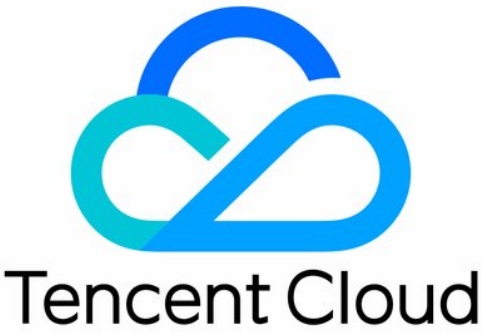
TencentDB for PostgreSQL

Kernel Version Introduction

Product Documentation



Contents

Kernel Version Introduction
Kernel Version Overview

Last updated：2024-04-03 16:44:55

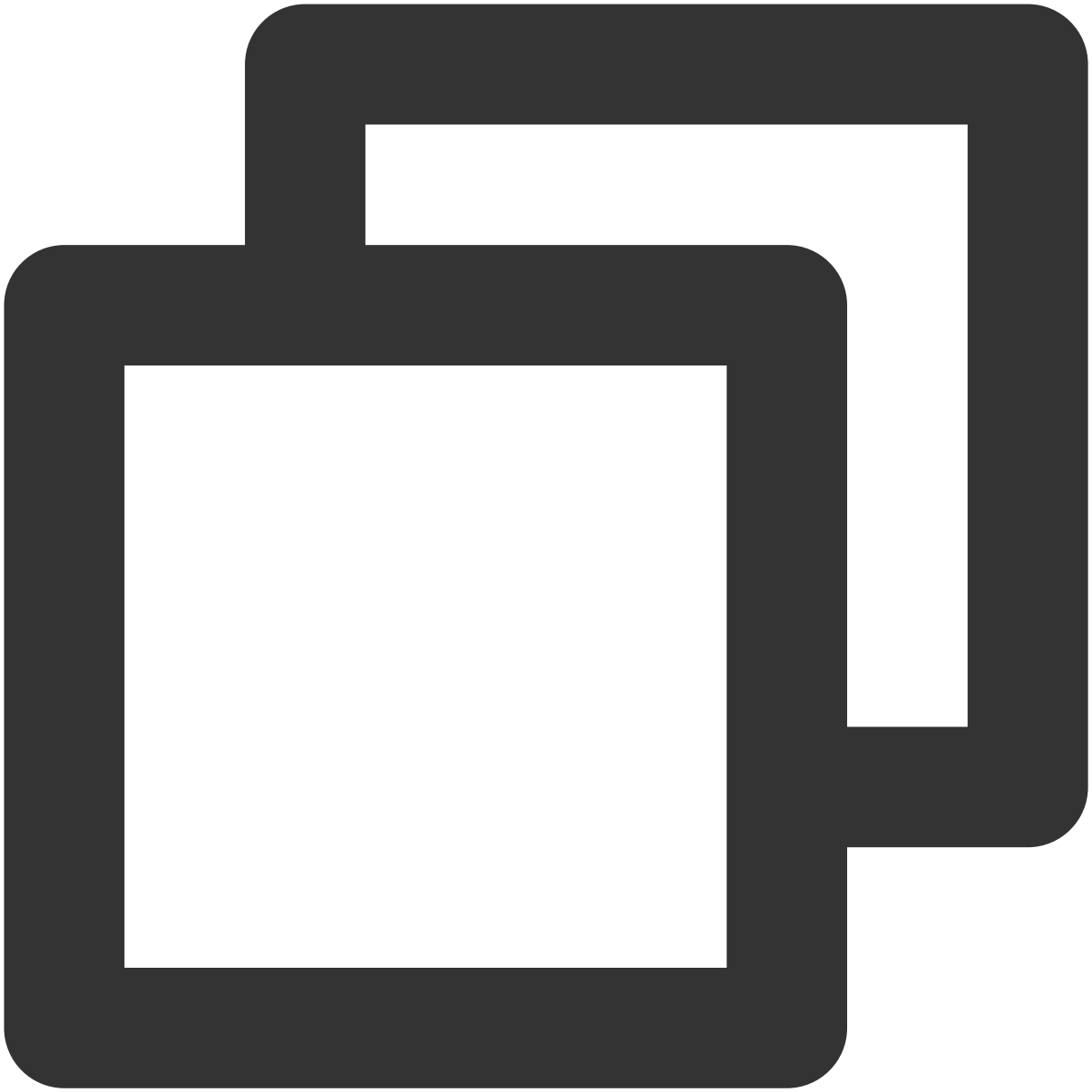
This document describes the background and strengths of TencentDB for PostgreSQL kernel versions.

Overview

PostgreSQL (PG) is a globally popular enterprise-grade open-source database. It won the DB-Engines DBMS of the Year awards for two consecutive years in 2017–2018 and again in 2020. It also won the O'Reilly Open Source Convention (OSCON) Lifetime Achievement Award in 2019.
Built on top of the community version of PostgreSQL, TencentDB for PostgreSQL offers specific features and optimized kernels based on its many years of business expertise. It not only optimizes the PostgreSQL database engine, permission management, and replication performance, but also improves the database usability and maintainability in the cloud.

Kernel Version Number

The kernel version number consists of two parts: **community version number** and **internal version number**.



```
v{community major version number.community minor version number}_r{incompatible ver
Example: v12.7_r1.1
```

v stands for version, and r stands for release; for example, v12.7_r1.1 corresponds to the community version 12.7, where r1.1 indicates Tencent Cloud's iteration version composed of `r{incompatible version number}.{compatible version number}`.

Note：

Version: It is the PostgreSQL community version number composed of two numbers: major version and minor version. The former starts from 10 (9.3, 9.4, 9.5, 9.6, and so on), and the latter starts from 1. When a community minor version

is released, Tencent Cloud will regularly follow the community to update the version.
Release: It is the version number used for TencentDB for PostgreSQL iteration and represents the modifications made by TencentDB for PostgreSQL. It consists of two numbers: incompatible version (starting from 1) and compatible version (starting from 0).
Incompatible version number
Definition of "incompatible": When a new database kernel program is used to start a database instance, if the start fails or some features cannot be used normally, the modification is incompatible.
When the kernel contains incompatible features, the incompatible version number needs to be increased by 1. It will be used to determine whether the minor version upgrade can be performed directly.
Compatible version number
When a compatible modification is added, the compatible version number needs to be increased by 1.

Supported Versions

PostgreSQL 10
PostgreSQL 11
PostgreSQL 12
PostgreSQL 13
PostgreSQL 14
PostgreSQL 15
PostgreSQL 16
You can view the database version of TencentDB for PostgreSQL in the instance list in the console. Different version numbers represent different compatible community versions; for example, TencentDB for PostgreSQL 12 is compatible with all subversions of open-source PostgreSQL 12.x, 12.1, 12.2, 12.3, etc.For information on version lifecycles, please see Major version lifecycle documentation.

Version Rules

Version and release together form a complete version called database kernel version.
A major version corresponds to a community evolution branch iterated separately, such as 10, 11, and 12, who have their own respective version number sequence.
When the community announces that a major version will no longer be maintained, the version will become unavailable for purchase. However, Tencent Cloud will continue to support database instances running on this version until they are actively terminated or migrated.
The minor version follows the changes in the community, while the release changes according to the feature changes made by the kernel of TencentDB.

Advantages

Compared with open-source PostgreSQL, TencentDB for PostgreSQL has the following strengths:

Greater stability

On the basis of open-source PostgreSQL, TencentDB for PostgreSQL further optimizes some underlying implementations of the database service, substantially reducing the problem of database crashes due to external factors and improving the database stability.

