

GaussDB

User Guide

Issue	01
Date	2025-09-19



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 Instance Selection.....	1
2 Permissions Management.....	8
2.1 Creating a User and Granting Permissions.....	8
2.2 Creating a Custom Policy.....	9
3 Buying a GaussDB Instance.....	11
4 GaussDB Instance Connection.....	24
4.1 Connecting to a GaussDB Instance.....	24
4.2 Connecting to an Instance Through DAS.....	27
4.3 Using gsql to Connect to an Instance.....	30
4.4 Using Navicat to Connect to an Instance.....	37
4.5 Using DBeaver to Connect to an Instance.....	40
5 Database Migration.....	46
5.1 Overview of GaussDB Migration Solutions.....	46
5.2 Using DRS to Migrate Data from Oracle Database to GaussDB.....	51
5.3 Using DRS to Migrate Data from MySQL Database to GaussDB.....	88
5.4 Migrating Data to GaussDB Using the Export and Import Functions of DAS.....	114
5.5 Using the copy to/from Command to Export and Import Data.....	117
5.6 Using CopyManager in JDBC to Export and Import Data.....	120
5.7 Using gs_dump and gs_dumpall to Export Data.....	123
5.8 Using gs_restore to Import Data.....	137
5.9 Using gs_loader to Import Data.....	139
6 Database Use.....	145
6.1 Overview of Using a Database.....	145
6.2 Creating a GaussDB Database.....	147
6.3 Creating a GaussDB Database User.....	148
6.4 Authentication Configurations.....	149
6.4.1 Adding a Client Authentication Configuration.....	149
6.4.2 Deleting a Client Authentication Configuration.....	152
6.4.3 Modifying a Client Authentication Configuration.....	154
6.4.4 Viewing the Modification History of Client Authentication Configurations.....	156
6.4.5 Restoring Client Authentication Configurations.....	159

7 Instance Management.....	161
7.1 Viewing GaussDB Instance Overview Data.....	161
7.2 Configuring Security Group Rules for a GaussDB Instance.....	164
7.3 Binding and Unbinding an EIP for a GaussDB Instance.....	167
7.4 Modifying the Recycle Bin Policy for a GaussDB Instance.....	169
7.5 Exporting Information About All GaussDB Instances.....	171
7.6 Unsubscribing a Yearly/Monthly GaussDB Instance.....	172
7.7 Stopping a GaussDB Instance.....	174
7.8 Starting a GaussDB Instance.....	175
7.9 Rebooting a GaussDB Instance.....	176
7.10 Deleting a Pay-per-Use GaussDB Instance.....	178
7.11 Rebuilding a GaussDB Instance.....	180
7.12 Stopping a GaussDB Node.....	188
7.13 Starting a GaussDB Node.....	190
7.14 Rebooting a GaussDB Node.....	191
8 Instance Modifications.....	193
8.1 Changing the Name of a GaussDB Instance.....	193
8.2 Changing the Database Port of a GaussDB Instance.....	194
8.3 Changing the M Compatibility Port.....	195
8.4 Changing the CPU and Memory Specifications of a GaussDB Instance.....	197
8.5 Configuring Read Replicas.....	200
8.6 Synchronizing Data to a Single-Replica Instance.....	204
8.7 Viewing and Modifying Advanced Features.....	205
8.8 Scaling In and Out an Instance.....	207
8.8.1 Overview of Scaling In and Out an Instance.....	207
8.8.2 Adding Coordinator Nodes for an Instance (Distributed).....	210
8.8.3 Adding Shards for an Instance (Distributed).....	212
8.8.4 Deleting Coordinator Nodes for an Instance (Distributed).....	215
8.8.5 Deleting Shards for an Instance (Distributed).....	217
8.9 Scaling Up Storage Space.....	219
8.9.1 Overview of Scaling Up Storage Space.....	219
8.9.2 Manually Scaling Up Storage Space for an Instance.....	224
8.9.3 Manually Scaling Up Storage Space of Specified Shards.....	226
8.9.4 Configuring Storage Autoscaling for an Instance.....	229
8.10 Changing the Deployment Model.....	234
8.10.1 Overview of Changing the Deployment Model.....	234
8.10.2 Changing the Deployment Model of a Single-Replica Instance (Centralized).....	236
8.10.3 Changing Standby DN to Log Nodes (for a Distributed Instance).....	240
8.11 Performing a Primary/Standby DN Switchover.....	241
8.11.1 Overview of Performing a Primary/Standby DN Switchover.....	241
8.11.2 Changing the DN Failover Priority.....	242
8.11.3 Performing a Primary/Standby Switchover.....	243

9 Instance Upgrade.....	246
9.1 Overview.....	246
9.2 Hot Patch Update.....	249
9.3 In-place Upgrade.....	259
9.4 Gray Upgrade.....	264
10 Add-on Management.....	279
10.1 Installing the Add-on.....	279
10.2 Enabling or Disabling Extensions.....	282
10.3 Viewing Extensions.....	283
11 Data Backup.....	285
11.1 Overview.....	285
11.2 Backup Execution.....	287
11.2.1 Configuring an Automated Backup Policy for GaussDB Instances.....	287
11.2.2 Configuring an Automated Backup Policy for GaussDB Tables.....	292
11.2.3 Setting a Cross-Region GaussDB Backup Policy.....	295
11.2.4 Creating a Manual Backup for GaussDB Instances.....	297
11.2.5 Creating a Manual Backup for GaussDB Tables.....	299
11.3 Backup Management.....	300
11.3.1 Downloading a GaussDB Backup.....	300
11.3.2 Exporting Backup Information About GaussDB Instances.....	303
11.3.3 Stopping a Backup for a GaussDB Instance.....	303
11.3.4 Deleting a Manual Backup of a GaussDB Instance.....	304
12 Data Restoration.....	306
12.1 GaussDB Restoration Methods for Data Misoperations.....	306
12.2 Restoring a Backup File to a GaussDB Instance.....	309
12.3 Restoring a GaussDB Database or Table Using a Backup File.....	317
12.4 Restoring a GaussDB Instance to a Specific Point in Time.....	328
12.5 Restoring a GaussDB Database or Table to a Specific Point in Time.....	337
13 Account and Network Security.....	351
13.1 Resetting the Administrator Password of a GaussDB Instance.....	351
13.2 Enabling Transparent Data Encryption for a GaussDB Instance.....	352
14 Parameter Management.....	355
14.1 Configurable DB Instance Parameters.....	355
14.2 Modifying GaussDB Instance Parameters.....	497
14.3 Viewing Parameter Change History of a GaussDB Instance.....	501
14.4 Exporting Parameters of a GaussDB Instance.....	503
14.5 Creating a Custom Parameter Template for GaussDB Instances.....	504
14.6 Managing Parameter Templates for GaussDB Instances.....	505
15 Monitoring and Alarming.....	510
15.1 Supported Metrics of GaussDB.....	510

15.2 Querying GaussDB Monitoring Metrics.....	536
15.3 Checking GaussDB Monitoring Dashboards.....	537
15.4 Creating Alarm Rules for a GaussDB Instance.....	538
15.5 Event Monitoring.....	541
15.5.1 Supported Events of GaussDB.....	541
15.5.2 Checking GaussDB Event Monitoring Data.....	548
15.5.3 Creating an Alarm Rule to Monitor a GaussDB Event.....	548
16 Logs and Auditing.....	551
16.1 Downloading Error Logs and Slow Query Logs of a GaussDB Instance.....	551
16.2 Downloading Switchover/Failover Logs of a GaussDB Instance.....	558
16.3 Viewing GaussDB Operation Logs on CTS.....	559
16.4 Interconnecting with LTS and Querying Database Audit Logs.....	561
17 Quota Adjustment.....	565
17.1 Adjusting Cloud Service Resource Quotas of GaussDB.....	565
17.2 Adjusting GaussDB Resource Quotas of an Enterprise Project.....	566
18 Diagnostics and Optimization.....	570
18.1 Viewing Slow SQL Statements.....	570
18.2 Viewing Top SQL Statements.....	577
18.3 Viewing Full SQL Statistics Using SQL Explorer.....	581
18.4 Performing Full-Link Analysis for a SQL Record.....	594
18.5 Configuring SQL Throttling Rules.....	597
18.6 Data Storage Insights.....	605
18.7 Performance Reports.....	609
18.7.1 Generating a Workload Diagnosis Report.....	609
18.7.2 Generating an Active Session Profile.....	611
19 Session Management.....	613
19.1 Managing Real-Time Sessions.....	613
19.2 Viewing Session Statistics.....	621
20 Disaster Recovery Management.....	623
20.1 Constraints.....	623
20.2 Creating a DR Relationship.....	624
20.3 Checking DR Task Statuses.....	626
20.4 Promoting the DR Instance to Primary.....	628
20.5 Stopping a DR Task.....	629
20.6 Deleting a DR Task.....	629
20.7 Re-creating a DR Task After the Primary Instance Is Faulty.....	630
20.8 Switching Roles of Primary and DR Instances.....	630
20.9 Re-establishing a DR Relationship.....	631
20.10 Performing a DR Drill.....	631
20.11 Enabling or Disabling Log Cache.....	632

20.12 Restrictions on the DR Instance.....633

21 Managing GaussDB Tasks..... 641

22 Managing GaussDB Tags..... 644

1 Suggestions on Instance Selection

Overview

Before purchasing a GaussDB instance, consider key factors such as cost, performance, workload capacity, and your intended use case. This section describes the billing modes, editions, instance types, deployment models, storage types, and specifications of GaussDB to help you quickly select the right instance.

Billing Modes

GaussDB provides two billing modes: yearly/monthly and pay-per-use. Choose the billing mode that best fits your usage patterns and workload requirements. For details, see [Table 1-1](#).

Table 1-1 Billing modes

Billing Mode	Yearly/Monthly	Pay-per-use
Payment	Prepaid You pay upfront for a subscription term in exchange for a discounted rate. The longer the subscription term, the greater the discount.	Postpaid You can start using the database instance first and then pay as you go.
Billing Cycle	Billed by the subscription term you purchase	Billed by the second and settled by the hour
Billing Item	Instance specifications (vCPUs and memory) and storage	Instance specifications (vCPUs and memory), storage space, and backup space

Billing Mode Change	You can change the billing mode from yearly/monthly to pay-per-use. The change takes effect only after the yearly/monthly subscription expires. For details, see Yearly/Monthly to Pay-per-Use .	You can change the billing mode from pay-per-use to yearly/monthly. For details, see Pay-per-Use to Yearly/Monthly .
Use Case	Applicable to scenarios where the resource usage duration is predictable and intended for long-term use. This mode is more cost-effective.	Applicable to scenarios where compute resource requirements fluctuate. You can enable and delete resources at any time.

Editions

GaussDB provides two editions: basic edition and enterprise edition. Choose an edition that best fits your usage patterns and workload requirements. For details, see [Table 1-2](#).

Table 1-2 Differences between the basic and enterprise editions of GaussDB instances

Edition Type	Basic edition	Enterprise edition
Supported Instance Type	• Centralized • Distributed	• Centralized • Distributed
Supported Deployment Model	• Centralized – 1 primary + 2 standby – 1 primary + 1 standby + 1 log – Single • Distributed • Combined	• Centralized – 1 primary + 2 standby – 1 primary + 1 standby + 1 log – Single • Distributed • Independent
Available Instance Specifications	Supported specification types: • General-purpose • Dedicated • Kunpeng dedicated	Supported specification types: • Dedicated • Kunpeng dedicated

	Minimum specifications: • Centralized 4 vCPUs 16 GB (General-purpose, Dedicated, and Kunpeng dedicated) • Distributed Combined deployment: 16 vCPUs 64 GB (Dedicated and Kunpeng dedicated) NOTE For a combined deployment, all components must be deployed on a single ECS, which demands higher CPU resources to maintain optimal performance. As a result, a minimum of 16 vCPUs is necessary for a combined deployment.	Minimum specifications: • Centralized 4 vCPUs 16 GB (Dedicated and Kunpeng dedicated) • Distributed Independent deployment: 8 vCPUs 64 GB (Dedicated and Kunpeng dedicated)
	Maximum specifications: • x86: 128 vCPUs 1,024 GB • Arm: 128 vCPUs 1,024 GB	Maximum specifications: • x86: 128 vCPUs 1,024 GB • Arm: 128 vCPUs 1,024 GB
Supported Advanced Feature	Advanced compression	Advanced compression, dynamic data masking, and anti-tampering
Performance (Benchmark : Enterprise Edition)	• Centralized instance: same as the enterprise edition • Distributed instance in a combined deployment: 50% of the performance of the enterprise edition, when the same specifications are configured	-

Price (Benchmark : Enterprise Edition)	- When the specifications are the same, the cost of a centralized instance in the basic edition is about 30% of the cost of the enterprise edition. - When the specifications are the same, the cost of a distributed instance in a combined deployment of the basic edition is about 7.5% of the cost of the enterprise edition.	-

Instance Types

Table 1-3 lists the instance types provided by GaussDB. Choose a type that best fits your usage patterns and workload requirements.

Table 1-3 Instance types

Instance Type	Scale-out Supported	Use Case	Component	Service Processing
Distributed	Yes	The data volume is large, and large data capacity and high concurrency are required.	- OM - CM - GTM - ETCD - CN - DN	An application sends a SQL query request to a CN. The CN uses the optimizer of the database to generate an execution plan and sends the plan to DNs. Each DN processes data based on the execution plan. After the processing is complete, DNs return the result sets to the CN for summary. Finally, the CN returns the summary result to the application.
Centralized	No	The data volume is small and stable, and data reliability and service availability are important.	- OM - CM - ETCD - DN	An application sends a task directly to DNs, and the DNs then process and return the results to the application.

Deployment Models

Table 1-4 describes the deployment models supported by GaussDB. Choose a deployment model that best fits your usage patterns and workload requirements.

Table 1-4 Deployment models

Deployment Model		Characteristic	Use Case
Distributed	Independent deployment	The CN, DN, CMS, and GTM components of the database are deployed on different nodes.	Applicable to scenarios where there is a need to process a large amount of data, improve the storage capacity and concurrency of instances through scale-out, and the performance requirements cannot be met by deploying a distributed instance in combined deployment.
	Combined deployment	Different database components are deployed on the same node.	Applicable to scenarios with large data volumes, potential future data volume growth, moderate performance requirements, and a need for a distributed database.
Centralized	1 primary + 2 standby	The 1 primary + 2 standby deployment model is a 3-node deployment where there is one shard.	Applicable to scenarios with small and stable volumes of data, where data reliability and service availability are important.
	1 primary + 1 standby + 1 log	The 1 primary + 1 standby + 1 log deployment model is a 3-node deployment where there is one shard. The log node is only used to store logs and does not incur fees. This reduces costs and resource consumption.	Applicable to the same scenario as the centralized model of 1 primary + 2 standby, but with slightly lower reliability.
	Single (also known as single-replica)	The single-replica deployment model is a single-node deployment where there is only one CMS and one DN.	Applicable only to test scenarios with small data volume.

Storage Types

GaussDB supports **ultra-high I/O** and extreme SSD storage types. Choose a type that best fits your usage patterns and workload requirements. For details, see [Table 1-5](#).

Table 1-5 Storage types

Storage Type	Use Case
Ultra-high I/O	Ultra-high performance cloud disks are excellent for enterprise mission-critical services as well as workloads demanding high throughput and low latency.
Extreme SSD	Superfast disks are ideal for workloads demanding ultra-high bandwidth and ultra-low latency. NOTE Extreme SSD storage is only available for authorized users. To apply for the permissions needed, submit a service ticket .

Specifications

[Table 1-6](#) lists the specifications supported by GaussDB. Choose a specification that best fits your usage patterns and workload requirements.

Table 1-6 Specifications

Specification		CPU Architecture	Supported Edition	Description
General-purpose	General-purpose (1:4)	x86	Basic edition	This specification type uses general-computing ECSs, which prioritize resource sharing and offer better cost-effectiveness. However, if the database is heavily loaded, it can lead to resource contention and fluctuations in computing performance. Consequently, this specification type is not suitable for large services, and it is not offered in the enterprise edition.
Dedicated	- Dedicated (1:4) - Dedicated (1:8)	x86	- Basic edition - Enterprise edition	This specification type uses general computing-plus x86 ECSs. There is no resource contention between ECSs, and the performance is stable.

Specification		CPU Architecture	Supported Edition	Description
Kunpeng dedicated	• Kunpeng dedicated (1:4) • Kunpeng dedicated (1:8)	Arm	• Basic edition • Enterprise edition	This specification type uses general computing-plus ECSs with Kunpeng CPUs. There is no resource contention between ECSs, and the performance is stable.

2 Permissions Management

2.1 Creating a User and Granting Permissions

This section describes how to use [Identity and Access Management \(IAM\)](#) for fine-grained permissions management for your GaussDB resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing GaussDB resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account of Huawei Cloud or a cloud service to perform professional and efficient O&M on your GaussDB resources.

If your Huawei Cloud account does not require individual IAM users, skip this section.

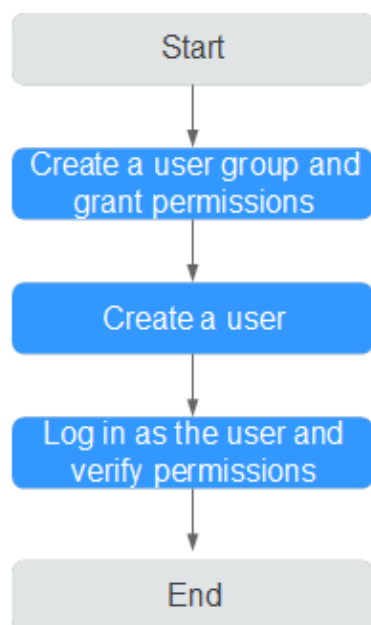
[Figure 2-1](#) describes the process for granting permissions.

Prerequisites

Before assigning permissions to user groups, you should learn about the system-defined permissions of GaussDB listed in [System-defined Permissions](#). For the system policies of other services, see [System-defined Permissions](#).

Process Flow

Figure 2-1 Process of granting GaussDB permissions



1. **Create a user group and assign permissions to it.**
Create a user group on the IAM console, and attach the **GaussDB ReadOnlyAccess** policy to the group.
2. **Create an IAM user and add it to the user group.**
Create a user on the IAM console and add the user to the group created in 1.
3. **Log in** and verify permissions.
Log in to the console by using the created user, and verify that the user only has read permissions for GaussDB.
 - Under the service list, choose **GaussDB**. In the navigation pane on the left, choose **GaussDB > Instances**. Click **Buy DB Instance** in the upper right corner. If a message appears indicating that you have insufficient permissions to perform the operation, the GaussDB ReadOnlyAccess policy has already taken effect.
 - Choose any other service in the service list. If a message appears indicating that you have insufficient permissions to access the service, the **GaussDB ReadOnlyAccess** policy has already taken effect.

2.2 Creating a Custom Policy

Custom policies can be created to supplement the system-defined policies of GaussDB. For the actions supported for custom policies, see [Permissions Policies and Supported Actions](#).

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.

- JSON: Create a policy in JSON format or edit the JSON strings of an existing policy.

For details about how to create a custom policy, see the section [Creating a Custom Policy](#). The following contains examples of common GaussDB custom policies.

Example Custom Policy

- Example 1: Allowing users to create GaussDB instances

```
{
  "Version": "V2.0-8.103",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["gaussdb:instance:create"]
  }]
}
```

- Example 2: Denying GaussDB instance deletion

A policy with only "Deny" permissions must be used in conjunction with other policies. If the permissions assigned to a user include both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

The following method can be used if you need to assign permissions of the **GaussDB FullAccess** policy to a user but you want to prevent the user from deleting GaussDB instances. Create a custom policy for denying GaussDB instance deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on GaussDB instances except deleting GaussDB instances. The following is an example of a deny policy:

```
{
  "Version": "V2.0-8.103",
  "Statement": [{
    "Action": ["gaussdb:instance:delete"],
    "Effect": "Deny"
  }]
}
```

3 Buying a GaussDB Instance

Scenarios

You can buy a GaussDB instance on the management console.

GaussDB supports pay-per-use and yearly/monthly billing. GaussDB allows you to tailor your computing resources and storage to your business needs.

Prerequisites

- Before purchasing an instance, ensure that there is a valid payment method configured.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click **Buy DB Instance**.

Step 5 On the displayed page, select a billing mode, configure parameters about the instance, and click **Next**.

Figure 3-1 Billing mode and basic information

Billing Mode

Yearly/Monthly

Pay-per-use ⓘ

Region

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

Project

DB Instance Name

X ⓘ

Edition Type

Enterprise edition

Basic edition

DB Engine Version

V2.0-8.103

V2.0-3.227

DB Instance Type

Distributed

Centralized

Deployment Model

Independent ⓘ

Log Nodes Supported

☐ Yes

Transaction Consistency

Strong consistency

Eventual consistency ⓘ

Failover Priority

Reliability

Availability ⓘ

Replicas

−

3

+

ⓘ

Shards

−

3

+

Coordinator Nodes

−

3

+

ⓘ

If Coordinator Nodes is set to 1, the instance can only be used for testing.

AZ

cn-north-4a

cn-north-4b

cn-north-4c

AZ7

Table 3-1 Basic information

Parameter	Description
Billing Mode	GaussDB provides yearly/monthly billing and pay-per-use billing. • Yearly/Monthly: You pay upfront for the amount of time you expect to use the DB instance for. You will need to make sure you have a top-up account with a sufficient balance or have a valid payment method configured first. For distributed instances using the combined deployment model, yearly/monthly billing is only available to authorized users. To apply for the permissions needed, submit a service ticket. • Pay-per-use: You can start using the DB instance first and then pay as you go. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.
Region	A region where the tenant is located. You can change the region on the instance creation page, or go back to the Instances page and change it in the upper left corner. Products in different regions cannot communicate with each other over a private network. After the instance is created, you cannot change its region.
DB Instance Name	The instance name is case-sensitive, must start with a letter, and can contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.
Edition Type	GaussDB provides Basic edition and Enterprise edition. The basic edition lacks certain advanced features that are available in the enterprise edition. The basic edition delivers the same level of performance as the enterprise edition at a more affordable price. This edition is ideal for users who prioritize cost and do not need advanced features. For more details about the differences between the two editions, see Differences Between the Basic and Enterprise Editions of GaussDB Instances.
DB Engine Version	Select the GaussDB database version of the new instance. NOTE • Instances of V2.0-8.102 or later run on Huawei Cloud EulerOS. • A DR relationship cannot be established between instances running on Huawei Cloud EulerOS and those running on EulerOS.

Parameter	Description
DB Instance Type	• Distributed: You can add nodes for distributed instances as needed to handle large volumes of concurrent requests. • Centralized: Centralized instances are suitable for scenarios with small and stable volumes of data, where data reliability and service availability are extremely important.
Deployment Model	• Distributed – Independent: Database components are deployed on different nodes. This model is suitable for where high availability and stability are required and the instance scale is large. – Combined: Different database components are deployed on the same node. This model is only available for basic edition instances of V2.0-3.223 or later. Storage autoscaling is not supported by this model. • Centralized – HA (1 primary + 2 standby): 3-node deployment where there is one shard. The shard contains one primary DN and two standby DNs. – Single: single-node deployment where there is only one CMS component and one DN. This model is only available for instances of V2.0-2.2 or later. This single-node deployment cannot guarantee service availability (SLA) and is not suitable for production environments. It is only recommended for R&D and testing purposes. – 1 primary + 1 standby + 1 log: 3-node deployment where there is one shard with three replicas. The shard contains one primary DN, one standby DN, and one log-dedicated DN. This model is only available for instances of V2.0-3.200 or later. This option is available only when Resource Type is set to Elastic Cloud Server (ECS). CAUTION • Single: The availability (or SLA) cannot be guaranteed because the instance is deployed on a single server.
Log Nodes Supported	This parameter is only available for distributed instances. If this option is selected, the distributed instance will be created using the 1 primary + 1 standby + 1 log deployment model. By default, centralized instances support the 1 primary + 1 standby + 1 log deployment model. You can simply set Deployment Model to 1 primary + 1 standby + 1 log as needed when creating a centralized instance.

Parameter	Description
Transaction Consistency	This parameter is only available for distributed instances. • Strong consistency: When an application updates data, every user can query all data that has been successfully committed, but performance is affected. • Eventual consistency: When an application updates data, the data users queried may be different, and some users may not obtain the most current value. The most current data may take a bit of time to become available for query by all users. However, DB instances with eventual consistency generally have higher performance. Eventual consistency cannot ensure strong read consistency of distributed transactions and consistency of transactions that depend on query results, such as <code>INSERT INTO SELECT * FROM</code>. Write operations that are split into multiple statements or involve in multiple nodes are not supported.
Failover Priority	This parameter is only available for distributed instances. This parameter is only available for authorized users. To apply for the permissions needed, submit a service ticket. The default value is Reliability. For details about how to change the failover priority for an existing instance, see Changing Failover Priority. • Reliability: Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency. • Availability: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. If Availability is selected, exercise caution when modifying the following database parameters. For details about how to modify parameters, see Modifying Instance Parameters. – recovery_time_target: If the value of this parameter is changed, the instance will experience frequent forced failovers. Contact technical support before modifying this parameter. – audit_system_object: If the value of this parameter is changed, DDL audit logs will be lost. Contact technical support before modifying this parameter.
Replicas	This parameter is only available for distributed instances. Total number of DN's each shard, primary and standby DN's combined. There are three replicas in a shard, indicating that there are one primary and two standby DN's in a shard.

Parameter	Description
Shards	This parameter is only available for distributed instances. It specifies the number of shards in an instance. A shard contains multiple DNs. The number of DNs in a shard depends on the value of Replicas, for example, if Replicas is set to 3, there are three DNs (one primary and two standby DNs) in a shard. The value ranges from 1 to 128. You can add up to 256 shards for an existing instance. For details, see Adding Shards for an Instance.
Coordinator Nodes	This parameter is only available for distributed instances. It indicates the number of CNs in an instance. The value ranges from 1 to 128. You can add up to 256 CNs for an existing instance. For details, see Adding Coordinator Nodes for an Instance. It is recommended that at least two CNs be configured for an instance. Configuring only one CN results in low service reliability, making it best suited for test environments rather than production use. A CN provides the following functions: • It receives access requests from applications and returns execution results to clients. • It breaks down tasks and distributes task fragments to different DNs for parallel processing.
AZ	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. A DB instance can be deployed in one AZ or three AZs. Three AZs are recommended. To deploy a GaussDB instance in Intelligent EdgeCloud (IEC), select an IEC HomeZones for AZ. For assistance with this setup, submit a service ticket to contact Huawei engineers.
Time Zone	Select a time zone according to the region hosting your DB instance when you buy the instance.

Figure 3-2 Specifications and storage

Instance Specifications

Dedicated(1:8)

Flavor Name

4 vCPUs | 32 GB

Unavailable for production environment

8 vCPUs | 64 GB

16 vCPUs | 128 GB

32 vCPUs | 256 GB

64 vCPUs | 512 GB

DB Instance Specifications

Dedicated(1:8) | 8 vCPUs | 64 GB

Storage Type

Ultra-high I/O

Learn more about storage types.

Storage Space (GB)

480

120

14480

28840

43200

72000

-

480

+

GaussDB provides free backup storage equal to the amount of your purchased storage space. After the free backup space is used up, charges are applied based on the [backup space pricing details](#).

Disk Encryption

Disable

Enable

Table 3-2 Specifications and storage

Parameter	Description
Instance Specifications	CPU and memory specifications of the instance. Different instance specifications support different numbers of database connections. For details, see Instance Specifications.
Storage Type	The storage type determines the read/write speed of an instance. The higher the maximum throughput is, the higher the instance read/write speed can be. GaussDB supports the ultra-high I/O and extreme SSD storage types. - Ultra-high I/O: Ultra-high performance cloud disks excellent for enterprise mission-critical services as well as workloads demanding high throughput and low latency. When striped across multiple disks, they can achieve up to 800 MB/s throughput. - Extreme SSD: Superfast disks ideal for workloads demanding ultra-high bandwidth and ultra-low latency. When striped across multiple disks, they can achieve up to 2,500 MB/s throughput. NOTE Extreme SSD storage is only available for authorized users. To apply for the permissions needed, submit a service ticket.

Parameter	Description
Storage Space	The allocated storage space contains necessary file system overhead required for inodes, reserved blocks, and database operations. When you create an instance, the storage for a single shard starts from 40 GB. You can increase the storage in increments of 4 GB. If too little storage is configured, when traffic volume is high, the storage can be used up quickly, and the instance will change to read-only. Select an amount of storage based on how much traffic you expect there to be. After buying an instance, you can scale up its storage. For details, see Scaling Up Storage Space.
Free Backup Space	GaussDB provides free backup storage equal to the amount of your purchased storage space. After the free backup space is used up, you will be billed for the additional space used.
Disk Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. Key Name: If disk encryption is enabled, you need to select or create a key. NOTE • When creating a pay-per-use instance with disk encryption enabled, ensure that your account has been granted the following action: kms:cmk:createGrant • When you create a yearly/monthly instance with disk encryption enabled, if the KMS key shared by RAM is used for the first time, create an agency as prompted. • When using the KMS key shared by RAM to create an instance, ensure that your account has also been granted the following actions: – iam:agencies:listAgencies – iam:agencies:createAgency – iam:roles:listRoles – iam:permissions:listRolesForAgency – iam:permissions:grantRoleToAgency – kms:cmk:createGrant If a shared KMS key is used, the corresponding CTS event is createGrant. Only the key owner can receive this event.
Transparent Data Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. If you select Enable, Key Name indicating the tenant key needs to be specified.

Figure 3-3 Network and database configuration

Relationship among VPCs, subnets, security groups, and DB instances. ⓘ

VPC

default_vpc

default_subnet

ⓘ

If you want to create a VPC, go to [the VPC console](#).

Security Group

default

[View Security Group](#) ⓘ

In a security group, rules that authorize connections to DB instances apply to all DB instances associated with the security group.

Ensure that the TCP ports in the inbound rule of the selected security group contain 8000-8100, 20050, 5000-5001, 2379-2380, 6000, 6500, 40000-60480.

Security Group Rules

▼

[Add Inbound Rule](#)

Database Port

Default port: 8000

Administrator

root

Administrator Password

ⓘ

Keep your password secure. The system cannot retrieve your password.

Confirm Password

ⓘ

Parameter Template

Default-Enterprise-Edition-GaussDB-8.10...

[View Parameter Template](#) ⓘ

Enterprise Project

default

[View Enterprise Projects](#) ⓘ ⓘ

Tag

TMS's predefined tags are recommended for adding the same tag to different cloud resources. [Create predefined tags](#) ⓘ ⓘ

[+ Add Tag](#)

You can add 20 more tags.

Table 3-3 Network

Parameter	Description
VPC	A virtual network where your GaussDB instances are located. A VPC isolates networks for different workloads. You need to create or select the required VPC. If no VPC is available, GaussDB allocates a default VPC for you. After the GaussDB instance is created, the VPC cannot be changed. For details about how to create a VPC, see Creating a VPC. With VPC sharing, you can also use a VPC and subnet shared by another account. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). This allows for more efficient use of network resources and reduces O&M costs. For more information about VPC subnet sharing, see VPC Sharing in the <i>Virtual Private Cloud User Guide</i>.

Parameter	Description
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within a specific AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets in which you plan to create GaussDB instances and cannot be disabled. GaussDB supports automatic IP address allocation during instance creation. If an IEC HomeZones is selected for AZ, create a subnet in that zone for this parameter. NOTE By default, a subnet supports up to 256 IP addresses. A distributed instance can require up to 1,286 IP addresses. You are advised to use a subnet that can provide 2,048 IP addresses.
Security Group	Controls access to and from an instance. When you buy an instance, the selected security group must meet the following requirements: - When buying a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6001, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound security group rules allow all outbound traffic. - When buying a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.) The security group enhances security by controlling access to GaussDB from other services. Ensure that the security group you selected allows your client to access the instance. If you do not need to specify a security group when creating an instance, you can submit a service ticket. If no security group is available, GaussDB allocates a default security group for you.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: 8000. The following ports are used by the system and cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.

Parameter	Description
Single Private IP Address	Specifies whether to enable the single private IP address policy. If this policy is enabled, only one private IP address is assigned to an instance and is bound to the primary node. The private IP address does not change after a primary/standby switchover. If this policy is disabled, each node is bound to a private IP address, and the private IP address changes after a primary/standby switchover. The constraints on the single private IP address policy are as follows: • This policy is only available for centralized instances of version V2.0-3.207 or later. • This policy is configurable only at the time of instance creation and cannot be changed afterward.

Table 3-4 Database configuration

Parameter	Description
Administrator	Database administrator. The default username is root .
Administrator Password	Enter a strong password and periodically change it to improve security, preventing security risks such as brute force cracking. The password must contain: • 8 to 32 characters. • At least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*-_+=?, Keep the password secure. Password retrieval is not supported. After a DB instance is created, you can reset this password. For details, see Resetting the Administrator Password.
Confirm Password	Enter the administrator password again.

Table 3-5 Parameter templates

Parameter	Description
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. After creating an instance, you can modify the parameter template. You can modify the instance parameters as required after the instance is created. For details, see Modifying Parameters in a Parameter Template.
Enterprise Project	If the instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the Enterprise Project Management console to create a project. For details, see Enterprise Management User Guide.

Table 3-6 Tags

Parameter	Description
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each instance can have up to 20 tags. If your organization has configured tag policies for GaussDB, add tags to instances based on the policies. If a tag does not comply with the policies, instance creation may fail. Contact your organization administrator to learn more about tag policies.

If you have any questions about the price, click **Pricing details** at the bottom of the page.

NOTE

The performance of your GaussDB instance depends on its settings. Hardware items include the instance specifications, storage type, and storage space.

Step 6 Confirm the displayed details.

For pay-per-use instances, confirm the specifications.

- If you need to modify your settings, click **Previous**.
- If you do not need to modify your settings, click **Submit**.

For yearly/monthly instances, confirm the order.

- If you need to modify your settings, click **Previous**.

- If you do not need to modify your settings, click **Pay Now** to go to the payment page. On the displayed page, select a payment method and click **Pay**.

Step 7 After the creation task is submitted, check the instance status on the **Instances** page. When a GaussDB instance is being created, its status is **Creating**. This process takes about 10 to 20 minutes. Once created, the instance becomes **Available**. You can view and manage it on the **Instances** page.

An automated full backup is immediately triggered once your instance is created.

----End

Related Operations

Once your instance is ready, you can create alarm rules, configure automated backup policies, scale in or out the instance, scale up its storage, modify instance parameters, and change the database port. For details, see:

- [Creating Alarm Rules](#)
- [Configuring an Automated Backup Policy](#)
- [Scaling In and Out an Instance](#)
- [Scaling Up Storage Space](#)
- [Modifying Instance Parameters](#)
- [Changing the Database Port](#)

FAQ

- [What Are the Differences Between the Independent Deployment Model and Combined Deployment Model of GaussDB Distributed Instances?](#)
- [What Are the Differences Between the General-Purpose, Dedicated, and Kunpeng Dedicated Specification Types Provided by GaussDB?](#)
- [Can I Switch Between the GaussDB Enterprise Edition and Basic Edition?](#)

4 GaussDB Instance Connection

4.1 Connecting to a GaussDB Instance

GaussDB instances can be connected using gsql, DBeaver, Navicat, or Data Admin Service (DAS).

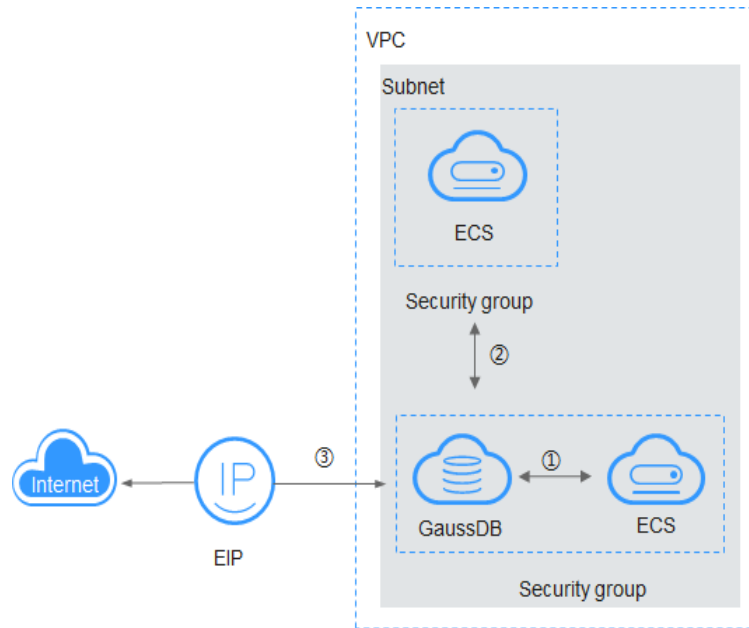
Table 4-1 GaussDB instance connection modes

Connect Through	IP Address	Description	Comments
DAS	Not required	Huawei Cloud DAS enables you to manage databases on a web-based console. It supports SQL execution, advanced database management, and intelligent O&M, simplifying database management and improving both efficiency and data security. The permissions required for connecting to a GaussDB instance through DAS are enabled by default.	Easy to use, secure, advanced, and intelligent

Connect Through	IP Address	Description	Comments
gsql	Private IP address/EIP	gsql is a client tool provided by GaussDB. You can use gsql to connect to the database and then enter, edit, and execute SQL statements in an interactive manner.	To achieve a higher data transmission rate and security level, migrate your applications to a server that is in the same subnet as your GaussDB instance and use a private IP address to access the instance. The bandwidth is not limited for private network connections.
DBeaver	EIP	DBeaver is a GUI-based database management tool. You can use this tool to view database schemas, execute SQL queries and scripts, browse and export data, process BLOB/CLOB data, and modify database schemas.	Open-source and easy-to-use
Navicat	EIP	Navicat is a database management tool. You can easily view and edit data on its graphical interface. For example, you can insert, delete, update, and query data, process SQL statements or scripts, use functions, and generate data.	Stable and easy to use

[Figure 4-1](#) shows how an instance is connected.

Figure 4-1 Connecting to an instance through a private network and an EIP



- ① Connect through a private network (ECS and GaussDB in the same security group)
- ② Connect through a private network (ECS and GaussDB in different security groups)
- ③ Connect through a public network

 NOTE

- If the ECS and GaussDB instance are in the same VPC and security group, they can communicate with each other through the private network by default. In this case, you can connect to the instance through a private IP address.
- If the ECS and GaussDB instance are in the same VPC but different security groups, you need to set security group rules for both the GaussDB instance and ECS, and then connect to the instance through a private IP address.
 - GaussDB instance: Configure an **inbound** rule for the security group with which the GaussDB instance is associated. For details, see [Configuring Security Group Rules for a GaussDB Instance](#).
 - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security group rule for the ECS. If **not all outbound traffic is allowed** in the security group, you need to configure an **outbound** rule for the ECS to allow all outbound packets.
- If the ECS and GaussDB instance are in different VPCs, you can bind an EIP to the ECS and use the EIP to connect to the instance. Ensure that both the ECS and GaussDB instance have EIPs.
 - For details about how to bind an EIP to an ECS, see [Binding an EIP](#).
 - For details about how to bind an EIP to a GaussDB instance, see [Binding an EIP](#).
- Database connection specifications:
 - As concurrency increases, establishing a new connection takes longer, growing from milliseconds to seconds.
 - When the database is under high load (CPU usage $\geq 80\%$), the connection establishment time increases significantly, for example, from 210 ms to 1,400 ms for 50 concurrent connections.
 - To ensure connection establishment time remains below 500 ms, limit the connection rate to no more than 20 connections per second.

4.2 Connecting to an Instance Through DAS

Scenarios

Data Admin Service (DAS) is a one-stop management platform that allows you to manage Huawei Cloud databases on a web console. It offers database development, O&M, and intelligent diagnosis, making it easy for you to use and maintain databases.

This section describes how to connect to a GaussDB instance through DAS.

Procedure

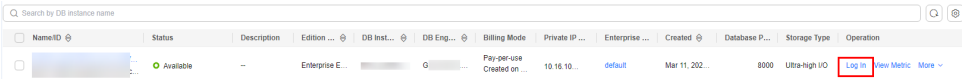
Step 1 [Log in to the management console](#).

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

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

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

Figure 4-2 Logging in to an instance



Alternatively, click the DB instance name on the **Instances** page. On the displayed **Basic Information** page, click **Log In** in the upper right corner of the page.

Figure 4-3 Logging in to an instance



Step 5 On the **Custom Login** page, select the node to be logged in to. Enter the correct database username and password, and click **Test Connection**. After the connection test is successful, click **Log In**.

Figure 4-4 Login page

A screenshot of the 'Instance Login Information' page. It shows the 'DB Instance Name' and 'DB Engine Version'. Under the 'Custom Login' tab, there is a 'Node Information' table with columns 'Name', 'Role', and 'Status'. Three nodes are listed: 'gauss-7264-root_0' (master, Available), 'gauss-7264-root_1' (slave, Available), and 'gauss-7264-root_2' (slave, Available). Below the table, there are input fields for 'Database Name' (postgres), 'Login Username' (root), and 'Password' (masked). A 'Test Connection' button is next to the password field. Below the password field, there is a message 'Connection is successful.' and a checked checkbox for 'Remember Password' with a note 'Your password will be encrypted and stored securely.' There is also a toggle for 'Show Executed SQL Statements' which is currently turned on. At the bottom right, there are 'Cancel' and 'Log In' buttons.

Table 4-2 Parameter description

Parameter	Description
Login Username	Username of the GaussDB database account. The default administrator is root .
Database Name	Name of the database to be connected. The default management database is postgres .
Password	Password of the database user.

Parameter	Description
Show Executed SQL Statements	You are advised to enable Show Executed SQL Statements . With it enabled, you can view the executed SQL statements under SQL Operations > SQL History and execute them again without entering the SQL statements.

For details about how to use DAS to manage databases, see [GaussDB Management](#).

----End

Follow-up Operations

After logging in to the instance, you can create databases, create database users, and migrate databases.

- [Creating a Database](#)
- [Migrating a Database](#)
- [Creating a Database Through DAS](#)
- [Creating a Database User Through DAS](#)

What Can I Do If the DAS Console Is Not Displayed After I Click Log In in the Operation Column of an Instance on the Instances Page?

Set your browser to allow pop-ups and try again.

What Should I Do If I Can't Connect to My DB Instance Due to Insufficient Permissions?

1. Error message: You do not have the required permission. The policy does not allow action das:connections:xxx.
Error cause: Your account does not have the DAS FullAccess permission.
Solution: Add the DAS FullAccess permission by referring to [Creating a User and Granting Permissions](#).
2. Error message: You do not have the permission to perform this operation. Contact your administrator to request the required permission.
Error cause: Your account does not have the DAS FullAccess permission.
Solution: Add the DAS FullAccess permission by referring to [Creating a User and Granting Permissions](#).
3. Error message: Your current account only has the read-only permission and cannot perform this operation. To ensure that you can use DAS smoothly, add the DAS Administrator permission.
Error cause: Your account does not have the DAS FullAccess permission.
Solution: Add the DAS FullAccess permission by referring to [Creating a User and Granting Permissions](#).

What Should I Do if I Forget My Password When Connecting to an Instance?

On the GaussDB console, reset the administrator password. For details, see [Resetting the Administrator Password](#).

4.3 Using gsql to Connect to an Instance

This section describes how to use the gsql client to connect to a GaussDB instance you have bought on the GaussDB management console.

- [Step 1: Buy an ECS](#)
- [Step 2: Query the IP Address and Port Number of the Instance to Be Connected](#)
- [Step 3: Test the Connectivity](#)
- [Step 4: Obtain the Driver Package](#)
- [Step 5: Connect to the Database](#)
 - [Non-SSL connection](#)
 - [SSL connection](#)

Buying an ECS

If you want to connect to a database using the command-line interface (CLI), like gsql, you need to create an ECS and install gsql on it.

1. [Log in to the management console](#) and check whether there is an available ECS.
 - If there is, go to [3](#).
 - If there is not, go to [2](#).

Figure 4-5 ECS instances



NameID	Monitor	Security	Status	AZ	Specifications	OS Type	IP Address	Billing Mode	Enterprise Project	Tag	Operation
ecs-8989			Running	AZ1	2 vCPUs, 4 GB (7 large-2)	Linux	122.122.122.122 (Private IP)	Pay-as-you-go	default	--	Restart Login More

2. Buy an ECS that runs EulerOS.
For details about how to buy a Linux ECS, see [Purchasing an ECS](#) in *Elastic Cloud Server Getting Started*.
3. On the **ECS Information** page of the target ECS, view the region and VPC of the ECS.

Figure 4-6 ECS basic information

ECS Information	
ID	
Name	ecs-6969 ↗
Description	-- ↗
Region	
AZ	AZ1
Specifications	General computing-plus 2 vCPUs 4 GiB c7.large.2
Image	CentOS 8.2 64bit Public image
VPC	default_vpc
Global EIP	-- Bind

Billing Mode	Pay-per-use
Created	Sep 24, 2024 18:03:03 GMT+08:00
Launched	Sep 24, 2024 18:03:17 GMT+08:00
Deletion Time	-- Modify

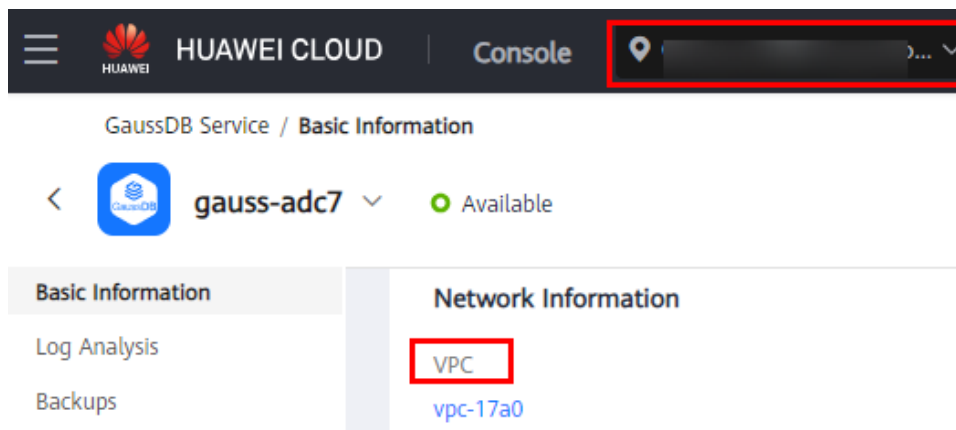
NOTE

The ECS must run on EulerOS. gsql supports the following OS versions:

For x86 servers: EulerOS V2.0SP5 and Kylin V10 SP2

For Kunpeng servers: EulerOS V2.0SP8 and Kylin V10 SP1

4. On the **Basic Information** page of your GaussDB instance, view the region and VPC of the instance.

Figure 4-7 Basic information about a GaussDB instance

5. Check whether the ECS and GaussDB instance are in the same region and VPC.
 - If the ECS and GaussDB instance are in the same region and VPC, the DB instance can be connected through a private network. For details about how to obtain the private IP address, see [Querying the IP Address of the Instance to Be Connected](#).
 - If the ECS and DB instance are in different VPCs, the DB instance must be connected over a public network. For details about how to obtain the public IP address, see [Querying the IP Address of the Instance to Be Connected](#). Ensure that both the ECS and GaussDB instance have EIPs.
 - For details about how to bind an EIP to an ECS, see [Binding an EIP](#).
 - For details about how to bind an EIP to a GaussDB instance, see [Binding an EIP](#).

Querying the IP Address and Port Number of the Instance to Be Connected

1. [Log in to the management console](#).
2. Click  in the upper left corner and select a region and project.
3. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
4. On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
5. In the **Node List** area, view the IP address. In the **Network Information** area, view the port number.

For a centralized instance, obtain the IP address of the primary DN. For a distributed instance, obtain the IP address of any CN.

- If the ECS and GaussDB instance are in the same VPC, obtain the private IP address.
- If the ECS and GaussDB instance are in different VPCs, obtain the EIP.

Testing Connectivity

1. Log in to the ECS. For details, see [Logging In to a Linux ECS Using VNC](#) in *Elastic Cloud Server User Guide*.
2. On the ECS, check whether it can connect to the target GaussDB instance using the IP address and port number obtained in [Querying the IP Address and Port Number of the Instance to Be Connected](#).

telnet *IP address Port number*

Example:

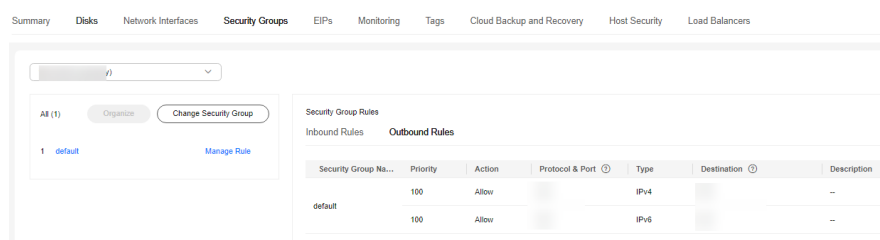
telnet 192.168.0.16 8000

NOTE

If the message "command not found" is displayed, install a Telnet client that matches the OS of the ECS.

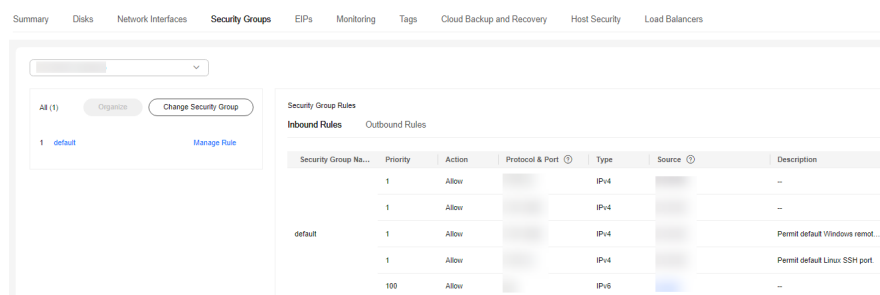
- If the ECS can connect to the DB instance, no further action is required.

- If the communication fails, check the security group rules.
 - On the **Outbound Rules** page of the ECS, add the IP address and port of the GaussDB instance to the outbound rules.
 - If the ECS and GaussDB instance are in the same VPC, add the private IP address and port of the GaussDB instance to the outbound rules.
 - If the ECS and GaussDB instance are in different VPCs, add the EIP address and port of the GaussDB instance to the outbound rules.

Figure 4-8 ECS security group

- On the **Inbound Rules** page of the GaussDB instance, add the IP address and port of the ECS to the inbound rules.
 - If the ECS and GaussDB instance are in the same VPC, add the private IP address and port of the ECS to the inbound rules.
 - If the ECS and GaussDB instance are in different VPCs, add the EIP address and port of the ECS to the inbound rules.

For details, see [Configuring Security Group Rules](#).

Figure 4-9 GaussDB security group

Obtaining the Driver Package

Download particular packages listed in [Table 4-3](#) based on the version of your instance.

Table 4-3 Driver package download list

Version	Download Address
V2.0-8.x	Driver package Verification package for the driver package
V2.0-3.x	Driver package Verification package for the driver package
V2.0-2.x	Driver package Verification package for the driver package

To prevent a software package from being tampered with during transmission or storage, download the corresponding verification package and perform the following steps to verify the software package:

1. Upload the software package and verification package to the same directory on a Linux VM.
2. Run the following command to verify the integrity of the software package:

```
cat GaussDB_driver.zip.sha256 | sha256sum --check
```

If **OK** is displayed in the command output, the verification is successful.

```
GaussDB_driver.zip: OK
```

Connecting to a Database

GaussDB lets you select either SSL or non-SSL connections as required.

Non-SSL Connection

1. Log in as the **root** user to the ECS you have created.
2. Upload the client tool package and configure gsql environment variables.
 - a. Run the following command to create the **/tmp/tools** directory for storing the client tool package:

```
mkdir /tmp/tools
```
 - b. Download the **GaussDB_driver.zip** driver package of the required version by referring to [Obtaining the Driver Package](#), and upload it to the **/tmp/tools** directory of the created ECS.
 - c. Run the following commands to decompress the **GaussDB_driver.zip** driver package:

```
cd /tmp/tools
unzip GaussDB_driver.zip
```
 - d. Run the following commands to copy the decompressed **GaussDB-Kernel_***_EULER_64bit-Gsql.tar.gz** client tool package to the **/tmp/tools** directory:

 NOTE

This section uses the gsql tool package suitable for the centralized instances running on Euler2.5_x86_64 as an example. The relative path of the tool package varies depending on where you decompressed it.

```
cd /tmp/tools/GaussDB_driver/Centralized/Euler2.5_X86_64/  
cp GaussDB-Kernel_***_EULER_64bit-Gsql.tar.gz /tmp/tools
```

- e. Run the following commands to decompress the package:

```
cd /tmp/tools  
tar -zxvf GaussDB-Kernel_***_EULER_64bit-Gsql.tar.gz
```

- f. Configure environment variables.

Run the following command to open the ~/.bashrc file:

```
vim ~/.bashrc
```

Press **G** to move the cursor to the last line, press **i** to enter Insert mode, and type the following information. Then, press **Esc** to exit Insert mode, and run **:wq** to save the settings and exit.

```
export PATH=/tmp/tools/bin:$PATH  
export LD_LIBRARY_PATH=/tmp/tools/lib:$LD_LIBRARY_PATH
```

Run the following command to make the environment variables take effect permanently:

```
source ~/.bashrc
```

3. Enter the password when prompted to connect to the database.

After an instance is created, a **postgres** database is generated by default. Database **postgres** is used as an example.

```
gsql -d postgres -h 10.0.0.0 -U root -p 8000  
Password for user root:
```

postgres is the name of the database you want to connect. **10.0.0.0** is the IP address of the instance obtained in [Querying the IP Address of the Instance to Be Connected](#). **root** is the username for logging in to the database. **8000** is the database port obtained in [Querying the Port Number of the Instance to Be Connected](#).

If the following information is displayed, the connection is successful:
gaussdb=>

For more information about gsql commands, see [Tool Reference](#).

SSL Connection

1. [Log in to the management console](#).
2. Click  in the upper left corner and select a region and project.
3. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
4. On the **Instances** page, click the name of the target instance. In the **Configuration** area on the **Basic Information** page, click  next to the **SSL** field to download the root certificate or certificate bundle.
5. Upload the root certificate to the ECS or save it to the device to be connected to the GaussDB instance.

Import the root certificate to the Linux ECS. For details, see [How Can I Import the Root Certificate to a Windows or Linux OS?](#)

6. Connect to a GaussDB instance.

A Linux ECS is used in this example. Run the following command to set environment variables on the ECS:

```
export PGSSLMODE=<sslmode>  
export PGSSLROOTCERT=<ca-file-directory>
```

```
gsql -h <host> -p <port> -d <database> -U <user>
```

Example:

```
export PGSSLMODE="verify-ca"  
export PGSSLROOTCERT="/home/Ruby/ca.pem"
```

```
gsql -h 10.0.0.0 -p 8000 -d postgres -U root
```

Password for user root:

Table 4-4 Parameter description

Parameter	Description
<host>	IP address of the DB instance. To obtain the IP address, click the instance name on the Instances page to go to the Basic Information page of the instance. The IP address can be found in the Private IP Address column of the Node List area.
<port>	Database port number. The default value is 8000 . To obtain the database port, click the instance name on the Instances page to go to the Basic Information page of the instance. The database port can be founded in the Database Port field of the Network Information area.
<database>	Name of the database to connect to. The default database is postgres .
<user>	Username of the GaussDB database account. The default administrator is root .
<ca-file-directory>	Directory of the CA certificate for SSL connection.
<sslmode>	SSL connection mode. Set it to verify-ca to verify that the server is trustworthy by checking the certificate chain.

For more information about gsql commands, see [Tool Reference](#).

7. Check the command output after you log in to the database. If information similar to the following is displayed, the SSL connection has been established.
- ```
SSL connection (cipher: DHE-RSA-AES256-GCM-SHA384, bits: 256)
```

## FAQ

### What Should I Do If I Can't Connect to My GaussDB Instance?

## 4.4 Using Navicat to Connect to an Instance

Navicat Premium 16.2.8 for Windows PC now supports GaussDB management and development. This section describes how to use Navicat to connect to a GaussDB instance.

### NOTE

This section uses Navicat 16.3.3 as an example. Navicat's graphical user interfaces (GUIs) differ depending on the version you are using. To learn more about how to perform operations in different versions, refer to the product documentation specific to that version.

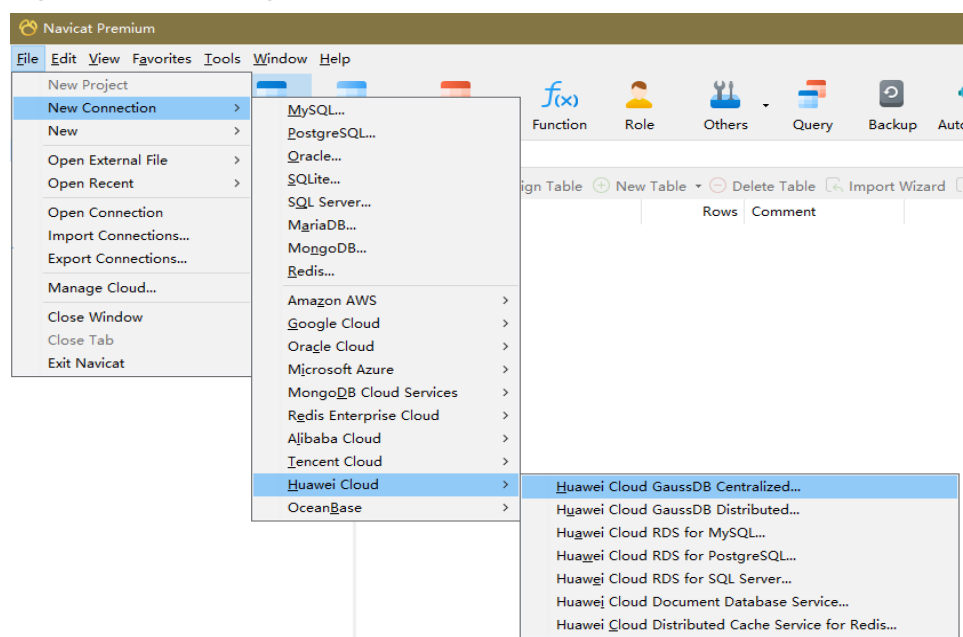
### Prerequisites

You have [downloaded](#) or bought Navicat Premium and installed it on the local PC.

### Procedure

- Step 1** Start the Navicat Premium client and choose **File > New Connection > Huawei Cloud > Huawei Cloud GaussDB Centralized or Huawei Cloud GaussDB Distributed**.

**Figure 4-10** Creating a connection



- Step 2** In the **New Connection** window, enter the correct connection name, host, port, initial database, user name, and password.



**Figure 4-11** Setting information for connecting to a centralized instance

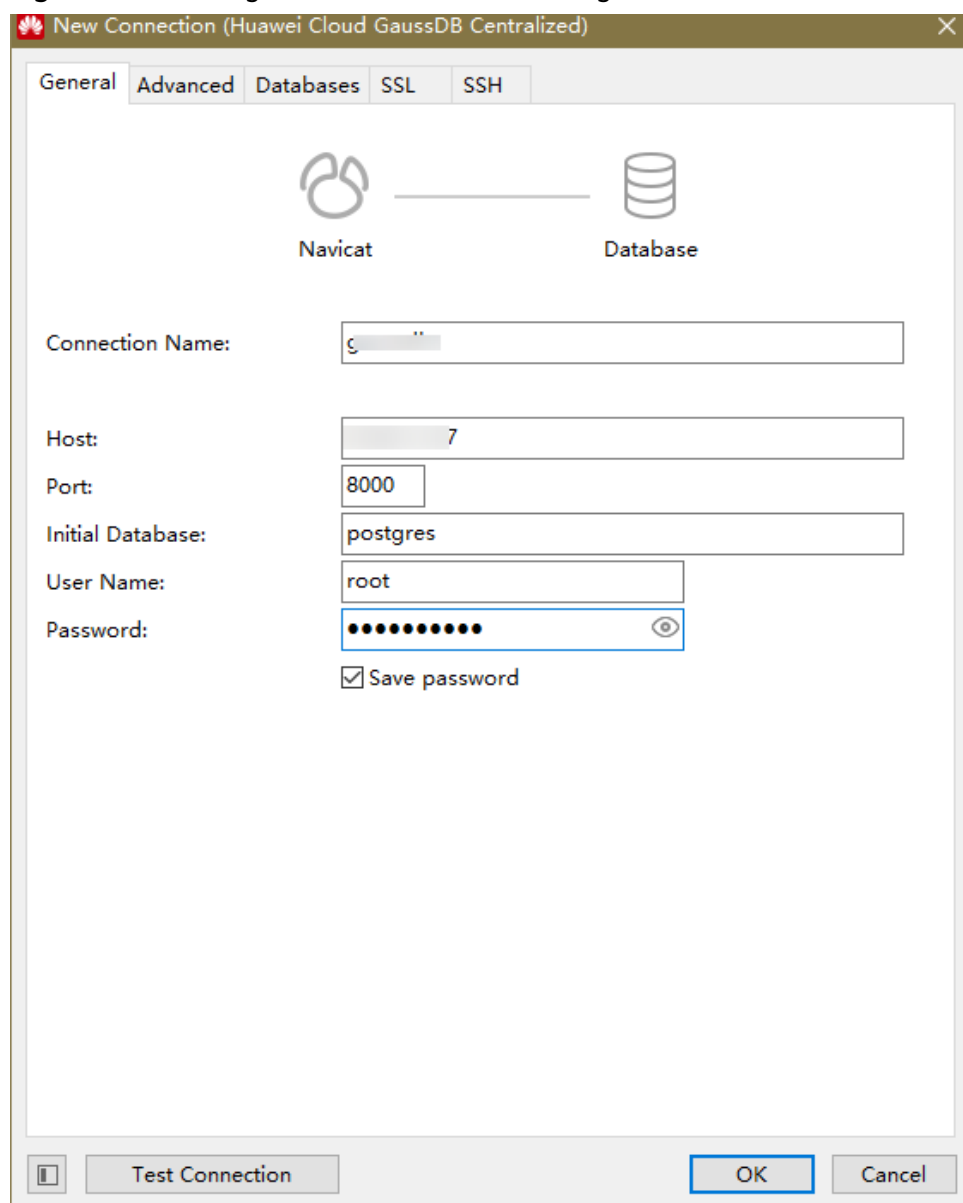
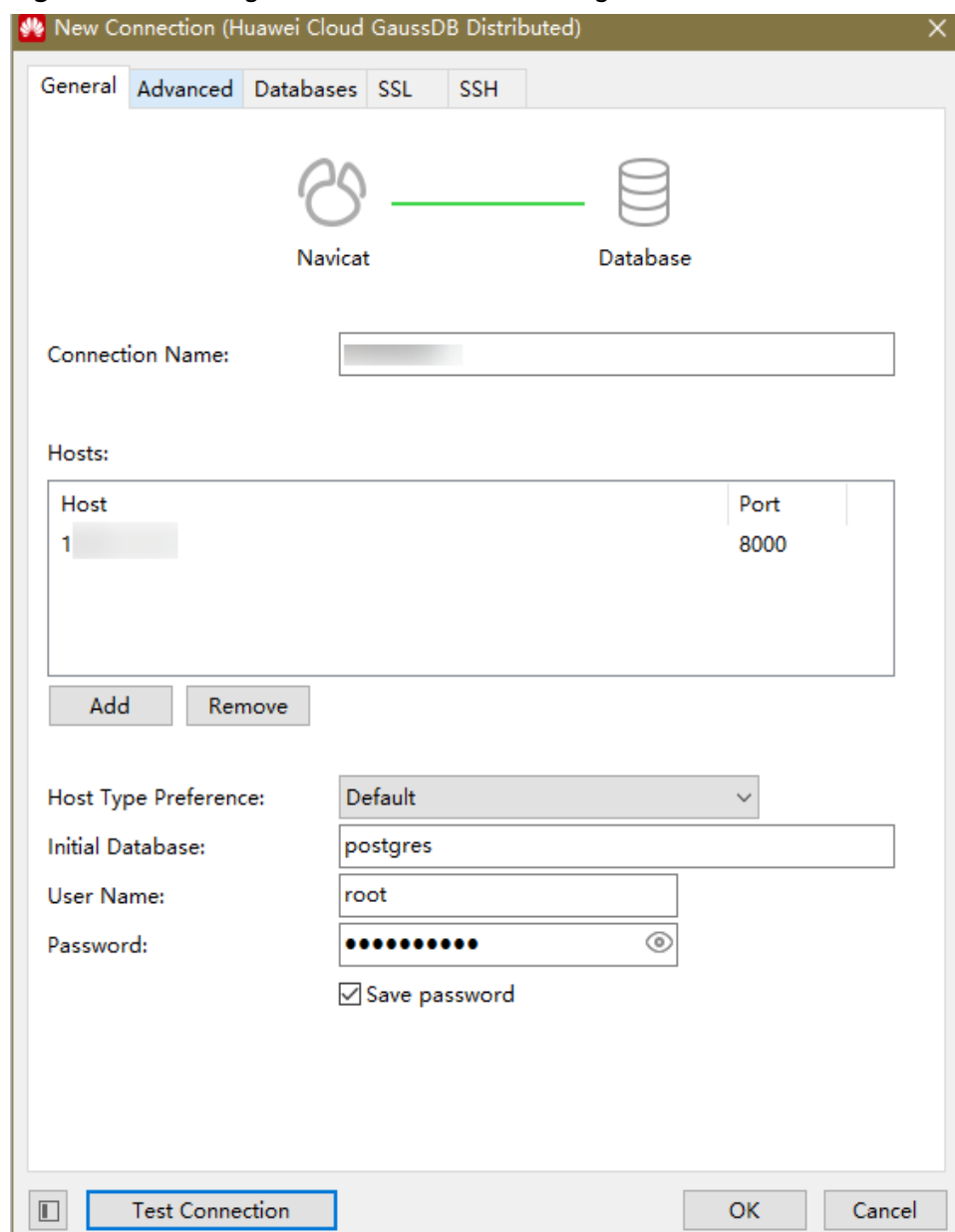


**Figure 4-12** Setting information for connecting to a distributed instance**Table 4-5** Parameters

| Parameter       | Description                          |
|-----------------|--------------------------------------|
| Connection Name | Use a name that is easy to identify. |

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Host             | Private IP address of the DB instance to be connected. To obtain the IP address, perform the following steps: <ol style="list-style-type: none"><li>1. Log in to the GaussDB management console.</li><li>2. Select the region in which the target instance is located.</li><li>3. Click the name of the target instance to enter the <b>Basic Information</b> page.</li><li>4. In the <b>Node List</b> area, view the EIP of the instance. If no EIP is bound to the instance, bind one to the instance first. For details, see <a href="#">Binding an EIP</a>.</li></ol> |
| Port             | Port of your DB instance specified during instance creation. The default port of a GaussDB instance is 8000.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Initial Database | Name of the database to be connected. After a DB instance is created, a database named <b>postgres</b> is generated by default.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| User Name        | Name of the user who will access the GaussDB instance. The default user is <b>root</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Password         | Password of the user who will access the GaussDB instance.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

**Step 3** Click **Test Connection**. If **Connection Successful** is displayed in the dialog box, the connection is normal. Click **OK** to close the dialog box.

**Step 4** Click **OK**. The connection is disabled by default after being created.

**Step 5** Right-click the connection name and choose **Open Connection** from the shortcut menu.

**Step 6** Right-click the database name and choose **Open Database** from the shortcut menu.

----End

## 4.5 Using DBeaver to Connect to an Instance

DBeaver is a multi-platform database client for you to connect to different databases using particular drivers. This section describes how to use DBeaver to connect to a GaussDB instance.

### NOTE

This section uses DBeaver 24.1.4 as an example. DBeaver's GUIs differ depending on the version you are using. To learn more about how to perform operations in different versions, refer to the product documentation specific to that version.

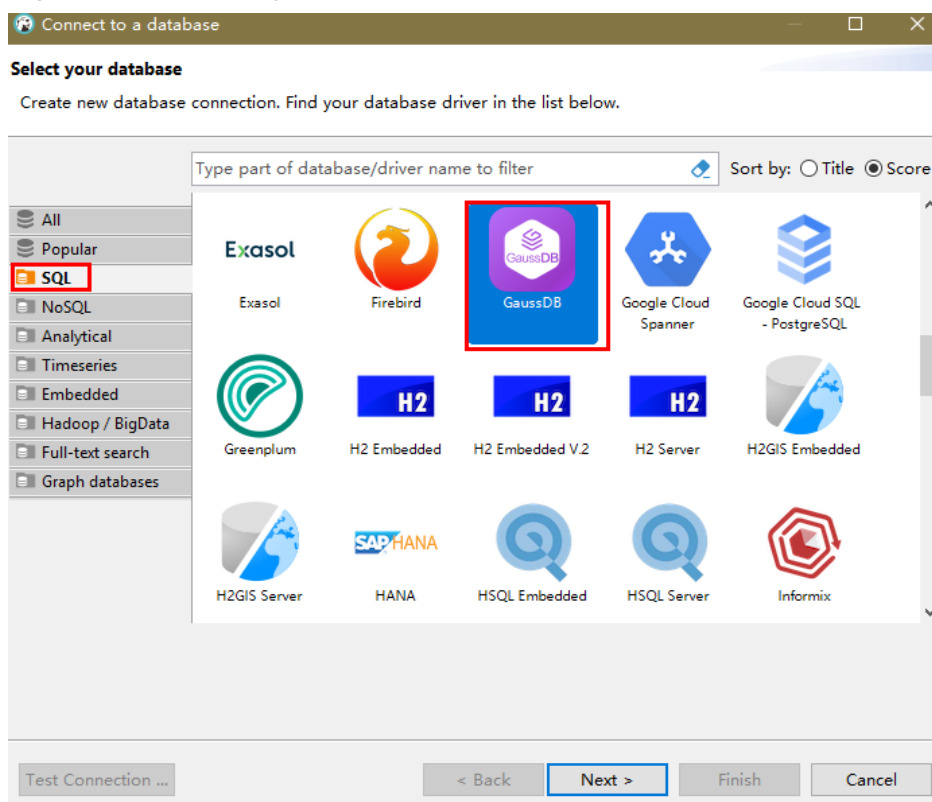
## Step 1: Obtain the DBeaver Client Installation Package

The DBeaver official website provides client installation packages for different OSs. [Download](#) the required DBeaver client installation package, and install it on the local PC.

## Step 2: Connect to the Database

1. On the DBeaver client, choose  to create a connection.
2. On the **SQL** tab, select **GaussDB** and click **Next**.

**Figure 4-13** Selecting a database



3. Enter the host IP address, port number, database name, username, and password.

**Figure 4-14** Configuring the connection

Connect to a database

**Connection Settings**

GaussDB connection settings

Main GaussDB Driver properties SSH + Network configurations...

Server

Connect by: ☒ Host ☐ URL

URL: jdbc:postgresql://localhost:8000/postgres

Host: Port: 8000

Database: postgres

Authentication

Authentication: Database Native

Username: root

Password:  ☒ Save password

[Connection variables information](#) Connection details (name, type, ...)

Driver name: GaussDB Driver Settings

Test Connection ... < Back Next > Finish Cancel

**Table 4-6** Parameters

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Host      | Private IP address of the DB instance to be connected. To obtain the IP address, perform the following steps: <ol style="list-style-type: none"><li>1. Log in to the GaussDB management console.</li><li>2. Select the region in which the target instance is located.</li><li>3. Click the name of the target instance to enter the <b>Basic Information</b> page.</li><li>4. In the <b>Node List</b> area, view the EIP of the instance.<br/>If no EIP is bound to the instance, bind one to the instance first. For details, see <a href="#">Binding an EIP</a>.</li></ol> |

| Parameter | Description                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------|
| Port      | Port of your DB instance specified during instance creation. The default port of a GaussDB instance is 8000.                    |
| Database  | Name of the database to be connected. After a DB instance is created, a database named <b>postgres</b> is generated by default. |
| Username  | Name of the user who will access the GaussDB instance. The default user is <b>root</b> .                                        |
| Password  | Password of the user who will access the GaussDB instance.                                                                      |

4. Click **Test Connection**. In the **Driver settings** dialog box, click **Download** to enable the server to automatically download driver files. If **Connected** is displayed in the dialog box, the connection is successful. Click **OK**.
- Alternatively, [obtain driver files](#) by yourself and click **Edit Driver**. Then, click the **Libraries** tab and add the local driver files to the driver editor.

**Figure 4-15** Automatically downloading driver files

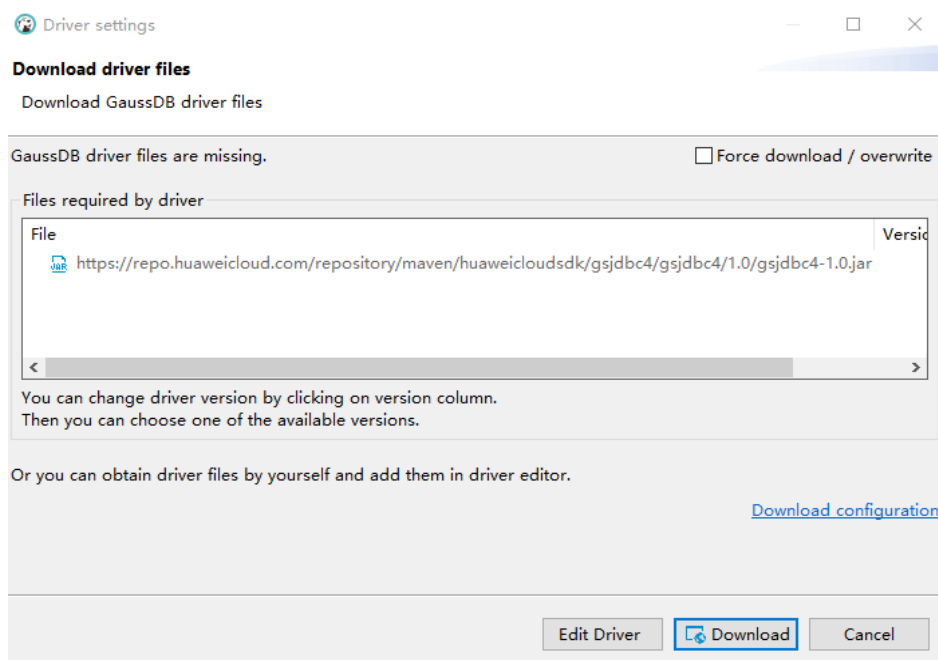
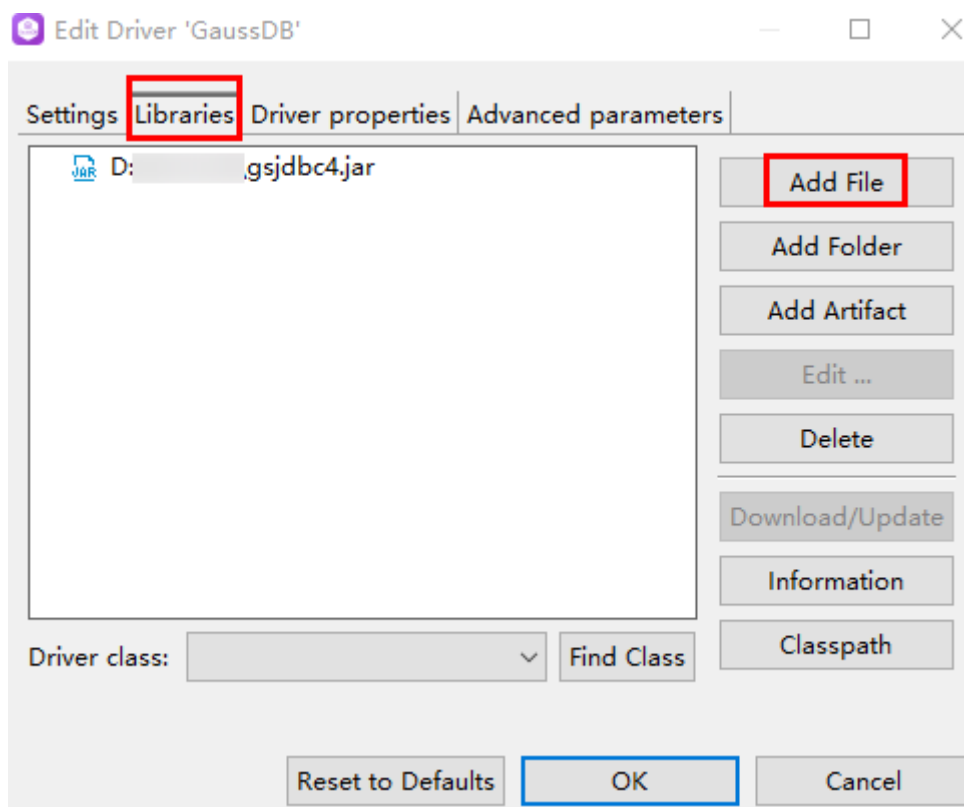


**Figure 4-16** Manually adding local driver files

5. Click **Finish** to connect to the database. You can view information about the connected database in the **Database Navigator** area.

## Related Operations

1. Obtain the driver package and its verification package.  
Download the driver package and its verification package of the relevant version to any local directory. [Table 4-7](#) lists the download list.

**Table 4-7** Driver package download list

| Version  | Download Address                                                                              |
|----------|-----------------------------------------------------------------------------------------------|
| V2.0-8.x | <a href="#">Driver package</a><br><a href="#">Verification package for the driver package</a> |
| V2.0-3.x | <a href="#">Driver package</a><br><a href="#">Verification package for the driver package</a> |
| V2.0-2.x | <a href="#">Driver package</a><br><a href="#">Verification package for the driver package</a> |

2. Verify the driver package.

To prevent the driver package from being maliciously tampered during transfer or storage, perform the following steps to verify the driver package:

- a. Press **Win+R** to open the **Run** text box. Type **cmd** in the **Open** field and press **Enter** to open the **Command Prompt** window.
- b. Run the following command to obtain the hash value of the driver package:

```
certutil -hashfile {Local directory of the driver package}\{Driver package name} sha256
```

- Replace *{Local directory of the driver package}* with the actual download path, for example, **C:\Users**.
- Replace *{Driver package name}* with the name of the downloaded driver package, for example, **GaussDB\_driver.zip**.

Example: **certutil -hashfile C:\Users\GaussDB\_driver.zip sha256**

- c. Compare the hash value obtained in [2.b](#) with the hash value of the verification package obtained in [1](#).
  - If they are consistent, the verification is successful.
  - If they are inconsistent, download the driver package again and repeat [2.a](#) to [2.c](#) to verify the driver package.

3. Extract the **gsjdbc4.jar** package from the driver package.

Decompress the driver package obtained in [1](#) to the local PC. Then, go to any OS directory in the directory of the driver package corresponding to the type of the instance to be connected, extract the **gsjdbc4.jar** package from the **GaussDB-Kernel\_Database version\_OS version\_64bit\_Jdbc.tar.gz** package, and save it to any local directory. The following is an example:

To connect to a distributed instance, go to the **GaussDB\_driver\Distributed\Euler2.5\_X86\_64** directory, find the **GaussDB-Kernel\_503.1.0.SPC2300\_Euler\_64bit\_Jdbc.tar.gz** package, and extract the **gsjdbc4.jar** package from it.

#### NOTE

The same JDBC driver package is used across different operating systems and CPU architectures. You only need to focus on the instance type when obtaining the required **gsjdbc4.jar** package.



