

**GaussDB**

# Best Practices

**Issue**            01  
**Date**             2025-01-22



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2025. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI 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 Suggestions on GaussDB Security Configuration..... 1**

# 1 Suggestions on GaussDB Security Configuration

---

Security is a shared responsibility between Huawei Cloud and you. Huawei Cloud ensures the security of cloud services for a secure cloud. As a tenant, you should utilize the security capabilities provided by cloud services to protect data and use the cloud securely. For details, see [Shared Responsibilities](#).

This section provides actionable guidance for enhancing the overall security of using GaussDB. You can continuously evaluate the security status of your GaussDB and enhance their overall security defense by combining different security capabilities provided by GaussDB. By doing this, data stored in GaussDB can be protected from leakage and tampering both at rest and in transit.

You can make security configurations from the following dimensions to match your workloads.

- [Maximum Number of Connections](#)
- [Security Authentication](#)
- [User Password Security](#)
- [Permissions Management](#)
- [Database Audit](#)
- [WAL Archiving](#)
- [Backup Management](#)

## Maximum Number of Connections

Excessive GaussDB connections can consume excessive server resources, leading to sluggish operation responses. You can adjust the maximum allowed connections using the **max\_connections** parameter. For details, see [Connection Settings](#).

**max\_connections**: maximum number of concurrent connections to the database. This parameter affects the concurrency capability of the cluster.

## Security Authentication

To ensure user experience and prevent accounts from being cracked, you can configure the following parameters to set the maximum number of login retries and the automatic unlocking time:

- **failed\_login\_attempts**: maximum number of failed login attempts.
- **password\_lock\_time**: number of days after which a locked account is automatically unlocked.

Once detecting that an account is stolen or the account is used to access the database without being authorized, administrators can manually lock the account. Administrators can manually unlock the account if the account becomes normal again.

For example, run the following commands to manually lock and unlock the **joe** user:

- Manually lock the account.  

```
gaussdb=# ALTER USER joe ACCOUNT LOCK;  
ALTER ROLE
```
- Manually unlock the account.  

```
gaussdb=# ALTER USER joe ACCOUNT UNLOCK;  
ALTER ROLE
```

## User Password Security

GaussDB enhances user account security in the following aspects:

- User passwords are stored in the system catalog **pg\_authid**. To prevent password leakage, GaussDB encrypts user passwords before storing them. The cryptographic algorithm is determined by the configuration parameter **password\_encryption\_type**.
- All passwords in GaussDB must have a validity period. You can configure the **password\_effect\_time** parameter to set a validity period for each database user password.

## Permissions Management

- A VPC provides an isolated virtual network for GaussDB instances. You can configure and manage the network as required. A subnet provides dedicated network resources that are logically isolated from other networks for security. If you need to assign different permissions (also known as privileges) to different employees in your enterprise to access your DB instance resources, IAM is a good choice. For details, see [Permissions Management](#).
- To ensure database security and reliability, configure security groups before using a DB instance. For details, see [Configuring Security Group Rules](#).
- Run the following SQL statement to check whether the **PUBLIC** role has the **CREATE** privilege in public schema. If so, any user can create and modify tables or database objects in public schema.

```
SELECT CAST(has_schema_privilege('public','public','CREATE') AS TEXT);
```