Enhanced primary-standby sync performance

The primary-standby sync performance of open-source PostgreSQL is severely compromised when a high number of DDL statements are processed. TencentDB for PostgreSQL is greatly optimized in this regard, with an overall improvement of up to 30,000 times.

Shorter access latency

TencentDB for PostgreSQL features special optimization for connection establishment in scenarios with a large number of non-persistent connections. This reduces the database connection performance loss by 80% and greatly improves the non-persistent connection performance.

Kernel Version Release Notes

Last updated：2024-04-03 17:54:52

This document describes the TencentDB for PostgreSQL kernel version updates. For the kernel minor version upgrade, please refer to kernel minor version upgrade; for the major version upgrade, consult upgrading the major version of the database.

PostgreSQL 16

V16.0_r1.2

Rectified the memory leak caused by the long connection of audit logs.
Resolved the core caused by the illegal parameter of dblink.
For security reasons, ordinary logs record the COPY statement;
The tencentdb_superuser supports the permission of creating replication origin.
Fixed the timeout error issue when cos_fdw scans large partition tables.

V16.0_r1.1

Rectified CVE-2023-5869.

V16.0_r1.0

Compatible with Community Version 16.0.

PostgreSQL 15

V15.1_r1.9

Rectified the memory leakage caused by the long connection of audit logs;
Resolved the core caused by the illegal parameter of dblink;
For security reasons, ordinary logs record the COPY statement;
The tencentdb_superuser supports the permission of creating replication origin.
Fixed the timeout error issue when cos_fdw scans large partition tables.

V15.1_r1.8

Rectified CVE-2023-5869.

V15.1_r1.7

Optimized the audit log performance.
Resolved the issue where slave initiation from the slave machine fails when the failover slot is slower than the slave machine's checkpoint position.

V15.1_r1.6

Supported major version upgrades.
Updated certain extension versions.

V15.1_r1.5

Compatible with community version 15.1.
Rectified CVE-2023-39417 security vulnerability.
Supported tencentdb_system_stat process monitoring extension.
Supported pg_squeeze and pg_similarity extensions.

V15.1_r1.4

Compatible with community version 15.1.
Supported dblink for cross-instance access.
The rds_server_handler extension allows command-based cache clearance.
Supported comprehensive database auditing.
Prohibited COPY PROGRAM actions.
Supported the tencentdb_superuser extension.

V15.1_r1.3

Compatible with community version 15.1.
Supported the pgvector extension.

V15.1_r1.2

Compatible with Community Version 15.1.
Strengthened security measures to address potential vulnerabilities during the creation and upgrade of certain extensions.

V15.1_r1.0

Compatible with Community Version 15.1.
Introduced the tencentdb_superuser permission, allowing the creation of event_trigger.

PostgreSQL 14

V14.2_r1.18

Rectified the memory leakage caused by the long connection of audit logs.
Resolved the core caused by the illegal parameter of dblink;
For security reasons, ordinary logs record the COPY statement;
The tencentdb_superuser supports the permission of creating replication origin.
Fixed the timeout error issue when cos_fdw performs a large partition table scan.
Community bug fix: Data corruption caused by timeline switchback after promote is executed on the slave machine.

V14.2_r1.16

Rectified CVE-2023-5869.

V14.2_r1.15

Optimized the audit log performance.

V14.2_r1.14

Supported major version upgrades.
Updated certain extension versions.

V14.2_r1.12

Compatible with Community Version 14.2.
Rectified CVE-2023-39417 security vulnerability.
Supported tencentdb_system_stat process monitoring extension.
Supported the pg_squeeze and pg_similarity extensions.

V14.2_r1.12

Rectified CVE-2023-39417 security vulnerability.
Supported the tencentdb_system_stat process monitoring extension.++
Supported the pg_squeeze and pg_similarity extensions.

V14.2_r1.11

Supported cross-instance access via dblink.
The rds_server_handler extension allows for command-based cache clearance;
Supported audit logs.
Prohibited COPY PROGRAM actions.
Supported the pgvector extension.
Supported the tencentdb_superuser extension.

V14.2_r1.9

Rectified permission escalation vulnerability during extension creation: The tencentdb_superuser no longer escalates to superuser during extension creation.

V14.2_r1.7

Added GUC to regulate user COPY actions, prohibiting the use of COPY actions during extension creation.

V14.2_r1.4

Supported failover slot.
tencentdb_superuser supports the creation/modification of event_trigger.
Table oid information is returned when parsing drop schema fails during logical replication.
The parameter level of session_replication_role has been changed to PGC_USERSET.
Fixed the issue of cross-database access permissions within the same instance of dblink.

V14.2_r1.3

Supported the tencentdb_pwdcheck strong password authentication extension.

V14.2_r1.2

Compatible with Community Version 14.2.
Supported the dblink extension.

v14.2_r1.1

Compatible with Community Version 14.2.
Supported the cos_fdw extension.

v14.2_r1.0

Compatible with Community Version 14.2.
Integrated with the ability to modify TencentDB for PostgreSQL kernel.
Supported the pg_cron extension.
Supported the TopN extension.
Supported TDE.

PostgreSQL 14 babelfish edition

v14.3_r1.19

Community bug fix: Data corruption caused by timeline switchback after promote is executed on the slave machine.

v14.3_r1.12

Rectified the CVE-2023-39417 security vulnerability.

v14.3_r1.10

Rectified permission escalation vulnerability during extension creation: The tencentdb_superuser no longer escalates to superuser during extension creation.

v14.3_r1.8

Added GUC to regulate user COPY actions, prohibiting the use of COPY actions during extension creation.

v14.3_r1.6

Supported the conversion of NUL to custom characters.

v14.3_r1.3

Supported the merging and filtering of files in pg_basebackup dual stream.
Supported tencentdb_superuser.
Supported soft_limit_connection for soft link restrictions.
Supported the QPS statistical extension.
Prohibited the superuser from executing functions owned by ordinary users.
Supported third-party extensions.
Disabled dblink.
Fixed the bug of slave asynchronous invalid buffer.
Supported the topn and pg_cron extensions;
When tencentdb_superuser CREATE/ALTER superuser is used, the operation of CREATE/ALTER a tencentdb_superuser will be performed by default.
Supported TDE.
Supported file (folder) filtering during server-side basebackup.
Supported cos_fdw cold and hot storage separation extension.
Fixed the security issue of pg_cron.
Supported dblink cross-database access within the same instance.
Allowed the superuser to execute all functions with oid less than 16384.
Fixed the issue of data loss in community CIC/RIC index.
Supported the initial kernel and 2.1 version extension code of babelfish.

PostgreSQL 13

V13.3_r1.14

Rectified the memory leak caused by the long connection of audit logs.

Resolved the core caused by the illegal parameter of dblink.
For security reasons, ordinary logs record the COPY statement;
The tencentdb_superuser supports the permission of creating replication origin.
Fixed the timeout error issue when cos_fdw scans large partition tables.
Community bug fix: Data corruption caused by timeline switchback after promote is executed on the slave machine.

V13.3_r1.13

Rectified CVE-2023-5869.

V13.3_r1.12

Optimized the audit log performance.

V13.3_r1.11

Supported major version upgrades.
Updated certain extension versions.

V13.3_r1.10

Compatible with Community Version 13.3.
Rectified CVE-2023-39417 security vulnerability.
Supported the tencentdb_system_stat process monitoring extension.
Support the pg_squeeze, pg_similarity extensions.

V13.3_r1.9

Compatible with Community Version 13.3.
Supported dblink for cross-instance access.
The rds_server_handler extension allows command-based cache clearance.
Supported comprehensive database auditing.
Prohibited COPY PROGRAM actions.
Supported the tencentdb_superuser extension.

V13.3_r1.8

Compatible with community version 13.3.
Strengthened security measures to address potential issues that may arise during the creation and upgrading of certain extensions.