# 5 Database Migration

---

## 5.1 Overview of GaussDB Migration Solutions

You can migrate data from MySQL, PostgreSQL, Oracle, Db2 for LUW, RDS for SQL Server, or Microsoft SQL Server databases to GaussDB, or from one GaussDB instance to another GaussDB instance.

Data migration tools include DRS, DAS, and `gs_loader`. You are advised to use DRS because it is easy to use and can complete a migration task in minutes. GaussDB migration service helps you reduce DBA labor costs, hardware costs, and data transmission costs.

Data migration tools include `gs_dump`, `gs_dumpall`, `gs_restore`, **copy** commands, and CopyManager.

**Table 5-1** GaussDB migration solutions

| Solution                            | Data Source                      | Description                                                                                                                                                                                            | Reference                                                                                                                                                                                     |
|-------------------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Using DRS to import data to GaussDB | MySQL                            | Real-time data synchronization of DRS allows you to copy data from a data source to GaussDB to implement real-time data flow of key services. It focuses on the synchronous import of tables and data. | <a href="#">Using DRS to Migrate Data from MySQL Database to GaussDB</a>                                                                                                                      |
|                                     | Oracle                           |                                                                                                                                                                                                        | <a href="#">Using DRS to Migrate Data from Oracle Database to GaussDB</a>                                                                                                                     |
|                                     | GaussDB (distributed deployment) |                                                                                                                                                                                                        | <ul style="list-style-type: none"><li>• <a href="#">From GaussDB Distributed to GaussDB Distributed</a></li><li>• <a href="#">From GaussDB Distributed to GaussDB Centralized</a></li></ul>   |
|                                     | GaussDB (centralized deployment) | For billing details, see <a href="#">DRS Billing</a> .                                                                                                                                                 | <ul style="list-style-type: none"><li>• <a href="#">From GaussDB Centralized to GaussDB Distributed</a></li><li>• <a href="#">From GaussDB Centralized to GaussDB Centralized</a></li></ul>   |
|                                     | Db2 for LUW                      |                                                                                                                                                                                                        | <ul style="list-style-type: none"><li>• <a href="#">From Db2 for LUW to GaussDB Centralized</a></li><li>• <a href="#">From Db2 for LUW to GaussDB Distributed</a></li></ul>                   |
|                                     | PostgreSQL                       |                                                                                                                                                                                                        | <ul style="list-style-type: none"><li>• <a href="#">From PostgreSQL to GaussDB Centralized</a></li><li>• <a href="#">From PostgreSQL to GaussDB Distributed</a></li></ul>                     |
|                                     | SQL Server                       |                                                                                                                                                                                                        | <ul style="list-style-type: none"><li>• <a href="#">From Microsoft SQL Server to GaussDB Centralized</a></li><li>• <a href="#">From Microsoft SQL Server to GaussDB Distributed</a></li></ul> |

| Solution                                                        | Data Source              | Description                                                                                                                                                                                                                                                                                               | Reference                                                                              |
|-----------------------------------------------------------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| Using DAS to export and import data                             | SQL/CSV files            | You can use DAS to export data from the source database first and then import the data from your local PC or OBS bucket to the destination database.                                                                                                                                                      | <a href="#">Migrating Data to GaussDB Using the Export and Import Functions of DAS</a> |
| Using the <b>copy to/from</b> command to export and import data | CSV files                | The gsql tool provides the <b>\copy</b> meta-command to import or export data. <b>\copy</b> applies only to small-scale data import in good format. It does not preprocess invalid characters or provide error tolerance. Therefore, <b>\copy</b> cannot be used in scenarios where abnormal data exists. | <a href="#">Using the copy to/from Command to Export and Import Data</a>               |
| Using CopyManager in JDBC to export and import data             | Other files or databases | When you use Java to develop applications, the CopyManager API of the JDBC driver is called to write data from files or other databases to GaussDB.                                                                                                                                                       | <a href="#">Using CopyManager in JDBC to Export and Import Data</a>                    |

| Solution                                    | Data Source                                                                                                                                                              | Description                                                                                                                                                                                                                                        | Reference                                                   |
|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| Using gs_dump and gs_dumpall to export data | <ul style="list-style-type: none"><li>• Plain-text archives</li><li>• Custom-format archives</li><li>• Directory-format archives</li><li>• TAR-format archives</li></ul> | <p>gs_dump can export a single database or its objects.</p> <p>gs_dumpall can export all databases or global objects in a cluster.</p> <p>You can use a tool to import the exported metadata to a destination database for database migration.</p> | <a href="#">Using gs_dump and gs_dumpall to Export Data</a> |

| Solution                        | Data Source       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Reference                                       |
|---------------------------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Using gs_restore to import data | SQL/TMP/TAR files | <p>During database migration, you can use gs_restore to import files exported by gs_dump to GaussDB. In this way, metadata, such as table definitions and database object definitions, can be imported. The imported data includes:</p> <ul style="list-style-type: none"><li>• Object definitions of all databases</li><li>• Object definitions of a single database</li><li>• Definitions of a single schema</li><li>• Definitions of a single table</li></ul> | <a href="#">Using gs_restore to Import Data</a> |

| Solution                       | Data Source | Description                                                                                                                                                                                                                                                                                                                                           | Reference                                      |
|--------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|
| Using gs_loader to import data | CSV files   | You can use gs_loader to import the files exported by using the <b>copy to</b> command. The gs_loader tool converts the syntax supported by control files into <b>\copy</b> syntax, then leverages the existing <b>\copy</b> function to import main data. At the same time, gs_loader logs the results of the <b>\copy</b> operations to a log file. | <a href="#">Using gs_loader to Import Data</a> |

## 5.2 Using DRS to Migrate Data from Oracle Database to GaussDB

### Scenarios

This section describes how to use real-time synchronization of DRS to migrate data from an on-premises Oracle database to Huawei Cloud GaussDB in real time. Full+incremental synchronization can ensure that data is always in sync between the source Oracle database and the destination GaussDB instance. Full synchronization is used to synchronize data. Incremental synchronization is used to synchronize data between the source and destination databases in real time.

[Step 1: Create a VPC and Security Group](#)

[Step 2: Create a GaussDB Instance](#)

[Step 3: Construct Data Before Migration](#)

[Step 4: Migrating the Database](#)

[Step 5: Verify Data After Migration](#)

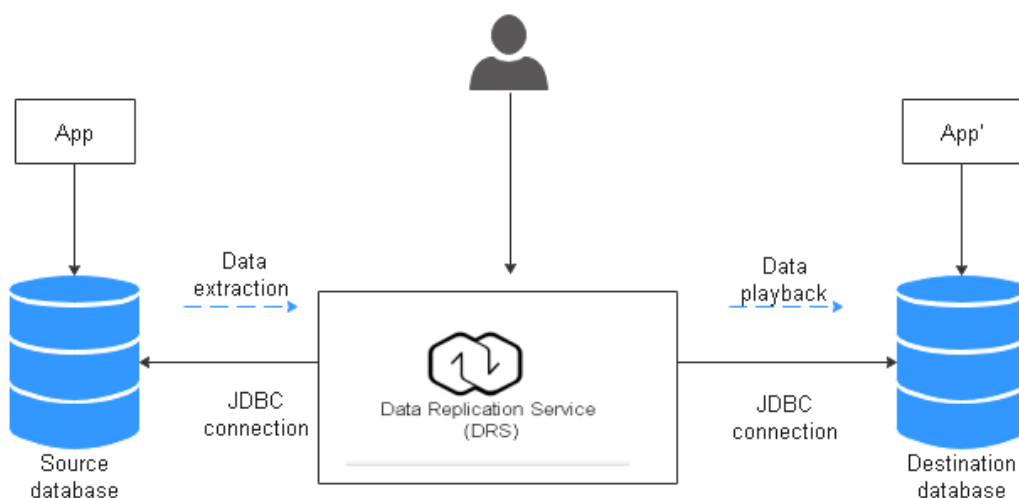
## Problems to Resolve

- Enterprise workloads have been growing and evolving fast, and traditional databases lack the scalability needed to keep up. Enterprises need distributed databases.
- Building a traditional database means purchasing and installing servers, systems, databases, and other software. The O&M is expensive and difficult.
- Traditional databases have poor performance when it comes to handling complex queries.
- It is hard for traditional databases to smoothly synchronize data with no downtime.

## Prerequisites

- You have registered with Huawei Cloud and completed account authentication.
- Your account balance is greater than or equal to \$0 USD.
- In a testing scenario, you have set up an on-premises Oracle database.
- You have obtained the IP address, port number, username, and password of the Oracle database to be migrated.

## Service Architecture

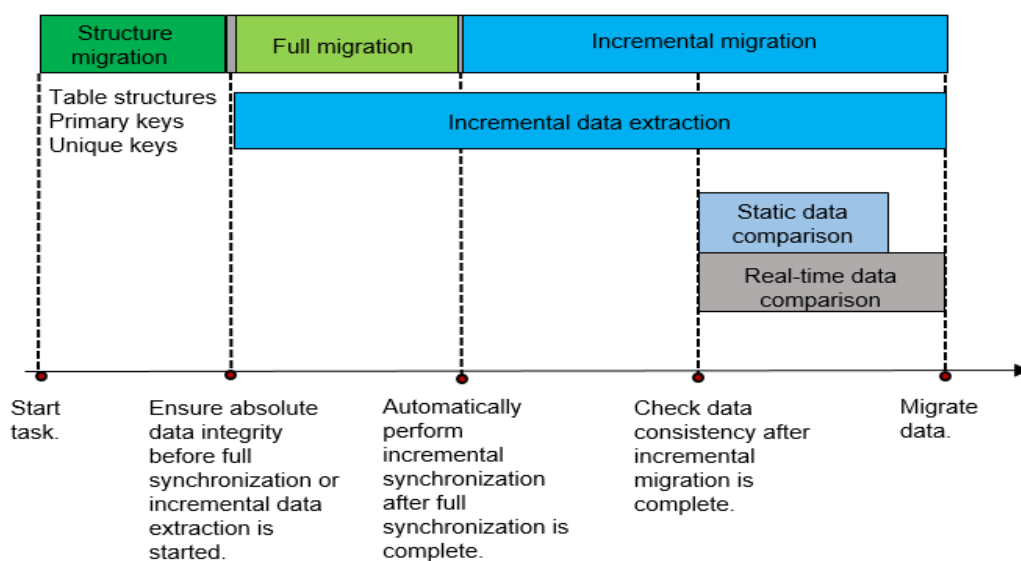


## How Data Migration Works

The data migration process is completed using full and incremental synchronization, which includes the following operations:

1. In the full synchronization phase, schemas, including tables, primary keys, and unique keys, are synchronized first.
2. After schemas are synchronized, incremental data extraction is started to ensure that the incremental data generated during full data synchronization is completely extracted to the DRS instance.

3. A full migration task is started.
4. An incremental synchronization is automatically started after the full migration is complete. The replay starts from the position where the full synchronization starts.
5. A comparison task is started after the incremental replay is complete to check the data consistency. Real-time comparison is supported.
6. Workloads synchronization is started if the data is consistent between the source and destination databases.

**Figure 5-1** Migration principle

## Resource Planning

The resource planning in this section is just an example. You need to adjust it as needed.

**Table 5-2** Resource planning

| Category | Item              | Planned Value | Remarks                                                             |
|----------|-------------------|---------------|---------------------------------------------------------------------|
| VPC      | VPC name          | vpc-src-172   | Specify a name that is easy to identify.                            |
|          | Region            | Test region   | To achieve lower network latency, select the region nearest to you. |
|          | AZ                | AZ 3          | -                                                                   |
|          | Subnet CIDR block | 172.16.0.0/16 | Select a subnet with sufficient network resources.                  |



| Category                    | Item                    | Planned Value                                    | Remarks                                                                                                                                                                                       |
|-----------------------------|-------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             | Subnet name             | subnet-src-172                                   | Specify a name that is easy to identify.                                                                                                                                                      |
| On-premises Oracle database | Name                    | orcl                                             | Specify a name that is easy to identify.                                                                                                                                                      |
|                             | Specifications          | 16 vCPUs   32 GB                                 | -                                                                                                                                                                                             |
|                             | Database version        | 11.2.0.1                                         | -                                                                                                                                                                                             |
|                             | Database user           | test_info                                        | Specify a username. The user must have the following permissions during migration: CREATE SESSION, SELECT ANY TRANSACTION, SELECT ANY TABLE, SELECT ANY DICTIONARY, and EXECUTE_CATALOG_ROLE. |
| GaussDB                     | Instance name           | Auto-drs-gaussdbv5-tar-1                         | Specify a name that is easy to identify.                                                                                                                                                      |
|                             | Database version        | GaussDB V2.0-8.103 Enterprise edition            | -                                                                                                                                                                                             |
|                             | Instance type           | Distributed (3 CNs, 3 DN shards, and 3 replicas) | In this example, a distributed instance will be created.                                                                                                                                      |
|                             | Deployment model        | Independent                                      | -                                                                                                                                                                                             |
|                             | Transaction consistency | Strong consistency                               | -                                                                                                                                                                                             |
|                             | Shards                  | 3                                                | -                                                                                                                                                                                             |
|                             | Coordinator nodes       | 3                                                | -                                                                                                                                                                                             |
|                             | Storage type            | Ultra-high I/O                                   | -                                                                                                                                                                                             |

| Category                               | Item                    | Planned Value                    | Remarks                                                                                                                                                            |
|----------------------------------------|-------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                        | AZ                      | AZ 2                             | In this example, a single AZ is select. You are advised to select multiple AZs to improve instance availability in actual use.                                     |
|                                        | Instance Specifications | Dedicated (1:8); 8 vCPUs   64 GB | Small specifications are selected for this test instance. You are advised to configure specifications based on service requirements in actual use.                 |
|                                        | Storage space           | 480 GB                           | A small storage space is selected for this test instance. You are advised to configure the storage space based on service requirements in actual use.              |
|                                        | Disk encryption         | Disable                          | In this example, disk encryption is disabled. Enabling disk encryption improves the security of data, but may slightly affect the database read/write performance. |
| Logging in to the database through DAS | Database engine         | GaussDB                          | -                                                                                                                                                                  |
|                                        | Database source         | GaussDB                          | Select the GaussDB instance created in this example.                                                                                                               |
|                                        | Database name           | postgres                         | -                                                                                                                                                                  |
|                                        | Username                | root                             | -                                                                                                                                                                  |
|                                        | Password                | -                                | Enter the password of the <b>root</b> user of the GaussDB instance created in this example.                                                                        |
| DRS migration task                     | Migration task name     | DRS-test-info                    | Specify a name that is easy to identify.                                                                                                                           |

| Category | Item                        | Planned Value      | Remarks                                                                                             |
|----------|-----------------------------|--------------------|-----------------------------------------------------------------------------------------------------|
|          | Destination database name   | test_database_info | Specify a name that is easy to identify. The name must be compatible with the Oracle database name. |
|          | Source database engine      | Oracle             | -                                                                                                   |
|          | Destination database engine | GaussDB            | -                                                                                                   |
|          | Network type                | Public network     | In this example, a public network is used.                                                          |

## Step 1: Create a VPC and Security Group

Create a VPC and security group for the GaussDB instance.

### Creating a VPC

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.
3. Click the service list icon on the left and choose **Networking > Virtual Private Cloud**. The VPC console is displayed.
4. Click **Create VPC**.

**Figure 5-2** Basic information

< | Create VPC ⓘ

Basic Information

Region



Name

vpc-9aac

IPv4 CIDR Block

192

168

0

0

/

16

 Recommended: [10.0.0.0/8-24](#) [Select](#) | [172.16.0.0/12-24](#) [Select](#) | [192.168.0.0/16-24](#) [Select](#)  
To enable communications between VPCs or between a VPC and an on-premises data center, ensure their CIDR blocks do not overlap. [Learn more about network planning](#)

Enterprise Project

--Select-- ⓘ [Create Enterprise Project](#)

Advanced Settings (Optional)

Tag: --

Description: --

Figure 5-3 Setting a subnet

Subnet Setting1

Subnet Name

subnet-9ab7

AZ

AZ1

AZ2

AZ3

AZ7

?

IPv4 CIDR Block

192

168

0

0

/

24

▼

Available IP Addresses: 251

⚠

The CIDR block cannot be modified after the subnet is created. Before creating a subnet, [plan subnet CIDR blocks](#) as required.

IPv6 CIDR Block (Optional)

☐ Enable

?

Associated Route Table

Default

?

▼ Advanced Settings (Optional)

Gateway: 192.168.0.1

DNS Server Address: 100.125.1.250,100.125.129....

Domain Name: --

NTP Server Address: --

...

Table 5-3 VPC parameters

| Parameter                          | Description                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Region                             | Region where the VPC will be created. Select the region nearest to you to ensure the lowest latency possible.                                                                                                                                                                                                                                          |
| Name                               | VPC name. The name can contain: <ul style="list-style-type: none"><li>1 to 64 characters.</li><li>Letters, digits, underscores (_), hyphens (-), and periods (.).</li></ul>                                                                                                                                                                            |
| IPv4 CIDR Block                    | IPv4 CIDR block of the VPC. Consider the following when specifying a CIDR block: <ul style="list-style-type: none"><li>IP addresses: Reserve sufficient IP addresses for subsequent service expansion.</li><li>IP address ranges: Avoid IP address conflicts if you need to connect a VPC to an on-premises data center or connect two VPCs.</li></ul> |
| Enterprise Project                 | Enterprise project that the VPC belongs to.<br>An enterprise project facilitates project-level management and grouping of cloud resources and users. The name of the default project is <b>default</b> .                                                                                                                                               |
| Advanced Settings (Optional) > Tag | Click ▼ next to <b>Advanced Settings (Optional)</b> to set this parameter.<br>Add tags to help you quickly identify, classify, and search for your VPCs.                                                                                                                                                                                               |

| Parameter                                  | Description                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Advanced Settings (Optional) > Description | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>Enter the description about the VPC in the text box as required.</p> <p>The description can contain a maximum of 255 characters and cannot contain angle brackets (&lt; or &gt;).</p> |

**Table 5-4** Subnet parameters

| Parameter   | Description                                                                                                                                                                                                                                                                                                                   |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Subnet Name | <p>Subnet name. The name can contain:</p> <ul style="list-style-type: none"><li>• 1 to 64 characters.</li><li>• Letters, digits, underscores (_), hyphens (-), and periods (.).</li></ul>                                                                                                                                     |
| AZ          | <p>A geographic location with independent power supply and network facilities in a region. AZs are physically isolated, and AZs in the same VPC are interconnected through an internal network.</p> <p>Each region contains multiple AZs. If one AZ is faulty, other AZs in the same region continue to provide services.</p> |

| Parameter                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IPv4 CIDR Block                        | <p>This parameter is displayed only in regions where IPv4/IPv6 dual stack is supported.</p> <p>A subnet is a unique CIDR block with a range of IP addresses in a VPC. Comply with the following principles when planning subnets:</p> <ul style="list-style-type: none"><li>• Planning CIDR block size: After a subnet is created, the CIDR block cannot be changed. You need to plan the CIDR block in advance based on the number of IP addresses required by your service.<ul style="list-style-type: none"><li>– The subnet CIDR block cannot be too small. Ensure that the number of available IP addresses in the subnet meets service requirements. The first and last three addresses in a subnet CIDR block are reserved for system use. For example, in subnet 10.0.0.0/24, 10.0.0.1 is the gateway address, 10.0.0.253 is the system interface address, 10.0.0.254 is used by DHCP, and 10.0.0.255 is the broadcast address.</li><li>– The subnet CIDR block cannot be too large, either. If you use a CIDR block that is too large, you may not have enough CIDR blocks available for new subnets, which can be a problem when you want to scale out services.</li></ul></li><li>• Avoiding subnet CIDR block conflicts: If you need to connect two VPCs or connect a VPC to an on-premises data center, there cannot be any CIDR block conflicts. If the subnet CIDR blocks at both ends of the network conflict, create a subnet. For details, see <a href="#">Creating a Subnet for an Existing VPC</a>.</li></ul> <p>A subnet mask can be between the netmask of its VPC CIDR block and /29 netmask. If a VPC CIDR block is 10.0.0.0/16, its subnet mask can be between 16 and 29.</p> <p>For more information about VPC subnet planning, see <a href="#">VPC and Subnet Planning</a>.</p> |
| IPv6 CIDR Block (Optional)             | <p>This parameter is displayed only in regions where IPv4/IPv6 dual stack is supported.</p> <p>If you enable this function, the system automatically assigns an IPv6 CIDR block to the subnet. Currently, the IPv6 CIDR block cannot be customized. Once enabled, this function cannot be disabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Advanced Settings (Optional) > Gateway | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>Retain the default value unless there are special requirements.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Parameter                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Advanced Settings (Optional) > DNS Server Address | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>Huawei Cloud private DNS server addresses are entered by default. This allows ECSs in a VPC to communicate with each other. You can also access cloud services directly through private DNS servers.</p> <p>You can change the default DNS server address if needed. If you delete the default DNS server addresses, you may fail to access other services on the cloud. Exercise caution when performing this operation.</p> <p>You can click <b>Reset</b> on the right to restore DNS server addresses to the default value.</p> <p>A maximum of two DNS server IP addresses are allowed. Separate them by a comma (,).</p>                                                                                                                                                                                                                                                                                                                                                                      |
| Advanced Settings (Optional) > Domain Name        | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>Enter domain names separated by spaces. A maximum of 254 characters are allowed. Each label in a domain name can contain a maximum of 63 characters. (For example, <b>test</b> and <b>com</b> are two labels in <b>test.com</b>.)</p> <p>To access a domain name, simply enter the domain name prefix. ECSs in the subnet automatically match the configured domain name suffix.</p> <p>If the domain names are changed, ECSs newly added to this subnet will use the new domain names.</p> <p>If an existing ECS in this subnet needs to use the new domain names, restart the ECS or run a command to restart the DHCP Client service or network service.</p> <p><b>NOTE</b></p> <p>The command for updating the DHCP configuration depends on the ECS OS. The commands here are only for your reference.</p> <ul style="list-style-type: none"><li>Restart the DHCP Client service: <b>service dhcpcd restart</b></li><li>Restart the network service: <b>service network restart</b></li></ul> |

| Parameter                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Advanced Settings (Optional) > NTP Server Address   | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>If you want to add NTP server addresses for a subnet, you can specify this parameter. The IP addresses are added in addition to the default NTP server addresses.</p> <ul style="list-style-type: none"><li>• If you add or change the NTP server addresses of a subnet, you need to renew the DHCP lease for or restart all the ECSs in the subnet to make the change take effect immediately.</li><li>• If the NTP server addresses have been cleared out, restarting the ECSs will not help. You must renew the DHCP lease for all ECSs to make the change take effect immediately.</li></ul>                                                                                                                                                                                                                                                                                                                       |
| Advanced Settings (Optional) > IPv4 DHCP Lease Time | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>This parameter is displayed only in regions where IPv4/IPv6 dual stack is supported.</p> <p>You can set the DHCP lease time of an IPv4 address.</p> <p>This parameter specifies the period during which a client can use an IP address automatically assigned by the DHCP server. After the lease time expires, a new IP address will be assigned to the client.</p> <ul style="list-style-type: none"><li>• <b>Limited:</b> Set the DHCP lease time. The unit can be days or hours.</li><li>• <b>Unlimited:</b> The DHCP lease time does not expire.</li></ul> <p>If the time period is changed, the new lease time takes effect when the instance (such as an ECS) in the subnet is renewed next time. You can wait for the instance to be renewed automatically or manually modify the lease time. If you want the new lease time to take effect immediately, manually renew the lease or restart the instance.</p> |
| Advanced Settings > IPv6 DHCP Lease Time            | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>This parameter is displayed for regions where IPv4/IPv6 dual stack is supported and <b>IPv6</b> is enabled.</p> <p>You can set the DHCP lease time of an IPv6 address in the same way as how you do with an IPv4 address.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Advanced Settings (Optional) > Tag                  | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>You can add tags to help you quickly identify, classify, and search for your subnets.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |



| Parameter                                  | Description                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Advanced Settings (Optional) > Description | <p>Click  next to <b>Advanced Settings (Optional)</b> to set this parameter.</p> <p>You can enter the subnet description in the text box as required.</p> <p>The description can contain a maximum of 255 characters and cannot contain angle brackets (&lt; or &gt;).</p> |

5. Configure parameters as needed and click **Create Now**.
6. Return to the VPC list and check whether the VPC is created.  
If the VPC status becomes available, the VPC has been created.

## Creating a Security Group

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.
3. Click the service list icon on the left and choose **Networking > Virtual Private Cloud**. The VPC console is displayed.
4. In the navigation pane, choose **Access Control > Security Groups**.
5. Click **Create Security Group**.
6. Specify a security group name and other information.

**Figure 5-4** Basic information

< | Create Security Group

Summary

Region   
Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

Name

Enterprise Project   [Create Enterprise Project](#)

Tag (Optional) TMS's predefined tags are recommended for adding the same tag to different cloud resources. [Create predefined tags](#)  
[+ Add Tag](#)  
You can add 20 more tags.

Description (Optional)   
0/255

**Table 5-5** Parameter description

| Parameter          | Description                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Region             | Region where the VPC will be created. Select the region nearest to you to ensure the lowest latency possible.<br>An instance must be in the same region as its associated security group.                                                                                                                                                         |
| Name               | The name of the security group. The name can contain: <ul style="list-style-type: none"><li>1 to 64 characters.</li><li>Letters, digits, underscores (_), hyphens (-), and periods (.).</li></ul> You can change the security group name after a security group is created. It is recommended that you give each security group a different name. |
| Enterprise Project | When creating a security group, you can add the security group to an enabled enterprise project.<br>An enterprise project facilitates project-level management and grouping of cloud resources and users. The name of the default project is <b>default</b> .                                                                                     |
| Tag                | You can add tags to help you identify, classify, and search for your security groups.                                                                                                                                                                                                                                                             |
| Description        | The description of the security group.<br>The description can contain a maximum of 255 characters and cannot contain angle brackets (< or >).                                                                                                                                                                                                     |

- Click the **Inbound Rules** tab and then click **Add Rule**.
- Configure an inbound rule, add the IP address of the source database, and click **OK**.

**Figure 5-5** Configuring an inbound rule

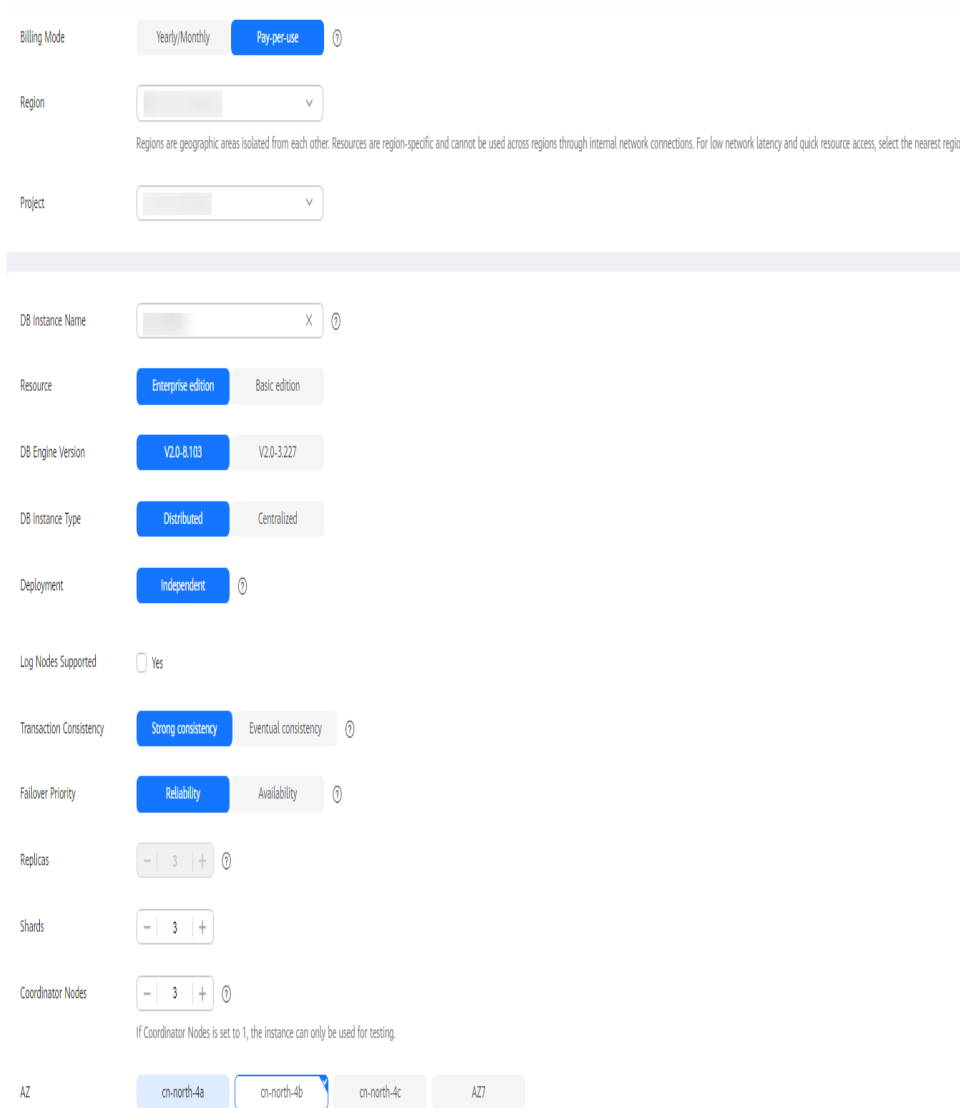
The screenshot displays the 'Inbound Rules' configuration page. At the top, there are tabs for 'Inbound Rules' and 'Outbound Rules'. Below the tabs are two buttons: 'Add Rule' and 'Fast-Add Rule'. The main area contains a table of existing rules with columns: Priority, Action, Type, Protocol & Port, Source, Description, and Operation. The table lists three rules: 1. Priority 1, Action Allow, Type IPv4, Protocol & Port Protocol / TCP (Custom ports), Source IP address, Description (empty), Operation Replicate Delete. 2. Priority 1, Action Allow, Type IPv4, Protocol & Port Protocol / All, Source Security group, Description Allows the instances in the security group to, Operation Delete. 3. Priority 1, Action Allow, Type IPv6, Protocol & Port Protocol / All, Source Security group, Description Allows the instances in the security group to, Operation Delete. Below the table, there is a 'Create Now' button.

- Configure parameters as needed and click **Create Now**.

## Step 2: Create a GaussDB Instance

Create a GaussDB instance as the destination database of the migration task.

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.
3. Click the service list icon on the left and choose **Databases > GaussDB**.
4. In the navigation pane on the left, choose **GaussDB > Instances**.
5. Click **Buy DB Instance**.
6. On the page shown in [Figure 5-6](#), configure basic information about the instance, including the instance name, billing mode, edition type, DB engine version, instance type, transaction consistency, number of shards, number of coordinator nodes, and deployment AZ.

**Figure 5-6** Basic information

Billing Mode: ☐ Yearly/Monthly ☒ Pay-per-use ⓘ

Region:  ⓘ  
Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

Project:  ⓘ

---

DB Instance Name:  X ⓘ

Resource: ☒ Enterprise edition ☐ Basic edition

DB Engine Version: ☒ V2.0-8.103 ☐ V2.0-3.227

DB Instance Type: ☒ Distributed ☐ Centralized

Deployment: ☒ Independent ⓘ

Log Nodes Supported: ☐ Yes

Transaction Consistency: ☒ Strong consistency ☐ Eventual consistency ⓘ

Failover Priority: ☒ Reliability ☐ Availability ⓘ

Replicas:  - 3 + ⓘ

Shards:  - 3 +

Coordinator Nodes:  - 3 + ⓘ  
If Coordinator Nodes is set to 1, the instance can only be used for testing.

AZ: ☐ cn-north-4a ☒ cn-north-4b ☐ cn-north-4c ☐ AZ7

7. Select the instance specifications and storage space.

**Figure 5-7** Instance specifications

Instance Specifications **Dedicated(1:8)** ⓘ

Flavor Name

☐ 4 vCPUs | 32 GB Unavailable for production environment

☒ 8 vCPUs | 64 GB

☐ 16 vCPUs | 128 GB

☐ 32 vCPUs | 256 GB

☐ 64 vCPUs | 512 GB

DB Instance Specifications **Dedicated(1:8) | 8 vCPUs | 64 GB**

Storage Type **Ultra-high I/O** [Learn more about storage types.](#)

Storage Space (GB) **480**

120 14480 28840 43200 72000

GaussDB provides free backup storage equal to the amount of your purchased storage space. After the free backup space is used up, charges are applied based on the [backup space pricing details](#).

Disk Encryption **Disable** **Enable** ⓘ

8. Select the VPC created in [Creating a VPC](#) and security group created in [Creating a Security Group](#) for the instance and configure the database port.

**Figure 5-8** Selecting a VPC and security group

Relationship among VPCs, subnets, security groups, and DB instances. ⓘ

VPC **default\_vpc** **default\_subnet** ⓘ

If you want to create a VPC, go to [the VPC console](#).

Security Group **default** [View Security Group](#) ⓘ

In a security group, rules that authorize connections to DB instances apply to all DB instances associated with the security group.

Ensure that the TCP ports in the inbound rule of the selected security group contain 8000-8100, 20050, 5000-5001, 2379-2380, 6000, 6500, 40000-60480.

Security Group Rules **Add Inbound Rule**

Database Port **Default port: 8000**

9. Configure the password and other information.

**Figure 5-9** Configuring the password and other information

Administrator **root**

Administrator Password **\*\*\*\*\*** ⓘ Keep your password secure. The system cannot retrieve your password.

Confirm Password **\*\*\*\*\*** ⓘ

Parameter Template **Default-Enterprise-Edition-GaussDB-8.10...** [View Parameter Template](#)

Enterprise Project **default** [View Enterprise Projects](#) ⓘ

Tag **+ Add Tag** ⓘ TMS's predefined tags are recommended for adding the same tag to different cloud resources. [Create predefined tags](#) ⓘ

You can add 20 more tags.

10. Click **Next**, confirm the information, and click **Submit**.

11. Go to the instance list.

If status of the instance becomes **Available**, the instance has been created.

### Step 3: Construct Data Before Migration

Before the migration, prepare some data types in the source database for verification after the migration is complete. The end-to-end test data in this section is for reference only.

The following table lists data types supported by DRS.

**Table 5-6** Data type mapping

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key) | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key)                    | Comparison (Source Data Type as Non-Primary Key)                | Remarks                                                                                             |
|------------------|-----------------------|----------------------------------------|--------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| CHAR             | character             | Supported                              | Supported                                  | Supported. The spaces before and after a character are ignored. | Supported. The spaces before and after a character are ignored. | -                                                                                                   |
| VARCHAR          | character varying     | Supported                              | Supported                                  | Supported                                                       | Supported                                                       | The precision ranges of the source and destination databases are different, causing precision loss. |
| VARCHAR2         | character varying     | Supported                              | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key)                                                           | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key)                    | Comparison (Source Data Type as Non-Primary Key)                | Remarks                                                                                             |
|------------------|-----------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| NCHAR            | character             | Supported                                                                                        | Supported                                  | Supported. The spaces before and after a character are ignored. | Supported. The spaces before and after a character are ignored. | -                                                                                                   |
| NVARCHAR2        | nvarchar2             | Supported                                                                                        | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |
| NUMBER           | numeric               | Supported                                                                                        | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |
| NUMBER (6,3)     | numeric(6,3)          | Supported                                                                                        | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |
| NUMBER (6,0)     | Integer               | Supported                                                                                        | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |
| NUMBER (3)       | smallint              | Supported                                                                                        | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |
| NUMBER (6,-2)    | integer               | Supported                                                                                        | Supported                                  | Supported                                                       | Supported                                                       | -                                                                                                   |
| BINARY_FLOAT     | real                  | Not supported (The destination database does not support creating tables using the primary key.) | Supported                                  | Not supported                                                   | Supported                                                       | The precision ranges of the source and destination databases are different, causing precision loss. |

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key)                                                           | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks                                                                                             |
|------------------|-----------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| BINARY_DOUBLE    | double precision      | Not supported (The destination database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Supported                                        | -                                                                                                   |
| FLOAT            | real                  | Not supported (The destination database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Supported                                        | The precision ranges of the source and destination databases are different, causing precision loss. |
| INT              | numeric               | Supported                                                                                        | Supported                                  | Supported                                    | Supported                                        | -                                                                                                   |
| INTEGER          | numeric               | Supported                                                                                        | Supported                                  | Supported                                    | Supported                                        | -                                                                                                   |

| Source Data Type | Destination Data Type          | Sync (Source Data Type as Primary Key) | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks                                                                                                                                                                                                                                        |
|------------------|--------------------------------|----------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DATE             | date                           | Supported                              | Supported                                  | Not supported                                | Supported                                        | If a table with the date type is created in the destination database, the data type precision range in the source database is different from that in the destination database, causing precision loss. Therefore, comparison is not supported. |
| TIMESTAMP        | timestamp(6) without time zone | Supported                              | Supported                                  | Not supported                                | The value is accurate to six decimal places.     | Restrictions on the source database: The maximum precision supported by the source database is 6.                                                                                                                                              |



| Source Data Type | Destination Data Type       | Sync (Source Data Type as Primary Key)                                                           | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks |
|------------------|-----------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|---------|
| TIMESTAMP_TZ     | timestamp(6) with time zone | Not supported (The source database does not support creating tables using the primary key.)      | Supported                                  | Not supported                                | Filter out this column.                          | -       |
| TIMESTAMP_LTZ    | timestamp(6) with time zone | Not supported (The destination database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Filter out this column.                          | -       |

| Source Data Type | Destination Data Type  | Sync (Source Data Type as Primary Key) | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks                                                                                                                                                   |
|------------------|------------------------|----------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| INTERVAL_YM      | interval year to month | Supported                              | Supported                                  | Not supported                                | Not supported                                    | Incremental synchronization does not support this type.                                                                                                   |
| INTERVAL_DS      | interval day to second | Supported                              | Supported                                  | Not supported                                | Not supported                                    | Incremental synchronization does not support this type. Restrictions on the source database: The maximum precision supported by the source database is 6. |

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key)                                                      | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks |
|------------------|-----------------------|---------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|---------|
| BLOB             | bytea                 | Not supported (The source database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Filter out this column.                          | -       |
| CLOB             | text                  | Not supported (The source database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Filter out this column.                          | -       |

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key)                                                      | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks |
|------------------|-----------------------|---------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|---------|
| NCLOB            | text                  | Not supported (The source database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Filter out this column.                          | -       |
| LONG             | text                  | Not supported (The source database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Filter out this column.                          | -       |

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key)                                                           | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks |
|------------------|-----------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|---------|
| LONG_RAW         | bytea                 | Not supported (The source database does not support creating tables using the primary key.)      | Supported                                  | Not supported                                | Filter out this column.                          | -       |
| RAW              | bytea                 | Not supported (The destination database does not support creating tables using the primary key.) | Supported                                  | Not supported                                | Supported                                        | -       |
| RowID            | character varying(18) | Supported                                                                                        | Supported                                  | Supported                                    | Supported                                        | -       |

| Source Data Type | Destination Data Type | Sync (Source Data Type as Primary Key) | Sync (Source Data Type as Non-Primary Key) | Comparison (Source Data Type as Primary Key) | Comparison (Source Data Type as Non-Primary Key) | Remarks                                                                      |
|------------------|-----------------------|----------------------------------------|--------------------------------------------|----------------------------------------------|--------------------------------------------------|------------------------------------------------------------------------------|
| BFILE            | -                     | Not supported                          | Not supported                              | Not supported                                | Not supported                                    | Restrictions on the source database: The BFILE type is not supported.        |
| XMLTYPE          | -                     | Not supported                          | Not supported                              | Not supported                                | Not supported                                    | Restrictions on the source database: The XMLTYPE type is not supported.      |
| UROWID           | -                     | Not supported                          | Not supported                              | Not supported                                | Not supported                                    | Full and incremental synchronizations are not supported.                     |
| sdo_geometry     | -                     | Not supported                          | Not supported                              | Not supported                                | Not supported                                    | Restrictions on the source database: The sdo_geometry type is not supported. |
| NUMBER(*, 0)     | numeric               | Supported                              | Supported                                  | Supported                                    | Supported                                        | -                                                                            |

Perform the following steps to construct data in the source database:

1. Use a database connection tool to connect to the source Oracle database based on its IP address.
2. Construct data in the source database based on data types supported by DRS.

- a. Create a test user.

```
create user test_info identified by xxx;
```

*test\_info* indicates the user created for the migration, and *xxx* indicates the password of the user.

- b. Assign privileges to the user.

```
grant dba to test_info;
```

- c. Create a data table for the user.

```
CREATE TABLE test_info.DATATYPELIST(
 ID INT,
 COL_01_CHAR_____E CHAR(100),
 COL_02_NCHAR_____E NCHAR(100),
 COL_03_VARCHAR___E VARCHAR(1000),
 COL_04_VARCHAR2___E VARCHAR2(1000),
 COL_05_NVARCHAR2_E NVARCHAR2(1000),
 COL_06_NUMBER_____E NUMBER(38,0),
 COL_07_FLOAT_____E FLOAT(126),
 COL_08_BFLOAT_____E BINARY_FLOAT,
 COL_09_BDOUBLE_____E BINARY_DOUBLE,
 COL_10_DATE_____E DATE DEFAULT SYSTIMESTAMP,
 COL_11_TS_____E TIMESTAMP(6),
 COL_12_TSTZ_____E TIMESTAMP(6) WITH TIME ZONE,
 COL_13_TSLTZ_____E TIMESTAMP(6) WITH LOCAL TIME ZONE,
 COL_14_CLOB_____E CLOB DEFAULT EMPTY_CLOB(),
 COL_15_BLOB_____E BLOB DEFAULT EMPTY_BLOB(),
 COL_16_NCLOB_____E NCLOB DEFAULT EMPTY_CLOB(),
 COL_17_RAW_____E RAW(1000),
 COL_19_LONGRAW___E LONG RAW,
 COL_24_ROWID_____E ROWID,
 PRIMARY KEY(ID)
);
```

- d. Insert two rows of data.

```
insert into test_info.DATATYPELIST
values(4,'huawei','xian','shanxi','zhongguo','shijie',
666,12.321,1.123,2.123,sysdate,sysdate,sysdate,sysdate,'hw','cb','df','F'
F','FF','AAAYEVAJAAAACrAAA');
```

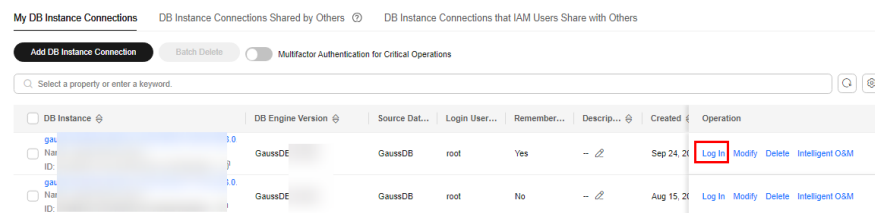
```
insert into test_info.DATATYPELIST values(2,'Migrate-
test','test1','test2','test3','test4',
666,12.321,1.123,2.123,sysdate,sysdate,sysdate,sysdate,'hw','cb','df','F'
F','FF','AAAYEVAJAAAACrAAA');
```

- e. Commit the changes to the database.

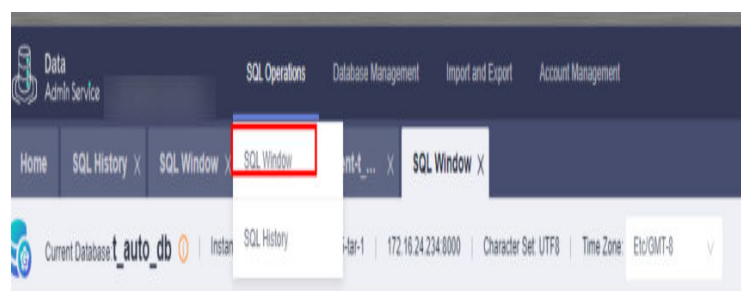
```
commit;
```

3. Create a database in the destination GaussDB instance.

- a. Log in to the [Huawei Cloud console](#).
- b. Click  in the upper left corner and select a region.
- c. Click the service list icon on the left and choose **Databases > Data Admin Service**.
- d. In the navigation pane on the left, choose **Development Tool** to go to the login list page.
- e. Click **Add Login**.
- f. On the displayed page, select the DB engine, source database, and target instance, enter the login username, password, and description (optional), and enable **Show Executed SQL Statements**.
- g. Click **Test Connection** to check whether the connection is successful.  
If a message is displayed indicating the connection is successful, continue with the operation. If a message is displayed indicating the connection failed and the failure cause is provided, make modifications according to the error message.
- h. Click **OK**.
- i. Locate the added record and click **Log In** in the **Operation** column.

**Figure 5-10** Logging in to an instance

- j. Choose **SQL Operations > SQL Window** on the top menu bar.



- k. Run the following statement to create an Oracle-compatible database:  
**test\_database\_info** indicates the database name. Replace it as required.  
`CREATE DATABASE test_database_info DBCOMPATIBILITY 'ORA';`

## Step 4: Migrating the Database

Create a DRS instance and migrate data from the **test\_info** database in the on-premises Oracle database to the **test\_database\_info** database in the GaussDB instance.



## Performing a Pre-migration Check

Before creating a migration task, check the migration conditions to ensure smooth migration.

Before the migration, you need to obtain the [notes on migration to the cloud](#).

## Creating a Migration Task

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.  
Select the region where the destination instance is deployed.
3. Click the service list icon on the left and choose **Databases > Data Replication Service**.
4. In the navigation pane on the left, choose **Data Synchronization Management**. On the displayed page, click **Create Synchronization Task**.
5. Configure synchronization instance information.
  - a. Select a region, billing mode, and project, and enter a task name.

Billing Mode: Yearly/Monthly, Pay-per-use

Region: [dropdown]

Regions are geographic areas isolated from each other. For low network latency and quick resource access, select the nearest region.

Project: [dropdown]

★ Task Name: DRS-2843

Description: [text area]

0/256

- b. Configure synchronization instance details. Specifically, specify **Data Flow**, **Source DB Engine**, **Destination DB Engine**, **Network Type**, **DRS Task Type**, **Destination DB Instance**, **Synchronization Instance Subnet**, **Synchronization Mode**, **Specify EIP** (mandatory when **Network Type** is set to **Public Network**), **Specifications**, **AZ**, **Enterprise Project**, and **Tags** (optional).

Synchronization Instance Details

The following information cannot be modified after you go to the next page.

• Data Flow: In the cloud, Out of the cloud, Self-built to self-built

• Source DB Engine: MySQL, Oracle, DB2 for LUW, CDC, MariaDB, MongoDB, PostgreSQL, Microsoft SQL Server, GaussDB(for MySQL), TDB

• Destination DB Engine: MySQL, GaussDB(DWS), GaussDB Distributed, GaussDB Primary/Standby, MariaDB, PostgreSQL, GaussDB(for MySQL)

• Network Type: Public network

• DRS Task Type: Single-AZ, Dual-AZ

• Destination DB Instance: No DB instance available. View DB Instance, View Unavailable DB Instance

• Synchronization Instance Subnet: Select the subnet. View Subnets, View Occupied IP Address

• Security Group: default

• Synchronization Mode: Full, Incremental

• Specify EIP: [input field], Create an EIP

The screenshot shows the configuration page for a GaussDB instance. It includes sections for Specifications (Micro, Small, Medium, Large, Ultra-large), AZ (az1, az2, az3, az7), Enterprise Project (a dropdown menu), and Tags (a text input field and an 'Add' button). The 'Medium' specification is selected, and 'az1' is selected for the AZ. The Enterprise Project is set to 'Selected'. The Tags section has a text input field and an 'Add' button.

- c. Click **Create Now**.
6. Configure the source and destination database information.
  - a. Configure **DNS Server** as required. Specify connection information about the source database, including the IP address, port, username, and password.

Click **Test Connection**.

The screenshot shows the 'Configure Your Own DNS Server' configuration page. It includes a 'DNS Server' toggle switch, a 'Source Database' section with a 'Select Connection' dropdown, and fields for 'IP Address or Domain Name', 'Port', 'Database Service Name', 'PDB Name', 'Database Username', 'Database Password', 'SSL Connection', and 'Encryption Certificate'. The 'DNS Server' toggle is turned on. The 'Source Database' section has a 'Select Connection' dropdown. The 'IP Address or Domain Name' field is empty. The 'Port' field is empty. The 'Database Service Name' field is empty. The 'PDB Name' field is empty. The 'Database Username' field is empty. The 'Database Password' field is empty. The 'SSL Connection' toggle is turned on. The 'Encryption Certificate' field is empty. The 'Test Connection' button is disabled.

- b. Enter the username and password of the destination database.  
Click **Test Connection**.

### Destination Database

DB Instance Name

Database Username

Database Password

This button is available only after the replication instance is created successfully.

- c. Click **Next**. In the displayed box, read the message carefully and click **Agree**.
7. Configure the synchronization task.
  - a. Select the databases and tables of the source database to be migrated. In this example, select the **DATATYPELIST** table from the **test\_info** database.

**Basic Information**

|                           |                                      |                              |                    |
|---------------------------|--------------------------------------|------------------------------|--------------------|
| Task ID                   | 5b99e983-f78a-42c9-aa73-ba921d11b20r | Task Name                    | DRS-test-info      |
| Created                   | Dec 30, 2021 16:50:36 GMT+08:00      | Source Database IP           | 10.154.219.69      |
| Destination Database Name | Auto-drs-gaussdbv5-tar-1             | Destination Database IP Port | 172.16.24.234:8000 |

Flow Control   ⓘ

Synchronization Object

Only some DDL statements can be synchronized. For details, see precautions of the current scenario in Real-Time Synchronization > Before You Start. After objects are synchronized, they will be saved in the destination database with their names in all lowercase. If any data in the source database changes, click the refresh button below. Move objects to be migrated from list of unselected objects on left side to the list of selected objects on right side.

☒ test

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

- b. Locate the database and table, respectively, and click **Edit** to change the database name and table name as needed.

☒ test\_info

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☐ database

☒ database

☒ TEST\_INFO

☐ DATATYPELIST

- c. On the displayed dialog box, enter a new name, for example, **DATATYPELIST\_After**.  
The name cannot include special characters. Otherwise, an error will be reported during SQL statement execution after the migration.

## New Table Name

The new table will be used in the destination database.

Edit Table Name

**OK** Cancel

- d. Confirm the settings and click **Next**.

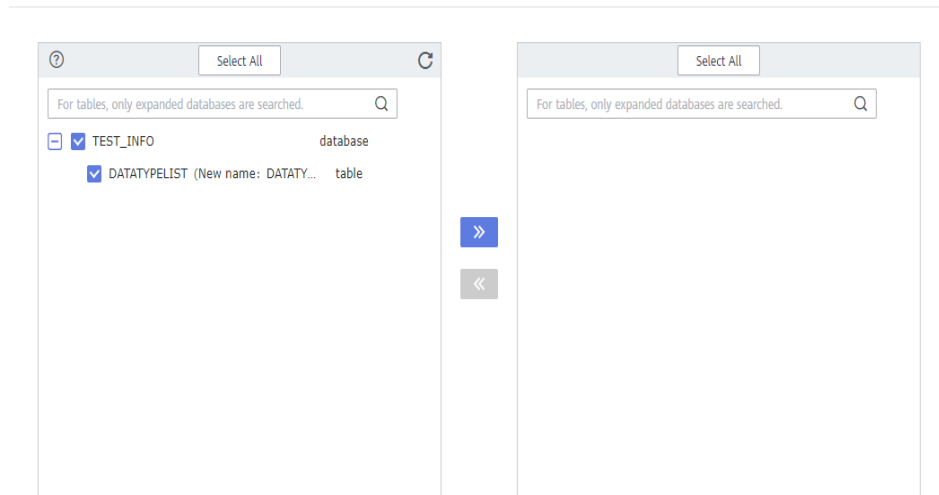
8. Confirm advanced settings.

The information on the **Advanced Settings** page is for confirmation only and cannot be modified. After confirming the information, click **Next**.

9. Process data.

On this page, you can process the table to be migrated. For example, you can select the column to be migrated and change its name. In this example, change the column name **COL\_01\_CHAR\_\_\_\_\_E** to **new-line**.

- a. Click **Edit** next to the table to be processed.



- b. Edit the **COL\_01\_CHAR\_\_\_\_\_E** column.



- c. Enter the new name **new-line** and click **Confirm**.

Database Name: SYNC\_FULL\_INCR\_VPC\_001 Table Name: COMMON

Enter a column name

| Column Name                                | New Column Name | Type   | Constraint Type |
|--------------------------------------------|-----------------|--------|-----------------|
| <input checked="" type="checkbox"/> ID     |                 | NUMBER | Primary Key     |
| <input checked="" type="checkbox"/> COL_03 | new_line        | NUMBER |                 |
| <input checked="" type="checkbox"/> COL_04 |                 | NUMBER |                 |
| <input checked="" type="checkbox"/> COL_06 |                 | NUMBER |                 |
| <input checked="" type="checkbox"/> COL_07 |                 | NUMBER |                 |
| <input checked="" type="checkbox"/> COL_09 |                 | FLOAT  |                 |
| <input checked="" type="checkbox"/> COL_10 |                 | NUMBER |                 |

10 Total Records: 20 < 1 2 >

Confirm Cancel

- d. Click **Next**.
10. Perform a pre-check.
- a. After all settings are complete, perform a pre-check to ensure that the migration is successful.

Basic Information

|                           |                                      |                              |                    |
|---------------------------|--------------------------------------|------------------------------|--------------------|
| Task ID                   | 5b99e983-f78a-42c9-aa73-ba921d11b20r | Task Name                    | DRS-test-info      |
| Created                   | Dec 30, 2021 16:50:36 GMT+08:00      | Source Database IP           | 10.154.219.69      |
| Destination Database Name | Auto-drs-gaussdbv5-tar-1             | Destination Database IP Port | 172.16.24.234:8000 |

Check Again

Check success rate 4%

All checks must pass before you can continue. If any check requires confirmation, check and confirm the results before proceeding to the next step.

| Check Item                                                                                | Check Result |
|-------------------------------------------------------------------------------------------|--------------|
| Database parameters                                                                       |              |
| Whether the source database contains unsupported table field types                        | Checking     |
| Whether the destination database is compatible with the source database                   | Checking     |
| Whether the character set of the source database matches that of the destination database | Checking     |
| Whether the destination database has sufficient available connections                     | Checking     |
| Whether the selected objects exist in the destination database                            | Checking     |
| Whether the destination database contains the configured databases                        | Checking     |
| Whether there are source database foreign keys                                            | Checking     |
| Whether tables to be migrated contain primary keys                                        | Checking     |

- b. If any check item fails, review the cause and rectify the fault. Then, click **Check Again**.

Check Again

Check success rate 

100%

 All checks must pass before you can continue. If any check requires confirmation, check and confirm the results before proceeding to the next step.

| Check Item                                                                                | Check Result |
|-------------------------------------------------------------------------------------------|--------------|
| Database parameters                                                                       |              |
| Whether the source database contains unsupported table field types                        | Passed       |
| Whether the destination database is compatible with the source database                   | Passed       |
| Whether the character set of the source database matches that of the destination database | Passed       |
| Whether the destination database has sufficient available connections                     | Passed       |
| Whether the selected objects exist in the destination database                            | Passed       |
| Whether the destination database contains the configured databases                        | Passed       |
| Whether there are source database foreign keys                                            | Passed       |
| Whether tables to be migrated contain primary keys                                        | Passed       |
| Whether existing data meets the constraints                                               | Passed       |
| Whether the source database character set is supported                                    | Passed       |
| Whether the source database has sufficient available connections                          | Passed       |
| Whether the source database container type is correct                                     | Passed       |
| Whether archive logs are enabled on the source database                                   | Passed       |
| Whether the source database name is valid                                                 | Passed       |
| Whether the supplementary log is enabled for the source database.                         | Passed       |
| Whether OGG log reading is enabled on the source database                                 | Passed       |
| Whether the source database table name is valid                                           | Passed       |

- c. After all check items pass the pre-check, click **Next**.
11. Confirm the task.
- a. Check that all configured information is correct.

Start Time

Start upon task creation

Start at a specified time

Send Notifications

If disabled, DMS alarms, such as task failure, high latency, and frozen, cannot be received.

Stop Abnormal Tasks After

14

Abnormal tasks run longer than the period you set (unit: day) will automatically stop.

Details

| Product Name         | Configuration                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------|
| Task Information     |                                                                                                               |
| Name                 | DMS-test-info                                                                                                 |
| Description          | Source Database IP Address or Domain Name: 10.154.219.69 Destination DB Instance Name: Auto-ds-gaussdb5-tan-1 |
| Synchronization Mode | Full/Incremental synchronization                                                                              |
| Data Flow            | To the cloud                                                                                                  |

- b. Click **Submit**. In the display box, select **I have read the precautions**.
- c. Click **Submit**.

## Notice



During the synchronization, do not perform any operations on the destination DB instance through the management console. To ensure migration success, we strongly recommend that you read the [migration precautions](#) carefully before starting migration tasks and follow the instructions to ensure migration stability.



If the task status is abnormal for more than 14 days, the task automatically stops. Pay attention to the alarms you received and handle the task in time to resume the download and avoid task retry failure.



I have read the precautions.

Submit

12. After the task is submitted, view and manage it.

After the task is created, return to the task list to view the status of the created task.

|                          |                                                             |                     |       |          |                   |                  |                         |                           |             |                                  |                     |           |
|--------------------------|-------------------------------------------------------------|---------------------|-------|----------|-------------------|------------------|-------------------------|---------------------------|-------------|----------------------------------|---------------------|-----------|
| Batch Operations         |                                                             | View Abnormal Tasks |       | Oracle   | All network types | All statuses     | Enter a task name or ID | Search by Tag             |             |                                  |                     |           |
| <input type="checkbox"/> | Task Name/ID                                                | Status              | Delay | Charging | Data Flow         | DB Engine        | Synchronization         | Created                   | Network     | Billing Mode                     | Description         | Operation |
| <input type="checkbox"/> | <a href="#">DRS-test-info</a><br>5b99e983-f78a-42c9-aa73... | Starting            | --    | No       | To the cloud      | Oracle-GaussD... | Full-Incremental        | Dec 30, 2021 16:50:36 ... | Public n... | Pay per-Use<br>Created on Dec... | Source Databases... | Stop      |

## Step 5: Verify Data After Migration

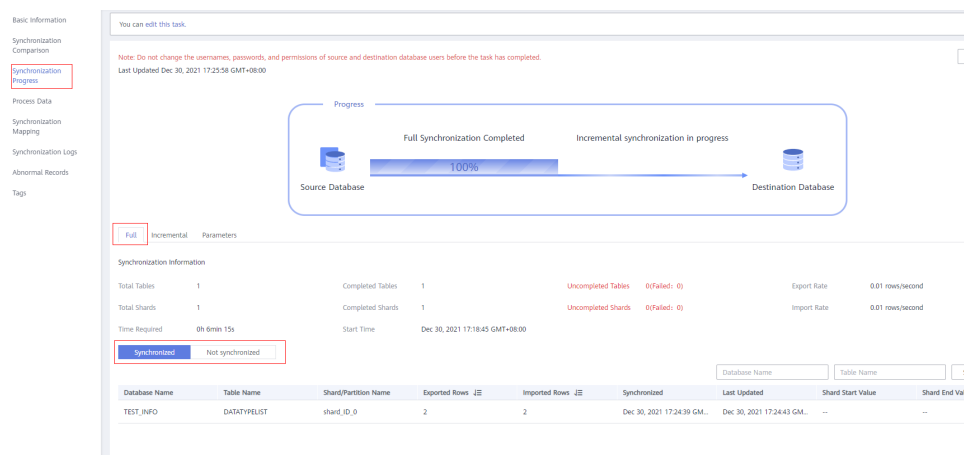
When the task status changes to **Incremental**, the full synchronization is complete. You can log in to GaussDB and view the data migration result.

1. Wait until the migration task status becomes **Incremental**.

|                          |                                             |             |    |    |              |                  |                  |                           |             |                                  |        |      |      |       |
|--------------------------|---------------------------------------------|-------------|----|----|--------------|------------------|------------------|---------------------------|-------------|----------------------------------|--------|------|------|-------|
| <input type="checkbox"/> | DRS-test-info<br>5b99e983-f78a-42c9-aa73... | Incremental | -- | No | To the cloud | Oracle-GaussD... | Full-Incremental | Dec 30, 2021 16:50:36 ... | Public n... | Pay-per-Use<br>Created on Dec... | Source | Edit | Stop | Speed |
|--------------------------|---------------------------------------------|-------------|----|----|--------------|------------------|------------------|---------------------------|-------------|----------------------------------|--------|------|------|-------|

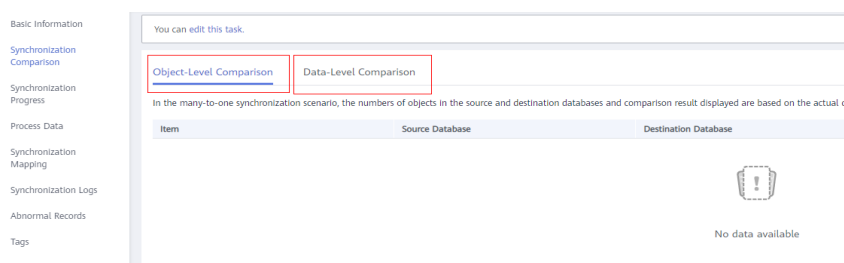
2. Click the task name to go to the **Basic Information** page.
3. On the **Synchronization Progress** page, view the full synchronization result.  
As shown in the following figure, the **DATATYPELIST** table in the **TEST\_INFO** database has been migrated to **shard\_0**. Two rows of data were migrated successfully.





#### 4. Verify data consistency.

- a. Choose **Synchronization Comparison > Object-Level Comparison** to view the database and table migration results.



- b. Choose **Synchronization Comparison > Data-Level Comparison**, click **Create Comparison Task**, and view the migration results of the rows in the table.

Create Comparison Task

Some comparison results may be inconsistent because data changes during the comparison cannot be synchronized to the destination in real time. You are advised to select a scheduled time to start the comparison during off-peak hours so that you can get an accurate comparison result.

\* Comparison Type: Row

\* Comparison Time: Start upon task creation Start at a specified time

\* Object: If any data in the source database changes, click the refresh button below.

Select All

For tables, only expanded databases are searched.

TEST\_INFO database

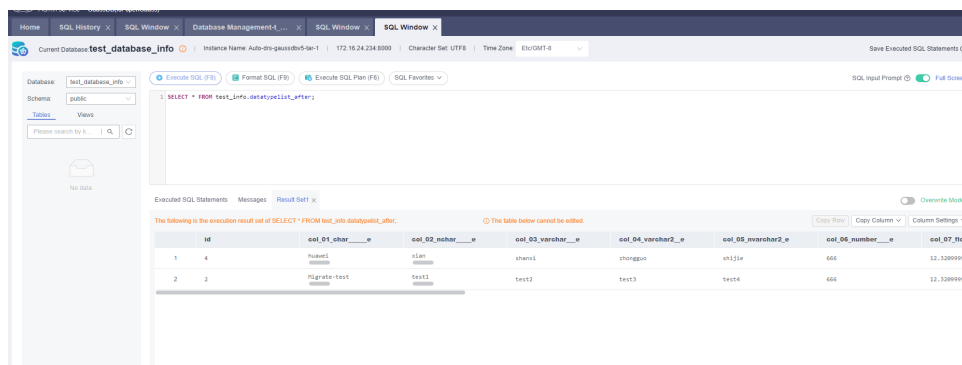
DATATYPELIST (New name: DATATY... table)

OK Cancel

5. Connect to the **test\_database\_info** database in GaussDB using DAS.
6. Run the following statement to query the full synchronization result:
- ```
SELECT * FROM test_info.datatypeplist_after;
```

After the schema in Oracle is migrated, it will be used as the schema in GaussDB. Therefore, it is required to add the schema in the query statement for exact query.

As shown in the following figure, all data types in the table were successfully migrated and the data is correct.



7. Verify incremental synchronization.

In full+incremental synchronization, after the full synchronization is complete, the data that is written to the source database after the task is created can still be synchronized to the destination database until the task is stopped. The following describes how to synchronize incremental data from the source database to the destination database:

- Use a database connection tool to connect to the source Oracle database based on its IP address.
- Run the following statement to insert a data record into the source database:

Insert a data record whose ID is 1.

```
insert into test_info.DATATYPELIST values(1,'Migrate-test','test1','test2','test3','test4',  
666,12.321,1.123,2.123,sysdate,sysdate,sysdate,sysdate,'hw','cb','df','FF','FF','AAAYEVAJAAAAACrA  
AA');  
commit;
```

- Run the following statement in the destination database to query the result:

```
SELECT * FROM test_info.datatypeelist_after;
```

As shown in the following figure, the new data inserted in the source database has been synchronized to the destination database in real time.



8. Stop the migration task.

After data is completely migrated to the destination database, stop the synchronization task.

- Locate the task and click **Stop** in the **Operation** column.



- In the display box, click **Yes**.

Stop Task ×

Are you sure you want to stop this task?

Name	Status
DRS-test-info	 Incremental

If you forcibly stop a task, the migration task will be stopped first. ×☐ Force stop task

Description:

· Once this task is stopped, it cannot be recovered.

Yes

No

9. After the migration is complete, test the GaussDB performance.

For details about how to test GaussDB performance, see [Performance White Paper](#).

5.3 Using DRS to Migrate Data from MySQL Database to GaussDB

Scenarios

This section describes how to use DRS to migrate data from an on-premises MySQL database to Huawei Cloud GaussDB in real time. Full+incremental synchronization can ensure that data is always in sync between the source MySQL database and the destination GaussDB instance.

Step 1: Create a VPC and Security Group

Step 2: Create a GaussDB Instance

Step 3: Construct Data Before Migration

Step 4: Migrating the Database

Step 5: Verify Data After Migration

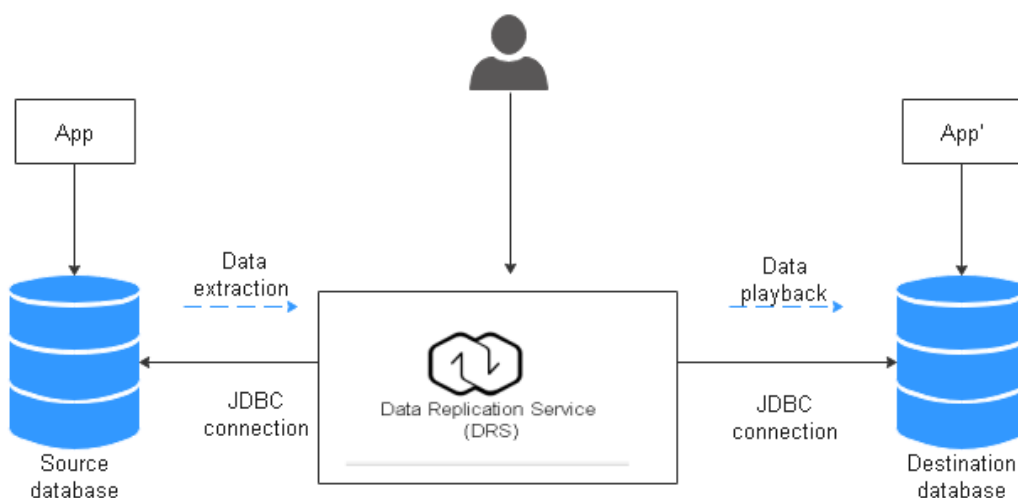
Problems to Resolve

- Enterprise workloads have been growing and evolving fast, and traditional databases lack the scalability needed to keep up. Enterprises need distributed databases.
- Building a traditional database means purchasing and installing servers, systems, databases, and other software. The O&M is expensive and difficult.
- Traditional databases have poor performance when it comes to handling complex queries.
- It is hard for traditional databases to smoothly synchronize data with no downtime.

Prerequisites

- You have registered with Huawei Cloud and completed account authentication.
- Your account balance is greater than or equal to \$0 USD.
- In a testing scenario, you have set up an on-premises MySQL database.
- You have obtained the IP address, port number, username, and password of the MySQL database to be migrated.

Service Architecture

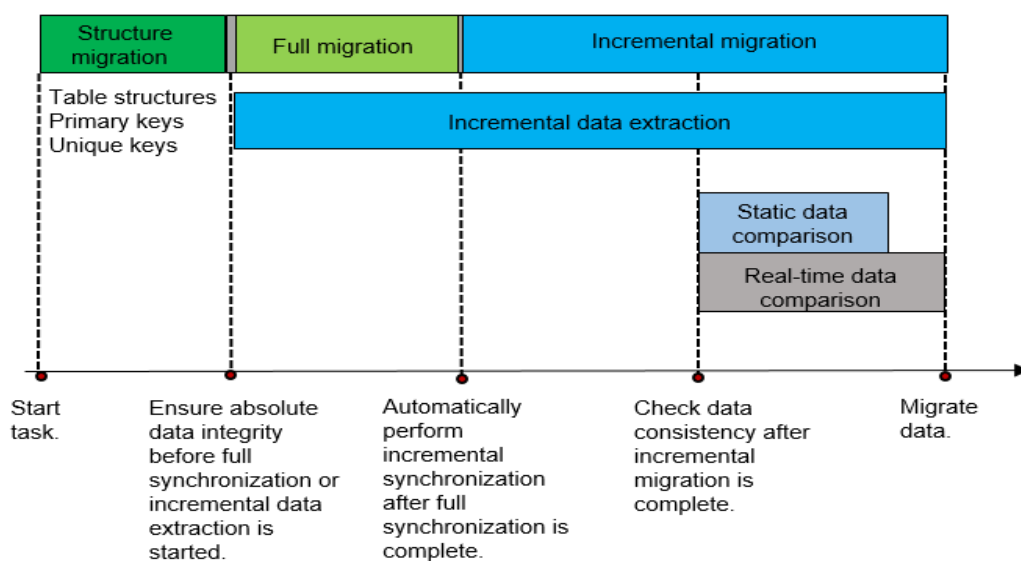


How Data Migration Works

The data migration process is completed using full and incremental synchronization, which includes the following operations:

1. In the full synchronization phase, schemas, including tables, primary keys, and unique keys, are synchronized first.
2. After schemas are synchronized, incremental data extraction is started to ensure that the incremental data generated during full data synchronization is completely extracted to the DRS instance.

3. A full migration task is started.
4. An incremental synchronization is automatically started after the full migration is complete. The replay starts from the position where the full synchronization starts.
5. A comparison task is started after the incremental replay is complete to check the data consistency. Real-time comparison is supported.
6. Workloads synchronization is started if the data is consistent between the source and destination databases.

Figure 5-11 Migration principle

Resource Planning

The resource planning in this section is just an example. You need to adjust it as needed.

Table 5-7 Resource planning

Category	Item	Planned Value	Remarks
VPC	VPC name	vpc-src-172	Specify a name that is easy to identify.
	Region	Test region	To achieve lower network latency, select the region nearest to you.
	AZ	AZ 3	-
	Subnet CIDR block	172.16.0.0/16	Select a subnet with sufficient network resources.

Category	Item	Planned Value	Remarks
	Subnet name	subnet-src-172	Specify a name that is easy to identify.
On-premises MySQL database	Database version	5.7.38	-
	Database user	test_info	Specify a database user. The user must at least have the following privileges: SELECT, LOCK TABLES, REPLICATION SLAVE, and REPLICATION CLIENT.
GaussDB	Instance name	Auto-drs-gaussdbv5-tar-1	Specify a name that is easy to identify.
	Database version	GaussDB V2.0-8.103 Enterprise edition	-
	Instance type	Distributed (3 CNs, 3 DN shards, and 3 replicas)	In this example, a distributed instance will be created.
	Deployment model	Independent	-
	Transaction consistency	Strong consistency	-
	Shards	3	-
	Coordinator nodes	3	-
	Storage type	Ultra-high I/O	-
	AZ	AZ 2	In this example, a single AZ is select. You are advised to select multiple AZs to improve instance availability in actual use.

Category	Item	Planned Value	Remarks
	Instance Specifications	Dedicated (1:8); 8 vCPUs 64 GB	Small specifications are selected for this test instance. You are advised to configure specifications based on service requirements in actual use.
	Storage space	480GB	A small storage space is selected for this test instance. You are advised to configure the storage space based on service requirements in actual use.
	Disk encryption	Disable	In this example, disk encryption is disabled. Enabling disk encryption improves the security of data, but may slightly affect the database read/write performance.
Logging in to the database through DAS	Database engine	GaussDB	-
	Database source	GaussDB	Select the GaussDB instance created in this example.
	Database name	postgres	-
	Username	root	-
	Password	-	Enter the password of the root user of the GaussDB instance created in this example.
DRS migration task	Migration task name	DRS-test-info	Specify a name that is easy to identify.
	Destination database name	test_database_info	Specify a name that is easy to identify. The name must be compatible with the MySQL database name.
	Source database engine	MySQL	-

Category	Item	Planned Value	Remarks
	Destination database engine	GaussDB	-
	Network type	Public network	In this example, a public network is used.

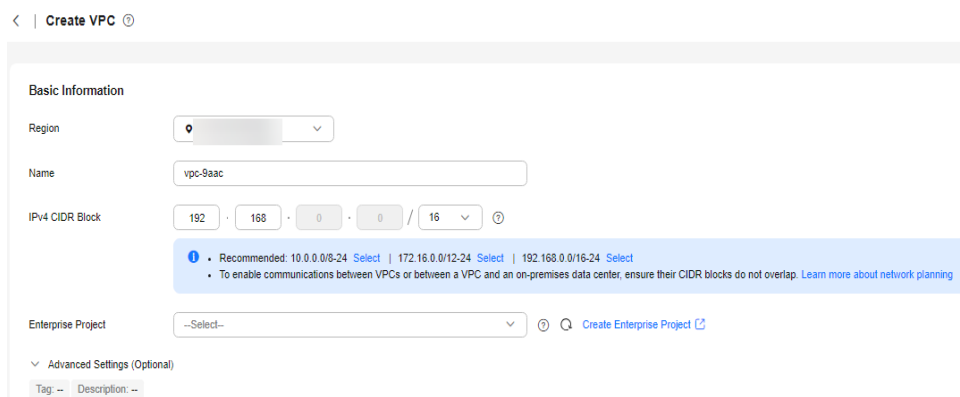
Step 1: Create a VPC and Security Group

Create a VPC and security group for the GaussDB instance.

Creating a VPC

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.
3. Click the service list icon on the left and choose **Networking > Virtual Private Cloud**. The VPC console is displayed.
4. Click **Create VPC**.

Figure 5-12 Basic information



The screenshot shows the 'Create VPC' console interface. At the top, there is a breadcrumb navigation: '< | Create VPC ⓘ'. Below this is the 'Basic Information' section. It contains the following fields and options:

- Region:** A dropdown menu with a location pin icon and a downward arrow.
- Name:** A text input field containing 'vpc-9aac'.
- IPv4 CIDR Block:** A field with four input boxes containing '192', '168', '0', and '0', followed by a slash and a box containing '16'. A help icon ⓘ is to the right.
- IPv4 CIDR Block Hint:** A blue information box with the text: 'Recommended: 10.0.0.0/8-24 [Select](#) | 172.16.0.0/12-24 [Select](#) | 192.168.0.0/16-24 [Select](#). To enable communications between VPCs or between a VPC and an on-premises data center, ensure their CIDR blocks do not overlap. [Learn more about network planning](#)'.
- Enterprise Project:** A dropdown menu showing '--Select--' and a help icon ⓘ. A link 'Create Enterprise Project' is also present.
- Advanced Settings (Optional):** A collapsed section with a downward arrow.
- Tag:** A field with a dropdown menu showing '--'.
- Description:** A field with a dropdown menu showing '--'.

Figure 5-13 Setting a subnet

Subnet Setting1

Subnet Name

subnet-9ab7

AZ

AZ1

AZ2

AZ3

AZ7

?

IPv4 CIDR Block

192

168

0

0

/

24

▼

Available IP Addresses: 251

⚠

The CIDR block cannot be modified after the subnet is created. Before creating a subnet, [plan subnet CIDR blocks](#) as required.

IPv6 CIDR Block (Optional)

☐ Enable

?

Associated Route Table

Default

?

▼ Advanced Settings (Optional)

Gateway: 192.168.0.1

DNS Server Address: 100.125.1.250,100.125.129....

Domain Name: --

NTP Server Address: --

...

Table 5-8 VPC parameters

Parameter	Description
Region	Region where the VPC will be created. Select the region nearest to you to ensure the lowest latency possible.
Name	VPC name. The name can contain: 1 to 64 characters. - Letters, digits, underscores (_), hyphens (-), and periods (.).
IPv4 CIDR Block	IPv4 CIDR block of the VPC. Consider the following when specifying a CIDR block: - IP addresses: Reserve sufficient IP addresses for subsequent service expansion. - IP address ranges: Avoid IP address conflicts if you need to connect a VPC to an on-premises data center or connect two VPCs.
Enterprise Project	Enterprise project that the VPC belongs to. An enterprise project facilitates project-level management and grouping of cloud resources and users. The name of the default project is default .
Advanced Settings (Optional) > Tag	Click ▼ next to Advanced Settings (Optional) to set this parameter. Add tags to help you quickly identify, classify, and search for your VPCs.

Parameter	Description
Advanced Settings (Optional) > Description	Click  next to Advanced Settings (Optional) to set this parameter. Enter the description about the VPC in the text box as required. The description can contain a maximum of 255 characters and cannot contain angle brackets (< or >).

Table 5-9 Subnet parameters

Parameter	Description
Subnet Name	Subnet name. The name can contain: • 1 to 64 characters. • Letters, digits, underscores (_), hyphens (-), and periods (.).
AZ	A geographic location with independent power supply and network facilities in a region. AZs are physically isolated, and AZs in the same VPC are interconnected through an internal network. Each region contains multiple AZs. If one AZ is faulty, other AZs in the same region continue to provide services.

Parameter	Description
IPv4 CIDR Block	This parameter is displayed only in regions where IPv4/IPv6 dual stack is supported. A subnet is a unique CIDR block with a range of IP addresses in a VPC. Comply with the following principles when planning subnets: • Planning CIDR block size: After a subnet is created, the CIDR block cannot be changed. You need to plan the CIDR block in advance based on the number of IP addresses required by your service. – The subnet CIDR block cannot be too small. Ensure that the number of available IP addresses in the subnet meets service requirements. The first and last three addresses in a subnet CIDR block are reserved for system use. For example, in subnet 10.0.0.0/24, 10.0.0.1 is the gateway address, 10.0.0.253 is the system interface address, 10.0.0.254 is used by DHCP, and 10.0.0.255 is the broadcast address. – The subnet CIDR block cannot be too large, either. If you use a CIDR block that is too large, you may not have enough CIDR blocks available for new subnets, which can be a problem when you want to scale out services. • Avoiding subnet CIDR block conflicts: If you need to connect two VPCs or connect a VPC to an on-premises data center, there cannot be any CIDR block conflicts. If the subnet CIDR blocks at both ends of the network conflict, create a subnet. For details, see Creating a Subnet for an Existing VPC. A subnet mask can be between the netmask of its VPC CIDR block and /29 netmask. If a VPC CIDR block is 10.0.0.0/16, its subnet mask can be between 16 and 29. For more information about VPC subnet planning, see VPC and Subnet Planning.
IPv6 CIDR Block (Optional)	This parameter is displayed only in regions where IPv4/IPv6 dual stack is supported. If you enable this function, the system automatically assigns an IPv6 CIDR block to the subnet. Currently, the IPv6 CIDR block cannot be customized. Once enabled, this function cannot be disabled.
Advanced Settings (Optional) > Gateway	Click  next to Advanced Settings (Optional) to set this parameter. Retain the default value unless there are special requirements.

