = mysql_client.fetch_one(sql)#Search for user_id in RDS
#Obtain and download the user avatar to the corresponding path
oss.get_object_to_file(bucket, object, your_path/user_id+’ .png’)
#Process the uploaded user avatar
oss.put_object_from_file(bucket, object, your_path/user_id+’ .png’)
```
```

Appendix

Commonly used SQL commands (MySQL)

Instruction

This document lists some of the commonly used SQL commands. Only the sytaxes are explained. For the detailed information on SQL commands, including command parameters and restrictions, see MySQL 5.7 Reference Manual.

Database-related commands

Operation	Command

Create a database and designate a character set	create database db01 DEFAULT CHARACTER SET gbk COLLATE gbk_chinese_ci;
Delete a database	drop database db01;

Account-related commands

Note: If an instance has the high-privilege account, the passwords of other accounts/users under this instance cannot be changed through the high-privilege account. If the password needs to be changed, you need to delete this account and create a new one.

Operation	Command
Create an account	CREATE USER 'username'@'host' IDENTIFIED BY 'password';
Delete an account	DROP USER 'username'@'host';
Authorization	GRANT SELECT ON db01.* TO 'username'@'host';
Query the created accounts in the database	SELECT user,host,password FROM mysql.user_view; or show grants for xxx
Reclaim the authority	- Reclaim all the authority REVOKE ALL PRIVILEGES,GRANT OPTION FROM 'username'@'host'; - Reclaim the specified authority REVOKE UPDATE ON *.* FROM 'username'@'host';