TaurusDB for PostgreSQL

FAQs

Issue 01

Date 2025-11-14

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Product Consulting

1.1 Is There Anything Special I Need to Note When Using TaurusDB for PostgreSQL?

1. DB instance operating systems (OSs) are invisible to you. Your applications can access a database only through an IP address and a port.
2. The backup files stored in Object Storage Service (OBS) and the Elastic Cloud Server (ECS) used by ECS are invisible to you. They are visible only to the TaurusDB for PostgreSQL instance management system.
3. Before viewing the DB instance list, ensure that the region is the same as the region where the DB instance is purchased.
4. After purchasing TaurusDB for PostgreSQL instances, you do not need to perform basic database O&M operations, such as applying HA and security patches. However, you must still pay attention to whether:
   a. The CPU, input/output operations per second (IOPS), and space of the TaurusDB for PostgreSQL instance are sufficient. If any of these becomes insufficient, scale up the instance.
   b. The performance of the TaurusDB for PostgreSQL instances is adequate, a large number of slow SQL queries exist, SQL statements need to be optimized, or any indexes are redundant or missing.

1.2 Will My TaurusDB for PostgreSQL Instances Be Affected by Other Users' Instances?

No. Your TaurusDB for PostgreSQL instances and resources are isolated from other users' TaurusDB for PostgreSQL instances.

1.3 Will Different TaurusDB for PostgreSQL Instances Share CPU and Memory Resources?

Yes, that depends on the instance specifications.

- General-purpose:

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

- Dedicated:

  Your instance gets dedicated vCPUs and memory, so the performance is stable. It is not affected by other instances on the same physical machine. Dedicated instances are good for scenarios that require stable performance.

1.4 How Long Does It Take to Create a TaurusDB for PostgreSQL Instance?

It takes 5 to 7 minutes to create a single-node or primary/standby instance.

If creating an instance takes much more time than described above, there may be problems during the creation. In this case, contact customer service by choosing **Service Tickets > Create Service Ticket** in the upper right corner of the management console.

1.5 What Can I Do About Slow Responses of Websites When They Use TaurusDB for PostgreSQL?

To solve this problem:

- Check the performance of TaurusDB for PostgreSQL instances on the management console.
- Compare the database connection status of local databases and TaurusDB for PostgreSQL instances. This problem depends on web applications.

1.6 What Is the Time Delay for Primary/Standby Replication?

When standby instances cannot keep up with the updates on the primary, this generates replication delay. If the standby SQL and I/O thread are running, the replication delay is a positive value measured in seconds. If the standby SQL thread is not running, or if the SQL thread has consumed all of the relay log and the standby I/O thread is running, then it is **NULL** (undefined or unknown)

The delay for primary/standby replication cannot be calculated using a formula as the delay is affected by the following factors:

- Network communication status
- Transaction workload on the primary DB instance in transactions per second (TPS)
- The size of the transaction executed by the primary DB instance (this affects the duration of transaction executions)

- Load balancing of the standby DB instance

If the primary DB instance has a heavy load for a certain period of time and executes a large number of transactions per second, replication to the standby DB instance will be delayed. This delay is generally a few seconds.

1.7 Can Multiple ECSs Connect to the Same TaurusDB for PostgreSQL Instance?

Multiple ECSs can connect to the same TaurusDB for PostgreSQL instance as long as the capability limits of a database are not exceeded.

1.8 What Is the Availability of TaurusDB for PostgreSQL Instances?

Formula for the TaurusDB for PostgreSQL instance availability:

DB instance availability = (1 – Failure duration/Total service duration) × 100%

1.9 Does TaurusDB for PostgreSQL Support Cross-AZ High Availability?

Yes. When you buy a DB instance, you can select **Primary/Standby** for **DB Instance Type** and then select different AZs for **Primary AZ** and **Standby AZ**.

NOTE

TaurusDB for PostgreSQL does not support 3-AZ deployment.

An AZ is a physical region where resources have independent power supplies and networks. AZs are physically isolated but interconnected through an internal network. You can deploy your instance across AZs in some regions.

For example, if your VPC subnet is deployed in AZ 3, you can select AZ 1 and AZ 2 that are in the same region as AZ 3 when buying a primary/standby instance. The three AZs can communicate with each other.

TaurusDB for PostgreSQL allows you to deploy primary/standby instances in an AZ or across AZs. You can determine whether the standby AZ is the same as the primary AZ.

- If they are different (default setting), the primary and standby instances are deployed in different AZs to ensure failover support and high availability.
- If they are the same, the primary and standby instances are deployed in the same AZ. If an AZ failure occurs, high availability cannot be ensured.

1.10 Can TaurusDB for PostgreSQL Primary/Standby DB Instances Be Changed to Single DB Instances?

No. You can use Data Replication Service (DRS) or the export and import tool of the client to migrate data from primary/standby DB instances to single DB instances.

1.11 Does TaurusDB for PostgreSQL Support CloudPond?

No. For details, see **CloudPond and Other Services**.

1.12 Can I Use an Encrypted Password to Log In to a TaurusDB for PostgreSQL Instance?

No. When you log in to a TaurusDB for PostgreSQL instance, use the password set on the console for authentication. Encrypted password authentication is not supported.

1.13 What Can I Do If I Can't Find My TaurusDB for PostgreSQL Resources?

Symptom

After I logged in to the management console, I could not find my purchased TaurusDB for PostgreSQL resources.

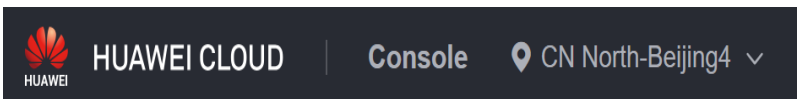
Possible Causes

- Your purchased resources are not in the selected region.
- Your purchased resources are not under the selected service.

Solution 1

1. **Log in to the management console**.
2. In the upper part of the page, switch to the region that your TaurusDB for PostgreSQL resources belong to.

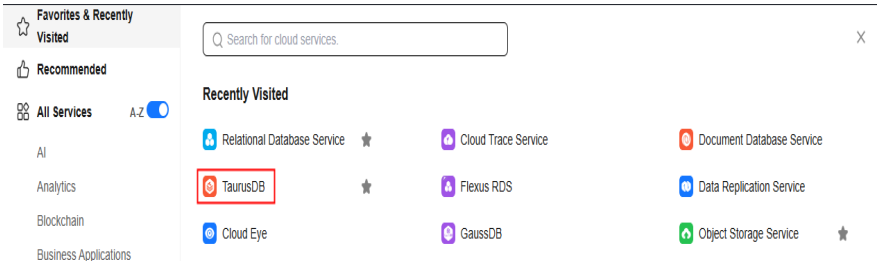
   Figure 1-1 Changing a region

3. Click ☰ in the upper left corner and select the correct service name.

To search for TaurusDB for PostgreSQL instances, choose **Databases** > **TaurusDB**.