Parameter	Description
Advanced Settings (Optional) > DNS Server Address	Click  next to Advanced Settings (Optional) to set this parameter. Huawei Cloud private DNS server addresses are entered by default. This allows ECSs in a VPC to communicate with each other. You can also access cloud services directly through private DNS servers. You can change the default DNS server address if needed. If you delete the default DNS server addresses, you may fail to access other services on the cloud. Exercise caution when performing this operation. You can click Reset on the right to restore DNS server addresses to the default value. A maximum of two DNS server IP addresses are allowed. Separate them by a comma (,).
Advanced Settings (Optional) > Domain Name	Click  next to Advanced Settings (Optional) to set this parameter. Enter domain names separated by spaces. A maximum of 254 characters are allowed. Each label in a domain name can contain a maximum of 63 characters. (For example, test and com are two labels in test.com.) To access a domain name, simply enter the domain name prefix. ECSs in the subnet automatically match the configured domain name suffix. If the domain names are changed, ECSs newly added to this subnet will use the new domain names. If an existing ECS in this subnet needs to use the new domain names, restart the ECS or run a command to restart the DHCP Client service or network service. NOTE The command for updating the DHCP configuration depends on the ECS OS. The commands here are only for your reference. - Restart the DHCP Client service: service dhcpcd restart - Restart the network service: service network restart

Parameter	Description
Advanced Settings (Optional) > NTP Server Address	Click  next to Advanced Settings (Optional) to set this parameter. If you want to add NTP server addresses for a subnet, you can specify this parameter. The IP addresses are added in addition to the default NTP server addresses. • If you add or change the NTP server addresses of a subnet, you need to renew the DHCP lease for or restart all the ECSs in the subnet to make the change take effect immediately. • If the NTP server addresses have been cleared out, restarting the ECSs will not help. You must renew the DHCP lease for all ECSs to make the change take effect immediately.
Advanced Settings (Optional) > IPv4 DHCP Lease Time	Click  next to Advanced Settings (Optional) to set this parameter. This parameter is displayed only in regions where IPv4/IPv6 dual stack is supported. You can set the DHCP lease time of an IPv4 address. This parameter specifies the period during which a client can use an IP address automatically assigned by the DHCP server. After the lease time expires, a new IP address will be assigned to the client. • Limited: Set the DHCP lease time. The unit can be days or hours. • Unlimited: The DHCP lease time does not expire. If the time period is changed, the new lease time takes effect when the instance (such as an ECS) in the subnet is renewed next time. You can wait for the instance to be renewed automatically or manually modify the lease time. If you want the new lease time to take effect immediately, manually renew the lease or restart the instance.
Advanced Settings > IPv6 DHCP Lease Time	Click  next to Advanced Settings (Optional) to set this parameter. This parameter is displayed for regions where IPv4/IPv6 dual stack is supported and IPv6 is enabled. You can set the DHCP lease time of an IPv6 address in the same way as how you do with an IPv4 address.
Advanced Settings (Optional) > Tag	Click  next to Advanced Settings (Optional) to set this parameter. You can add tags to help you quickly identify, classify, and search for your subnets.

Parameter	Description
Advanced Settings (Optional) > Description	Click  next to Advanced Settings (Optional) to set this parameter. You can enter the subnet description in the text box as required. The description can contain a maximum of 255 characters and cannot contain angle brackets (< or >).

5. Configure parameters as needed and click **Create Now**.
6. Return to the VPC list and check whether the VPC is created.
If the VPC status becomes available, the VPC has been created.

Creating a Security Group

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.
3. Click the service list icon on the left and choose **Networking > Virtual Private Cloud**. The VPC console is displayed.
4. In the navigation pane, choose **Access Control > Security Groups**.
5. Click **Create Security Group**.
6. Specify a security group name and other information.

Figure 5-14 Basic information

< | Create Security Group

Summary

Region 
Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

Name

Enterprise Project   [Create Enterprise Project](#)

Tag (Optional) TMS's predefined tags are recommended for adding the same tag to different cloud resources. [Create predefined tags](#)
[+ Add Tag](#)
You can add 20 more tags.

Description (Optional)
0/255

Table 5-10 Parameter description

Parameter	Description
Region	Region where the VPC will be created. Select the region nearest to you to ensure the lowest latency possible. An instance must be in the same region as its associated security group.
Name	The name of the security group. The name can contain: 1 to 64 characters. - Letters, digits, underscores (_), hyphens (-), and periods (.). You can change the security group name after a security group is created. It is recommended that you give each security group a different name.
Enterprise Project	When creating a security group, you can add the security group to an enabled enterprise project. An enterprise project facilitates project-level management and grouping of cloud resources and users. The name of the default project is default .
Tag	You can add tags to help you identify, classify, and search for your security groups.
Description	The description of the security group. The description can contain a maximum of 255 characters and cannot contain angle brackets (< or >).

- Click the **Inbound Rules** tab and then click **Add Rule**.
- Configure an inbound rule, add the IP address of the source database, and click **OK**.

Figure 5-15 Configuring an inbound rule

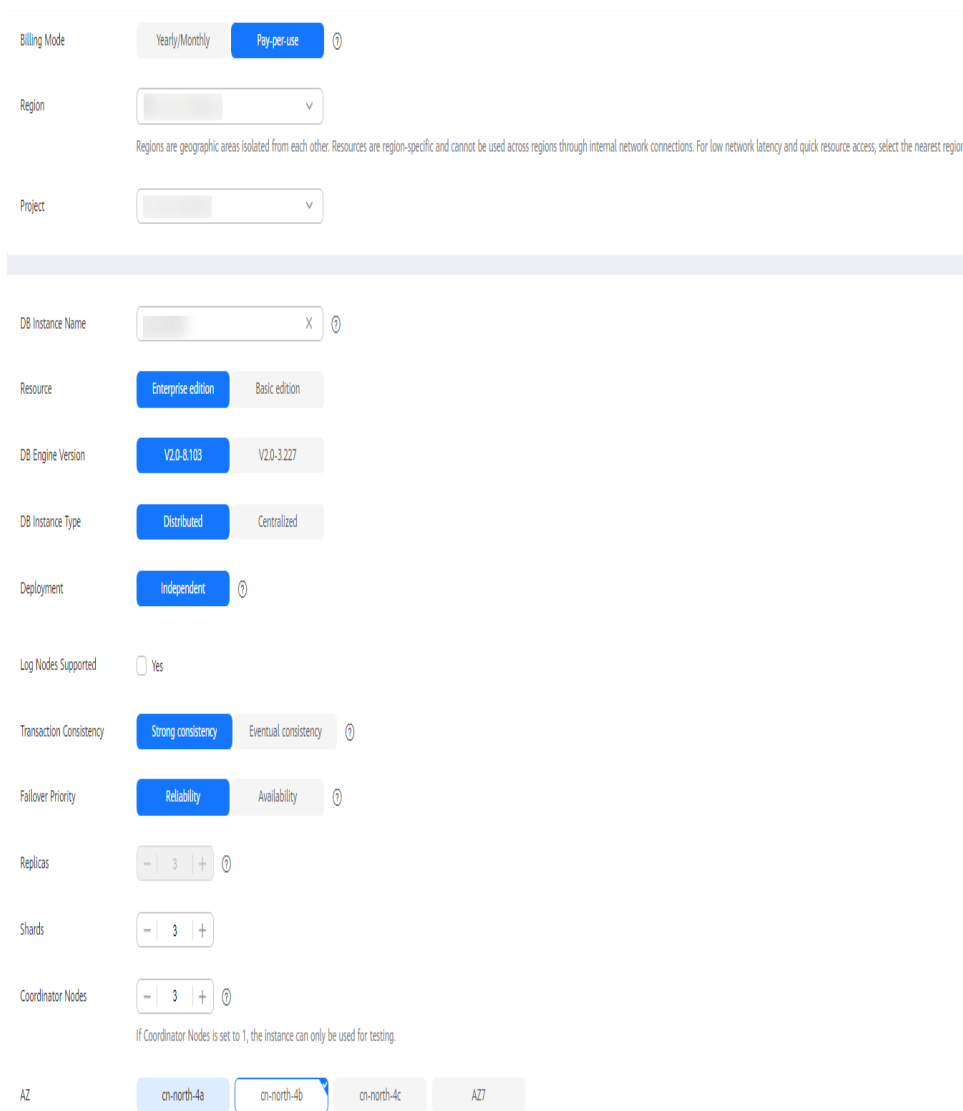
The screenshot displays the 'Inbound Rules' configuration page. At the top, there are tabs for 'Inbound Rules' and 'Outbound Rules'. Below the tabs are buttons for 'Add Rule' and 'Fast-Add Rule'. A table lists existing rules with columns: Priority, Action, Type, Protocol & Port, Source, Description, and Operation. The table shows three rules: 1. Priority 1, Action Allow, Type IPv4, Protocol & Port Protocol / TCP (Custom ports), Source IP address, Description (empty), Operation (Replicate, Delete). 2. Priority 1, Action Allow, Type IPv4, Protocol & Port Protocol / All, Source Security group, Description Allows the instances in the security group to, Operation Delete. 3. Priority 1, Action Allow, Type IPv6, Protocol & Port Protocol / All, Source Security group, Description Allows the instances in the security group to, Operation Delete. Below the table is a 'Create Now' button.

- Configure parameters as needed and click **Create Now**.

Step 2: Create a GaussDB Instance

Create a GaussDB instance as the destination database of the migration task.

1. Log in to the [Huawei Cloud console](#).
2. Click  in the upper left corner and select a region.
3. Click the service list icon on the left and choose **Databases > GaussDB**.
4. In the navigation pane on the left, choose **GaussDB > Instances**.
5. Click **Buy DB Instance**.
6. On the page shown in [Figure 5-16](#), configure basic information about the instance, including the instance name, billing mode, edition type, DB engine version, instance type, transaction consistency, number of shards, number of coordinator nodes, and deployment AZ.

Figure 5-16 Basic information

The screenshot displays the 'Basic information' configuration page for a GaussDB instance. The settings are as follows:

- Billing Mode:** Yearly/Monthly (disabled), Pay-per-use (selected).
- Region:** A dropdown menu with a placeholder text. A note below states: 'Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.'
- Project:** A dropdown menu with a placeholder text.
- DB Instance Name:** A text input field with a placeholder text and a clear button (X).
- Resource:** Enterprise edition (selected), Basic edition (disabled).
- DB Engine Version:** V2.0-8.103 (selected), V2.0-3.227 (disabled).
- DB Instance Type:** Distributed (selected), Centralized (disabled).
- Deployment:** Independent (selected).
- Log Nodes Supported:** Yes (unchecked).
- Transaction Consistency:** Strong consistency (selected), Eventual consistency (disabled).
- Failover Priority:** Reliability (selected), Availability (disabled).
- Replicas:** A numeric input field set to 3.
- Shards:** A numeric input field set to 3.
- Coordinator Nodes:** A numeric input field set to 3. A note below states: 'If Coordinator Nodes is set to 1, the instance can only be used for testing.'
- AZ:** cn-north-4a (disabled), cn-north-4b (selected), cn-north-4c (disabled), AZ7 (disabled).

7. Select the instance specifications and storage space.

Figure 5-17 Instance specifications

Instance Specifications **Dedicated(1:8)** ⓘ

Flavor Name

- ☐ 4 vCPUs | 32 GB Unavailable for production environment
- ☒ 8 vCPUs | 64 GB
- ☐ 16 vCPUs | 128 GB
- ☐ 32 vCPUs | 256 GB
- ☐ 64 vCPUs | 512 GB

DB Instance Specifications Dedicated(1:8) | 8 vCPUs | 64 GB

Storage Type **Ultra-high I/O** [Learn more about storage types.](#)

Storage Space (GB) **480**

120 14480 28840 43200 72000

GaussDB provides free backup storage equal to the amount of your purchased storage space. After the free backup space is used up, charges are applied based on the [backup space pricing details](#).

Disk Encryption **Disable** **Enable** ⓘ

8. Select the VPC created in [Creating a VPC](#) and security group created in [Creating a Security Group](#) for the instance and configure the database port.

Figure 5-18 Selecting a VPC and security group

Relationship among VPCs, subnets, security groups, and DB instances. ⓘ

VPC **default_vpc** **default_subnet** ⓘ

If you want to create a VPC, go to [the VPC console](#).

Security Group **default** [View Security Group](#) ⓘ

In a security group, rules that authorize connections to DB instances apply to all DB instances associated with the security group.

Ensure that the TCP ports in the inbound rule of the selected security group contain 8000-8100, 20050, 5000-5001, 2379-2380, 6000, 6500, 40000-60480.

Security Group Rules **▼** [Add Inbound Rule](#)

Database Port **Default port: 8000**

9. Configure the password and other information.

Figure 5-19 Configuring the password and other information

Administrator **root**

Administrator Password ********* ⓘ Keep your password secure. The system cannot retrieve your password.

Confirm Password ********* ⓘ

Parameter Template **Default-Enterprise-Edition-GaussDB-8.10...** [View Parameter Template](#) ⓘ

Enterprise Project **default** [View Enterprise Projects](#) ⓘ

Tag **+ Add Tag** ⓘ

TMS's predefined tags are recommended for adding the same tag to different cloud resources. [Create predefined tags](#) ⓘ

You can add 20 more tags.

10. Click **Next**, confirm the information, and click **Submit**.
11. Go to the instance list.

If status of the instance becomes **Available**, the instance has been created.

Step 3: Construct Data Before Migration

Before the migration, prepare some data types in the source database for verification after the migration is complete. The end-to-end test data in this section is for reference only.

For details about the data types supported by DRS, see [MySQL->GaussDB](#).

Perform the following steps to construct data in the source database:

1. Use a database connection tool to connect to the source MySQL database based on its IP address.
2. Construct data in the source database based on data types supported by DRS.

- a. Create a test user.

```
create user test_info identified by xxx;
```

test_info indicates the user created for the migration, and *xxx* indicates the password of the user.

- b. Create a database named **test_info** under the user.

```
CREATE DATABASE test_info;
```

- c. Create a table in the **test_info** database.

```
CREATE TABLE `test_info`.`test_table` (
```

```
  `id` int NOT NULL,
```

```
  `c1` char(10) DEFAULT NULL,
```

```
  `c2` varchar(10) DEFAULT NULL,
```

```
  `c3` binary(10) DEFAULT NULL,
```

```
  `c4` varbinary(10) DEFAULT NULL,
```

```
  `c5` tinyblob,
```

```
  `c6` mediumblob,
```

```
  `c7` longblob,
```

```
  `c8` tinytext,
```

```
  `c9` text,
```

```
  `c10` mediumtext,
```

```
  `c11` longtext,
```

```
  `c12` enum('1','2','3') DEFAULT NULL,
```

```
  `c13` set('1','2','3') DEFAULT NULL,
```

```
  `c14` tinyint DEFAULT NULL,
```

```
  `c15` smallint DEFAULT NULL,
```

```
  `c16` mediumint DEFAULT NULL,
```

```
  `c17` bigint DEFAULT NULL,
```

```
  `c18` float DEFAULT NULL,
```

```
  `c19` double DEFAULT NULL,
```

```
`c20` date DEFAULT NULL,  
`c21` datetime DEFAULT NULL,  
`c22` timestamp,  
`c23` time DEFAULT NULL,  
`c24` year DEFAULT NULL,  
`c25` bit(10) DEFAULT NULL,  
`c26` json DEFAULT NULL,  
`c27` decimal(10,0) DEFAULT NULL,  
`c28` decimal(10,0) DEFAULT NULL,  
PRIMARY KEY (`id`)  
);
```

- d. Assign privileges to the user.

```
GRANT SELECT,LOCK TABLES ON <database>.<table> to test_info;  
GRANT REPLICATION SLAVE,REPLICATION CLIENT ON *.* to test_info;
```

In the preceding commands, *test_info* indicates the user created for this migration task, *<database>* indicates the name of the database to be synchronized, and *<table>* indicates the name of the table to be synchronized. Replace them as required.

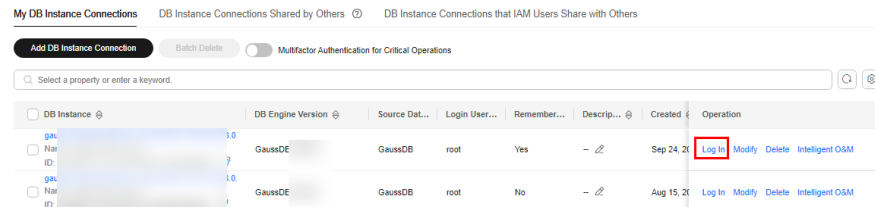
- e. Insert two rows of data into the table.

```
insert into test_info.test_table values  
(1,'a','b','111','111','tinyblob','mediumblob','longblob','tinytext','text',  
mediumtext,'longtext','1','3',1,2,3,4,1.123,1.1234,'2024-03-08','2024-0  
3-08 08:00:00','2024-03-08  
08:00:00','08:00:00','2024','1010',{'a':"b"},1.23,1.234);  
insert into test_info.test_table values  
(2,'a','b','111','111','tinyblob','mediumblob','longblob','tinytext','text',  
mediumtext,'longtext','1','3',1,2,3,4,1.123,1.1234,'2024-03-08','2024-0  
3-08 08:00:00','2024-03-08  
08:00:00','08:00:00','2024','1010',{'a':"b"},1.23,1.234);
```

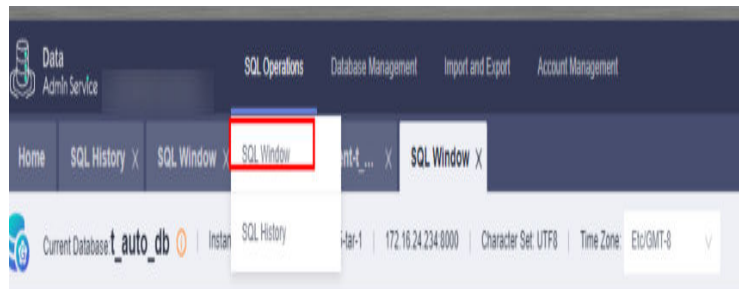
3. Create a database in the destination GaussDB instance.

- Log in to the [management console](#).
- Click  in the upper left corner and select a region.
- Click the service list icon on the left and choose **Databases > Data Admin Service**.
- In the navigation pane on the left, choose **Development Tool** to go to the login list page.
- Click **Add Login**.
- On the displayed page, select the DB engine and source database, enter the login username, password, and description (optional), and enable **Show Executed SQL Statements**.
- Click **Test Connection** to check whether the connection is successful.
If a message is displayed indicating the connection is successful, continue with the operation. If a message is displayed indicating the connection failed and the failure cause is provided, rectify the fault according to the error message.

- h. Click **OK**.
- i. Locate the added record and click **Log In** in the **Operation** column.

Figure 5-20 Logging in to an instance

- j. Choose **SQL Operations > SQL Window** on the top menu bar.



- k. Run the following statement to create a database compatible with MySQL:
test_database_info indicates the database name. Replace it as required.
`CREATE DATABASE test_database_info DBCOMPATIBILITY 'mysql';`

Step 4: Migrating the Database

Create a DRS instance and synchronize data from the **test_info** database in the on-premises MySQL database to the **test_database_info** database in the GaussDB instance.

Performing a Pre-migration Check

Before creating a migration task, check the migration conditions to ensure smooth migration.

Before the migration, you need to obtain the [notes on migration to the cloud](#).

Creating a Migration Task

1. Log in to the [management console](#).
2. Click  in the upper left corner and select a region.
Select the region where the destination instance is deployed.
3. Click the service list icon on the left and choose **Databases > Data Replication Service**.
4. In the navigation pane on the left, choose **Data Synchronization Management**. On the displayed page, click **Create Synchronization Task**.
5. Configure synchronization instance information.

- a. Select a region, billing mode, and project, and enter a task name.

The screenshot shows a configuration form with the following fields:

- Billing Mode:** Two buttons, "Yearly/Monthly" and "Pay-per-use". "Pay-per-use" is highlighted in blue.
- Region:** A dropdown menu with a location pin icon and a downward arrow. Below it, a note states: "Regions are geographic areas isolated from each other. For low network latency and quick resource access, select the nearest region."
- Project:** A dropdown menu with a downward arrow.
- * Task Name:** A text input field containing "DRS-2843" and a help icon (?) on the right.
- Description:** A large text area with a height indicator "0/256" at the bottom right and a help icon (?) on the right.

- b. Specify **Data Flow**, **Source DB Engine**, **Destination DB Engine**, **Network Type**, **DRS Task Type**, **Destination DB Instance**, **Synchronization Instance Subnet** (optional), **Synchronization Mode**, **Specifications**, **AZ**, and **Tags** (optional).

The screenshot shows the "Synchronization Instance Details" page with the following sections:

- Synchronization Instance Details** (with a help icon ?)
- The following information cannot be modified after you go to the next page:**
- * Data Flow:** Three buttons: "To the cloud" (highlighted in blue), "Out of the cloud", and "Self-built to self-built". A note below states: "The destination database must be a database in the current cloud. If you want to synchronize data between databases, select To the cloud."
- * Source DB Engine:** A row of buttons: MySQL, Oracle, Cassandra, DB2 for LUW, ODM, MariaDB, MongoDB, PostgreSQL, Microsoft SQL Server, GaussDB(for MySQL), and TiDB. "MySQL" is highlighted in blue.
- * Destination DB Engine:** A row of buttons: MySQL, GaussDB Distributed (highlighted in blue), GaussDB Primary/Standby, MariaDB, PostgreSQL, and GaussDB(for MySQL).
- * Network Type:** A dropdown menu with "Public network" selected and a help icon (?). Below it, a note states: "DRS will automatically bind the specified EIP to the DRS instance and unbind the EIP after the task is complete. For details about the data transmission fee when an EIP is specified, see the pricing details of the EIP service."
- * Destination DB Instance:** A dropdown menu with "No DB instance available." selected and a help icon (?). To the right are links: "View DB Instance" and "View Unselectable DB Instance".
- * Synchronization Instance Subnet:** A dropdown menu with "Select the subnet" selected and a help icon (?). To the right are links: "View Subnets" and "View Occupied IP Address".
- * Synchronization Mode:** Three buttons: "Full-Incremental" (highlighted in blue), "Full", and "Incremental". A note below states: "This synchronization type synchronizes data in real time. After a full synchronization initializes the destination database, an incremental synchronization parses logs to ensure data consistency between the source and destination databases."
- * Specify EIP:** A dropdown menu and a "Create an EIP" button.

- c. Click **Create Now**.
6. Configure the source and destination database information.
- a. Enter the IP address, port number, username, and password of the source database.
- Click **Test Connection**.

Configure Your Own DNS Server

DNS Server



Source Database

Select Connection



System databases, users, parameters, and jobs will not be migrated. You need to manually import users and jobs to the destination database and configure parameters in parameter templates of the destination database.

Database Type

Self-built on ECS

RDS DB instance

VPC

vpc-17a0(192.168.0.0/16)

[View VPC](#)

Subnet

subnet-17d0(192.168.0.0/24)

[View Subnets](#)

IP Address or Domain Name

Port

Database Username

Database Password

SSL Connection



If you want to enable SSL connection, ensure that SSL has been enabled on the source database, related parameters have been correctly configured, and an SSL certificate has been uploaded.

Encryption Certificate

Select

Test Connection

- b. Enter the username and password of the destination database.
Click **Test Connection**.

Destination Database

DB Instance Name

Database Username

Database Password

Test Connection

This button is available only after the replication instance is created successfully.

- c. Click **Next**. In the displayed box, read the message carefully and click **Agree**.
7. Configure the synchronization task.
- a. Select the object type for full synchronization. If the table structure to be synchronized has not been created in the destination database, select **Table structure** (the table structure contains primary keys and unique keys) for **Synchronization Object Type**. Otherwise, deselect **Table structure**. Select **Index** for **Synchronization Object Type** as needed.

Synchronization Object Type



Table structure



Data



Index

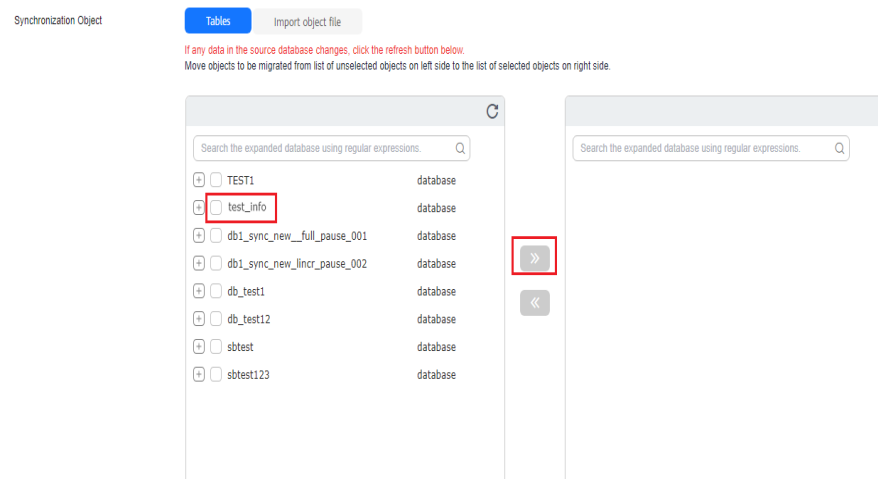
When you manually create a table structure in the destination database, for details about the data type, see [Mapping Data Types](#).

- b. Specify **Incremental Conflict Policy**. This option determines how the system reacts when there is a data conflict (for example, duplicate primary or unique keys) between the source and destination databases.

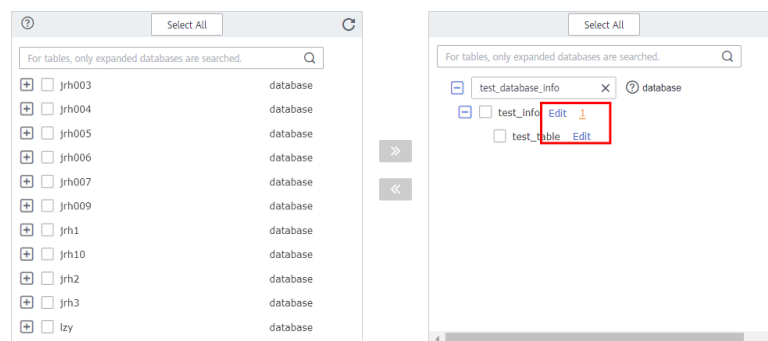
- **Ignore**: The system will ignore the conflicting data in the source database and continue the subsequent synchronization process. If you select **Ignore**, data in the source database may be inconsistent with that in the destination database.
- **Report error**: The synchronization task will be stopped and fail. You can view the details in synchronization logs.
- **Overwrite**: Conflicting data in the destination database will be overwritten.



- c. Select the databases and tables of the source database to be migrated. In this example, select the **test_table** table from the **test_info** database.



- d. Locate the database and table, respectively, and click **Edit** to change the database name and table name as needed.



- e. On the displayed dialog box, enter a new name, for example, **DATATYPELIST_After**.

The name cannot include special characters. Otherwise, an error will be reported during SQL statement execution after the migration.

New Table Name

The new table will be used in the destination database.

Edit Table Name

OK

Cancel

- f. Confirm the settings and click **Next**.

Select All

For tables, only expanded databases are searched.

jrh003

database

jrh004

database

jrh005

database

jrh006

database

jrh007

database

jrh009

database

jrh1

database

jrh10

database

jrh2

database

jrh3

database

lzy

database

>>

<<

Select All

For tables, only expanded databases are searched.

test_database_info

database

test_info

Edit 1

test_table

New name: DATATYPELIST_After Edit

8. Confirm advanced settings.

The information on the **Advanced Settings** page is for confirmation only and cannot be modified. After confirming the information, click **Next**.

Create Synchronization Instance

Configure Source and Destination Databases

Set Synchronization Task

Advanced Settings

Process Data

Check Task

Confirm Task

Basic Information

Task ID

S09N983-F78a-42c9-aa73-ba92141j32Dr

Task Name

DMS-test-info

Created

Dec 30, 2021 16:50:36 GMT+08:00

Source Database IP

Destination Database Name

Auto-dm-gaussdbv5-tan-1

Destination Database IP Port

8000

Full Synchronization Settings

Synchronization Object Type

☒ Table structure ☒ Data ☒ Constraint(excluding foreign keys)

Stream Mode

☒

Concurrent Export Tasks

-

8

+

Concurrent Import Tasks

-

8

+

Import Mode

COPY

INSERT

Rows per Shard

500000

Incremental Data Capture Settings

Concurrent Log Capture Tasks

-

1

+

Incremental Replay Settings

Concurrent Replay Tasks

-

64

+

Price

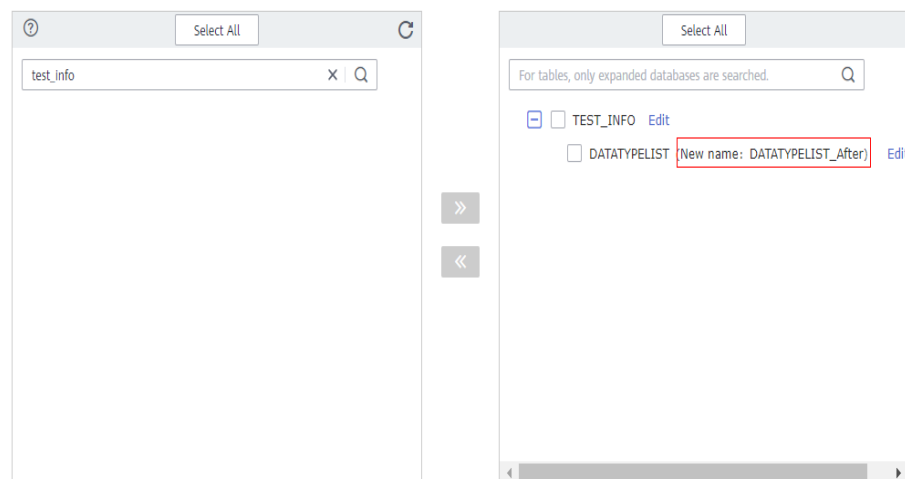
Previous

Next

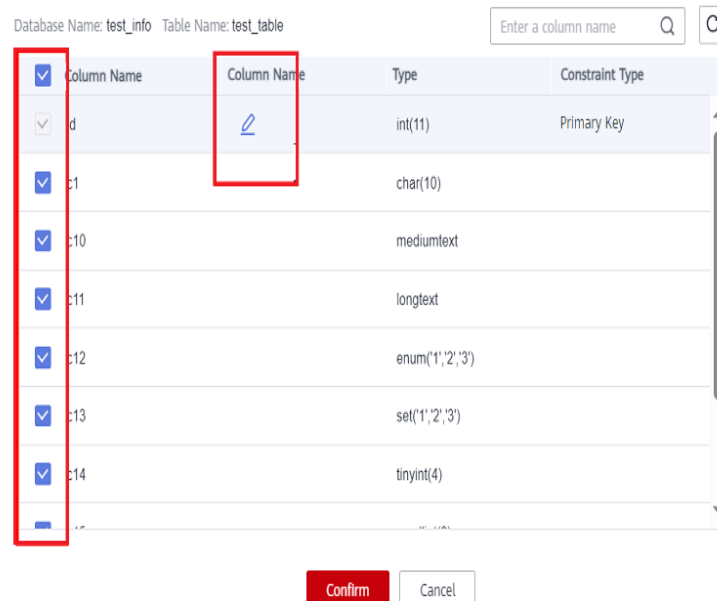
9. Process data.

On this page, you can process the table to be migrated. For example, you can select the column to be migrated and change its name. In this example, change the column name **c1** to **new-line**.

- a. Click **Edit** next to the table to be processed.



b. Edit the **c1** column.



c. Enter the new name **new-line** and click **Confirm**.

d. Click **Next**.

10. Perform a pre-check.

- After all settings are complete, perform a pre-check to ensure that the migration is successful.
- If any check item fails, review the cause and rectify the fault. Then, click **Check Again**.

Check Again

Check success rate 100% All checks must pass before you can continue. If any check requires confirmation, check and confirm the results before proceeding to the next step.

Check Item	Check Result
Database parameters	
Whether the source database contains unsupported table field types	Passed
Whether the destination database is compatible with the source database	Passed
Whether the character set of the source database matches that of the destination database	Passed
Whether the destination database has sufficient available connections	Passed
Whether the selected objects exist in the destination database	Passed
Whether the destination database contains the configured databases	Passed
Whether there are source database foreign keys	Passed
Whether tables to be migrated contain primary keys	Passed
Whether existing data meets the constraints	Passed
Whether the source database character set is supported	Passed
Whether the source database has sufficient available connections	Passed
Whether the source database container type is correct	Passed
Whether archive logs are enabled on the source database	Passed
Whether the source database name is valid	Passed
Whether the supplementary log is enabled for the source database.	Passed
Whether OGG log reading is enabled on the source database	Passed
Whether the source database table name is valid	Passed

- c. After all check items pass the pre-check, click **Next**.
11. Confirm the task.
- a. Check that all configured information is correct.

Start Time Start upon task creation Start at a specified time ⓘ

Send Notifications ⓘ If disabled, DMS alerts, such as task failure, high latency, and frozen, cannot be received.

Stop Abnormal Task After 14 ⓘ Abnormal tasks run longer than the period you set (unit: day) will automatically stop.

Details

Product Name	Configuration
Task Information	
Name	DMS task info
Description	Source Database IP Address or Domain Name: 10.154.210.68 Destination DB Instance Name: Auto-dts-gaussdb5-ge-1
Synchronization Mode	Full-Incremental synchronization
Data Flow	To the cloud

- b. Click **Submit**. In the display box, select **I have read the precautions**.
- c. Click **Submit**.

Notice



During the synchronization, do not perform any operations on the destination DB instance through the management console. To ensure migration success, we strongly recommend that you read the [migration precautions](#) carefully before starting migration tasks and follow the instructions to ensure migration stability.



If the task status is abnormal for more than 14 days, the task automatically stops. Pay attention to the alarms you received and handle the task in time to resume the download and avoid task retry failure.



I have read the precautions.

Submit

12. After the task is submitted, view and manage it.

After the task is created, return to the task list to view the status of the created task.

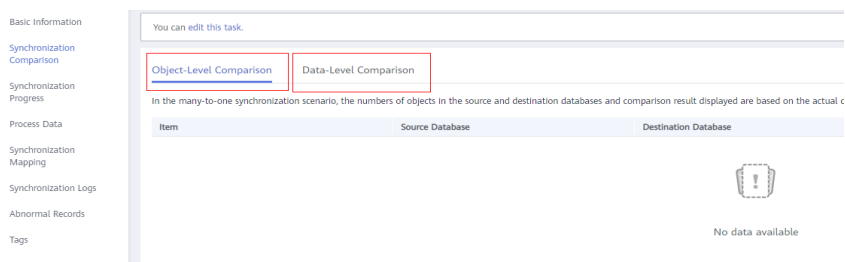
Step 5: Verify Data After Migration

When the task status changes to **Incremental**, the full synchronization is complete. You can log in to GaussDB and view the data migration result.

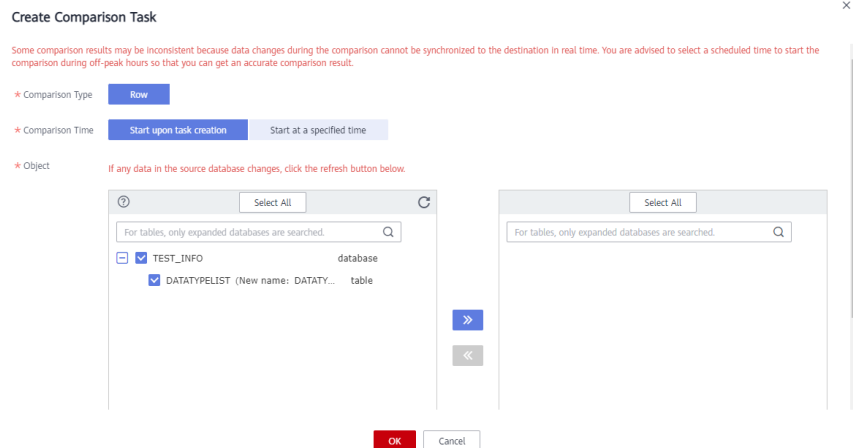
1. Wait until the migration task status becomes **Incremental**.



2. Click the task name to go to the **Basic Information** page.
3. Verify data consistency.
 - a. Choose **Synchronization Comparison** > **Object-Level Comparison** to view the database and table migration results.



- b. Choose **Synchronization Comparison** > **Data-Level Comparison**, click **Create Comparison Task**, and view the migration results of the rows in the table.



4. Connect to the **test_database_info** database in GaussDB using DAS.
For details about how to connect to an instance through DAS, see [Adding DB Instance Login Information](#).
5. Run the following statement to query the full synchronization result:

```
SELECT * FROM test_info.datatypelist_after;
```

After the schema in the MySQL database is migrated, it will be used as the schema in GaussDB. Therefore, it is required to add the schema in the query statement for exact query.

The query result shows that all data types in the table were successfully synchronized and the data is correct.
6. Verify incremental synchronization.

In full+incremental synchronization, after the full synchronization is complete, the data that is written to the source database after the task is created can still be synchronized to the destination database until the task is stopped. The following describes how to synchronize incremental data from the source database to the destination database:

 - a. Use a database connection tool to connect to the source MySQL database based on its IP address.
 - b. Run the following statement to insert a data record into the source database:
 Insert a data record whose ID is 3.

```
insert into test_info.test_table values
(3,'a','b','111','111','tinyblob','mediumblob','longblob','tinytext','text','mediumtext','longtext','1','3',1
,2,3,4,1.123,1.1234,'2024-03-08','2024-03-08 08:00:00','2024-03-08
08:00:00','08:00:00','2024','1010',{'a':"b"}',1.23,1.234);
```
 - c. Run the following statement in the destination database to query the result:

```
SELECT * FROM test_info.datatypelist_after;
```

The query result shows that new data in the source database has been synchronized to the destination database in real time.
7. Stop the migration task.

After data is completely migrated to the destination database, stop the synchronization task.

 - a. Locate the task and click **Stop** in the **Operation** column.

- b. In the display box, click **Yes**.

5.4 Migrating Data to GaussDB Using the Export and Import Functions of DAS

Scenarios

Data Admin Service (DAS) is a one-stop management platform that allows you to manage Huawei Cloud databases on a web console. It offers database development, O&M, and intelligent diagnosis, making it easy for you to use and maintain databases.

To back up or migrate data, you can use DAS to export data from the source database first and then import the data from your local PC or OBS bucket to the destination database.

For more information, see [Data Import and Export](#).

Constraints

- The file to be imported should be no larger than 1 GB.
- Only data files in the CSV or SQL format can be imported.
- Binary fields such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB are not supported.
- Data cannot be exported or imported using cross-region OBS buckets.

Exporting Data

Step 1 [Log in to the management console](#).

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

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

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

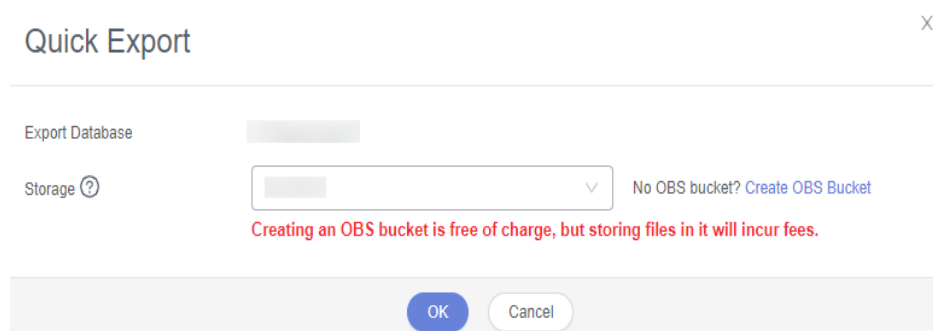
Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **Import and Export > Export**.

Step 7 On the displayed page, click **Create Task** and choose **Export Database** or **Export SQL Result** as required. The following takes database export as an example.

Alternatively, click **Quick Export** and select the target database. On the displayed page, select a storage path and click **OK**.

Figure 5-21 Quick export



Quick Export

Export Database: [text box]

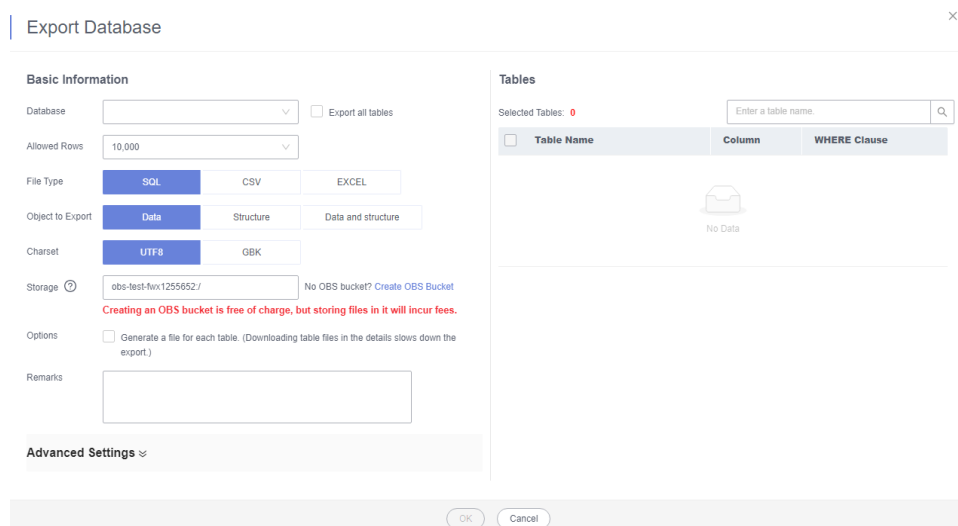
Storage: [dropdown menu] No OBS bucket? [Create OBS Bucket](#)

Creating an OBS bucket is free of charge, but storing files in it will incur fees.

OK Cancel

Step 8 On the displayed page, set parameters as required in areas **Basic Information** and **Advanced Settings**. Then, select the tables to be exported on the right.

Figure 5-22 Creating an export task



Export Database

Basic Information

Database: [dropdown menu] ☐ Export all tables

Allowed Rows: 10,000 [dropdown menu]

File Type: **SQL** CSV EXCEL

Object to Export: **Data** Structure Data and structure

Charset: **UTF8** GBK

Storage: [text box] No OBS bucket? [Create OBS Bucket](#)

Creating an OBS bucket is free of charge, but storing files in it will incur fees.

Options: ☐ Generate a file for each table. (Downloading table files in the details slows down the export.)

Remarks: [text box]

Advanced Settings

Tables

Selected Tables: 0 [text box] [search icon]

Table Name	Column	WHERE Clause
No Data		

OK Cancel

NOTE

- In a SQL result export task, the executed SQL statements cannot exceed 5 MB.
- Databases are classified as user databases or system databases. System databases cannot be exported. If system database data is required, deploy system database services in a created user database, so that you can export the system database data from the user database.
- DAS connects to your standby database to export data. This prevents the primary database from being affected by data export. However, if the standby database has a high replication delay, the exported data may not be the latest.

Step 9 After settings are complete, click **OK**.

Step 10 In the task list, view the task ID, type, status, and progress.

Step 11 Click **Details** in the **Operation** column to view task details.

Figure 5-23 Task list

Create Task		Quick Export		Delete Task		Time Range		Start Date		End Date		Search		Q	
☐	Task ID	Task Type	Database	Started	Ended	File Size	File Type	Status	Elapsed Time	Exports of Rows	Progress	Remarks	Operation		
☐	c48d5366ad427db5c368d8682021	Quick E...	db_01	2020-09-07 20:16:45	2020-09-07 20:16:55	4.53 MB	SQL	Successful	10 secs.	202415	100%		Details Download		
☐	c48d53765847c7a3b437f58a7e7a2	Database	create_new_db1	2020-09-03 16:50:45	2020-09-03 16:52:14	16.36 MB	SQL	Successful	1 min.	10000	100%		Details Download		
☐	7d566a2f56b4a6789966c3db56b61ca	Database	create_new_db1	2020-09-03 16:47:05	2020-09-03 16:47:22	3.94 MB	SQL	Successful	17 secs.	2414	100%		Details Download		

----End

Importing Data

Step 1 On the top menu bar, choose **Import and Export > Import**.

Step 2 Import a file from your local PC or an OBS bucket.

Figure 5-24 Creating an import task

Create Task

X

Import Type

sql

CSV

File Source

Upload file

Choose from OBS

Attachment Storage ?

407154

▼

No OBS bucket? [Create OBS Bucket](#)

Creating an OBS bucket is free of charge, but storing files in it will incur fees.

Attachment

+

Click here to upload a file, or drag one here. (.sql)

Upload only one attachment that is no larger than 1 GB.

Database

db_4eb3_0000

▼

Charset

Auto Detect

UTF8

GBK

Options

☐ Ignore errors, that is, skip the step where the SQL statement fails to be executed.

☒ Delete the uploaded file upon an import success.

Remarks

Create

Cancel

- From your local PC
In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Upload file** for **File Source**, set the attachment storage, and upload the file. Then, set other parameters as required.
For security purposes, imported files are stored in OBS buckets.

 NOTE

- To keep your data secure, provide your own OBS bucket to store the attachments you upload. In this way, DAS automatically connects to your OBS bucket for in-memory reading.
 - If you select **Delete the uploaded file upon an import success**, the file you uploaded will be automatically deleted from the OBS bucket after being imported to the destination database.
- From an OBS bucket
In the upper left corner, click **Create Task**. On the displayed page, select an import type, select **Choose from OBS** for **File Source**, and select a file from the bucket. Then, set other parameters as required.

 NOTE

The file uploaded from an OBS bucket will not be deleted upon an import success.

Step 3 After setting the import parameters, click **Create**. Confirm the information again before you click **OK** because original data may be overwritten after data import.

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

----End

5.5 Using the copy to/from Command to Export and Import Data

Scenarios

The gsql tool provides the `\copy` meta-command to import or export data. `\copy` applies only to small-scale data import in good format. It does not preprocess invalid characters or provide error tolerance. Therefore, `\copy` cannot be used in scenarios where abnormal data exists.

Preparing for Data Migration

1. Prepare an ECS or a device that can access the GaussDB instance over EIP.
 - To connect to a GaussDB instance through an ECS, you must first create an ECS.
For details on how to create and log in to an ECS, see [Purchasing an ECS](#) and [Logging In to an ECS](#) in *Elastic Cloud Server Getting Started*.
 - To connect to a GaussDB instance through a device that can access the GaussDB instance over EIP, you must:
 - i. Bind an EIP to the GaussDB instance. For details, see [Binding an EIP](#).
 - ii. Ensure that the local device can access the EIP that has been bound to the GaussDB instance.
2. Install the gsql client on the prepared ECS or device that can access the GaussDB database, and connect it to the GaussDB instance. For details, see [Using gsql to Connect to a Database](#).

Exporting Data

On the prepared ECS or device that can access GaussDB, connect to the GaussDB instance and export the content of the **copy_example** table.

- Method 1: Export the content of the **copy_example** table to **stdout** in CSV format. Use quotation marks (") as the quotes, and use the quotes to enclose the fourth and fifth columns.

```
\copy copy_example to stdout CSV quote as '"' force quote col_4,col_5;
1,iamtext,iamvarchar,"2006-07-07","12:00:00"
2,sometext,somevarchar,"2006-07-07","12:00:00"
3,sometext,somevarchar,"2006-07-07","12:00:00"
4,sometext,somevarchar,"2022-07-07","19:00:02"
5,sometext,somevarchar,"2006-07-07",
6,sometext,somevarchar,"2022-07-07","19:00:02"
```

- Method 2: Export the content of the **copy_example** table to the **copy_example.csv** file under the local path **/tmp/data/**. Use vertical bars (|) as the delimiters and quotation marks (") as the quotes.

```
\copy copy_example to '/tmp/data/copy_example.csv' csv delimiter '|' quote '"';
```

Check the **/tmp/data/copy_example.csv** file to ensure that the data has been exported.

```
1|iamtext|iamvarchar|2006-07-07|12:00:00
2|sometext|somevarchar|2006-07-07|12:00:00
3|sometext|somevarchar|2006-07-07|12:00:00
4|sometext|somevarchar|2022-07-07|19:00:02
5|sometext|somevarchar|2006-07-07|
6|sometext|somevarchar|2022-07-07|19:00:02
```

- Method 3: Export the query result set of the **copy_example** table to the **copy_example2.csv** file under the local path **/tmp/data/**. Use commas (,) as the delimiters and quotation marks (") as the quotes.

```
\copy (select * from copy_example where col_1 = 1) to '/tmp/data/copy_example2.csv' csv delimiter ',' quote '"';
```

Check the **/tmp/data/copy_example2.csv** file to ensure that the data has been exported.

```
1,iamtext,iamvarchar,2006-07-07,12:00:00
```

Importing Data

Import data to a GaussDB instance. For example, import data to the target table **copy_example**, the schema is as follows:

```
create table copy_example
(
  col_1 integer,
  col_2 text,
  col_3 varchar(12),
  col_4 date,
  col_5 time
);
```

On the prepared ECS or device that can access GaussDB, connect to the GaussDB instance and import data to the target table **copy_example**.

- Method 1: Copy data from **stdin** and import data to the target table **copy_example**.

```
\copy copy_example from stdin csv;
```

When **>>** is displayed, enter data. To end your input, enter a backslash and a period (**\.**).

Enter data to be copied followed by a newline.
End with a backslash and a period on a line by itself.

```
>> 1,"iamtext","iamvarchar",2006-07-07,12:00:00
>> 2,"sometext","somevarchar",2006-07-07,12:00:00
>> \.
```

View the imported data.

```
select * from copy_example;
col_1 | col_2 | col_3 | col_4 | col_5
-----+-----+-----+-----+-----
 1 | iamtext | iamvarchar | 2006-07-07 00:00:00 | 12:00:00
 2 | sometext | somevarchar | 2006-07-07 00:00:00 | 12:00:00
(2 rows)
```

- Method 2: The **example.csv** file exists in the local **/tmp/data/** directory. The file contains the header row. Use vertical bars (|) as the delimiters and quotation marks (") as the quotes. The content is as follows:

```
header
3|"sometext"|"somevarchar"|2006-07-07|12:00:00
4|"sometext"|"somevarchar"|2022-07-07|19:00:02
```

Import data from the local file **example.csv** to the target table **copy_example**. If **header** is set to **on**, the first row is automatically ignored. Use quotation marks (") as the quotes by default.

```
\copy copy_example from '/tmp/data/example.csv' with(header 'on', format 'csv', delimiter '|',
date_format 'yyyy-mm-dd', time_format 'hh24:mi:ss');
```

View the imported data.

```
select * from copy_example;
col_1 | col_2 | col_3 | col_4 | col_5
-----+-----+-----+-----+-----
 1 | iamtext | iamvarchar | 2006-07-07 00:00:00 | 12:00:00
 2 | sometext | somevarchar | 2006-07-07 00:00:00 | 12:00:00
 3 | sometext | somevarchar | 2006-07-07 00:00:00 | 12:00:00
 4 | sometext | somevarchar | 2022-07-07 00:00:00 | 19:00:02
(4 rows)
```

- Method 3: The **example2.csv** file exists in the local directory **/tmp/data/**. Use commas (,) as the delimiters and quotation marks (") as the quotes. The last field is missing in the first line, and the last field is added in the second line. The content is as follows:

```
5,"sometext","somevarchar",2006-07-07
6,"sometext","somevarchar",2022-07-07,19:00:02,12:00:00
```

Import data from the local file **example2.csv** to the target table **copy_example**. The default delimiters are commas (,). Because the error tolerance parameters **IGNORE_EXTRA_DATA** and **FILL_MISSING_FIELDS** are specified, the missing fields are replaced with **NULL**, and the extra fields are ignored.

```
\copy copy_example from '/tmp/data/example2.csv' with( format 'csv', date_format 'yyyy-mm-dd',
time_format 'hh24:mi:ss', IGNORE_EXTRA_DATA 'true', FILL_MISSING_FIELDS 'true');
```

View the imported data.

```
select * from copy_example;
col_1 | col_2 | col_3 | col_4 | col_5
-----+-----+-----+-----+-----
 1 | iamtext | iamvarchar | 2006-07-07 00:00:00 | 12:00:00
 2 | sometext | somevarchar | 2006-07-07 00:00:00 | 12:00:00
 3 | sometext | somevarchar | 2006-07-07 00:00:00 | 12:00:00
 4 | sometext | somevarchar | 2022-07-07 00:00:00 | 19:00:02
 5 | sometext | somevarchar | 2006-07-07 00:00:00 | 
 6 | sometext | somevarchar | 2022-07-07 00:00:00 | 19:00:02
(6 rows)
```

Helpful Links

For more information, see:

- [COPY \(distributed\)](#)
- [COPY \(centralized\)](#)

5.6 Using CopyManager in JDBC to Export and Import Data

Scenarios

When you use Java to develop applications, the CopyManager API of the JDBC driver is called to write data from files or other source databases to GaussDB.

Example 1: Importing and Exporting Data Through Local Files

Call the CopyManager API of the JDBC driver to export data from the database to a local file or import a local file to the database in streaming mode. The file format can be CSV or TEXT.

The sample program is as follows. Load the GaussDB JDBC driver before executing it.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.io.IOException;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.sql.SQLException;
import org.postgresql.copy.CopyManager;
import org.postgresql.core.BaseConnection;

public class Copy{

    public static void main(String[] args)
    {
        String urls = new String("jdbc:postgresql://localhost:8000/postgres"); // URL of the database
        String username = System.getenv("EXAMPLE_USERNAME_ENV"); // Username
        String password = System.getenv("EXAMPLE_PASSWORD_ENV"); // Password
        String tablename = new String("migration_table"); // Define table information.
        String tablename1 = new String("migration_table_1"); // Define table information.
        String driver = "org.postgresql.Driver";
        Connection conn = null;

        try {
            Class.forName(driver);
            conn = DriverManager.getConnection(urls, username, password);
        } catch (ClassNotFoundException e) {
            e.printStackTrace(System.out);
        } catch (SQLException e) {
            e.printStackTrace(System.out);
        }

        // Export data from the migration_table table to the local d:/data.txt file.
        try {
            copyToFile(conn, "d:/data.txt", "(SELECT * FROM migration_table)");
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        // Import data from the d:/data.txt file to the migration_table_1 table.
```

```
try {
    copyFromFile(conn, "d:/data.txt", tablename1);
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

// Export data from the migration_table_1 table to the local d:/data1.txt file.
try {
    copyToFile(conn, "d:/data1.txt", tablename1);
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

public static void copyFromFile(Connection connection, String filePath, String tableName)
    throws SQLException, IOException {

    FileInputStream fileInputStream = null;

    try {
        CopyManager copyManager = new CopyManager((BaseConnection)connection);
        fileInputStream = new FileInputStream(filePath);
        copyManager.copyIn("COPY " + tableName + " FROM STDIN ", fileInputStream);
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

public static void copyToFile(Connection connection, String filePath, String tableOrQuery)
    throws SQLException, IOException {

    FileOutputStream fileOutputStream = null;

    try {
        CopyManager copyManager = new CopyManager((BaseConnection)connection);
        fileOutputStream = new FileOutputStream(filePath);
        copyManager.copyOut("COPY " + tableOrQuery + " TO STDOUT", fileOutputStream);
    } finally {
        if (fileOutputStream != null) {
            try {
                fileOutputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
}
```

Example 2: Migrating Data from Database B

The following example shows how to use CopyManager to migrate data from database B to GaussDB.

```
import java.io.StringReader;
import java.sql.Connection;
```

```
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.postgresql.copy.CopyManager;
import org.postgresql.core.BaseConnection;

public class Migration{

    public static void main(String[] args) {
        String url = new String("jdbc:postgresql://localhost:8000/postgres"); //URL of the database
        String user = new String("username"); // GaussDB database username
        String pass = new String("passwd"); // GaussDB database password
        String tablename = new String("migration_table_1"); // Define table information.
        String delimiter = new String(","); // Define a delimiter.
        String encoding = new String("UTF8"); // Define a character set.
        String driver = "org.postgresql.Driver";
        StringBuffer buffer = new StringBuffer(); // Define a buffer to store formatted data.

        try {
            // Obtain the query result set of the source database.
            ResultSet rs = getDataSet();

            // Traverse the result set and obtain records row by row.
            // The values of columns in each record are separated by the specified delimiter and end with a
            // newline character to form strings.
            // Add the strings to the buffer.
            while (rs.next()) {
                buffer.append(rs.getString(1) + delimiter
                    + rs.getString(2) + delimiter
                    + rs.getString(3) + delimiter
                    + rs.getString(4)
                    + "\n");
            }
            rs.close();

            try {
                // Connect to the destination database.
                Class.forName(driver);
                Connection conn = DriverManager.getConnection(url, user, pass);
                BaseConnection baseConn = (BaseConnection) conn;
                baseConn.setAutoCommit(false);

                // Initialize table information.
                String sql = "Copy " + tablename + " from STDIN with (DELIMITER " + "'" + delimiter + "'" + " "
                + "ENCODING " + "'" + encoding + "'");

                // Commit data in the buffer.
                CopyManager cp = new CopyManager(baseConn);
                StringReader reader = new StringReader(buffer.toString());
                cp.copyIn(sql, reader);
                baseConn.commit();
                reader.close();
                baseConn.close();
            } catch (ClassNotFoundException e) {
                e.printStackTrace(System.out);
            } catch (SQLException e) {
                e.printStackTrace(System.out);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    //*****
    // Return the query result set from the source database.
    //*****
}
```

```
private static ResultSet getDataSet() {
    ResultSet rs = null;
    try {
        Class.forName("com.B.jdbc.Driver").newInstance();
        Connection conn = DriverManager.getConnection("jdbc:MY://10.119.179.227:3306/jack?
useSSL=false&allowPublicKeyRetrieval=true", "jack", "xxxxxxx");
        Statement stmt = conn.createStatement();
        rs = stmt.executeQuery("select * from migration_table");
    } catch (SQLException e) {
        e.printStackTrace();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return rs;
}
```

Helpful Links

For more information, see:

- [CopyManager \(distributed\)](#)
- [CopyManager \(centralized\)](#)

5.7 Using gs_dump and gs_dumpall to Export Data

Scenarios

GaussDB provides `gs_dump` and `gs_dumpall` to export required database objects and related information. You can use a tool to import the exported data to a destination database for database migration. `gs_dump` can export a single database or its objects. `gs_dumpall` can export all databases or global objects in the database. For details, see [Table 5-11](#).

NOTE

In the multitenancy scenario, `gs_dump` can be used to export a single PDB or its objects, but `gs_dumpall` cannot support this scenario.

Table 5-11 Scenarios

Scenario	Export Granularity	Export Format	Import Method
Exporting a single database	Database-level export (see Exporting a Database). - Export full information of a database. You can use the exported information to create the same database containing the same data. - Export all object definitions of a database, including the definitions of the database, functions, schemas, tables, indexes, and stored procedures. You can use the exported object definitions to quickly create the same database, without data. - Export data of a database.	- Plain-text - Custom - Directory - TAR	- Use <code>gsql</code> to import plain-text data files. For details, see gsql in <i>Tool Reference</i>. - For details about how to import data files in .tar, directory, or custom format, see Using gs_restore to Import Data.
	Schema-level export (see Exporting a Schema). - Export full information of a schema. - Export data of a schema. - Export all object definitions of a schema, including the definitions of tables, stored procedures, and indexes.		
	Table-level export (see Exporting a Table). - Export full information of a table. - Export data of a table. - Export the definition of a table.		

Scenario	Export Granularity	Export Format	Import Method
Exporting all databases	Database-level export (see Exporting All Databases). - Export full information of databases. You can use the exported full information to create the same host environment containing the same databases and public global objects, with the same data. - Export all object definitions of databases, including the definitions of tablespaces, databases, functions, schemas, tables, indexes, and stored procedures. You can use the exported object definitions to quickly create the same host environment containing the same databases and tablespaces but without data. - Export data of databases.	Plain-text	For details about how to import data files, see Using copy from to Import Data .
	Public global object export (see Exporting Global Objects). - Export tablespaces. - Export roles. - Export tablespaces and roles.		

gs_dump and gs_dumpall use **-U** to specify the user that performs the export. If the specified user does not have the required permissions, data cannot be exported. For details about the scenarios where this function can be used, see [Table 5-11](#).

Precautions

gs_dump and gs_dumpall encrypt the exported data files. These files are decrypted before being imported to prevent data disclosure for higher database security. Note that gsql cannot decrypt and import stored procedures and functions for plain-text files encrypted using gs_dump. Therefore, if the exported database contains stored procedures or functions, use the other three modes to export the database and use gs_restore to restore the database.

When gs_dump or gs_dumpall is used to export data, other users can still operate (read or write) the database.

gs_dump and gs_dumpall can export complete, consistent data. For example, if gs_dump exports data from database A or gs_dumpall exports data from GaussDB database at T1, the exported data is the data status of database A or GaussDB database at T1. Modified data of database A or GaussDB database after T1 will not be exported.

When gs_dump or gs_dumpall is used to export data, generated columns are not dumped.

- Do not modify the files and contents exported using the **-F c/d/t** format. Otherwise, the restoration may fail. For files exported using the **-F p** format, edit the exported files with caution if necessary.
- If your database contains over 500,000 objects (data tables, views, and indexes), [submit a service ticket](#) to contact Huawei Cloud technical support to improve performance and avoid memory problems.
- To ensure data consistency and integrity, the export tools will set a shared lock for the tables to be dumped. If a shared lock has been set for the table in other transactions, gs_dump and gs_dumpall lock the table after it is released. If the table cannot be locked within the specified time, the dump fails. You can customize the timeout duration to wait for lock release by specifying the **--lock-wait-timeout** parameter.
- During an export, gs_dumpall reads tables in all databases. Therefore, you need to connect to the databases as a database administrator to export a complete file. When you use gsql to execute scripts, administrator permissions are also required so as to add users and user groups, and create databases.

Preparing for Data Migration

1. Prepare an ECS or a device that can access the GaussDB instance over EIP.
 - To connect to a GaussDB instance through an ECS, you must first create an ECS.
For details on how to create and log in to an ECS, see [Purchasing an ECS](#) and [Logging In to an ECS](#) in *Elastic Cloud Server Getting Started*.
 - To connect to a GaussDB instance through a device that can access the GaussDB instance over EIP, you must:
 - i. Bind an EIP to the GaussDB instance. For details, see [Binding an EIP](#).
 - ii. Ensure that the local device can access the EIP that has been bound to the GaussDB instance.
2. Install the gsql client on the prepared ECS or device that can access the GaussDB database, and connect it to the GaussDB database instance. For details, see [Using gsql to Connect to a Database](#).

Exporting a Database

1. Create the database and table to be exported and insert data into them.

```
create database gs_example;

\c gs_example
password:

create schema gs_sch_example;
set search_path to gs_sch_example;
create table gs_table_example
```

```
(
col_1 integer,
col_2 text,
col_3 varchar(12),
col_4 date,
col_5 time
);
insert into gs_table_example values(1,'iamtext','iamvarchar','2006-07-07','12:00:00');
insert into gs_table_example values(2,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_table_example values(3,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_table_example values(4,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_table_example values(5,'sometext','somevarchar','2006-07-07', null);
insert into gs_table_example values(6,'sometext','somevarchar','2006-07-07','19:00:02');
```

2. Use **gs_dump** to export data of the **gs_example** database.

- Example 1: Use **gs_dump** to export full information of the **gs_example** database by specifying the database IP address. The exported files are in .sql format.

```
gs_dump -U root -f /tmp/data/gs_example_dump.sql -p 8000 gs_example -F p -h 192.*.*.139;
Password:
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:04:20]: The total
objects number is 458.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:04:20]: [100.00%]
458 objects have been dumped.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:04:20]: dump
database gs_example successfully
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:04:20]: total time:
8779 ms
```

- Example 2: Use **gs_dump** to export full information of the **gs_example** database by specifying the database IP address. The exported information is archived to the **/tmp/data/gs_example_dump.tar** file in .tar format.

```
gs_dump -U root -f /tmp/data/gs_example_dump.tar -p 8000 gs_example -F t -h 192.*.*.139;
Password:
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 14:58:49]: The total
objects number is 458.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 14:58:49]: [100.00%]
458 objects have been dumped.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 14:58:49]: dump
database gs_example successfully
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 14:58:49]: total time:
8201 ms
```

- Example 3: Use **gs_dump** to export data of the **gs_example** database by specifying the database IP address. The exported data does not contain object definitions of the database. The exported files are in custom format.

```
gs_dump -U root -f /tmp/data/gs_example_dump.dmp -p 8000 gs_example -a -F c -h
192.*.*.139;
Password:
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:07:23]: dump
database gs_example successfully
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:07:23]: total time:
8369 ms
```

- Example 4: Use **gs_dump** to export all object definitions of the **gs_example** database by specifying the database IP address. The exported files are in .sql format.

```
gs_dump -U root -f /tmp/data/gs_example_dump.s.sql -p 8000 gs_example -s -F p -h
192.*.*.139;
Password:
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:09:37]: The total
objects number is 457.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:09:37]: [100.00%]
457 objects have been dumped.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:09:37]: dump
database gs_example successfully
```

```
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:09:37]: total time: 8523 ms
```

- Example 5: Use `gs_dump` to export all object definitions of the **gs_example** database by specifying the database IP address. The exported files are encrypted in .txt format.

```
gs_dump -U root -f /tmp/data/gs_example_dump_s_key.sql -p 8000 gs_example --with-encryption AES128 --with-key abcdefg_?1234567 -s -F p -h 192.*.*.139;
```

Password:

```
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:10:38]: The total objects number is 457.
```

```
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:10:38]: [100.00%] 457 objects have been dumped.
```

```
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:10:38]: dump database gs_example successfully
```

```
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:10:38]: total time: 9101 ms
```

Table 5-12 Common parameters

Parameter	Description	Example
-U	Username for database connection. NOTE If the username for connecting to the database is not specified, the initial system administrator created during installation is used for connection by default.	-U jack
-W	User password for database connection. • This parameter is not required for database administrators if the trust policy is used for authentication. • If you connect to the database without specifying this parameter and you are not a database administrator, you will be prompted to enter the password.	-W *****
-f	Folder to store exported files. If this parameter is not specified, the exported files are stored in the standard output. If the output format is (-F c/-F d/-F t) , the -f parameter must be specified.	-f /home/omm/backup/ <i>MPPDB_backup.tar</i>

Parameter	Description	Example
-p	TCP port or local Unix-domain socket file name extension on which the server is listening for connections.	-p 8000
dbname	Name of the database to be exported.	testdb
-F	Format of exported files. The values are as follows: • p: plain-text • c: custom • d: directory • t: TAR	-F t

Exporting a Schema

1. Create a schema to be exported and insert data into it.

```
create database gs_example;

\c gs_example
password:

create schema gs_sch_example;
create schema gs_sch_1_example;
create table gs_sch_example.gs_table_example
(
  col_1 integer,
  col_2 text,
  col_3 varchar(12),
  col_4 date,
  col_5 time
);
create table gs_sch_1_example.gs_table_example
(
  col_1 integer,
  col_2 text,
  col_3 varchar(12),
  col_4 date,
  col_5 time
);
insert into gs_sch_example.gs_table_example values(1,'iamtext','iamvarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example
values(2,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example
values(3,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example
values(4,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_sch_example.gs_table_example values(5,'sometext','somevarchar','2006-07-07', null);
insert into gs_sch_example.gs_table_example
values(6,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_sch_1_example.gs_table_example values(7,'iamtext','iamvarchar','2006-07-07','12:00:00');
insert into gs_sch_1_example.gs_table_example
values(8,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_1_example.gs_table_example
values(9,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_1_example.gs_table_example
values(10,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_sch_1_example.gs_table_example values(11,'sometext','somevarchar','2006-07-07', null);
```

```
insert into gs_sch_1_example.gs_table_example  
values(12,'sometext','somevarchar','2006-07-07','19:00:02');
```

2. Use `gs_dump` to export schemas from the **gs_example** database at the same time.
 - Example 1: Use `gs_dump` to export the **gs_sch_example** and **gs_sch_1_example** schemas at the same time by specifying the database IP address. The exported files are in directory format.

```
gs_dump -U root -f /tmp/data/gs_sch_dump -p 8000 gs_example -n gs_sch_example -n  
gs_sch_1_example -F d -h 192.*.*.139;  
Password:  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:37:11]: The total  
objects number is 460.  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:37:11]: [100.00%]  
460 objects have been dumped.  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:37:11]: dump schema  
gs_sch_example gs_sch_1_example successfully  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:37:11]: dump  
database gs_example successfully  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:37:11]: total time:  
9602 ms
```
 - Example 2: Use `gs_dump` to export full information of the **gs_sch_example** schema by specifying the database IP address. The exported files are in .txt format.

```
gs_dump -U root -f /tmp/data/gs_sch_dump.sql -p 8000 gs_example -n gs_sch_example -F p -h  
192.*.*.139;  
Password:  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:39:00]: The total  
objects number is 457.  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:39:00]: [100.00%]  
457 objects have been dumped.  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:39:00]: dump schema  
gs_sch_example successfully  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:39:00]: dump  
database gs_example successfully  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:39:00]: total time:  
8582 ms
```
 - Example 3: Use `gs_dump` to export data from the **gs_example** database by specifying the database IP address. The exported data does not contain the **gs_sch_example** schemas. The exported files are in custom format.

```
gs_dump -U root -f /tmp/data/gs_sch_dump.dmp -p 8000 gs_example -N gs_sch_example -F c -  
h 192.*.*.139;  
Password:  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:41:14]: The total  
objects number is 458.  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:41:14]: [100.00%]  
458 objects have been dumped.  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:41:14]: dump  
database gs_example successfully  
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:41:14]: total time:  
8323 ms
```

Table 5-13 Common parameters

Parameter	Description	Example
-U	Username for database connection.	-U jack

Parameter	Description	Example
-W	User password for database connection. - This parameter is not required for database administrators if the trust policy is used for authentication. - If you connect to the database without specifying this parameter and you are not a database administrator, you will be prompted to enter the password.	-W *****
-f	Folder to store exported files. If this parameter is not specified, the exported files are stored in the standard output.	-f /home/omm/backup/MPPDB_schema_backup
-p	TCP port or local Unix-domain socket file name extension on which the server is listening for connections.	-p 8000
dbname	Name of the database to be exported.	human_resource
-n	Names of schemas to be exported. This option contains the schema and all its contained objects. - Single schema: Enter -n <i>schemaname</i>. - Multiple schemas: Enter -n <i>schemaname</i> for each schema.	- Single schemas: -n hr - Multiple schemas: -n hr -n public
-F	Format of exported files. The values are as follows: p: plain-text c: custom d: directory t: TAR	-F d

Exporting a Table

1. Create a schema to be exported and insert data into it.

```
create database gs_example;
```

```
\c gs_example
password:

create schema gs_sch_example;
create table gs_sch_example.gs_table_example
(
  col_1 integer,
  col_2 text,
  col_3 varchar(12),
  col_4 date,
  col_5 time
);
create table gs_sch_example.gs_table_example_2
(
  col_1 integer,
  col_2 text,
  col_3 varchar(12),
  col_4 date,
  col_5 time
);
insert into gs_sch_example.gs_table_example values(1,'iamtext','iamvarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example values(2,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example values(3,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example values(4,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_sch_example.gs_table_example values(5,'sometext','somevarchar','2006-07-07', null);
insert into gs_sch_example.gs_table_example values(6,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_sch_example.gs_table_example_2 values(7,'iamtext','iamvarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example_2 values(8,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example_2 values(9,'sometext','somevarchar','2006-07-07','12:00:00');
insert into gs_sch_example.gs_table_example_2 values(10,'sometext','somevarchar','2006-07-07','19:00:02');
insert into gs_sch_example.gs_table_example_2 values(11,'sometext','somevarchar','2006-07-07', null);
insert into gs_sch_example.gs_table_example_2 values(12,'sometext','somevarchar','2006-07-07','19:00:02');
```

2. Use `gs_dump` to export the **gs_sch_example.gs_table_example** and **gs_sch_example.gs_table_example_2** tables at the same time.

NOTE

1. In the following example, after the export, ensure that the schema to which the exported table belongs exists before the import.
- Example 1: Use `gs_dump` to export the **gs_sch_example.gs_table_example** and **gs_sch_example.gs_table_example_2** tables at the same time by specifying the database IP address. The exported files are in directory format.

```
gs_dump -U root -f /tmp/data/gs_table_dump -p 8000 gs_example -t
gs_sch_example.gs_table_example -t gs_sch_example.gs_table_example_2 -F d -h 192.*.*.139;
Password:
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:49:06]: The total
objects number is 458.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:49:06]: [100.00%]
458 objects have been dumped.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:49:06]: dump table
gs_sch_example.gs_table_example gs_sch_example.gs_table_example_2 successfully
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:49:06]: dump
database gs_example successfully
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:49:06]: total time:
7694 ms
```

- Example 2: Use `gs_dump` to export the tables excluding the **gs_sch_example.gs_table_example_2** table by specifying the database IP address. The exported files are in custom format.

```
gs_dump -U root -f /tmp/data/gs_table_dump.dmp -p 8000 gs_example -T
gs_sch_example.gs_table_example_2 -F c -h 192.*.*.139;
Password:
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:52:07]: The total
objects number is 461.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:52:07]: [100.00%]
461 objects have been dumped.
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:52:07]: dump
database gs_example successfully
gs_dump[user='root'][localhost][port='8000'][gs_example][2024-07-26 15:52:07]: total time:
8203 ms
```

Table 5-14 Common parameters

Parameter	Description	Example
-U	Username for database connection.	-U jack
-W	User password for database connection. • This parameter is not required for database administrators if the trust policy is used for authentication. • If you connect to the database without specifying this parameter and you are not a database administrator, you will be prompted to enter the password.	-W *****
-f	Folder to store exported files. If this parameter is not specified, the exported files are stored in the standard output.	-f /home/omm/backup/MPPDB_table_backup
-p	TCP port or local Unix-domain socket file name extension on which the server is listening for connections.	-p 8000
dbname	Name of the database to be exported.	human_resource

Parameter	Description	Example
-t	Tables (or views, sequences, foreign tables) to export. You can specify multiple tables by listing them or using wildcard characters. When you use wildcard characters, quote the pattern to prevent the shell from expanding the wildcard characters. - Single table: Enter -t <i>schema.table</i>. - Multiple tables: Enter -t <i>schema.table</i> for each table.	- Single table: -t <i>hr.staffs</i> - Multiple tables: -t <i>hr.staffs</i> -t <i>hr employments</i>
-F	Format of exported files. The values are as follows: p: plain-text c: custom d: directory t: TAR	-F d
-T	A list of tables, views, sequences, or foreign tables not to be dumped. You can use multiple -t parameters or wildcard characters to specify tables. When -t and -T are input, the object will be stored in -t list not -T table object.	-T <i>table1</i>

Exporting All Databases

Use `gs_dumpall` to export all database information at a time.

- Example 1: Use `gs_dumpall` to export all database information by specifying the database IP address. The exported file is in `.sql` format. After the command is executed, a large amount of output information will be displayed. **total time** will be displayed at the end of the information, indicating that the command is executed successfully. In this example, only relative output information is included.

```
gs_dumpall -U root -f /tmp/data/dumpall.sql -p 8000 -h 192.**.139;  
Password:
```

```
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:02:15]: dumpall operation successful  
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:02:15]: total time: 35133 ms
```

- Example 2: Use `gs_dumpall` to export all database definitions by specifying the database IP address. The exported file is in `.sql` format. After the command is executed, a large amount of output information will be displayed. **total time** will be displayed at the end of the information,

indicating that the command is executed successfully. In this example, only relative output information is included.

```
gs_dumpall -U root -f /tmp/data/dumpall_def.sql -p 8000 -s -h 192.*.*.139;
```

Password:

```
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:07:50]: dumpall operation successful
```

```
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:07:50]: total time: 21239 ms
```

Table 5-15 Common parameters

Parameter	Description	Example
-U	Username for database connection. The user must be a database administrator.	-U omm
-W	User password for database connection. - This parameter is not required for database administrators if the trust policy is used for authentication. - If you connect to the database without specifying this parameter and you are not a database administrator, you will be prompted to enter the password.	-W *****
-f	Folder to store exported files. If this parameter is not specified, the exported files are stored in the standard output.	-f /home/omm/backup/MPPDB_backup.sql
-p	TCP port or local Unix-domain socket file name extension on which the server is listening for connections.	-p 8000

Exporting Global Objects

Use `gs_dumpall` to export tablespace object information.

- Example 1: Use `gs_dumpall` to export the global tablespace and user information of all databases by specifying the database IP address. The exported files are in .sql format. In this example, only relative output information is included.

```
gs_dumpall -U root -f /tmp/data/dumpall_tablespace.sql -p 8000 -t -h 192.*.*.139;
```

Password:

```
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:10:42]: dumpall operation successful
```

```
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:10:42]: total time: 1800 ms
```

- Example 2: Use `gs_dumpall` to export the global user information of all databases by specifying the database IP address. The exported files are in .txt format. In this example, only relative output information is included.

```
gs_dumpall -U root -f /tmp/data/dumpall_user.sql -p 8000 -r -h 192.*.*.139;
```

Password:

```
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:12:15]: dumpall operation successful  
gs_dumpall[user='root'][localhost][port='8000'][2024-07-26 16:12:15]: total time: 1269 ms
```

Table 5-16 Common parameters

Parameter	Description	Example
-U	Username for database connection. The user must be a database administrator.	-U omm
-W	User password for database connection. - This parameter is not required for database administrators if the trust policy is used for authentication. - If you connect to the database without specifying this parameter and you are not a database administrator, you will be prompted to enter the password.	-W *****
-f	Folder to store exported files. If this parameter is not specified, the exported files are stored in the standard output.	-f /home/omm/backup/ <i>MPPDB_tablespace.sql</i>
-p	TCP port or local Unix-domain socket file name extension on which the server is listening for connections.	-p 8000
-t	Dumping only tablespaces. You can also use --tablespaces-only alternatively.	-t

Helpful Links

For more information, see:

- [gs_dump \(distributed\)](#)
- [gs_dump \(centralized\)](#)
- [gs_dumpall \(distributed\)](#)
- [gs_dumpall \(centralized\)](#)

5.8 Using gs_restore to Import Data

Scenarios

gs_restore is an import tool provided by GaussDB and works together with gs_dump. You can use gs_restore to import the files exported by gs_dump to a database. gs_restore can import the files in .tar, custom, or directory format.

gs_restore can:

- Import data to a database.
If a database is specified, data is imported to the database. If parallel import is enabled, the password for connecting to the database also needs to be specified. During data import, the generated columns are automatically updated and saved as common columns.
- Import data to a script file.
If no database is specified, a script containing the SQL statements to rebuild the database is created and written to a file or standard output. This script file is equivalent to the plain-text output of gs_dump.

You can specify and sort the data to import.

Precautions

gs_restore incrementally imports data by default. To prevent data exception caused by consecutive imports, use the **-c** and **-e** parameters for each import. **-c** indicates that the database objects that already exist in the database to be restored are cleared (deleted) before the database objects are rebuilt. **-e** indicates that if an error occurred when an SQL statement is sent to the database, the system exits. By default, the system continues to import data and displays a series of error information after the import is complete.

Preparing for Data Migration

1. Prepare an ECS or a device that can access the GaussDB instance over EIP.
 - To connect to a GaussDB instance through an ECS, you must first create an ECS.
For details on how to create and log in to an ECS, see [Purchasing an ECS](#) and [Logging In to an ECS](#) in *Elastic Cloud Server Getting Started*.
 - To connect to a GaussDB instance through a device that can access the GaussDB instance over EIP, you must:
 - i. Bind an EIP to the GaussDB instance. For details, see [Binding an EIP](#).
 - ii. Ensure that the local device can access the EIP that has been bound to the GaussDB instance.
2. Install the gsql client on the prepared ECS or device that can access the GaussDB database, and connect it to the GaussDB instance. For details, see [Using gsql to Connect to a Database](#).

Procedure

Step 1 Upload the file exported by `gs_dump` to the device. For details about the file exported by `gs_dump`, see [Using gs_dump and gs_dumpall to Export Data](#).

Step 2 Use `gs_restore` to import the definitions of all database objects from the exported file to the destination database.

- Example 1: Use `gs_restore` to import the data and object definitions of the **gs_example** database from the **gs_example_dump.tar** file (in .tar format) by specifying the database IP address and an existing database (for example, **gs_example_restore**). In this example, only relative output information is included.