- If **TRUE** is returned, run the following SQL statement to revoke the privilege:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
```

- All users are attached to the **PUBLIC** role. If all privileges of an object are granted to the **PUBLIC** role, any user can inherit all the privileges of the object, which violates the principle of least privilege. For this reason, this role should have the minimal privileges for database security purposes. Run the following SQL statement to check whether all privileges are granted to the **PUBLIC** role:  
**SELECT relname,relacl FROM pg\_class WHERE (CAST(relacl AS TEXT) LIKE '%,=arwdDxt/%}' OR CAST(relacl AS TEXT) LIKE '{=arwdDxt/%}') AND (CAST(relacl AS TEXT) LIKE '%,=APmiv/%}' OR CAST(relacl AS TEXT) LIKE '{=APmiv/%}');**
  - If the returned value is empty, all privileges have been granted. In this case, run the following SQL statement to revoke the privileges:  
**REVOKE ALL ON <OBJECT\_NAME> FROM PUBLIC;**
- The **pg\_authid** system catalog in the **pg\_catalog** schema contains information about all roles in a database. To prevent sensitive information from being disclosed or modified, the **PUBLIC** role is not allowed to have any privilege on this system catalog. Run the following SQL statement to check whether privileges on the **pg\_authid** system catalog have been granted:  
**SELECT relname,relacl FROM pg\_class WHERE relname = 'pg\_authid' AND CAST(relacl AS TEXT) LIKE '%,=%}';**
  - If the returned value is not empty, the privileges have been granted. In this case, run the following SQL statement to revoke the privileges:  
**REVOKE ALL ON pg\_authid FROM PUBLIC;**
- Common users are non-administrator users who perform common service operations. Common users should not have administrative privileges beyond their normal scope of responsibilities, such as the privilege to create roles, create databases, audit, monitor, perform O&M operations, or manage security policies. To ensure the principle of least privilege for common users, unnecessary administrative privileges should be revoked while meeting normal business requirements.
- The SECURITY DEFINER function is executed with the privileges of the creator. Improper use of SECURITY DEFINER may cause the function executor to perform unauthorized operations with the privileges of the creator. For this reason, ensure that this function is not misused. For security purposes, the **PUBLIC** role is not allowed to execute functions of the SECURITY DEFINER type. Run the following SQL statement to check whether the **PUBLIC** role has access to any SECURITY DEFINER functions:  
**SELECT a.proname, b.nspname FROM pg\_proc a, pg\_namespace b where a.pronamespace=b.oid and b.nspname <> 'pg\_catalog' and a.prosecdef='t';**
  - If the returned value is not empty, run the following SQL statement to check whether it has the **EXECUTE** privilege:  
**SELECT CAST(has\_function\_privilege('public', 'function\_name([arg\_type][, ...])', 'EXECUTE') AS TEXT);**
    - If **TRUE** is returned, the role has the privilege. In this case, run the following SQL statement to revoke the privilege:  
**REVOKE EXECUTE ON FUNCTION function\_name([arg\_type][, ...]) FROM PUBLIC;**
- The SECURITY INVOKER function is executed with the privileges of the invoker. Improper use of SECURITY INVOKER may cause the function creator

to perform unauthorized operations with the privileges of the executor. Before invoking a function not created by yourself, check the function content to prevent the function creator from performing unauthorized operations with your privileges.

## Database Audit

- GaussDB can record operations you perform on your DB instances. However, only operations supported by Cloud Trace Service (CTS) can be recorded. View the supported operations before performing operations. For details, see [Key Operations Supported by CTS](#).
- Ensure that auditing is enabled for the creation, deletion, and modification of database objects. For details, see [Database Audit](#).
- To view audit logs in a visualized manner, enable **Upload Audit Logs to LTS**. For details, see [Interconnecting with LTS and Querying Database Audit Logs](#).

## WAL Archiving

The Write Ahead Log (WAL) is another term for the transaction log, which is also referred to as the Xlog. It records changes made to the database before they are written to the main storage, ensuring data consistency and durability in case of failures. The **wal\_level** parameter specifies the level of information to be written into a WAL. To enable read-only queries on a standby node, you need to set the **wal\_level** parameter to **hot\_standby** on the primary node and set **hot\_standby** to **on** on the standby node.

## Backup Management

GaussDB provides instance backup and restoration to ensure data reliability. Backups are stored in unencrypted form. To prevent data loss caused by misoperations or service exceptions, you can:

- Configure automated backups and create manual backups. For details, see [Working with Backups](#). When you create a GaussDB instance, the instance-level automated backup policy is enabled by default. After your instance is created, you can modify the automated backup policy as needed.
- Configure an automated backup policy to periodically back up databases. For details, see [Configuring an Automated Backup Policy](#).
- [Export backup information](#).