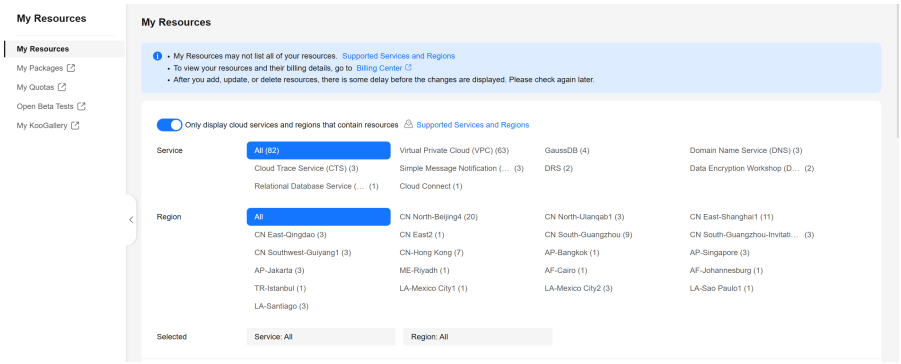
Figure 1-2 Selecting a service



Solution 2

1. **Go to the My Resources page**.
2. Select the correct service and region to view the purchased resources.

Figure 1-3 My Resources



2 Resource and Disk Management

2.1 Which Types of Logs and Files Occupy TaurusDB for PostgreSQL Storage Space?

The following logs and files occupy TaurusDB for PostgreSQL storage space.

Table 2-1 PostgreSQL database file types

DB Engine	File Type
PostgreSQL	Log files: database error log and transaction log files
	Data files: database content, index, replication slot data, transaction status data, and database configuration files
	Other files: temporary files

Solution

1. If the original storage space is insufficient as your workloads grow, scale up storage space of your DB instance.
2. If data occupies too much storage space, run **DROP** or **TRUNCATE** to delete useless historical table data to release storage space. If no historical data can be deleted, scale up your storage space.
3. If temporary files generated by sorting queries occupy too much storage space, optimize your SQL query statements.
   a. A large number of temporary files are generated if there are a large number of sorting queries executed by applications.
   b. A large number of WAL logs are generated and occupy space if there are large amounts of insert, delete, and update operations in a short period.
4. Use Cloud Eye to monitor the size, usage, and utilization of storage space of your DB instance and set alarm policies.

2.2 Which Items Occupy the Storage Space of My TaurusDB for PostgreSQL Instance?

Both your regular data (excluding backup data) and the data required for the operation of your TaurusDB for PostgreSQL instances (such as system database data, WAL logs, and indexes) take up storage space on your TaurusDB for PostgreSQL instances. The storage space includes the file system overhead required for inode, reserved blocks, and database operations. Log files generated by TaurusDB for PostgreSQL database servers also occupy storage space.

These files ensure the stability of TaurusDB for PostgreSQL instances.

2.3 How Much Storage Space Is Required for DDL Operations?

Data Definition Language (DDL) operations may increase storage usage sharply. To ensure that services are running properly, do not perform DDL operations during peak hours. If DDL operations are required, ensure that storage space is at least twice the tablespace size plus 10 GB. For example, if your tablespace is 500 GB, ensure that storage space is at least 1,010 GB (500 GB x 2 + 10 GB).

2.4 What Are the Differences Between the Storage Space and Backup Space of a TaurusDB for PostgreSQL Primary/Standby Instance?

In a TaurusDB for PostgreSQL primary/standby instance, the node accessed by applications is called the primary instance, and data on the primary instance is synchronized to the other node (called the standby instance) in real time. The standby instance acts as a failover target for the primary. If the primary instance fails, TaurusDB for PostgreSQL promotes the standby instance to primary for high availability.

- Storage space

  Storage space of the standby instance is the same as that of the primary instance. Scaling up the storage space of the primary instance will also scale up that of the standby instance.

  Data on the primary instance may not be synchronized to the standby instance fast enough when there are a large number of data writes or a long replication delay. If this happens, the primary instance retains WAL logs for replication. WAL logs will be stacked and occupy storage space.

- Backup space

  Backup space is used to store automated backups, manual backups, and SQL audit logs. You can get a free backup space of the same size as your purchased storage space. If the free backup space is used up, the additional space will be billed. You can configure a custom backup policy to best fit your workloads' needs.

To release backup space, see **How Do I Clear TaurusDB for PostgreSQL Backup Space?**.

3 Database Connection

3.1 What Do I Do If There Are Too Many Database Connections?

The number of database connections indicates the number of applications that can be simultaneously connected to a database, and is irrelevant to the maximum number of users allowed by your applications or websites.

If there is an excessive number of database connections, applications may fail to be connected, and the full and incremental backups may fail, affecting services.

Fault Locating

1. Check whether applications are connected, optimize the connections, and release unnecessary connections.
2. On the Cloud Eye console, view metrics of your DB instance to identify performance issues and set alarms for metric thresholds. Cloud Eye monitors metrics of different categories, including CPU, memory, storage, and connections. For details, see the *Cloud Eye User Guide*.

Solution

1. Connect to a DB instance through a private network. Using a private network prevents congestion caused by insufficient bandwidth.
2. On the management console, set the parameter **innodb_adaptive_hash_index** to **off** to reduce lock wait time.
3. Optimize slow queries.

3.2 What Is the Maximum Number of Connections to a TaurusDB for PostgreSQL Instance?

TaurusDB for PostgreSQL does not have constraints on the number of connections. It depends on the default values and value ranges of certain parameters in your DB engine.

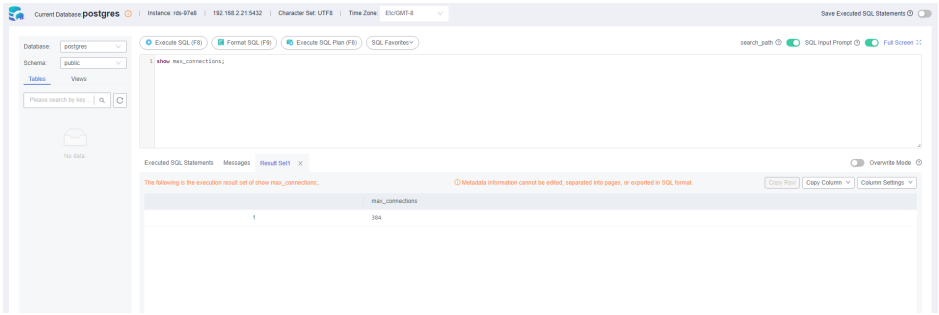
Definition

The maximum number of connections refers to the concurrent connections allowed for a DB instance.

How to Change It

- You can change the maximum number of connections allowed on the console. For details, see **Modifying Instance Parameters**.
- You can run the following command to query the maximum number of connections allowed:

```
show max_connections;
```



If you want to change the maximum number of connections by running commands, **submit a service ticket** to apply for required permissions.

Setting the Maximum Number of Connections to an Appropriate Value

The **max_connections** parameter is closely related to the memory (GB) of the DB instance.

Table 3-1 Calculation formula

Parameter	Formula	Version Involved	Unit
max_connections	min((total_mem_kb – shared_buffer) x 1,024.0/16 MB, 5,000)	All versions	Count

The following table lists the default values of **max_connections** for different memory specifications.