```
gs_restore -U root /tmp/data/gs_example_dump.tar -p 8000 -d gs_example_restore -e -h 192.**.139;  
Password:  
restore operation successful  
total time: 1430 ms
```

- Example 2: Use `gs_restore` to import the data and object definitions of the **gs_example** database from the **gs_example_dump.tar** file (in .tar format) by specifying the database IP address and an existing database (for example, **gs_example_restore**). In addition, the database objects that already exist in the database to be imported are cleared (deleted). In this example, only relative output information is included.

```
gs_restore -U root /tmp/data/gs_example_dump.tar -p 8000 -d gs_example_restore -e -c -h 192.**.139;  
Password:  
restore operation successful  
total time: 1621 ms
```

Table 5-17 Common parameters

Parameter	Description	Example
-U	Username for database connection.	-U jack
-W	User password for database connection. - This parameter is not required for database administrators if the trust policy is used for authentication. - If you connect to the database without specifying this parameter and you are not a database administrator, you will be prompted to enter the password.	-W *****
-d	Name of a database to which data will be imported.	-d backupdb
-p	TCP port or local Unix-domain socket file name extension on which the server is listening for connections.	-p 8000

Parameter	Description	Example
-e	If an error occurs when you send the SQL statement to the database, the system exits. Error messages are displayed after the import process is complete.	-e
-c	Before re-creating database objects, clear (delete) the database objects that exist in the database to be imported.	-c
-s	Only schema definitions are imported. Sequence values and data will not be imported.	-s

----End

Helpful Links

For more information, see:

- [gs_restore \(distributed\)](#)
- [gs_restore \(centralized\)](#)

5.9 Using gs_loader to Import Data

Scenarios

You can use `gs_loader` to import the files exported by using the **copy to** command. The `gs_loader` tool converts the syntax supported by control files into **\copy** syntax, then leverages the existing **\copy** function to import main data. At the same time, `gs_loader` logs the results of the **\copy** operations to a log file.

Precautions

- `gs_loader` does not support M-compatible databases.
- Before using `gs_loader`, ensure that the `gs_loader` version is consistent with the `gsql` version and database version.
- Currently, `gs_loader` is only available for centralized instances.
- When you use `gs_loader` to import data, if transcoding is not required, the size of a single row of data (including tuple metadata, same as mentioned below) is less than 1 GB minus 1 byte. If transcoding is required, the size of a single row of data is less than 256 MB minus 1 byte. Special handling has been applied to the following transcoding scenarios: the size for UTF-8 to GB18030/GB18030_2022 transcoding is less than 512 MB minus 1 byte, and the size for UTF-8 to GBK transcoding is less than 1 GB minus 1 byte.
- It is recommended that the size of a single file to be imported be less than or equal to 1 GB. `gs_loader` has no limit on the size of a single file to be

imported. However, importing a large file is time-consuming. Therefore, you are advised to split a large file, and start multiple `gs_loader` processes to write data to the table in append mode. (If there is a need to truncate data, it should be done by performing a separate `TRUNCATE` operation, rather than writing the `TRUNCATE` statement into the control file.) When the CPU resources are sufficient, this method can effectively improve the import speed.

Preparing for Data Migration

1. Prepare an ECS or a device that can access the GaussDB instance over EIP.
 - To connect to a GaussDB instance through an ECS, you must first create an ECS.
For details on how to create and log in to an ECS, see [Purchasing an ECS](#) and [Logging In to an ECS](#) in *Elastic Cloud Server Getting Started*.
 - To connect to a GaussDB instance through a device that can access the GaussDB instance over EIP, you must:
 - i. Bind an EIP to the GaussDB instance. For details, see [Binding an EIP](#).
 - ii. Ensure that the local device can access the EIP that has been bound to the GaussDB instance.
2. Install the `gsqsl` client on the prepared ECS or device that can access the GaussDB database, and connect it to the GaussDB instance. For details, see [Using gsql to Connect to a Database](#).

Procedure

Step 1 Create a control file and prepare a data file.

1. Create a control file, for example, `/tmp/data/loader.ctl`, and import data to the `loader_tbl` table. `WHEN` requires that the second character in each row be a comma (,).

```
LOAD DATA
truncate into table loader_tbl
WHEN (2:2) = ','
fields terminated by ','
trailing nullcols
(
  id integer external,
  name char(32),
  con ":id || '-' || :name",
  dt date
)
```

2. Create a GUC parameter file, for example, `/tmp/data/guc.txt`.
`set a_format_copy_version='s1';`
3. Create a data file, for example, `/tmp/data/data.csv`.

```
1,OK,,2007-07-8
2,OK,,2008-07-8
3,OK,,2009-07-8
4,OK,,2007-07-8
43,DISCARD,,2007-07-8

'''
32,DISCARD,,2007-07-8
a,ERROR int,,2007-07-8
8,ERROR date,,2007-37-8
'''
,
8,ERROR fields,,2007-37-8
'''
5,OK,,2021-07-30
```

Step 2 Create a user and grant privileges to the user.

```
CREATE USER load_user WITH PASSWORD '*****!';
GRANT ALL ON SCHEMA public TO load_user;
SELECT copy_summary_create() WHERE NOT EXISTS(SELECT * FROM pg_tables WHERE
schemaname='public' AND tablename='gs_copy_summary');
GRANT ALL PRIVILEGES ON public.gs_copy_summary To load_user;
SELECT copy_error_log_create() WHERE NOT EXISTS(SELECT * FROM pg_tables WHERE
schemaname='public' AND tablename='pgxc_copy_error_log');
GRANT ALL PRIVILEGES ON public.pgxc_copy_error_log To load_user;
\c - load_user
Password for user load_user:
```

Step 3 Create a target table to import data. For example, you can create the **loader_tbl** table in the **gs_example** database.

```
\c gs_example
Password for user load_user:
CREATE TABLE loader_tbl
(
  ID NUMBER,
  NAME VARCHAR2(20),
  CON VARCHAR2(20),
  DT DATE
);
```

Step 4 Import the data.

Exit the current login connection.

```
\q
```

Before importing data, ensure that **gs_loader** has the required privileges. Ensure that the current directory has write privileges (**gs_loader** generates some temporary files when importing data. The files are automatically deleted after the import is completed).

```
gs_loader control=/tmp/data/loader.ctl data=/tmp/data/data.csv db=gs_example bad=/tmp/data/loader.bad
guc_param=/tmp/data/guc.txt errors=5 port=8000 passwd=***** user=load_user -h 192.*.139;
```

The output is shown as follows:

```
gs_loader: version 0.1
```

```
5 Rows successfully loaded.
```

```
log file is:
/tmp/data/loader.log
```

In the **/tmp/data/data.csv** file, the first four rows and the last row are imported. Rows 5 and 7 were ignored because the second character was not a comma (,). Rows 6, 10, and 13 were skipped because all fields were empty. Rows 8, 9, and 12 were not imported due to incorrect field values. The file defined as **bad=/tmp/data/loader.bad** can be used to view the erroneous lines, and the execution result is in **/tmp/data/loader.log**, which records the imported log information.

----End

Table 5-18 gs_loader parameters

Parameter	Description	Parameter Type/Value Range
help	Help information.	-

Parameter	Description	Parameter Type/Value Range
user	Database connection user (equivalent to -U).	Character string
-U	Database connection user (equivalent to user).	Character string
passwd	User password (equivalent to -W).	Character string
-W	User password (equivalent to passwd).	Character string
db	(Required) Database name. This parameter is equivalent to -d .	Character string
-d	(Required) Database name. This parameter is equivalent to db .	Character string
host	Host name of the running server, the path of the Unix domain socket, or the domain name. You can specify multiple host addresses by using character strings separated by commas (,). This parameter is equivalent to -h . If multiple host addresses are specified, the primary node is connected by default.	See the gsql --host parameter.
-h	Host name of the running server, the path of the Unix domain socket, or the domain name. You can specify multiple host addresses by using character strings separated by commas (,). This parameter is equivalent to host . If multiple host addresses are specified, the primary node is connected by default.	See the gsql --host parameter.
port	Port number of the database server. One or more port numbers can be configured. When one port number is configured, all IP addresses use the same port for connection. When multiple port numbers are configured, the sequence is the same as the IP address sequence, and the number of port numbers must be the same as the number of IP addresses. If they are different, an error is reported. This parameter is equivalent to -p .	See the gsql --port parameter.

Parameter	Description	Parameter Type/Value Range
-p	Port number of the database server. One or more port numbers can be configured. When one port number is configured, all IP addresses use the same port for connection. When multiple port numbers are configured, the sequence is the same as the IP address sequence, and the number of port numbers must be the same as the number of IP addresses. If they are different, an error is reported. This parameter is equivalent to port .	See the gsql --port parameter.
create	Specifies whether to create the pgxc_copy_error_log and gs_copy_summary tables. In the current version, the two tables are created by default. Therefore, this parameter is meaningless. This parameter is reserved only for compatibility.	The value can be true or false . The default value is true .
data	(Required) Data file. You can specify multiple data files or use wildcards (*) and question marks (?) to represent multiple data files.	Character string
control	(Required) Name of a control file.	Character string
log	Name of a log file.	Character string
bad	Name of the file that records the error rows and details. You can also specify a directory. If you do not specify a directory, the file is generated based on the data file name.	Character string
errors	Maximum number of error rows in a data file.	Integer. The default value is 0 .
limit	Maximum number of rows that can be imported.	Integer. By default, the value is infinite.

 CAUTION

- All parameters are in lowercase and are compatible with the `gsql` login mode, including `-p` port number, `-h` host, `-d` database, `-U` username, and `-W` password.
- When the **rows** parameter is specified, the number of commit times cannot exceed 1,000. Otherwise, the performance will be affected. The number of commit times is approximately equal to the number of data rows in the data file divided by the value of **rows**. If the **rows** parameter is not specified, there is no default value for **rows**. In this case, the transaction is committed only once after all data is imported to the table.
- Frequent commit of a small amount of data affects the data import performance. You are advised to set the **rows** parameter properly to ensure that the amount of data committed each time is greater than 5 MB. For common servers with 16 vCPUs | 128 GB specifications, in the scenario where one primary node and two standby nodes are deployed and 13 GB of data is imported to a table with five columns, the rate of multiple commits is about 10 MB/s, which is basically same as that of a single commit (5 MB data is committed each time; network impacts are not considered).
- Currently, `gs_loader` is only compatible with NUL characters in data files. It does not support NUL characters in CTL files. If the .ctl file contains the nul character, unexpected problems may occur.

For details about other parameters and control file syntax, see [gs_loader](#) in *Tool Reference for Centralized Instances*.

6 Database Use

6.1 Overview of Using a Database

After creating a GaussDB instance, you can use DAS or SQL statements to perform basic database operations such as creating a database, creating a database user, creating tables, inserting data into tables, and deleting data from tables based on your requirements. For details, see [Table 6-1](#).

Basic operations should comply with relevant design specifications. For details, see [Development and Design Proposal](#).

You can also develop applications based on the JDBC, ODBC, libpq, Psycopg, ecpg, and Go drivers. For details, see [Application Development Guide](#).

Table 6-1 Database operations

Database Operation		Description
Creating a database	• Creating a Database Through DAS • Creating a Database Using SQL Statements	These sections describe how to create a database using DAS and the CREATE DATABASE command.
Creating a database user	• Creating a Database User Through DAS • Creating a Database Account Using SQL Statements	These sections describe how to use the CREATE USER command to create a database user. Data is not shared between users.

Database Operation		Description
Creating a table	Creating a Table Using SQL Statements	A table is created in a database and can be stored in different databases. Tables under different schemas in a database can have the same name. This section describes how to use the CREATE TABLE command to create a table.
Inserting data to a table	Inserting Data to Tables Using SQL Statements	A new table contains no data. You need to insert data to the table before using it. This section describes how to insert one or more rows of data using the INSERT command.
Updating data in a table	Updating Data in a Table Using SQL Statements	You can update one row, all rows, or specified rows of data. You can also update data in a single column without affecting the data in the other columns. This section describes how to use the UPDATE command to update data in a table.
Viewing data in a table	• Viewing Data in a Table Using SQL Statements • Opening a Table and Viewing Table Details Through DAS	These sections describe how to use SQL statements to view data in tables and how to open tables and view table details on the DAS console.
Deleting data from a table	Deleting Data from a Table Using SQL Statements	Outdated data in existing tables may need to be deleted. This section describes how to use SQL statements to delete tables or table data.
Creating and managing views	• Creating and Managing Views Using SQL Statements • Opening a View and Viewing Details of a View Through DAS	These sections describe how to create, query, and delete views using SQL statements, and how to open a view and view details of a view through DAS.
Creating and managing tablespaces	Creating and Managing Tablespaces Using SQL Statements	This section describes how to create tablespaces, create objects in tablespaces, query tablespaces, query tablespace usage, change tablespace names, and delete tablespaces.

Database Operation		Description
Querying system catalogs	Querying System Catalogs Using SQL Statements	This section describes how to query system catalogs.
Creating and managing schemas	Creating and Managing Schemas Using SQL Statements	This section describes how to create a schema, use a schema, view the search path of a schema, control schema privileges, and delete a schema.
Creating and managing partitioned tables	Creating and Managing Partitioned Tables Using SQL Statements	This section describes how to create a partitioned table, insert data into a table, modify the row migration attributes of a partitioned table, delete a partition, add a partition, rename a partition, modify the tablespace of a partition, query a partition, and delete a partitioned table and tablespace.
Creating and managing indexes	Creating and Managing Indexes Using SQL Statements	This section describes how to create an index, modify the tablespace of an index partition, rename an index partition, query an index, and delete an index.
Creating and managing sequences	Creating and Managing Sequences Using SQL Statements	This section describes how to use a sequence to make a column a unique identifier.
Creating and managing scheduled jobs	Creating and Managing Scheduled Jobs Using SQL Statements	This section describes how to create a job, view job details, start a job, stop a job, modify job attributes, and delete a job.

6.2 Creating a GaussDB Database

Scenarios

After creating a GaussDB instance, you can create more databases as required. Database creation should comply with relevant design specifications. For details, see [Development and Design Proposal](#).

You can create a database by using either of the following methods:

- [Creating a Database Through DAS](#)
- [Creating a Database Using SQL Statements](#)

Creating a Database Through DAS

Step 1 [Log in to the management console](#).

- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, locate the DB instance you want to log in to and click **Log In** in the **Operation** column.
- Alternatively, click the DB instance name on the **Instances** page. On the displayed **Basic Information** page, click **Log In** in the upper right corner of the page.
- Step 5** On the **Custom Login** page, select the node to be logged in to. Enter the correct database username and password, and click **Test Connection**. After the connection test is successful, click **Log In**.
- Step 6** Create a database.
- In the database list of the homepage, click **Create Database**. In the displayed dialog box, enter a database name and specify a character set, template, and other required parameters. Click **OK**.
 - Alternatively, on the top menu bar, choose **SQL Operations > SQL Query**. Run the following command to create a database:

```
create database database_name;
```


----End

Creating a Database Using SQL Statements

You can also use the [CREATE DATABASE](#) command to create a database.

6.3 Creating a GaussDB Database User

Scenarios

When you create a GaussDB instance, the **root** user is created at the same time by default. You can add other users as required.

You can create a database user by one of the following methods:

- [Creating a Database User Through DAS](#)
- [Creating a Database User Through SQL Statements](#)

Constraints

Only centralized instances of version V2.0-8.100.0 or later are supported.

Creating a Database User Through DAS

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.

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

Alternatively, click the DB instance name on the **Instances** page. On the displayed **Basic Information** page, click **Log In** in the upper right corner of the page.

Step 5 On the displayed login page, enter the username and password and click **Log In**.

Step 6 On the top menu bar, choose **SQL Operations > SQL Query**.

Step 7 Create a database user.