V13.3_r1.6

Compatible with community version 13.3.
Added the tencentdb_failover_slot extension, supporting the failover slot feature.

Added the tencentdb_superuser permission, allowing the use of the checkpoint command.
Added the tencentdb_superuser permissions, allowing the creation of event_trigger.++
Supported cross-database access within the same instance using dblink.
Optimized the performance of walsender parsing the drop publication statement.

V13.3_r1.5

Compatible with Community Version 13.3.
Supported the tencentdb_pwdcheck strong password authentication extension.

V13.3_r1.4

Compatible with community version 13.3.
Supported the db_link extension.

v13.3_r1.3

Compatible with Community Version 13.3.
Supported the cos_fdw extension.

v13.3_r1.2

Compatible with Community Version 13.3.
Supported the pg_cron extension.
Supported the TopN extension.
Supported TDE.

v13.3_r1.1

Compatible with Community Version 13.3.
Fixed some extension management bugs.

v13.3_r1.0

Compatible with Community Version 13.3.
Integrated all the features of open-source PostgreSQL 13.3.
Added the admin role `pg_tencentdb_superuser` .
Integrated Tencent Cloud's historical optimization feature packages for PostgreSQL.

PostgreSQL 12

V12.18_r1.19

Upgraded the community minor version to 12.18.

V12.7_r1.18

Rectified the memory leak caused by the long connection of audit logs.

Fixed the core caused by the illegal parameter of dblink.

For security reasons, ordinary logs record the COPY statement;

The tencentdb_superuser supports the permission of creating replication origin.

Resolved the timeout error issue when cos_fdw scans large partition tables.

Community bug fix: Data corruption caused by timeline switchback after promote is executed on the slave machine.

V12.7_r1.17

Supported the orafce extension.

V12.7_r1.16

Rectified CVE-2023-5869.

V12.7_r1.15

Optimized the audit log performance.

V12.7_r1.14

Supported major version upgrades.

Updated certain extension versions.

V12.7_r1.13

Compatible with community version 12.7.

Rectified CVE-2023-39417 security vulnerability.

Supported tencentdb_system_stat process monitoring extension.

Supported the pg_squeeze and pg_similarity extensions.

V12.7_r1.12

Compatible with community version 12.7.

Supported the pg_squeeze estension in version 1.5.2.

Upgraded the zhparser extension to version 2.2.

V12.7_r1.10

Compatible with community version 12.7.

Enhanced security to address potential issues that may arise during the creation and upgrade of certain extensions.

V12.7_r1.8

Compatible with community version 12.7.

Added the tencentdb_failover_slot extension, enabling the failover slot feature.
Added the tencentdb_superuser permission, allowing the use of the checkpoint command.
Added the tencentdb_superuser permission, allowing the creation of event_trigger.
Supported cross-database access within the same instance using dblink.
Optimized the performance of walsender in parsing drop publication statements.

V12.7_r1.7

Compatible with Community Version 12.7.
Supported the tencentdb_pwdcheck strong password authentication extension.

V12.7_r1.6

Compatible with community version 12.7.
Supported the timescaledb extension in versions 1.7.5 and 2.6.0.

v12.7_r1.5

Compatible with community version 12.7.
Supported the dblink extension.

v12.7_r1.3

Compatible with community version 12.7.
Supported the cos_fdw extension.

v12.7_r1.2

Compatible with community version 12.7.
Supported the pg_cron extension.
Supported the TopN extension.
Supported TDE.

v12.7_r1.1

Compatible with community version 12.7.
Integrated all the features of open-source PostgreSQL 12.7.

v12.4_r1.0

Compatible with community version 12.7.
Added the admin role `pg_tencentdb_superuser` .
Integrated Tencent Cloud's historical optimization feature packages for PostgreSQL.

PostgreSQL 11

V11.12_r1.15

Community bug fix: Data corruption caused by timeline reversion after promote is executed on the slave machine.

V11.12_r1.14

Optimized the Audit Logs Performance.

V11.12_r1.13

Supported major version upgrades.

Updated certain extension versions.

V11.12_r1.12

Compatible with community version 11.12.

Resolved CVE-2023-39417 security vulnerability.

Supported the tencentdb_system_stat process monitoring extension.

Supported the pg_squeeze and pg_similarity extensions.

V11.12_r1.11

Compatible with community version 11.12.

Supported dblink for cross-instance access.

The rds_server_handler extension allows command-based cache clearance.

Supported comprehensive database auditing.

Prohibited COPY PROGRAM actions.

Supported the tencentdb_superuser extension.

v11.12_r1.9

Compatible with community version 11.12.

Enhanced security to address potential issues that may arise during the creation and upgrade of certain extensions.

V11.12_r1.7

Compatible with community version 11.12.

Added the tencentdb_failover_slot extension, enabling the failover slot feature.

Added the tencentdb_superuser permission, allowing the use of the checkpoint command.

Added the tencentdb_superuser permissions, allowing the creation of event_trigger.++

Supported cross-database access within the same instance using dblink.

Optimized the performance of walsender in parsing drop publication statements.

V11.12_r1.6

Compatible with Community Version 11.12.

Supported the tencentdb_pwdcheck strong password authentication extension.

V11.12_r1.5

Compatible with Community Version 11.12.

Supported the dblink extension.

v11.12_r1.3

Compatible with community version 11.12.

Supported the cos_fdw extension.

v11.12_r1.2

Compatible with community version 11.12.

Supported the pg_cron extension.

Supported the TopN extension.

Supported TDE.

v11.12_r1.1

Compatible with community version 11.12.

v11.8_r1.0

Compatible with community version 11.8.

Integrated all the features of open-source PostgreSQL 11.8.

Added the admin role `pg_tencentdb_superuser` .

Integrated Tencent Cloud's historical optimization feature packages for PostgreSQL.

PostgreSQL 10

V10.17_r1.13

Community bug fix: Data corruption caused by timeline switchback after promote is executed on the slave machine.

V10.17_r1.12

Compatible with Community Version 10.17.

Resolved CVE-2023-39417 security vulnerability.

V10.17_r1.11

Compatible with community version 10.17.

Enhanced against potential security issues that may arise during the creation and upgrade of certain extensions.

V10.17_r1.8

Compatible with community version 10.17.

Added the tencentdb_failover_slot extension, supporting the failover slot feature.

Added the tencentdb_superuser permission, allowing the use of the checkpoint command.

Added the tencentdb_superuser permissions, allowing the creation of event_trigger.++

Optimized the performance of walsender in parsing drop publication statements.

V10.17_r1.7

Compatible with Community Version 10.17.

Supported the tencentdb_pwdcheck strong password authentication extension.

V10.17_r1.6

Compatible with community version 10.17.

Resolved the issue of tencentdb_superuser using dblink.

V10.17_r1.5

Compatible with Community Version 10.17.

Supported the dblink extension.

v10.17_r1.3

Compatible with Community Version 10.17.

Supported the cos_fdw extension.

v10.17_r1.2

Compatible with Community Version 10.17.

Supported the pg_cron extension.

Supported the TopN extension.

Supported TDE.

v10.17_r1.1

Compatible with Community Version 10.17.

v10.4_r1.1

Compatible with Community Version 10.4.

Fixed some extension management bugs.

v10.4_r1.0

Compatible with Community Version 10.4.

Added the admin role `pg_tencentdb_superuser` .

Integrated Tencent Cloud's historical optimization feature packages for PostgreSQL.