Table 3-2 Default values of **max_connections** for different memory specifications

Memory (GB)	Max Client Connections (max_connections)
512	5,000
384	5,000
256	5,000

Memory (GB)	Max Client Connections (max_connections)
128	5,000
64	3,072
32	1,536
16	768
8	384
4	192
2	96

3.3 What Should I Do If an ECS Cannot Connect to a TaurusDB for PostgreSQL Instance over a Private Network?

Perform the following steps to identify the problem:

Step 1 Check whether the ECS and the TaurusDB for PostgreSQL instance are located in the same VPC.

- If they are in the same VPC, go to **Step 2**.
- If they are in different VPCs, create an ECS in the VPC in which the TaurusDB for PostgreSQL instance is located.

Step 2 Check whether the security group rules of the TaurusDB for PostgreSQL instance are appropriate.

Step 3 On the ECS, check whether the TaurusDB for PostgreSQL instance port can be connected.

The default port of TaurusDB for PostgreSQL is **5432**.

```
curl -kv Private_IP_address:5432
```

- If the ECS can connect to the DB instance, no further action is required.
- If the ECS cannot connect to the DB instance, contact technical support.

----End

3.4 What Should I Do If My TaurusDB for PostgreSQL Instance Fails to Be Connected Due to Database Client Problems?

Check the following items to troubleshoot TaurusDB for PostgreSQL connection failures caused by a client problem:

1. ECS security policy

   In Windows, check whether the TaurusDB for PostgreSQL instance port is enabled in the Windows security policy. In Linux, run **iptables** to check whether the TaurusDB for PostgreSQL instance port is enabled in firewall settings.

2. Application configuration

   Check whether the connection address, port parameter configuration, and JDBC connection parameter configuration are correct.

3. Username or password

   Check whether the username or password is correct if an error similar to the following occurs during TaurusDB for PostgreSQL connection:

   - [Warning] Access denied for user 'username'@'yourIp' (using password: NO)
   - [Warning] Access denied for user 'username'@'yourIp' (using password: YES)

NOTE

If the problem persists, contact post-sales technical support.

3.5 What Should I Do If a Database Server Problem Causes a Connection Failure to a TaurusDB for PostgreSQL Instance?

Check whether any of the following problems occur on the TaurusDB for PostgreSQL instance:

1. The TaurusDB for PostgreSQL instance is not properly connected.

   Solution: Check the connection. If you connect to the TaurusDB for PostgreSQL instance over a private network, the ECS and the TaurusDB for PostgreSQL instance must be in the same VPC and the TaurusDB for PostgreSQL instance can be accessed only through an ECS in the same VPC. If you connect to the TaurusDB for PostgreSQL instance over a public network, the ECS and the TaurusDB for PostgreSQL instance can be in different VPCs.

2. The maximum number of connections has been reached.

   Solution: Use TaurusDB for PostgreSQL resource monitoring to check if the CPU usage and the number of connections are within the allowed ranges. If either of them has reached the maximum, reboot or disconnect the TaurusDB for PostgreSQL instance.

3. The TaurusDB for PostgreSQL instance is abnormal. For example, the TaurusDB for PostgreSQL instance fails to be rebooted, the system is faulty, or the instance or table is locked.

   Solution: Reboot the TaurusDB for PostgreSQL instance. If the problem persists, contact post-sales technical support.

3.6 Do Applications Need to Support Automatic Reconnections to TaurusDB for PostgreSQL Databases?

It is recommended that your applications support automatic reconnections to a database. After a database reboot, your applications will automatically reconnect to the database to increase service availability and continuity.

To reduce resource consumption and improve performance, configure your applications to connect to a database using a persistent connection.

3.7 Can I Access a TaurusDB for PostgreSQL Instance over an Intranet Across Regions?

By default, TaurusDB for PostgreSQL instances cannot be accessed over an intranet across regions. Cloud services in different regions cannot communicate with each other over an intranet. You can use Cloud Connect (CC) or Virtual Private Network (VPN) to connect to instances across regions.

- CC allows you to connect VPCs in different regions, even if they are not owned by the same account. For details, see **Communications Among VPCs of the Same Account**.
- VPN uses an encrypted tunnel to connect VPCs in different regions and sends traffic over the Internet. It is inexpensive, easy to configure, and easy to use. However, the quality of VPN connections depends on the quality of your Internet connection. For details, see **Connecting to a VPC Through a VPN**.

3.8 Why Did the New Password Not Take Effect After I Reset the Administrator Password of My TaurusDB for PostgreSQL Instance?

Possible Causes

You may have restored from a backup before you reset the administrator password.

Locating Method

Check whether the DB instance was restored after you reset the administrator password.

Solution

Log in to the console and reset the administrator password again.

3.9 Can I Access TaurusDB for PostgreSQL Standby Instances?

No. You can directly access primary instances. Standby instances are not visible to users and therefore you cannot access them directly.

TaurusDB for PostgreSQL supports primary/standby failover and switchover. Data is synchronized between the primary and standby instances in real time.

3.10 Will My Access Be Restricted by Bandwidth When I Connect to My TaurusDB for PostgreSQL Instance from an ECS over a Private Network?

No.

4 Database Migration

4.1 What Types of DB Engines Does TaurusDB for PostgreSQL Support for Importing Data?

- Exporting or importing data between DB engines of the same type is called homogeneous database export or import.
- Exporting or importing data between DB engines of different types is called heterogeneous database export or import. For example, import data from Oracle to DB engines supported by TaurusDB for PostgreSQL.

  Generally, data cannot be exported or imported between heterogeneous databases due to the different data formats involved. However, if the data formats are compatible, table data can, in theory, be migrated between them.

  Third-party software is usually required for data replication to export and import between heterogeneous databases.

4.2 Why Do I Need to Use the pg_dump Tool for Migration?

The pg_dump tool is easy to use for data migration. However, when you use this tool, the server is stopped for a long period of time during data migration. Only use these tools when there is not much data to migrate or if stopping the server for a long period of time is not an issue.

TaurusDB for PostgreSQL is compatible with source database services. The procedure for migrating data from your database to TaurusDB for PostgreSQL is similar to the procedure for migrating data from one database server to another.

5 Database Permission

5.1 Why Does the Root User of My TaurusDB for PostgreSQL Instance Not Have the Super Permissions?

TaurusDB for PostgreSQL does not provide the super permissions for the **root** user. The super permissions allow you to execute management commands, such as **reset master**, **set global**, **kill** *thread_ID*, and **reset slave**. These operations may cause primary/standby replication errors.

If you need to perform operations that require super permissions, TaurusDB for PostgreSQL provides alternative methods.

- Scenario 1: If you cannot run the following command on a TaurusDB for PostgreSQL instance to modify parameter values, you can modify parameter values through the TaurusDB for PostgreSQL console.

  set global parameter name=*Parameter value*;

  If the script contains the **set global** command, delete the **set global** command and modify parameter values on the TaurusDB for PostgreSQL console.