```
create user username;
```

```
----End
```

Creating a Database User Through SQL Statements

You can also use **CREATE USER** to create a database user.

6.4 Authentication Configurations

6.4.1 Adding a Client Authentication Configuration

Scenarios

When a host needs to connect to a database remotely, its information must be added to the database system's configuration file, along with client authentication rules. To simplify the process, GaussDB automatically includes the following default rules in the client authentication configuration file when an instance is created:

- Default configurations for a centralized instance (assuming that the subnet CIDR block of the instance is 192.168.0.0)

TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	0:0:0:0/0	sha256
host	all	all	192.168.0.0/16	sha256
host	replication	all	192.168.0.0/16	sha256
host	replication	all	0:0:0:0/0	sha256

- The first record allows all users from any IPv4 client to access all databases using the SHA-256 authentication method.
- The second record allows all users from IPv4 clients within the current instance's subnet to access all databases using SHA-256.
- The third record allows all users from IPv4 clients within the current instance's subnet to request a replication connection using SHA-256.
- The fourth record allows all users from any IPv4 client to request a replication connection using SHA-256.

- Default configurations for a distributed instance

This record allows all users from any IPv4 client to access all databases using the SHA-256 authentication method.

TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	0:0:0:0/0	sha256

In most cases, the default configurations are sufficient for typical remote connection requirements. However, if you require more granular control over client

access or the existing authentication settings do not meet your operational requirements, GaussDB allows you to add custom client authentication configurations on the management console. This section describes how to add client authentication configurations on the console.

Constraints

- Client authentication can only be configured for an instance when all its nodes are in the available state.
- In cross-region disaster recovery scenarios, the client authentication configurations added to the primary instance will not be automatically synchronized to the DR instance.
- Instances of V2.0-8.1.0 or later support IPv6 addresses. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- For centralized instances, configuration changes are synchronized across all nodes. For distributed instances, changes are synchronized across all CN components.
- When an external client uses the M-compatible protocol to remotely connect to an M-compatible database, only the SHA-256 authentication method is supported.
- Added authentication configurations are only applied for newly created sessions.
- A maximum of 10 authentication rules can be added in a single batch operation.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Management > Client Authentication**.

Step 6 On the **Authentication Configurations** tab, click **Batch Add**.

Figure 6-1 Adding a client authentication configuration

Authentication Configurations

Configuration History

Batch Add

Restore Defaults

Search by type

Type	Database	User	Address	Method	Operation
host	all	all	0.0.0.0/0	sha256	Modify Delete
host	replication	all	192.168.0.0/16	sha256	Modify Delete
host	replication	all	0.0.0.0/0	sha256	Modify Delete

Total Records: 3

Step 7 Configure authentication rules and click **OK**.

Figure 6-2 Configuring access authentication rules**Table 6-2** Parameter description

Parameter	Description
Type	Client connection type. • host: The DB instance accepts both standard TCP/IP socket connections and SSL-encrypted TCP/IP socket connections. • hostssl: The DB instance only accepts SSL-encrypted TCP/IP socket connections. • hostnossl: The DB instance only accepts standard TCP/IP socket connections.
Database	Databases that can be accessed. • all: The rule matches all databases. • A specific database name or a list of databases NOTE – The value replication matches requests for replication connections, which are not associated with any specific database. If you intend to access a database named replication, you must explicitly specify "replication" as the database name in the Database column.
User	Users who are allowed to access the databases. • all: The record matches all users. • A specific database username or a list of users
Address	Allowed IP address range. • Currently, only the IP address/mask length format is supported. • For instances with a DB engine version of V2.0-8.1.0 or later, IPv6 addresses are also supported in the Address column.

Parameter	Description
Method	Authentication method used for connection. • reject: A connection is rejected unconditionally. It is commonly used to block certain hosts. • md5: MD5 is used for authentication. This method is considered insecure and is not recommended. It is disabled by default. To enable it, modify the password_encryption_type parameter. For details about how to modify the parameter, see Modifying Instance Parameters. • sha256: The client is required to provide a password encrypted using the SHA-256 algorithm. During transmission, the password is combined with a salt (a server-provided random number) and processed using a one-way SHA-256 hash, enhancing security. • sm3: The client is required to provide a password encrypted using the SM3 algorithm. During transmission, the password is combined with a salt (a server-provided random number) and processed using a one-way SM3 hash, enhancing security. • cert: A client certificate is used for authentication. This method requires SSL configuration and a valid client SSL certificate. No password is required. The cert authentication method only supports rules with the hostssl client connection type.

Step 8 After the configurations are added, check the added configuration information in the list.

----End

6.4.2 Deleting a Client Authentication Configuration

Scenarios

You can delete a client authentication configuration record based on your application requirements. After a record is deleted, that configuration becomes invalid. Existing sessions that were authenticated under the configuration continue to work, but new sessions that would rely on the deleted configuration will fail to connect. Ensure that the configuration you are deleting aligns with your application requirements.

Constraints

- Client authentication configurations can only be deleted for an instance when all its nodes are in the available state.
- In a cross-region disaster recovery scenario, deleting a client authentication configuration on the primary instance is not automatically synchronized to the DR instance. If you also need to remove the same configuration on the DR instance, manually delete it on the DR instance.

- For centralized instances, configuration changes are synchronized across all nodes. For distributed instances, changes are synchronized across all CN components.

Procedure

Step 1 Log in to the management console.

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

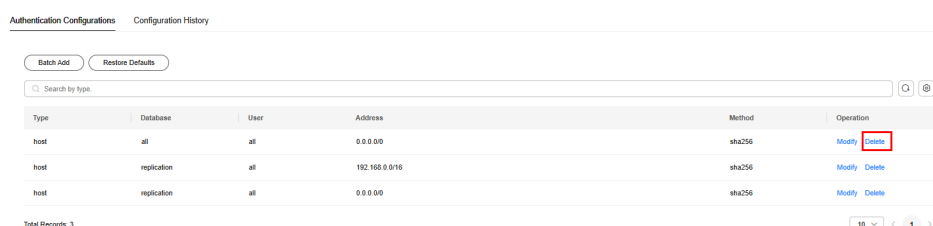
Step 3 Click  in the upper left corner of the page and choose **Databases** > **GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Management** > **Client Authentication**.

Step 6 On the **Authentication Configurations** tab, locate the target configuration and click **Delete** in the **Operation** column.

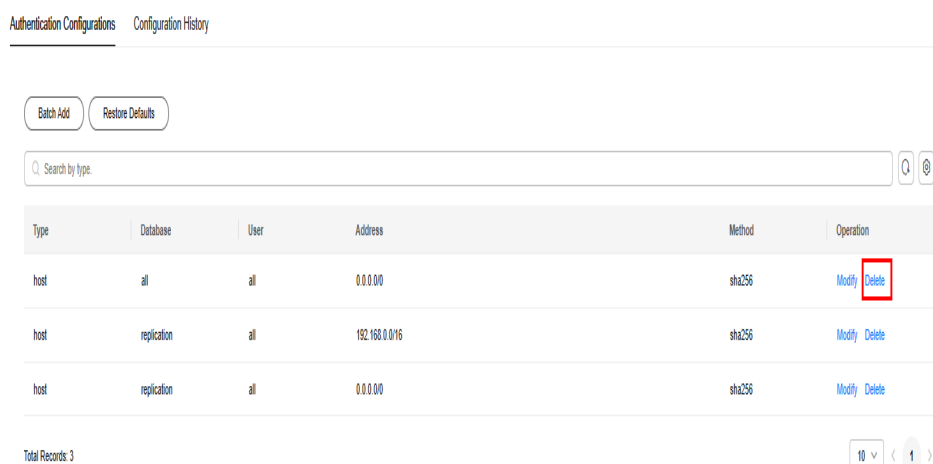
Figure 6-3 Deleting a client authentication configuration



Authentication Configurations Configuration History					
Batch Add Restore Defaults Search by type:					
Type	Database	User	Address	Method	Operation
host	all	all	0.0.0.0/0	sha256	Modify Delete
host	replication	all	192.168.0.0/16	sha256	Modify Delete
host	replication	all	0.0.0.0/0	sha256	Modify Delete
Total Records: 3					

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

Figure 6-4 Deleting a client authentication configuration



Authentication Configurations Configuration History					
Batch Add Restore Defaults Search by type:					
Type	Database	User	Address	Method	Operation
host	all	all	0.0.0.0/0	sha256	Modify Delete
host	replication	all	192.168.0.0/16	sha256	Modify Delete
host	replication	all	0.0.0.0/0	sha256	Modify Delete
Total Records: 3					

Step 8 Check whether the target configuration is deleted.

----End

6.4.3 Modifying a Client Authentication Configuration

Scenarios

You can modify a client authentication configuration record based on your application requirements. After the modification, existing sessions that have already been authenticated using the original configuration remain unaffected and continue to function. New sessions, however, will fail to connect if they do not meet the updated authentication requirements. Ensure that the modified configuration aligns with your application requirements.

Constraints

- Client authentication configurations can only be modified for an instance when all its nodes are in the available state.
- In a cross-region disaster recovery scenario, modifying a client authentication configuration on the primary instance is not automatically synchronized to the DR instance. If you also need to modify the same configuration on the DR instance, manually modify it on the DR instance.
- Instances of V2.0-8.1.0 or later support IPv6 addresses. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- For centralized instances, configuration changes are synchronized across all nodes. For distributed instances, changes are synchronized across all CN components.
- When an external client uses the M-compatible protocol to remotely connect to an M-compatible database, only the SHA-256 authentication method is supported.
- Modified authentication configurations are only applied for newly created sessions.

Procedure

Step 1 [Log in to the management console](#).

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

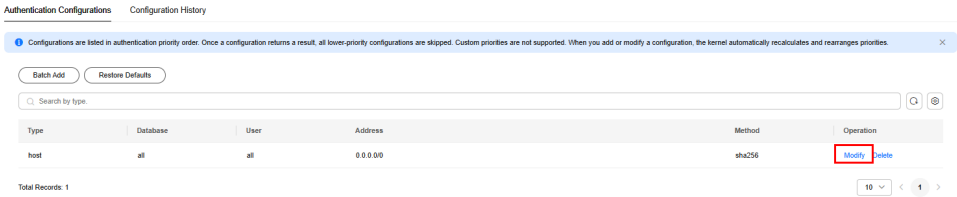
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Management > Client Authentication**.

Step 6 On the **Authentication Configurations** tab, locate the target configuration and click **Modify** in the **Operation** column.

Figure 6-5 Modifying a client authentication configuration



Step 7 After a record is modified, click **Save** in the **Operation** column.

Figure 6-6 Modifying a client authentication configuration

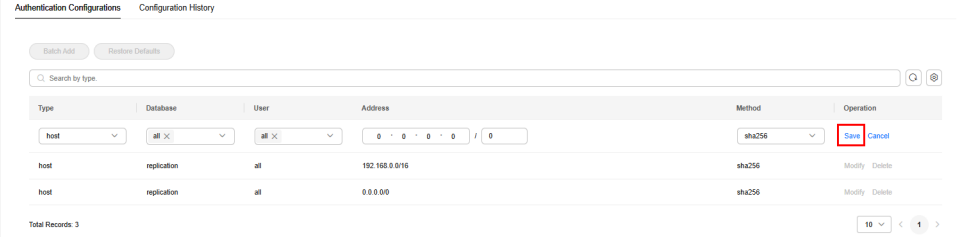


Table 6-3 Parameter description

Parameter	Description
Type	Client connection type. ● host: The DB instance accepts both standard TCP/IP socket connections and SSL-encrypted TCP/IP socket connections. ● hostssl: The DB instance only accepts SSL-encrypted TCP/IP socket connections. ● hostnossl: The DB instance only accepts standard TCP/IP socket connections.
Database	Databases that can be accessed. ● all: The rule matches all databases. ● A specific database name or a list of databases NOTE – The value replication matches requests for replication connections, which are not associated with any specific database. If you intend to access a database named replication, you must explicitly specify "replication" as the database name in the Database column.
User	Users who are allowed to access the databases. ● all: The record matches all users. ● A specific database username or a list of users

Parameter	Description
Address	Allowed IP address range. - Currently, only the IP address/mask length format is supported. - For instances with a DB engine version of V2.0-8.1.0 or later, IPv6 addresses are also supported in the Address column.
Method	Authentication method used for connection. reject: A connection is rejected unconditionally. It is commonly used to block certain hosts. md5: MD5 is used for authentication. This method is considered insecure and is not recommended. It is disabled by default. To enable it, modify the password_encryption_type parameter. For details about how to modify the parameter, see Modifying Instance Parameters. sha256: The client is required to provide a password encrypted using the SHA-256 algorithm. During transmission, the password is combined with a salt (a server-provided random number) and processed using a one-way SHA-256 hash, enhancing security. sm3: The client is required to provide a password encrypted using the SM3 algorithm. During transmission, the password is combined with a salt (a server-provided random number) and processed using a one-way SM3 hash, enhancing security. cert: A client certificate is used for authentication. This method requires SSL configuration and a valid client SSL certificate. No password is required. The cert authentication method only supports rules with the hostssl client connection type.

Step 8 After the modification is saved, check the updated configuration information in the list.

----End

6.4.4 Viewing the Modification History of Client Authentication Configurations

Scenarios

You can view the modification history of client authentication configurations.

Constraints

Only the modification history from the past month is available for review.

Procedure

Step 1 [Log in to the management console](#).

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

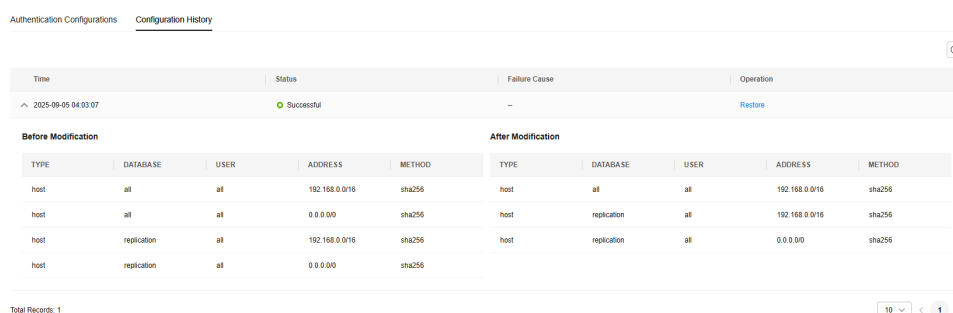
Step 3 Click  in the upper left corner of the page and choose **Databases** > **GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Management** > **Client Authentication**.

Step 6 Click the **Configuration History** tab. On the displayed page, locate the record you want to check and click  to view the modification details.

Figure 6-7 Configuration History



Time	Status	Failure Cause	Operation
2025-09-05 04:03:07	Successful	-	Restore

Before Modification					After Modification				
TYPE	DATABASE	USER	ADDRESS	METHOD	TYPE	DATABASE	USER	ADDRESS	METHOD
host	all	all	192.168.0.0/16	sha256	host	all	all	192.168.0.0/16	sha256
host	all	all	0.0.0.0/0	sha256	host	replication	all	192.168.0.0/16	sha256
host	replication	all	192.168.0.0/16	sha256	host	replication	all	0.0.0.0/0	sha256
host	replication	all	0.0.0.0/0	sha256					

Total Records: 1

Table 6-4 Parameter description

Parameter	Description
Time	Time when client authentication configurations are modified.
Status	Status of the modification. - Successful - Failed
Failure Cause	The reason for failure when the modification status is Failed .
TYPE	Client connection type. host: The DB instance accepts both standard TCP/IP socket connections and SSL-encrypted TCP/IP socket connections. hostssl: The DB instance only accepts SSL-encrypted TCP/IP socket connections. hostnossl: The DB instance only accepts standard TCP/IP socket connections.

Parameter	Description
DATABASE	Databases that can be accessed. • all: The rule matches all databases. • A specific database name or a list of databases NOTE – The value replication matches requests for replication connections, which are not associated with any specific database. If you intend to access a database named replication, you must explicitly specify "replication" as the database name in the Database column.
USER	Users who are allowed to access the databases. • all: The record matches all users. • A specific database username or a list of users
ADDRESS	Allowed IP address range. • Currently, only the IP address/mask length format is supported. • For instances with a DB engine version of V2.0-8.1.0 or later, IPv6 addresses are also supported in the Address column.
METHOD	Authentication method used for connection. • reject: A connection is rejected unconditionally. It is commonly used to block certain hosts. • md5: MD5 is used for authentication. This method is considered insecure and is not recommended. It is disabled by default. To enable it, modify the password_encryption_type parameter. For details about how to modify the parameter, see Modifying Instance Parameters. • sha256: The client is required to provide a password encrypted using the SHA-256 algorithm. During transmission, the password is combined with a salt (a server-provided random number) and processed using a one-way SHA-256 hash, enhancing security. • sm3: The client is required to provide a password encrypted using the SM3 algorithm. During transmission, the password is combined with a salt (a server-provided random number) and processed using a one-way SM3 hash, enhancing security. • cert: A client certificate is used for authentication. This method requires SSL configuration and a valid client SSL certificate. No password is required. The cert authentication method only supports rules with the hostssl client connection type.

----End

6.4.5 Restoring Client Authentication Configurations

Scenarios

If a client authentication configuration is modified or deleted by mistake, you can restore the configuration to its state prior to the change based on the recorded modification history. You can also restore the client authentication configurations to the original defaults, in which case all your custom configurations will be reset. Ensure that the restoration aligns with your application requirements.

Constraints

- Only successfully modified records in the client authentication configuration history are available for restoration.
- Client authentication configurations can only be restored for an instance when all its nodes are in the available state.
- In a cross-region disaster recovery scenario, restoring a client authentication configuration on the primary instance is not automatically synchronized to the DR instance. If you also need to restore the same configuration on the DR instance, manually restore it on the DR instance.
- For centralized instances, configuration changes are synchronized across all nodes. For distributed instances, changes are synchronized across all CN components.
- Restored authentication configurations are only applied for newly created sessions.

Restoring a Client Authentication Configuration from the Modification History

Step 1 [Log in to the management console.](#)

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

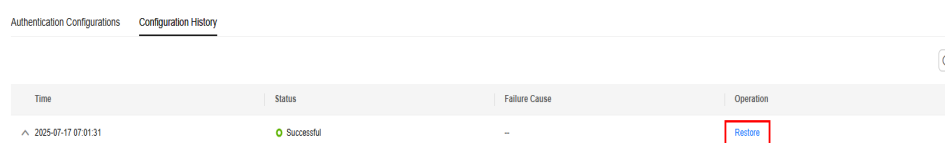
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Management > Client Authentication**.

Step 6 Click the **Configuration History** tab. On the displayed page, locate the modification record you want to restore, and click **Restore** in the **Operation** column.

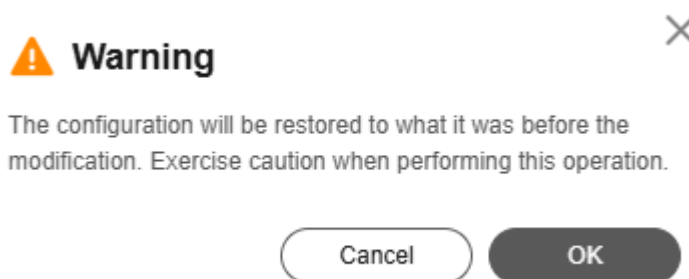
Figure 6-8 Restoring client authentication configurations



Authentication Configurations		Configuration History	
Time	Status	Failure Cause	Operation
2025-07-17 07:01:31	Successful	-	Restore

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

Figure 6-9 Confirming the restoration



Step 8 After the restoration is complete, click the **Authentication Configurations** tab and check that the configurations have been restored to their state before the modification.

----End

Restoring Default Client Authentication Configurations

Step 1 [Log in to the management console](#).

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

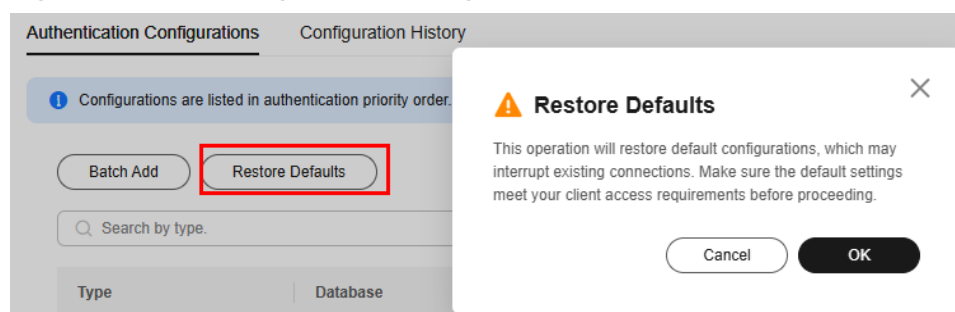
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Database Management > Client Authentication**.

Step 6 Click **Restore Defaults**. In the displayed dialog box, click **OK**.

Figure 6-10 Restoring default configurations



Step 7 After the restoration is complete, check that the default configurations are displayed in the list.

----End

7 Instance Management

7.1 Viewing GaussDB Instance Overview Data

Scenarios

GaussDB offers an instance overview page for you to review details of all database instances under your account, including status-based instance statistics and alarm data.

Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane, choose **Overview**.

If there are no DB instances, no DB instance information will be displayed on the **Overview** page. Instead, you can only create a DB instance on this page. For new users, follow the prompts to create your first instance. If there are existing instances, you can view instance information on this page.

- Instances by Status
You can also view instance status statistics on the **Instances** page.

Figure 7-1 Instances by Status



Table 7-1 Parameters for instance status statistics

Item	Description	Operation	Handling Suggestion
Total instances	Number of instances installed in the current GaussDB system	Click Total instances to go to the instance list and view all instances.	-
Abnormal	Number of instances that are in an abnormal state on the Instances page If the storage space of an instance is full, the instance is also considered abnormal.	Click Abnormal to go to the instance list and view abnormal instances.	To handle this issue, submit a service ticket .
Creation failed	Number of instances that are in the creation failed state on the Instances page	Click Creation failed to go to the instance list and view instances that fail to be created.	To handle this issue, submit a service ticket .
Frozen	Number of instances that are in the frozen state on the Instances page	Click Frozen to go to the instance list and view frozen instances.	For details, see Resource Freezing, Unfreezing, Release, Deletion, and Unsubscription .
Creating	Number of instances that are in the creating state on the Instances page	Click Creating to go to the instance list and view instances that are being created.	-
Running	Number of instances that are in the available state on the Instances page	Click Running to go to the instance list and view running instances.	-
Stopped	Number of instances that are in the stopped state on the Instances page	Click Stopped to go to the instance list and view stopped instances.	-

- Alarm Statistics

You can view alarm statistics in the last 1 hour, last 3 hours, last 12 hours, last day, last three days, or last week. Instance alarm details are displayed from multiple dimensions. For details, see [Table 7-2](#) and [Table 7-3](#).

Table 7-2 Description of the Top 5 Instances by Total Number of Alarms area

Item	Description
Alarm Severity	Number of all uncleared alarms of different severities in the specified period The alarm severity can be critical, major, minor, or warning.
Top 5 Instances by Total Number of Alarms	Number of alarms of the top 5 instances with the largest number of uncleared alarms in a specified period

Table 7-3 Alarm list description

Item	Description
Alarm Name/ID	Name and ID of an alarm
Instance Name/ID	Name and ID of the instance for which an alarm is generated
Alarm Severity	Severity of an alarm CAUTION Critical alarms are generated for faults that affect system-provided services. You need to take countermeasures immediately. For example, if a device or resource is unavailable, fix it immediately.
Alarm Type	Type of an alarm, which can be Metric or Event
First Reported	Time when an alarm was reported for the first time
Last Reported	Lastest time when an alarm was reported

- DR Overview
You can view information about the primary and DR instances in a DR task.

Figure 7-2 DR Overview

DR Overview

Task Name/ID	Status	DR Role	DR Type	Primary Instance	Primary Region	DR Instance	DR Region	Created
DR task-0025 dbC5da-9f1a-4e74-9a0b-c0eaf1d0fa	Available	DR instance	Streaming	tc947d24054c27b477e05a61aa3a1e14		gauss-0025 127c0b6c40aa4790a4e4db00ee051e14	or	Oct 26, 2024 15:59:32 GMT+08:00
DR task-0025 0601c0e4ee74111a0e0b-af0c30e0635f	Available	Primary in...	Streaming	gauss-0025 tc947d24054c27b477e05a61aa3a1e14		127c0b6c40aa4790a4e4db00ee051e14	or	Oct 26, 2024 15:57:23 GMT+08:00

----End

7.2 Configuring Security Group Rules for a GaussDB Instance

Scenarios

A security group is a collection of access control rules for ECSs and GaussDB instances that are within the same VPC, have the same security requirements, and are mutually trusted.

If you have applied for the whitelist of not specifying a security group when creating an instance, skip this section. The security group information will not be displayed in the DB instance information area.

To ensure database security and reliability, you need to configure security group rules to allow specific IP addresses and ports to access the GaussDB instances.

- When you attempt to connect to a GaussDB instance through a private network, check whether the ECS and GaussDB instance are in the same security group.
 - If they are in the same security group, they can communicate with each other by default. No security group rule needs to be configured.
 - If they are in different security groups, you need to configure security group rules for the ECS and GaussDB instance, respectively.
 - GaussDB instance: Configure an **inbound rule** for the security group with which the GaussDB instance is associated.
 - ECS: The default security group rule allows all outgoing data packets. In this case, you do not need to configure a security group rule for the ECS. If **not all outbound traffic is allowed** in the security group, you need to configure an **outbound** rule for the ECS to allow all outbound packets.
- When you attempt to connect to a GaussDB instance using an EIP, you need to configure an **inbound rule** for the security group associated with the instance.

This section describes how to configure an inbound rule for a GaussDB instance.

For details about the requirements of security group rules, see [Adding a Security Group Rule](#) in the *Virtual Private Cloud User Guide*.

Precautions

The default security group rule allows all outbound data packets. This means that ECSs and GaussDB instances associated with the same security group can access each other by default. After a security group is created, you can add security group rules to control the access from and to the GaussDB instance.

- By default, you can create up to 500 security group rules.

- Ensure that each security group has no more than 50 rules.
- To access a GaussDB instance from resources outside the security group, configure an **inbound rule** for the security group associated with the instance.
- All Kunpeng ECS flavors do not support inconsecutive ports.
If you use inconsecutive port numbers in a security group rule of a Kunpeng ECS, this rule and rules configured after this one do not take effect.
For example, if you configure security group rule A with inconsecutive ports **22, 24** and then configure security group rule B with port 9096, both rule A and rule B do not take effect.
- Outbound rules typically do not apply to DB instances. The rules are used only when a DB instance acts as a client.
- If a DB instance resides in a VPC but is not publicly accessible, you can also use a VPN connection to connect to it.
- When creating a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6500, and $\langle \text{database\_port} \rangle - (\langle \text{database\_port} \rangle + 100)$. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound rules allow all outbound traffic.
- When creating a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and $\langle \text{database\_port} \rangle - (\langle \text{database\_port} \rangle + 100)$. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.)
- To ensure data and instance security, grant only the required permissions to users. Apply the principle of least privilege for database access. Limit access by setting the allowed IP address to either the remote server's specific address or its smallest subnet range.
- The default value of **Source** is **0.0.0.0/0**, indicating that all IP addresses can access the GaussDB instance as long as they are associated with the same security group as the instance.

For details about the requirements of security group rules, see [Adding a Security Group Rule](#) in the *Virtual Private Cloud User Guide*.

Procedure

Step 1 [Log in to the management console](#).

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

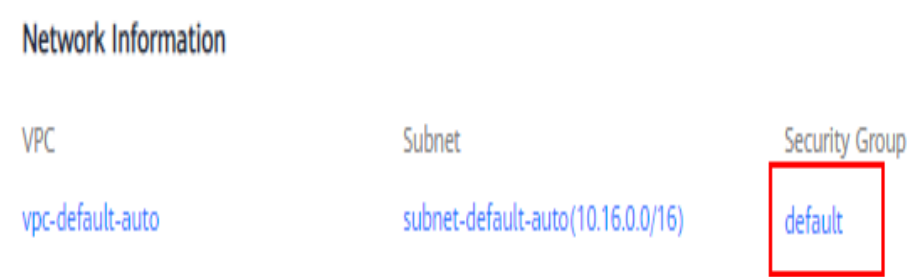
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Configure security group rules.

In the **Network Information** area, click the security group name.

Figure 7-3 Network information



Step 6 On the **Inbound Rules** tab, click **Add Rule**. In the displayed dialog box, configure the required parameters and click **OK**.

You can click + to add more inbound rules.

Table 7-4 Inbound rule parameter description

Parameter	Description	Example Value
Protocol & Port	Network protocol. Currently, the value can be All , TCP , UDP , ICMP , GRE , or others.	TCP (Custom ports)
	Port: port or port range over which the traffic can reach your ECS. The value ranges from 1 to 65535.	When connecting to your instance through a private network, enter the port of the ECS used to connect to your instance.
Type	IP address type. - IPv4 - IPv6	IPv4
Source	Source of the security group rule. The value can be a security group or an IP address. Examples: - xxx.xxx.xxx.xxx/32 (IPv4 address) - xxx.xxx.xxx.0/24 (subnet) 0.0.0.0/0 (any IP address)	0.0.0.0/0
Description	Provides supplementary information about the security group rule. This parameter is optional. The description can contain up to 255 characters and cannot contain angle brackets (<) or (>).	-

----End

7.3 Binding and Unbinding an EIP for a GaussDB Instance

Scenarios

You can bind an EIP to a GaussDB instance for public access and can unbind the EIP from an instance as required.

Precautions

- To ensure that the DB instance can be accessed, the security group used by the DB instance must allow access to the database port. For example, if the database port is **1611**, ensure that the security group allows access to the port **1611**.
- If a DB instance has already been bound with an EIP, you must unbind the EIP from the instance first before binding a new EIP to it.
- An EIP can be bound to only one node IP address of a DB instance.
- Security risks may arise after an EIP is bound to an instance. To enhance database security, set strict inbound and outbound rules for the security group. For details about how to configure a security group, see [Configuring Security Group Rules](#).

Binding an EIP

Step 1 [Log in to the management console](#).

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

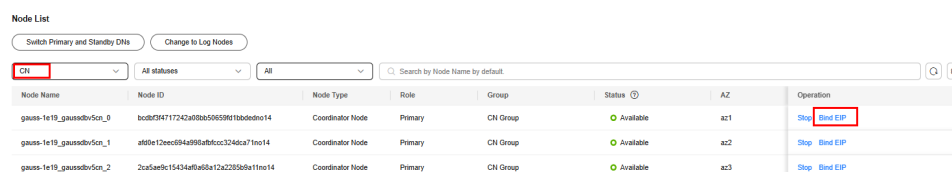
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Node List** area, click **Bind EIP** in the **Operation** column.

For a distributed instance, switch to **CN** in the filter area first and then bind an EIP.

Figure 7-4 Binding an EIP (distributed instance)



The screenshot shows the 'Node List' interface with a table of nodes. The 'CN' filter is selected, and the 'Bind EIP' button is highlighted in the 'Operation' column for the first node.

Node Name	Node ID	Node Type	Role	Group	Status	AZ	Operation
gauss-1e19_gaussdbv5cn_0	bcd8f94717242a0ba5595981b0dedno14	Coordinator Node	Primary	CN Group	Available	az1	Stop Bind EIP
gauss-1e19_gaussdbv5cn_1	a85fe12ec694a990abfccc3246ca71no14	Coordinator Node	Primary	CN Group	Available	az2	Stop Bind EIP
gauss-1e19_gaussdbv5cn_2	2ca5a6dc15434af0a85a12x22958a11no14	Coordinator Node	Primary	CN Group	Available	az3	Stop Bind EIP

Figure 7-5 Binding an EIP (centralized instance)

Node List

Switch Primary and Standby DBs

Node Name	Node ID	Role	Status	AZ	Private IP Address	IPv6 Address	Operation
gauss-bb51_node_0	0b4d41bc2534fbb2130e52894b040e14	Primary	Available	az1	192.168.3.125	--	Reboot Stop Bind EIP
gauss-bb51_node_1	1cb7050382148c3ae5e81494208a5ae014	Standby	Available	az2	192.168.3.63	--	Reboot Stop Bind EIP
gauss-bb51_node_2	53b4a0b7e094a096e713d05c2526e40e14	Standby	Available	az3	192.168.3.104	--	Reboot Stop Bind EIP

Step 6 In the displayed dialog box, all available unbound EIPs are listed. Select the required EIP and click **OK**.

If no available EIPs are displayed, click **View EIP** and obtain an EIP.

Figure 7-6 Binding an EIP

Bind EIP

For security purposes, after the EIP is bound, use [SSL](#) to connect to the database and add inbound and outbound rules in the security group.

Select EIP

	EIP	Status	Bandwidth
☒	10.83.83.51	☒ Unbound	5 Mbit/s

View EIP

Canceled

OK

Step 7 In the **Node List** area, check the operation result in the **EIP** column.

Figure 7-7 Checking the binding result

AZ	IP Address	IPv6 Address	EIP	Operation
az1	10.16.224.241	--	10.83.83.51 View	Reboot Stop Node More

To unbind the EIP from the instance, see [Unbinding an EIP](#).

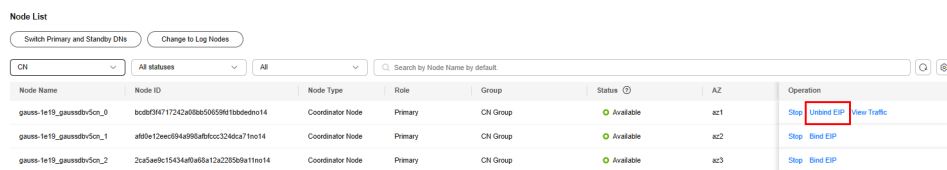
-----End

Unbinding an EIP

Step 1 [Log in to the management console](#).

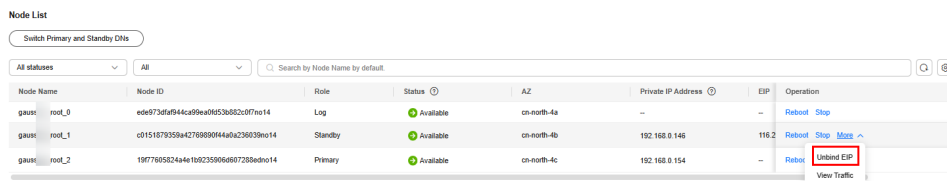
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the instance that has been bound with an EIP.
- Step 5** In the **Node List** area, click **Unbind EIP** in the **Operation** column.
- For a distributed instance, switch to **CN** in the filter area first and then click **Unbind EIP** in the **Operation** column.
 - For a centralized instance, click **More** and choose **Unbind EIP** in the **Operation** column.

Figure 7-8 Unbinding an EIP (distributed instance)



Node Name	Node ID	Node Type	Role	Group	Status	AZ	Operation
gauss-1e19_gaussdbv5cn_0	b0bf3f4717242a0b65065961b0bde014	Coordinator Node	Primary	CN Group	Available	az1	Stop Unbind EIP View Traffic
gauss-1e19_gaussdbv5cn_1	a0f0e12ee094a99a0a0cc324ca71no14	Coordinator Node	Primary	CN Group	Available	az2	Stop Bind EIP
gauss-1e19_gaussdbv5cn_2	2ca5a6c15434fa0a58a12a22058a11no14	Coordinator Node	Primary	CN Group	Available	az3	Stop Bind EIP

Figure 7-9 Unbinding an EIP (centralized instance)



Node Name	Node ID	Role	Status	AZ	Private IP Address	EIP	Operation
gauss-root_0	ede073afa944c90e06536082c07no14	Log	Available	cn-north-4a	--	--	Reboot Stop
gauss-root_1	c015167959a4276889044a0a236039no14	Standby	Available	cn-north-4b	192.168.0.146	116.2	Reboot Stop More
gauss-root_2	19077605824a4e1b0235906060728b0no14	Primary	Available	cn-north-4c	192.168.0.154	--	Reboot Unbind EIP View Traffic

- Step 6** In the displayed dialog box, click **OK** to unbind the EIP.
- Step 7** If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see [Identity and Access Management User Guide](#).

- Step 8** In the **Node List** area, check the operation result in the **EIP** column.

To bind an EIP to the instance again, see [Binding an EIP](#).

----End

7.4 Modifying the Recycle Bin Policy for a GaussDB Instance

GaussDB automatically moves deleted instances to the recycle bin. This section describes how to set the retention period of instances in the recycle bin.

Precautions

- The recycle bin is enabled by default and cannot be disabled. The deleted instances can be retained for 7 days by default.
- The new recycling policy only applies to instances that were put in the recycle bin after the new policy was put into effect. Those already in the recycle bin follow the original policy.

Billing

Instances in the recycle bin will not incur charges.

Procedure

Step 1 [Log in to the management console](#).

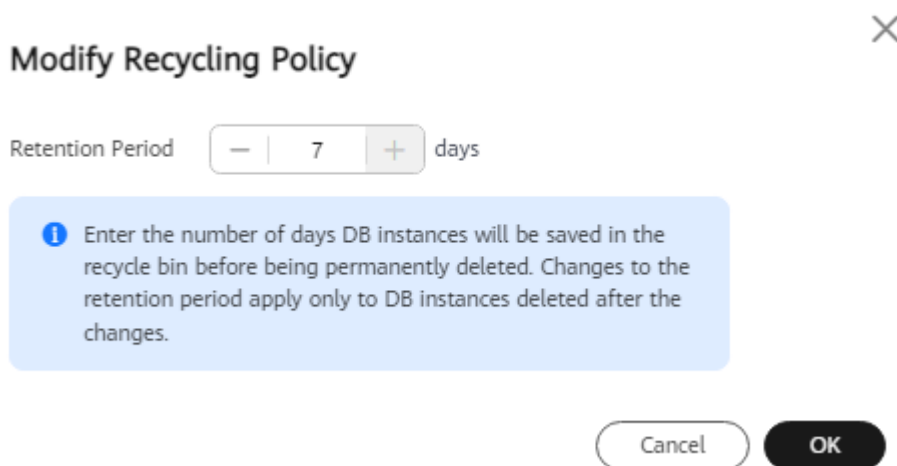
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 Click **Modify Recycling Policy**. In the displayed dialog box, set the retention period for the deleted DB instances from 1 day to 7 days.

Figure 7-10 Modifying the recycling policy



Step 6 Click **OK**.

----End

Related Operations

[Rebuilding a GaussDB Instance](#)

7.5 Exporting Information About All GaussDB Instances

Scenarios

You can export information about all instances in the list for review and analysis.

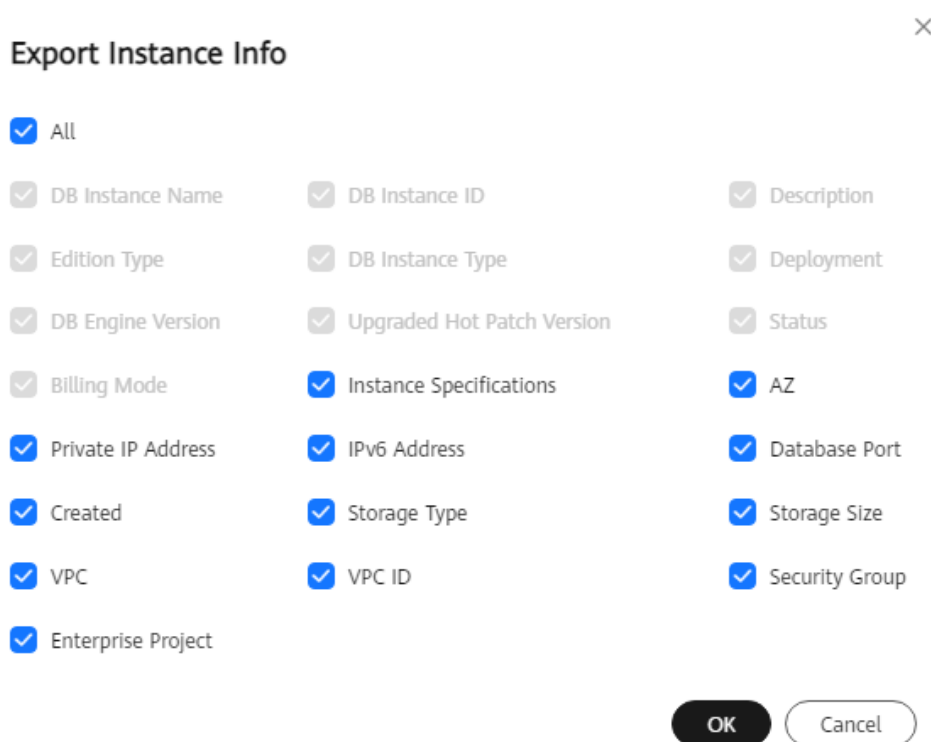
Constraints

A tenant can export a maximum of 3,000 instances at a time. The time required for the export depends on the number of instances.

Exporting All Instance Information

- Step 1** [Log in to the management console.](#)
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click **Export Instance Info**. By default, all instance information is exported.
- Step 5** In the displayed dialog box, select the items to be exported and click **OK**.

Figure 7-11 Exporting information about all instances



Step 6 After the export task is complete, a CSV file is generated on the local PC.

----End

7.6 Unsubscribing a Yearly/Monthly GaussDB Instance

Scenarios

To delete a DB instance billed on a yearly/monthly basis, you need to unsubscribe the order. Currently, DB instances cannot be unsubscribed in batches. You can unsubscribe only one instance at a time. For details, see [Unsubscribing a Single Instance](#). For unsubscription fees, see [Unsubscription Rules](#).

For pay-per-use DB instances, you need to delete them on the **Instances** page. For details, see [Deleting a Pay-per-Use GaussDB Instance](#).

Precautions

- After an unsubscription request is submitted, resources and data will be deleted and cannot be retrieved.
- If you want to retain data, complete a manual backup before submitting the unsubscription request.
- A maximum of 100 resources can be unsubscribed at a time.
- Unsubscribed instances will be moved to the recycle bin, but will be permanently deleted after a length of time determined by the recycling policy. The system retains the most recent automated full backup of a deleted instance from one day ago (or, if none exists, the latest automated full backup) and then creates a new full backup. You can restore the instance later by using any of the backups. All manual backups are retained and will incur additional costs. To delete the manual backups, go to the **Backups** page on the console.

Unsubscribing a Single Instance (Method 1)

Unsubscribe a yearly/monthly DB instance on the **Instances** page.

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, locate the instance, click **More** in the **Operation** column, and choose **Unsubscribe**.

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

For unsubscription details, see [Unsubscription Rules](#).

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

Step 7 View the unsubscription result. After the DB instance order is successfully unsubscribed, the DB instance is no longer displayed in the instance list on the **Instances** page.

----End

Unsubscribing a Single Instance (Method 2)

Unsubscribe a yearly/monthly instance on the **Billing Center** page.

Step 1 [Log in to the management console](#).

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

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

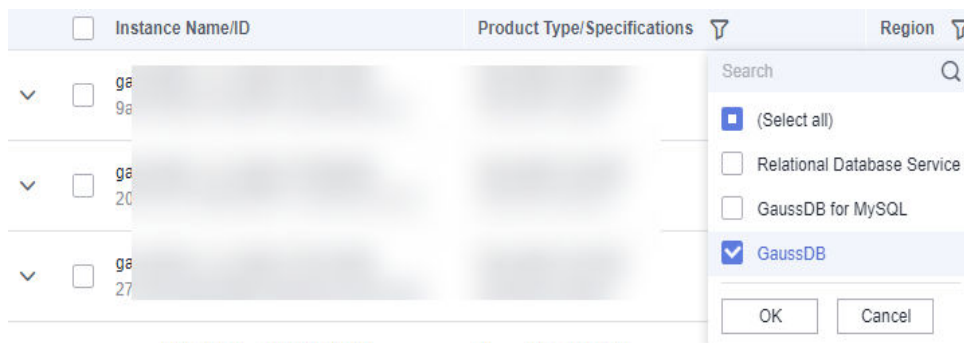
Step 4 In the upper right corner, click **Billing**.

Step 5 In the navigation pane, choose **Orders > Unsubscriptions**.

Step 6 On the displayed page, select the order to be unsubscribed and click **Unsubscribe from Resource** in the **Operation** column.

- You can select **GaussDB** in the **Product Type/Specifications** filter box to filter all GaussDB orders.

Figure 7-12 Filtering all GaussDB orders



- Alternatively, you can search for orders by name, order number, or ID in the search box above the order list.

Step 7 On the displayed page, confirm the order to be unsubscribed and select a reason. Then, click **Confirm**.

For unsubscription details, see [Unsubscription Rules](#).

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

Step 9 View the unsubscription result. After the instance order is successfully unsubscribed, the instance will be deleted.

----End

7.7 Stopping a GaussDB Instance

Scenarios

You can manually stop a GaussDB instance. This operation terminates the database processes across all nodes within the instance and shuts down the VMs hosting the nodes. However, resources like EIPs, storage, and backups are retained. If you are using DB instances mainly for routine development, you can stop the pay-per-use instances during idle periods to cut down on costs.

Constraints

- The following operations cannot be performed when an instance is being stopped and after it is stopped: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting the instance, and deleting the instance.
- After an instance is stopped, its component information cannot be queried.
- After an instance is stopped, it cannot provide services. Exercise caution when performing this operation.
- Instance parameters cannot be modified after an instance is stopped.
- Stopping an instance will stop all component processes in it and workloads will be interrupted. Exercise caution when performing this operation.

Billing

For pay-per-use instances, billing for the VMs hosting it stops when the instance is stopped, but billing continues for associated resources like EIPs, storage, and backups. For yearly/monthly instances, all related resources are billed even after the instance is stopped.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Stop**.

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

Figure 7-13 Stopping an instance

Stop DB Instance

Stop this DB instance?

Name/ID	DB Instance Type	Status
gauss-af3d 3b61b47619a94b8b9304e...	Centralized	Available

Stopping an instance will stop all component processes in it, causing interruptions to your workloads and preventing backups from being created. Exercise caution when performing this operation. For pay-per-use instances, billing for the VMs hosting it stops when the instance is stopped, but billing continues for associated resources like EIPs, storage, and backups. For yearly/monthly instances, all related resources are billed even after the instance is stopped. If these resources are no longer required, delete them to [stop further billing](#).

To confirm stopping the instance, enter "YES" below. [Auto Enter](#)

YES

Cancel

OK

Step 6 Refresh the instance list and view the status of the instance. If its status is **Stopped**, it has been stopped successfully.

----End

7.8 Starting a GaussDB Instance

Scenarios

GaussDB allows you to manually start a stopped DB instance.

Precautions

- Only instances in the **Stopped** state can be started.
- During instance startup, the following operations cannot be performed: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting the instance, and deleting the instance.
- If the number of stopped nodes in a DN shard exceeds half of the replicas of the shard, the instance may be abnormal. You are advised to stop all nodes and then start the instance.

Procedure

Step 1 Log in to the management console.

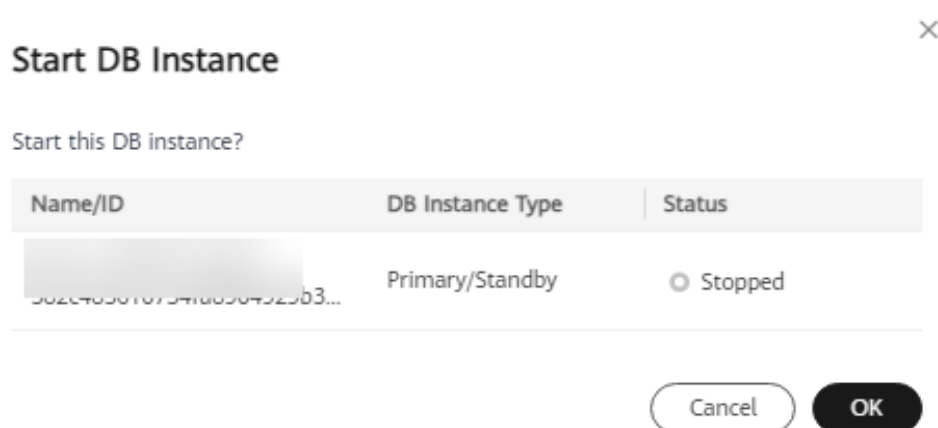
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Start**.

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

Figure 7-14 Starting an instance



Step 6 Refresh the instance list and view the status of the instance. If its status is **Available**, it has been started successfully.

----End

7.9 Rebooting a GaussDB Instance

Scenarios

You can reboot a DB instance for the modifications to take effect.

Precautions

- You can reboot a DB instance only when its status is **Available**. Your database may be unavailable in some cases, for example, when some modifications are being made.
- Rebooting a DB instance will cause service interruptions. During this period, the DB instance status is **Rebooting**.
- An instance is not available when it is being rebooted. After the reboot completes, the cached memory will be automatically cleared. You are advised to reboot the instance during off-peak hours.

- To quickly reboot a DB instance, perform fewer operations on the DB instance.
- If there are a large number of slow SQL statements or sessions, or if the thread pool is full, the reboot process may take a longer time than usual.

Procedure

Step 1 Log in to the management console.

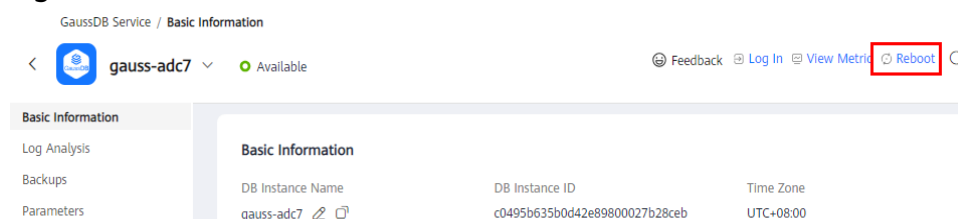
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, locate the instance you want to reboot and choose **More > Reboot** in the **Operation** column.

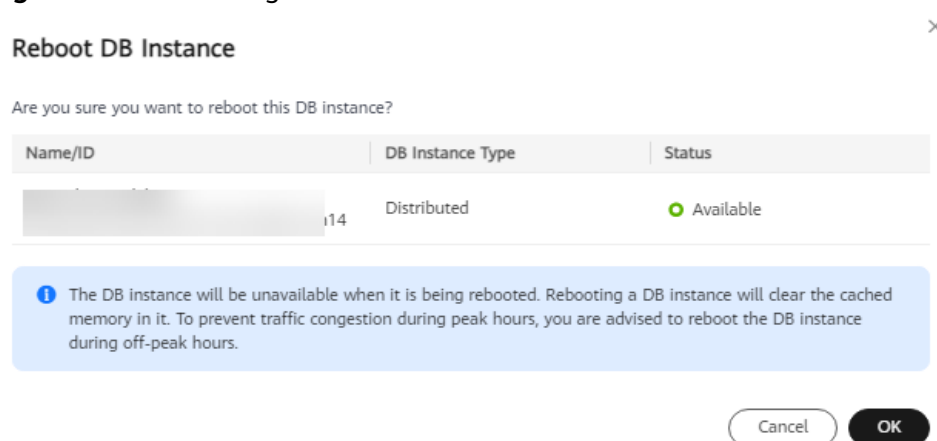
Alternatively, click the instance name to go to the **Basic Information** page. Click **Reboot** in the upper right corner of the page.

Figure 7-15 Basic instance information



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

Figure 7-16 Rebooting an instance



The instance status becomes **Rebooting**.

Step 6 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

Two-factor authentication improves the security of your account. For details about how to view and enable high-risk operation protection, see [Identity and Access Management User Guide](#).

Step 7 Refresh the DB instance list and view the status of the DB instance. If its status is **Available**, it has been rebooted.

----End

7.10 Deleting a Pay-per-Use GaussDB Instance

Scenarios

On the **Instances** page, you can manually delete DB instances that fail to be created or unnecessary instances after service changes.

Billing

If a pay-per-use instance is deleted, its automated backups will also be deleted and you will no longer be billed for them. Manual backups, however, will be retained and continue to incur costs.

Precautions

- Instance deletion cannot be undone. Exercise caution when performing this operation. Back up your data first if you want to keep it after deleting an instance.
- Instances cannot be deleted while an operation is being performed on them. They can only be deleted once the operations are complete.
- You can restore a DB instance that was deleted up to 7 days ago from the recycle bin. For details, see [Modifying the Recycle Bin Policy for a GaussDB Instance](#).
- If you delete a stopped instance, it cannot be rebuilt from the recycle bin. Exercise caution when performing this operation. If you might want to rebuild it later, start the instance first and make sure it is running properly before deleting it.
- Instances in the recycle bin will not incur charges.
- When **pay-per-use** instances are deleted, manual backups are retained. (To delete a yearly/monthly instance, you need to unsubscribe the order first. For details, see [Unsubscribing a Yearly/Monthly GaussDB Instance](#).)
- After an instance is deleted, the system retains the most recent automated full backup of the instance from one day ago (or, if none exists, the latest automated full backup) and then creates a new full backup. You can restore the instance later by using any of the backups.

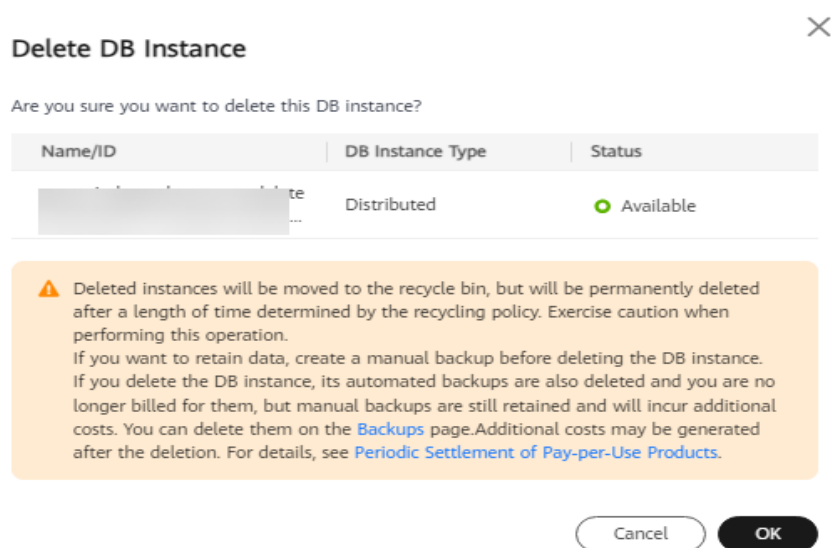
Procedure

Step 1 [Log in to the management console](#).

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

- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, locate the instance you want to delete and click **More > Delete** in the **Operation** column.
- Step 5** In the displayed dialog box, click **OK**. Refresh the **Instances** page later to check that the deletion is successful.

Figure 7-17 Deleting an instance



- Step 6** If you have enabled the operation protection function, click **Start Verification** in the **Delete DB Instance** dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

Two-factor authentication improves the security of your account. For details about how to view and enable high-risk operation protection, see [Identity and Access Management User Guide](#).

----End

FAQ

Question:

Why is my instance still billed after I delete it?

Answer:

1. Pay-per-use instances are postpaid resources. You are not billed immediately after a bill is generated. For example, if you delete a GaussDB instance (which is billed on an hourly basis) at 08:30, the expenditures for that hour from 08:00 to 09:00 are usually not billed until about 10:00.
2. The instance has manual backups. After an instance is deleted, its manual backups are retained by default and will be billed based on the backup space pricing. If you no longer need the manual backups and want to stop being billed for them, simply [delete them](#).

Related Operations

[Modifying the Recycle Bin Policy for a GaussDB Instance](#)

[Rebuilding a GaussDB Instance](#)

[Unsubscribing a Yearly/Monthly GaussDB Instance](#)

7.11 Rebuilding a GaussDB Instance

Scenarios

When you delete an instance, the system automatically creates a full backup and stores it in the recycle bin. Within the retention period, you can restore (also known as rebuild in this context) this instance by creating a new instance using the backup file. However, note that the restored instance will have a different private IP address from the original one.

Procedure

Step 1 [Log in to the management console](#).

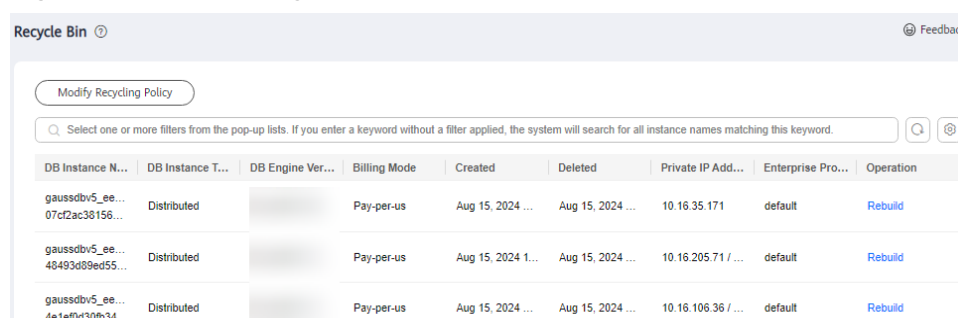
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 In the navigation pane on the left, choose **Recycle Bin**.

Step 5 Locate the instance to be rebuilt and click **Rebuild** in the **Operation** column.

Figure 7-18 Rebuilding an instance



DB Instance N...	DB Instance T...	DB Engine Ver...	Billing Mode	Created	Deleted	Private IP Add...	Enterprise Pro...	Operation
gaussdbv5_ee... 07cf2ac38156...	Distributed		Pay-per-us	Aug 15, 2024 ...	Aug 15, 2024 ...	10.16.35.171	default	Rebuild
gaussdbv5_ee... 48493d09ed55...	Distributed		Pay-per-us	Aug 15, 2024 1...	Aug 15, 2024 ...	10.16.205.71 / ...	default	Rebuild
gaussdbv5_ee... 4e1ef0d30fb34...	Distributed		Pay-per-us	Aug 15, 2024 ...	Aug 15, 2024 ...	10.16.106.36 / ...	default	Rebuild

Step 6 On the displayed page, configure required parameters and submit the task.

Figure 7-19 Billing mode and basic information

Billing Mode

Yearly/Monthly

Pay-per-use

?

Region

Project

DB Instance Name

gauss-7e3d

?

Edition Type

Enterprise edition

DB Engine Version

V2.0-8.210

DB Instance Type

Centralized

Deployment Model

1 primary + 2 standby

?

AZ

cn-north-4a

cn-north-4b

cn-north-4c

AZ7

Only one or three AZs can be selected.

Time Zone

(UTC+08:00) Beijing, Chongqing, Hong K...

Table 7-5 Basic information

Parameter	Description
Billing Mode	GaussDB provides yearly/monthly billing and pay-per-use billing. Yearly/Monthly: You pay upfront for the amount of time you expect to use the DB instance for. You will need to make sure you have a top-up account with a sufficient balance or have a valid payment method configured first. For distributed instances using the combined deployment model, yearly/monthly billing is only available to authorized users. To apply for the permissions needed, submit a service ticket. Pay-per-use: You can start using the DB instance first and then pay as you go. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.
DB Instance Name	The instance name is case-sensitive, must start with a letter, and can contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Failover Priority	This parameter is only available for distributed instances. This parameter is only available for authorized users. To apply for the permissions needed, submit a service ticket. The default value is Reliability. For details about how to change the failover priority for an existing instance, see Changing Failover Priority. • Reliability: Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency. • Availability: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. If Availability is selected, exercise caution when modifying the following database parameters. For details about how to modify parameters, see Modifying Instance Parameters. – recovery_time_target: If the value of this parameter is changed, the instance will experience frequent forced failovers. Contact technical support before modifying this parameter. – audit_system_object: If the value of this parameter is changed, DDL audit logs will be lost. Contact technical support before modifying this parameter.
AZ	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. A DB instance can be deployed in one AZ or three AZs. Three AZs are recommended. To deploy a GaussDB instance in Intelligent EdgeCloud (IEC), select an IEC HomeZones for AZ. For assistance with this setup, submit a service ticket to contact Huawei engineers.
Time Zone	Select a time zone according to the region hosting your DB instance when you buy the instance.

Figure 7-20 Specifications and storage

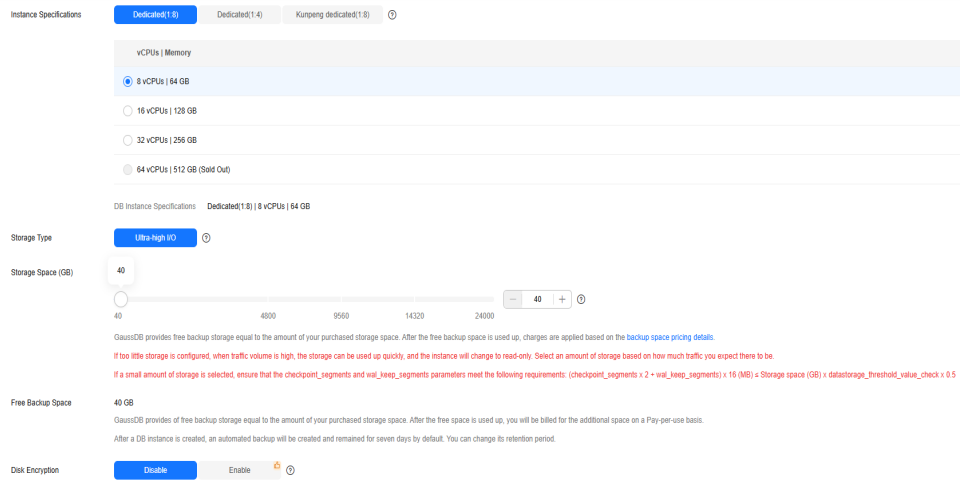


Table 7-6 Specifications and storage

Parameter	Description
Instance Specifications	CPU and memory specifications of the instance. Different instance specifications support different numbers of database connections. For details, see Instance Specifications .
Storage Space	The allocated storage space contains necessary file system overhead required for inodes, reserved blocks, and database operations. The storage for a single shard starts from 40 GB and can be increased at a step of 4 GB.

Parameter	Description
Disk Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. Key Name: If disk encryption is enabled, you need to select or create a key. NOTE • When creating a pay-per-use instance with disk encryption enabled, ensure that your account has been granted the following action: kms:cmk:createGrant • When you create a yearly/monthly instance with disk encryption enabled, if the KMS key shared by RAM is used for the first time, create an agency as prompted. • When using the KMS key shared by RAM to create an instance, ensure that your account has also been granted the following actions: – iam:agencies:listAgencies – iam:agencies:createAgency – iam:roles:listRoles – iam:permissions:listRolesForAgency – iam:permissions:grantRoleToAgency – kms:cmk:createGrant If a shared KMS key is used, the corresponding CTS event is createGrant. Only the key owner can receive this event.

Figure 7-21 Network and database configuration

Relationship among VPCs, subnets, security groups, and DB instances. ⓘ

VPC ⓘ ⓘ ⓘ

If you want to create a VPC, go to [the VPC console](#).

The subnet your DB instance will be deployed on cannot be changed later. Make sure it has enough private IP addresses available for future scale-ups. Available private IP addresses in the selected subnet: 239

Security Group ⓘ [View Security Groups](#) ⓘ ⓘ

In a security group, rules that authorize connections to DB instances apply to all DB instances associated with the security group.

Ensure that the TCP ports in the inbound rules of the selected security group contain 8000-8100, 20050, 5000-5001, 2379-2380, 6000, 6500.

Security Group Rules

Database Port

Single Private IP Address ☐

The single private IP address policy is disabled by default, but you can enable it if you want to create an instance with only one private IP address. Instances with this policy enabled do not support IPv6.

Table 7-7 Network and database configuration

Parameter	Description
VPC	A virtual network where your GaussDB instances are located. A VPC isolates networks for different workloads. You need to create or select the required VPC. If no VPC is available, GaussDB allocates a default VPC for you. After the GaussDB instance is created, the VPC cannot be changed. For details about how to create a VPC, see Creating a VPC. With VPC sharing, you can also use a VPC and subnet shared by another account. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). This allows for more efficient use of network resources and reduces O&M costs. For more information about VPC subnet sharing, see VPC Sharing in the <i>Virtual Private Cloud User Guide</i>.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within a specific AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets in which you plan to create GaussDB instances and cannot be disabled. GaussDB supports automatic IP address allocation during instance creation. If an IEC HomeZones is selected for AZ, create a subnet in that zone for this parameter. NOTE By default, a subnet supports up to 256 IP addresses. A distributed instance can require up to 1,286 IP addresses. You are advised to use a subnet that can provide 2,048 IP addresses.

Parameter	Description
Security Group	Controls access to and from an instance. When you buy an instance, the selected security group must meet the following requirements: - When buying a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6001, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound security group rules allow all outbound traffic. - When buying a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.) The security group enhances security by controlling access to GaussDB from other services. Ensure that the security group you selected allows your client to access the instance. If you do not need to specify a security group when creating an instance, you can submit a service ticket. If no security group is available, GaussDB allocates a default security group for you.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: 8000. The following ports are used by the system and cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.
Single Private IP Address	Specifies whether to enable the single private IP address policy. If this policy is enabled, only one private IP address is assigned to an instance and is bound to the primary node. The private IP address does not change after a primary/standby switchover. If this policy is disabled, each node is bound to a private IP address, and the private IP address changes after a primary/standby switchover. The constraints on the single private IP address policy are as follows: - This policy is only available for centralized instances of version V2.0-3.207 or later. - This policy is configurable only at the time of instance creation and cannot be changed afterward.

Figure 7-22 Database configuration

Administrator: root

Administrator Password: [password field] [Keep your password secure. The system cannot retrieve your password.](#)

Confirm Password: [password field]

Parameter Template: [Default-Enterprise-Edition-GaussDB-V2.0...] [View Parameter Templates](#)

Enterprise Project: [default] [View Enterprise Projects](#)

Tag: TMS's predefined tags are recommended for adding the same tag to different cloud resources. [Create predefined tags](#)

[+ Add Tag](#)

Tags you can still add: 20

Table 7-8 Database configuration

Parameter	Description
Administrator	Database administrator. The default username is root .
Administrator Password	Enter a strong password and periodically change it to improve security, preventing security risks such as brute force cracking. The password must contain: • 8 to 32 characters. • At least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*-_+=? , Keep the password secure. Password retrieval is not supported.
Confirm Password	Enter the administrator password again.
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. After creating an instance, you can modify the parameter template.
Enterprise Project	If the instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the Enterprise Project Management console to create a project. For details, see Enterprise Management User Guide.

Parameter	Description
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each instance can have up to 20 tags. If your organization has configured tag policies for GaussDB, add tags to instances based on the policies. If a tag does not comply with the policies, instance creation may fail. Contact your organization administrator to learn more about tag policies.

Step 7 After the rebuilding task is submitted, check the instance status on the **Instances** page. The rebuild is complete when the status shows **Available**.

----End

7.12 Stopping a GaussDB Node

Scenarios

You can manually stop a GaussDB instance node. This operation terminates all database processes on that node and shuts down the VM hosting it. However, resources like EIPs, storage, and backups are retained. If you are using DB instances mainly for routine development, you can stop nodes of pay-per-use instances during idle periods to cut down on costs.

Constraints

- Stopping a node may expose the instance to higher risk of downtime. Exercise caution when performing this operation.
- The following operations cannot be performed when a node is being stopped: changing specifications, backing up data, resetting passwords, rebooting an instance, and deleting an instance. When you perform these operations, nodes cannot be stopped.
- HA monitoring will be disabled for a node before it is stopped. HA monitoring will be enabled when you start the node. You can also manually enable HA monitoring for a node. After HA monitoring is disabled, the node status will not be monitored.
- If a stopped node contains at least half of the replicas of the corresponding DN shard, the instance may be abnormal and a single node may fail to be started.
- After a node is stopped, you can still log in to the node using CLI commands, but all database-related operations cannot be performed.
- For a distributed instance, at least one CN in the instance must be available, or the entire instance will be unavailable.
- Instance parameters cannot be modified if there are stopped nodes in an instance.
- If a node is stopped for too long, an alarm is triggered. You can [start the node](#) to rectify the fault and clear the alarm.

Billing

For nodes of pay-per-use instances, billing for the VM hosting it stops when the node is stopped, but billing continues for associated resources like EIPs, storage, and backups. For nodes of yearly/monthly instances, all related resources are billed even after the node is stopped.

Procedure

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the **Node List** area, click **Stop** in the **Operation** column.
- Step 6** In the **Stop Node** dialog box, enter **YES**, and click **OK**.

Figure 7-23 Stopping a node

Stop Node

Stop this instance node?

Node Name	Role	Status
gauss-af3d_root...	Primary	 Available



- If a primary DN is stopped, a primary/standby switchover will be triggered, which may interrupt services for several seconds or minutes. So you should stop such nodes during off-peak hours.
- Stopping a node reduces the number of replicas of an instance, which reduces the instance HA capability.
- If a stopped node contains at least half of the replicas of the corresponding DN shard, the instance may be abnormal and that single node may be unable to start. You are advised to stop all nodes and then start the instance. Exercise caution when performing this operation.
- After a node is stopped, the primary and standby statuses of instance nodes may be unbalanced, and they cannot be automatically balanced after the node starts. You can contact the administrator to balance the primary and standby nodes. If you have to stop a node, restart it as soon as possible.

For nodes of pay-per-use instances, billing for the VM hosting it stops when the node is stopped, but billing continues for associated resources like EIPs, storage, and backups. For nodes of yearly/monthly instances, all related resources are billed even after the node is stopped.

To confirm stopping the node, enter "YES" below. [Auto Enter](#)

YES

Cancel

OK

Step 7 Refresh the instance and view the status of the node. If its status is **Stopped**, it has been stopped successfully.

----End

7.13 Starting a GaussDB Node

Scenarios

GaussDB allows you to manually start a stopped node.

Precautions

- Only nodes in the **Stopped** state can be started.
- The following operations cannot be performed when a node is being started: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting an instance, and deleting an instance. When you perform these operations, nodes cannot be started.
- After a node is started, the distribution of primary and standby nodes for the instance may be unbalanced. Contact O&M personnel to determine whether to balance the primary and standby statuses for the instance.
- If the number of stopped nodes in a DN shard exceeds half of the replicas of the shard, a single node may fail to be started.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Node List** area, click **Start** in the **Operation** column.

Figure 7-24 Starting a node

Node Name	Node ID	Role	Status	Operation
gaus-ct_0	21f5c546f21e4f559b3b43da36f0f75eno14	Primary	Available	Reboot Stop Node Bind the EIP
gaus-ct_1	6cc9f25cdf8347e5bd8cb60a35cf6333no14	Standby	Available	Reboot Stop Node Bind the EIP
gaus-ct_2	1d30cec4f61c4d35b81282534c0366ceno14	Standby	Stopped	Reboot Start Node Bind the EIP...

Step 6 In the displayed dialog box, enter **start**, and click **OK**.

Step 7 Refresh the instance and view the status of the node. If its status is **Available**, it has been started successfully.

----End

7.14 Rebooting a GaussDB Node

Scenarios

If the status of a GaussDB instance node is abnormal, you can reboot the node to restore the node status. You can also reboot a node when it is in the **Available** state. A node is not available when it is being rebooted.

Precautions

- You can reboot a node when the DB instance is in the following state or performing the following operations:
 - Backup and restoration failed
 - Changing the billing mode from pay-per-use to yearly/monthly
 - DR in progress for the primary instance in a streaming DR task
 - Caching logs for the primary instance in a streaming DR task
 - DR simulation in progress for the DR instance in a streaming DR task
 - DR in progress for the DR instance in a streaming DR task
 - DR instance promoted to primary in a streaming DR task
- Rebooting nodes will clear the cached memory in them. To prevent traffic congestion during peak hours, you are advised to reboot nodes during off-peak hours.
- Only nodes of centralized instances can be rebooted.
- A primary/standby switchover will be triggered if a primary node is rebooted.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 Click the target instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, click **Reboot** in the **Operation** column of a node. Confirm information about the node to be restarted, enter **reboot**, and click **OK**.

Figure 7-25 Rebooting a node

Node List

Switch Primary and Standby DN's

All statuses  

Node Name	Node ID	Role	Status	Operation
gauss-1e4d-kyl_root_0	ade2d8c650da446e92bcac5dd772b49no14	Primary	 Available	Reboot
gauss-1e4d-kyl_root_1	1b5866a2778947e58667d5bc4bd6f8dano14	Standby	 Available	Reboot
gauss-1e4d-kyl_root_2	5f678df49d6468983711b822c191aa3no14	Standby	 Available	Reboot

Figure 7-26 Confirming the reboot

Reboot

Are you sure you want to reboot this node?

Node Name	Role	Status
gauss-1e4d-kyl_root_0	Primary	Available

This node is not available when it is being rebooted. Rebooting the node will clear the cached memory in them. To prevent traffic congestion during peak hours, you are advised to reboot the node during off-peak hours.

To confirm the reboot, enter "reboot" below. [One-key input](#)

Enter reboot

Cancel

OK

The node status becomes **Rebooting node**.

Step 6 Refresh the instance basic information and check the reboot result. If the node status is **Available**, the reboot is successful.

----End

8 Instance Modifications

8.1 Changing the Name of a GaussDB Instance

Scenarios

You can change the name of an instance.

Constraints

You cannot perform the following operations when the instance name is being changed:

- Binding an EIP
- Deleting the instance
- Creating a backup for the instance

Precautions

- The new name of an instance can be the same as an existing instance name.
- Changing the name of a DB instance does not disassociate the associated tags from the instance.
- If a DB instance is renamed, backups of the DB instance are still retained.

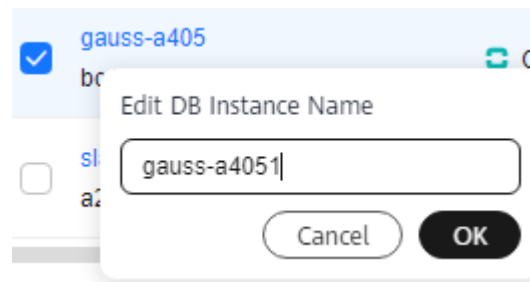
Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, locate the instance whose name you want to edit and click  next to the instance name. Then, edit the name and click **OK**.

Figure 8-1 Changing the name of an instance

Alternatively, click the instance name to go to the **Basic Information** page. In the **Basic Information** area, click  next to the **DB Instance Name** field to edit the instance name.

The name must start with a letter and consist of 4 to 64 characters. It can contain only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_).

- To submit the change, click .
- To cancel the change, click .

Step 5 View the new instance name.

----End

8.2 Changing the Database Port of a GaussDB Instance

Scenarios

You can change the database port of your GaussDB instance.

Precautions

- The database port can be changed only for instances of version V2.0-2.0 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- Changing the port of an instance will reboot all nodes of the instance, during which services will be intermittently interrupted.
- The GaussDB port ranges from 1024 to 39989, but the following ports that are reserved for system use cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.
- The port cannot be a number in the range [*Database_port*, *Database_port* + 10].

Procedure

Step 1 [Log in to the management console](#).

- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases** > **GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance.
- Step 5** In the **Network Information** area on the **Basic Information** page, click  in the **Database Port** field.

Figure 8-2 Changing the database port

Network Information			
VPC	Subnet	Security Group	Database Port
vpc-default-auto	subnet-default-auto(10.16.0.0/16)	Sys_default	8000 

- To submit the change, click .
 - In the displayed dialog box, click **Yes** to submit the change.
 - In the displayed dialog box, click **No** to cancel the change.
- To cancel the change, click .

Step 6 View the result of the change on the **Basic Information** page.

----End

8.3 Changing the M Compatibility Port

Scenarios

GaussDB allows you to use the **templatem** template to create an M-compatible database. You can enable, disable, or change the M compatibility port for a centralized instance on the console for better compatibility.

Constraints

- M-compatible databases can be created only for centralized instances of version V2.0-8.100 or later and distributed instances of version V2.0-8.200 or later. The distributed instances must be newly created, instead of those upgraded from an earlier version. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- For centralized instances earlier than V2.0-8.210 and distributed instances, only one M-compatible database can be created. For centralized instances of version V2.0-8.210 or later, you can create multiple M-compatible databases.
- M-compatible databases do not support database- and table-level backup and restoration.
- The M compatibility port cannot be changed for GaussDB instances that contain a database named **templatem**.
- This operation is not allowed for DR instances. To enable the M compatibility port for a DR instance, delete its DR relationship first.

- Enabling the M compatibility port will reboot the DB instance, during which services are interrupted. Perform this operation during off-peak hours.
- The port must have been enabled in the inbound rule of the security group.

Procedure

Step 1 [Log in to the management console](#).

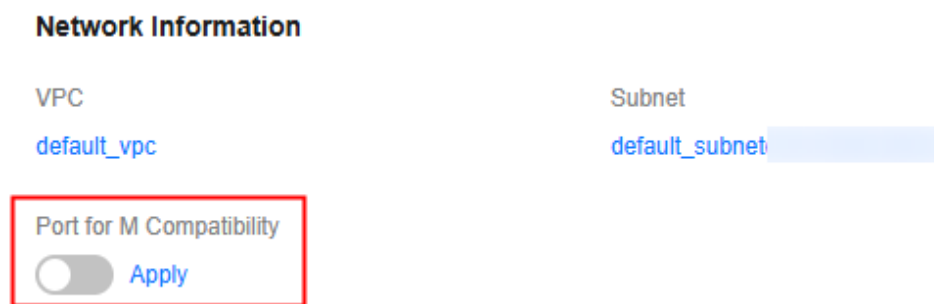
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Network Information** area, click  or **Apply** in the **Port for M Compatibility** field.

Figure 8-3 Setting the M compatibility port



Step 6 In the displayed dialog box, set the M compatibility port, and click **OK**.

- The port ranges from 1024 to 39989, but the following ports that are reserved for system use cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009, 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001
- The port cannot be a number in the range [*Database port*, *Database port* + 10].

Figure 8-4 Enabling M compatibility port

Enable M Compatibility Port

Warning: The port must be enabled in the inbound rule of the security group. Changing the port used for M compatibility will reboot the DB instance, during which services will be intermittently interrupted. Therefore, perform this operation during off-peak hours.

DB Instance Name

Port for M Compatibility

Enter a port. ?

Cancel OK

Step 7 Check the result.

During the modification, the instance is in the **Changing M compatibility port** state. After the instance status becomes **Available**, check the port displayed in the **Port for M Compatibility** field.

----End

8.4 Changing the CPU and Memory Specifications of a GaussDB Instance

Scenarios

You can change the CPU and memory specifications of an instance to match your workload requirements. This includes scaling up or down the specifications of your GaussDB instance.

Precautions

- Before you change the instance specifications, ensure that the instance is available. If the instance or node is abnormal, or the storage space is full, you cannot perform this operation.
- During the specification change for an HA (1 primary + 2 standby) instance, a primary/standby failover is triggered. During the failover, services are interrupted for about 1 minute.
- For a single-replica instance, changing instance specifications will reboot the instance and interrupt services for 5 to 10 minutes.
- After you change instance specifications, the DB instances will be rebooted and services will be interrupted. You are advised to perform this operation during off-peak hours.

- If the instance load is heavy, it takes a longer time to change its instance specifications.
- If there is only one coordinator node in an instance, services will be interrupted during the specification change. Exercise caution when performing this operation.

Billing

Billing Mode	Operation	Impact on Fees
Yearly/ Monthly	Specification upgrade	After instance specifications are upgraded, the new instance specifications take effect in the original usage period. You need to pay for the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition 1 primary + 2 standby instance of V2.0-8.210 (instance specifications: dedicated, 4 vCPUs 16 GB; storage: ultra-high I/O, 160 GB) on June 1, 2025. The instance price was \$2,064 USD/month. On June 15, 2025, you changed the instance specifications to 8 vCPUs 32 GB. The instance price became \$4,032 USD/month. Price difference = Price for the new instance specifications x Remaining period – Price for the original instance specifications x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 4032 x 0.5 – 2064 x 0.5 = \$984 USD

Billing Mode	Operation	Impact on Fees
	Specification downgrade	After instance specifications are downgraded, the new instance specifications take effect in the original usage period. GaussDB refunds the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition 1 primary + 2 standby instance of V2.0-8.210 (instance specifications: dedicated, 8 vCPUs 32 GB; storage: ultra-high I/O, 160 GB) on June 1, 2025. The instance price was \$4,032 USD/month. On June 15, 2025, you changed the instance specifications to 4 vCPUs 16 GB. The instance price became \$2,064 USD/month. Price difference = Price for the original instance specifications x Remaining period – Price for the new instance specifications x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 4032 x 0.5 – 2064 x 0.5 = \$984 USD
Pay-per-use	Specification upgrade	After instance specifications are changed, the new instance specifications are billed by hour. For details, see Product Pricing Details .
	Specification downgrade	

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, locate the instance and choose **More > Change Instance Specifications** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Configuration** area, click **Change** in the **Instance Specifications** field.

Step 5 On the displayed page, specify the new instance specifications and click **Next**.

Step 6 Confirm the specifications and click **Submit**.

Step 7 View the new instance specifications.

After the task is submitted, click **Go to Instance List**. On the **Instances** page, the DB instance status is **Changing instance specifications**. After a few minutes, check the new instance specifications on the **Basic Information** page.

-----End

Reference

[Instance Specifications](#)

8.5 Configuring Read Replicas

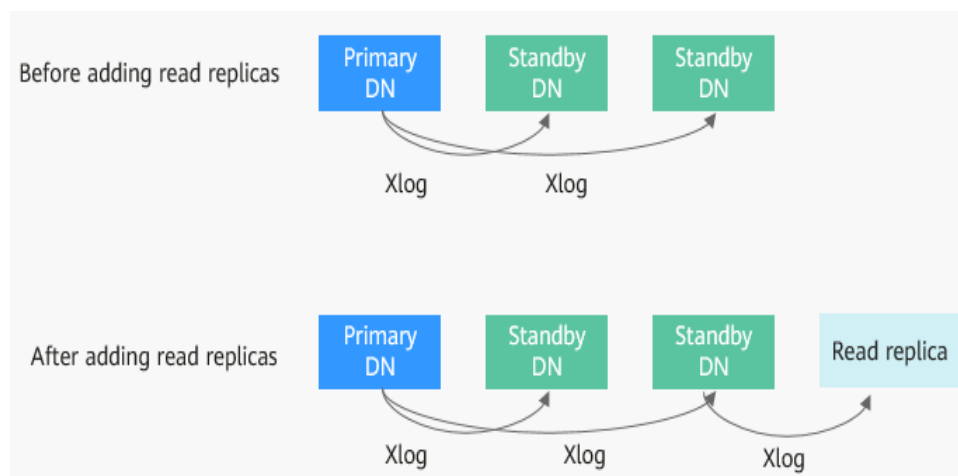
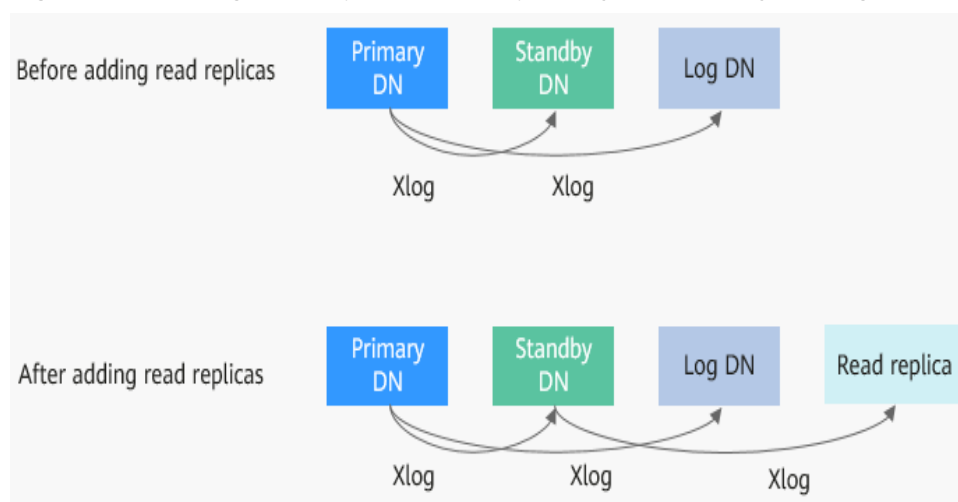
Scenarios

In a centralized instance, a surge in read requests can consume excessive compute resources. This heavy load may interfere with transaction log (Xlog) replay on standby nodes, leading to increased replication lag between primary and standby nodes. If the primary node fails, the accumulated Xlogs may delay failover. To mitigate this, you can add read replicas to offload and distribute read workloads.

You can also delete read replicas or adjust their specifications to better match changing read demands.

Technical Details

The primary node handles both read and write operations, while standby and log nodes synchronize Xlogs from it. When read traffic increases, high compute resource usage can hinder Xlog replay on standby nodes. Adding read replicas helps offload read requests. They synchronize Xlogs from standby nodes, reducing the primary node's workload and maintaining stable Xlog replay.

Figure 8-5 Adding read replicas for a 1 primary + 2 standby instance**Figure 8-6** Adding read replicas for a 1 primary + 1 standby + 1 log instance

Precautions

- Read replica-related operations are supported for centralized (1 primary + 2 standby) instances of V2.0-3.209 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- Read replica-related operations are supported for centralized (1 primary + 1 standby + 1 log) instances of V2.0-8.0.0 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- After snapshot-based backup is enabled, read replicas cannot be added.
- Adding read replicas temporarily stops backups. Once the replicas are ready, the system re-enables backups and runs a full backup automatically.

Billing

Billing Mode	Operation	Impact on Fees
Pay-per-use	Adding a read replica	Added read replicas are billed by the hour.
	Deleting a read replica	Deleted read replicas are no longer billed.

Adding a Read Replica

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Configuration** area, click **Add** in the **Read Replicas** field.

Step 6 Select the specifications and parameter template of the read replica, and click **Add Read Replica**.

Figure 8-7 Adding a read replica

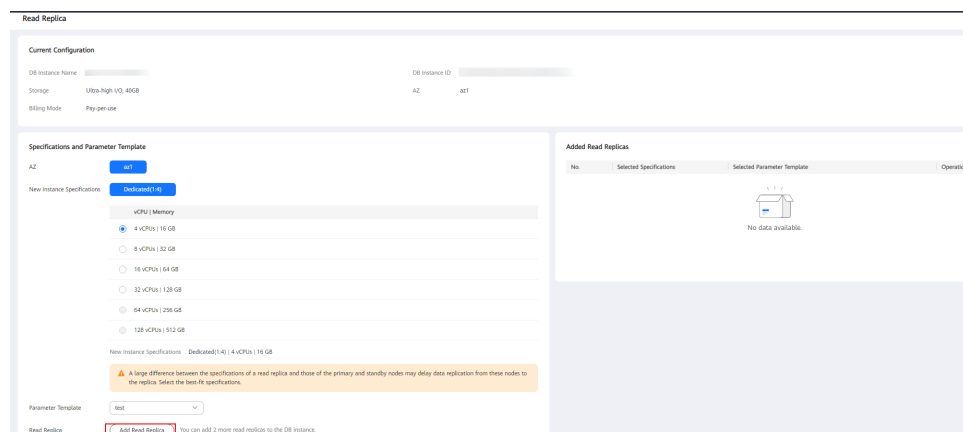


Table 8-1 Parameter description

Parameter	Description
AZ	An AZ is a physical region where resources have independent power supply and networks. AZs are physically isolated but interconnected through an internal network. Select either one or three AZs.
New Instance Specifications	CPU and memory specifications of the instance. Different instance specifications have different numbers of database connections. For details, see Instance Specifications .
Read Replica	A maximum of three read replicas can be added to a single instance.

Step 7 Click **Next**, confirm the information, and click **Submit**.

Step 8 Check the result of adding a read replica.

On the **Instances** page, the instance status is **Creating read replicas**. After the instance status becomes **Available**, click the instance name. On the **Basic Information** page, check the number of read replicas.

----End

Deleting a Read Replica

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Configuration** area, click **Delete** in the **Read Replicas** field.

Step 6 Select the read replica to be deleted and click **Next**.

Step 7 Confirm the information and click **Submit**.

Step 8 Check the result of deleting a read replica.

On the **Instances** page, the instance status is **Deleting a read replica**. After the instance status becomes **Available**, click the instance name. On the **Basic Information** page, check the number of read replicas.

----End

Changing the Specifications of Read Replicas

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Configuration** area, click **Change** in the **Read Replicas** field.

Step 6 On the displayed page, select the target read replica and new specifications, and click **Next**.

Step 7 Confirm the specifications and click **Submit**.

Step 8 Check the specification change result.

After the task is submitted, click **Go to Instance List**. On the **Instances** page, the instance status is **Changing instance specifications**. After a few minutes, check the new instance specifications on the **Basic Information** page.

----End

8.6 Synchronizing Data to a Single-Replica Instance

Scenarios

GaussDB supports data synchronization from a three-replica instance to a single-replica instance. After replication, the single-replica instance contains the same data as the three-replica instance at the time of synchronization.

Precautions

- If the instance or any of its nodes is abnormal, you cannot perform this operation.
- If you are performing other operations on a single-replica instance, this operation is unavailable.
- After this operation is performed, all data on the target single-replica instance, including its password, will be overwritten.

Procedure

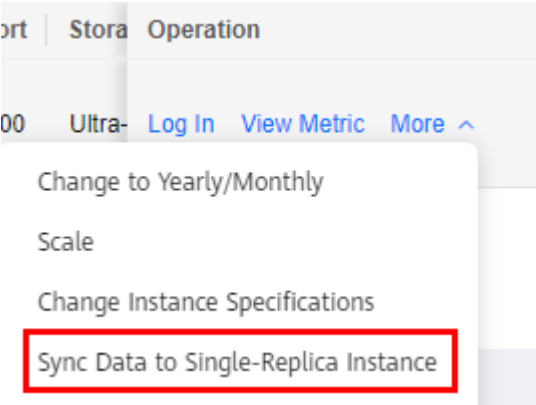
Step 1 [Log in to the management console.](#)

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

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

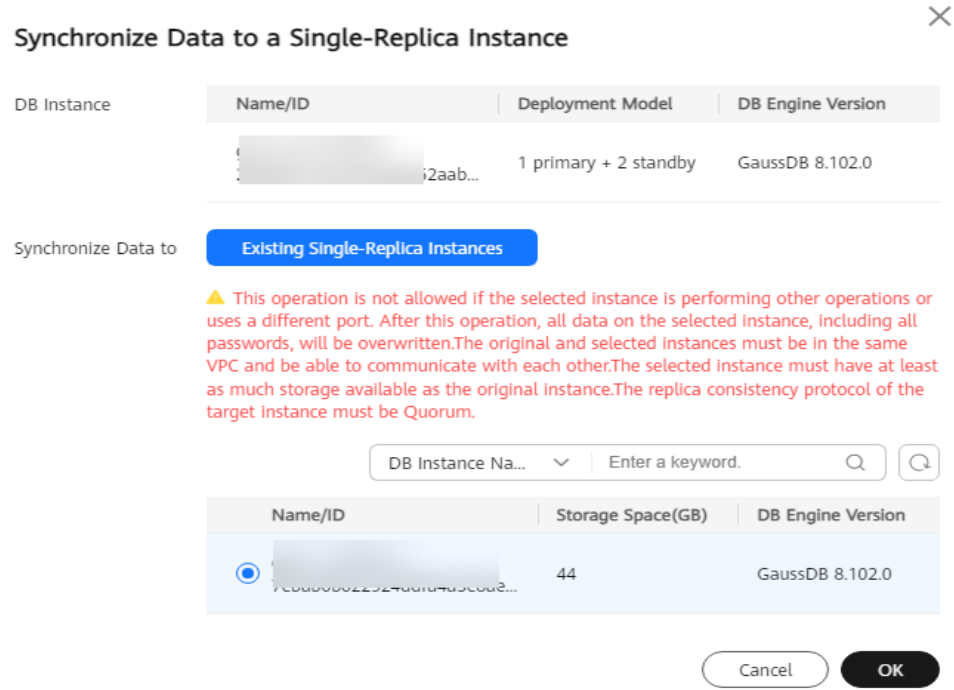
Step 4 On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Sync Data to Single-Replica Instance**.

Figure 8-8 Choosing Sync Data to Single-Replica Instance



Step 5 In the displayed dialog box, select the target instance and click **OK**.

Figure 8-9 Synchronizing data to a single-replica instance



-----End

8.7 Viewing and Modifying Advanced Features

Scenarios

You can enable advanced features such as advanced compression, dynamic data masking, and anti-tampering for your instance on the instance details page.

Precautions

- Advanced compression status can only be changed for instances of the enterprise or basic edition whose version is V2.0-8.1.0 or later.
- Dynamic data masking and anti-tampering are only supported for instances of the enterprise edition whose version is V2.0-8.0.0 or later.

For details about how to view the instance version, see [Checking the DB Engine Version](#).

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Click **View and Modify** in the **Advanced Features** field of the **Configuration** area.

Step 6 In the feature list, change the feature statuses or values, enter **Yes**, and click **OK**.

Figure 8-10 Modifying advanced features for an instance of the enterprise edition

View and Modify Advanced Features

Advanced Feature	Value	Allowed Values	Description
ledger	off	on off	Specifies whether to enable the le...
ilm	off	on off	Specifies whether to enable OLTP ...
security	off	on off	If a security policy is required, set t...

To confirm the enablement, enter "YES" below. [Auto Enter](#)

Cancel

OK

Figure 8-11 Modifying the advanced feature for an instance of the basic edition

View and Modify Advanced Features ✕

Advanced Feature	Value	Allowed Values	Description
ilm	off	on/off	Specifies whether to enable OLTP ...

To confirm the enablement, enter "YES" below. [Auto Enter](#)

Cancel OK

Step 7 Wait until the modification is complete.

-----End

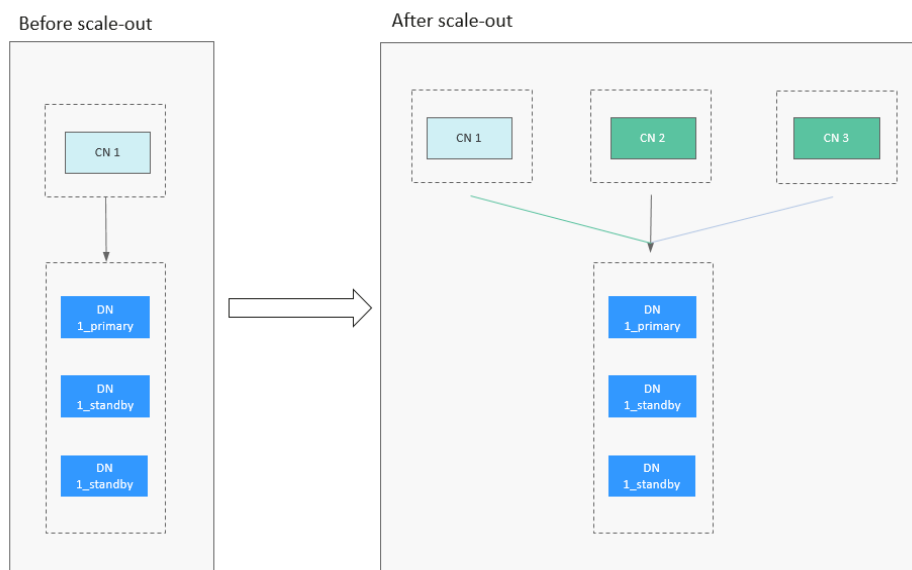
8.8 Scaling In and Out an Instance

8.8.1 Overview of Scaling In and Out an Instance

After you purchase a GaussDB instance, your resource needs might change due to workload changes. For such cases, GaussDB distributed instances that are independently deployed can be scaled in or out. You can adjust resources as required. Currently, instances can be scaled in or out in the following ways:

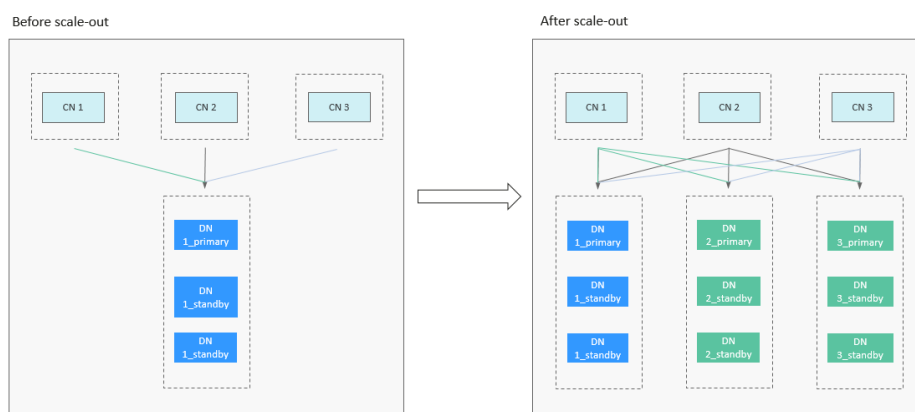
Adding CNs for an Instance

Instances can be scaled out by adding coordinator nodes (CNs). For example, if the original instance is configured with one CN, three replicas, and one shard, the instance will have three CNs, three replicas, and one shard after two CNs are added. For details, see [Adding Coordinator Nodes for an Instance \(Distributed\)](#).

Figure 8-12 Adding CNs for an instance

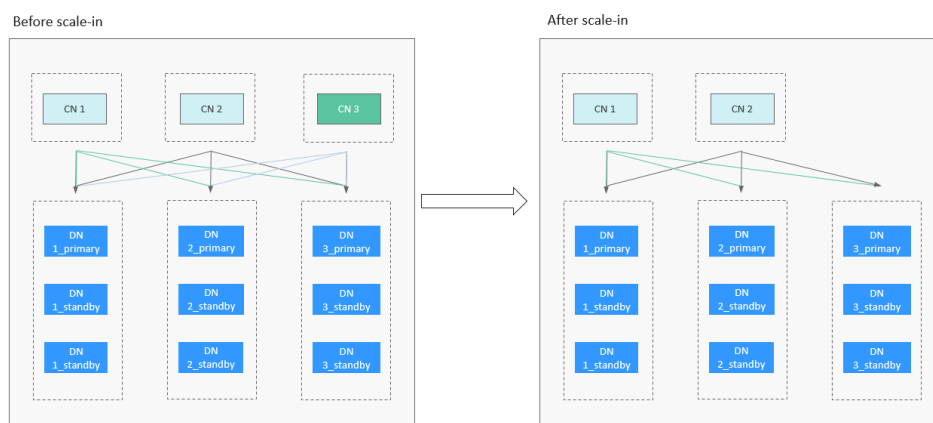
Adding Shards for an Instance

Instances can be scaled out by adding shards. For example, if the original instance is configured with three CNs, three replicas, and one shard, the instance will have three CNs, three replicas, and three shards after two shards are added. For details, see [Adding Shards for an Instance \(Distributed\)](#).

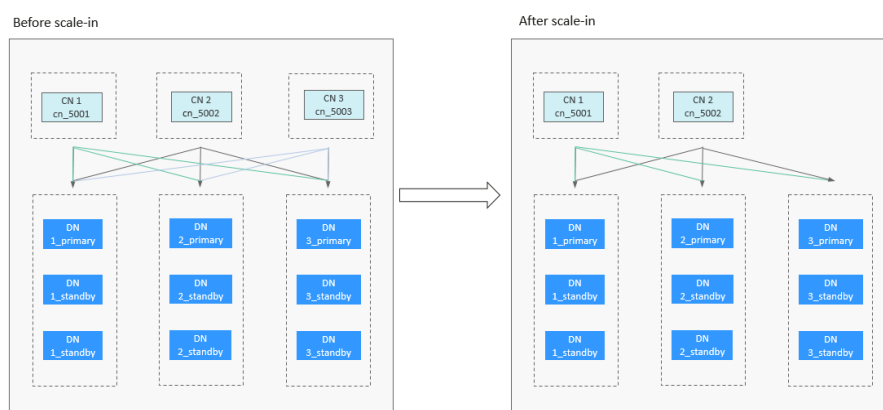
Figure 8-13 Adding shards for an instance

Deleting CNs for an Instance

Instances can be scaled in by deleting CNs. For example, if the original instance is configured with three CNs, three replicas, and three shards, the instance will have two CNs, three replicas, and three shards after one CN is deleted. For details, see [Deleting Coordinator Nodes for an Instance \(Distributed\)](#).

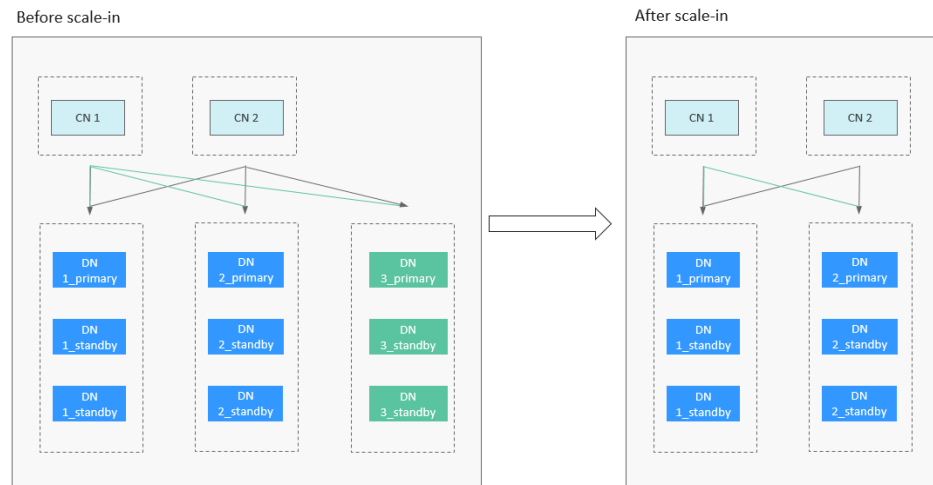
Figure 8-14 Deleting CNs for an instance

Main processes are running on the main CN (that is, the CN whose component ID is cn_5001), so this CN cannot be deleted for scale-in. You can call the [Querying the Components of a DB Instance](#) API to query cn_5001. To scale in the instance, another CN will be deleted. In the example shown in [Figure 8-15](#), the CN whose component ID is cn_5003 is deleted instead.

Figure 8-15 Deleting CNs for an instance

Deleting Shards for an Instance

Instances can be scaled in by deleting shards. For example, if the original instance is configured with two CNs, three replicas, and three shards, the instance will have two CNs, three replicas, and two shards after one shard is deleted. For details, see [Deleting Shards for an Instance \(Distributed\)](#).

Figure 8-16 Deleting shards for an instance

8.8.2 Adding Coordinator Nodes for an Instance (Distributed)

Scenarios

As the instance deployment time and data increase, the database performance and storage will gradually reach the bottleneck. Adding nodes can improve the instance performance and storage capacity. You can only add nodes for distributed GaussDB instances that are deployed independently.

Precautions

- The scaling duration depends on the amount of data. The default timeout period is seven days. When an instance is being scaled out, the instance is available, but you are not allowed to perform other operations on the console. If you need to perform any operations, [submit a service ticket](#).
- You can flexibly add coordinator nodes (CNs) or shards as needed. It is recommended that the number of CNs of a GaussDB instance do not exceed twice the number of shards.
- Instances can be scaled out only when they are in the **Available** state.
- You can add up to 64 CNs at a time, but the system supports no more than 256 CNs in total.

Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After a CN is added, you will be billed for the new CN based on the time remaining in the requested period of your instance. You need to pay for the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNs; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, you added a CN (specifications: dedicated, 8 vCPUs 64 GB) for the instance. Its price was \$1,640 USD/month. Price difference = Fee of the new CN x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 1640 x 0.5 = \$820 USD
Pay-per-use	Added CNs are billed by the hour. For details, see Product Pricing Details .

Procedure

Step 1 [Log in to the management console](#).

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

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

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

Step 5 On the **Basic Information** page, click **Add** in the **Coordinator Nodes** field.

Step 6 Specify the number of CNs to be added, the subnet, and the AZ.

Figure 8-17 Adding coordinator nodes

Add Coordinator Node

Current Configuration

DB Instance Name	gauss-yl	DB Instance ID	8fe2f6f0ac4d6aaa44f48cc3873e27m14
Storage	Extreme SSD, 40GB	AZ	az1
Shards	1	Coordinator Nodes	1

Coordinator Node Specifications gaussdb.opengauss.ee.cn.m6.2xlarge.8.in | 8 vCPUs | 64 GB

New Nodes Coordinator nodes you can still create: 255 (Max. allowed each time: 64)

AZ Deployment

AZ

Current Subnet subnet-7517(172.16.1.0/24)

Subnet Available private IP addresses in the selected subnet: 170

Search

subnet-7517(172.16.1.0/24)

subnet-9222(172.16.0.0/24)

erred. If you want to use a different subnet, ensure that the subnet of the instance can communicate with the selected subnet.
s added.
sts for nodes. The settings of the two subnets are the same. Otherwise, nodes may be unable to communicate with each other.

- If single-AZ deployment is specified during instance creation, CNs are only added to the AZ you specified.
- You can assign a different subnet for new CNs, though the instance's current subnet is recommended. If you want to use another subnet, ensure that it can communicate with the subnet of the instance. Otherwise, the scale-out will fail. You can select up to two different subnets for nodes, but their settings must be identical. If they differ, the nodes might not be able to communicate with each other.

Step 7 Click **Next**.

Step 8 Confirm the information and click **Submit**.

Step 9 Check the result.

On the **Instances** page, check that the instance is in the **Adding CNs** state during the scale-out. After the instance status changes to **Available**, click the instance name. On the **Basic Information** page, check that the number of CNs matches the required quantity.

-----End

8.8.3 Adding Shards for an Instance (Distributed)

Scenarios

As the instance deployment time and data increase, the database performance and storage will gradually reach the bottleneck. In this case, you need to add hosts to improve the instance performance and storage capability. This function is available only for distributed GaussDB instances that are deployed independently.

Precautions

- The scaling duration depends on the amount of data. The default timeout period is seven days. When an instance is being scaled out, the instance is

available, but you are not allowed to perform other operations on the console. If you need to perform any operations, [submit a service ticket](#).

- Instances can be scaled out only when they are in the **Available** state. When shards are being added, you can still query and insert data, query services are not interrupted, and the data insertion performance is not affected. The performance of join queries on local tables across node groups during redistribution may be affected.
- You can add up to 64 shards at a time, but the system supports no more than 256 shards in total.

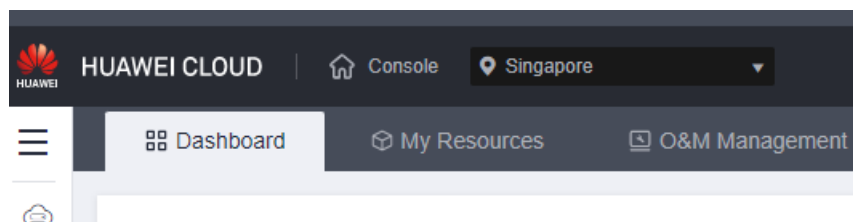
Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After a shard is added, you will be billed for the new shard based on the time remaining in the requested period of your instance. You need to pay for the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNs; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, you added a shard for the instance, which was priced at \$4,944 USD/month. Price difference = Fee of the new shard x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 4944 x 0.5 = \$2,472 USD
Pay-per-use	Added shards are billed by the hour. For details, see Product Pricing Details .

Procedure

Step 1 [Log in to the management console](#).

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



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

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

Step 5 On the **Basic Information** page, click **Add** in the **Shards** field.

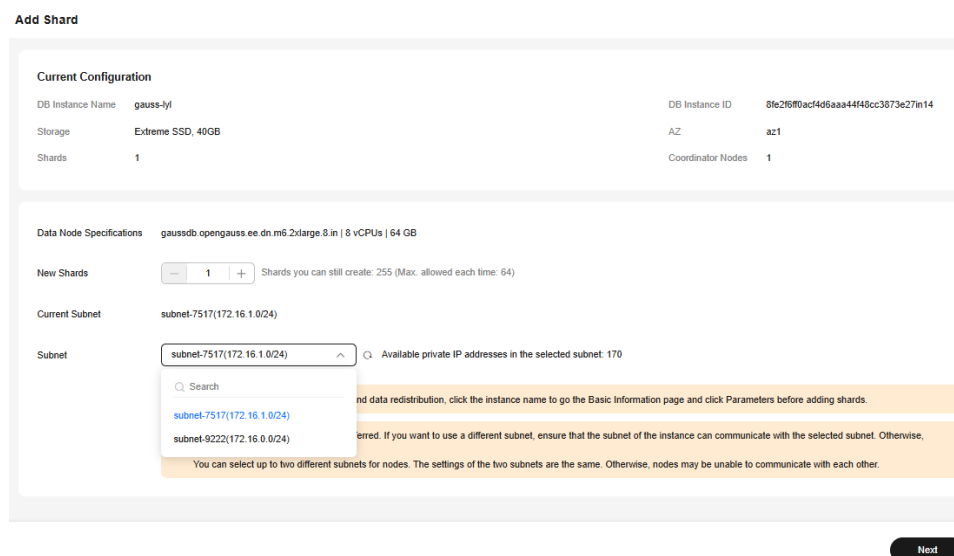
Figure 8-18 Adding shards



Step 6 Specify the number of shards to be added and the subnet. Click **Next**.

You can assign a different subnet for new shards, though the instance's current subnet is recommended. If you want to use another subnet, ensure that it can communicate with the subnet of the instance. Otherwise, the scale-out will fail. You can select up to two different subnets for nodes, but their settings must be identical. If they differ, the nodes might not be able to communicate with each other.

Figure 8-19 Adding shards



Step 7 Confirm the information and then click **Submit**.

 NOTE

By default, a shard contains three replicas (a primary DN and two standby DNs). Each time you add a shard, three replicas will be added.

Step 8 Check the result.

On the **Instances** page, check that the instance is in the **Adding shards** state during the scale-out. After the instance status changes to **Available**, click the instance name. On the **Basic Information** page, check that the number of shards matches the required quantity.

----End

8.8.4 Deleting Coordinator Nodes for an Instance (Distributed)

Scenarios

As service demand decreases, some coordinator nodes (CNs) are left idle. To improve resource utilization, you can delete unnecessary coordinator nodes. This function is available only for distributed GaussDB instances that are deployed independently.

Precautions

- Deleting CNs does not interrupt ongoing services.
- You can only delete the CNs of instances that were deployed independently.
- At least one CN needs to be reserved for each DB instance.
- Before deleting a CN, ensure that the CN is not in a JDBC connection configuration, or the high availability of the JDBC connection may be affected.
- DDL operations will be rolled back when CNs are being deleted.
- PITR backup is suspended during the deletion and is automatically restored after deletion is complete.
- After the deletion is complete, a full backup is performed automatically.
- Before you delete CNs, you need to ensure that the instance status and all CNs are normal.
- Main processes are running on the main CN (that is, the CN whose component ID is cn_5001), so this CN cannot be deleted for scale-in. You can call the [Querying the Components of a DB Instance](#) API to query cn_5001. If the CN to be deleted is cn_5001, the system will randomly select another CN to delete.

Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After a CN is deleted, the system will stop charging you for it in the original subscription period. GaussDB refunds the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNs; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, you deleted a CN from the instance, which was priced at \$1,640 USD/month. Price difference = Fee of the deleted CN x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: - Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 - Price difference = 1640 x 0.5 = \$820 USD
Pay-per-use	Deleted CNs are no longer billed.

Procedure

Step 1 [Log in to the management console.](#)

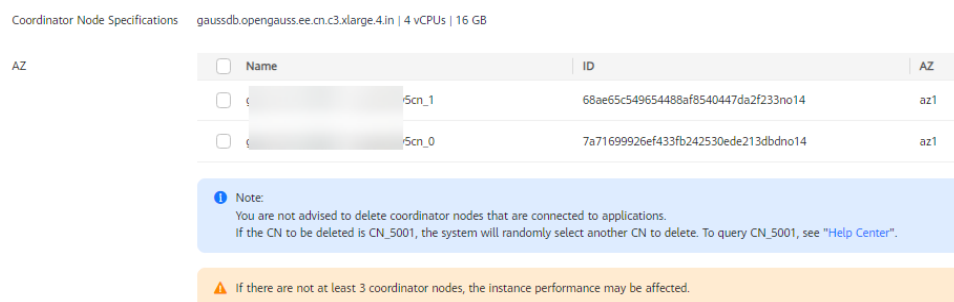
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, click the name of the instance for which you want to delete CNs.

Step 5 In the **DB Information** area of the **Basic Information** page, delete CNs.

1. Click **Delete** next to **Coordinator Nodes**.
2. Select the coordinator nodes to be deleted.

Figure 8-20 Deleting CNs

3. Click **Next**.
4. Confirm the information about the CNs to be deleted and click **Submit**.
5. Check the result.

On the **Instances** page, check that the instance is in the **Deleting coordinator nodes** state during the scale-in. After the instance status changes to **Available**, click the instance name. On the **Basic Information** page, check that the number of CNs matches the required quantity.

----End

8.8.5 Deleting Shards for an Instance (Distributed)

Scenarios

There may be more than enough DNs in your DB instance after read/write splitting is enabled or redundant business data is cleared. You can delete shards as needed to avoid cost waste. This function is available only for distributed GaussDB instances that are deployed independently.

Precautions

- The scaling duration depends on the amount of data. The default timeout period is seven days. When an instance is being scaled in, the instance is available, but you are not allowed to perform other operations on the console. If you need to perform any operations, [submit a service ticket](#).
- When shards are being deleted, existing sessions on the DNs in the deleted shards will be cleared, and some services will be affected. Therefore, delete shards during off-peak hours.
- There must be at least one shard in a DB instance. The storage space of the instance after the deletion must meet and following condition: Used space of the current instance/Number of DNs after the deletion + Maximum capacity of a table/Number of DNs after the deletion) < Read-only threshold (85%) x Disk capacity
- PITR backup is suspended during the deletion and is automatically restored after deletion is complete.
- After the deletion is complete, a full backup is performed automatically.
- DB instances can be scaled in only when they are in the **Available** state. When shards are being deleted, you can still query and insert data, query

services are not interrupted, and the data insertion performance is not affected. The performance of join queries on local tables across node groups during redistribution may be affected.

Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After a shard is deleted, the system will stop charging you for it in the original subscription period. GaussDB refunds the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNs; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, you deleted a shard from the instance, which was priced at \$4,920 USD/month. Price difference = Fee of the deleted shard x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: - Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 - Price difference = 4920 x 0.5 = \$2,460 USD
Pay-per-use	Deleted shards are no longer billed.

Procedure

Step 1 [Log in to the management console.](#)

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

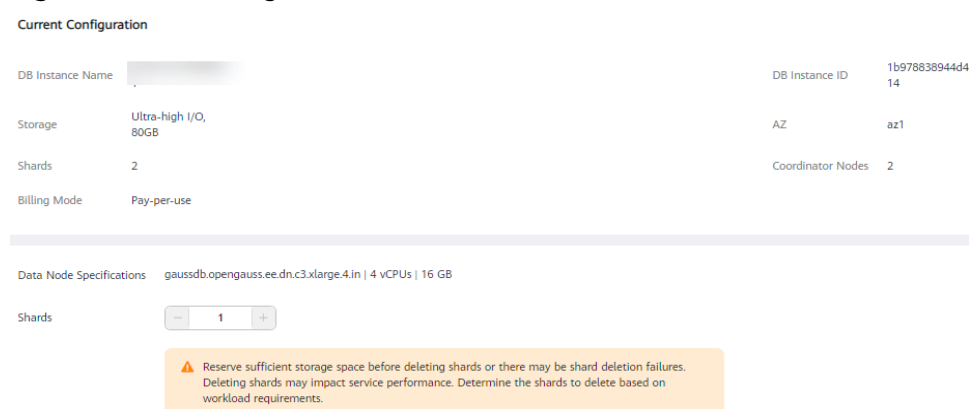
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

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

Step 5 On the **Basic Information** page, click **Delete** in the **Shards** field.

Figure 8-21 Deleting shards

Step 6 Select the number of shards to be deleted and click **Next**.

Figure 8-22 Deleting shards

Step 7 Confirm the information and then click **Submit**.

NOTE

By default, a shard contains three replicas (a primary DN and two standby DNs). Each time you delete a shard, three replicas will be deleted.

Step 8 Check the result.

On the **Instances** page, check that the instance is in the **Deleting shards** state during the scale-in. After the instance status changes to **Available**, click the instance name. On the **Basic Information** page, check that the number of shards matches the required quantity.

-----End

8.9 Scaling Up Storage Space

8.9.1 Overview of Scaling Up Storage Space

As more data is added, you may start to run out of space. This section describes how to scale up the storage space of a DB instance. [Table 8-2](#) describes the scale-up methods provided by GaussDB.

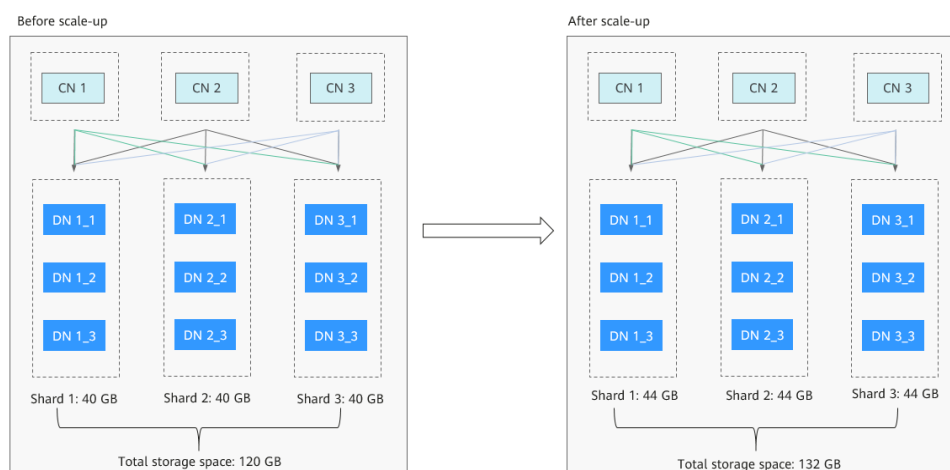
Table 8-2 Scale-up methods

Scale-up Method	Supported Instance Type	Scope	Description
Manually Scaling Up Storage Space for an Instance	• Distributed • Centralized	All shards	Shard-level scale-up is performed based on the selected storage space. All shards are scaled up at a time. The added storage space must be a multiple of 4 GB x <i>Number of shards</i> .
Manually Scaling Up Storage Space of Specified Shards	Distributed	• Specified shards • All shards	Shard-level scale-up is performed based on the selected storage space. One or more shards are scaled up at a time. The added storage space must be a multiple of 4 GB. This method is only available for distributed instances using independent deployment.

Scale-up Method	Supported Instance Type	Scope	Description
Configuring Storage Autoscaling for an Instance	• Distributed • Centralized	• Specified shards • All shards	Shard-level scale-up is performed based on the user-defined scaling increment (specified by the Scale Up By parameter). When the available storage of a DN component drops to or below the specified threshold, an automatic scale-up task is triggered to scale up the shard where the DN component with insufficient available storage is located. Storage can be scaled by percentage or a fixed amount. • Percentage: The storage is scaled up by a percentage you specify. The added storage space is the storage space of the shard multiplied by the scaling increment. • Fixed amount: The storage is scaled up by a fixed amount you specify. The added storage space is a fixed value. Autoscaling is unavailable for distributed instances using combined deployment.

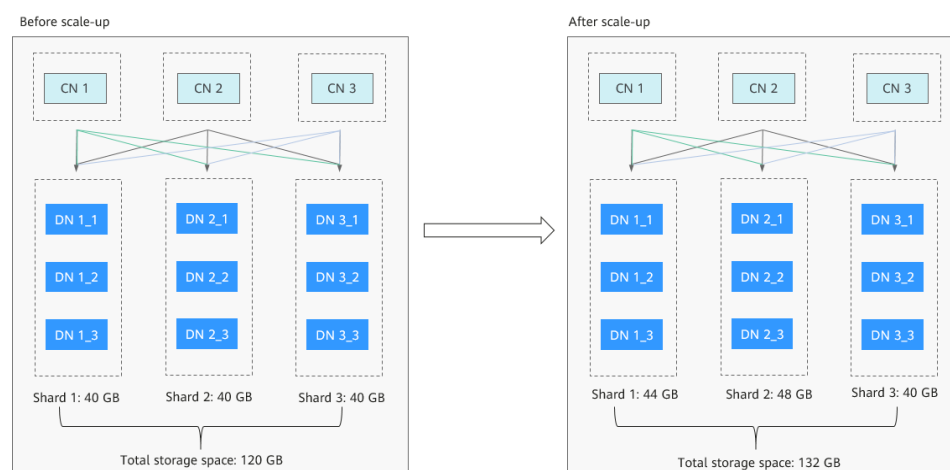
Manually Scaling Up Storage Space for an Instance

Take an independently deployed distributed instance as an example. This instance has three shards, three replicas, and three CNs, which are the default configurations, with a total storage space of 120 GB before the scale-up. This means that each shard has a storage space of 40 GB. If 12 GB of storage is added, the total storage space is 132 GB after scale-up, including 4 GB for each shard. After the scale-up, each shard has a storage space of 44 GB.

Figure 8-23 Scaling up storage space

Manually Scaling Up Storage Space of Specified Shards

Take an independently deployed distributed instance as an example. This instance has three shards, three replicas, and three CNs, which are the default configurations, with a total storage space of 120 GB before the scale-up. This means that each shard has a storage space of 40 GB. If you increase the storage space of shard 1 and shard 2 by 4 GB and 8 GB, respectively, the total storage space after scale-up is 132 GB. The storage spaces of shard 1 and shard 2 are 44 GB and 48 GB, respectively.

Figure 8-24 Scaling up storage space

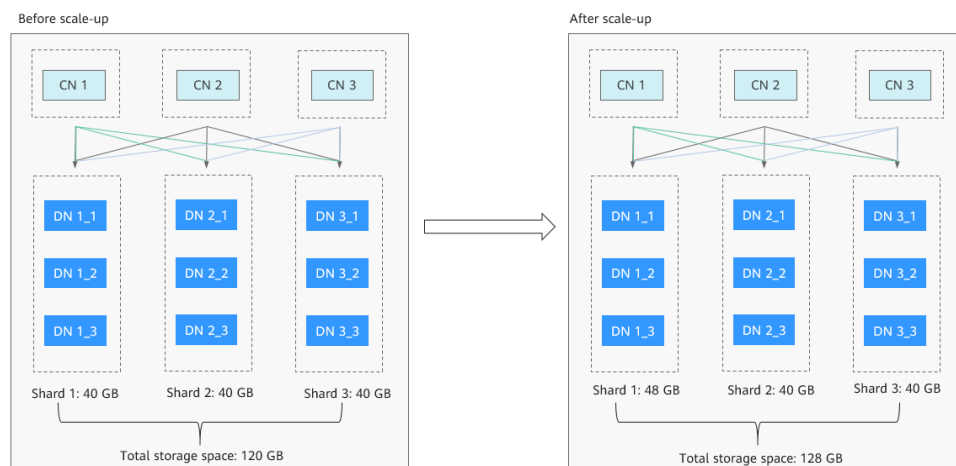
Configuring Storage Autoscaling for an Instance

- By percentage

Take the independently deployed distributed instance with three shards, three replicas, and three CNs as an example. Its total storage space before the scale-up is 120 GB, with each shard holding 40 GB. In the autoscaling policy, both the **Trigger If Available Storage Drops To** and **Scale Up By** parameters are set to **20%**. If the available storage usage of shard 1 drops to 20% or lower, a scale-up action is automatically triggered. The added storage space

of shard 1 is 8 GB (40 x 20%). After the scale-up, the storage space of shard 1 is 48 GB, and the total storage space of the instance is 128 GB.

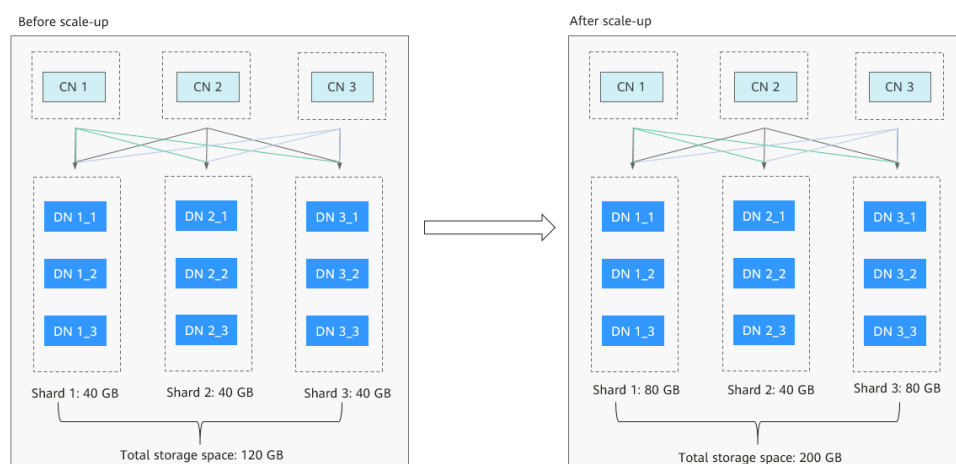
Figure 8-25 Scaling up storage space



- By fixed amount

Take the independently deployed distributed instance with three shards, three replicas, and three CNs as an example. Its total storage space is 120 GB before scale-up, and each shard has a storage space of 40 GB. In the autoscaling policy, the **Trigger If Available Storage Drops To** parameter is set to **20%**, and the **Scale Up By** parameter is set to **40 GB**. If the available storage usage of the DN 1_1 and DN 3_3 components drops to or below 20%, a scale-up action is automatically triggered for the two shards where DN 1_1 and DN 3_3 are located. A 40 GB storage increment is added to both shards. After the scale-up, their storage space is both 80 GB, and the total storage space of the instance is 200 GB.

Figure 8-26 Scaling up storage space



8.9.2 Manually Scaling Up Storage Space for an Instance

Scenarios

As more data is added, you may start to run out of space. If the kernel system detects that the storage usage exceeds the predefined threshold, the instance is set to read-only and no data can be written to it. (The default threshold is 85%. You can [modify the instance parameter `cms:datastorage\_threshold\_value\_check`](#) to change the usage threshold.) This section describes how to scale up the storage space of a DB instance. Services will not be interrupted during storage scale-up.

Precautions

- Within the maximum allowed range, the storage usage cannot exceed the predefined threshold (85% by default) after the scale-up.
- If any node becomes faulty, [submit a service ticket](#) to contact the O&M engineers for troubleshooting before the scale-up.
- The storage space must be a multiple of (Number of shards x 4 GB).
- A single shard can hold up to 24 TB
- If a DB instance is unavailable because the storage space is used up, you can scale up the storage space.
- If a DR relationship has been established for the instance and the instance functions as the primary one, ensure that the storage of its DR instances is greater than or equal to that of the primary instance during storage scale-up. Otherwise, disaster recovery may fail if the data volume of the primary instance is greater than that of the DR instance.

Constraints

- You can scale up storage space only when your account balance is greater than or equal to \$0 USD.
- The maximum allowed storage for a single shard is 24 TB by default. There is no limit on the number of scale-ups.
- The DB instance is in the **Scaling up** state when its storage space is being scaled up and the backup services are not affected.
- Do not reboot or delete the instance whose storage is being scaled up.
- Storage space can only be scaled up, not down.

Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After storage is scaled up, you will be billed for the increased storage based on the time remaining in the requested period of your instance. You need to pay for the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNS; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, you added 132 GB of storage for the instance at a price of \$7.2 USD/month. Price difference = Fee of the increased storage x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 7.2 x 0.5 = \$3.6 USD
Pay-per-use	Increased storage is billed by the hour. For details, see Product Pricing Details .

Procedure

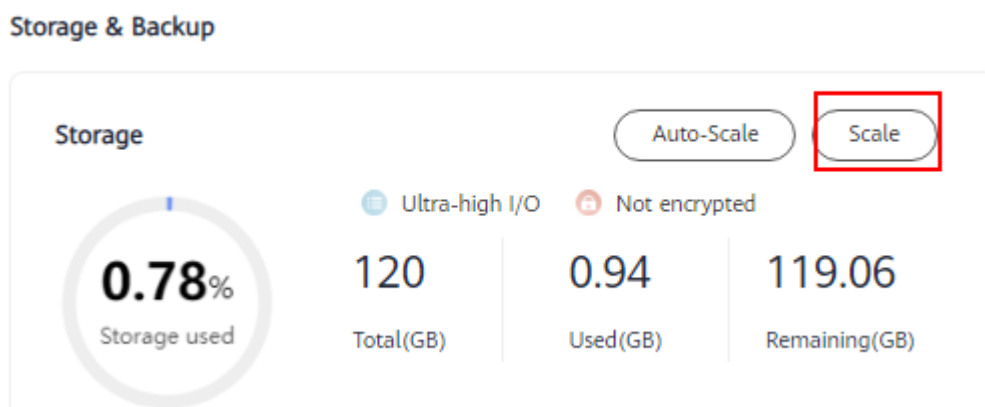
Step 1 [Log in to the management console](#).

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

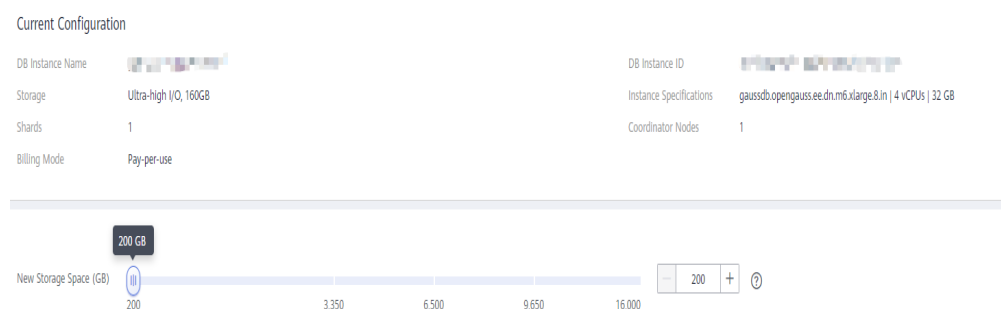
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, locate the instance you want to scale up and click **More > Scale Storage Space** in the **Operation** column.

Alternatively, click the instance name to go to the **Basic Information** page. In the **Storage** section of the **Storage & Backup** area, click **Scale**.

Figure 8-27 Scaling up storage

Step 5 On the displayed page, specify the new storage space and click **Next**.

Figure 8-28 Setting the space size

When you scale up the storage space, ensure that the usage of the new storage space is less than the predefined threshold (85% by default). An instance can be restored from read-only to the read/write state only when the storage usage is lower than the specified threshold.

Step 6 Confirm settings.

- If you need to modify your settings, click **Previous**.
- If the settings are correct, click **Submit**.

Step 7 View the storage scale-up results.

During the scale-up, the status of the instance on the **Instances** page is **Scaling up**. This process may take 3 to 5 minutes. Once the scale-up is complete, click the instance name to go the **Basic Information** page and you can see the new storage space.

----End

8.9.3 Manually Scaling Up Storage Space of Specified Shards

Scenarios

As more data is added, you may start to run out of space. If the kernel system detects that the storage usage exceeds the predefined threshold, the instance is

set to read-only and no data can be written to it. (The default threshold is 85%. You can modify the instance parameter `cms:datastorage_threshold_value_check` to change the usage threshold.) You can scale up the storage space of one or more specified shards. Services are not interrupted during storage scale-up.

Precautions

- This function is only available for distributed instances using independent deployment.
- Within the maximum allowed range, the storage usage cannot exceed the predefined threshold (85% by default) after the scale-up.
- If any node becomes faulty, [submit a service ticket](#) to contact the O&M engineers for troubleshooting before the scale-up.
- The storage space must be a multiple of (Number of shards x 4 GB).
- Each shard can hold up to 24 TB, so 24 TB of storage can be added for each shard added.
- If a DB instance is unavailable because the storage space is used up, you can scale up the storage space.
- The disk size of all shards must be the same.
- When you restore backup data to a new instance, the number of disks of the new instance is the number of disks in the largest shard of the original instance multiplied by the number of shards.

Constraints

- You can scale up storage space only when your account balance is greater than or equal to \$0 USD.
- The maximum allowed storage is 24 TB. There is no limit on the number of scale-ups.
- The DB instance is in the **Scaling up** state when its storage space is being scaled up and the backup services are not affected.
- Reboot is not required during instance scale-up.
- You cannot reboot or delete an instance that is being scaled up.
- Storage space can only be scaled up, not down.

Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After storage is scaled up, you will be billed for the increased storage based on the time remaining in the requested period of your instance. You need to pay for the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNS; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, you added 40 GB of storage for one of the shards in the instance at a price of \$24 USD/month. Price difference = Fee of the increased storage x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 24 x 0.5 = \$12 USD
Pay-per-use	Increased storage is billed by the hour.

Procedure

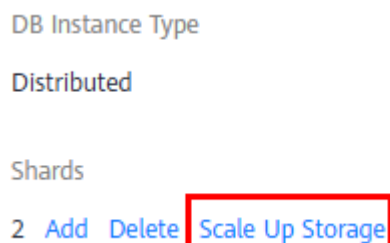
Step 1 [Log in to the management console.](#)

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

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

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

Step 5 On the **Basic Information** page, click **Scale Up Storage** in the **Shards** field.

Figure 8-29 Basic information

Step 6 On the **Scale Storage Space** page, select target shards one by one, set the new storage space, and click **Add to Scale**. Then, click **Next**.

Figure 8-30 Scaling up storage for specified shards

New Storage Space for Shards

Shard: gaussdbv5dn1(69bcd96b4ba144a9ace01ac3683bf421gr14)

Shard Name/ID	Storage	Used	Usage
gaussdbv5dn1 69bcd96b4ba144a9ace01ac3683bf421gr14	40 GB	1.05 GB	2.63%

Current Shard Storage

New Storage Space (GB): 44 GB

44 4830 9620 14410 24000

If the storage sizes of shards in the instance are different after the scale-up, no shards can be added or deleted until all shards are scaled to the same size.

GaussDB provides free backup storage equal to the amount of your purchased storage space. After the free backup space is used up, charges are applied based on the [backup space pricing details](#).

Add to Scale Add to Scale

Ensure that the usage of the new storage space is less than the predefined threshold (85% by default) after the scale-up. An instance can be restored from read-only to the read/write state only when the storage usage is lower than the specified threshold.

Step 7 Confirm settings.

- If you need to modify your settings, click **Previous**.
- If your settings are correct, click **Submit**.

Step 8 View the results.

During the scale-up, the status of the instance on the **Instances** page is **Scaling up**. Later, click the instance name to go the **Basic Information** page and view the new storage space. This process may take 3 to 5 minutes.

-----End

8.9.4 Configuring Storage Autoscaling for an Instance

Scenarios

You can enable autoscaling for a GaussDB instance so that its storage can be automatically scaled up when the disk usage reaches the specified threshold.

Precautions

- Autoscaling is unavailable for distributed instances using combined deployment.
- All nodes in the target instance must be in an available state.
- Storage autoscaling is mutually exclusive with the following operations: manually scaling up storage space, adding nodes, changing the disk type, deleting an instance, checking snapshots, updating agents, and storage autoscaling. That is, storage autoscaling cannot be performed when any of the preceding operations is ongoing, even if the autoscaling policy is configured.
- Autoscaling for centralized instances is at the instance level.
- Autoscaling for distributed instances is at the shard level.
- If the storage sizes of shards in a distributed instance are different after the scale-up, no shards can be added or deleted until all shards are scaled to the same size.
- During storage autoscaling, the storage space is increased in increments of 40 GB.
 - If the space to increase exceeds the upper limit you have specified, only the space size equal to the upper limit will be increased.
 - If the space to increase exceeds the upper limit specified by the system, only the space size equal to the upper limit will be increased.
- An alarm will be generated when autoscaling fails. This alarm will be automatically cleared when the disk usage is lower than the specified threshold.
- If a yearly/monthly DB instance has pending orders, autoscaling will fail.
- If your account balance is insufficient, autoscaling will fail.

Billing

Billing Mode	Impact on Fees
Yearly/ Monthly	After storage is scaled up, you will be billed for the increased storage based on the time remaining in the requested period of your instance. You need to pay for the difference in price based on the remaining period. Example (The prices below are for reference only. The actual prices are displayed on the console.) Suppose you purchased a one-month, enterprise-edition distributed instance of V2.0-8.210 (configurations: 3 shards, 3 replicas, 3 CNS; instance specifications: dedicated, 8 vCPUs 64 GB; storage: ultra-high I/O, 120 GB) on June 1, 2025. The instance price was \$19,752 USD/month. On June 15, 2025, 40 GB of storage was added to a shard of the instance as the instance's storage usage reached the predefined threshold. The storage was priced at \$24 USD/month. Price difference = Fee of the increased storage x Remaining period The remaining period in the formula is the number of remaining days in each calendar month divided by the maximum number of days in the calendar month. In this example, the remaining period and price difference are calculated as follows: • Remaining period = 15 (Remaining days in June)/30 (Maximum number of days in June) = 0.5 • Price difference = 24 x 0.5 = \$12 USD
Pay-per-use	Increased storage is billed by the hour.

Procedure

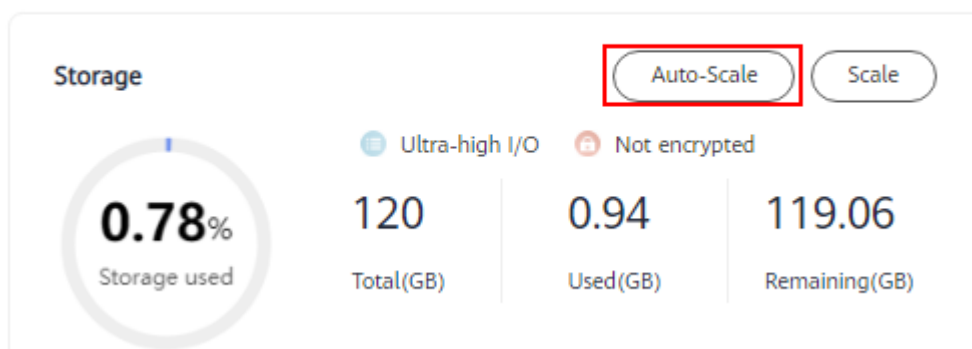
Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Storage & Backup** area, click **Auto-Scale**.

Figure 8-31 Enabling autoscaling**Storage & Backup**

Step 6 In the displayed **Configure Storage Autoscaling** dialog box, set the following parameters:

Figure 8-32 Configuring autoscaling

The figure shows the 'Configure Storage Autoscaling' dialog box. It has a title bar with a close button (X). The main content area includes: a toggle switch for 'Autoscaling' (currently off); a note 'Once enabled, an agency will be created.'; a dropdown menu for 'Trigger If Available Storage Drops To' set to '20 %'; a numeric input for 'Autoscaling Limit' set to '80 GB'; a 'Scaling Method' section with two buttons: 'By percentage' (selected) and 'By fixed amount'; a numeric input for 'Scale Up By' set to '20 %'; and a detailed note: 'If available storage drops to or below 20%, your storage will automatically scale up by 20% (in increments of 40 GB) of your allocated storage. If your account balance is insufficient, autoscaling will fail.' At the bottom right, there are 'Cancel' and 'OK' buttons.

Table 8-3 Parameters

Parameter	Description
Autoscaling	Specifies whether to enable automatic scale-out. By default, automatic scale-out is disabled.
Trigger If Available Storage Drops To	The storage will be automatically scaled up if the available storage drops to or below the threshold specified by this parameter. The default value is 20% . The value can be 20% , 25% , or 50% .
Autoscaling Limit	Upper limit of the storage space in GB that can be automatically scaled to. The value of this parameter must be greater than the current storage of the instance. Value range: - Centralized instances: [<i>Current storage</i> + 40 GB, 24,000 GB] - Distributed instances: [<i>Current storage</i> + 40 GB, 24,000 GB x <i>Number of shards</i>]
Scaling Method	The value can be By percentage or By fixed amount . - If By percentage is selected, the storage space to be expanded increases each time. - If By fixed amount is selected, a fixed volume of storage will be expanded each time. Evaluate your workloads and costs and select a method as required.

Parameter	Description
Scale Up By	Size of the storage space to be expanded each time, which depends on the selected scaling method. Storage can be scaled by percentage or a fixed amount. • If By percentage is selected for Scaling Method, the default value of this parameter is 20%, and the value range is [1%, 100%]. If the available storage drops to or below the specified threshold, the storage will be automatically scaled up by a percentage specified by this parameter (in increments of 40 GB). For example, if the current storage space of an instance is 40 GB and this parameter is set to 20%, the volume to be expanded is 8 GB, which will be rounded up to 40 GB. • If By fixed amount is selected for Scaling Method, the default value of this parameter is 40 GB, and the value range is [40 GB, <i>Current storage space</i> + 40 GB]. If the available storage drops to or below the specified threshold, the storage will be automatically scaled up by a fixed amount specified by this parameter.

Step 7 Click **OK**.

----End

8.10 Changing the Deployment Model

8.10.1 Overview of Changing the Deployment Model

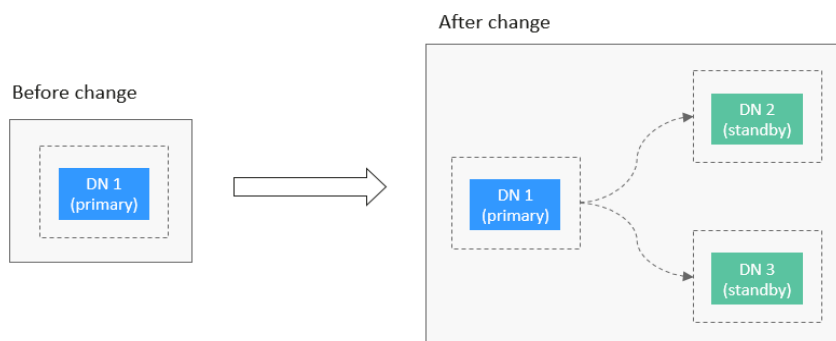
To meet various service requirements, GaussDB allows you to change the instance deployment model. Currently, the following deployment models are supported:

- For centralized instances, a single-replica instance can be changed into a 1 primary + 2 standby instance or a 1 primary + 1 standby + 1 log instance. For details, see [Changing the Deployment Model of a Single-Replica Instance \(Centralized\)](#). Note that rollback is not supported after the deployment model of a single-replica instance is changed.
- For distributed instances, a standby DN can be changed into a log node. For details, see [Changing Standby DNs to Log Nodes \(for a Distributed Instance\)](#). The log node is only used to store logs and does not incur fees, which reduces costs and resource consumption. During the change, services will be interrupted for about 1 minute. So, change the deployment model during off-peak hours.

Deployment Model Change Principle of a Single-Replica Centralized Instance

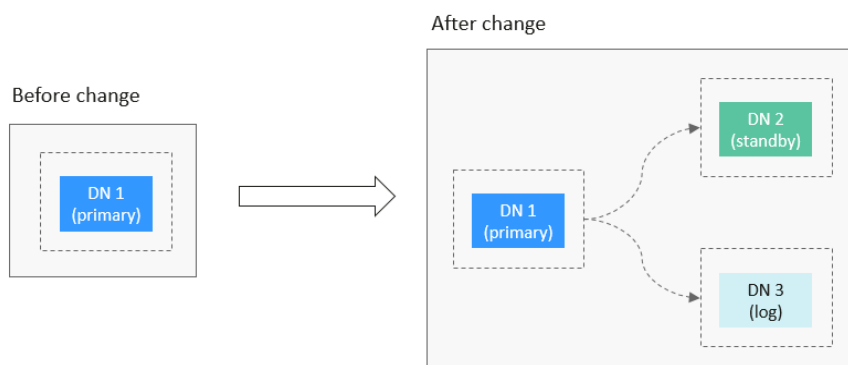
- Changing a single-replica instance into a 1 primary + 2 standby instance: In this case, two standby nodes will be added after the change.

Figure 8-33 Changing a single-replica instance into a 1 primary + 2 standby instance



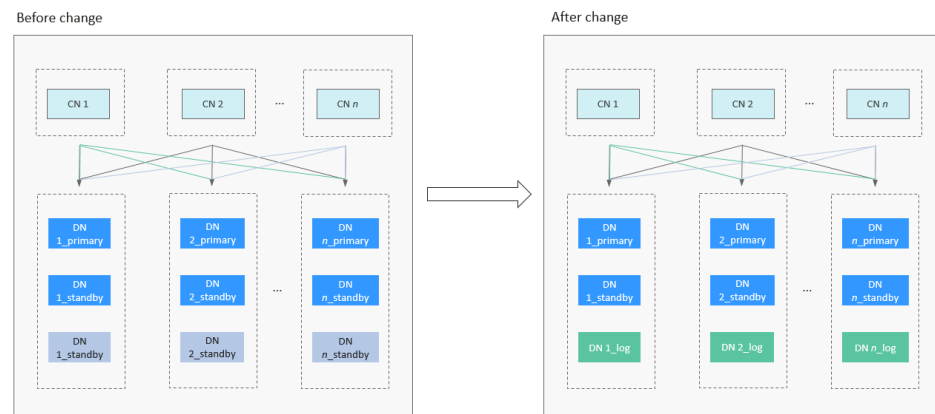
- Changing a single-replica instance into a 1 primary + 1 standby + 1 log instance: In this case, one standby node and one log node will be added after the change.

Figure 8-34 Changing a single-replica instance into a 1 primary + 1 standby + 1 log instance



Deployment Model Change Principle of a Distributed Instance

- Changing standby DNs to log nodes: Before the change, each shard of the instance has one primary DN and two standby DNs. After the change, one of the standby DNs in each shard is changed to a log node.

Figure 8-35 Changing standby DNs into log nodes

8.10.2 Changing the Deployment Model of a Single-Replica Instance (Centralized)

Scenarios

You can change the deployment model of a single-replica GaussDB instance to an instance with multiple replicas.

NOTE

This function is only available to whitelisted users. To apply for the permissions needed, [submit a service ticket](#).

Precautions

- Currently, the following deployment model change scenarios are supported:
 - Change a single-replica centralized instance to a 1 primary + 2 standby instance.
 - Change a single-replica centralized instance to a 1 primary + 1 standby + 1 log instance.
- Before changing the deployment model, ensure that the instance status is normal.
- The following operations cannot be performed when the deployment model of an instance is being changed: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting the instance, and deleting the instance.
- Changing the deployment model will interrupt services. Therefore, perform this operation during off-peak hours.
- After the deployment model is changed, the specifications of the new nodes are the same as those of the original nodes. The specifications of the log node use the configured specifications for log nodes.
- After a single-replica centralized instance is changed to a 1 primary + 1 standby + 1 log instance, its replica consistency protocol changes to Paxos.
- After the deployment model is changed, an automated backup will be triggered and log archiving will be enabled.

- Only the deployment model of single-replica instances whose version is V2.0-8.0 or later can be changed. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- Currently, only the deployment model of pay-per-use instances can be changed.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

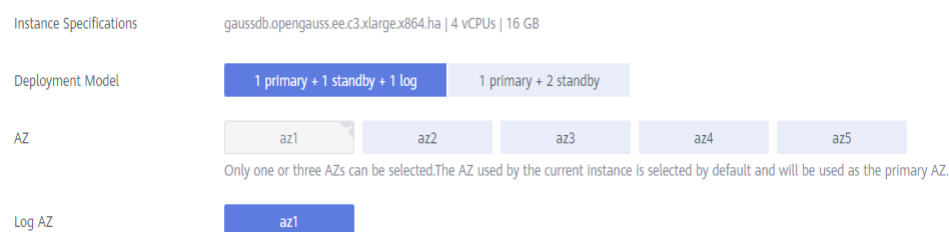
Step 5 In the **Configuration** area, click **Change** in the **Deployment Model** field. The **Change Deployment Model** page is displayed.

Figure 8-36 Changing the deployment model



Step 6 On the **Change Deployment Model** page, select the new deployment model and AZ, and click **Next**.

Figure 8-37 Changing the deployment model



Step 7 Confirm the displayed details.

- If you need to modify your settings, click **Previous**.
- If the information is correct, click **Submit** to submit the change request.

Figure 8-38 Submitting the request

Change Deployment Model		
Resource	Configuration	
DB Instance	DB Instance Name	gauss-3153-00036590
	DB Instance ID	c28659e8027c48e999e11fa028f5c0fa14
	New deployment model	Enterprise edition
	AZ	ecsaz3-x862

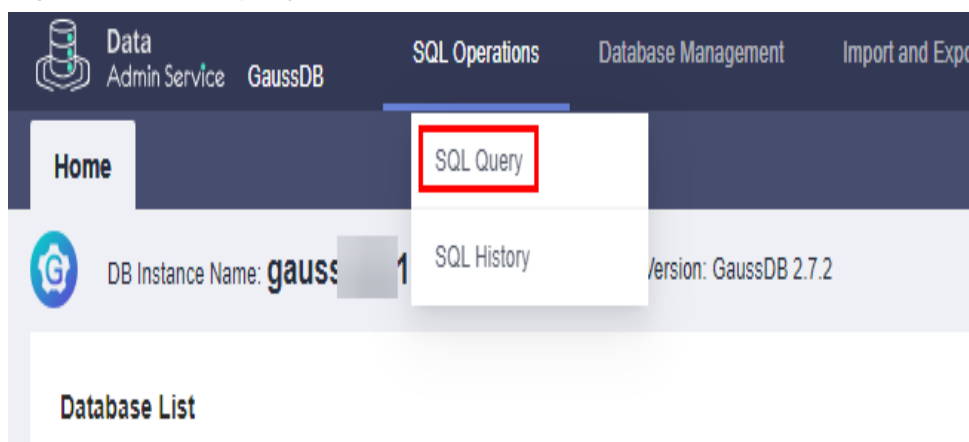
After the task is submitted, the instance status will be **Changing deployment model**.

----End

Verification After the Change

After the change is complete, check the instance status, backup creation status, and instance connectivity, and whether you can add, delete, update, and query data in the instance.

- Step 1** On the **Instances** page, check whether **Status** of the target instance is **Available**.
- Step 2** Click the name of the target instance to go to the **Basic Information** page. In the **Node List** area, check that the statuses of all nodes are **Available**.
- Step 3** Check that the automated backup triggered after the change is successfully created.
- On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
 - In the navigation pane, choose **Backups**. Check that a backup has been created and the backup status is **Completed**.
- Step 4** Check that the instance is properly connected and you can add, delete, update, and query data in the instance.
- Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
 - Go to the **SQL Query** page.

Figure 8-39 SQL query

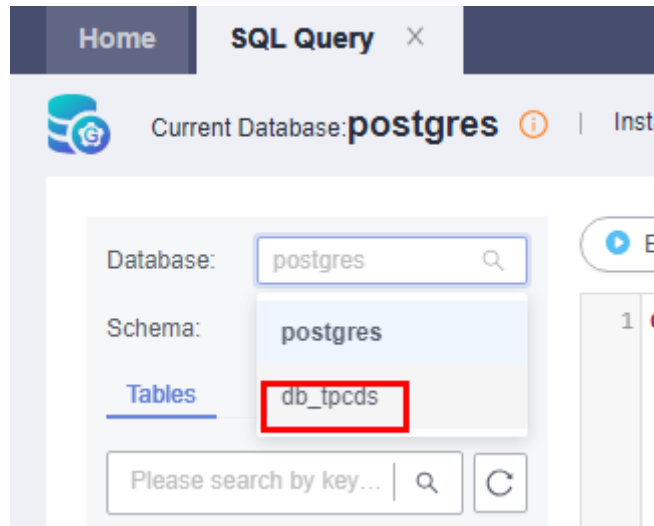
- Create a database.
CREATE DATABASE *database_name*;

In this example, run the following command to create a database named **db_tpcds**:

```
CREATE DATABASE db_tpcds;
```

Switch to the newly created database in the upper left corner.

Figure 8-40 Switching to the new database



4. Create a table and add, delete, update, and query data in the table.

- a. Create a schema.