Viewing Kernel Version

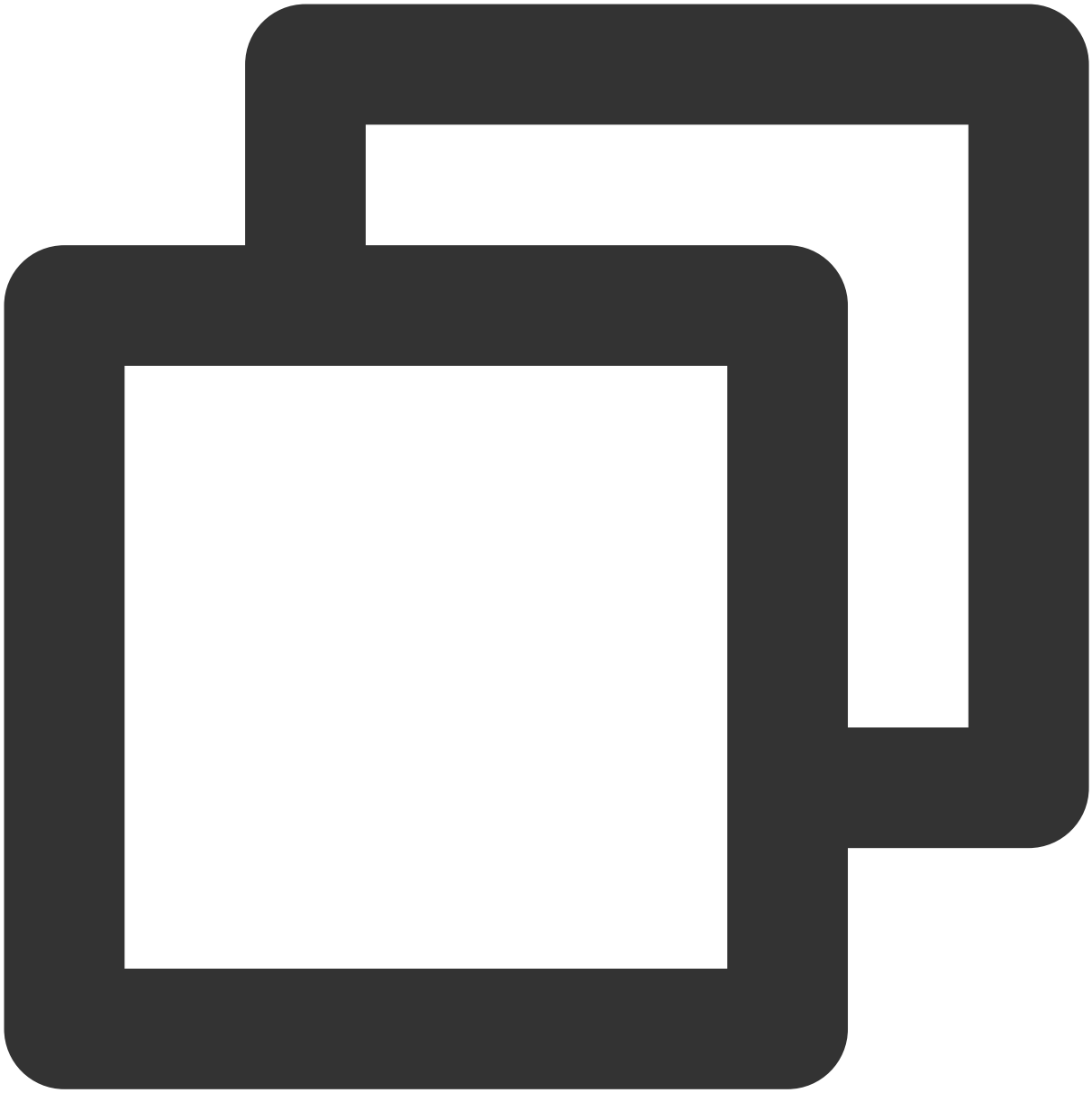
Last updated：2024-01-24 11:08:34

This document describes how to view the kernel version information of TencentDB for PostgreSQL.

Directions

In TencentDB for PostgreSQL instance

1. You can view the kernel version by connecting to the TencentDB for PostgreSQL instance through CVM. For more information on how to connect and log in to the instance, see Connecting to PostgreSQL Instances.
2. After login, run the following command to view the kernel version.



```
show tencentdb_version;
```

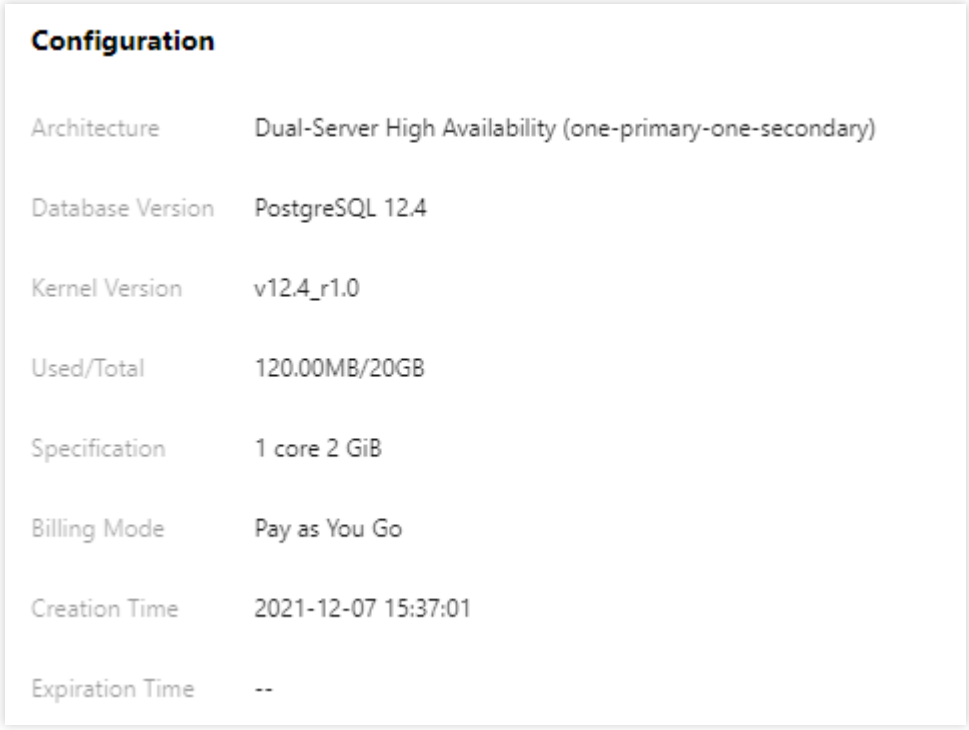
Note:

For some existing instances, if the command fails or the returned result is v1, the release version is r1.0. If the database version is PostgreSQL 10.4, then the kernel version is v10.4_r1.0.

In TencentDB for PostgreSQL console

1. Log in to the TencentDB for PostgreSQL console. In the instance list, click an instance ID or **Manage** in the **Operation** column to access the instance management page.