- Scenario 2: An error is reported after you run the following command because the **root** user does not have the super permissions. To solve this problem, delete **definer='root'** from the command.

  create definer='root'@'%' trigger(procedure)...

5.2 How Do I View Authorized Databases After a Local Client Is Connected to a TaurusDB for PostgreSQL Instance?

After connecting to the database on a local client, run the following command to grant permissions to view the database. In the command, *ip* indicates the local IP address.

show grants for root@'*ip*';

show grants for root@'%';

5.3 Can Multiple Users Log In to a TaurusDB for PostgreSQL Instance Through DAS at the Same Time? Will the Accounts Be Locked If I Enter Wrong Passwords Several Times in a Row?

Multiple users can log in to a TaurusDB for PostgreSQL instance through DAS at the same time. The passwords will not be locked after multiple failed login attempts.

If you forget the password of your database account when using TaurusDB for PostgreSQL, you can reset the password. On the **Instances** page of the TaurusDB for PostgreSQL console, locate the target DB instance and choose **More** > **Reset Password** in the **Operation** column.

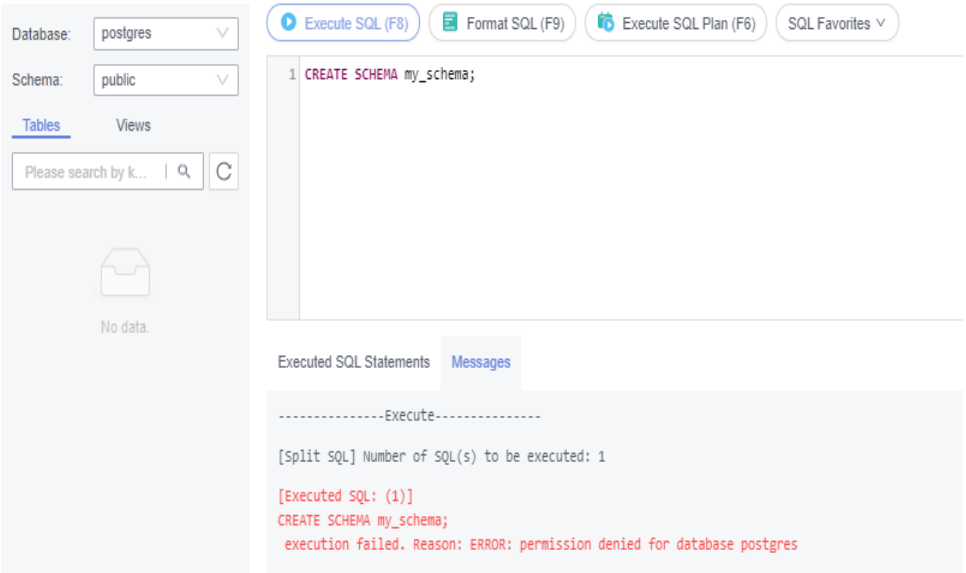
5.4 Why Did I Fail to Create an Object on the postgres Database as a Common User?

Symptom

- **Creating a Schema as a Common User**

  Command: **CREATE SCHEMA my_schema;**

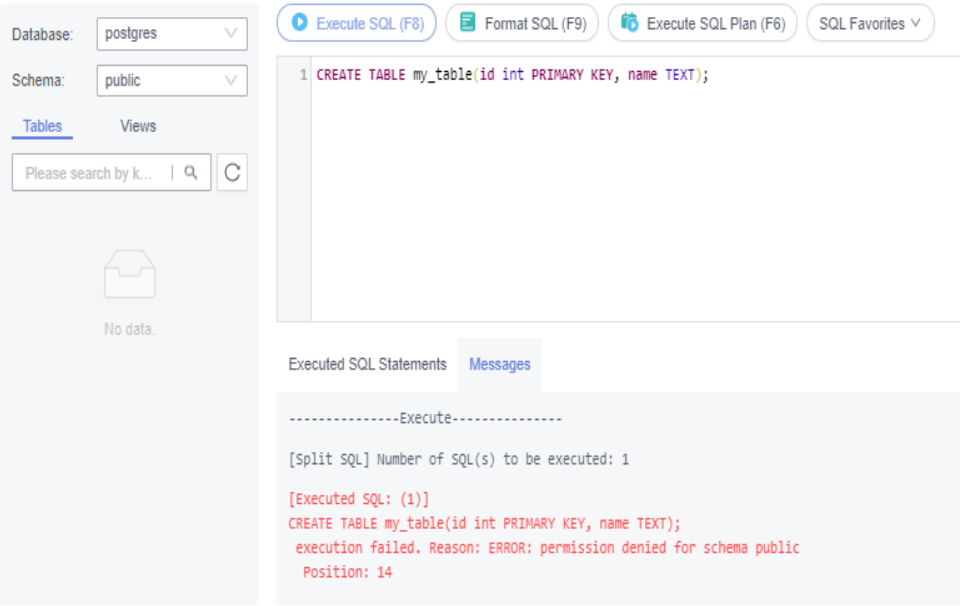
  Error: ERROR: permission denied for database postgres



- **Creating a Table as a Common User**

  Command: **CREATE TABLE my_table(id int PRIMARY KEY,name VARCHAR(30));**

  ERROR: permission denied for schema public



Solution

A common user cannot create objects on the **postgres** database, but a **root** user can.

NOTE

Switch to user **root** and ensure that your database kernel allows for root privilege escalation. For details, see **Privileges of the root User**.

Privileges of the root User

TaurusDB for PostgreSQL provides permissions for the **root** user. To create objects on a TaurusDB for PostgreSQL database without operation risks, escalate your account to root privileges when necessary.

The following table describes root privilege escalation in different versions.

Table 5-1 Privileges of the **root** user

Version	Whether to Escalate Privileges	Initial Version for Privilege Escalation
pgcore16	Yes	16.4

Escalate to root privileges when you need to:

- Create an event trigger.
- Create a wrapper.
- Create a logical replication publication.
- Create a logical replication subscription.
- Query and maintain replication sources.

- Create a replication user.
- Create a full-text index template and parser.
- Run the **vacuum** command on a system catalog.
- Run the **analyze** command on a system catalog.
- Create an extension.
- Grant an object permission to a user.

5.5 What Should I Do If a Role Failed to Be Deleted from a TaurusDB for PostgreSQL Instance?

Symptom

Role **test** failed to be deleted from a TaurusDB for PostgreSQL instance by running the **DROP ROLE test;** command on the DAS console.