```
CREATE SCHEMA myschema;
```

- b. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.

```
CREATE TABLE myschema.mytable (firstcol int);
```

- c. Insert data into the table.

```
INSERT INTO myschema.mytable values (100);
```

- d. View data in the table.

```
SELECT * FROM myschema.mytable;
```

	firstcol
1	100

- e. Update data in the table.

```
UPDATE myschema.mytable SET firstcol = 200;
```

- f. View the data in the table again.

```
SELECT * FROM myschema.mytable;
```

	firstcol
1	200

- g. Delete the table.

```
DROP TABLE myschema.mytable;
```

----End

8.10.3 Changing Standby DN to Log Nodes (for a Distributed Instance)

Scenarios

Each shard of a distributed GaussDB instance consists of one primary DN and two standby DNs. However, this architecture may not meet your application requirements. GaussDB provides the flexibility to convert one standby DN in each shard into a log node.

Precautions

- This function is only available for distributed instances of V2.0-3.200.0 or later that have one primary DN and two standby DNs in each shard. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- PITR backup is suspended during the process of changing standby DNs to log nodes and automatically resumes after the operation is complete.
- After standby DNs are changed to log nodes, a full backup is automatically performed.

Procedure

Step 1 [Log in to the management console](#).

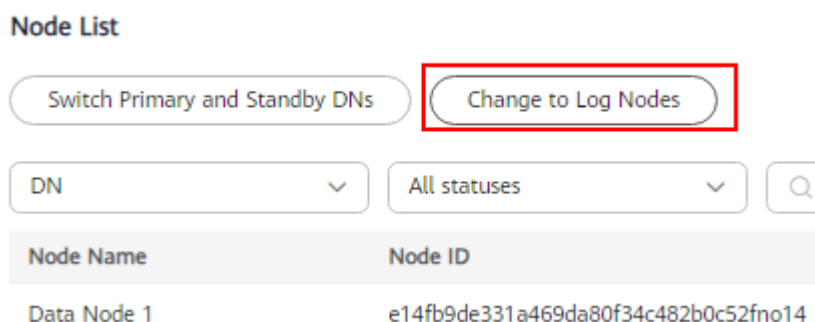
Step 2 Click  in the upper left corner and select the desired region and project.

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the **Node List** area, click **Change to Log Nodes**.

Figure 8-41 Changing the deployment model to 1 primary + 1 standby + 1 log



Step 6 In the **Change Standby Data Nodes to Log Nodes** dialog box, select an AZ and click **OK**.

Step 7 Check the change result.

After the task is submitted, click **Back to DB Instance List**. On the **Instances** page, the instance status is **Changing to log node**. After the task is complete, go to the **Basic Information** page of the instance and check that the deployment model is changed to 1 primary + 1 standby + 1 log.

----End

8.11 Performing a Primary/Standby DN Switchover

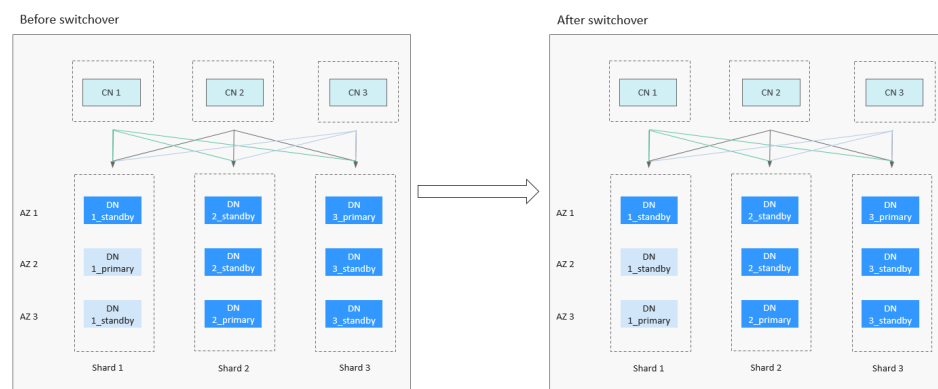
8.11.1 Overview of Performing a Primary/Standby DN Switchover

Performing a Primary/Standby DN Switchover

GaussDB supports primary/standby DN switchover in a shard in a distributed instance. You can promote a standby DN to the primary DN in a shard.

Take the distributed instances that are independently deployed as an example. The instance is configured with 3 shards, 3 replicas, and 3 AZs. If you perform a primary/standby switchover on a node in shard 1 and promote the standby node in AZ 3 to primary, the node in AZ 2 becomes the standby node and the node in AZ 3 becomes the primary node after the switchover.

Figure 8-42 Performing a primary/standby DN switchover



Changing the Failover Priority of DNs

GaussDB provides failover priority on reliability or availability.

- Reliability applies to scenarios that require high data consistency. In these scenarios, if the primary DN is faulty, services are provided only after log replay is complete to prevent data loss. During this period, connections cannot be established.
- Availability applies to scenarios that require uninterrupted online services. In these scenarios, if the primary DN is faulty and no new primary node is selected within 10 minutes in a shard, a standby node is forcibly started as the

new primary node to provide services to ensure cluster availability. In this way, the cluster can be recovered as soon as possible at the cost of data loss.

8.11.2 Changing the DN Failover Priority

Scenarios

GaussDB provides failover priority on availability or reliability. You can change the failover priority of a GaussDB instance on the **Basic Information** page. Reliability applies to scenarios that require high data consistency, and availability applies to scenarios that require uninterrupted online services.

NOTE

This function is only available to authorized users. To apply for the permissions needed, [submit a service ticket](#).

Precautions

- This function is only available to distributed instances.
- If you select **Availability** for the failover priority, exercise caution when changing the following database parameters:
 - **recovery_time_target**: Specifies the time for the standby node to write and replay logs. The value ranges from 0 to 3600, in seconds. The default value is **60**. **0** indicates that log flow control is disabled. A value from 1 to 3600 indicates that a standby node can write and replay logs within the period specified by this parameter, so that the standby node can quickly assume the primary role. If **recovery_time_target** is set to a small value, the performance of the primary node is affected. If it is set to a large value, the log flow is not effectively controlled. You are advised to retain the default value.
 - **audit_system_object**: Specifies whether to audit the CREATE, DROP, and ALTER operations on GaussDB database objects. GaussDB database objects include databases, users, schemas, and tables. The value of this parameter ranges from 0 to 536,870,911. The default value is **67121159**. You can change the value of this parameter to audit only the operations on required database objects. In the scenario where the leader node is forcibly selected, you are advised to set **audit_system_object** to the maximum value and audit all DDL objects.

Procedure

Step 1 [Log in to the management console](#).

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

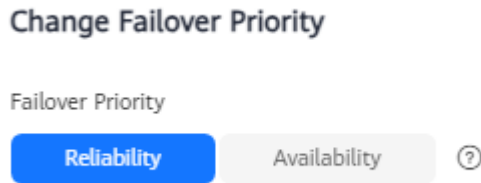
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 Click the name of the target instance to go to the **Basic Information** page.

Step 5 Click **Change** in the **Failover Priority** field.

Step 6 In the displayed dialog box, select **Reliability** or **Availability** as required.

Figure 8-43 Changing the failover priority



- **Reliability:** Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency.
- **Availability:** Database availability is prioritized during a failover. This is recommended for applications that require their databases to provide uninterrupted online services.

Step 7 Click **OK**.

Step 8 After you change some parameters, manually reboot the instance for the changes to take effect. For details, see [Rebooting a GaussDB Instance](#).

The failover priority cannot be changed when the DB instance is in the **Rebooting** state.

----End

8.11.3 Performing a Primary/Standby Switchover

Scenarios

GaussDB supports primary/standby DN switchover in a shard of an instance when the instance is available. You can promote a standby DN to the primary DN in a shard.

Constraints

- This operation cannot be performed when the node status is abnormal.
- Only one standby node can be specified as the primary node in a shard.
- Single-node instances do not support primary/standby DN shard switchovers.
- During a primary/standby switchover, the following operations cannot be performed:
 - Rebooting a DB instance
 - Switching AZs
 - Changing CPU and memory specifications of an instance
 - Repairing a node
 - Replacing a node

- Adding nodes
- Backing up and restoring an instance

Precautions

- Services may be interrupted for several seconds or minutes during the switchover. You are advised to perform this operation during off-peak hours.
- Switchovers will not change private IP addresses of an instance.

Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 Click the target instance name to go to the **Basic Information** page.

Step 5 In the **Node List** area, click **Switch Primary and Standby DNs**.

Step 6 Select an AZ to view the DN shard of the primary DN in the selected AZ. Select the standby DN to be promoted to primary, enter **rearrange**, and click **OK**.

- If there is no primary DN in the selected AZ, shard information is not displayed.
- Services may be interrupted for several seconds or minutes during the switchover. You are advised to perform this operation during off-peak hours.
- Primary/Standby switchovers can be performed on a maximum of 30 shards at a time.

Figure 8-44 Switching primary and standby DN

×

Switch Primary and Standby DNs

⚠ Services may be interrupted for several seconds or minutes during the switchover. You are advised to perform this operation during off-peak hours.

AZ of Primary DN

All AZs

⌵

⌂

Group ID	Compon...	Node
60011	dn_6001	

All AZs

az1

Group ID	Compo...	Node	Role	Status	AZ
60011	dn_6003		Standby	Av...	az1
60011	dn_6002		Standby	Av...	az1

To confirm the switchover, enter "rearrange" below. [One-key input](#)

Enter rearrange

Cancel

OK

-----End

9 Instance Upgrade

9.1 Overview

You can manually upgrade the GaussDB kernel version of a single instance or multiple instances in batches using in-place upgrade, gray upgrade, or hot patch update to improve performance, add new functions, and fix bugs.

Checking the Current Kernel Version

To check the version of an instance, go to the **Basic Information** page of the instance and check the value of **DB Engine Version** in the **Configuration** area.

Figure 9-1 Basic information



Configuration	
DB Engine Version	Kernel Engine Version
GaussDB 505.1.0 Upgrade Instance	505.1.0

You can learn [details about kernel versions](#) and determine the target version to upgrade.

Upgrade Methods

The following table describes the upgrade methods supported by GaussDB.

Table 9-1 Upgrade methods

Upgrade Method	Action	Type	Scenario	Rollback Method	Impact on Services	Suggestions
Hot patch	Auto-commit	Online upgrade	Fix product issues.	- Automatic - Manual	No service is interrupted during the upgrade.	None
In-place upgrade	N/A	Offline upgrade	- Add new functions. - Fix product issues.	Automatic	Services are interrupted for about 30 minutes during the in-place upgrade.	Stop all workloads during the upgrade.

Upgrade Method	Action	Type	Scenario	Rollback Method	Impact on Services	Suggestions
Gray upgrade	Auto-commit	Online upgrade	• Add new functions. • Fix product issues.	Automatic	Services are interrupted for about 10s during the upgrade of primary DN and during the upgrade of CNs. During upgrade commit, primary/standby distribution balancing may be performed. Services may be interrupted for different periods of time based on factors such as data volume.	Add the service reconnection mechanism. It is recommended that the retry interval be 1s. During low-pressure periods (less than 3,000 TPS + 4,000 QPS for each shard), the total retry duration is 25s. During high-pressure periods (less than 6,000 TPS + 10,000 QPS for each shard), the total retry duration is 100s. The upgrade is not recommended when the pressure is out of the acceptable range.
	Rolling upgrade	Online upgrade	• Add new functions. • Fix product issues.	• Automatic • Manual	If the AZ to be upgraded contains primary DNs, services will be interrupted for about 10s during the upgrade of each primary DN. If the AZ to be upgraded contains CNs, services will be interrupted for about 10s during the upgrade of each CN.	

9.2 Hot Patch Update

Scenarios

You can install a hot patch for your GaussDB instance to rectify product issues. A hot patch can be loaded without interrupting services and can be used to resolve some emergent database kernel problems online without affecting services. Hot patch update supports manual rollback.

Precautions

- During the update, hot patch packages will be downloaded and decompressed, which occupies certain disk space. It is recommended that the disk usage on the DN be less than or equal to the disk usage threshold minus 10%.

NOTE

To check the current DN disk usage, go to the metric monitoring page on the management console.

To obtain the disk usage threshold, [submit a service ticket](#) to contact technical support.

- Version upgrade is unavailable if instance nodes are in an abnormal state.
- If a hot patch conflicts with the backup, the differential backup and full backup of the instance will be stopped during hot patch installation.
- During an upgrade or rollback, the following operations cannot be performed: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting instances, and deleting instances.
- You are advised to perform an upgrade during off-peak hours because there are more idle CPU, disk, and memory resources.
- Hot patch update is available only when there is a hot patch for installation. If no hot patch is available, hot patch update is not displayed.
- Hot patch update and rollback can be performed in batches for different patch versions of a single instance. During the update, hot patches are installed in ascending order of version numbers. During the rollback, hot patches are rolled back in descending order of version numbers.
- If the upgrade fails, the system automatically rolls back the instance to the source version. You can [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you upgrade the instance if necessary.
- After the upgrade is complete, you can manually roll back the upgrade.
- A maximum of 30 instances can be selected at a time for batch upgrade.
- GaussDB supports automatic hot patch update after instance creation or cold patch update. To enable automatic hot patch update, [submit a service ticket](#). Note that automatic hot patch update is not supported for new instances created using backups of existing instances.

Step 1: Perform a Pre-upgrade Check

Before an upgrade, check the instance status and whether monitoring metrics such as the CPU usage, memory usage, and disk usage of the instance are normal.

1. Check instance status.
 - a. [Log in to the management console](#).
 - b. Click  in the upper left corner and select a region and project.
 - c. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
 - d. On the **Instances** page, check whether **Status** of the target instance is **Available**.

Figure 9-2 Instance status

☐ Name/ID ↕	Status	Description
☐ gau 5d1 [redacted] 2ab18ae51486in14	 Available	--

If the instance is in an abnormal state, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you analyze and handle the abnormal instance.

2. Check monitoring metrics.
 - a. Click  in the upper left corner of the page, and choose **Management & Governance > Cloud Eye**.
 - b. In the navigation pane, choose **Cloud Service Monitoring > GaussDB**.
 - c. On the **Cloud Service Monitoring** page, click the target instance to go to the metric monitoring page.
 - On the **DB Instance** tab, view the value of **Instance Disk Usage** to check whether the disk usage is insufficient.
 - On the **Node** tab, view the value of **CPU Usage** to check whether the CPU usage remains high for a long time.
 - On the **Node** tab, view the value of **Memory Usage** to check whether the memory usage increases sharply.

If any metric is abnormal, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you analyze and handle the abnormal metrics.

Step 2: Perform the Upgrade

[Method 1: upgrading a single instance]

1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.

2. On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Upgrade**.
3. In the **Upgrade Instance** dialog box, select **Hot patch** for **Upgrade Method**, enter **YES**, and click **OK**.

NOTE

All available patch versions are displayed in the **Target Version** area. If multiple patches are to be installed, they will be installed in ascending order of version numbers after the upgrade task is submitted.

Figure 9-3 Upgrading an instance

Upgrade Instance

After the instances are upgraded, upgrade your application's driver to the [driver version](#) that matches the target instance version.

ID

DB Instance Name

Upgrade Method: **Hot patch** Gray upgrade In-place

Action: **Auto-commit**

Target Version: 08.0.7 Universal

Multiple patches will be installed from minor to major versions.

To confirm the upgrade, enter "YES" below. [Auto Enter](#)

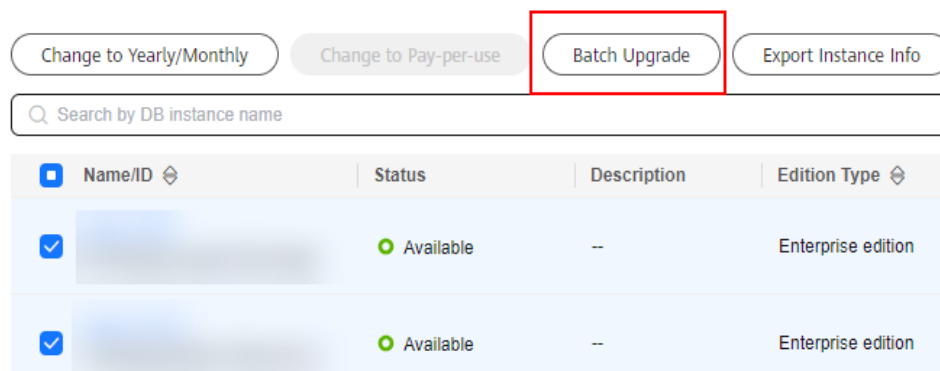
YES

Cancel OK

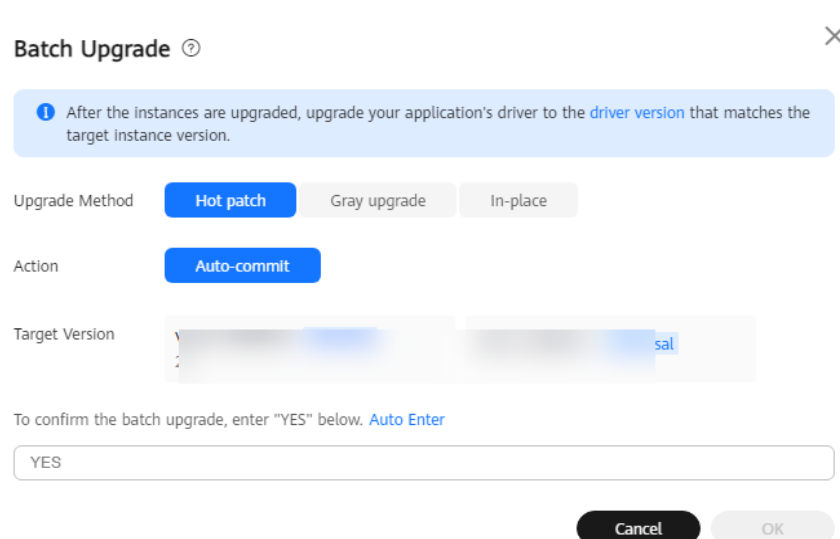
4. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Available**. In the **Configuration** area on the **Basic Information** page of the instance, check that the target version is displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields.
- Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target version is displayed in the two columns on the **Instances** page.

[Method 2: upgrading instances in batches]

1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
2. On the **Instances** page, select the target instances and click **Batch Upgrade**.

Figure 9-4 Batch upgrade

3. In the **Batch Upgrade** dialog box, select **Hot patch** for **Upgrade Method**.
4. Enter **YES** and click **OK**.

Figure 9-5 Auto-commit of a hot patch update**NOTE**

All available patch versions are displayed in the **Target Version** area. If multiple patches are to be installed, they will be installed in ascending order of version numbers after the upgrade task is submitted.

5. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Available**. In the **Configuration** area on the **Basic Information** page of the instance, check that the target version is displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields.

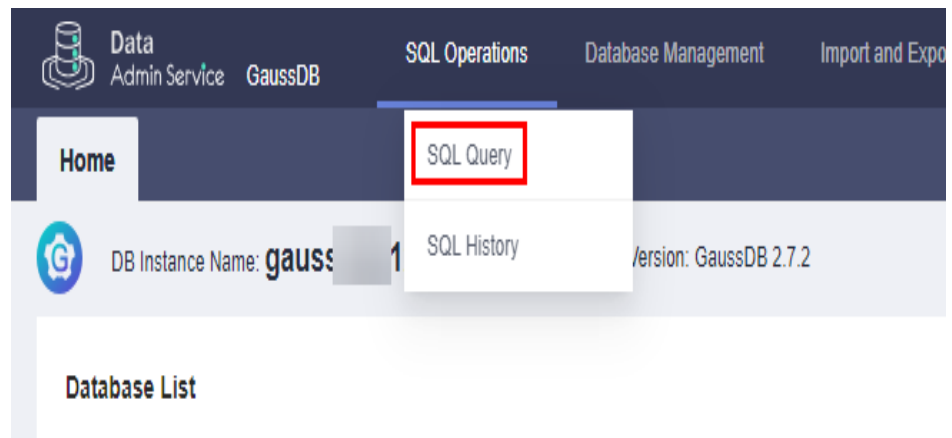
Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target version is displayed in the two columns on the **Instances** page.

Step 3: Verify the Upgrade

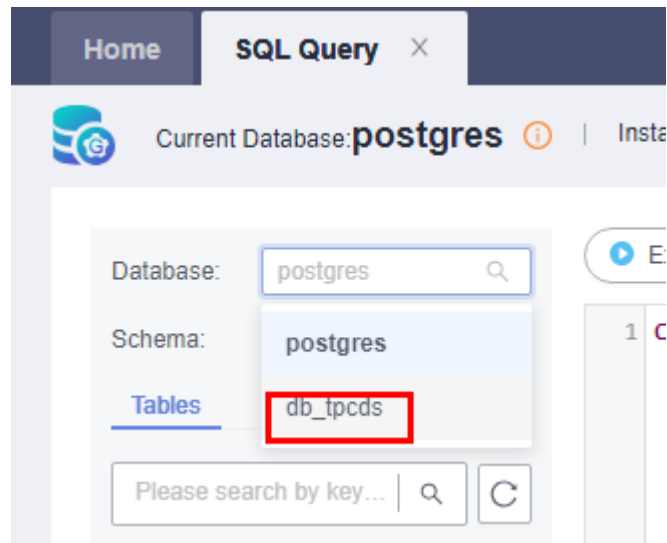
After the upgrade is complete, check the instance status, backup creation status, and instance connectivity, and whether you can add, delete, update, and query data in the instance.

1. On the **Instances** page, check whether **Status** of the target instance is **Available**.
2. On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check that the target versions are displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields in the **Configuration** area. Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target versions are displayed in the two columns on the **Instances** page.
3. Check that the instance is properly connected and you can add, delete, update, and query data in the instance.
 - a. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
 - b. Go to the **SQL Query** page.

Figure 9-6 SQL query



- c. Create a database.
CREATE DATABASE *database_name*;
In this example, run the following command to create a database named **db_tpcds**:
CREATE DATABASE db_tpcds;
Switch to the newly created database in the upper left corner.

Figure 9-7 Switching to the new database

d. Create a table and add, delete, update, and query data in the table.

i. Create a schema.

```
CREATE SCHEMA myschema;
```

ii. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.

```
CREATE TABLE myschema.mytable (firstcol int);
```

iii. Insert data into the table.

```
INSERT INTO myschema.mytable VALUES (100);
```

iv. View data in the table.

```
SELECT * FROM myschema.mytable;
```

firstcol
1

v. Update data in the table.

```
UPDATE myschema.mytable SET firstcol = 200;
```

vi. View the data in the table again.

```
SELECT * FROM myschema.mytable;
```

firstcol
1

vii. Drop the table.

```
DROP TABLE myschema.mytable;
```

Rollback Operations

If a rollback is required after the upgrade, perform the following operations to roll back an instance to the source version.

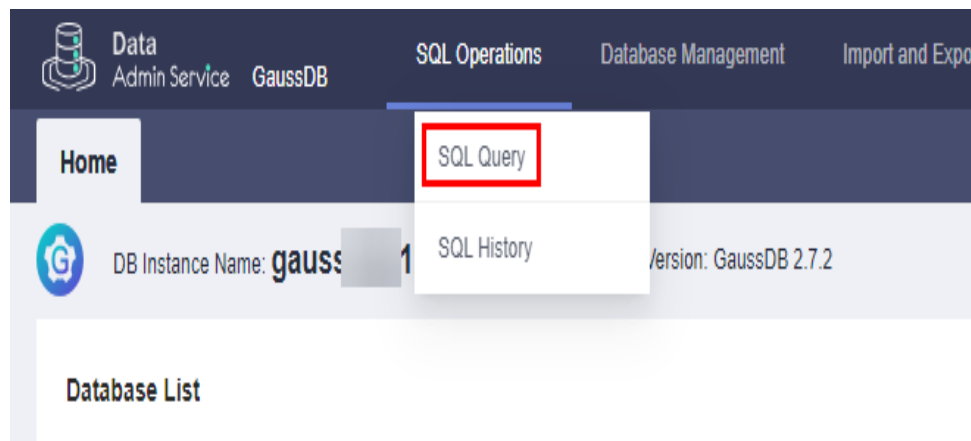
NOTE

After the upgrade is rolled back, you can perform the upgrade again. If the problem persists, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you upgrade the instance if necessary.

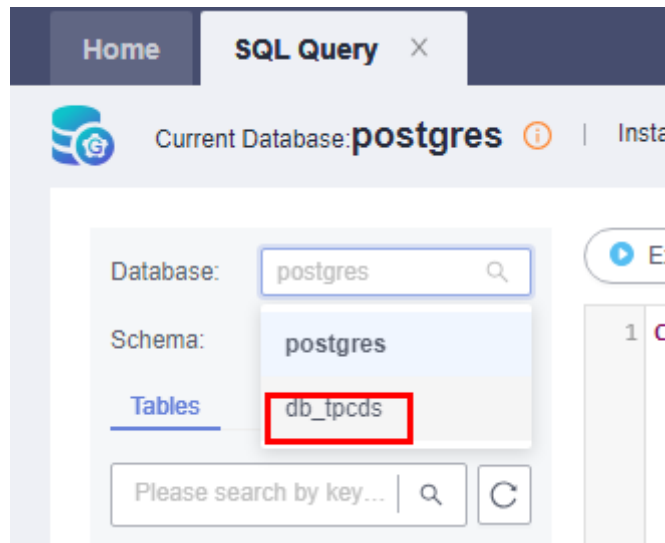
[Method 1: Rolling Back a Single Instance]

- Step 1** In the **Upgrade Instance** dialog box, select **Hot patch** for **Upgrade Method**.
- Step 2** Select **Rollback** for **Action**.
- Step 3** Select the target version, enter **YES**, and click **OK**.
- Step 4** On the **Instances** page, check the rollback status. After the rollback is complete, the instance status changes to **Available**.
- Step 5** On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check that the target versions are not displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields in the **Configuration** area. Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target version is not displayed in the two columns on the **Instances** page.
- Step 6** Check that the instance is properly connected and you can add, delete, update, and query data in the instance.
1. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
 2. Go to the **SQL Query** page.

Figure 9-8 SQL query



3. Create a database.
CREATE DATABASE *database_name*;
In this example, run the following command to create a database named **db_tpcds**:
CREATE DATABASE db_tpcds;
Switch to the newly created database in the upper left corner.

Figure 9-9 Switching to the new database

4. Create a table and add, delete, update, and query data in the table.
 - a. Create a schema.
`CREATE SCHEMA myschema;`
 - b. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.
`CREATE TABLE myschema.mytable (firstcol int);`
 - c. Insert data into the table.
`INSERT INTO myschema.mytable values (100);`
 - d. View the data in the table.
`SELECT * FROM myschema.mytable;`

firstcol
100
 - e. Update data in the table.
`UPDATE myschema.mytable SET firstcol = 200;`
 - f. View the data in the table again.
`SELECT * FROM myschema.mytable;`

firstcol
200
 - g. Drop the table.
`DROP TABLE myschema.mytable;`

----End

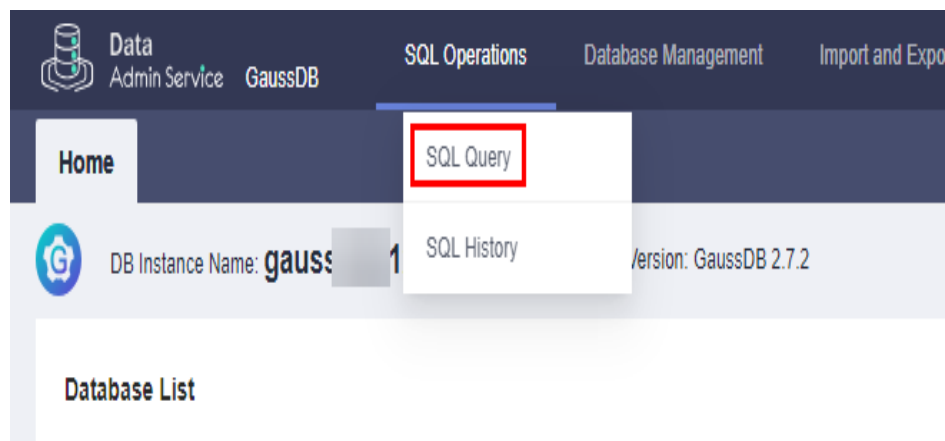
[Method 2: Rolling Back Instances in Batches]

Step 1 On the **Instances** page, select the target instances and click **Batch Upgrade**.

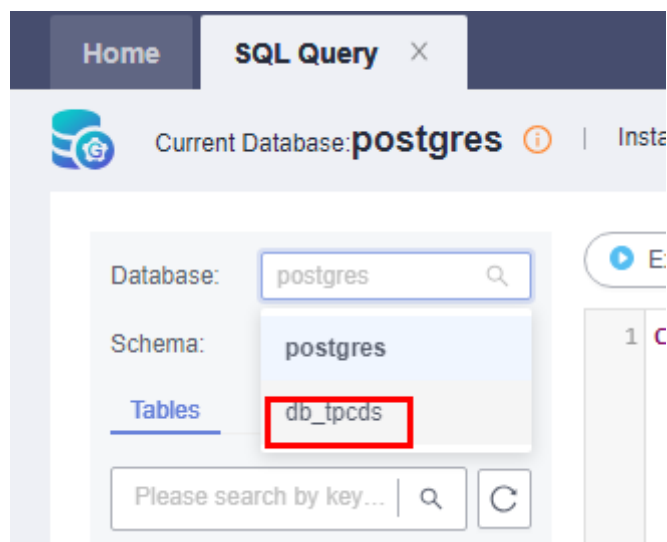
Step 2 In the **Batch Upgrade** dialog box, select **Hot patch** for **Upgrade Method** and **Rollback** for **Action**, select a target version, enter **YES**, and click **OK**.

- Step 3** On the **Instances** page, check the rollback status. After the rollback is complete, the instance status changes to **Available**.
- Step 4** On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check that the target versions are not displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields in the **Configuration** area. Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target versions are not displayed in the two columns on the **Instances** page.
- Step 5** Check that the instance is properly connected and you can add, delete, update, and query data in the instance.
1. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
 2. Go to the **SQL Query** page.

Figure 9-10 SQL query



3. Create a database.
CREATE DATABASE *database_name*;
In this example, run the following command to create a database named **db_tpcds**:
CREATE DATABASE db_tpcds;
Switch to the newly created database in the upper left corner.

Figure 9-11 Switching to the new database

4. Create a table and add, delete, update, and query data in the table.
 - a. Create a schema.
`CREATE SCHEMA myschema;`
 - b. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.
`CREATE TABLE myschema.mytable (firstcol int);`
 - c. Insert data into the table.
`INSERT INTO myschema.mytable VALUES (100);`
 - d. View data in the table.
`SELECT * FROM myschema.mytable;`

	firstcol
1	100
 - e. Update data in the table.
`UPDATE myschema.mytable SET firstcol = 200;`
 - f. View the data in the table again.
`SELECT * FROM myschema.mytable;`

	firstcol
1	200
 - g. Drop the table.
`DROP TABLE myschema.mytable;`

----End

9.3 In-place Upgrade

Scenarios

You can use in-place upgrade to upgrade your instance if a version upgrade is required for new functions or issue rectification. During an in-place upgrade, all nodes are upgraded at the same time, and all services are interrupted.

Precautions

- In-place upgrade is only available for single-replica instances.
- The DN disk usage cannot be greater than the configured disk usage threshold minus 10%.

NOTE

To check the current DN disk usage, go to the metric monitoring page on the management console.

To obtain the disk usage threshold, [submit a service ticket](#) to contact technical support.

- Version upgrade is unavailable if instance nodes are in an abnormal state.
- Before the upgrade, ensure that there is a public schema in each database. Otherwise, the upgrade will fail.
- During an upgrade, the following operations cannot be performed: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting instances, and deleting instances.
- If this method is used for a major version upgrade, log archiving will be disabled before the upgrade, and you cannot use archive logs for Point-In-Time Recovery (PITR), which may result in data loss.

NOTE

Example major version upgrade: from V2.0-1.x to V2.0-2.x or from V2.0-2.x to V2.0-2.y

- If the upgrade fails, the system automatically rolls back the instance to the source version. You can [submit a service ticket](#) to contact Huawei Cloud customer service, and the Huawei Cloud engineers will help you upgrade the instance if necessary.
- Services are interrupted for about 30 minutes during the in-place upgrade.
- After the upgrade is complete, an automated backup task will be triggered and log archiving will be enabled. However, for a single-replica instance upgraded to V2.0-3.0 or later from earlier versions, automated backup is disabled by default and will not be triggered. An automated backup task will also not be triggered in the case of minor version upgrades.

NOTE

Example minor version upgrade: from V2.0-1.a.x to V2.0-1.a.y or from V2.0-2.a.x to V2.0-2.a.y

- In-place upgrade does not require manual rollback.
- A maximum of 30 instances can be selected at a time for batch upgrade.

Step 1: Perform a Pre-upgrade Check

Before an upgrade, check the instance status and whether monitoring metrics such as the CPU usage, memory usage, and disk usage of the instance are normal.

1. Check instance status.
 - a. [Log in to the management console](#).
 - b. Click  in the upper left corner and select a region and project.
 - c. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
 - d. On the **Instances** page, check whether **Status** of the target instance is **Available**.

Figure 9-12 Instance status

☐ Name/ID ↕	Status	Description
☐ gau 5d1 [redacted] 2ab18ae51486in14	 Available	--

If the instance is in an abnormal state, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you analyze and handle the abnormal instance.

2. Check monitoring metrics.
 - a. Click  in the upper left corner of the page, and choose **Management & Governance > Cloud Eye**.
 - b. In the navigation pane, choose **Cloud Service Monitoring > GaussDB**.
 - c. On the **Cloud Service Monitoring** page, click the target instance to go to the metric monitoring page.
 - On the **DB Instance** tab, view the value of **Instance Disk Usage** to check whether the disk usage is insufficient.
 - On the **Node** tab, view the value of **CPU Usage** to check whether the CPU usage remains high for a long time.
 - On the **Node** tab, view the value of **Memory Usage** to check whether the memory usage increases sharply.

If any metric is abnormal, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you analyze and handle the abnormal metrics.

Step 2: Perform the Upgrade

[Method 1: upgrading a single instance]

1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.

2. On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Upgrade**.

Figure 9-13 Instances



NameID	Status	Description	DB Inst...	DB Engine Versi...	Target Versi...	Upgraded H...	Billing Mode	Private IP A...	Created	Database Port	Storage	Operation
	Available	--			--	--	Pay-per-use Created on A...	192.168...	Apr 18, 2024...	8000	Ultra-High	Log In View Metrics More
	Available	--			--	--	Pay-per-use Created on A...	19.16.11...	Apr 18, 2024...	8000		Change to Hourly/Monthly
	Creating	--			--	--	Pay-per-use		Apr 18, 2024...	8000		Scale Storage Space
	Available	--			--	--	Pay-per-use Created on A...	19.16.14...	Apr 18, 2024...	8000		Change Instance Specifications
	Available	--			--	--	Pay-per-use Created on A...	19.16.13...	Apr 18, 2024...	8000		Sync Data to Single-Replica Instance
	Available	--			--	--	Pay-per-use Created on A...					Create Backup
	Available	--			--	--	Pay-per-use Created on A...					Reset Password
	Available	--			--	--	Pay-per-use Created on A...					Reboot
	Available	--			--	--	Pay-per-use Created on A...					Upgrade
	Available	--			--	--	Pay-per-use Created on A...					Delete

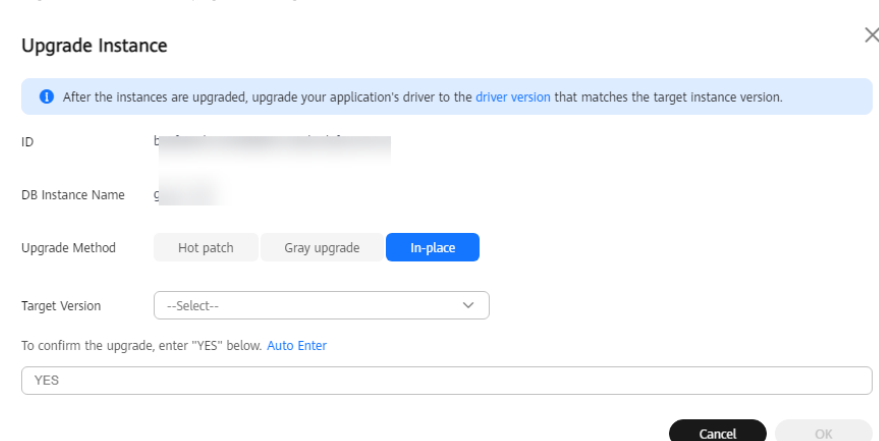
Alternatively, click the target instance name to go to the **Basic Information** page. In the **Configuration** area, click **Upgrade Instance** in the **DB Engine Version** field.

Figure 9-14 Basic information



3. In the **Upgrade Instance** dialog box, select **In-place** for **Upgrade Method**, select the target version, enter **YES**, and click **OK**.

Figure 9-15 Upgrading an instance



Upgrade Instance

After the instances are upgraded, upgrade your application's driver to the driver version that matches the target instance version.

ID b...

DB Instance Name c...

Upgrade Method Hot patch Gray upgrade In-place

Target Version --Select--

To confirm the upgrade, enter "YES" below. Auto Enter

YES

Cancel OK

4. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Available**. In the **Configuration** area on the **Basic Information** page of the instance, check that the target version is displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields.

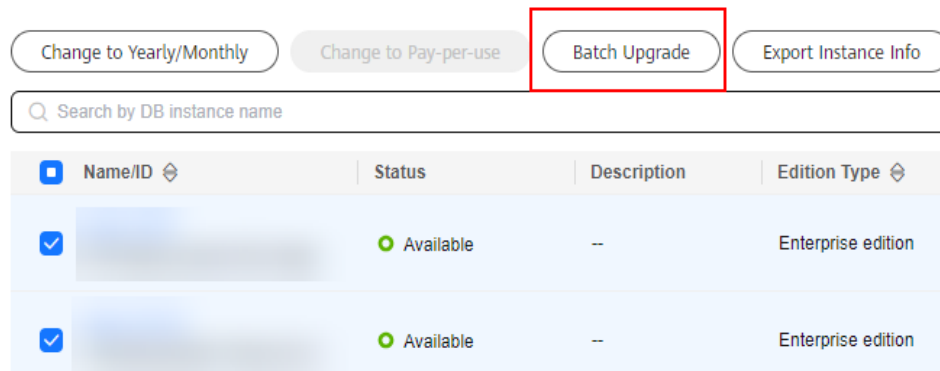
Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and

Upgraded Kernel Hot Patch Version columns. Then, check that the target version is displayed in the two columns on the **Instances** page.

[Method 2: upgrading instances in batches]

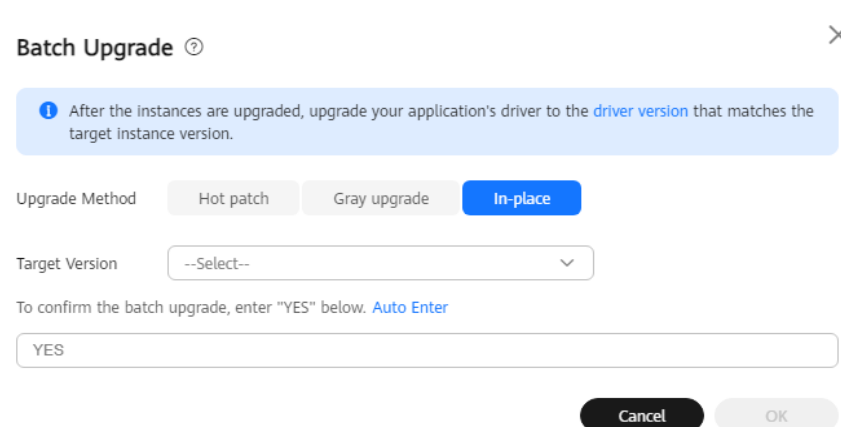
1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
2. On the **Instances** page, select the target instances and click **Batch Upgrade**.

Figure 9-16 Batch upgrade



3. In the **Batch Upgrade** dialog box, select **In-place** for **Upgrade Method**, select the target version, enter **YES**, and click **OK**.

Figure 9-17 In-place upgrade



4. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Available**. In the **Configuration** area on the **Basic Information** page of the instance, check that the target version is displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields.Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and

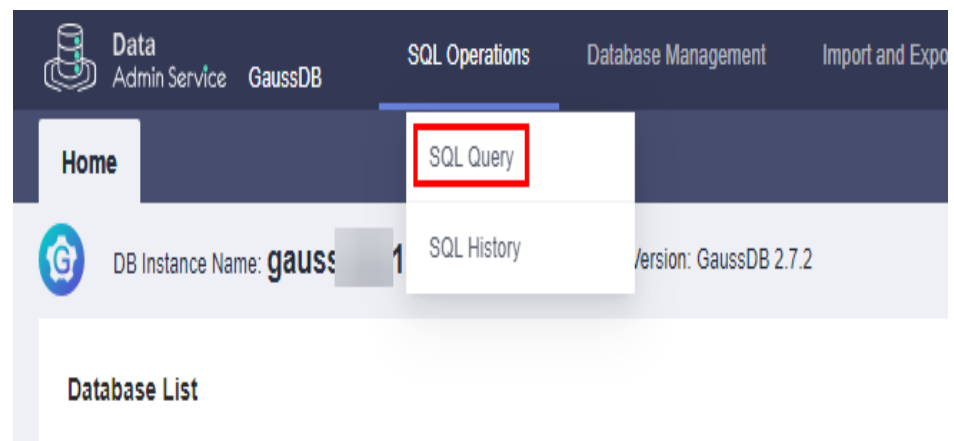
Upgraded Kernel Hot Patch Version columns. Then, check that the target version is displayed in the two columns on the **Instances** page.

Step 3: Verify the Upgrade

After the upgrade is complete, check the instance status, backup creation status, and instance connectivity, and whether you can add, delete, update, and query data in the instance.

1. On the **Instances** page, check whether **Status** of the target instance is **Available**.
2. On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check whether the value of **DB Engine Version** in the **Configuration** area is the target version.
3. Check that the automated backup triggered after the upgrade is successfully created.
 - a. On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
 - b. In the navigation pane, choose **Backups**. Check that a backup has been created and the backup status is **Completed**.
4. Check that the instance is properly connected and you can add, delete, update, and query data in the instance.
 - a. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
 - b. Go to the **SQL Query** page.

Figure 9-18 SQL query



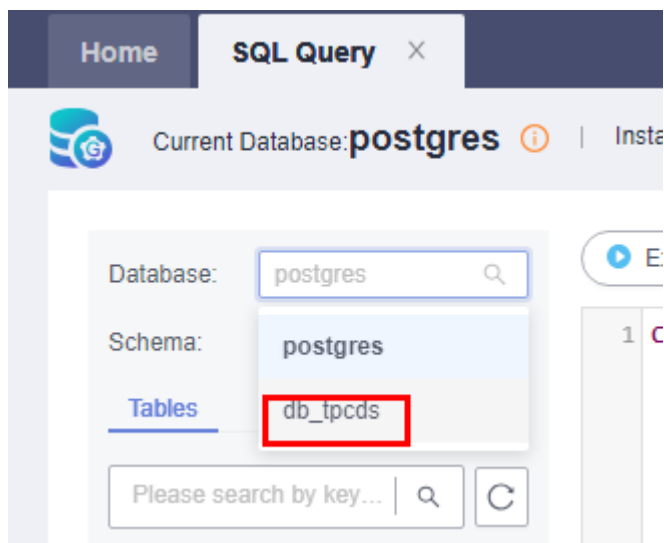
- c. Create a database.

CREATE DATABASE *database_name*;

In this example, run the following command to create a database named **db_tpcds**:

CREATE DATABASE db_tpcds;

Switch to the newly created database in the upper left corner.

Figure 9-19 Switching to the new database

d. Create a table and add, delete, update, and query data in the table.

i. Create a schema.

```
CREATE SCHEMA myschema;
```

ii. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.

```
CREATE TABLE myschema.mytable (firstcol int);
```

iii. Insert data into the table.

```
INSERT INTO myschema.mytable VALUES (100);
```

iv. View data in the table.

```
SELECT * FROM myschema.mytable;
```

firstcol
1

v. Update data in the table.

```
UPDATE myschema.mytable SET firstcol = 200;
```

vi. View the data in the table again.

```
SELECT * FROM myschema.mytable;
```

firstcol
1

vii. Drop the table.

```
DROP TABLE myschema.mytable;
```

9.4 Gray Upgrade

Scenarios

You can use gray upgrade to upgrade your GaussDB instance if a version upgrade is required for new functions or issue rectification. You can either select auto-commit after the upgrade or perform a rolling upgrade.

- In the auto-commit mode, all standby DN nodes are upgraded first and then primary DN nodes and CN nodes in sequence. After the upgrade is complete, the upgrade is automatically committed.
- The rolling upgrade mode is also known as the upgrade observation mode. In this mode, the system enters the observation state after the upgrade is complete. During this period, you can observe the service status and either commit or roll back the upgrade based on service status.
 - Distributed instances are upgraded by shard. For details, see [Upgrading a Distributed Instance](#).
 - Centralized instances are upgraded by AZ. For details, see [Upgrading a Centralized Instance](#).

Procedure

Step	Description
Step 1: Perform a Pre-upgrade Check	Before an upgrade, check the instance status and whether monitoring metrics such as the CPU usage, memory usage, and disk usage of the instance are normal.
Step 2: Perform the Upgrade	Select either auto-commit after the upgrade or perform a rolling upgrade. You can upgrade a single instance or multiple instances in batches as required.
Step 3: Verify the Upgrade	After the upgrade is complete, check the instance status, backup creation status, and instance connectivity, and whether you can add, delete, update, and query data in the instance.

Precautions

- The DN disk usage cannot be greater than the configured disk usage threshold minus 10%.

NOTE

To check the current DN disk usage, go to the metric monitoring page on the management console.

To obtain the disk usage threshold, [submit a service ticket](#) to contact technical support.

- Version upgrade is unavailable if instance nodes are in an abnormal state.
- Before the upgrade, ensure that there is a public schema in each database. Otherwise, the upgrade will fail.
- The rolling upgrade mode supports manual rollback, but the auto-commit mode does not support manual rollback.
- During the upgrade, automatic reduction of replicas is not supported.
- During an upgrade or rollback, the following operations cannot be performed: scaling up storage, changing specifications, backing up data, resetting passwords, rebooting instances, and deleting instances.
- You are advised to perform an upgrade during off-peak hours because there are more idle CPU, disk, and memory resources.

- If upgrade auto-commit is used for a major version upgrade, log archiving will be disabled before the upgrade, and you cannot use archive logs for PITR, which may result in data loss.
- If rolling upgrade is used for a major version upgrade, full backup cannot be triggered during the upgrade, and differential backup may fail. Manual full backups cannot be created until the upgrade operations in all AZs are complete during the rolling upgrade and observation period. Archive logs are still generated before the upgrade is committed, and you can use archive logs for PITR to prevent data loss. In the upgrade commit phase, log archiving is disabled.
- If the upgrade fails, the system automatically rolls back the instance to the source version. You can [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you upgrade the instance if necessary.
- Services are interrupted for about 10 seconds during the upgrade of primary DN and during the upgrade of CNs.
- After the upgrade is complete, an automated backup task will be triggered and log archiving will be enabled. However, an automated backup task will not be triggered in the case of minor version upgrades. Log archiving is only available for instances of versions later than V2.0-2.2.

NOTE

Example minor version upgrade: from V2.0-1.a.x to V2.0-1.a.y or from V2.0-2.a.x to V2.0-2.a.y

Example major version upgrade: from V2.0-1.x to V2.0-2.x or from V2.0-2.x to V2.0-2.y

Step 1: Perform a Pre-upgrade Check

Before an upgrade, check the instance status and whether monitoring metrics such as the CPU usage, memory usage, and disk usage of the instance are normal.

1. Check instance status.
 - a. [Log in to the management console](#).
 - b. Click  in the upper left corner and select a region and project.
 - c. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
 - d. On the **Instances** page, check whether **Status** of the target instance is **Available**.

Figure 9-20 Instance status

☐ Name/ID	Status	Description
☐ gau5d1-2ab18ae51486in14	 Available	--

If the instance is in an abnormal state, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you analyze and handle the abnormal instance.

2. Check monitoring metrics.
 - a. Click  in the upper left corner of the page, and choose **Management & Governance > Cloud Eye**.
 - b. In the navigation pane, choose **Cloud Service Monitoring > GaussDB**.
 - c. On the **Cloud Service Monitoring** page, click the target instance to go to the metric monitoring page.
 - On the **DB Instance** tab, view the value of **Instance Disk Usage** to check whether the disk usage is insufficient.
 - On the **Node** tab, view the value of **CPU Usage** to check whether the CPU usage remains high for a long time.
 - On the **Node** tab, view the value of **Memory Usage** to check whether the memory usage increases sharply.

If any metric is abnormal, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you analyze and handle the abnormal metrics.

Step 2: Perform the Upgrade

You can select auto-commit after the upgrade or perform a rolling upgrade for gray upgrade as required.

Upgrade Auto-commit

[Method 1: upgrading a single instance]

1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
2. On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Upgrade**.
3. In the **Upgrade Instance** dialog box, select **Gray upgrade** for **Upgrade Method**.
4. Select **Auto-commit** for **Action**.

Figure 9-21 Upgrading an instance

Upgrade Instance

After the instances are upgraded, upgrade your application's driver to the [driver version](#) that matches the target instance version.

ID

DB Instance Name

Upgrade Method: Hot patch, **Gray upgrade**, In-place

Action: **Auto-commit**, Rolling upgrade

Target Version: --Select--

To confirm the upgrade, enter "YES" below. [Auto Enter](#)

YES

Cancel OK

5. Select the target version, enter **YES**, and click **OK**.
6. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Available**. In the **Configuration** area on the **Basic Information** page of the instance, check that the target version is displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields.

Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target version is displayed in the two columns on the **Instances** page.

[Method 2: upgrading instances in batches]

1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
2. On the **Instances** page, select the target instances and click **Batch Upgrade**.

Figure 9-22 Batch upgrade

Change to Yearly/Monthly Change to Pay-per-use **Batch Upgrade** Export Instance Info

Search by DB instance name

Name/ID	Status	Description	Edition Type
☒	Available	--	Enterprise edition
☒	Available	--	Enterprise edition

3. In the **Batch Upgrade** dialog box, select **Gray upgrade** for **Upgrade Method**.

4. Select **Auto-commit** for **Action**.
5. Select the target version, enter **YES**, and click **OK**.

Figure 9-23 Auto-commit of a gray upgrade

Batch Upgrade ?

After the instances are upgraded, upgrade your application's driver to the driver version that matches the target instance version.

Upgrade Method: Hot patch, **Gray upgrade**, In-place

Action: **Auto-commit**, Rolling upgrade

Target Version: --Select--

To confirm the batch upgrade, enter "YES" below. Auto Enter

YES

Cancel OK

6. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Available**. In the **Configuration** area on the **Basic Information** page of the instance, check that the target version is displayed in the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** fields.Alternatively, on the **Instances** page, click  in the upper-right corner of the instance list and select the **Upgraded Hot Patch Version** and **Upgraded Kernel Hot Patch Version** columns. Then, check that the target version is displayed in the two columns on the **Instances** page.

Rolling Upgrade

[Method 1: upgrading a single instance]

- Upgrading a distributed instance
 - a. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
 - b. On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Upgrade**.Alternatively, click the target instance name to go to the **Basic Information** page. In the **Configuration** area, click **Upgrade Instance** in the **DB Engine Version** field.

Figure 9-24 Basic information**Configuration**

DB Engine Version

GaussDB  Upgrade Instance

Kernel Engine Version

505.1.0

- c. In the **Upgrade Instance** dialog box, select **Gray upgrade** for **Upgrade Method**.
- d. Select **Rolling upgrade** for **Action**.
- e. Set **Shards to Upgrade**, select a target version, enter **YES**, and click **OK**.
- f. Check the upgrade result on the **Instances** page.

- i. During the upgrade, the instance status is **Upgrading version**.
- ii. After the upgrade is complete, the instance status changes to **Observing version upgrade**.

- g. Check that all shards are upgraded and services are running properly before committing the upgrade.

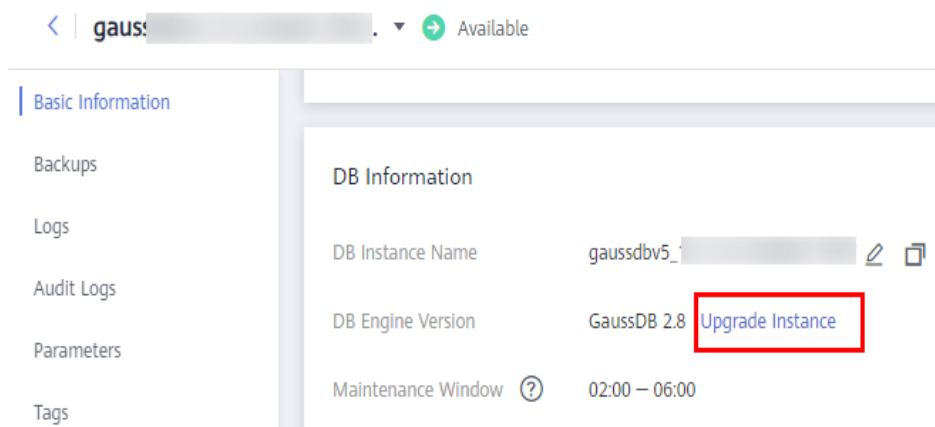
In the **Upgrade Instance** dialog box, select **Commit** for **Action**, select a target version, enter **YES**, and click **OK**.

If you choose to upgrade shards one by one, repeat **b** to **f** until all shards are upgraded, and then commit the upgrade.

- Upgrading a centralized instance

- a. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- b. On the **Instances** page, click **More** in the **Operation** column of the target instance and choose **Upgrade**.

Alternatively, click the target instance name to go to the **Basic Information** page. In the **Configuration** area, click **Upgrade Instance** in the **DB Engine Version** field.

Figure 9-25 Basic information

- c. In the **Upgrade Instance** dialog box, select **Gray upgrade** for **Upgrade Method**.

- d. Select **Rolling upgrade** for **Action**.
- e. Set **AZs to Upgrade**, select a target version, enter **YES**, and click **OK**.

 NOTE

You can upgrade a single AZ or multiple AZs at a time as needed.

- f. Check the upgrade result on the **Instances** page.
 - i. During the upgrade, the instance status is **Upgrading version**.
 - ii. After the upgrade is complete, the instance status changes to **Observing version upgrade**.
- g. Check that all AZs are upgraded and services are running properly before committing the upgrade.

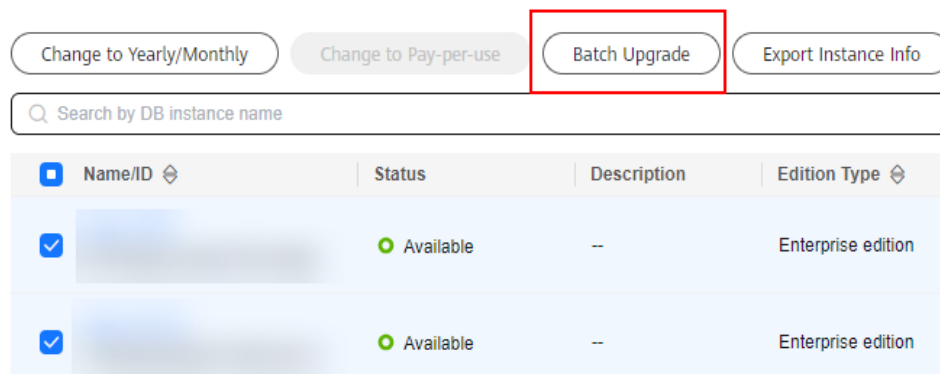
In the **Upgrade Instance** dialog box, select **Commit** for **Action**, select a target version, enter **YES**, and click **OK**.

If you choose to upgrade AZs one by one, repeat **b** to **f** until all AZs are upgraded, and then commit the upgrade.

[Method 2: upgrading instances in batches]

1. Click  in the upper left corner of the page and choose **Databases > GaussDB**.
2. On the **Instances** page, select the target instances and click **Batch Upgrade**.

Figure 9-26 Batch upgrade



3. In the **Batch Upgrade** dialog box, select **Gray upgrade** for **Upgrade Method**.
4. Select **Rolling upgrade** for **Action**.
5. Select the target version, enter **YES**, and click **OK**.

Figure 9-27 Rolling upgrade of a gray upgrade

The image shows a 'Batch Upgrade' dialog box with a close button (X) in the top right corner. At the top, there is a blue information banner with a question mark icon and text: 'After the instances are upgraded, upgrade your application's driver to the driver version that matches the target instance version.' Below this, there are two rows of buttons. The first row, labeled 'Upgrade Method', contains three buttons: 'Hot patch', 'Gray upgrade' (which is highlighted in blue), and 'In-place'. The second row, labeled 'Action', contains two buttons: 'Auto-commit' and 'Rolling upgrade' (which is highlighted in blue). Below these buttons is a 'Target Version' dropdown menu currently showing '--Select--'. Underneath the dropdown is a text prompt: 'To confirm the batch upgrade, enter "YES" below. Auto Enter'. Below this prompt is a text input field containing the word 'YES'. At the bottom right of the dialog are two buttons: 'Cancel' and 'OK'.

NOTE

During a batch upgrade in rolling upgrade mode, all AZs or shards of the selected instances are upgraded by default.

6. Check the upgrade result on the **Instances** page.
 - During the upgrade, the instance status is **Upgrading version**.
 - After the upgrade is complete, the instance status changes to **Observing version upgrade**.
7. Check that all shards or AZs are upgraded and services are running properly before committing the upgrade.

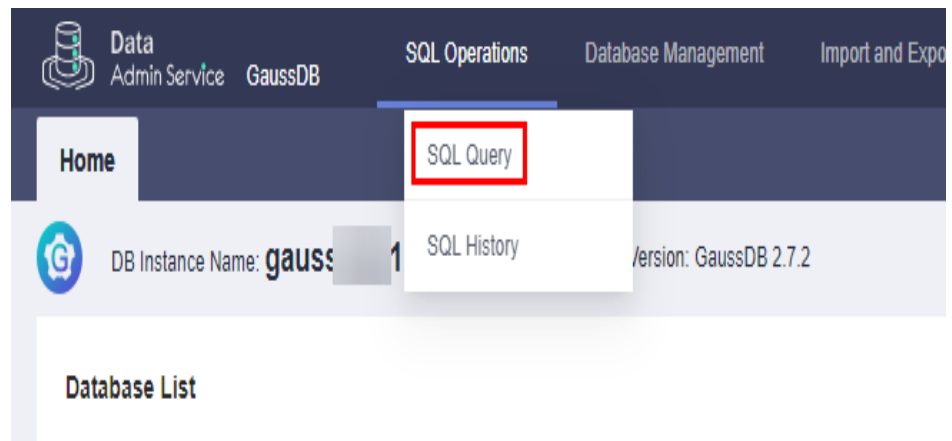
In the **Batch Upgrade** dialog box, select **Commit** for **Action**, select a target version, enter **YES**, and click **OK**.

Step 3: Verify the Upgrade

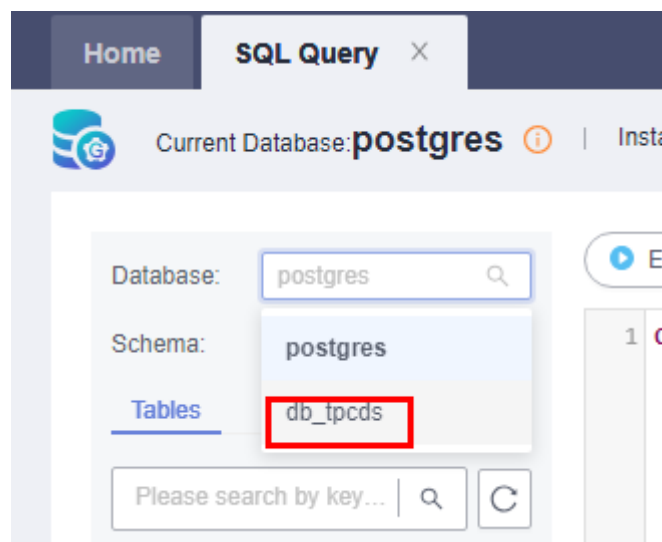
After the upgrade is complete, check the instance status, backup creation status, and instance connectivity, and whether you can add, delete, update, and query data in the instance.

1. On the **Instances** page, check whether **Status** of the target instance is **Available**.
2. On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check whether the value of **DB Engine Version** in the **Configuration** area is the target version.
3. Check that the automated backup triggered after the upgrade is successfully created.
 - a. On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
 - b. In the navigation pane, choose **Backups**. Check that a backup has been created and the backup status is **Completed**.
4. Check that the instance is properly connected and you can add, delete, update, and query data in the instance.

- a. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
- b. Go to the **SQL Query** page.

Figure 9-28 SQL query

- c. Create a database.
CREATE DATABASE *database_name*;
In this example, run the following command to create a database named **db_tpcds**:
CREATE DATABASE db_tpcds;
Switch to the newly created database in the upper left corner.

Figure 9-29 Switching to the new database

- d. Create a table and add, delete, update, and query data in the table.
 - i. Create a schema.
CREATE SCHEMA *myschema*;
 - ii. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.
CREATE TABLE *myschema.mytable (firstcol int);*

- iii. Insert data into the table.

```
INSERT INTO myschema.mytable VALUES (100);
```

- iv. View data in the table.

```
SELECT * FROM myschema.mytable;
```

```
| firstcol |  
-----+  
1 | 100 |
```

- v. Update data in the table.

```
UPDATE myschema.mytable SET firstcol = 200;
```

- vi. View the data in the table again.

```
SELECT * FROM myschema.mytable;
```

```
| firstcol |  
-----+  
1 | 200 |
```

- vii. Drop the table.

```
DROP TABLE myschema.mytable;
```

Rollback Operations

During upgrade observation, if a rollback is required due to service reasons or the upgrade using the rolling upgrade mode fails, you can manually roll back the upgrade by performing the steps in this section.

NOTE

- If the rollback is successful, you can perform the upgrade again.
- If the rollback fails, you can perform the rollback again.

If the problem persists, [submit a service ticket](#) to contact Huawei Cloud technical support, and the Huawei Cloud engineers will help you upgrade the instance if necessary.

[Method 1: Rolling Back a Single Instance]

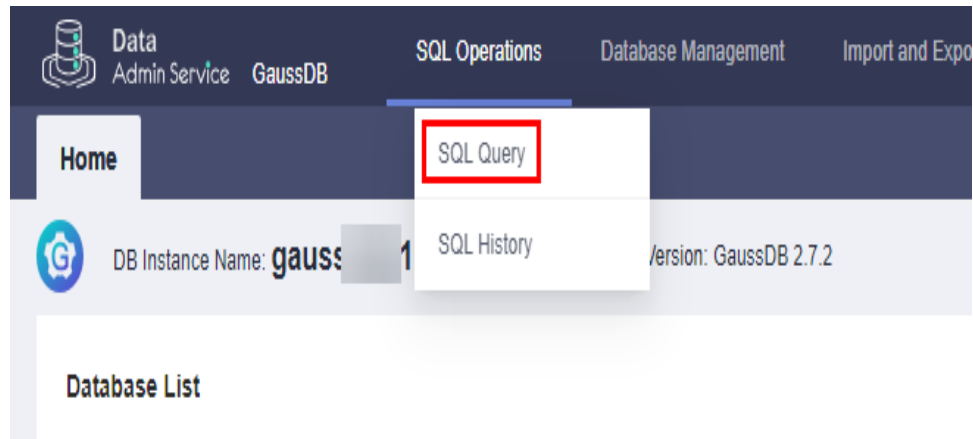
Step 1 In the **Upgrade Instance** dialog box, select **Rollback** for **Action**, select a target version, enter **YES**, and click **OK**.

Step 2 On the **Instances** page, check the rollback status. After the rollback is complete, the instance status changes to **Available**.

Step 3 On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check that the value of **DB Engine Version** in the **Configuration** area is the source version, that is, the version before upgrade.

Step 4 Check that the instance is properly connected and you can add, delete, update, and query data in the instance.

1. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
2. Go to the **SQL Query** page.

Figure 9-30 SQL query

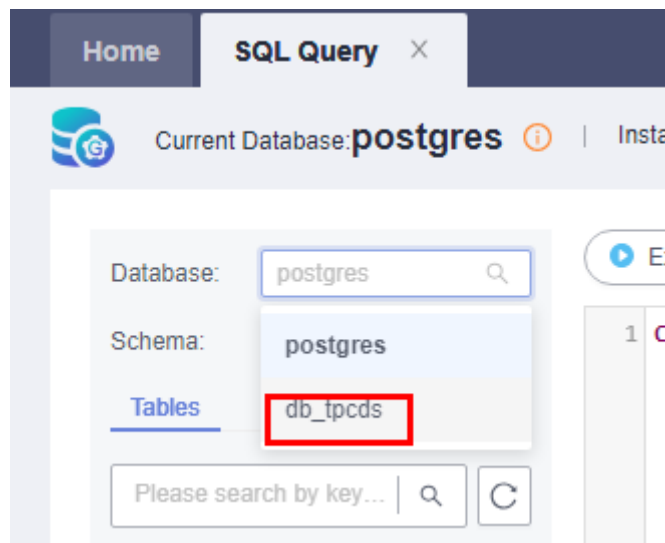
3. Create a database.

CREATE DATABASE *database_name*;

In this example, run the following command to create a database named **db_tpcds**:

CREATE DATABASE db_tpcds;

Switch to the newly created database in the upper left corner.

Figure 9-31 Switching to the new database

4. Create a table and add, delete, update, and query data in the table.

- a. Create a schema.

CREATE SCHEMA *myschema*;

- b. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.

CREATE TABLE *myschema.mytable (firstcol int);*

- c. Insert data into the table.

INSERT INTO *myschema.mytable* values (100);

- d. View data in the table.

```
SELECT * FROM myschema.mytable;
```

	firstcol
1	100

- e. Update data in the table.

```
UPDATE myschema.mytable SET firstcol = 200;
```

- f. View the data in the table again.

```
SELECT * FROM myschema.mytable;
```

	firstcol
1	200

- g. Drop the table.

```
DROP TABLE myschema.mytable;
```

----End

[Method 2: Rolling Back Instances in Batches]

Step 1 On the **Instances** page, select the target instances and click **Batch Upgrade**.

Step 2 In the **Batch Upgrade** dialog box, select **Rollback** for **Action**, select a target version, enter **YES**, and click **OK**.

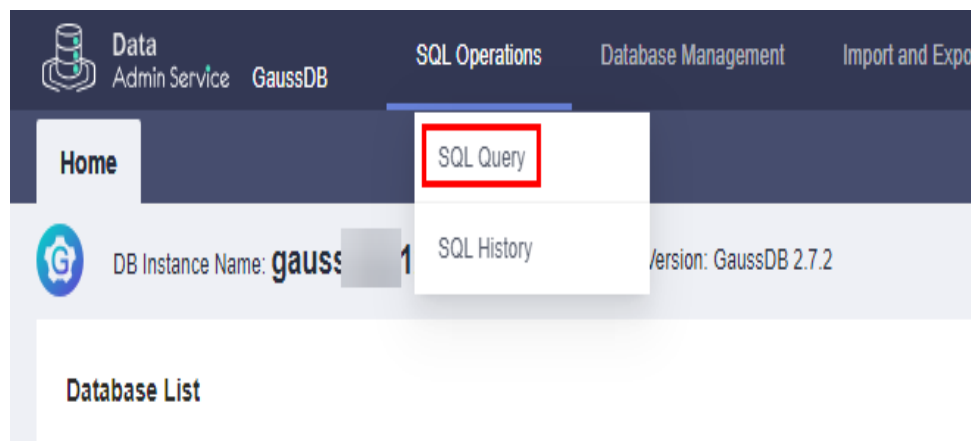
Step 3 On the **Instances** page, check the rollback status. After the rollback is complete, the instance status changes to **Available**.

Step 4 On the **Instances** page, click the name of the target instance. On the **Basic Information** page that is displayed, check that the value of **DB Engine Version** in the **Configuration** area is the source version, that is, the version before upgrade.

Step 5 Check that the instance is properly connected and you can add, delete, update, and query data in the instance.

1. Log in to the database. For details, see [Connecting to an Instance Through DAS](#).
2. Go to the **SQL Query** page.

Figure 9-32 SQL query



3. Create a database.

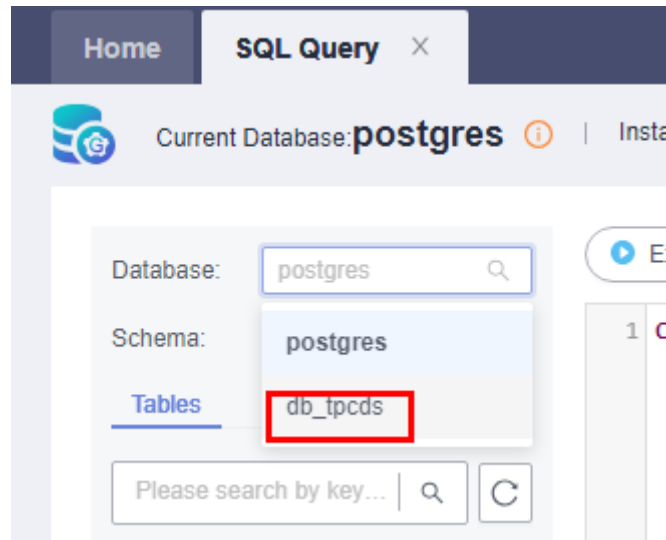
CREATE DATABASE *database_name*;

In this example, run the following command to create a database named **db_tpcds**:

CREATE DATABASE db_tpcds;

Switch to the newly created database in the upper left corner.

Figure 9-33 Switching to the new database



4. Create a table and add, delete, update, and query data in the table.

- a. Create a schema.

CREATE SCHEMA *myschema*;

- b. Create a table named **mytable** that has only one column. The column name is **firstcol** and the column type is integer.

CREATE TABLE *myschema.mytable (firstcol int);*

- c. Insert data into the table.

INSERT INTO *myschema.mytable* values (100);

- d. View data in the table.

SELECT * FROM *myschema.mytable*;

```
| firstcol |
-----+
1 | 100 |
```

- e. Update data in the table.

UPDATE *myschema.mytable* SET firstcol = 200;

- f. View the data in the table again.

SELECT * FROM *myschema.mytable*;

```
| firstcol |
-----+
1 | 200 |
```

- g. Drop the table.