2. On the instance details page, view the kernel version under **Configuration**.



Proprietary Kernel Features
Logical Replication Slot Failover

Last updated：2024-01-24 11:08:34

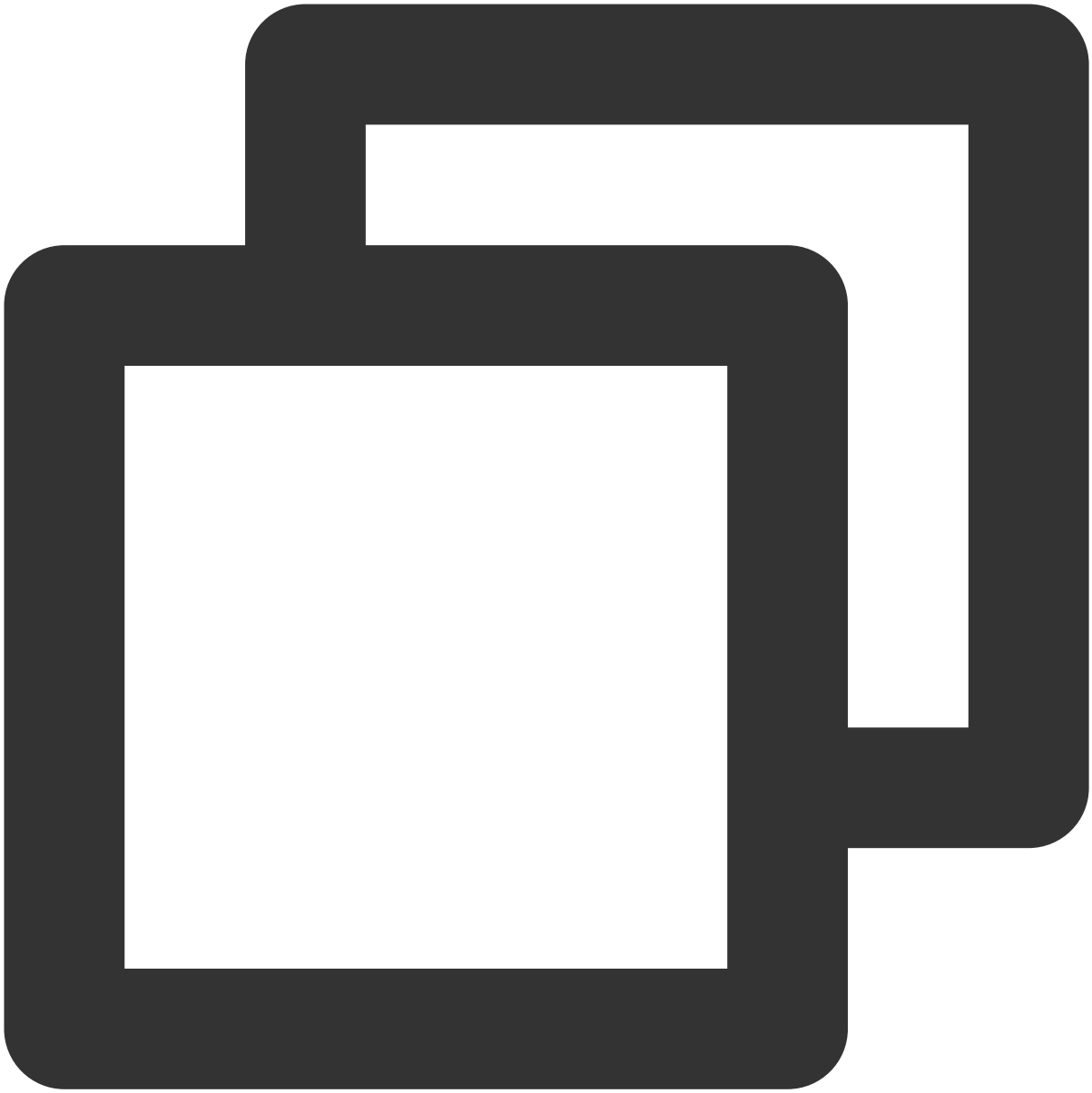
Background

Currently, you will create a logical replication slot when using logical replication in the TencentDB for PostgreSQL. As this information is not recorded in the catalog, it will not be synchronized from the primary database to the standby database. If an HA switchover occurs, the logical replication slot will be lost, causing the logical replication to be disconnected and causing inconvenience to the business. Therefore, the logical replication slot need to be failed over.

Feature Overview

The failover slot feature can synchronize the slot information from the primary database to the standby database, so that the disconnection of the logical subscription can be avoided after the HA switchover, and the user can continue to use the logical subscription without disruption. TencentDB for PostgreSQL provides `tencent_failover_slot`, a plugin for users to create failover slots, convert them to non-failover slots, view failover slot information, and delete failover slots. At the same time, GUC parameters are added to help you configure for abnormal situations. The details are as follows.

Creating a failover slot



```
create extension tencentdb_failover_slot;--Create a plugin first
select pg_create_logical_gailover_slot('slotname','pluginname');--Create a failover
```

Note:

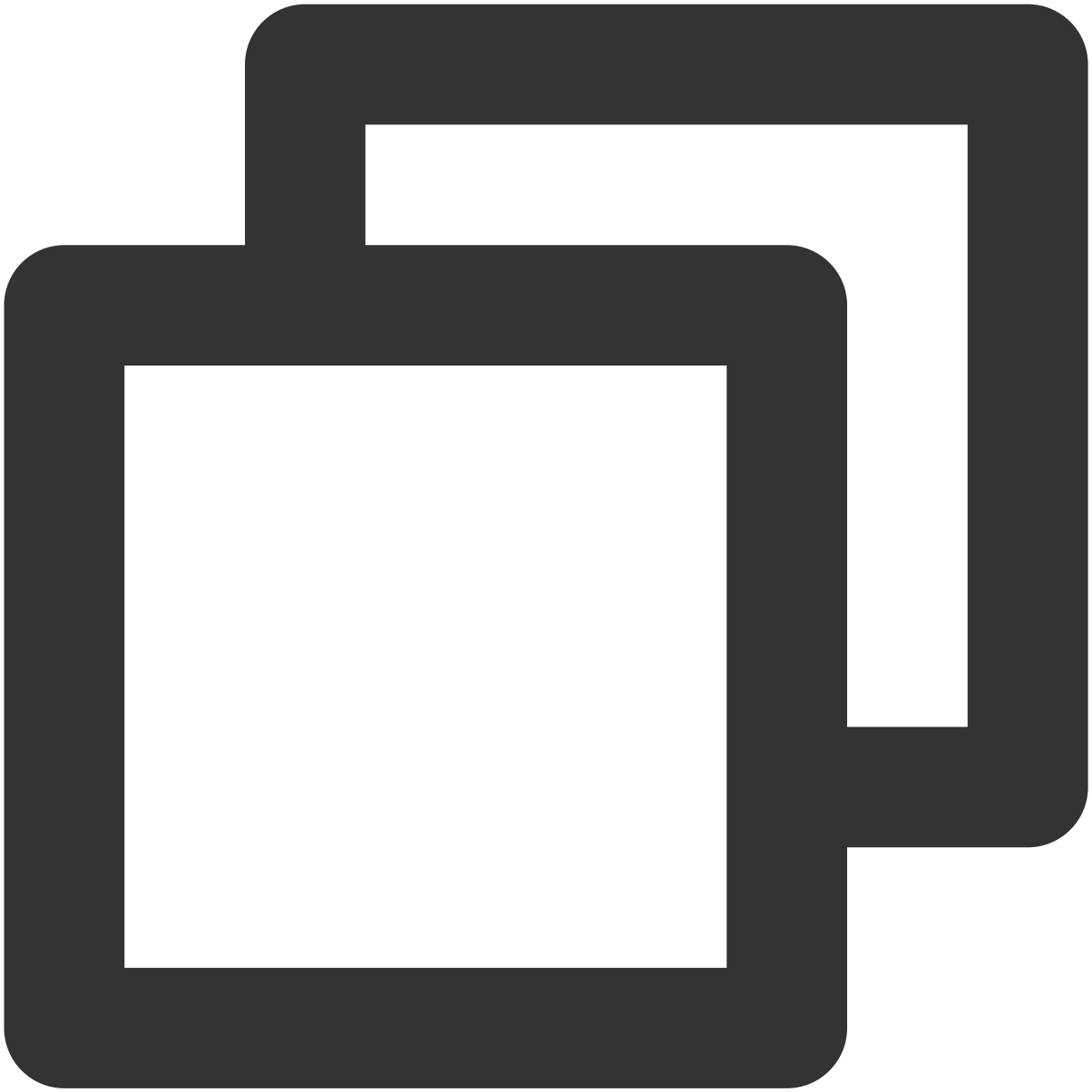
Slots are at the instance level, while plugins are at the database level. If you need to call functions in the plugins after switching the database, you need to recreate the `tencentdb_failover_slot` plugin.