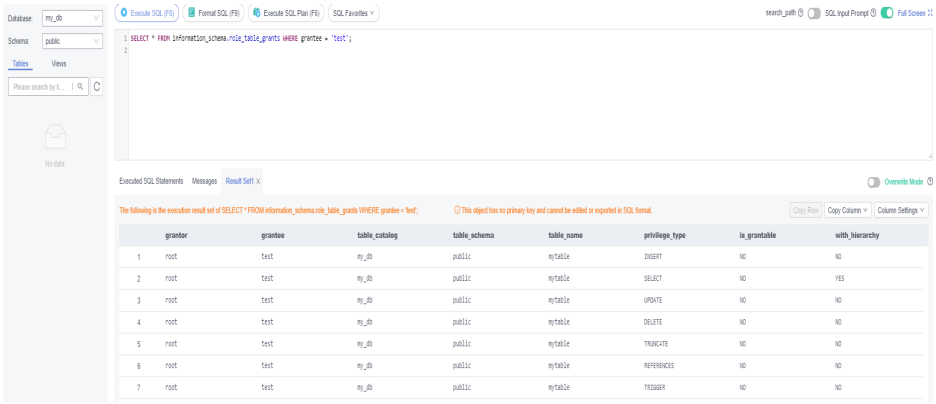
Possible Causes

Role **test** cannot be deleted probably because it is associated with objects. Before deleting this role, revoke the permissions of its associated objects.

Solution

1. Check the permissions of the role.

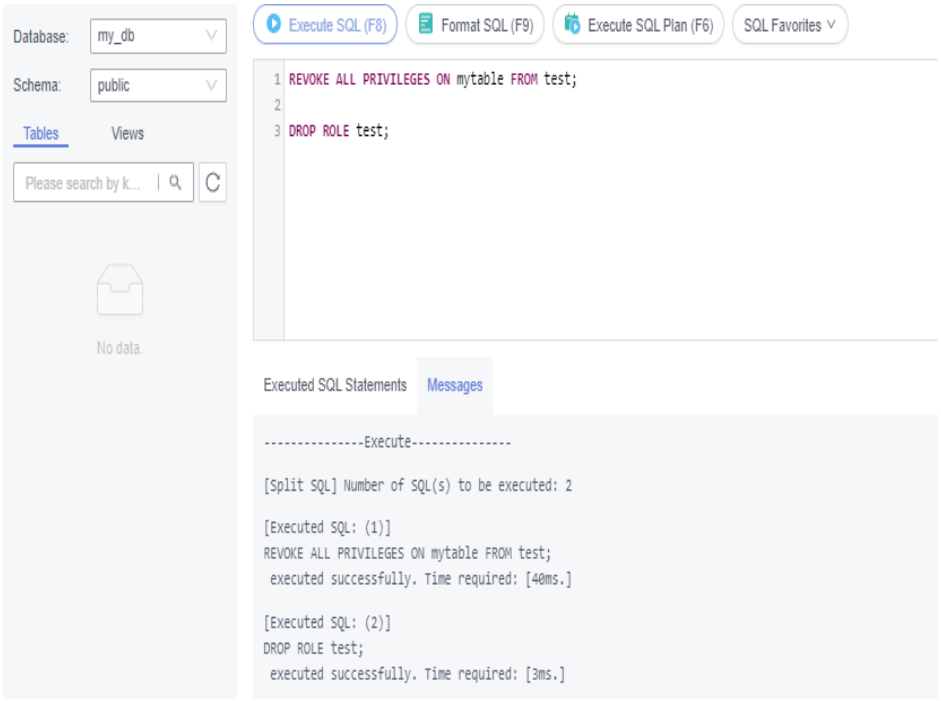
   select * from INFORMATION_SCHEMA.role_table_grants WHERE grantee='test';



2. If role **test** is associated with objects, revoke the permissions of the objects and then delete the role.

   REVOKE ALL PRIVILEGES ON mytable FROM test;

   DROP ROLE test;



5.6 Why Did My TaurusDB for PostgreSQL Migration Fail?

Symptom

An error is reported when user **root** uses DRS to migrate TaurusDB for PostgreSQL database data. In this case, you need to switch to another account or grant permissions to the current account.

Possible Causes

The migration may have failed because user **root** may not have permissions to operate certain objects when migrating the database using DRS.

Solution

1. Log in to the database as user **root** and run the following statements.
   ```
   grant USAGE on schema public to root;
   grant SELECT,REFERENCES,TRIGGER on all tables in schema public to root;
   grant EXECUTE on ALL FUNCTIONS IN SCHEMA public to root;
   ```
2. After the migration is complete, revoke the permissions.
   ```
   revoke USAGE schema public from root;
   revoke SELECT,REFERENCES,TRIGGER on all tables in schema public from root;
   revoke EXECUTE on ALL FUNCTIONS IN SCHEMA public from root;
   ```

NOTE

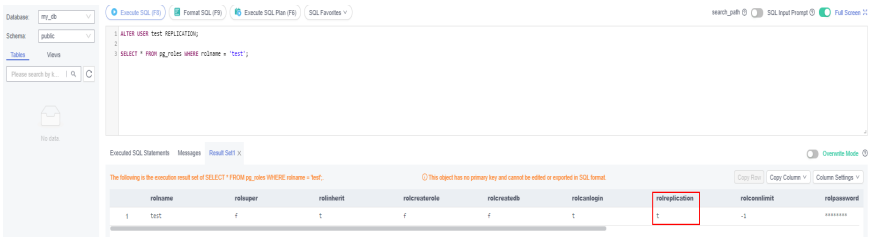
You can run the commands to grant permissions as user **root** only when your database kernel version supports root privilege escalation. For details, see **Privileges of the root User**.

5.7 How Do I Grant the REPLICATION Permission to a TaurusDB for PostgreSQL Database User?

1. Log in to the database as user **root**.
2. Grant the REPLICATION permission to your account and query the pg_roles table to verify that the permission has been granted.

   ALTER USER <user> REPLICATION;

   SELECT * FROM pg_roles;



NOTE

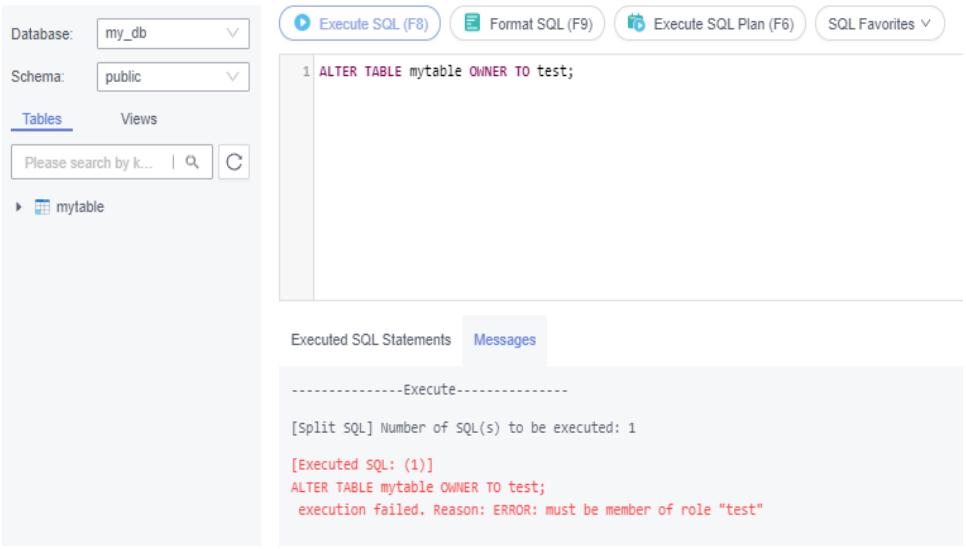
You can run the commands to grant permissions as user **root** only when your database kernel version supports root privilege escalation. For details, see **Privileges of the root User**.