```
DROP TABLE myschema.mytable;
```

```
----End
```

10 Add-on Management

10.1 Installing the Add-on

Scenarios

Yukon is a geographic information system (GIS) add-on developed by SuperMap. It enhances geospatial data storage and management, supports 2D/3D integrated spatial data storage, and provides advanced GIS features to standard relational databases. For more information about the add-on, see [KooGallery](#) > **Support** > **Yukon Product Documentation**.

You can install kernel add-ons (also known as plug-ins or extensions) for your GaussDB instances to enhance kernel functions. Only the PostGIS add-on provided by Yukon is supported.

Prerequisites

- You have purchased a license on [KooGallery](#) and obtained the download URL and SHA-256 hash of the add-on package from the third-party vendor. For details about the contact information, see [KooGallery](#) > **Support** > **Support Range**.
- The add-on package has been uploaded to the OBS of the end tenant. For details about how to upload an add-on package, see [Uploading an Object](#).

Precautions

- During the installation, the instance will be rebooted, which will temporarily interrupt database workloads.
- The add-on installation requests can be submitted repeatedly. If the add-on is installed for the first time, a license must be configured.
- If you set the **enable_default_ustore_table** parameter to **off**, new tables will use Astore for storage, while existing tables keep their current storage mode.
- The database version must match the GIS add-on version.

Constraints

- This function is only available for instances of version V2.0-8.218.
- If the instance or node status is abnormal, the add-on cannot be installed.
- The add-on cannot be uninstalled after being installed.
- During add-on installation, operations such as node repair and replacement, scale-out and scale-up, as well as hot patch installation are not supported.
- The Yukon GIS add-on version must match the required version of your DB instance. If you upgrade the instance, the GIS features from the installed add-on will stop working. The GIS add-on cannot be upgraded on the console.
- PostGIS extensions can be enabled or disabled for a maximum of 200 databases in an instance and for a maximum of 10 databases at a time.
- Before using the PostGIS add-on, check the value of the **behavior_compat_options** parameter. If the value contains **allow_procedure_compile_check**, **proc_implicit_for_loop_variable**, or **proc_outparam_override**, the add-on may fail to be used. You can modify the parameter on the console. For details, see [Modifying GaussDB Instance Parameters](#).
- Before using the PostGIS add-on, check whether the values of **enable_default_ustore_table** and **forbid_public_funcname_same_with_sysfunc** are **off**. If their values are **on**, the add-on cannot be used. You can change the parameter value to **off** on the console. For details, see [Modifying GaussDB Instance Parameters](#).
- Before using the PostGIS add-on and its extensions, check whether the value of **enable_astore_gist** is **on**. If the value is **off**, Generalized Search Tree (GiST) indexes cannot be used. You can change the parameter value to **on** on the console. For details, see [Modifying GaussDB Instance Parameters](#).
- The PostGIS add-on cannot be installed on primary and standby instances involved in a DR relationship.

Procedure

Step 1 [Log in to the management console](#).

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

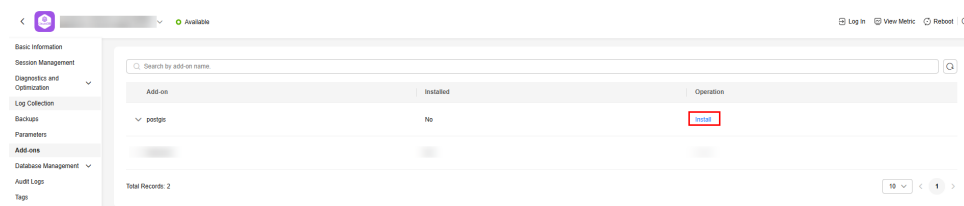
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Add-ons**.

Step 6 Click **Install** in the **Operation** column of the row containing the **postgis** add-on.

Figure 10-1 Installing the PostGIS add-on



Step 7 Enter the license, download URL, and SHA-256 hash, and click **OK**.

Figure 10-2 Installing the add-on

Install Add-on

The DB instance will be rebooted during add-on installation. The instance will be temporarily unavailable and the cache will be cleared. You are advised to install add-ons during off-peak hours. PostGIS does not support the Ustore storage engine. Make sure the current instance uses Astore when installing PostGIS. To enable Astore, set the GUC parameter `enable_default_ustore_table` to 'off'. If the DB engine version is V2.0-8.200 or later, set `forbid_public_funcname_same_with_sysfunc` to 'off', too.

License

Enter a license.

Add-on

postgis

Download URL

Enter a URL.

SHA-256 Hash

Enter the SHA-256 hash of the plug-in package.

Cancel

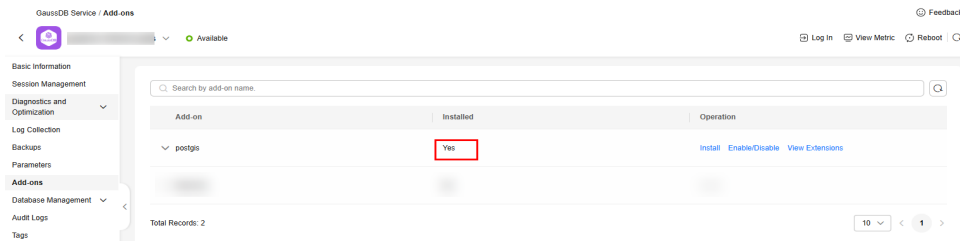
OK

Table 10-1 Parameter description

Parameter	Description
License	License provided by the third-party vendor. To obtain it, send the equipment serial number (ESN) to the third-party vendor. The ESN is the ID of the instance where the add-on is to be installed.
Add-on	Name of the add-on to be installed. The default add-on name is postgis .
Download URL	Shared object URL of the add-on package provided in OBS. For details, see Accessing an Object Using Its URL .
SHA-256 Hash	SHA-256 hash provided by the third-party vendor.

Step 8 Check the installation result.

During add-on installation, the instance status is **Installing add-on**. After the add-on is installed, the instance status becomes **Available**. On the **Add-ons** page, if **Installed** is **Yes** in the row containing **postgis**, the add-on is installed.

Figure 10-3 Checking the installation result

----End

10.2 Enabling or Disabling Extensions

You can enable or disable extensions provided by the installed add-on for a GaussDB instance. The PostGIS add-on only supports the following extensions: **postgis**, **postgis_sfcgal**, **postgis_raster**, **yukon_geomodel**, **yukon_geogridcoder**, and **postgis_topology**.

Precautions

- The extension enabling or disabling requests can be submitted repeatedly.
- A single extension can be enabled or disabled for multiple user databases.
- When enabling extensions, enable **postgis** first and then the other extensions.

Constraints

- If the instance or node status is abnormal, the extensions cannot be enabled or disabled.
- Before using the PostGIS add-on, check whether the values of **enable_default_ustore_table** and **forbid_public_funcname_same_with_sysfunc** are **off**. If their values are **on**, the add-on cannot be used. You can change the parameter value to **off** on the console. For details, see [Modifying GaussDB Instance Parameters](#).

Procedure

Step 1 [Log in to the management console](#).

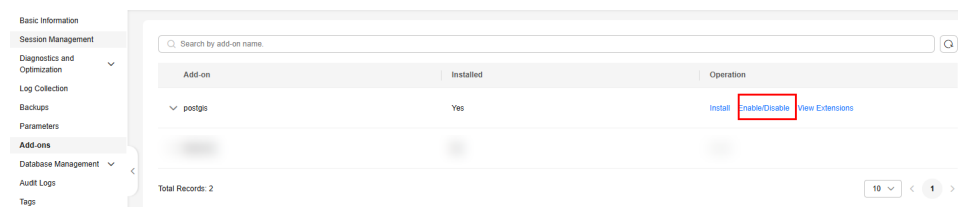
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page. In the navigation pane, choose **Add-ons**.

Step 5 Click **Enable/Disable** in the **Operation** column of the row containing the **postgis** add-on.

Figure 10-4 Enabling or disabling extensions



Step 6 Set **User Database**, **Extension**, and **Enable/Disable**, and click **OK**.

Table 10-2 Parameter description

Parameter	Description
User Database	Select one or more user-created service databases.
Extension	Six types of expansions are supported. The available extensions displayed vary from instance to instance. • postgis • postgis_sfcgal • postgis_raster • yukon_geomodel • yukon_geogridcoder • postgis_topology
Enable/Disable	Select Enable or Disable .

----End

10.3 Viewing Extensions

After the add-on is installed, you can check whether an extension is enabled or disabled for a specified user database.

Procedure

Step 1 [Log in to the management console](#).

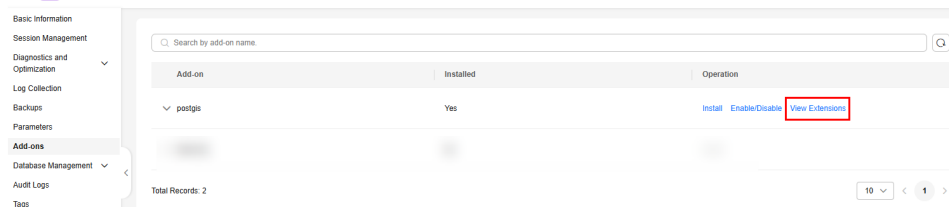
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page. In the navigation pane, choose **Add-ons**.

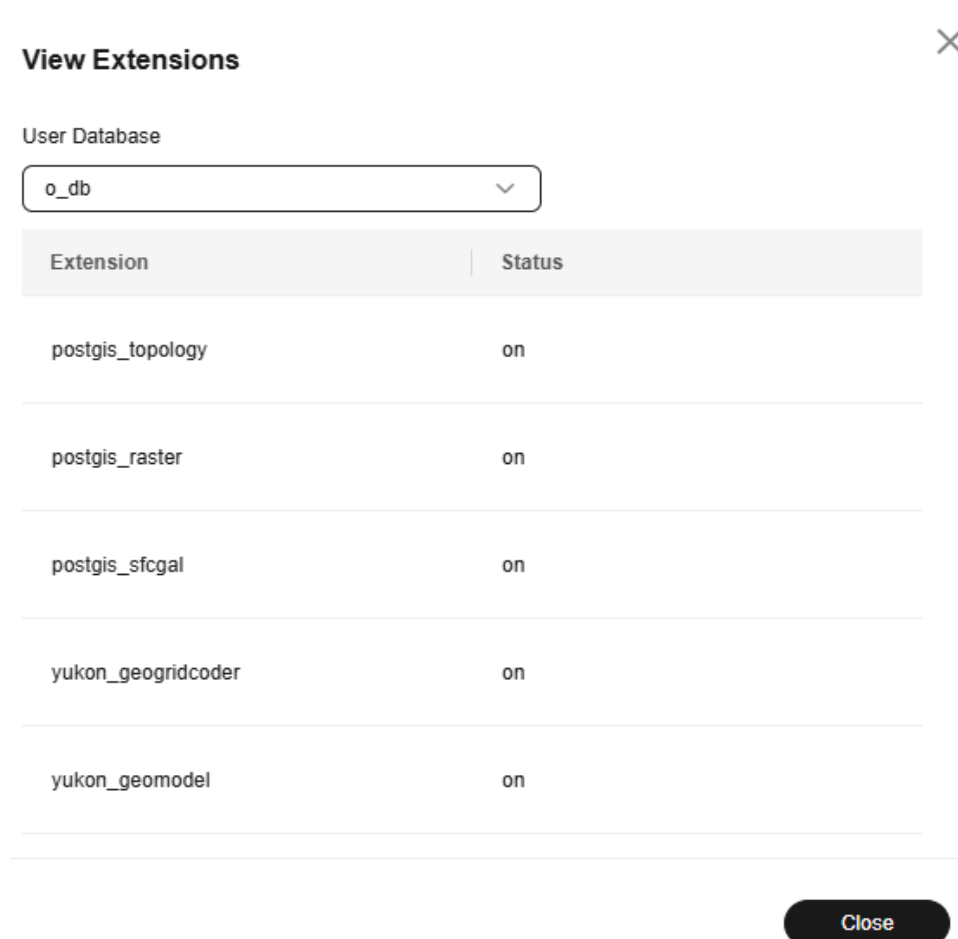
Step 5 Click **View Extensions** in the **Operation** column of the row containing the **postgis** add-on.

Figure 10-5 Viewing extensions



Step 6 In the dialog box that is displayed, check whether an extension is enabled or disabled.

Figure 10-6 Viewing extensions



-----End

11 Data Backup

11.1 Overview

You can back up your GaussDB instances to ensure data reliability. Currently, backups are stored in an unencrypted form.

Backups are stored in OBS buckets.

In a standard Huawei Cloud environment, a full backup or restoration of 2 TB of data can be completed within 8 hours.

Precautions

Xlogs are not reclaimed during backup.

Functions

Although GaussDB supports high availability, if a database or table is maliciously or mistakenly deleted, data on the standby nodes is also deleted. In this case, you can only restore the deleted data from backups.

Full Backup

A full backup involves all data of a database at the backup point in time. The time required for full backup is long (in direct proportion to the total data volume of the database). You can use a full backup to restore data of a complete database. A full backup backs up all data even if the data has not changed since the last backup.

Differential Backup

A differential backup involves only incremental data modified after a specified time point. It takes less time than a full backup in direct proportion to how much data has changed (The total data volume is irrelevant). However, a differential backup cannot be used to restore all of the data of a database. By default, the system automatically backs up updated data every 30 minutes since the last automated backup. The backup period can be changed from 15 minutes to 1,440 minutes.

Table-Level Full Backup

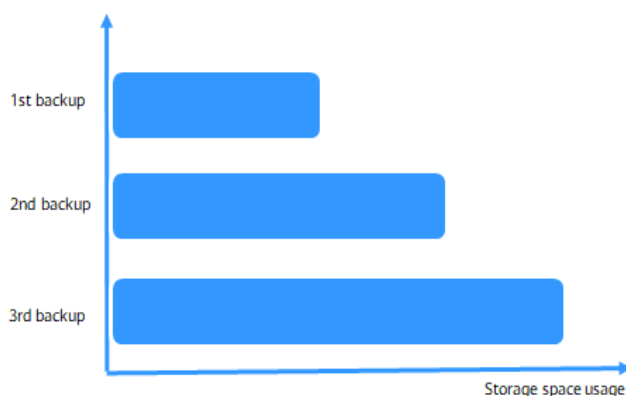
A table-level full backup involves all the data of a specified table at a specific point in time. The time required for table backup is long (in direct proportion to the total data volume of the table). You can use a full backup to restore all the data of a complete table. A table full backup backs up all data even if the data has not changed since the last backup.

Table-Level Differential Backup

A table differential backup involves only incremental data modified after a specified time point. It takes a short period of time (in direct proportion to the incremental data volume and irrelevant to the total data volume). However, a differential cannot restore all data of a table. By default, the system automatically backs up the updated table data every 30 minutes since the last automated backup. The backup period can be changed from 15 minutes to 72 hours.

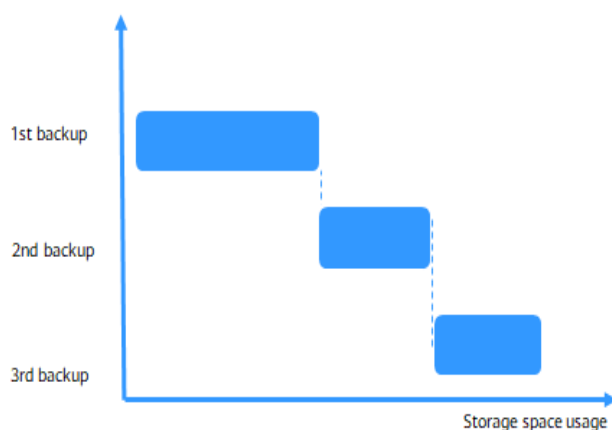
Backup Principles

Figure 11-1 Full backup



Full backup: After the first full backup, all data is backed up in the second and third backups regardless of whether the data is changed.

Figure 11-2 Differential backup



Differential backup: After the first full backup, the second backup backs up only the changed data, and the third backup backs up only the data changed after the second backup.

Automated Backup

Automated backups are created during the backup time window of your GaussDB instances. The system saves automated backups based on a retention period you specify. An automated backup is triggered after CNs or shards are added.

Table-Level Automated Backup

Automated backups of specified tables are created during the backup time window of your GaussDB instances. GaussDB saves automated table backups based on a retention period you specify. If necessary, you can restore the saved backups. If you have set a table-level backup policy, the system automatically backs up the table after CNs or shards are added.

Manual Backup

Manual backups are user-initiated full backups of instances. They are retained until you delete them manually.

Table-Level Manual Backup

Table-level manual backups are user-initiated full backups of instance tables. They are retained until you delete them manually.

11.2 Backup Execution

11.2.1 Configuring an Automated Backup Policy for GaussDB Instances

Scenarios

When you create a GaussDB instance, an instance-level automated backup policy is enabled by default. After your instance is created, you can modify the automated backup policy as needed. GaussDB backs up data based on the automated backup policy you specified.

If a database is faulty or data is damaged, you can restore it from backups to ensure data reliability. Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects the database read and write performance, you are advised to perform automated backups during off-peak hours.

Constraints

- The instance-level automated backup policy cannot be configured for GaussDB single-replica instances of versions earlier than V2.0-3.0. For details

about how to view the instance version, see [Checking the DB Engine Version](#).

- To ensure that data can be restored to a specific point in time, the most recent full backup that exceeds the backup retention period will not be deleted immediately.

For example, with the automated backup policy set to perform daily backups and retain them for one day, if backup 1 is generated on November 1, this backup will not be deleted on November 2 when backup 2 is generated, but will be deleted on November 3 when backup 3 is generated and backup 2 is retained.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see [How Is GaussDB Backup Data Charged?](#)

Modifying an Automated Backup Policy

[Method 1: modifying the automated backup policy for a single instance]

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click **Modify Backup Policy**. You can view the configured backup policy. To modify the backup policy, adjust the parameter values as needed.

Figure 11-3 Modifying the backup policy

×

Modify Backup Policy

Full Backup Policy

Automated Backup ☒

Retention Period
Enter an integer from 1 to 732.

Backup Flow Control
The upload speed when backing up data to OBS. Value range: 0 to 1024. Default value 75. 0 indicates that the speed is not limited.

Standby Node Backup ☒
If the standby node is abnormal, backups will be switched to the primary node.

Time Zone GMT+08:00

Time Window

Backup Cycle

Monday × Tuesday × Wednesday ×
Thursday × Friday × Saturday ×

A minimum of one day must be selected. If there is a long backup cycle, the point-in-time recovery (PITR) will take longer.

Differential Backup Policy

Backup Cycle
▲ If a short interval is selected, backup task conflicts may occur.

Cancel

OK

Step 6 Configure parameters.

Table 11-1 Full backup policy

Parameter	Description
Retention Period	Number of days that your automated backups can be retained. Increasing the retention period will improve data reliability. The default value is 7. Full backups are retained till the retention period expires. However, even if the retention period has expired, the most recent backup will be retained. • Extending the retention period improves data reliability. You can extend the retention period as needed. • If you shorten the retention period, the new backup policy takes effect for existing backups. Any automated backups (including full and incremental backups) that have expired will be automatically deleted. Manual backups will not be automatically deleted but you can delete them manually. Policy for automatically deleting automated full backups: To ensure data integrity, even after the retention period expires, the most recent backup will be retained. If Backup Cycle was set to Monday and Tuesday and the Retention Period was set to 2: • The full backup generated on Monday will be automatically deleted on Thursday. The reasons are as follows: The backup generated on Monday expires on Wednesday, but it was the last backup, so it will be retained until a new backup expires. The next backup will be generated on Tuesday and will expire on Thursday. So the full backup generated on Monday will not be automatically deleted until Thursday. • The full backup generated on Tuesday will be automatically deleted on the following Wednesday. The reasons are as follows: The backup generated on Tuesday will expire on Thursday, but as it is the last backup, so it will be retained until a new backup expires. The next backup will be generated the next Monday and will expire on the next Wednesday. So the full backup generated on Tuesday will not be automatically deleted until the next Wednesday.
Backup Flow Control	Rate at which data is uploaded from the data disk of the instance to the backup storage device (such as OBS). The default rate is 75 MB/s. The value 0 indicates that the upload rate is not limited. However, the actual upload rate is still restricted by factors such as the network, instance specifications, and disk I/O.
Standby Node Backup	If this policy is enabled, full and differential backups of the instance are performed on the hosts where standby DN's reside. By default, this feature is enabled.

Parameter	Description
Time Window	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00. The backup time is in UTC format. If the DST or standard time is switched, the backup time segment changes with the time zone.
Backup Cycle	One or more days from Monday to Sunday when a backup task will be triggered. By default, all options are selected. Select at least one day as required. Backups can be retained for 1 to 732 days. To extend the retention period, submit a service ticket. Automated backups can be retained for up to 2,562 days. A time window is one hour. A total of 24 time windows are available. You are advised to select an off-peak time window for full backups. By default, each day of the week is selected for Backup Cycle. You can change the backup cycle. At least one day must be selected. A full backup is immediately triggered after a DB instance is created. Then, a full backup or differential backup is performed based on the time window and backup cycle you specified. We recommend that you set the full backup time window to an off-peak hour.

Table 11-2 Differential backup policy

Parameter	Description
Backup Cycle	You need to select the backup cycle for performing a differential backup. The default value is 30 minutes.
Prefetch Pages	This parameter specifies the number of prefetch pages from the modified pages in the disk table file during a differential backup. The default value is 64 . When modified pages are adjacent (for example, with a bulk data load), you can set this parameter to a large value. When modified pages are scattered (for example, random update), you can set this parameter to a small value. If this parameter is set to a large value, the occupied I/O increases. In this case, other services are affected and database performance deteriorates.

Step 7 Click **OK**.

Step 8 Check the result.

After the task is submitted, click **Modify Backup Policy** to check whether the modification is successful.

----End

[Method 2: modifying the automated backup policy for multiple instances]

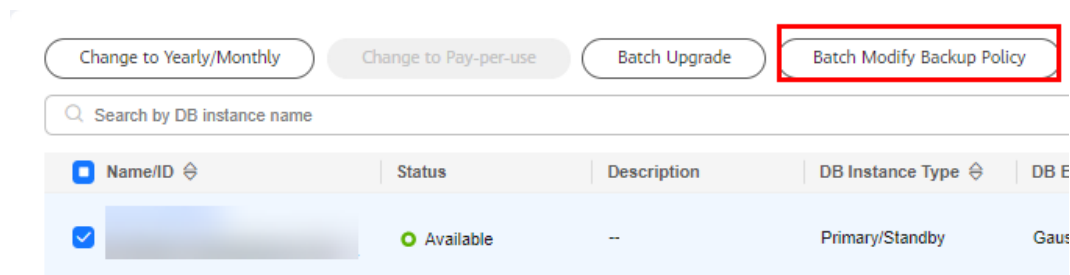
Step 1 Log in to the management console.

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

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

Step 4 On the **Instances** page, select the target instances and click **Batch Modify Backup Policy**.

Figure 11-4 Modifying the backup policy in batches



Step 5 In the **Batch Modify Backup Policy** dialog box, set the instance-level backup policy as required.

Step 6 Click **OK**.

 NOTE

- A maximum of 30 instances can be selected for modifying the backup policy in batches.
- If the backup policy is disabled for a selected instance, the batch modification operation will not update the backup policy for the instance.
- **Time Window** and **Backup Cycle** use UTC. To avoid scheduling mismatches in your local time zone, modify both fields if you want to update either of them.

Step 7 View the modification result.

After the task is submitted, click a target instance name, and choose **Backups** in the navigation pane on the left. Then, click **Modify Backup Policy** to check whether the backup policy is modified successfully.

----End

FAQ

Why Does Automated Backup for a GaussDB Instance Fail?

11.2.2 Configuring an Automated Backup Policy for GaussDB Tables

Scenarios

When you create a GaussDB instance, the table-level automated backup policy is disabled by default. After your instance is created, no table-level full backup task will be triggered. You can modify the automated backup policy for tables as needed. If such a policy is configured, GaussDB backs up data of specified tables based on the automated backup policy you specified.

Precautions

- After an instance-level restoration is complete, the databases or tables configured in the table-level automated backup policy may no longer exist. In this case, you need to configure the table-level automated backup policy again.
- To ensure that data can be restored to a point in time, the latest full backup that exceeds the backup retention period will not be deleted immediately. For example, if **Backup Cycle** is set to **All** and **Retention Period** to one day and backup 1 is generated on November 1, this table backup will not be deleted on November 2 when backup 2 is generated, but will be deleted on November 3 when backup 3 is generated.
- Tables in the **postgres**, **template0**, **template1**, **templatem**, **templatea**, and **templatepdb** system databases cannot be backed up using the table-level backup method.
- Tables in system schemas (for example, **public**) cannot be backed up using the table-level backup method.
- Tables in M-compatible databases cannot be backed up using the table-level backup method.
- A maximum of 100 databases or tables can be restored at the same time. If there are more than 100 databases or tables, you are advised to use instance-level backup and restoration.
- Table-level restoration does not support segment-page tables, column-store tables, tables containing user-defined types, synonym tables, temporary tables (including global tables), encrypted tables, TDE-enabled tables, unlogged tables, compressed tables, tables of private users, and tables of ledger databases.
- Table-level backup is unavailable to single-node instances.
- If the name of a table, schema, or database to be backed up is changed, you need to reconfigure the table-level backup policy.
- Backups are saved as packages in OBS buckets to ensure data confidentiality and durability. Since backing up data affects database read and write performance, you are advised to perform automated backups for tables during off-peak hours.
- If there is a new table available for backup, the new table will be backed up during the next scheduled full or differential backup.
- If a table covered by the table-level backup policy is deleted, modify the policy immediately to remove that table. Otherwise, the table-level backup will keep failing.
- To disable the table-level differential backup policy, [submit a service ticket](#).

Billing

Backups are saved as packages in OBS buckets. For the billing details, see [How Is GaussDB Backup Data Charged?](#)

Modifying an Automated Backup Policy for Tables

Step 1 [Log in to the management console](#).

- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane on the left, choose **Backups**. On the displayed page, choose **Full Backups > Table Backup**. Click **Modify Backup Policy**.
- Step 6** Configure parameters.

Table 11-3 Backup policies

Backup Policy	Parameter	Description
Full Backup Policy	Retention Period	Number of days that your automated backups can be retained. Increasing the retention period will improve data reliability. The default value is 7. If you shorten the retention period, the new backup policy will be applied for all backup files. The backup files that have expired will be deleted. Backups can be retained for 1 to 732 days. To extend the retention period, submit a service ticket . Automated backups can be retained for up to 2,562 days.
	Time Window	A one-hour period the backup will be scheduled within 24 hours, such as 01:00-02:00 or 12:00-13:00. To minimize the potential impact on services, set the time window to an off-peak hour.
	Backup Cycle	One or more days from Monday to Sunday when a backup task will be triggered. By default, all options are selected. Select at least one day as required.
Differential Backup Policy	Backup Cycle	Interval at which table-level differential backup is performed. The default cycle is 30 minutes.

- Step 7** Click **Modify Backup Policy**.

----End

FAQ

Why Does Automated Backup for a GaussDB Instance Fail?

11.2.3 Setting a Cross-Region GaussDB Backup Policy

Scenarios

To ensure remote disaster recovery, GaussDB can store backups in a region different from the region where your instance is located, allowing you to restore a new GaussDB instance there to recover your workloads. After you enable cross-region backup for an instance, the system automatically stores its backups to the specified target region based on the configured backup policy. On the **Backups** page, you can click **View Backup** in the **Operation** column and manage cross-region backup files.

NOTE

This function is only available for authorized users. To apply for the permissions needed, [submit a service ticket](#).

Precautions

- After the cross-region backup policy is enabled, automated full backups of your GaussDB instance can be synchronized to the region you specify. Differential backups will be synchronized to the target region only after the corresponding full backups and all related differential backups have been successfully synchronized. The cross-region backup policy is not suitable for manual backups, archive logs, and table-level backups.
- If the cross-region backup policy is disabled, the cross-region backup task will be stopped immediately, and all cross-region backup files will be immediately deleted. Operations related to cross-region backup may fail.
- After your instance is deleted, all cross-region backup files of the instance are retained. The retention period depends on the **Retention Period** that you specify when you enable cross-region backup policy.
- Cross-region backup cannot be enabled for multitenant container DB instances.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click **Set Cross-Region Backup Policy**.

Step 6 Configure parameters.

- **Cross-Region Full Backup**

Enabling it means that automated full backup files of your instance are stored in OBS in the region you specify.

- **Cross-Region Differential Backup**

Enabling it means that automated differential backup files of your instance are stored in OBS in the region you specify.

 NOTE

Before enabling this function, ensure that there is enough capability in OBS in the region you specify.

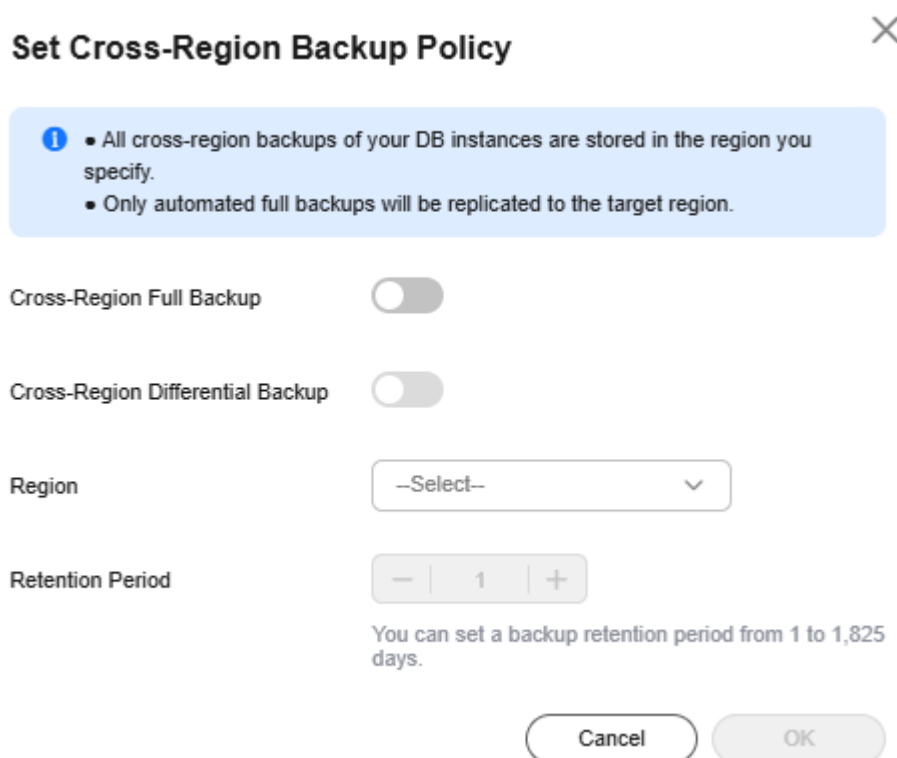
- **Region**

Region for which you back up data for.

- **Retention Period**

How long cross-region backups can be retained. Value range: 1 to 1825 days. Increasing the retention period improves data reliability but increases OBS space usage. Set this parameter based on service requirements.

Figure 11-5 Setting the cross-region backup policy



Set Cross-Region Backup Policy ✕

Information:

- All cross-region backups of your DB instances are stored in the region you specify.
- Only automated full backups will be replicated to the target region.

Cross-Region Full Backup ☐

Cross-Region Differential Backup ☐

Region --Select--

Retention Period 1

You can set a backup retention period from 1 to 1,825 days.

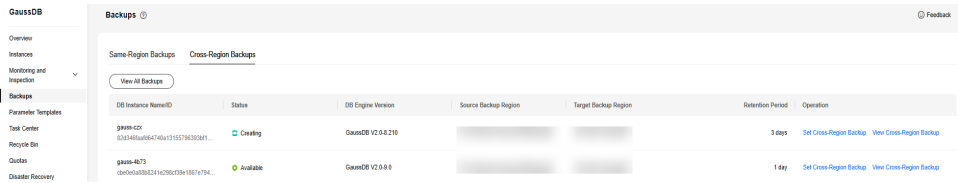
Cancel OK

Step 7 Click **OK**.

Step 8 Go back to the **Instances** page and choose **Backups** from the navigation pane on the left.

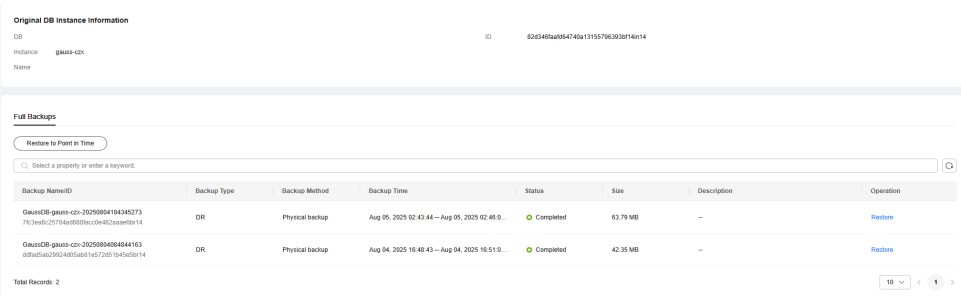
Step 9 On the **Cross-Region Backups** tab of the **Backups** page, manage cross-region backups.

Figure 11-6 Managing cross-region backups



- To modify the cross-region backup policy, click **Set Cross-Region Backup** in the **Operation** column.
- To view generated cross-region full and differential backups, click **View Cross-Region Backup** in the **Operation** column. You can use the backups to restore data to a new instance.

Figure 11-7 Cross-region backups



- Click **Restore** in the **Operation** column to restore data to a new instance using the full backup.
- Click **Restore to Point in Time** to restore data to a new instance at a specified time point using the generated full and differential backups.

----End

11.2.4 Creating a Manual Backup for GaussDB Instances

Scenarios

GaussDB allows you to create instance-level manual backups for running instances. You can restore data using manual backups to ensure data reliability.

Precautions

- Manual backups are user-initiated full backups of instances. They are retained until you delete them manually.
- You can back up data of instances that are in the **Available** state.
- A user can perform only one instance-level backup operation for a DB instance at a time.
- Instance-level manual backups cannot be created for GaussDB single-replica instances of versions earlier than V2.0-3.0. For details about how to view the instance version, see [Checking the DB Engine Version](#).

Billing

Backups are saved as packages in OBS buckets. For the billing details, see [How Is GaussDB Backup Data Charged?](#)

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required. For details, see [Product Pricing Details](#).

Method 1

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, locate the instance and choose **More > Create Backup** in the **Operation** column.

Step 5 In the displayed dialog box, enter a backup name and enter the description as needed. Then, click **OK**. If you want to cancel the backup creation task, click **Cancel**.

To check whether the backup has been successfully created, click  in the upper right corner of the page. If the instance status is **Available**, the backup has been created. You can manage the backup by following the instructions provided in [Step 6](#).

- The backup name is case-sensitive and can contain 4 to 64 characters. It must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- The description can contain up to 256 characters, but cannot contain carriage returns and the following special characters: >!<"&'='
- During the creation process, the instance status is **Backing up**. The time required for creating a manual backup depends on the data volume.

Step 6 View and manage the created backup on the **Backups** page.

Alternatively, click the instance name. On the **Backups** page, you can view and manage the manual backups.

----End

Method 2

Step 1 [Log in to the management console](#).

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

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

- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane on the left, choose **Backups**. On the displayed page, click **Create Backup**.
- Step 6** In the displayed dialog box, enter a backup name and description and click **OK**.
- The backup name is case-sensitive and can contain 4 to 64 characters. It must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.
 - The description can contain up to 256 characters, but cannot contain carriage returns and the following special characters: >!<"&'=
 - During the creation process, the manual backup status is **Creating**. The time required for creating a manual backup depends on the data volume.
- Step 7** View and manage the created backup on the current page.

Alternatively, go back to the instance list page, and click **Backups** to view and manage the backup.

----End

11.2.5 Creating a Manual Backup for GaussDB Tables

Scenarios

GaussDB allows you to create table-level manual backups for available instances. You can use these backups to restore data for ensuring data reliability.

Precautions

- You can back up data of instances that are in the **Available** state.
- A user can perform only one table-level backup operation for a DB instance at a time.
- Keep your account balance above zero so that table-level backups can be created.
- Tables in the **postgres**, **template0**, **template1**, **templatea**, and **templatepdb** system databases cannot be backed up using the table-level backup method.
- Tables in M-compatible databases cannot be backed up using the table-level backup method.
- Tables in system schemas (for example, **public**) cannot be backed up using the table-level backup method.

By default, **public** is a system schema. However, you can run commands to change the owner of **public**. When the owner is changed to a non-system user, you can back up tables in the **public** schema.

- When a DB instance is deleted, the automated backups for tables are also deleted, but manual table backups are retained.

Billing

Backups are saved as packages in OBS buckets. For the billing details, see [How Is GaussDB Backup Data Charged?](#)

After a DB instance is deleted, the free backup space of the DB instance is automatically canceled. Manual backups are billed based on the space required. For details, see [Product Pricing Details](#).

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click **Table Backup** under **Full Backups** and click **Create Backup**.

Step 6 Enter the backup name and description, select the required data and click **Create Backup**.

- The backup name is case-sensitive and can contain 4 to 64 characters. It must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.
- The description can contain up to 256 characters, but cannot contain carriage returns and the following special characters: >!<"&'='
- During the creation process, the status is **Creating**. The time required for creating a manual backup depends on the data volume.

Step 7 View and manage the created backup on the current page.

Alternatively, go back to the instance list page, and click **Backups** to view and manage the backup.

----End

11.3 Backup Management

11.3.1 Downloading a GaussDB Backup

Scenarios

You can download backup files from OBS, including instance-level and table-level full backups and differential backups, for local backup.

Constraints

- Snapshot-based backups cannot be downloaded.
- Only OBS Browser+ can be used for download.

Procedure

Step 1 Log in to the management console.

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

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

Step 4 In the navigation pane, choose **Backups**. On the displayed page, click **Download** in the **Operation** column of the backup you want to download.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane on the left, choose **Backups**. Click the **Instance Backup** tab under **Full Backups**. On the displayed page, click **Download** in the **Operation** column of the backup you want to download.

Step 5 In the displayed dialog box, retain the only option **Use OBS Browser+** for **Download Method** and click **OK**.

Figure 11-8 Downloading a backup

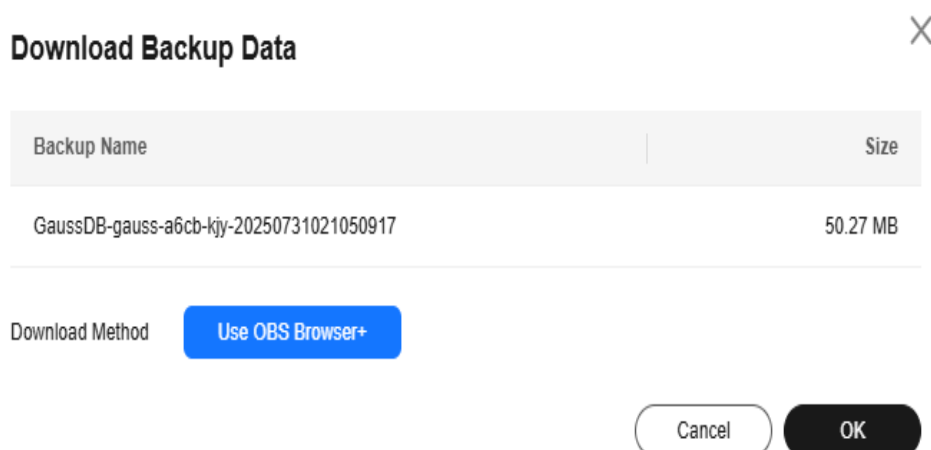
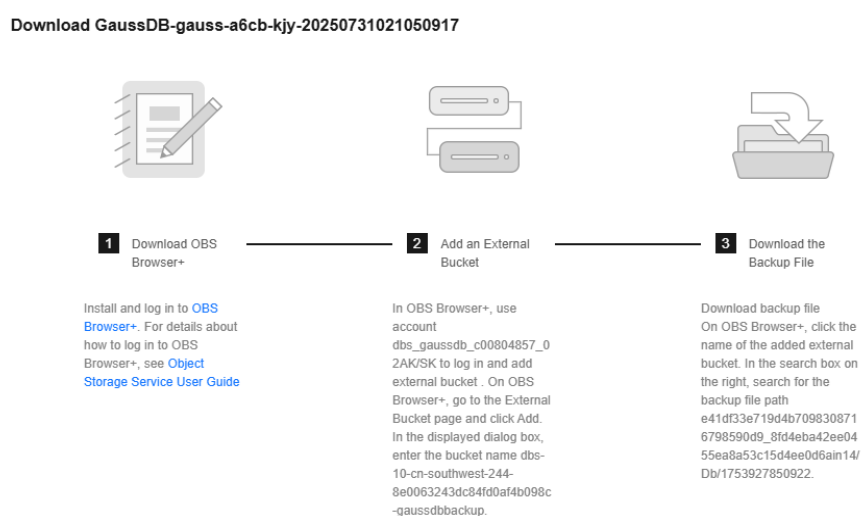


Figure 11-9 Using OBS Browser+ to download data

1. Download OBS Browser+.
For details, see [Downloading OBS Browser+](#).
2. Decompress the package and install OBS Browser+.
For details, see [Installing OBS Browser+](#).
3. Log in to OBS Browser+.
For details, see [Logging In to OBS Browser+](#).

NOTE

1. When logging in to OBS Browser+, you need to use the AK/SK for signature verification. To obtain the AK/SK, perform the following steps:
 1. Log in to the console.
 2. Click the username in the upper right corner and choose **My Settings** from the drop-down list.
 3. Click the **Access Keys** tab.
 4. Click **Add Access Key**. In the displayed dialog box, click **OK**. The access key file (**credentials.csv**) is automatically downloaded to your local computer. You can obtain the AK/SK in the file.
4. Add an external bucket.
On OBS Browser+, go to the **External Bucket** page and click **Add**. In the displayed dialog box, enter the bucket name provided on the download guideline page shown in [Figure 11-9](#) and click **OK**.
5. Download the backup file.
On OBS Browser+, click the name of the bucket that you added. In the search box on the right of the displayed page, enter the backup file name displayed on the download guideline page shown in [Figure 11-9](#) on the GaussDB console. In the search result, locate the target backup and download it.
6. After the backup is downloaded, enable OBS Browser+ certificate verification.

----End

11.3.2 Exporting Backup Information About GaussDB Instances

Scenarios

You can export backup information of instances to a CSV file for further analysis. The exported information includes the backup ID, backup name, instance name, instance ID, DB engine, backup type, backup start time, backup end time, backup status, backup size, and backup description.

Precautions

- Backup information cannot be exported for GaussDB single-replica instances of versions earlier than V2.0-3.0. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- You can only export the backup data shown on the current page; data on other pages cannot be exported.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 In the navigation pane on the left, choose **Backups**. Select the backups you want to export and click **Export** to export the backup information.

Alternatively, click the instance name. On the **Backups** page, select the backups you want to export and click **Export** above the backup list to export the backup information.

The exported backup information is in a CSV file which facilitates your analysis.

Step 5 View the exported backup information.

----End

11.3.3 Stopping a Backup for a GaussDB Instance

Scenarios

You can stop instance-level backup tasks as needed for your GaussDB instance, including automated full backups, manual full backups, and differential backups.

Precautions

Stopping a backup for an instance will stop all its ongoing full and differential backup tasks.

- Backups can be stopped only for instances whose DB engine version is V2.0-2.8 or later. You can view the backup stopping task information of an

instance on the **Task Center** page. For details about how to view the instance version, see [Checking the DB Engine Version](#).

- Backup tasks may fail to be stopped in the following scenarios:
 - The backup task is about to complete, that is, the backup is complete, but the status has not been updated. Wait until the backup task status changes to complete.
 - The backup task has just started executing and the backup process has not been started. In this case, try again later to stop the backup.
- You are advised not to stop the first automated backup after an instance is changed or restored. Forcibly stopping that backup may cause incremental and differential backups between the current time and the next automated full backup to fail, and point-in-time restoration may be unavailable. Stop that backup only when absolutely necessary.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click **Stop Backup**.

Step 6 Click **OK**.

Step 7 Choose **Task Center** in the navigation pane on the left. On the displayed page, view the task details.

----End

11.3.4 Deleting a Manual Backup of a GaussDB Instance

Scenarios

You can delete manual backups for GaussDB instances to release storage space.

Precautions

- Deleted manual backups cannot be recovered. Exercise caution when performing this operation.
- Automated backups cannot be manually deleted.
- Backups that are being restored cannot be deleted.
- To delete a backup, you must log in to the account that the backup belongs to.
- Manual backups cannot be deleted for GaussDB single-replica instances of versions earlier than V2.0-3.0. For details about how to view the instance version, see [Checking the DB Engine Version](#).

Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup you want to delete and click **Delete** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane on the left, choose **Backups**. On the displayed page, locate the manual backup you want to delete and click **Delete** in the **Operation** column.

Figure 11-10 Deleting a manual backup

☐ Backup Name/ID	Backup...	Backup Met...	Backup Time	Status	Size	Description	Operation
☐ backup-5ed3 1fa5708b4fa24b5c85...	Manual	Physical ba...	Aug 15, 2024 20:...	 Compl...	40.48 MB	--	Restore Delete
☐ GaussDB-gauss-124... ce9abbe2fcd484c84...	Automated	Physical ba...	Aug 15, 2024 17:...	 Compl...	33.46 MB	--	Restore

Step 5 Click **OK**.

After the backup is deleted, it will not be displayed on the **Backups** page.

Step 6 If you have enabled operation protection, click **Send Code** in the displayed **Identity Verification** dialog box and enter the obtained verification code. Then, click **OK**.

Two-factor authentication improves the security of your account. For details about how to view and enable high-risk operation protection, see [Identity and Access Management User Guide](#).

-----End

12 Data Restoration

12.1 GaussDB Restoration Methods for Data Misoperations

You can choose an appropriate data restoration method based on the site requirements.

Table 12-1 Restoration methods for misoperations

Scenario	Recovery Method	Restoration Scope	Restore To	Operation Guide
An instance is deleted by mistake.	Recycle bin: Locate the deleted instance in the recycle bin and rebuild the instance to restore it.	All databases and tables	A new instance	Rebuilding a GaussDB Instance
	Instance backup: If a manual backup has been created before the instance is deleted, restore the instance on the Backups page.	All databases and tables	• A new instance • An existing instance	Restoring a Backup File to a GaussDB Instance

Scenario	Recovery Method	Restoration Scope	Restore To	Operation Guide
A table is deleted by mistake.	Restore the tables that are deleted by mistake by referring to the methods of restoring an instance.	All databases and tables	• A new instance • An existing instance • Current instance	• Restoring a Backup File to a GaussDB Instance • Restoring a GaussDB Instance to a Specific Point in Time
	Use the database and table restoration method to restore the table.	• All databases and tables • Certain databases and tables		• Restoring a GaussDB Database or Table Using a Backup File • Restoring a GaussDB Database or Table to a Specific Point in Time
A database is deleted by mistake.	Restore the databases that are deleted by mistake by referring to the methods of restoring an instance.	All databases and tables	• A new instance • An existing instance • Current instance	• Restoring a Backup File to a GaussDB Instance • Restoring a GaussDB Instance to a Specific Point in Time

Scenario	Recovery Method	Restoration Scope	Restore To	Operation Guide
	Use the database and table restoration method to restore the database.	• All databases and tables • Certain databases and tables		• Restoring a GaussDB Database or Table Using a Backup File • Restoring a GaussDB Database or Table to a Specific Point in Time
An entire table is overwritten, or the columns, rows, or data in a table is deleted or modified by mistake.	Restore the data that is deleted by mistake by referring to the methods of restoring an instance.	All databases and tables	• A new instance • An existing instance • Current instance	• Restoring a Backup File to a GaussDB Instance • Restoring a GaussDB Instance to a Specific Point in Time
	Use the database and table restoration method to restore table data.	• All databases and tables • Certain databases and tables		• Restoring a GaussDB Database or Table Using a Backup File • Restoring a GaussDB Database or Table to a Specific Point in Time

12.2 Restoring a Backup File to a GaussDB Instance

Scenarios

You can use an instance-level automated or manual backup to restore data to the point in time when the backup was created. The restoration is at the DB instance level.

Data can be restored to a new DB instance, an existing DB instance, or the original DB instance.

Precautions

- Restoration will fail if the instance is in the **Abnormal**, or **Storage full** state.
- GaussDB currently only supports restoration between DB instances running the same major version. For example, backup data can only be restored from version V2.0-1.4.x to version V2.0-1.4.y.
- If parallel restoration is enabled, all primary and standby replicas will simultaneously download backup data from the storage server. Compared to serial restoration, this process demands a multiple of the bandwidth from the storage server equal to the number of replicas in each shard. If your instance has more than five shards, consult O&M personnel about the available bandwidth of the OBS server before deciding whether to enable parallel restoration. This will prevent a decrease in restoration speed due to bandwidth exhaustion.
- Centralized instances support only parallel restoration.
- Parallel restoration cannot be enabled if the DB engine version is earlier than V2.0-1.4.
- In addition to full backups and incremental backups, the system backs up incremental log files to ensure data consistency. It takes some time to back up and upload incremental log files (The time depends on the network and OBS traffic control). Note that the backup completion time does not represent the data consistency point that can be specified when this backup set is used to restore data. (Generally, the data consistency point is within several minutes before the backup completion time.) If you have strict requirements on data consistency after restoration, restore data to a specified point in time.

Restoring Data to a New Instance

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane, choose **Backups**. On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the name of the target instance on the **Instances** page. In the navigation pane, choose **Backups**. On the **Full Backups** page, click the **Instance**

Backup tab, and click **Restore** in the **Operation** column of the backup to be used for restoration.

Step 5 Set **Restoration Method** to **Create New Instance** and click **OK**.

- The new DB instance must have the same major version as the original instance when it was backed up. For example, backup data can only be restored from version V2.0-1.4.x to version V2.0-1.4.y.
- The storage space of the new instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance. The storage space for a single shard starts from 40 GB and can be increased at a step of 4 GB.
- The administrator password needs to be reset.
- By default, the instance specifications of the new instance are the same as those of the original instance. To change the instance specifications, ensure that the instance specifications of the new instance are at least those of the original instance.
- The backup media, CPU architecture, OS type and version, instance type (distributed or centralized), deployment model, replica consistency protocol, and transaction consistency settings of the new instance must be the same as those of the original instance to be restored.
- If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the new instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.

Step 6 On the displayed page, configure parameters of the new instance and click **Next**.

Table 12-2 Basic information

Parameter	Description
Billing Mode	GaussDB provides yearly/monthly billing and pay-per-use billing. • Yearly/Monthly: You pay upfront for the amount of time you expect to use the DB instance for. You will need to make sure you have a top-up account with a sufficient balance or have a valid payment method configured first. For distributed instances using the combined deployment model, yearly/monthly billing is only available to authorized users. To apply for the permissions needed, submit a service ticket. • Pay-per-use: You can start using the DB instance first and then pay as you go. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.
DB Instance Name	The instance name is case-sensitive, must start with a letter, and can contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Failover Priority	This parameter is only available for distributed instances. This parameter is only available for authorized users. To apply for the permissions needed, submit a service ticket. The default value is Reliability. For details about how to change the failover priority for an existing instance, see Changing Failover Priority. • Reliability: Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency. • Availability: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. If Availability is selected, exercise caution when modifying the following database parameters. For details about how to modify parameters, see Modifying Instance Parameters. – recovery_time_target: If the value of this parameter is changed, the instance will experience frequent forced failovers. Contact technical support before modifying this parameter. – audit_system_object: If the value of this parameter is changed, DDL audit logs will be lost. Contact technical support before modifying this parameter.
AZ	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. A DB instance can be deployed in one AZ or three AZs. Three AZs are recommended. To deploy a GaussDB instance in Intelligent EdgeCloud (IEC), select an IEC HomeZones for AZ. For assistance with this setup, submit a service ticket to contact Huawei engineers.
Time Zone	Select a time zone according to the region hosting your DB instance when you buy the instance.

Table 12-3 Specifications and storage

Parameter	Description
Instance Specifications	CPU and memory specifications of the instance. Different instance specifications support different numbers of database connections. For details, see Instance Specifications.

Parameter	Description
Storage Space	The allocated storage space contains necessary file system overhead required for inodes, reserved blocks, and database operations. The storage for a single shard starts from 40 GB and can be increased at a step of 4 GB.
Disk Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. Key Name: If disk encryption is enabled, you need to select or create a key. NOTE • When creating a pay-per-use instance with disk encryption enabled, ensure that your account has been granted the following action: kms:cmk:createGrant • When you create a yearly/monthly instance with disk encryption enabled, if the KMS key shared by RAM is used for the first time, create an agency as prompted. • When using the KMS key shared by RAM to create an instance, ensure that your account has also been granted the following actions: – iam:agencies:listAgencies – iam:agencies:createAgency – iam:roles:listRoles – iam:permissions:listRolesForAgency – iam:permissions:grantRoleToAgency – kms:cmk:createGrant If a shared KMS key is used, the corresponding CTS event is createGrant. Only the key owner can receive this event.

Table 12-4 Network and database configuration

Parameter	Description
VPC	A virtual network where your GaussDB instances are located. A VPC isolates networks for different workloads. You need to create or select the required VPC. If no VPC is available, GaussDB allocates a default VPC for you. After the GaussDB instance is created, the VPC cannot be changed. For details about how to create a VPC, see Creating a VPC. With VPC sharing, you can also use a VPC and subnet shared by another account. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). This allows for more efficient use of network resources and reduces O&M costs. For more information about VPC subnet sharing, see VPC Sharing in the <i>Virtual Private Cloud User Guide</i>.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within a specific AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets in which you plan to create GaussDB instances and cannot be disabled. GaussDB supports automatic IP address allocation during instance creation. If an IEC HomeZones is selected for AZ, create a subnet in that zone for this parameter. NOTE By default, a subnet supports up to 256 IP addresses. A distributed instance can require up to 1,286 IP addresses. You are advised to use a subnet that can provide 2,048 IP addresses.

Parameter	Description
Security Group	Controls access to and from an instance. When you buy an instance, the selected security group must meet the following requirements: - When buying a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6001, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound security group rules allow all outbound traffic. - When buying a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.) The security group enhances security by controlling access to GaussDB from other services. Ensure that the security group you selected allows your client to access the instance. If you do not need to specify a security group when creating an instance, you can submit a service ticket. If no security group is available, GaussDB allocates a default security group for you.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: 8000. The following ports are used by the system and cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.
Single Private IP Address	Specifies whether to enable the single private IP address policy. If this policy is enabled, only one private IP address is assigned to an instance and is bound to the primary node. The private IP address does not change after a primary/standby switchover. If this policy is disabled, each node is bound to a private IP address, and the private IP address changes after a primary/standby switchover. The constraints on the single private IP address policy are as follows: - This policy is only available for centralized instances of version V2.0-3.207 or later. - This policy is configurable only at the time of instance creation and cannot be changed afterward.

Table 12-5 Database configuration

Parameter	Description
Administrator	Database administrator. The default username is root .
Administrator Password	Enter a strong password and periodically change it to improve security, preventing security risks such as brute force cracking. The password must contain: • 8 to 32 characters. • At least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*_-=+?, Keep the password secure. Password retrieval is not supported.
Confirm Password	Enter the administrator password again.
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. After creating an instance, you can modify the parameter template.
Enterprise Project	If the instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the Enterprise Project Management console to create a project. For details, see Enterprise Management User Guide.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each instance can have up to 20 tags. If your organization has configured tag policies for GaussDB, add tags to instances based on the policies. If a tag does not comply with the policies, instance creation may fail. Contact your organization administrator to learn more about tag policies.

Step 7 View the restoration results.

A new instance that contains the same data as the backup is created. When the instance status changes from **Creating** to **Available**, the restoration is complete.

The new instance is independent of the original one.

----End

Restoring Data to the Original Instance

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

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

Step 3 In the navigation pane, choose **Backups**. On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the name of the target instance on the **Instances** page. In the navigation pane, choose **Backups**. On the **Full Backups** page, click the **Instance Backup** tab, and click **Restore** in the **Operation** column of the backup to be used for restoration.

Step 4 Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.

- The instance version and node configuration must be the same as those of the original instance when the backup was created.
- Restoring to the original DB instance will overwrite all data on it and cause the DB instance to be unavailable during the restoration.
- You are advised to manually back up data before the restoration.
- If you use a backup created before advanced compression is enabled to restore data to the original instance, you must enable this feature for the instance again.

Step 5 Confirm the backup task information and click **OK**.

Step 6 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. After the restoration is complete, an instance-level full backup will be automatically triggered.

After the restoration is complete, check whether the restored data is consistent with the time point to which the data is restored.

----End

Restoring Data to an Existing Instance

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

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

Step 3 In the navigation pane, choose **Backups**. On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

Alternatively, click the name of the target instance on the **Instances** page. In the navigation pane, choose **Backups**. On the **Full Backups** page, click the **Instance Backup** tab, and click **Restore** in the **Operation** column of the backup to be used for restoration.

Step 4 Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, select an existing instance, and click **Next**.

- The selected instance must have the same engine type, engine version, OS, resource type, topology, CPU architecture, case sensitivity of M-compatible table names (for a centralized instance), replica consistency (for a centralized instance), and transaction consistency (for a distributed instance) settings as the original instance had when it was backed up.
- The storage capacity of the selected instance must be at least equal to what the original instance had when it was backed up. (For an independently deployed distributed instance whose shards have different storage: Original instance storage = Maximum shard storage x Shard quantity; Target instance storage = Minimum shard storage x Shard quantity.)
- The specifications of the selected instance must be at least equal to what the original instance had when it was backed up.
- The selected instance cannot contain read replicas.
- Restoring data to an existing instance will overwrite all data on it and cause the instance to be unavailable during the restoration.
- Manually backing up data of the selected DB instance before the restoration.
- If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the new instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.
- An existing instance cannot be selected as the restoration target if it is currently undergoing a full or differential backup. If the backup is not required, you can stop it before proceeding with the restoration.
- If transparent data encryption (TDE) is enabled for the original instance, TDE must also be enabled for the selected target instance and the same key must be used.

Step 5 Confirm the backup task information and click **OK**.

Step 6 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. After the restoration is complete, an instance-level full backup will be automatically triggered.

----End

12.3 Restoring a GaussDB Database or Table Using a Backup File

Scenarios

You can use an instance-level automated or manual backup to restore data of specified databases or tables to the point in time when the backup was created. This operation restores data in specified databases or tables.

You can use a table-level automated or manual backup to restore data in tables to the point in time when the backup was created. This operation restores data only in specified tables.

Data can be restored to a new DB instance, an existing DB instance, or the original DB instance.

Precautions

- Keep your account balance above zero so that backups can be restored to a new instance.
- Only databases and tables in instances of version V2.0-3.200 or later can be restored. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- Restoration will fail if the instance is in the **Abnormal**, or **Storage full** state.
- Backup data cannot be restored across major versions. For example, backup data of an instance of version V2.0-3.200.x can only be restored to version V2.0-3.200.y.
- A maximum of 100 databases or tables can be restored at the same time. If there are more than 100 databases or tables, you are advised to use instance-level restoration.
- The table names in a given database and schema as well as the database names must be different before and after the restoration.
- Table-level restoration does not support column-store tables, user-defined tables, synonym tables, temporary tables (including global tables), unlogged tables, tables of private users, and tables of ledger databases.
- Currently, only heap data, index data, and auto-increment column data related to tables can be restored. Other data related to tables, such as foreign key information, triggers, and SQL throttling rules, cannot be restored.
- After table-level restoration, row-level access control and dynamic masking information will be lost.
- System databases (**postgres**, **template0**, **template1**, **templatem**, **templatea**, and **templatepdb**) and their tables cannot be selected for database- and table-level restoration.
- System schemas (for example, **public**) and their tables cannot be selected for database- and table-level restoration.
- In B-compatible mode, tables with the **on update timestamp** column do not support table-level physical restoration.
- M-compatible databases and their tables cannot be selected for database- and table-level restoration.
- Single-replica instances do not support database- and table-level backup and restoration.
- Snapshot-based backups cannot be used to restore databases or tables.
- If transparent data encryption (TDE) is enabled for an instance and the instance version is earlier than V2.0-8.200, database- and table-level restoration is not supported.
- Data cannot be restored to an existing instance for which a DR relationship has been established.
- Restoring a database or table using a backup file is only available to authorized users. To request this feature, [submit a service ticket](#).
- If parallel restoration is enabled, all primary and standby replicas will simultaneously download backup data from the storage server. Compared to

serial restoration, this process demands a multiple of the bandwidth from the storage server equal to the number of replicas in each shard. If your instance has more than five shards, consult O&M personnel about the available bandwidth of the OBS server before deciding whether to enable parallel restoration. This will prevent a decrease in restoration speed due to bandwidth exhaustion.

- Centralized instances support only parallel restoration.
- Parallel restoration cannot be enabled if the DB engine version is earlier than V2.0-1.4.
- In addition to table-level full backups and incremental backups, the system backs up incremental log files to ensure data consistency. It takes some time to back up and upload incremental log files (The time depends on the network and OBS traffic control). Note that the backup completion time does not represent the data consistency point that can be specified when this backup set is used to restore data. (Generally, the data consistency point is within several minutes before the backup completion time.) If you have strict requirements on data consistency after restoration, restore data to a specified point in time.

Restoring Data to a New Instance

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane, choose **Backups**. On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

- If the selected backup is an instance-level backup, you can restore data to specified databases or tables.
- If the selected backup is a table-level backup, you can only restore data to specified tables.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane on the left, choose **Backups**. Click **Instance Backup** or **Table Backup** under **Full Backups**. On the displayed page, locate the backup to be restored and click **Restore** in the **Operation** column.

Step 5 Set **Restoration Method** to **Create New Instance** and click **OK**.

When you attempt to restore data to tables, a maximum of 200 tables can be displayed for **Backup Tables** by default. If the required table is not displayed, click the + icon in the upper right corner to add a custom table. (You can also use this method to add required tables if the instance is faulty or deleted.)

- The new DB instance must have the same major version as the original instance when it was backed up. For example, backup data of an instance of version V2.0-3.200.x can only be restored to version V2.0-3.200.y.
- The storage space of the new instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.

- The administrator password needs to be reset.
- By default, the specifications of the new instance are the same as those of the original instance. You can configure higher specifications for the new instance. For instances of version V2.0-8.200 or later, an instance backup can be restored to an instance with lower specifications than the original instance. (Minimum specifications for the destination instance: 8 vCPUs | 32 GB)
- The backup media, CPU architecture, OS type and version, instance type (distributed or centralized), deployment model, replica consistency protocol, and transaction consistency settings of the new instance must be the same as those of the original instance to be restored.
- If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the new instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.
- Parameters of the original instance will not be automatically restored to the new instance. To use the original parameter settings, select the required parameter template for **Parameter Template** when creating an instance for restoring data to a new instance. After the new instance is created, click the instance name and manually change the parameter values on the **Parameters** page.

Step 6 On the displayed page, configure parameters of the new instance, select the table or database data to be restored, and click **Next**.

Table 12-6 Basic information

Parameter	Description
Billing Mode	GaussDB provides yearly/monthly billing and pay-per-use billing. • Yearly/Monthly: You pay upfront for the amount of time you expect to use the DB instance for. You will need to make sure you have a top-up account with a sufficient balance or have a valid payment method configured first. For distributed instances using the combined deployment model, yearly/monthly billing is only available to authorized users. To apply for the permissions needed, submit a service ticket. • Pay-per-use: You can start using the DB instance first and then pay as you go. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.
DB Instance Name	The instance name is case-sensitive, must start with a letter, and can contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Failover Priority	This parameter is only available for distributed instances. This parameter is only available for authorized users. To apply for the permissions needed, submit a service ticket. The default value is Reliability. For details about how to change the failover priority for an existing instance, see Changing Failover Priority. • Reliability: Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency. • Availability: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. If Availability is selected, exercise caution when modifying the following database parameters. For details about how to modify parameters, see Modifying Instance Parameters. – recovery_time_target: If the value of this parameter is changed, the instance will experience frequent forced failovers. Contact technical support before modifying this parameter. – audit_system_object: If the value of this parameter is changed, DDL audit logs will be lost. Contact technical support before modifying this parameter.
AZ	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. A DB instance can be deployed in one AZ or three AZs. Three AZs are recommended. To deploy a GaussDB instance in Intelligent EdgeCloud (IEC), select an IEC HomeZones for AZ. For assistance with this setup, submit a service ticket to contact Huawei engineers.
Time Zone	Select a time zone according to the region hosting your DB instance when you buy the instance.

Table 12-7 Specifications and storage

Parameter	Description
Instance Specifications	CPU and memory specifications of the instance. Different instance specifications support different numbers of database connections. For details, see Instance Specifications.

Parameter	Description
Storage Space	The allocated storage space contains necessary file system overhead required for inodes, reserved blocks, and database operations. The storage for a single shard starts from 40 GB and can be increased at a step of 4 GB.
Disk Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. Key Name: If disk encryption is enabled, you need to select or create a key. NOTE • When creating a pay-per-use instance with disk encryption enabled, ensure that your account has been granted the following action: kms:cmk:createGrant • When you create a yearly/monthly instance with disk encryption enabled, if the KMS key shared by RAM is used for the first time, create an agency as prompted. • When using the KMS key shared by RAM to create an instance, ensure that your account has also been granted the following actions: – iam:agencies:listAgencies – iam:agencies:createAgency – iam:roles:listRoles – iam:permissions:listRolesForAgency – iam:permissions:grantRoleToAgency – kms:cmk:createGrant If a shared KMS key is used, the corresponding CTS event is createGrant. Only the key owner can receive this event.

Table 12-8 Network and database configuration

Parameter	Description
VPC	A virtual network where your GaussDB instances are located. A VPC isolates networks for different workloads. You need to create or select the required VPC. If no VPC is available, GaussDB allocates a default VPC for you. After the GaussDB instance is created, the VPC cannot be changed. For details about how to create a VPC, see Creating a VPC. With VPC sharing, you can also use a VPC and subnet shared by another account. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). This allows for more efficient use of network resources and reduces O&M costs. For more information about VPC subnet sharing, see VPC Sharing in the <i>Virtual Private Cloud User Guide</i>.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within a specific AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets in which you plan to create GaussDB instances and cannot be disabled. GaussDB supports automatic IP address allocation during instance creation. If an IEC HomeZones is selected for AZ, create a subnet in that zone for this parameter. NOTE By default, a subnet supports up to 256 IP addresses. A distributed instance can require up to 1,286 IP addresses. You are advised to use a subnet that can provide 2,048 IP addresses.

Parameter	Description
Security Group	Controls access to and from an instance. When you buy an instance, the selected security group must meet the following requirements: - When buying a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6001, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound security group rules allow all outbound traffic. - When buying a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.) The security group enhances security by controlling access to GaussDB from other services. Ensure that the security group you selected allows your client to access the instance. If you do not need to specify a security group when creating an instance, you can submit a service ticket. If no security group is available, GaussDB allocates a default security group for you.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: 8000. The following ports are used by the system and cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.
Single Private IP Address	Specifies whether to enable the single private IP address policy. If this policy is enabled, only one private IP address is assigned to an instance and is bound to the primary node. The private IP address does not change after a primary/standby switchover. If this policy is disabled, each node is bound to a private IP address, and the private IP address changes after a primary/standby switchover. The constraints on the single private IP address policy are as follows: - This policy is only available for centralized instances of version V2.0-3.207 or later. - This policy is configurable only at the time of instance creation and cannot be changed afterward.

Table 12-9 Database configuration

Parameter	Description
Administrator	Database administrator. The default username is root .
Administrator Password	Enter a strong password and periodically change it to improve security, preventing security risks such as brute force cracking. The password must contain: • 8 to 32 characters. • At least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*_-=+?, Keep the password secure. Password retrieval is not supported.
Confirm Password	Enter the administrator password again.
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. After creating an instance, you can modify the parameter template.
Enterprise Project	If the instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the Enterprise Project Management console to create a project. For details, see Enterprise Management User Guide.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each instance can have up to 20 tags. If your organization has configured tag policies for GaussDB, add tags to instances based on the policies. If a tag does not comply with the policies, instance creation may fail. Contact your organization administrator to learn more about tag policies.

Step 7 View the restoration results.

A new instance that contains the same databases or tables as the backup is created. When the instance status changes from **Creating** to **Available**, the restoration is complete.

The new instance is independent of the original one.

----End

Restoring Data to the Original Instance

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane, choose **Backups**. On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

- If the selected backup is an instance-level backup, you can restore data to specified databases or tables.
- If the selected backup is a table-level backup, you can only restore data to specified tables.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane on the left, choose **Backups**. Click **Instance Backup** or **Table Backup** under **Full Backups**. On the displayed page, locate the backup to be restored and click **Restore** in the **Operation** column.

Step 5 Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.

When you attempt to restore data to tables, a maximum of 200 tables can be displayed for **Backup Tables** by default. If the required table is not displayed, click the + icon in the upper right corner to add a custom table. (You can also use this method to add required tables if the instance is faulty or deleted.)

- The instance version and node configuration must be the same as those of the original instance when the backup was created.
- Databases and tables will be created on the target DB instance. During the restoration, the source database can be used properly.
- You are advised to manually back up data before the restoration.
- After a database (for example, **db1**) is restored to the original instance, you need to wait for the DB instance to automatically perform a full or differential backup before restoring the data of **db1** using other restoration processes. The time to wait depends on the backup policy. During point-in-time restoration, if you select a time point that is later than when the current database-level restoration is complete but earlier than when the next backup is performed, **db1** cannot be restored.

Step 6 On the **Restore DB Instance** page, select the data to be restored and click **Submit**.

Step 7 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

Restoring Data to an Existing Instance

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane, choose **Backups**. On the **Backups** page, locate the backup to be restored and click **Restore** in the **Operation** column.

- If the selected backup is an instance-level backup, you can restore data to specified databases or tables.
- If the selected backup is a table-level backup, you can only restore data to specified tables.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane on the left, choose **Backups**. Click **Instance Backup** or **Table Backup** under **Full Backups**. On the displayed page, locate the backup to be restored and click **Restore** in the **Operation** column.

Step 5 Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, select an existing instance, and click **Next**.

When you attempt to restore data to tables, a maximum of 200 tables can be displayed for **Backup Tables** by default. If the required table is not displayed, click the + icon in the upper right corner to add a custom table. (You can also use this method to add required tables if the instance is faulty or deleted.)

- Databases and tables will be created on the target instance. During the restoration, the databases on the instance can be used properly.
- Manually backing up data of the selected DB instance before the restoration.
- The selected instance must have the same engine type, engine version, OS, resource type, topology, CPU architecture, case sensitivity of M-compatible table names (for a centralized instance), replica consistency (for a centralized instance), and transaction consistency (for a distributed instance) settings as the original instance had when it was backed up.
- The storage capacity of the selected instance must be at least equal to what the original instance had when it was backed up. (For an independently deployed distributed instance whose shards have different storage: Original instance storage = Maximum shard storage x Shard quantity; Target instance storage = Minimum shard storage x Shard quantity.)
- The specifications of the selected instance must be at least equal to what the original instance had when it was backed up.
- The selected instance cannot contain read replicas.
- An existing instance cannot be selected as the restoration target if it is currently undergoing a full or differential backup. If the backup is not required, you can stop it before proceeding with the restoration.

Step 6 Confirm the backup task information and click **OK**.

Step 7 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

12.4 Restoring a GaussDB Instance to a Specific Point in Time

Scenarios

You can use an instance-level automated backup to restore a GaussDB instance to a specified point in time.

You can restore backup data to the original GaussDB instance, an existing instance, or a new one.

Precautions

- Only instances of version V2.0-2.1 or later can be restored to any point in time. Single-replica instances are not supported. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- Data from the time period of the following operations cannot be restored: nodes are being added to an instance, the instance version is being upgraded, or the instance data is being restored.
- If an instance is faulty or a CN is removed, archive logs cannot be generated, and data from the affected time period cannot be restored.
- If you restore backup data to a new DB instance:
 - The DB engine and major version are the same as those of the original DB instance and cannot be changed.
 - The administrator password needs to be reset.
- If you restore backup data to the original DB instance, data on the original instance will be overwritten and the original DB instance will be unavailable during the restoration. Additionally, log archiving stops. After the restoration is complete, the **Confirm Data Integrity** button is displayed. Before clicking **Confirm Data Integrity**, you can restore data for multiple times. Once data integrity has been confirmed, any logs archived after the point in time data was restored from will be lost, but normal log archiving will be restored.
- When a DB instance is deleted, all archive logs are deleted by default and cannot be retained. After an instance is deleted, it cannot be rebuilt or restored to any point in time.
- If parallel restoration is enabled, all primary and standby replicas will simultaneously download backup data from the storage server. Compared to serial restoration, this process demands a multiple of the bandwidth from the storage server equal to the number of replicas in each shard. If your instance has more than five shards, consult O&M personnel about the available bandwidth of the OBS server before deciding whether to enable parallel restoration. This will prevent a decrease in restoration speed due to bandwidth exhaustion.
- Centralized instances support only parallel restoration.
- Parallel restoration cannot be enabled if the DB engine version is earlier than V2.0-1.4.

Restoring Data to a New Instance

Step 1 [Log in to the management console.](#)

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

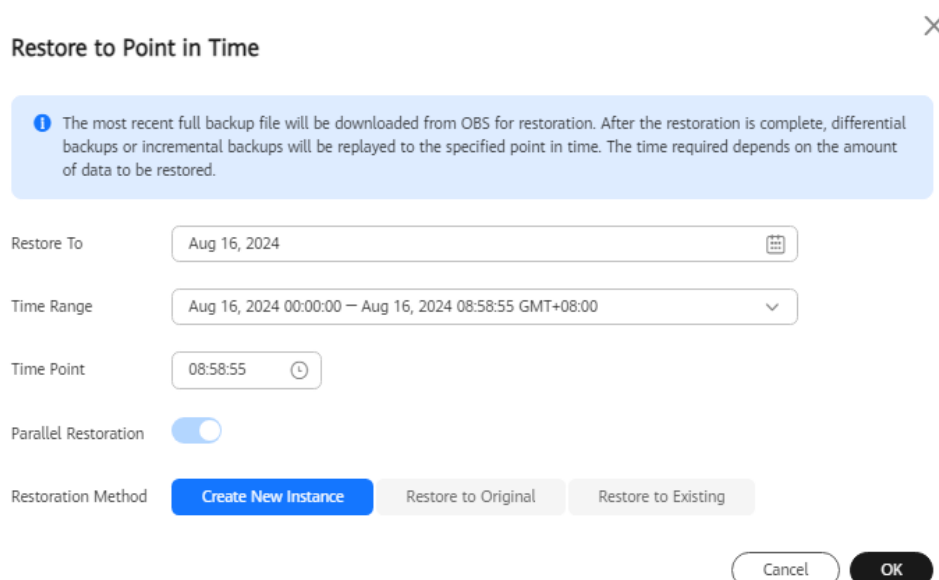
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Backups**. On the displayed page, click **Restore to Point in Time**.

Step 6 Set **Restoration Method** to **Create New Instance** and click **OK**.

Figure 12-1 Restoring data to a specified point in time



Restore to Point in Time

ⓘ The most recent full backup file will be downloaded from OBS for restoration. After the restoration is complete, differential backups or incremental backups will be replayed to the specified point in time. The time required depends on the amount of data to be restored.

Restore To: Aug 16, 2024

Time Range: Aug 16, 2024 00:00:00 – Aug 16, 2024 08:58:55 GMT+08:00

Time Point: 08:58:55

Parallel Restoration: ☒

Restoration Method: **Create New Instance** | Restore to Original | Restore to Existing

Cancel OK

- The new DB instance must have the same major version as the original instance when it was backed up. For example, backup data can only be restored from version V2.0-1.4.x to version V2.0-1.4.y.
- The storage space of the new instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.
- The administrator password needs to be reset.
- The backup media, CPU architecture, OS type and version, instance type (distributed or centralized), deployment model, replica consistency protocol, and transaction consistency settings of the new instance must be the same as those of the original instance to be restored.
- By default, the instance specifications of the new instance are the same as those of the original instance. To change the instance specifications, ensure that the instance specifications of the new instance are at least those of the original instance.

- If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the new instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.

Step 7 On the displayed page, configure parameters of the new instance and click **Next**.

Table 12-10 Basic information

Parameter	Description
Billing Mode	GaussDB provides yearly/monthly billing and pay-per-use billing. • Yearly/Monthly: You pay upfront for the amount of time you expect to use the DB instance for. You will need to make sure you have a top-up account with a sufficient balance or have a valid payment method configured first. For distributed instances using the combined deployment model, yearly/monthly billing is only available to authorized users. To apply for the permissions needed, submit a service ticket. • Pay-per-use: You can start using the DB instance first and then pay as you go. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.
DB Instance Name	The instance name is case-sensitive, must start with a letter, and can contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Failover Priority	This parameter is only available for distributed instances. This parameter is only available for authorized users. To apply for the permissions needed, submit a service ticket. The default value is Reliability. For details about how to change the failover priority for an existing instance, see Changing Failover Priority. • Reliability: Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency. • Availability: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. If Availability is selected, exercise caution when modifying the following database parameters. For details about how to modify parameters, see Modifying Instance Parameters. – recovery_time_target: If the value of this parameter is changed, the instance will experience frequent forced failovers. Contact technical support before modifying this parameter. – audit_system_object: If the value of this parameter is changed, DDL audit logs will be lost. Contact technical support before modifying this parameter.
AZ	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. A DB instance can be deployed in one AZ or three AZs. Three AZs are recommended. To deploy a GaussDB instance in Intelligent EdgeCloud (IEC), select an IEC HomeZones for AZ. For assistance with this setup, submit a service ticket to contact Huawei engineers.
Time Zone	Select a time zone according to the region hosting your DB instance when you buy the instance.

Table 12-11 Specifications and storage

Parameter	Description
Instance Specifications	CPU and memory specifications of the instance. Different instance specifications support different numbers of database connections. For details, see Instance Specifications.

Parameter	Description
Storage Space	The allocated storage space contains necessary file system overhead required for inodes, reserved blocks, and database operations. The storage for a single shard starts from 40 GB and can be increased at a step of 4 GB.
Disk Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. Key Name: If disk encryption is enabled, you need to select or create a key. NOTE • When creating a pay-per-use instance with disk encryption enabled, ensure that your account has been granted the following action: kms:cmk:createGrant • When you create a yearly/monthly instance with disk encryption enabled, if the KMS key shared by RAM is used for the first time, create an agency as prompted. • When using the KMS key shared by RAM to create an instance, ensure that your account has also been granted the following actions: – iam:agencies:listAgencies – iam:agencies:createAgency – iam:roles:listRoles – iam:permissions:listRolesForAgency – iam:permissions:grantRoleToAgency – kms:cmk:createGrant If a shared KMS key is used, the corresponding CTS event is createGrant. Only the key owner can receive this event.

Table 12-12 Network and database configuration

Parameter	Description
VPC	A virtual network where your GaussDB instances are located. A VPC isolates networks for different workloads. You need to create or select the required VPC. If no VPC is available, GaussDB allocates a default VPC for you. After the GaussDB instance is created, the VPC cannot be changed. For details about how to create a VPC, see Creating a VPC. With VPC sharing, you can also use a VPC and subnet shared by another account. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). This allows for more efficient use of network resources and reduces O&M costs. For more information about VPC subnet sharing, see VPC Sharing in the <i>Virtual Private Cloud User Guide</i>.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within a specific AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets in which you plan to create GaussDB instances and cannot be disabled. GaussDB supports automatic IP address allocation during instance creation. If an IEC HomeZones is selected for AZ, create a subnet in that zone for this parameter. NOTE By default, a subnet supports up to 256 IP addresses. A distributed instance can require up to 1,286 IP addresses. You are advised to use a subnet that can provide 2,048 IP addresses.

Parameter	Description
Security Group	Controls access to and from an instance. When you buy an instance, the selected security group must meet the following requirements: - When buying a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6001, 6500, and $\langle database\_port \rangle - (\langle database\_port \rangle + 100)$. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound security group rules allow all outbound traffic. - When buying a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and $\langle database\_port \rangle - (\langle database\_port \rangle + 100)$. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.) The security group enhances security by controlling access to GaussDB from other services. Ensure that the security group you selected allows your client to access the instance. If you do not need to specify a security group when creating an instance, you can submit a service ticket. If no security group is available, GaussDB allocates a default security group for you.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: 8000. The following ports are used by the system and cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.
Single Private IP Address	Specifies whether to enable the single private IP address policy. If this policy is enabled, only one private IP address is assigned to an instance and is bound to the primary node. The private IP address does not change after a primary/standby switchover. If this policy is disabled, each node is bound to a private IP address, and the private IP address changes after a primary/standby switchover. The constraints on the single private IP address policy are as follows: - This policy is only available for centralized instances of version V2.0-3.207 or later. - This policy is configurable only at the time of instance creation and cannot be changed afterward.

Table 12-13 Database configuration

Parameter	Description
Administrator	Database administrator. The default username is root .
Administrator Password	Enter a strong password and periodically change it to improve security, preventing security risks such as brute force cracking. The password must contain: • 8 to 32 characters. • At least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*_-=+?, Keep the password secure. Password retrieval is not supported.
Confirm Password	Enter the administrator password again.
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. After creating an instance, you can modify the parameter template.
Enterprise Project	If the instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the Enterprise Project Management console to create a project. For details, see Enterprise Management User Guide.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each instance can have up to 20 tags. If your organization has configured tag policies for GaussDB, add tags to instances based on the policies. If a tag does not comply with the policies, instance creation may fail. Contact your organization administrator to learn more about tag policies.

Step 8 View the restoration results.

A new instance that contains the same data as the backup is created. When the instance status changes from **Creating** to **Available**, the restoration is complete.

The new instance is independent of the original one.

----End

Restoring Data to the Original Instance

- Step 1** Click  in the upper left corner and select a region and project.
- Step 2** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 3** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 4** In the navigation pane on the left, choose **Backups**. On the displayed page, click **Restore to Point in Time**.
- Step 5** Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.
- The instance version and node configuration must be the same as those of the original instance when the backup was created.
 - Restoring to the original DB instance will overwrite all data on it and cause the DB instance to be unavailable during the restoration.
 - You are advised to manually back up data before the restoration.
 - If you use a backup created before advanced compression is enabled to restore data to the original instance, you must enable this feature for the instance again.
- Step 6** Confirm the backup task information and click **OK**.
- Step 7** View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. After the restoration is complete, a full backup will be automatically triggered.

After the restoration is complete, check whether the restored data is consistent with the time point to which the data is restored.

----End

Restoring Data to an Existing Instance

- Step 1** Click  in the upper left corner and select a region and project.
- Step 2** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 3** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 4** In the navigation pane on the left, choose **Backups**. On the displayed page, click **Restore to Point in Time**.
- Step 5** Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, select an existing instance, and click **Next**.
- Restoring data to an existing instance will overwrite all data on it and cause the instance to be unavailable during the restoration.
 - Manually backing up data of the selected DB instance before the restoration.

- If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the target instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.
- The selected instance must have the same engine type, engine version, OS, resource type, topology, CPU architecture, case sensitivity of M-compatible table names (for a centralized instance), replica consistency (for a centralized instance), and transaction consistency (for a distributed instance) settings as the original instance had when it was backed up.
- The storage capacity of the selected instance must be at least equal to what the original instance had when it was backed up. (For an independently deployed distributed instance whose shards have different storage: Original instance storage = Maximum shard storage x Shard quantity; Target instance storage = Minimum shard storage x Shard quantity.)
- The specifications of the selected instance must be at least equal to what the original instance had when it was backed up.
- The selected instance cannot contain read replicas.
- An existing instance cannot be selected as the restoration target if it is currently undergoing a full or differential backup. If the backup is not required, you can stop it before proceeding with the restoration.
- If transparent data encryption (TDE) is enabled for the original instance, TDE must also be enabled for the selected target instance and the same key must be used.

Step 6 Confirm the backup task information and click **OK**.

Step 7 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. After the restoration is complete, a full backup will be automatically triggered.

----End

12.5 Restoring a GaussDB Database or Table to a Specific Point in Time

Scenarios

You can use an instance-level automated backup to restore data of specified databases or tables to a specified point in time.

You can use a table-level automated backup to restore data in tables to a specified point in time.

You can restore backup data to the original GaussDB instance, an existing instance, or a new one.

Precautions

- To use this function, your DB instance cannot be a single-replica instance and its version must be V2.0-3.200 or later.

- Data from the time period of the following operations cannot be restored: nodes are being added to an instance, the instance version is being upgraded, or the instance data is being restored.
- If an instance is faulty or a CN is removed, archive logs cannot be generated, and data from the affected time period cannot be restored.
- If you restore backup data to a new DB instance:
 - The DB engine and major version are the same as those of the original DB instance and cannot be changed.
 - The administrator password needs to be reset.
- If you restore backup data to the original DB instance, new databases or tables are created on the original instance and the original databases are still available during the restoration. The archive logs are normal.
- When a DB instance is deleted, all archive logs are deleted by default and cannot be retained. After an instance is deleted, it cannot be rebuilt or restored to any point in time.
- The table names in a given database and schema as well as the database names must be different before and after the restoration. If they are same, you need to rename the tables and databases that the data is restored to.
- If **ALTER DATABASE SET TABLESPACE** and **ALTER TABLE SET TABLESPACE** are executed in the original instance, table-level data can be restored only after a differential backup or full backup is triggered in the original instance.
- A maximum of 100 databases or tables can be restored at the same time. If there are more than 100 databases or tables, you are advised to use instance-level restoration.
- Table-level restoration does not support column-store tables, user-defined tables, synonym tables, temporary tables (including global tables), unlogged tables, tables of private users, and tables of ledger databases.
- Currently, only heap data, index data, and auto-increment column data related to tables can be restored. Other data related to tables, such as foreign key information, triggers, and SQL throttling rules, cannot be restored.
- After table-level restoration, row-level access control and dynamic masking information will be lost.
- System databases (**postgres**, **template0**, **template1**, **templatem**, **templatea**, and **templatepdb**) and their tables cannot be selected for database- and table-level restoration.
- System schemas (for example, **public**) and their tables cannot be selected for database- and table-level restoration.
- In B-compatible mode, tables with the **on update timestamp** column do not support table-level physical restoration.
- M-compatible databases and their tables cannot be selected for database- and table-level restoration.
- Single-replica instances do not support database- and table-level backup and restoration.
- PITR based on snapshots does not support database- or table-level restoration.
- If transparent data encryption (TDE) is enabled for an instance and the instance version is earlier than V2.0-8.200, database- and table-level restoration is not supported.