The `pg_create_logical_failover_slot` function contains two parameters:

slotname: The name of the failover slot.

Pluginname: The name of the plugin.

Converting a failover slot to non-failover slot



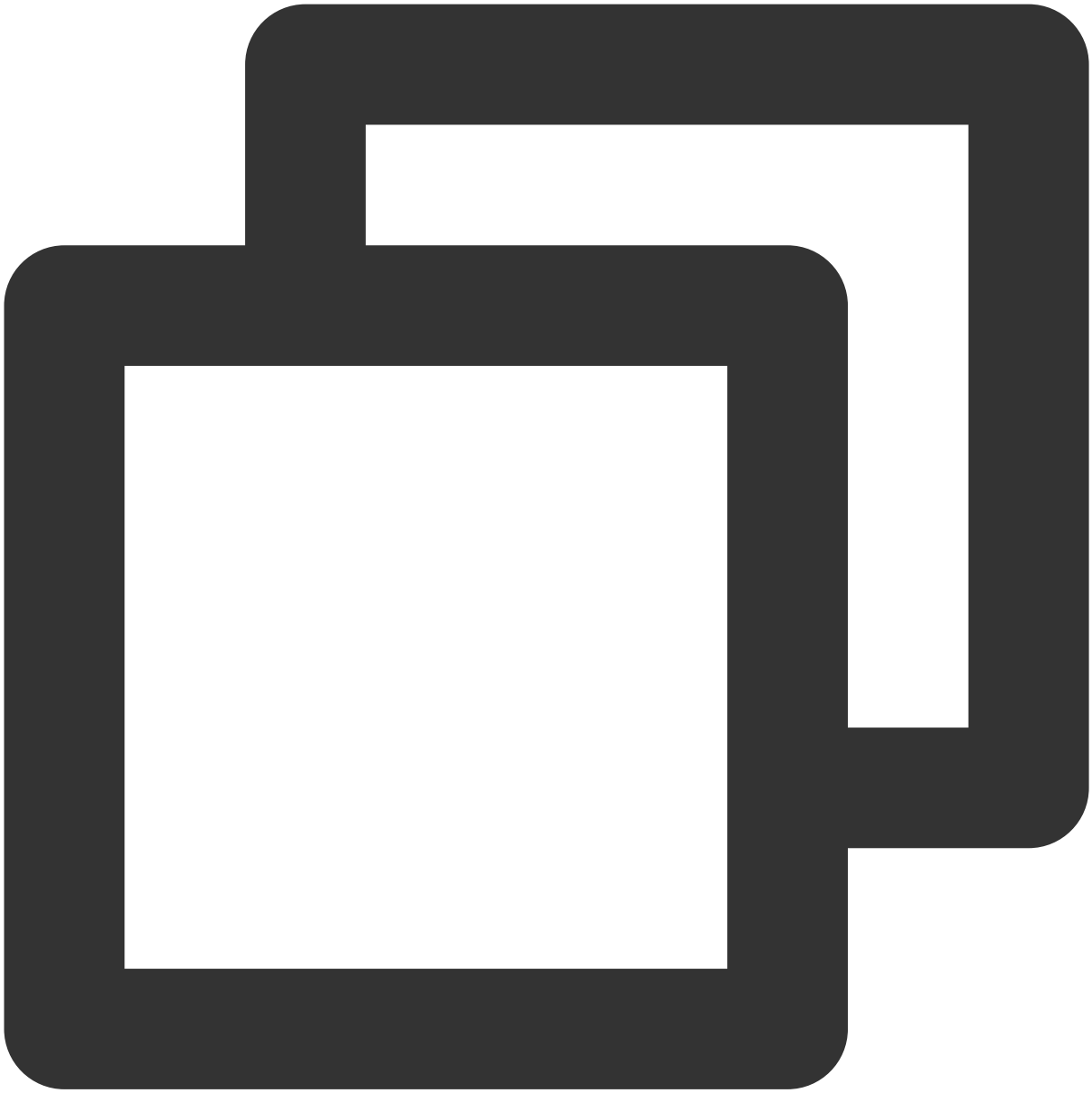
```
select transform_slot_to_nonfailover('slotname');
```

Note:

Slotname: The name of the slot.

Make sure that the slot is currently inactive, meaning that no publication or subscription is using it.

Viewing the information of a failover slot

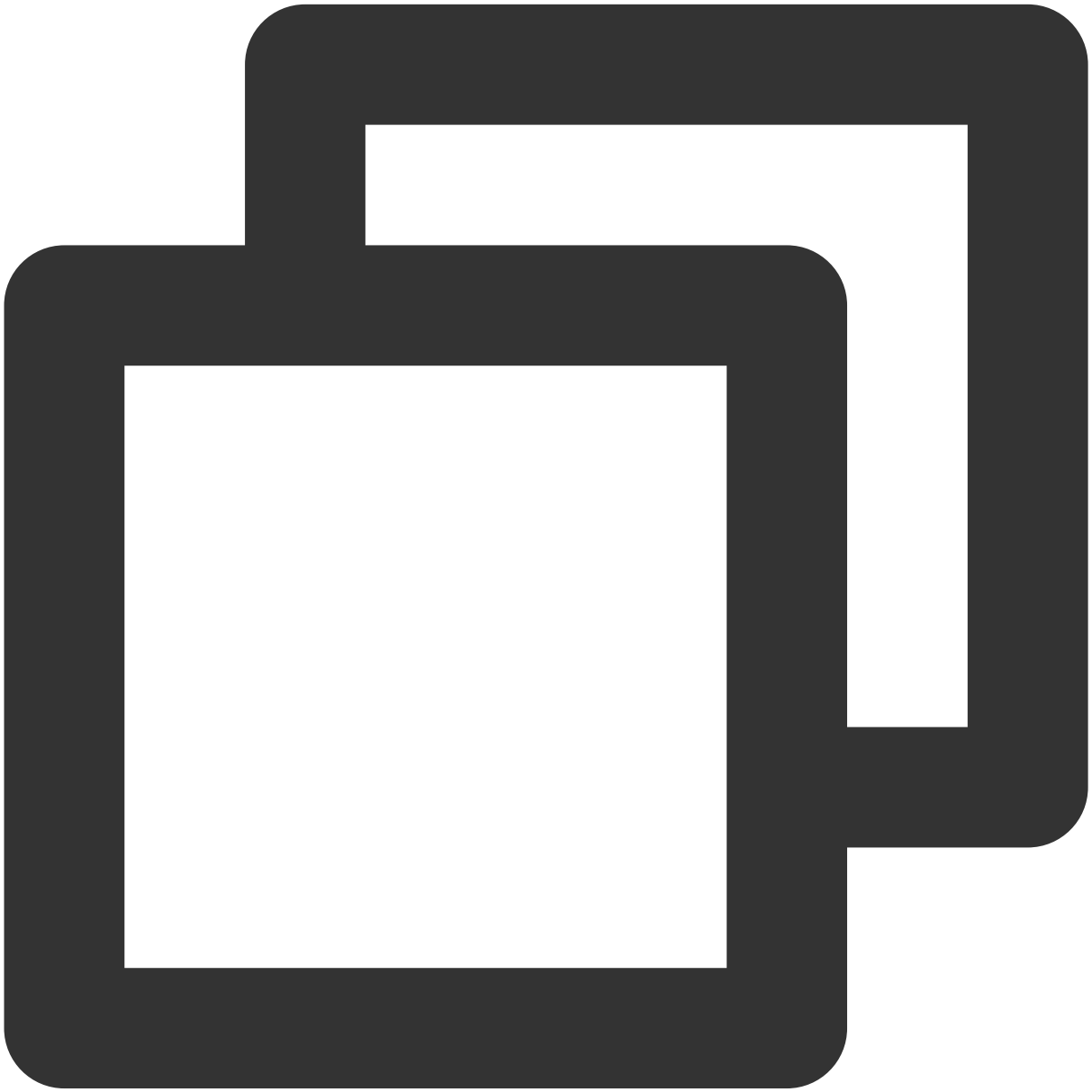


```
postgres=# select * from pg_failover_slots;
     slot_name
-----------------
     fs
     fsg
   (2 rows)
```

Note:

The `pg_failover_slots` view contains a column showing the names of all current failover slots. For more information about failover slots, go to the `pg_replication_slots` view.