5.8 Why Is An Error Reported When I Attempt to Change a Table Owner of the TaurusDB for PostgreSQL Database?

Symptom

An error is reported when the owner of a table named **mytable** is being changed to user **test**. The table is in the **my_db** database created by user **root**.

ALTER TABLE mytable OWNER TO test;



Possible Causes

The error is displayed because user **root** has not escalated the privilege.

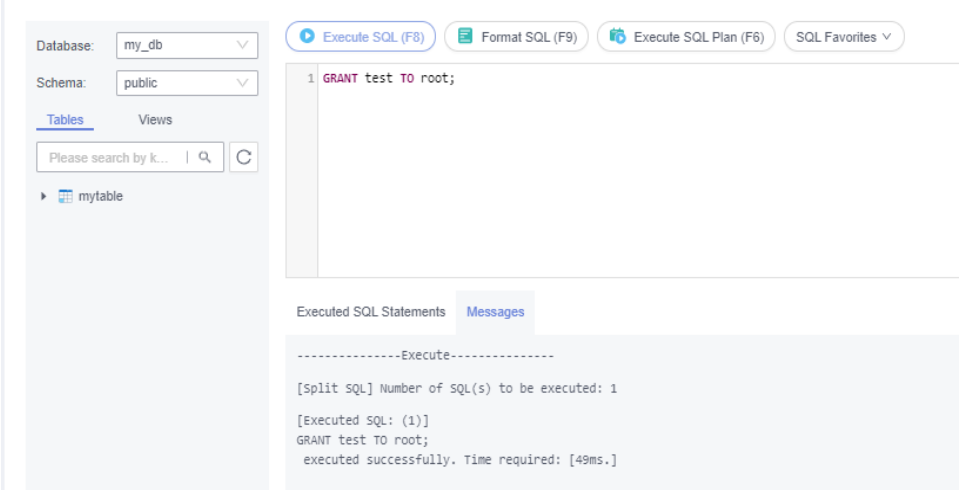
NOTE

After the privilege is escalated, user **root** has the permission to assign the table owner to another user.

Solution

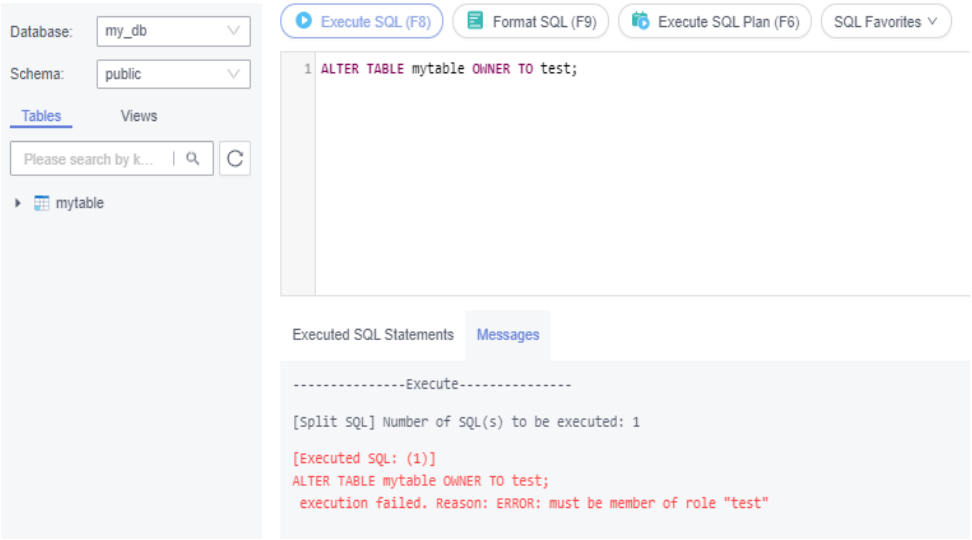
1. Log in to the database as user **test**.
2. Run the following command as user **test**:

   GRANT test TO root;



3. Log in to the database as user **root**.
4. Run the following command as user **root** to change the owner of the **mytable** table:

   ALTER TABLE mytable OWNER TO test;



6 Database Storage

6.1 What Types of Storage Does TaurusDB for PostgreSQL Use?

TaurusDB for PostgreSQL uses Elastic Volume Service (EVS) disks for storage. For details, see ***Elastic Volume Service Service Overview***.

The TaurusDB for PostgreSQL backup data is stored in OBS and does not occupy the database storage space. For details on the TaurusDB for PostgreSQL instance storage configuration, see **Object Storage Service User Guide**.

6.2 How Do I View the Storage Usage of My TaurusDB for PostgreSQL Instance?

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

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

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

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

Step 5 On the **Basic Information** page, view the storage space usage in the **Storage & Backup** area.

----End

7 Database Usage

7.1 How Do I View Session IDs and Login and Logout Time of a TaurusDB for PostgreSQL Database?

To view sessions, run the **show processlist;** command in the database.

7.2 What Should I Do If Garbled Characters Are Displayed After SQL Query Results Are Exported to an Excel File for My TaurusDB for PostgreSQL Instance?

The default code is utf8. You need to convert the default code to Unicode in the exported Excel file.

7.3 Does the OPTIMIZE TABLE Operation Lock Tables on a TaurusDB for PostgreSQL Instance?

When the OPTIMIZE TABLE operation is performed on a TaurusDB for PostgreSQL instance, the tables are locked only for a short period of time. During the table locking period, DML operations can be performed but DDL operations cannot. DML will recreate tables, which consumes CPU and disk resources. If there are a large number of concurrent DML operations, the table will be locked for longer. To avoid impacting services, perform the OPTIMIZE TABLE operation during off-peak hours.

8 Backup and Restoration

8.1 How Do I View My Backup Space Usage?

Step 1 Log in to the management console.

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

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

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

Step 5 On the **Basic Information** page, check the backup space usage in the **Storage/Backup Space** area.

NOTE

The storage spaces of primary and standby instances are the same because they both need to hold the same amount of data. Free backup storage equal to your purchased storage space is also provided. If free backup space is used up, the additional space will be billed. You need to configure an automated backup policy before using the backup space.

----End

8.2 How Is TaurusDB for PostgreSQL Backup Data Billed?

All the TaurusDB for PostgreSQL full and binlog backups are stored on OBS without occupying the storage of your DB instances. TaurusDB for PostgreSQL provides free backup space of the same size as your purchased storage.

The lifecycle of automated backups is the same as that of the DB instance. If you delete a DB instance, its automated backups are also deleted, but manual backups are not. For details, see **Deleting a Manual Backup**.

For example, if you purchase a DB instance with 200 GB of storage, you can get an additional 200 GB of backup space and will only be billed for backups in excess of 200 GB. The first 200 GB of backup data is free. When the 200 GB storage is used up, the backups will be billed on a pay-per-use basis.

NOTICE

If your storage is frozen, it is no longer billed and the free backup space is also unavailable.

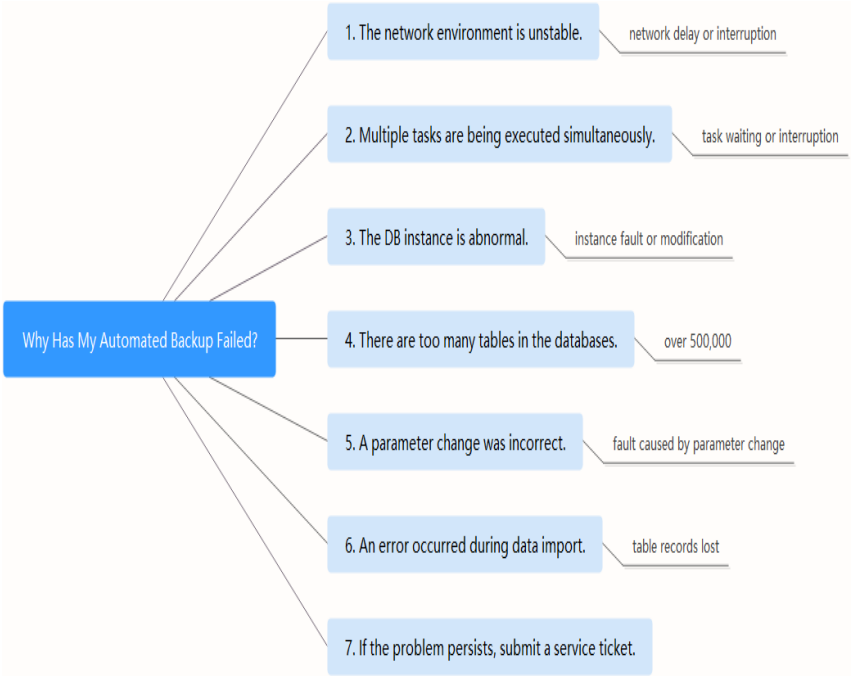
If your DB instance is frozen, no free backup space is available and the original backups generated before the instance is frozen will be billed.

- If you unfreeze the DB instance, the free backup space will be restored.
- If you delete the frozen DB instance, the existing automated backups will also be deleted. You need to manually delete the existing manual backups. After all the backups are removed, the backup space will no longer be billed.

8.3 Why Did My Automated Backup Fail?

The following figure shows the possible reasons for automated backup failures.

Figure 8-1 Reasons why automated backup fails



- The network environment may be unstable due to problems such as network delay or interruptions.

  If TaurusDB for PostgreSQL detects any of these problems, it triggers another automated backup half an hour later. Alternatively, you can perform a manual backup immediately.

- If multiple tasks are being executed simultaneously, there can be problems such as excessive task wait times or interruptions.

  If TaurusDB for PostgreSQL detects any of these problems, it triggers another automated backup half an hour later. Alternatively, you can perform a manual backup immediately.

- The DB instance is abnormal probably because it is faulty or being modified.

  If TaurusDB for PostgreSQL detects any of these problems, it triggers another automated backup half an hour later. Alternatively, you can perform a manual backup immediately.
- The backup speed depends on how many tables there are in the databases.

  If the number of tables exceeds 500,000, the backup will fail.
- A parameter change was incorrect.

  If your DB instance becomes faulty after you modify parameters of a parameter template and apply the template to the instance, check whether the modified parameters are set to correct values and whether there are any associated parameters that need to be changed, or reset the parameters to their defaults and reboot the DB instance.
- An error occurred during data import.

  If system catalog records are lost due to inappropriate data import, import the data again.
- If the problem persists, **submit a service ticket**.

8.4 Why Was Data Table Lost or Data Deleted from My TaurusDB for PostgreSQL Instance?

TaurusDB for PostgreSQL does not delete or perform any operations on any user data. If this problem occurs, check if there have been any misoperations and restore the data from backup files, if necessary.

Check for misoperations: If SQL auditing has been enabled, view data execution records in audit logs.

Restore data using backup files:

- Use the TaurusDB for PostgreSQL restoration function.
- Import the backup data from the ECS to TaurusDB for PostgreSQL.

8.5 How Long Does TaurusDB for PostgreSQL Store Backup Data?

Automated backup data is kept based on the backup retention period you specified.

There is no limit for the manual backup retention period. You can delete manual backups as needed.

The backup data is stored in OBS and does not occupy the database storage space you purchased.

8.6 How Do I Clear TaurusDB for PostgreSQL Backup Space?

The TaurusDB for PostgreSQL backup space stores automated backups, manual backups, and SQL audit logs.

- **Automated full and incremental backups**

  Automated backups cannot be manually deleted. To delete them, you can adjust the retention period specified in your backup policy. Retained backups will be automatically deleted at the end of the retention period.

- **Manual full backups**

  You can manually delete manual backups.

8.7 Can My TaurusDB for PostgreSQL Instance Still Be Used in the Backup Window?

A backup window is a user-specified time during which TaurusDB for PostgreSQL instances are backed up. With these periodic data backups, TaurusDB for PostgreSQL allows you to restore DB instances to a point in time within the backup retention period.

- During the backup window, you can still use your instance except rebooting it on the console.
- When starting a full backup task, TaurusDB for PostgreSQL first tests connectivity to your instance. If either of the following conditions is met, the test fails and a retry is performed. If the retry fails, the backup task fails.
  - DDL operations are being performed on the DB instance.
  - The backup lock fails to be obtained from the DB instance.

8.8 How Can I Back Up a TaurusDB for PostgreSQL Database to an ECS?

You can back up data to an ECS the same way you export SQL statements. The ECS service does not have restrictions on the types of data to be backed up as long as the data complies with local laws and regulations. You can store TaurusDB for PostgreSQL backup data on an ECS, but using an ECS is not recommended.

You are advised to use automated backup and manual backup to back up data to OBS for higher data reliability and service assurance.

8.9 Can I Dump TaurusDB for PostgreSQL Backups to My OBS Bucket?

No. Backups cannot be directly dumped to your OBS bucket.

You can download full backups or incremental backups to your local PC and dump them to your OBS bucket using OBS Browser+.

8.10 Does TaurusDB for PostgreSQL Support Table PITR?

No.

You can use a manual or an automated backup to restore data to the status when the backup was created. This operation restores the data of the entire DB instance. For details, see **Restoration Solutions**.

9 Database Monitoring

9.1 Which TaurusDB for PostgreSQL Instance Metrics Do I Need to Pay Attention To?

You need to pay attention to CPU, memory, and storage space usage.

You can configure the system to report alarms based on service requirements and take measures to handle any reported alarms.