Deleting a failover slot



```
select * from pg_drop_replication_slot('slotname');
```

Note:

Slotname: The name of the slot.

Parameter settings

TencentDB for PostgreSQL has added a new parameter `failover_slot_timeline_diverged_option` with the type of `enum`, which is convenient for you to configure for abnormal situations. You can log in to the TencentDB for PostgreSQL console, and configure this parameter on **Instance Details** > **Parameter Settings**. The default value of this parameter is `error`, and the optional values are `error` and `rewind`. In extreme cases, the log receiving speed of the standby database is slower than that of logical replication and the HA switch occurs. If `error` is set as the value, it means that logical replication will be paused, and both the publisher and subscriber can receive an error report and wait for you to process it. If the value is set to `rewind`, the logical replication will start from the time point when the switch is performed.

Note

Currently, only logical replication slot failover is supported, whereas physical replication slot failover is not.

TDE

Last updated：2024-01-24 11:08:34

Feature Background

As the requirements for data security protection get more and more strict, information security protection laws in many countries/regions stipulate that databases must encrypt the stored data to prevent data leakage caused by accidental loss of data files.

Feature Description

TencentDB for PostgreSQL comes with the transparent data encryption (TDE) feature. Transparent encryption means that the data encryption and decryption are transparent to users. TDE supports real-time I/O encryption and decryption of data files. It encrypts data before it is written to disk, and decrypts data when it is read into memory from disk, which meets the compliance requirements of static data encryption. The keys used for encryption are generated and managed by KMS.
KMS is a data and key security protection service of Tencent Cloud, where all involved processes use high-security communication protocols to guarantee high service security. In addition, it provides distributed cluster management and hot backup capabilities to ensure high service reliability and availability.
KMS uses a two-layer key system, which involves two types of keys: customer master key (CMK) and data encryption key (DEK). A CMK is used to encrypt small packet data (up to 4 KB in size), such as DEK, password, certificate, and configuration file. A DEK is used to encrypt massive amounts of business data in symmetric encryption method during storage or communication and is encrypted and protected in asymmetric encryption method with a CMK. In this way, data files can be encrypted.

Supported Versions

Kernel version: v10.17_r1.2、v11.12_r1.2、v12.7_r1.2、v13.3_r1.2、v14.2_r1.0.

Use Cases

TDE means that the data encryption and decryption are transparent to users. TDE supports real-time I/O encryption and decryption of data files. It encrypts data before it is written to disk, and decrypts data when it is read into memory from disk, which meets the compliance requirements of static data encryption.

Directions

For more information on how to enable TDE and encrypt a database with TDE, see Enabling TDE.

Making Buffer Invalid Asynchronously

Last updated：2024-01-24 11:08:34

Background

The primary-standby replication logic of PostgreSQL will severely slow down data recovery from the standby database if there are many DDL statements in tables. To avoid this, TencentDB for PostgreSQL is specially optimized and modified.

How It Works

PostgreSQL uses physical replication to implement primary-standby replication. After a log is synced to the standby node, the standby node will parse the WAL log to make the data consistent with that on the primary node. The standby node will perform the following operations when recovering a DROP statement:
1. Recover a system table, such as `pg_class`, `pg_attrbute`, and `pg_type`, which is equivalent to removing the table metadata.
2. Close the table file.
3. Traverse pages in the buffer and mark the found pages of the table as invalid, so subsequent processes can use them.
4. Send an async invalidation message to other backends to notify that the table has been deleted.
5. Delete the physical file of the table.
However, in the PostgreSQL kernel, the `DropRelFileNodesAllBuffers` function will be executed in step 3. It needs to traverse the entire `shard_buffer` to check whether the buffer has the data of the table to be deleted and mark corresponding pages as invalid. The default size of a page in PostgreSQL is 8 KB. If the size of `shard_buffer` is 16 GB for example, there will be 16 GB / 8 KB = 2 million pages. Therefore, every time a table is deleted, 2 million pages need to be checked one by one. If the table has indexes, 2 million pages will be checked one by one in each index.
From the perspective of business, when there are a high number of table changes and quick table deletions, as such operations can be executed concurrently on the primary database, the compromised performance is less obvious. However, as the PostgreSQL standby database recovers a table in a single thread, the primary-standby sync logs will be heaped, and data transfer will be delayed.

Solution

When TencentDB for PostgreSQL recovers a dropped table, it writes the table information into a shared hash table. If a table file is deleted, it won't be physically deleted directly; instead, TencentDB for PostgreSQL will store the actual file deletion operation as an async action in the hash table.

After invalid buffer processing is completed, the table will be removed from the hash table. In this way, if a file fails to be opened during the process, it will be sufficient to check whether the file exists in the hash table. In addition, if the queue is traversed when a file is created and a file with the same name in the hash table is undergoing invalid buffer processing, just wait.

PostgreSQL stores each table filename as a UInt32 integer and adopts the principle of "global assignment and local storage", where all databases in an instance use the same counter to generate file numbers, and generated files are stored in the databases' respective directories. When the system assigns a filename, if the current database has a file with the same name, the system will try the next name until there is no conflict. The counter will reset after it reaches the maximum value.

After optimization, the primary-standby sync performance is improved by over 30,000 times in similar scenarios.

Process Monitoring

Last updated：2024-03-20 10:32:43

Feature Background

In certain common scenarios, customer applications need not only the overall monitoring information of the database instance, but also need to use the process monitoring information in the instance to judge the business health, and then make business logic decisions. Typically, in SaaS scenarios, a single tenant of the user uses a database of the database instance alone. By monitoring the resource usage of the processes connected to the database, the business party can further judge the current health of the tenant and other situations.