Configuration examples:

- Configure TaurusDB for PostgreSQL to report alarms to Cloud Eye if its CPU utilization reaches or exceeds a specific value (for example, 90%) multiple times (for example, 3 times) within a set period (for example, 5 minutes).
- Configure TaurusDB for PostgreSQL to report alarms to Cloud Eye if its memory utilization reaches or exceeds a specific value (for example, 90%) multiple times (for example, 4 times) within a set period (for example, 5 minutes).
- Configure TaurusDB for PostgreSQL to report alarms to Cloud Eye if its storage utilization reaches or exceeds a specific value (for example, 85%) multiple times (for example, 5 times) within a set period (for example, 5 minutes).

NOTE

For details on Cloud Eye alarm configuration, see "Creating an Alarm Rule" in the *Cloud Eye User Guide*.

Measures:

- If a storage space usage alarm is reported, you can:
  - Check the storage space consumption to see if any space can be freed up by deleting data from DB instances or by dumping the data to another system.
  - Scale up the storage space.

10 Capacity Expansion and Specification Change

10.1 Are My TaurusDB for PostgreSQL Instances Still Available During Storage Scale-up?

Currently, you can scale up storage space of a TaurusDB for PostgreSQL instance.

When storage space is being scaled up, TaurusDB for PostgreSQL instances are still available and workloads are not affected. However, you cannot delete or reboot instances that are being scaled.

10.2 Why Does My TaurusDB for PostgreSQL Instance Become Faulty After Its Database Port Is Changed?

Symptom

- The DB instance is in **Faulty** state after the original database port is changed.
- The DB instance cannot be connected using the new database port.

Possible Causes

The submitted database port is occupied.

Procedure

Change the database port to the new port again.

- If the database port is changed successfully, the previous change failed because the submitted database port was occupied.
- If the original database port still fails to be changed, contact technical support.

10.3 Can I Change the VPC or Subnet that My TaurusDB for PostgreSQL Instance Belongs To?

No, you cannot directly change the VPC or subnet on the TaurusDB for PostgreSQL console.

However, you can change the VPC or subnet by restoring a full backup to a new DB instance.

11 Database Parameter Modification

11.1 Can I Use SQL Commands to Modify Global Parameters of My TaurusDB for PostgreSQL Instance?

Sorry, you cannot use SQL commands to modify global parameters, but you can modify specific parameters on the TaurusDB for PostgreSQL console.

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

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

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

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

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

Step 6 Change the value of the target parameter and click **Save**.

Step 7 In the displayed dialog box, click **OK**.

----End

11.2 How Do I Change the Time Zone of My TaurusDB for PostgreSQL Instance?

TaurusDB for PostgreSQL allows you to select a time zone when you create an instance and change the time zone after the instance is created.

NOTICE

After the time zone parameter is modified, you need to reconnect to the instance for the modification to take effect.

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

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

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

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

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

Step 6 Search for a time zone parameter in the search box, for example, **timezone**.

Step 7 Select a time zone, and click **Save**.

Step 8 In the displayed dialog box, click **OK**.

For example, to change the time zone to UTC+08:00, select **Asia/Shanghai** from the drop-down list.

----End

Time Zone Parameters

- **system_time_zone**: operating system (OS) time zone. This parameter cannot be changed and it has no impact on the database time zone.
- **timezone**: database time zone. You can modify this parameter to change the time zone for your instance.

11.3 What Inappropriate Parameter Settings Will Cause Unavailability of My TaurusDB for PostgreSQL Instance?

In the following cases, inappropriate parameter settings cause the database to be unavailable:

- Parameter value ranges are related to instance specifications.

  The maximum values of **shared_buffers** and **max_connections** are related to the instance physical memory. If you set these parameters inappropriately, the database will be unavailable.
- Parameter association is incorrect.

  max_connections, **autovacuum_max_workers**, and **max_worker_processes** must meet the following requirements. Otherwise, the database is unavailable.

  max_connections value + **autovacuum_max_workers** value + **max_worker_processes** value + 1 < 8388607

NOTE

For additional details, visit the **PostgreSQL official website**.

Solution:

1. Log in to the TaurusDB for PostgreSQL console and query the logs to locate the incorrectly configured parameters.

2. On the **Configuration** page, change parameters to default values and reboot the database.
3. Configure the incorrect parameter values and restore other parameters to their original default values.

11.4 How Do I Set the Upper Limit for the Storage Space Occupied by Temporary Files of My TaurusDB for PostgreSQL Instance?

Parameter Description

The **temp_file_limit** parameter specifies the maximum amount of storage space that a PostgreSQL process can use for temporary files.

When SQL statements are executed, temporary files, such as sort files and hash files, are generated. Any transaction using temporary files whose total size exceeds this limit will be terminated. For details, see **temp_file_limit**.

Parameter Modification

You can modify **temp_file_limit** of your TaurusDB for PostgreSQL instance on the console.

Precautions

Executing SQL statements or backing up or restoring data will generate temporary files. If the total size of temporary files exceeds the value of **temp_file_limit**, the operation fails. The value of **temp_file_limit** should not be too small.

temp_file_limit can be set to **-1**, indicating that the size of temporary files is not limited. To prevent too many temporary files from using up the storage space and causing service unavailability, do not set **temp_file_limit** to **-1**. In most cases, keep the default value for **temp_file_limit**.

11.5 How Do I Configure the test_decoding Extension for My TaurusDB for PostgreSQL Instance?

PostgreSQL 10, PostgreSQL 11, and PostgreSQL 13 support test_decoding. For more information about test_decoding, see **test_decoding introduction**.

To use test_decoding, set **wal_level** to **logical**.

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

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

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

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

Step 5 In the navigation pane, choose **Parameters**. On the **Parameters** tab page, locate **wal_level** and change its value to **logical**.

Step 6 Click **Save**. In the displayed dialog box, click **Yes**.

----End

12 Network Security

12.1 How Can I Prevent Untrusted Source IP Addresses from Accessing TaurusDB for PostgreSQL?

To prevent your database password from being cracked, set a strong password and periodically change it.

12.2 What Are the Possible Causes for Data Corruption of a TaurusDB for PostgreSQL Instance?

- Data tampering

  Lots of security measures are provided to ensure that only authenticated users have permissions to perform operations on database table records. Database tables can be accessed only through specific database ports.

  Verifying package during primary/standby synchronization can prevent data tampering.
- DB instance servers may be powered off suddenly, causing database page corruption and database rebooting failures.

  If a primary DB instance becomes faulty, TaurusDB for PostgreSQL switches to the standby DB instance within 1 to 5 minutes to provide services for you. DB instances cannot be accessed during the failover. You need to configure automatic reconnections between applications and DB instances to ensure near-continuous availability.

12.3 After My TaurusDB for PostgreSQL Instance Is Deleted, Why Can't the Associated Security Group Be Deleted Immediately?

When creating an instance, you must select a security group. If no security group is available or created, TaurusDB for PostgreSQL allocates a security group to you by default.

After an instance is deleted, it is moved to the recycle bin and retained for seven days by default. To modify the retention period, configure a recycling policy.

The deleted instance is not removed from the security group immediately until the instance is deleted from the recycle bin. Before deleting a security group, ensure that the security group is not associated with any instance. For details about how to query instances associated with a security group, see **How Do I Know the Instances Associated with a Security Group?**