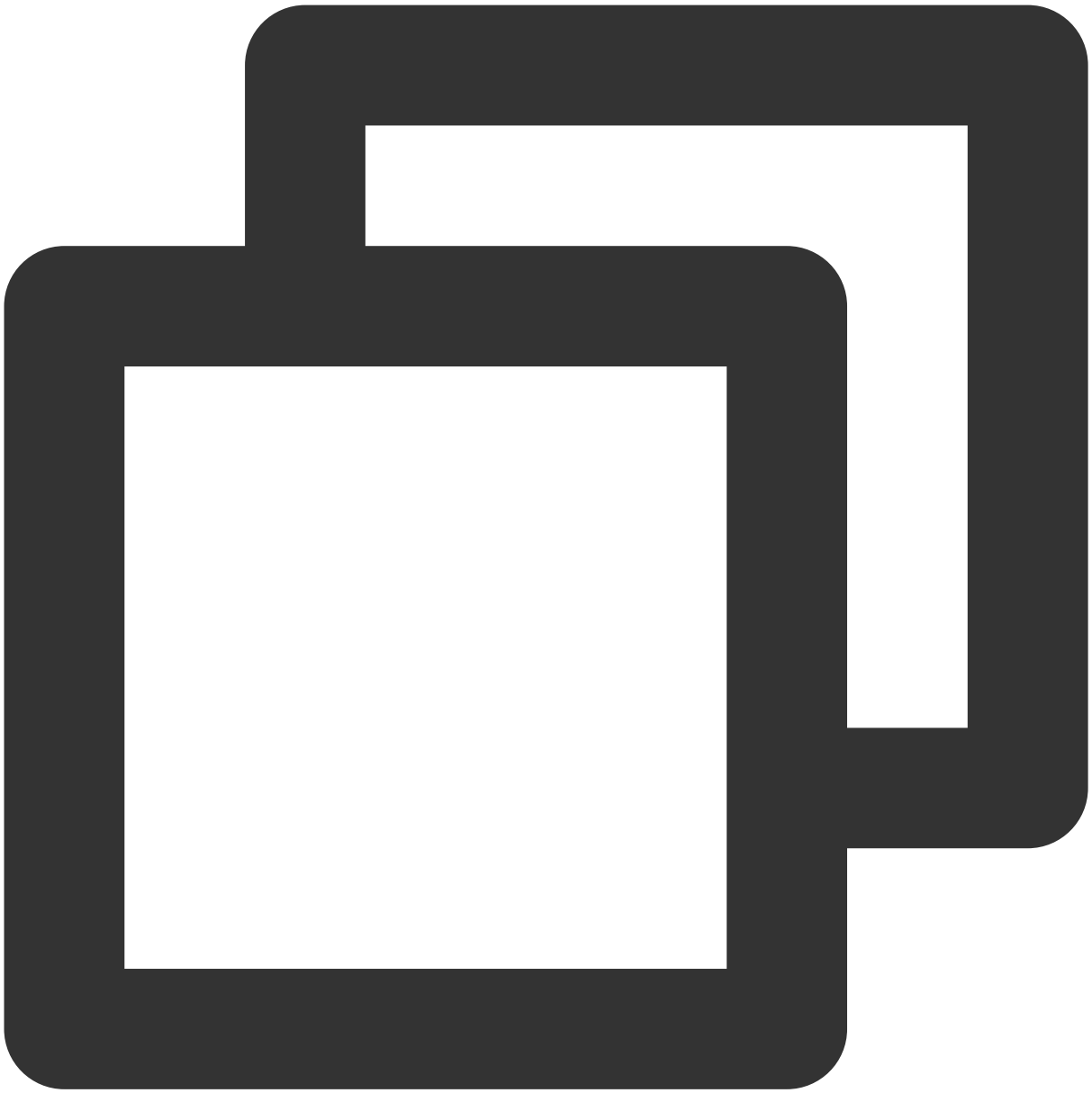
Description of the Feature

TencentDB for PostgreSQL provides process monitoring information for the database instance. The system provides a view named tencentdb_process_system_usage, which contains the following information, as explained below:

Field name	Description
pid	Process Number.
username	Username.
datname	Database Name.
backend_type	Process type. Possible types include autovacuum launcher, autovacuum worker, logical replication launcher, logical replication worker, parallel worker, background writer, client backend, checkpointer, startup, walreceiver, walsender, and walwriter. In addition, backend Workers registered by extensions may have additional types.
query	Refers to the query statement that is being executed currently.
cpu_usage	Refers to the CPU utilization during the current query sampling period. For the sampling interval, the parameter tencentdb_system_stat.sampling_interval is provided for settings. The unit of this parameter is milliseconds and the default value is 1000.
memory_bytes	Refers to the memory usage rate during the current sampling period, in bytes.

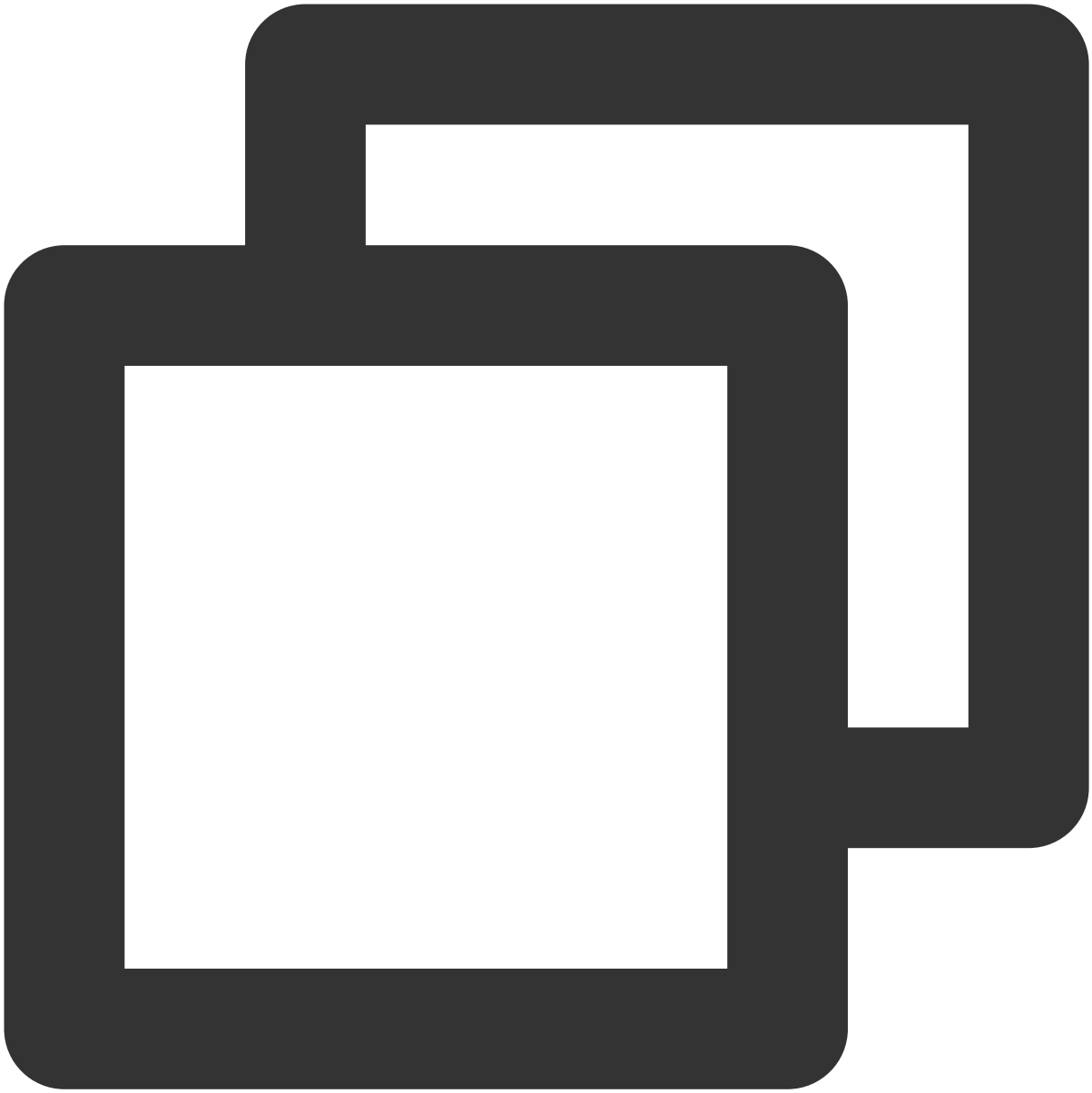
Use Instructions

1. Create a new extension with the following command:



```
postgres => create extension tencentdb_system_stat;
CREATE EXTENSION
postgres =>
```

2. The usage of the tencentdb_process_system_usage view is the same as other views. You only need to log in to the PostgreSQL database instance and perform a query. Here is an example statement:



```
select * from tencentdb_process_system_usage;
```