- If parallel restoration is enabled, all primary and standby replicas will simultaneously download backup data from the storage server. Compared to serial restoration, this process demands a multiple of the bandwidth from the storage server equal to the number of replicas in each shard. If your instance has more than five shards, consult O&M personnel about the available bandwidth of the OBS server before deciding whether to enable parallel restoration. This will prevent a decrease in restoration speed due to bandwidth exhaustion.
- Centralized instances support only parallel restoration.
- Parallel restoration cannot be enabled if the DB engine version is earlier than V2.0-1.4.
- Restoring data to existing instances is only available for authorized users. To apply for the permissions needed, [submit a service ticket](#).

Restoring Data to a New Instance

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Backups**. On the displayed page, click the **Instance Backup** or **Table Backup** tab.

- On the **Instance Backup** tab, click **Restore to Point in Time** to restore data of specified databases or tables.
- On the **Table Backup** tab, click **Restore to Point in Time** to restore data of specified tables.

Step 6 Set **Restoration Method** to **Create New Instance** and click **OK**.

- The new DB instance must have the same major version as the original instance when it was backed up. For example, backup data of an instance of version V2.0-3.200.x can only be restored to version V2.0-3.200.y.
- The storage space of the new instance is the same as that of the original DB instance by default and the new instance must be at least as large as the original DB instance.
- The administrator password needs to be reset.
- The backup media, CPU architecture, OS type and version, instance type (distributed or centralized), deployment model, replica consistency protocol, and transaction consistency settings of the new instance must be the same as those of the original instance to be restored.
- If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the new instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.
- By default, the instance specifications of the new instance are the same as those of the original instance. To change the instance specifications, ensure

that the instance specifications of the new instance are at least those of the original instance.

- Parameters of the original instance will not be automatically restored to the new instance. To use the original parameter settings, select the required parameter template for **Parameter Template** when creating an instance for restoring data to a new instance. After the new instance is created, click the instance name and manually change the parameter values on the **Parameters** page.

Step 7 On the displayed page, configure parameters of the new instance, select the table or database data to be restored, and click **Next**.

Table 12-14 Basic information

Parameter	Description
Billing Mode	GaussDB provides yearly/monthly billing and pay-per-use billing. Yearly/Monthly: You pay upfront for the amount of time you expect to use the DB instance for. You will need to make sure you have a top-up account with a sufficient balance or have a valid payment method configured first. For distributed instances using the combined deployment model, yearly/monthly billing is only available to authorized users. To apply for the permissions needed, submit a service ticket. Pay-per-use: You can start using the DB instance first and then pay as you go. Pricing is listed on a per-hour basis, but bills are calculated based on the actual usage duration.
DB Instance Name	The instance name is case-sensitive, must start with a letter, and can contain 4 to 64 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Description
Failover Priority	This parameter is only available for distributed instances. This parameter is only available for authorized users. To apply for the permissions needed, submit a service ticket. The default value is Reliability. For details about how to change the failover priority for an existing instance, see Changing Failover Priority. • Reliability: Data consistency is prioritized during a failover. This is recommended for applications whose highest priority is data consistency. • Availability: Database availability is given priority during a failover. This is recommended for applications that require their databases to provide uninterrupted online services. If Availability is selected, exercise caution when modifying the following database parameters. For details about how to modify parameters, see Modifying Instance Parameters. – recovery_time_target: If the value of this parameter is changed, the instance will experience frequent forced failovers. Contact technical support before modifying this parameter. – audit_system_object: If the value of this parameter is changed, DDL audit logs will be lost. Contact technical support before modifying this parameter.
AZ	An AZ is a physical region where resources have their own independent power supply and networks. AZs are physically isolated but interconnected through an internal network. A DB instance can be deployed in one AZ or three AZs. Three AZs are recommended. To deploy a GaussDB instance in Intelligent EdgeCloud (IEC), select an IEC HomeZones for AZ. For assistance with this setup, submit a service ticket to contact Huawei engineers.
Time Zone	Select a time zone according to the region hosting your DB instance when you buy the instance.

Table 12-15 Specifications and storage

Parameter	Description
Instance Specifications	CPU and memory specifications of the instance. Different instance specifications support different numbers of database connections. For details, see Instance Specifications.

Parameter	Description
Storage Space	The allocated storage space contains necessary file system overhead required for inodes, reserved blocks, and database operations. The storage for a single shard starts from 40 GB and can be increased at a step of 4 GB.
Disk Encryption	• Disable: Encryption is disabled. • Enable: Encryption is enabled, which improves data security but affects system performance. Key Name: If disk encryption is enabled, you need to select or create a key. NOTE • When creating a pay-per-use instance with disk encryption enabled, ensure that your account has been granted the following action: kms:cmk:createGrant • When you create a yearly/monthly instance with disk encryption enabled, if the KMS key shared by RAM is used for the first time, create an agency as prompted. • When using the KMS key shared by RAM to create an instance, ensure that your account has also been granted the following actions: – iam:agencies:listAgencies – iam:agencies:createAgency – iam:roles:listRoles – iam:permissions:listRolesForAgency – iam:permissions:grantRoleToAgency – kms:cmk:createGrant If a shared KMS key is used, the corresponding CTS event is createGrant. Only the key owner can receive this event.

Table 12-16 Network and database configuration

Parameter	Description
VPC	A virtual network where your GaussDB instances are located. A VPC isolates networks for different workloads. You need to create or select the required VPC. If no VPC is available, GaussDB allocates a default VPC for you. After the GaussDB instance is created, the VPC cannot be changed. For details about how to create a VPC, see Creating a VPC. With VPC sharing, you can also use a VPC and subnet shared by another account. VPC owners can share the subnets in a VPC with one or multiple accounts through Resource Access Manager (RAM). This allows for more efficient use of network resources and reduces O&M costs. For more information about VPC subnet sharing, see VPC Sharing in the <i>Virtual Private Cloud User Guide</i>.
Subnet	A subnet provides dedicated network resources that are logically isolated from other networks for network security. Subnets take effect only within a specific AZ. Dynamic Host Configuration Protocol (DHCP) is enabled by default for subnets in which you plan to create GaussDB instances and cannot be disabled. GaussDB supports automatic IP address allocation during instance creation. If an IEC HomeZones is selected for AZ, create a subnet in that zone for this parameter. NOTE By default, a subnet supports up to 256 IP addresses. A distributed instance can require up to 1,286 IP addresses. You are advised to use a subnet that can provide 2,048 IP addresses.

Parameter	Description
Security Group	Controls access to and from an instance. When you buy an instance, the selected security group must meet the following requirements: - When buying a distributed instance, ensure that the TCP ports in the inbound rule include the following: 40000-60480, 20050, 5000-5001, 2379-2380, 6000, 6001, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the security group must contain ports 8000 to 8100.) Additionally, ensure that the outbound security group rules allow all outbound traffic. - When buying a centralized instance, ensure that the TCP ports in the inbound rule include the following: 20050, 5000-5001, 2379-2380, 6000, 6500, and <code><database_port>-(<database_port> + 100)</code>. (For example, if the database port is 8000, the TCP ports for the security group must include 8000-8100.) The security group enhances security by controlling access to GaussDB from other services. Ensure that the security group you selected allows your client to access the instance. If you do not need to specify a security group when creating an instance, you can submit a service ticket. If no security group is available, GaussDB allocates a default security group for you.
Database Port	The port is used by applications to access the database. Value range: 1024 to 39989. Default value: 8000. The following ports are used by the system and cannot be used: 2378 to 2380, 2400, 4999 to 5001, 5100, 5500, 5999 to 6001, 6009 to 6010, 6500, 8015, 8097, 8098, 8181, 9090, 9100, 9180, 9187, 9200, 12016, 12017, 20049, 20050, 21731, 21732, 32122 to 32126, and 39001.
Single Private IP Address	Specifies whether to enable the single private IP address policy. If this policy is enabled, only one private IP address is assigned to an instance and is bound to the primary node. The private IP address does not change after a primary/standby switchover. If this policy is disabled, each node is bound to a private IP address, and the private IP address changes after a primary/standby switchover. The constraints on the single private IP address policy are as follows: - This policy is only available for centralized instances of version V2.0-3.207 or later. - This policy is configurable only at the time of instance creation and cannot be changed afterward.

Table 12-17 Restoration scenarios

Scenario	Description
Restoring tables	• If you use an instance-level backup to restore data to tables, you can select Time-specific Tables or Recent Tables for the Tables field. – Operations such as table creation may have been performed on the instance during the period from the time when the backup was created to the current time. Therefore, the tables listed when Time-specific Tables is selected may be different from those listed when Recent Tables is selected. If you attempt to restore a table in Recent Tables to a specified time point, the restoration may fail. ▪ Time-specific Tables indicates the table information stored when the backup was created. ▪ Recent Tables indicates the most recent table information of the current instance. – If some tables have been deleted from the instance, specific tables may not be available regardless of whether Time-specific Tables or Recent Tables is selected. If you are sure that a particular table is available when the backup was created, click the + icon in the upper right corner to add the table. • If you use a table-level backup to restore data to tables, a maximum of 200 tables can be displayed for Backup Tables by default. If the required table is not displayed, click the + icon in the upper right corner to add a custom table. (You can also use this method to add required tables if the instance is faulty or deleted.)
Restoring databases	If you use an instance-level backup to restore data to databases, you can select Time-specific Databases or Recent Databases for the Databases field.

Table 12-18 Database configuration

Parameter	Description
Administrator	Database administrator. The default username is root .

Parameter	Description
Administrator Password	Enter a strong password and periodically change it to improve security, preventing security risks such as brute force cracking. The password must contain: • 8 to 32 characters. • At least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*_-=+?, Keep the password secure. Password retrieval is not supported.
Confirm Password	Enter the administrator password again.
Parameter Template	A template of parameters for creating an instance. The template contains engine configuration values that are applied to one or more instances. After creating an instance, you can modify the parameter template.
Enterprise Project	If the instance has been associated with an enterprise project, select the target project from the Enterprise Project drop-down list. You can also go to the Enterprise Project Management console to create a project. For details, see Enterprise Management User Guide.
Tag	This parameter is optional. Adding tags helps you better identify and manage your DB instances. Each instance can have up to 20 tags. If your organization has configured tag policies for GaussDB, add tags to instances based on the policies. If a tag does not comply with the policies, instance creation may fail. Contact your organization administrator to learn more about tag policies.

Step 8 View the restoration results.

A new instance that contains the same data as the backup is created. When the instance status changes from **Creating** to **Available**, the restoration is complete.

The new instance is independent of the original one.

----End

Restoring Data to the Original Instance

Step 1 [Log in to the management console](#).

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

- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane, choose **Backups**. On the displayed page, click the **Instance Backup** or **Table Backup** tab.
- On the **Instance Backup** tab, click **Restore to Point in Time** to restore data of specified databases or tables.
 - On the **Table Backup** tab, click **Restore to Point in Time** to restore data of specified tables.
- Step 6** Set **Restoration Method** to **Restore to Original**, select the confirmation check box, and click **Next**.
- The instance version and node configuration must be the same as those of the original instance when the backup was created.
 - New databases or tables will be created on the original instance. The databases to be restored are available during the restoration.
 - After a database (for example, **db1**) is restored to the original instance, you need to wait for the DB instance to automatically perform a full or differential backup before restoring the data of **db1** using other restoration processes. The time to wait depends on the backup policy. During point-in-time restoration, if you select a time point that is later than when the current database-level restoration is complete but earlier than when the next backup is performed, **db1** cannot be restored.
 - You are advised to manually back up data before the restoration.
- Step 7** On the page that is displayed, select the data to be restored and click **Submit**.

Table 12-19 Restoration scenarios

Scenario	Description
Restoring tables	• If you use an instance-level backup to restore data to tables, you can select Time-specific Tables or Recent Tables for the Tables field. – Operations such as table creation may have been performed on the instance during the period from the time when the backup was created to the current time. Therefore, the tables listed when Time-specific Tables is selected may be different from those listed when Recent Tables is selected. If you attempt to restore a table in Recent Tables to a specified time point, the restoration may fail. ▪ Time-specific Tables indicates the table information stored when the backup was created. ▪ Recent Tables indicates the most recent table information of the current instance. – If some tables have been deleted from the instance, specific tables may not be available regardless of whether Time-specific Tables or Recent Tables is selected. If you are sure that a particular table is available when the backup was created, click the + icon in the upper right corner to add the table. • If you use a table-level backup to restore data to tables, a maximum of 200 tables can be displayed for Backup Tables by default. If the required table is not displayed, click the + icon in the upper right corner to add a custom table. (You can also use this method to add required tables if the instance is faulty or deleted.)
Restoring databases	If you use an instance-level backup to restore data to databases, you can select Time-specific Databases or Recent Databases for the Databases field.

Step 8 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete.

----End

Restoring Data to an Existing Instance

Step 1 [Log in to the management console.](#)

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

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

- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane, choose **Backups**. On the displayed page, click the **Instance Backup** or **Table Backup** tab.
- On the **Instance Backup** tab, click **Restore to Point in Time** to restore data of specified databases or tables.
 - On the **Table Backup** tab, click **Restore to Point in Time** to restore data of specified tables.
- Step 6** Set **Restoration Method** to **Restore to Existing**, select the confirmation check box, select an existing instance, and click **Next**.
- Databases and tables will be created on the target instance. During the restoration, the databases on the instance can be used properly.
 - You are advised to manually back up data of the selected instance before the restoration.
 - The selected instance must have the same engine type, engine version, OS, resource type, topology, CPU architecture, case sensitivity of M-compatible table names (for a centralized instance), replica consistency (for a centralized instance), and transaction consistency (for a distributed instance) settings as the original instance had when it was backed up.
 - The storage capacity of the selected instance must be at least equal to what the original instance had when it was backed up. (For an independently deployed distributed instance whose shards have different storage: Original instance storage = Maximum shard storage x Shard quantity; Target instance storage = Minimum shard storage x Shard quantity.)
 - The specifications of the selected instance must be at least equal to what the original instance had when it was backed up.
 - The selected instance cannot contain read replicas.
 - An existing instance cannot be selected as the restoration target if it is currently undergoing a full or differential backup. If the backup is not required, you can stop it before proceeding with the restoration.
 - If there are M-compatible databases in the instance to be restored, the **Table Names in M-compatible Databases** parameter of the new instance must be set to the same value as that of the original instance. The default value is **Case-sensitive**.
- Step 7** On the page that is displayed, select the data to be restored and click **Submit**.

Table 12-20 Restoration scenarios

Scenario	Description
Restoring tables	• If you use an instance-level backup to restore data to tables, you can select Time-specific Tables or Recent Tables for the Tables field. – Operations such as table creation may have been performed on the instance during the period from the time when the backup was created to the current time. Therefore, the tables listed when Time-specific Tables is selected may be different from those listed when Recent Tables is selected. If you attempt to restore a table in Recent Tables to a specified time point, the restoration may fail. ▪ Time-specific Tables indicates the table information stored when the backup was created. ▪ Recent Tables indicates the most recent table information of the current instance. – If some tables have been deleted from the instance, specific tables may not be available regardless of whether Time-specific Tables or Recent Tables is selected. If you are sure that a particular table is available when the backup was created, click the + icon in the upper right corner to add the table. • If you use a table-level backup to restore data to tables, a maximum of 200 tables can be displayed for Backup Tables by default. If the required table is not displayed, click the + icon in the upper right corner to add a custom table. (You can also use this method to add required tables if the instance is faulty or deleted.)
Restoring databases	If you use an instance-level backup to restore data to databases, you can select Time-specific Databases or Recent Databases for the Databases field.

Step 8 View the restoration results.

On the **Instances** page, when the instance status changes from **Restoring** to **Available**, the restoration is complete. After the restoration is complete, an instance-level full backup will be automatically triggered.

----End

13 Account and Network Security

13.1 Resetting the Administrator Password of a GaussDB Instance

Scenarios

If you forget the password of your **root** account when using GaussDB, you can reset the password.

Precautions

- If the password you provide is regarded as a weak password by the system, you will be prompted to enter a stronger password.
- If the DB instance is abnormal, the administrator password cannot be reset.
- The volume of data being processed by the instance determines how long it takes for the new password to take effect.
- To prevent brute force cracking and ensure system security, change your password periodically.
- You cannot reset the administrator password when the account is frozen.
- If you log in to the database as the **root** user, resetting the password may interrupt services. Exercise caution when performing this operation.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, locate the instance that you want to reset password for and click **More > Reset Password** in the **Operation** column.

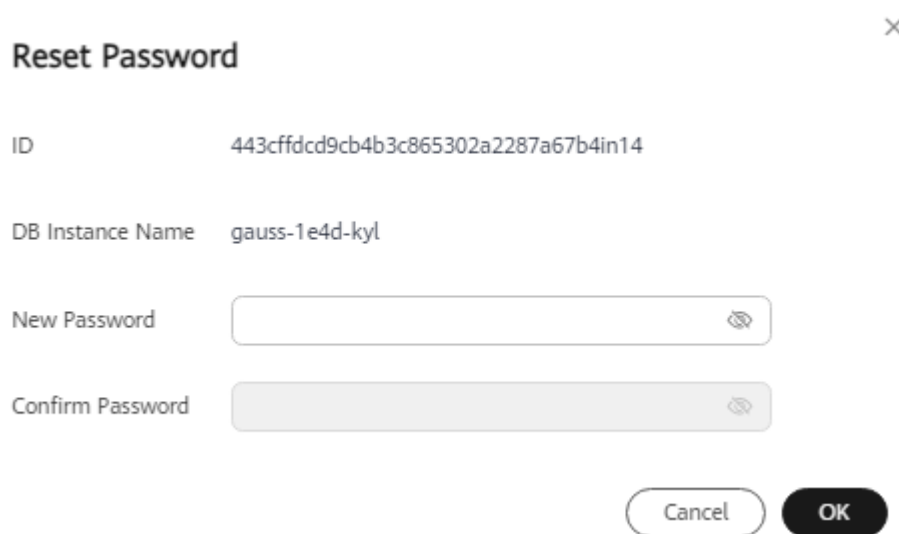
Alternatively, click the instance name to go to the **Basic Information** page. In the **Basic Information** area, click **Reset Password** next to the **Administrator** field.

Step 5 Enter a new password and confirm the password.

The new password must:

- Contain 8 to 32 characters.
- Contain at least three types of the following: uppercase letters, lowercase letters, digits, and special characters. Supported special characters: ~!@#%^*_-=+?,
- Be different from the old password or the old password written backwards.

Figure 13-1 Resetting a password

A screenshot of a 'Reset Password' dialog box. At the top right is a close button (X). The title 'Reset Password' is on the left. Below the title, there are two fields: 'ID' with the value '443cffdc9cb4b3c865302a2287a67b4in14' and 'DB Instance Name' with the value 'gauss-1e4d-kyl'. Below these are two password input fields: 'New Password' and 'Confirm Password'. Both fields have a toggle icon (an eye) on the right side. At the bottom right, there are two buttons: 'Cancel' and 'OK'.

Step 6 If you have enabled operation protection, click **Start Verification** in the displayed dialog box. On the displayed page, click **Send Code**, enter the obtained verification code, and click **Verify** to close the page.

Two-factor authentication improves the security of your account. For details about how to enable operation protection, see [Identity and Access Management User Guide](#).

----End

FAQ

[Why Did the New Password Not Take Effect After I Reset the Administrator Password of a GaussDB Instance?](#)

13.2 Enabling Transparent Data Encryption for a GaussDB Instance

Scenarios

You can enable transparent data encryption (TDE) for your instance on the instance details page.

Precautions

- TDE can be enabled only when KMS is used.
- After enabling TDE, you need to manually reboot the instance to apply the change.
- Cross-cloud DR is not supported.
- After TDE is enabled, the encryption key cannot be changed.
- If TDE is enabled for instances whose DB engine version is V2.0-8.200.0 or later, database- and table-level backup and restoration are supported.
- Encrypted tables cannot be migrated using DRS.
- TDE can be used only in instances whose kernel version is V2.0-3.300 or later.
- Do not disable or delete the KMS key specified here, or encrypted tables will be inaccessible.

Enabling Transparent Data Encryption

Step 1 [Log in to the management console](#).

Step 2 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 3 In the **Configuration** area, click **Enable** in the **Transparent Data Encryption** field.

Step 4 In the displayed dialog box, select a key from the **Key Name** drop-down list and enter **YES**.

- If no key is available, click **View Key Name List** next to the **Key Name** field to go to the **Key Management Service** page and create a key. For details about how to create a key, see [Creating a Custom Key](#).
- If the selected key has been authorized for the current user, click **OK** to enable TDE.
- If the selected key is not authorized for the current user, click **Authorize** next to the **Key Name** field to authorize key access. Then, click **OK** to enable TDE.

Figure 13-2 Enabling transparent data encryption

×

Enable Transparent Data Encryption

⚠

Transparent data encryption will not be used until after the next reboot. Do not disable or delete the used key, or encrypted tables will be inaccessible.

×

Encryption Type

KMS service

Key Name

KMS-a399

⌵

Ⓒ [Authorize](#) [View](#)

[Key Name List](#)

To confirm the enablement, enter "YES" below. [Auto Enter](#)

YES

Cancel

OK

Step 5 After a message indicating that TDE is enabled is displayed, reboot the instance for the modification to take effect.

----End

14 Parameter Management

14.1 Configurable DB Instance Parameters

This section describes the GaussDB instance parameters that can be modified.

- [Configurable Parameters for Version V2.0-8.x](#)
- [Configurable Parameters for Version V2.0-3.x](#)
- [Configurable Parameters for Version V2.0-2.x](#)

Configurable Parameters for Version V2.0-8.x

The following table describes the parameters that can be modified.

Configurable Parameters for Independently Deployed Distributed Instances of Version V2.0-8.x

Table 14-1 Parameters for distributed instances that are independently deployed

Parameter	Description
dn:qrw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:recovery_max_workers	Specifies the number of concurrent replayer threads.
cn:behavior_compat_options	Specifies database compatibility configuration items. After the value of proc_outparam_override is changed, the database must be connected again or the instance must be rebooted. Otherwise, stored procedures and functions cannot be correctly called.
dn:recyclebin_retention_time	Specifies how long files will be kept in the recycle bin, in seconds. Files in the recycle bin will be automatically deleted after this length of time.

Parameter	Description
dn:track_stmt_session_slot	Specifies the maximum number of full or slow SQL statements that can be cached in a session. If the number of full or slow SQL statements exceeds this value, new statements will not be traced until the flush thread flushes the cached statements to the disk to reserve free space. The default value is recommended.
timezone	Specifies the time zone for displaying and interpreting time stamps.
cn:auto_increment_offset	Specifies the initial value of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is greater than that of auto_increment_increment , there will be an error when the values in the auto-increment column automatically increase.
dn:enable_xid_abort_check	Specifies whether to check the status of transaction ID rollback when a transaction is committed.
cn:audit_internal_event	Specifies whether to audit the connections and operations of internal tools cm_agent, gs_clean, and WDRXdb, and whether to audit the logins and logouts from CNs on DN.
cn:codegen_compile_thread_num	Specifies the number of Codegen compilation threads.
dn:static_thread_pool_num	Specifies the number of threads used to create a static thread pool (static pool). This parameter takes effect only on CNs of a distributed instance after enable_thread_pool is enabled.
cn:auto_increment_increment	Specifies the auto-increment step of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is smaller than that of auto_increment_offset , there will be an error when the values in the auto-increment column automatically increase.
cn:sql_mode	Specifies the SQL behavior control configuration item in M-compatible mode.

Parameter	Description
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
dn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
cn:llvm_max_memory	Specifies the upper limit of the memory used by IRs (including cached and in-use IRs) generated during compilation in Codegen execution mode. The memory used by Codegen is not applied for by preoccupation. It is a part of max_dynamic_memory and is restricted by the llvm_max_memory parameter. Unit: KB
cn:hll_default_log2explicit	Specifies the threshold for switching from the explicit mode to the sparse mode.
dn:fix_func_selection	Specifies whether to optimize the function matching policy. The value catlist indicates the catlist sequence is optimized. (The non-B-compatible mode has been optimized. In non-B-compatible mode, system functions are always preferentially selected and executed. The policy in B-compatible mode is the same as that in versions earlier than 505.1.0. An error message indicating that the function is not unique may be displayed, or a system function may be selected for execution.)
dn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
global_syscache_threshold	Specifies the maximum memory usage of the global system cache. Recommended value range: 2,048 KB to 16,384 KB on average per database. If this parameter is set to a large value, the cache link may be too long and performance will deteriorate. If this parameter is set to a small value, the parameter is not applied, and after the memory usage can exceed the threshold, performance deteriorates. Unit: KB

Parameter	Description
dn:sql_mode	Specifies the SQL behavior control configuration item in M-compatible mode.
dn:verify_log_buffers	Specifies the size or pages of verifyLog buffers in memory mode. The unit is 8 KB. For example, if the value of this parameter is 4, the requested memory is 4 x 8 KB = 32 KB. This parameter is valid only when page_version_check is set to persistence . If page_version_check is set to another value, the parameter value will still be sent to the kernel, but the relevant function does not take effect until after page_version_check is set to persistence .
dn:resilience_ctrlstmt_control_iopslimit	Specifies the maximum IOPS that can be used by slow SQL statements after normal SQL statements are marked as slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. 0(None) : The IOPS is not limited. 10(LOW) : The limit level for IOPS is LOW . 20(MEDIUM) : The limit level for IOPS is MEDIUM . 50(HIGH) : The limit level for IOPS is HIGH .
cn:convert_illegal_char_mode	Specifies the placeholders of invalid characters that can be displayed on the client when enable_convert_illegal_char is enabled. Value range: 95 characters whose decimal codes range from 32 to 126 in the ASCII coding table.
cn:page_version_check	Specifies the type of page version verification. off indicates that page version verification is disabled. memory indicates that page version verification in pure memory mode is enabled. The page version information will be lost after a restart. persistence indicates that persistent page version verification is enabled. The page version information will not be lost after a restart.
cn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48.
recovery_time_target	Specifies whether the standby DB instance completes log writing and replay in streaming DR mode. If this parameter is set to a small value, the performance of the primary node is affected. If it is set to a large value, the log flow is not effectively controlled. The value 0 indicates that log flow control is disabled. Unit: second
dn:enable_convert_illegal_char	Specifies whether the database supports characters not included the character sets.

Parameter	Description
dn:system_view_version	Determines the version of the system view. All versions are backward compatible. For example, when system_view_version is set to 3 , all features of version 2 and version 1 are also supported. For details, see the product documentation.
dn:audit_dml_state_select	Determines whether to audit the SELECT operation.
dn:codegen_compile_thread_num	Specifies the number of Codegen compilation threads.
dn:resilience_ctrlstmt_detect_timelimit	Specifies the execution time of a normal SQL statement that will be marked as a slow SQL statement. The value 0 indicates that slow SQL statements are not identified. A value greater than 0 indicates that a normal SQL statement whose execution time exceeds the value of this parameter is marked as a slow SQL statement. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. Unit: millisecond
cn:a_format_date_timestamp	Specifies whether to enable current_date , current_timestamp , and localtimestamp to return the system time, instead of the transaction start time, when a transaction starts.
cn:recovery_max_workers	Specifies the number of concurrent replayer threads.
dn:max_compile_functions	Specifies the maximum number of function compilation results stored in the server. Excessive functions and compilation results of stored procedures may occupy large memory space. Setting this parameter to an appropriate value can reduce the memory usage and improve system performance. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
cn:audit_function_exec	Specifies whether to record the audit information during the execution of the stored procedures, anonymous blocks, or user-defined functions (excluding system functions). The value 0 means to disable the function, and 1 means to enable it.
dn:auto_explain_log_min_duration	Specifies the minimum duration of execution plans that are automatically printed. Only execution plans whose duration is greater than the value of auto_explain_log_min_duration will be printed. Unit: millisecond

Parameter	Description
cn:num_internal_lock_partitions	Specifies the number of internal lightweight lock partitions. Changing the value of this parameter affects performance and memory usage. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
dn:recovery_parse_workers	Specifies the number of ParseRedoRecord threads in the ultimate RTO feature. This parameter must be used together with recovery_redo_workers . If both recovery_parse_workers and recovery_redo_workers are greater than 1, ultimate RTO is enabled. If you do not want to enable ultimate RTO, retain the default value 1 for recovery_parse_workers . When enabling ultimate RTO, ensure that replication_type is set to 1. If both ultimate RTO and parallel replay are enabled, only ultimate RTO takes effect, and parallel replay is ineffective. Ultimate RTO does not support column-store tables. Therefore, disable ultimate RTO in a system where column-store tables are used or are to be used. Ultimate RTO also does not support flow control. Flow control is determined by the recovery_time_target parameter.
dn:enable_default_ustore_table	Specifies whether to enable the Ustore storage engine by default. If this parameter is set to on , all created tables are Ustore tables.
cn:enable_auto_explain	Specifies whether to automatically print execution plans. This parameter is used to locate slow stored procedures or slow queries.
dn:page_version_check	Specifies the type of page version verification. off indicates that page version verification is disabled. memory indicates that page version verification in pure memory mode is enabled. The page version information will be lost after a restart. persistence indicates that persistent page version verification is enabled. The page version information will not be lost after a restart.
cn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
cn:auto_explain_log_min_duration	Specifies the minimum duration of execution plans that are automatically printed. Only execution plans whose duration is greater than the value of auto_explain_log_min_duration will be printed. Unit: millisecond

Parameter	Description
dn:num_internal_lock_partitions	Specifies the number of internal lightweight lock partitions. Changing the value of this parameter affects performance and memory usage. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
cn:page_version_max_num	Specifies the maximum number of page version information entries that can be cached in memory. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
dn:enable_auto_explain	Specifies whether to automatically print execution plans. This parameter is used to locate slow stored procedures or slow queries.
dn:audit_function_exec	Specifies whether to record the audit information during the execution of the stored procedures, anonymous blocks, or user-defined functions (excluding system functions). The value 0 means to disable the function, and 1 means to enable it.
cn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.

Parameter	Description
dn:auto_increment_cache	Specifies the number of reserved auto-increment cache values when auto-increment is triggered by batch insertion or import of auto-increment columns. When auto-increment values are reserved, the auto-increment counter value is updated to the maximum auto-increment cache value. Before the cache values are used up, the auto-increment counter value remains unchanged, and the triggered auto-increment uses the cache values. If this parameter is set to 0, the auto-increment cache values are automatically reserved. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, 2^n values are reserved each time. For example, one value is reserved in the first auto-increment, two values are reserved in the second auto-increment, four values are reserved in the third auto-increment, and eight values are reserved in the fourth auto-increment. However, if the number of reserved values exceeds 65,535, 65,535 values are reserved. If this parameter is not set to 0, the number of reserved cache values is the value of this parameter. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, the value of auto_increment_cache is the number of auto-increment values reserved each time. The reserved cache values are valid only in the statement. If the reserved auto-increment cache values are used up and subsequent INSERT statements trigger auto-increment based on the auto-increment counter, the values in the auto-increment column in the table are discontinuous. This parameter does not affect the auto-increment column in the local temporary table.
dn:enable_codegen	Specifies whether code optimization can be enabled. Currently, code optimization uses the LLVM optimization.
dn:instr_unique_sql_combination_options	Specifies the configuration items of combining unique SQL statements of the same type. If this feature is enabled, the IDs of unique SQL statements of the same type are normalized, and the generated unique SQL strings are normalized.

Parameter	Description
dn:behavior_compat_options	Specifies database compatibility configuration items. After the value of proc_outparam_override is changed, the database must be connected again or the instance must be rebooted. Otherwise, stored procedures and functions cannot be correctly called.
dn:enable_early_free	Specifies whether the operator memory can be released in advance.
dn:page_missing_dirty_check	Specifies whether to enable the verification for pages not marked as dirty. The verification checks whether the modified pages are not marked as dirty. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
cn:undo_retention_time	Specifies how long undo logs are kept. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content. Unit: second
dn:wdr_snapshot_full_backup_interval	Specifies the interval at which a full WDR snapshot is created. The interval specified by this parameter is about a number instead of time. For example, if the parameter is set to 12 , a full snapshot and then 11 incremental snapshots are generated for each group. If the parameter is set to 1 , all snapshots generated are full snapshots.
cms:datastorage_threshold_value_check	Specifies the disk usage threshold to put a database node into read-only mode. If the disk usage of a data directory exceeds this threshold, the database node is automatically changed to read-only. Unit: percentage (%)
dn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48.
dn:wdr_snapshot_space_threshold	Specifies the threshold for controlling the space used by snapshots. When the space used by snapshots reaches 80% of the value of this parameter, the control logic of the database is enabled to stabilize the space usage. Unit: KB

Parameter	Description
cn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
dn:hll_duplicate_check	Specifies whether duplicate check is enabled by default.
cn:gs_perf_interval	Specifies the automatic perf data collection interval. The value 0 indicates that the collection is stopped. If the value is greater than 0 and less than 5, the value 5 is used. Unit: minute
cn:grw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:gs_perf_interval	Specifies the automatic perf data collection interval. The value 0 indicates that the collection is stopped. If the value is greater than 0 and less than 5, the value 5 is used. Unit: minute
dn:extra_float_digits	Adjusts the number of digits displayed for floating-point values, including float4, float8, and geometric data types. The parameter value is added to the standard number of digits (FLT_DIG or DBL_DIG as appropriate). This parameter can also be set to a negative value to suppress unwanted digits.
cn:disable_keyword_options	Specifies the configuration items for database compatibility. Multiple items are separated by commas (,). An identifier with this parameter set will not be used as a keyword.
dn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
cn:max_standby_archive_delay	Specifies the wait period before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode. -1 indicates that the standby node waits until the conflicting queries are complete. Unit: millisecond
track_stmt_stat_level	Controls the level of statement execution tracking.

Parameter	Description
dn:max_standby_archive_delay	Specifies the wait period before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode. -1 indicates that the standby node waits until the conflicting queries are complete. Unit: millisecond
cn:b_format_dev_version	Specifies the compatibility configuration item of database platform minor versions.
dn:copy_special_character_version	Specifies whether to report an error when there are invalid characters during data import and export using COPY FROM.
cn:page_version_recycler_thread_num	Specifies the number of threads for recycling and verifying page version information. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
enable_wdr_snapshot	Specifies whether to enable WDR snapshots. If this parameter is enabled, the kernel periodically takes performance snapshots.
dn:effective_cache_size	Specifies the size of the disk buffer available to the DN optimizer in a single query. Unit: 8 KB
cn:wdr_snapshot_full_backup_interval	Specifies the interval at which a full WDR snapshot is created. The interval specified by this parameter is about a number instead of time. For example, if the parameter is set to 12 , a full snapshot and then 11 incremental snapshots are generated for each group. If the parameter is set to 1 , all snapshots generated are full snapshots.
cn:extra_float_digits	Adjusts the number of digits displayed for floating-point values, including float4, float8, and geometric data types. The parameter value is added to the standard number of digits (FLT_DIG or DBL_DIG as appropriate). This parameter can also be set to a negative value to suppress unwanted digits.
cn:system_view_version	Determines the version of the system view. All versions are backward compatible. For example, when system_view_version is set to 3 , all features of version 2 and version 1 are also supported. For details, see the product documentation.

Parameter	Description
cn:recyclebin_retention_time	Specifies how long files will be kept in the recycle bin. Files in the recycle bin will be automatically deleted after this length of time. Unit: second
cn:password_encryption_type	Specifies how user passwords are encrypted. 0 : Passwords are encrypted using MD5. 1 : Passwords are encrypted using SHA-256 and MD5. 2 : Passwords are encrypted using SHA-256. 3 : Passwords are encrypted using SM3. MD5 is not recommended because it is a weak encryption algorithm.
cn:check_disconnect_query	Specifies whether to enable the function of terminating statement execution on the server after the client is disconnected due to timeout.
password_effect_time	Specifies the validity period of an account password. Unit: day
cn:verify_log_buffers	Specifies the size or pages of verifyLog buffers in memory mode. The unit is 8 KB. For example, if the value of this parameter is 4, the requested memory is 4 x 8 KB = 32 KB. This parameter is valid only when page_version_check is set to persistence . If page_version_check is set to another value, the parameter value will still be sent to the kernel, but the relevant function does not take effect until after page_version_check is set to persistence .
dn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
dn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
cn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
cn:enable_rls_match_index	Specifies whether indexes of a base table can be scanned based on target predicate conditions in row-level security scenarios. Target scenario: Row level security (RLS) policies are set and enabled in the base table, and the query predicate contains the unleakproof system function or like operator.

Parameter	Description
dn:enable_analyze_check	Specifies whether to check if statistics were collected about tables whose reltuples and relpages are displayed as 0 in pg_class during plan generation.
cn:resilience_ctrlstmt_control_iopslimit	Specifies the maximum IOPS that can be used by slow SQL statements after normal SQL statements are marked as slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. 0(None) : The IOPS is not limited. 10(LOW) : The limit level for IOPS is LOW . 20(MEDIUM) : The limit level for IOPS is MEDIUM . 50(HIGH) : The limit level for IOPS is HIGH .
dn:auto_increment_increment	Specifies the auto-increment step of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is smaller than that of auto_increment_offset , there will be an error when the values in the auto-increment column automatically increase.
cn:copy_special_character_version	Specifies whether to report an error when there are invalid characters during data import and export using COPY FROM.
cn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
dn:gs_format_behavior_compat_options	Specifies the configuration items of GaussDB internal system functions.
cn:hll_duplicate_check	Specifies whether duplicate check is enabled by default.
cn:max_standby_streaming_delay	Specifies the wait period before queries on the standby node are canceled when the queries conflict with WAL data receiving through streaming replication in hot standby mode. -1 indicates that the standby node waits until the conflicting queries are complete. Unit: millisecond
dn:b_format_dev_version	Specifies the compatibility configuration item of database platform minor versions.
dn:check_disconnect_query	Specifies whether to enable the function of terminating statement execution on the server after the client is disconnected due to timeout.

Parameter	Description
cn:gs_format_behavior_compat_options	Specifies the configuration items of GaussDB internal system functions.
cn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.
cn:static_thread_pool_num	Specifies the number of threads used to create a static thread pool (static pool). This parameter takes effect only on CNs of a distributed instance after enable_thread_pool is enabled.
dn:max_standby_streaming_delay	Specifies the wait period before queries on the standby node are canceled when the queries conflict with WAL data receiving through streaming replication in hot standby mode. -1 indicates that the standby node waits until the conflicting queries are complete. Unit: millisecond
audit_system_object	Specifies whether to audit the CREATE, DROP, and ALTER operations on GaussDB database objects. GaussDB database objects include databases, users, schemas, and tables. You can change the value of this parameter to audit only the operations on required database objects. In the scenario where the leader node is forcibly selected, you are advised to set audit_system_object to the maximum value and audit all DDL objects. For details about the value range, see the product documentation.
cn:enable_default_ustore_table	Specifies whether to enable the Ustore storage engine by default. If this parameter is set to on , all created tables are Ustore tables.
cn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.

Parameter	Description
dn:page_version_partitions	Specifies the number of hash table partitions used for caching page version information in memory. This parameter directly affects the hash query efficiency and hash collision probability, and is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
dn:page_version_recycler_thread_num	Specifies the number of threads for recycling and verifying page version information. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
dn:password_encryption_type	Specifies how user passwords are encrypted. 0 : Passwords are encrypted using MD5. 1 : Passwords are encrypted using SHA-256 and MD5. 2 : Passwords are encrypted using SHA-256. 3 : Passwords are encrypted using SM3. MD5 is not recommended because it is a weak encryption algorithm.
dn:dcf_thread_effective_time	Specifies the effective time of the DCF flushing thread. This parameter is used to determine whether the disk I/O hangs. If the DCF cannot access I/O resources within the period specified by this parameter, the DCF considers that the thread I/O hangs and a primary/standby switchover is triggered. If this parameter is set to 0 , I/O hang detection is disabled. Unit: second
dn:a_format_date_timestamp	Specifies whether to enable current_date , current_timestamp , and localtimestamp to return the system time, instead of the transaction start time, when a transaction starts.
dn:hll_default_log2sparse	Specifies the default threshold for switching from Sparse mode to Full mode.
cn:enable_xid_abort_check	Specifies whether to check the status of transaction ID rollback when a transaction is committed.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second

Parameter	Description
cn:hll_default_log2sparse	Specifies the default threshold for switching from Sparse mode to Full mode.
dn:disable_keyword_options	Specifies the configuration items for database compatibility. Multiple items are separated by commas (,). An identifier with this parameter set will not be used as a keyword.
cn:fix_func_selection	Specifies whether to optimize the function matching policy. The value catlist indicates the catlist sequence is optimized. (The non-B-compatible mode has been optimized. In non-B-compatible mode, system functions are always preferentially selected and executed. The policy in B-compatible mode is the same as that in versions earlier than 505.1.0. An error message indicating that the function is not unique may be displayed, or a system function may be selected for execution.)
cn:support_binary_copy_version	Specifies whether to verify the binary file encoding information when data is imported using COPY FROM in BINARY mode. If forward compatibility is required, leave this parameter empty. Otherwise, retain the default value.
autoanalyze	Specifies whether to automatically collect statistics on tables that have no statistics when a plan is generated.
password_lock_time	Specifies the lockout duration (in seconds) of an account after it exceeds the maximum number of failed login attempts specified by failed_login_attempts . The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
wdr_snapshot_interval	Specifies the interval at which the background thread Snapshot automatically takes snapshots of the database monitoring data. Unit: minute
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond

Parameter	Description
dn:undo_retention_time	Specifies how long undo logs are kept. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content. Unit: second
cn:recovery_parse_workers	Specifies the number of ParseRedoRecord threads in the ultimate RTO feature. This parameter must be used together with recovery_redo_workers . If both recovery_parse_workers and recovery_redo_workers are greater than 1, ultimate RTO is enabled. If you do not want to enable ultimate RTO, retain the default value 1 for recovery_parse_workers . When enabling ultimate RTO, ensure that replication_type is set to 1. If both ultimate RTO and parallel replay are enabled, only ultimate RTO takes effect, and parallel replay is ineffective. Ultimate RTO does not support column-store tables. Therefore, disable ultimate RTO in a system where column-store tables are used or are to be used. Ultimate RTO also does not support flow control. Flow control is determined by the recovery_time_target parameter.
cn:undo_space_limit_size	Specifies the threshold for forcibly recycling undo space. When the undo space usage reaches 80% of the threshold, forcible recycling starts. It is recommended that the value of this parameter be greater than or equal to the value of undo_limit_size_per_transaction . Unit: 8 KB

Parameter	Description
cn:auto_increment_cache	Specifies the number of reserved auto-increment cache values when auto-increment is triggered by batch insertion or import of auto-increment columns. When auto-increment values are reserved, the auto-increment counter value is updated to the maximum auto-increment cache value. Before the cache values are used up, the auto-increment counter value remains unchanged, and the triggered auto-increment uses the cache values. If this parameter is set to 0, the auto-increment cache values are automatically reserved. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, 2^n values are reserved each time. For example, one value is reserved in the first auto-increment, two values are reserved in the second auto-increment, four values are reserved in the third auto-increment, and eight values are reserved in the fourth auto-increment. However, if the number of reserved values exceeds 65,535, 65,535 values are reserved. If this parameter is not set to 0, the number of reserved cache values is the value of this parameter. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, the value of auto_increment_cache is the number of auto-increment values reserved each time. The reserved cache values are valid only in the statement. If the reserved auto-increment cache values are used up and subsequent INSERT statements trigger auto-increment based on the auto-increment counter, the values in the auto-increment column in the table are discontinuous. This parameter does not affect the auto-increment column in the local temporary table.
dn:enable_hotkeys_collection	Specifies whether to collect statistics on accessed key values in databases.
cn:enable_early_free	Specifies whether the operator memory can be released in advance.

Parameter	Description
cn:max_concurrent_autonomous_transactions	Specifies the maximum number of autonomous transaction connections, that is, the maximum number of concurrent autonomous transactions executed at the same time. If this parameter is set to 0 , autonomous transactions cannot be executed. The theoretical maximum value is 10000 . Set this parameter based on workload requirements and hardware configurations. It is recommended that this parameter be set to a value less than or equal to 1/10 of max_connections .
cn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
cn:enable_enhance_toast_table	Specifies whether to use the enhanced TOAST out-of-line storage table. The value on indicates that the enhanced TOAST out-of-line storage table is used. The value off indicates that the TOAST out-of-line storage table is used.
cn:archive_interval	Specifies the archiving interval. Log files are forcibly archived when the period specified by this parameter has elapsed. A large value of this parameter affects the RPO of PITR. The default value is recommended. Unit: second
dn:undo_limit_size_per_transaction	Specifies the maximum undo space for a single transaction. If the undo space of a transaction exceeds this parameter value, the transaction is rolled back due to an error. It is recommended that the value of this parameter be smaller than the value of undo_space_limit_size . If this parameter value is larger, the value of undo_space_limit_size will be used as the maximum undo space for a single transaction. If this undo space is greater than 1 TB, system performance and stability may be affected. Unit: 8 KB
dn:convert_illegal_char_mode	Specifies the placeholders of invalid characters that can be displayed on the client when enable_convert_illegal_char is enabled. Value range: 95 characters whose decimal codes range from 32 to 126 in the ASCII coding table.
cn:enable_analyze_check	Specifies whether to check if statistics were collected about tables whose reltuples and relpages are displayed as 0 in pg_class during plan generation.

Parameter	Description
dn:tde_index_default_encrypt	When tde_index_default_encrypt is set to on and an index is created based on an encrypted table, the database automatically converts the index to an encrypted index.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
dn:enable_bbox_dump	Specifies whether to enable the black box function so that core files can be generated even when the core mechanism is not configured in the system.
dn:enable_recyclebin	Enables or disables the recycle bin in real time.
cn:wdr_snapshot_space_threshold	Specifies the threshold for controlling the space used by snapshots. When the space used by snapshots reaches 80% of the value of this parameter, the control logic of the database is enabled to stabilize the space usage. Unit: KB
enable_global_syscache	Determines whether to enable global system cache.
dn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
cn:max_compile_functions	Specifies the maximum number of function compilation results stored in the server. Excessive functions and compilation results of stored procedures may occupy large memory space. Setting this parameter to an appropriate value can reduce the memory usage and improve system performance. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
dn:auto_increment_offset	Specifies the initial value of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times \text{auto\_increment\_increment}$. N is a positive integer. If the value of this parameter is greater than that of auto_increment_increment , there will be an error when the values in the auto-increment column automatically increase.

Parameter	Description
cn:track_stmt_session_slot	Specifies the maximum number of full or slow SQL statements that can be cached in a session. If the number of full or slow SQL statements exceeds this value, new statements will not be traced until the flush thread flushes the cached statements to the disk to reserve free space. The default value is recommended.
cn:page_version_partitions	Specifies the number of hash table partitions used for caching page version information in memory. This parameter directly affects the hash query efficiency and hash collision probability, and is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. The value 0 indicates that there is no timeout limit. Unit: second
dn:index_txntype	Sets the index page format to PCR or RCR. This parameter is left unconfigured during system initialization. By default, the created indexes are compatible with the index type (RCR) of earlier versions. Once this parameter is specified, it cannot be left unconfigured again.
cn:enable_recyclebin	Enables or disables the recycle bin in real time.
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
max_replication_slots	Specifies the number of log replication slots on the primary node.
cn:enable_bbox_dump	Specifies whether to enable the black box function so that core files can be generated even when the core mechanism is not configured in the system.
datestyle	Specifies the display format for date and time.

Parameter	Description
dn:hll_default_log2m	Specifies the number of buckets for HLL data.
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots. Currently, only archive slots and backup slots are involved. Set this parameter to on only in cloud scenarios where logical replication is enabled. In other scenarios, set this parameter to off .
cn:enable_convert_illegal_char	Specifies whether the database supports characters not included the character sets.
cn:enable_workload_rule	Specifies whether to enable SQL throttling.
cn:index_txntype	Sets the index page format to PCR or RCR. This parameter is left unconfigured during system initialization. By default, the created indexes are compatible with the index type (RCR) of earlier versions. Once this parameter is specified, it cannot be left unconfigured again.
dn:enable_enhance_toast_table	Specifies whether to use the enhanced TOAST out-of-line storage table. The value on indicates that the enhanced TOAST out-of-line storage table is used. The value off indicates that the TOAST out-of-line storage table is used.
dn:support_binary_copy_version	Specifies whether to verify the binary file encoding information when data is imported using COPY FROM in BINARY mode. If forward compatibility is required, leave this parameter empty. Otherwise, retain the default value.
enable_stream_operator	Specifies the query optimizer's use of streams. If enable_stream_operator is disabled, a large number of logs indicating that the plans cannot be pushed down are recorded. If you do not need these logs, you are advised to disable both enable_unshipping_log and enable_stream_operator . The default value is recommended.
cn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.
cn:gs_perf_retention_days	Specifies how many days the flame graph files in HTML format are retained. Unit: day

Parameter	Description
dn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
dn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.
dn:enable_rls_match_index	Specifies whether indexes of a base table can be scanned based on target predicate conditions in row-level security scenarios. Target scenario: Row level security (RLS) policies are set and enabled in the base table, and the query predicate contains the unleakproof system function or like operator.
dn:audit_internal_event	Specifies whether to audit the connections and operations of internal tools cm_agent, gs_clean, and WDRXdb, and whether to audit the logins and logouts from CNs on DN.
cn:effective_cache_size	Specifies the size of the disk buffer available to the CN optimizer in a single query. Unit: 8 KB
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.
dn:hll_default_log2explicit	Specifies the threshold for switching from the explicit mode to the sparse mode.
cn:instr_unique_sql_combination_options	Specifies the configuration items of combining unique SQL statements of the same type.
dn:undo_space_limit_size	Specifies the threshold for forcibly recycling undo space. When the undo space usage reaches 80% of the threshold, forcible recycling starts. It is recommended that the value of this parameter be greater than or equal to the value of undo_limit_size_per_transaction . Unit: 8 KB
dn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.

Parameter	Description
dn:llvm_max_memory	Specifies the upper limit of the memory used by IRs (including cached and in-use IRs) generated during compilation in Codegen execution mode. The memory used by Codegen is not applied for by preoccupation. It is a part of max_dynamic_memory and is restricted by the llvm_max_memory parameter. Unit: KB
dn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
cn:tde_index_default_encrypt	When tde_index_default_encrypt is set to on and an index is created based on an encrypted table, the database automatically converts the index to an encrypted index.
dn:enable_workload_rule	Specifies whether to enable SQL throttling.
dn:archive_interval	Specifies the archiving interval. Log files are forcibly archived when the period specified by this parameter has elapsed. A large value of this parameter affects the RPO of PITR. The default value is recommended. Unit: second
cn:enable_hotkeys_collection	Specifies whether to collect statistics on accessed key values in databases.
cn:enable_codegen	Specifies whether code optimization can be enabled. Currently, code optimization uses the LLVM optimization.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DNs connect to primary DNs to obtain physical logs, and logical replication tools connect to primary DNs to obtain logical logs. If this parameter is set to a value smaller than 20, scale-out may fail. The value of this parameter must be smaller than that of max_connections .
cn:undo_limit_size_per_transaction	Specifies the maximum undo space for a single transaction. If the undo space of a transaction exceeds this parameter value, the transaction is rolled back due to an error. It is recommended that the value of this parameter be smaller than the value of undo_space_limit_size . If this parameter value is larger, the value of undo_space_limit_size will be used as the maximum undo space for a single transaction. If this undo space is greater than 1 TB, system performance and stability may be affected. Unit: 8 KB

Parameter	Description
dn:max_concurrent_autonomous_transactions	Specifies the maximum number of autonomous transaction connections, that is, the maximum number of concurrent autonomous transactions executed at the same time. If this parameter is set to 0 , autonomous transactions cannot be executed. The theoretical maximum value is 10000 . Set this parameter based on workload requirements and hardware configurations. It is recommended that this parameter be set to a value less than or equal to 1/10 of max_connections .
cn:page_missing_dirty_check	Specifies whether to enable the verification for pages not marked as dirty. The verification checks whether the modified pages are not marked as dirty. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
cn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.
cn:audit_dml_state_select	Determines whether to audit the SELECT operation.
cn:hll_default_log2m	Specifies the number of buckets for HLL data.
dn:use_workload_manager	Specifies whether to enable resource management. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:use_workload_manager	Specifies whether to enable resource management. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_save_datachanged_timestamp	Specifies whether to record the time when INSERT, UPDATE, DELETE, or EXCHANGE/TRUNCATE/DROP PARTITION is performed on table data. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_save_datachanged_timestamp	Specifies whether to record the time when INSERT, UPDATE, DELETE, or EXCHANGE/TRUNCATE/DROP PARTITION is performed on table data. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:max_process_memory	Specifies the maximum physical memory allowed for a DN, in KB. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:max_process_memory	Specifies the maximum physical memory allowed for a CN, in KB. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_shared_buffers_fraction	Specifies the shared_buffers proportion used on the server where a standby instance is deployed. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:standby_shared_buffers_fraction	Specifies the shared_buffers proportion used on the server where a standby instance is deployed. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:pagewriter_sleep	Specifies the interval (in milliseconds) for the page writer thread to flush dirty pages to disks after the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:pagewriter_sleep	Specifies the interval (in milliseconds) for the page writer thread to flush dirty pages to disks after the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:xloginsert_locks	Specifies the number of locks on concurrent write-ahead logging. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:xloginsert_locks	Specifies the number of locks on concurrent write-ahead logging. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_light_proxy	Specifies whether the optimizer optimizes the execution of simple queries on CNs. This parameter does not take effect if the character set of the application side does not match that of the kernel side. You are advised to set the character set to UTF-8 when creating a database. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_light_proxy	Specifies whether the optimizer optimizes the execution of simple queries on CNs. This parameter does not take effect if the character set of the application side does not match that of the kernel side. You are advised to set the character set to UTF-8 when creating a database. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:synchronous_commit	Specifies the synchronization mode of the current transaction. Generally, logs generated by a transaction are synchronized in the following sequence: The primary node writes the logs to the local memory. The primary node writes logs in the local memory to the local file system. The primary node flushes the logs in the local file system to disks. The primary node sends the logs to the standby node. The standby node receives the logs and saves them to its local memory. The standby node writes the logs in the local memory to the local file system. The standby node flushes the logs in the local file system to disks. The standby node replays the logs to complete the incremental update of data files. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:synchronous_commit	Specifies the synchronization mode of the current transaction. Generally, logs generated by a transaction are synchronized in the following sequence: The primary node writes the logs to the local memory. The primary node writes logs in the local memory to the local file system. The primary node flushes the logs in the local file system to disks. The primary node sends the logs to the standby node. The standby node receives the logs and saves them to its local memory. The standby node writes the logs in the local memory to the local file system. The standby node flushes the logs in the local file system to disks. The standby node replays the logs to complete the incremental update of data files. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_incremental_checkpoint	Specifies whether to enable incremental checkpoint. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_incremental_checkpoint	Specifies whether to enable incremental checkpoint. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_gsstat_cache	Specifies whether to enable cache acceleration for PgStat functions. PgStat functions include <code>pg_stat_get_numscans</code>, <code>pg_stat_get_tuples_returned</code>, <code>pg_stat_get_tuples_fetched</code>, <code>pg_stat_get_tuples_inserted</code>, <code>pg_stat_get_tuples_updated</code>, <code>pg_stat_get_tuples_deleted</code>, <code>pg_stat_get_tuples_hot_updated</code>, <code>pg_stat_get_live_tuples</code>, <code>pg_stat_get_dead_tuples</code>, <code>pg_stat_get_autovacuum_count</code>, <code>pg_stat_get_autoanalyze_count</code>, <code>pg_stat_get_blocks_fetched</code>, <code>pg_stat_get_blocks_hit</code>, <code>pg_timestamp_stat_get_last_autoanalyze_time</code>, <code>pg_timestamp_stat_get_last_autovacuum_time</code>, and <code>pg_timestamp_stat_get_last_updated</code>. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:bypass_workload_manager	Enables or disables independent I/O control. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bypass_workload_manager	Enables or disables independent I/O control. This parameter is supported in GaussDB V2.0-8.201 or later.
enable_resource_track	Specifies whether to enable real-time resource monitoring. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_rt_percentile	Specifies whether to enable the function of calculating the response time of 80% and 95% of the SQL statements in the system. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_instr_rt_percentile	Specifies whether to enable the function of calculating the response time of 80% and 95% of the SQL statements in the system. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_cpu_timer	Specifies whether to capture the CPU time consumed during SQL statement execution. In the x86-based centralized deployment scenario where 32 vCPUs and 256 GB of memory are configured, the performance tested by BenchmarkSQL 5.0 fluctuates by about 3.5% by enabling or disabling this parameter. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_instr_cpu_timer	Specifies whether to capture the CPU time consumed during SQL statement execution. In the x86-based centralized deployment scenario where 32 vCPUs and 256 GB of memory are configured, the performance tested by BenchmarkSQL 5.0 fluctuates by about 3.5% by enabling or disabling this parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
track_counts	Determines whether to enable collection of statistics on database activities. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_user_metric_persistent	Specifies whether to dump the historical monitoring data of user resources. If this parameter is set to on , data in the PG_TOTAL_USER_RESOURCE_INFO view is periodically sampled and saved to the system catalog GS_WLM_USER_RESOURCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_user_metric_persistent	Specifies whether to dump the historical monitoring data of user resources. If this parameter is set to on , data in the PG_TOTAL_USER_RESOURCE_INFO view is periodically sampled and saved to the system catalog GS_WLM_USER_RESOURCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:log_hostname	By default, connection log messages only show the IP address of the connecting host. The host name can be recorded when this parameter is set to on . It may take some time to parse the host name. Therefore, the database performance may be affected. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:log_hostname	By default, connection log messages only show the IP address of the connecting host. The host name can be recorded when this parameter is set to on . It may take some time to parse the host name. Therefore, the database performance may be affected. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_track_wait	Specifies whether to enable real-time collection of wait event information. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_instr_track_wait	Specifies whether to enable real-time collection of wait event information. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instance_metric_persistent	Specifies whether to dump instance resource monitoring data. When this parameter is set to on , the instance monitoring data is saved to the system catalog GS_WLM_INSTANCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_instance_metric_persistent	Specifies whether to dump instance resource monitoring data. When this parameter is set to on , the instance monitoring data is saved to the system catalog GS_WLM_INSTANCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_logical_io_statistics	Specifies whether to collect logical I/O statistics during resource monitoring. If this function is enabled, the read_kbytes , write_kbytes , read_counts , write_counts , read_speed , and write_speed columns in the PG_TOTAL_USER_RESOURCE_INFO view will collect statistics on the byte count, number of times, and speed of logical read and write. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_logical_io_statistics	Specifies whether to collect logical I/O statistics during resource monitoring. If this function is enabled, the read_kbytes , write_kbytes , read_counts , write_counts , read_speed , and write_speed columns in the PG_TOTAL_USER_RESOURCE_INFO view will collect statistics on the byte count, number of times, and speed of logical read and write. Columns related to logical read and write in the system catalogs GS_WLM_USER_RESOURCE_HISTORY and GS_WLM_INSTANCE_HISTORY will collect statistics on the logical read and write of related users and instances. This parameter is supported in GaussDB V2.0-8.201 or later.
enable_sort	Specifies whether the query optimizer uses sort methods. It is impossible to suppress explicit sorts entirely, but setting this variable to off encourages the optimizer to choose other methods if available. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
enable_material	Specifies whether the query optimizer uses materialization. It is impossible to suppress materialization entirely, but setting this variable to off prevents the optimizer from inserting materialized nodes. This parameter is supported in GaussDB V2.0-8.201 or later.
gtm:gtm_max_trans	Specifies the maximum number of connections accepted by the GTM. You are advised not to change the value. This parameter is supported in GaussDB V2.0-8.201 or later.
gtm:csn_sync_interval	Specifies the interval for synchronizing CSNs between the primary and standby GTMs, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
gtm:restore_duration	Specifies the maximum number of transactions (identified by XIDs or CSNs) that the GTM will process in a single restoration cycle. This parameter is supported in GaussDB V2.0-8.201 or later.
cms:coordinator_heartbeat_timeout	Specifies the heartbeat timeout (in seconds) that triggers the automatic removal of faulty CNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:client_min_messages	Specifies which level of messages will be sent to the client. A higher level covers the messages of all the lower levels. The lower the level is, the fewer messages are sent. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:client_min_messages	Specifies which level of messages will be sent to the client. A higher level covers the messages of all the lower levels. The lower the level is, the fewer messages are sent. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:pagewriter_thread_num	Specifies the number of threads for background page flushing after the incremental checkpoint is enabled. Dirty pages are flushed in sequence to disks, updating recovery points. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:pagewriter_thread_num	Specifies the number of threads for background page flushing after the incremental checkpoint is enabled. Dirty pages are flushed in sequence to disks, updating recovery points. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:bgwriter_delay	Specifies the interval at which the background writer writes dirty shared buffers, in milliseconds. Each time, the background writer initiates write operations for some dirty buffers. In full checkpoint mode, the bgwriter_lru_maxpages parameter is used to control the amount of data to be written each time, and the process is restarted after the period of hibernation specified by bgwriter_delay (in milliseconds). In incremental checkpoint mode, the number of target idle buffer pages is calculated based on the value of candidate_buf_percent_target. If the number of idle buffer pages is insufficient, a batch of pages are flushed to disks at the interval specified by bgwriter_delay (in milliseconds). The number of flushed pages is calculated based on the target difference percentage. The maximum number of flushed pages is limited by max_io_capacity. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:bgwriter_delay	Specifies the interval at which the background writer writes dirty shared buffers, in milliseconds. Each time, the background writer initiates write operations for some dirty buffers. In full checkpoint mode, the bgwriter_lru_maxpages parameter is used to control the amount of data to be written each time, and the process is restarted after the period of hibernation specified by bgwriter_delay (in milliseconds). In incremental checkpoint mode, the number of target idle buffer pages is calculated based on the value of candidate_buf_percent_target. If the number of idle buffer pages is insufficient, a batch of pages are flushed to disks at the interval specified by bgwriter_delay (in milliseconds). The number of flushed pages is calculated based on the target difference percentage. The maximum number of flushed pages is limited by max_io_capacity. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:max_files_per_process	Specifies the maximum number of files that can be opened simultaneously by each server process. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:candidate_buf_percent_target	Specifies the proportion (%) of available buffers in shared_buffer in the candidate buffer chain when the incremental checkpoint is enabled. If the percentage is less than the value of this parameter, the bgwriter thread starts to flush dirty pages based on the max_io_capacity parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:candidate_buf_percent_target	Specifies the proportion (%) of available buffers in shared_buffer in the candidate buffer chain when the incremental checkpoint is enabled. If the percentage is less than the value of this parameter, the bgwriter thread starts to flush dirty pages based on the max_io_capacity parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:checkpoint_segments	Specifies the minimum number of WAL segment files in the period specified by checkpoint_timeout . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:checkpoint_segments	Specifies the minimum number of WAL segment files in the period specified by checkpoint_timeout . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:checkpoint_timeout	Specifies the maximum time between automatic WAL checkpoints, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:checkpoint_timeout	Specifies the maximum time between automatic WAL checkpoints, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:incremental_checkpoint_timeout	Specifies the maximum interval (in seconds) between automatic WAL checkpoints when the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:incremental_checkpoint_timeout	Specifies the maximum interval (in seconds) between automatic WAL checkpoints when the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:wal_writer_delay	Specifies the delay (in milliseconds) between activity rounds for the WAL writer. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:wal_writer_delay	Specifies the delay (in milliseconds) between activity rounds for the WAL writer. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:wal_buffers	Specifies the shared memory for storing WALs, that is, multiples of XLOG_BLCKSZ , in increments of 8 KB. For details about the value range, see wal_buffers . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:wal_buffers	Specifies the shared memory for storing WALs, that is, multiples of XLOG_BLCKSZ , in increments of 8 KB. For details about the value range, see wal_buffers . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bulk_write_ring_size	Specifies the size of the ring buffer used by the operation when a large amount of data is written (for example, the copy operation). This parameter is supported in GaussDB V2.0-8.201 or later.
cn:bulk_write_ring_size	Specifies the size of a ring buffer used for parallel data import. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:recovery_redo_workers	Specifies the number of PageRedoWorker threads corresponding to each ParseRedoRecord thread when ultimate RTO is enabled. recovery_redo_workers must be used together with recovery_parse_workers . The value of recovery_redo_workers takes effect only when recovery_parse_workers is greater than 1. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:recovery_redo_workers	Specifies the number of PageRedoWorker threads corresponding to each ParseRedoRecord thread when ultimate RTO is enabled. recovery_redo_workers must be used together with recovery_parse_workers . The value of recovery_redo_workers takes effect only when recovery_parse_workers is greater than 1. This parameter is supported in GaussDB V2.0-8.201 or later.
cms:phony_dead_effective_time	Specifies the maximum number of times DN processes are detected as zombie. If the number of times a process is detected as zombie is greater than the specified value, the process is considered to be a zombie process and will be restarted. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_ai_watchdog	Enables or disables the AI watchdog function. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_ai_watchdog	Enables or disables the AI watchdog function. This parameter is supported in GaussDB V2.0-8.201 or later.
enable_unshipping_log	Specifies whether to log statements that are not pushed down. The logs help locate performance issues that may be caused by statements not pushed down. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:max_connections	Specifies the maximum number of concurrent connections. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:max_connections	Specifies the maximum number of concurrent connections. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:resilience_memory_reject_percent	Specifies the dynamic memory usage for escape from memory overload. This parameter is only applied when use_workload_manager and enable_memory_limit are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:resilience_memory_reject_percent	Specifies the dynamic memory usage for escape from memory overload. This parameter is only applied when use_workload_manager and enable_memory_limit are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:resilience_threadpool_reject_cond	Specifies the proportion of accumulated sessions in the thread pool for escape from overload. This parameter is only applied when enable_thread_pool and use_workload_manager are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:resilience_threadpool_reject_cond	Specifies the proportion of accumulated sessions in the thread pool for escape from overload. This parameter is only applied when enable_thread_pool and use_workload_manager are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:vacuum_cost_delay	Specifies how long the process sleeps when vacuum_cost_limit has been exceeded, in milliseconds. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:vacuum_cost_delay	Specifies how long the process sleeps when vacuum_cost_limit has been exceeded, in milliseconds. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:b_format_version	Specifies the SQL behavior control configuration item in B-compatible mode. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:b_format_version	Specifies the forward compatibility in MySQL-compatible mode. This is a configuration item for controlling database platform compatibility behavior. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_control_group	Specifies whether to enable the Cgroups. This parameter must be applied on both CNs and DN. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_control_group	Specifies whether to enable the Cgroups. This parameter must be applied on both CNs and DN. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:track_stmt_parameter	After track_stmt_parameter is enabled, the executed statements recorded in statement_history are not normalized. The complete SQL statement information can be displayed to help the database administrator locate faults. For a simple query, the complete statement information is displayed. For a PBE statement, the complete statement information and information about each variable value are displayed. The format is "query string; parameters: \$1=value1,\$2=value2, ...". This parameter is used to display full SQL information for users and is not controlled by the track_activity_query_size parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:track_stmt_parameter	After track_stmt_parameter is enabled, the executed statements recorded in statement_history are not normalized. The complete SQL statement information can be displayed to help the database administrator locate faults. For a simple query, the complete statement information is displayed. For a PBE statement, the complete statement information and information about each variable value are displayed. The format is "query string; parameters: \$1=value1,\$2=value2, ...". This parameter is used to display full SQL information for users and is not controlled by the track_activity_query_size parameter. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:track_activity_query_size	Specifies byte counts of the current running commands used to trace each active session. If the number of bytes in a command is greater than the value of this parameter, the command will be truncated. This parameter affects the memory usage of multiple system views. Adjusting the value of this parameter has a great impact on the memory usage. You can estimate the memory usage based on the value calculated from instr_unique_sql_count times track_activity_query_size . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:track_activity_query_size	Specifies byte counts of the current running commands used to trace each active session. If the number of bytes in a command is greater than the value of this parameter, the command will be truncated. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_lock_account	Specifies whether to automatically lock an account when the number of consecutive login failures using incorrect passwords reaches the value of failed_login_attempts . Manual locking by administrators is not controlled by this parameter. Setting this parameter to off reduces account security. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_lock_account	Specifies whether to automatically lock an account when the number of consecutive login failures using incorrect passwords reaches the value of failed_login_attempts . Manual locking by administrators is not controlled by this parameter. Setting this parameter to off reduces account security. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:instr_unique_sql_track_type	Specifies how unique SQL statements are recorded. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:instr_unique_sql_track_type	Specifies how unique SQL statements are recorded. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:idle_in_transaction_timeout	Specifies the maximum idle duration (in seconds) of a transaction after the connection to the server is established and the transaction is started. The value 0 indicates that the timeout setting is disabled. A positive number indicates the maximum idle duration of a transaction after the connection to the server is established and the transaction is started. If the idle duration exceeds the value of this parameter, the client will be disconnected from the server. Set this parameter based on service security requirements. A smaller value indicates higher security but more inconvenient use. If this parameter is set to a small value, the client application may not send new requests in a timely manner after a transaction is started. As a result, the session times out and exits, affecting user experience. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:idle_in_transaction_timeout	Specifies the maximum idle duration (in seconds) of a transaction after the connection to the server is established and the transaction is started. The value 0 indicates that the timeout setting is disabled. A positive number indicates the maximum idle duration of a transaction after the connection to the server is established and the transaction is started. If the idle duration exceeds the value of this parameter, the client will be disconnected from the server. Set this parameter based on service security requirements. A smaller value indicates higher security but more inconvenient use. If this parameter is set to a small value, the client application may not send new requests in a timely manner after a transaction is started. As a result, the session times out and exits, affecting user experience. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_asp_standby	Specifies whether a standby node is enabled for ASP. If this function is enabled, the collected data of the standby node will also be sampled for generating an ASP report. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_asp_standby	Specifies whether a standby node is enabled for ASP. If this function is enabled, the collected data of the standby node will also be sampled for generating an ASP report. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:asp_flush_interval	Specifies the interval (in seconds) at which ASP flushes data. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:asp_flush_interval	Specifies the interval (in seconds) at which ASP flushes data. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:float_shortest_precision	Specifies whether to use the minimum precision method, instead of sprintf(), to serialize floating-point numbers. This minimum precision method is designed to enhance the performance of floating-point serialization system functions, specifically by reducing the single-threaded execution time of float4out() and float8out(). You are advised to set this parameter for new sites or sites that have performance requirements for floating-point number serialization. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:float_shortest_precision	Specifies whether to use the minimum precision method, instead of sprintf(), to serialize floating-point numbers. This minimum precision method is designed to enhance the performance of floating-point serialization system functions, specifically by reducing the single-threaded execution time of float4out() and float8out(). You are advised to set this parameter for new sites or sites that have performance requirements for floating-point number serialization. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_stream_pbe	Specifies whether the optimizer uses cache to optimize the STREAM distributed plan on DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_stream_pbe	Specifies whether the optimizer uses cache to optimize the STREAM distributed plan on DNs. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:gsi_fastpath_level	Specifies whether the optimizer executes the determination system of simple query rules of GSI. The simple query rules of GSI are as follows: A single table, no subquery, no sublink, or the target table is not a view. Query statements cannot generate an execution plan for a common index. Query statements cannot generate the IndexOnlyScan execution plan of GSI. Query statements do not contain scan hints, such as TableScan and IndexScan. The WHERE condition contains an equivalent condition that contains the index key of the candidate GSI (unique and not a partial GSI index). This parameter is supported in GaussDB V2.0-8.201 or later.
dn:gsi_fastpath_level	Specifies whether the optimizer executes the determination system of simple query rules of GSI. The simple query rules of GSI are as follows: A single table, no subquery, no sublink, or the target table is not a view. Query statements cannot generate an execution plan for a common index. Query statements cannot generate the IndexOnlyScan execution plan of GSI. Query statements do not contain scan hints, such as TableScan and IndexScan. The WHERE condition contains an equivalent condition that contains the index key of the candidate GSI (unique and not a partial GSI index). This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_case_intervalto numeric	Specifies whether INTERVAL can be converted to NUMERIC in CASE statements. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_case_intervalton umeric	Specifies whether INTERVAL can be converted to NUMERIC in CASE statements. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_wdr_cache	Adjusts the status of the GUC parameter enable_gssat_cache for WDR backend threads. If enable_wdr_cache is set to on , enable_gssat_cache will be set to on for WDR backend threads. Otherwise, enable_gssat_cache is set to off . This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_wdr_cache	Adjusts the status of the GUC parameter enable_gssstat_cache for WDR backend threads. If enable_wdr_cache is set to on , enable_gssstat_cache will be set to on for WDR backend threads. Otherwise, enable_gssstat_cache is set to off . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_uniq_idx_a_com pat	Specifies whether composite unique indexes are compatible with A databases for NULL values. This parameter is only applied in A-compatible databases. You are advised to retain the default value. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_uniq_idx_a_com pat	Specifies whether composite unique indexes are compatible with ORA databases for NULL values. This parameter is only applied in ORA-compatible databases. You are advised to retain the default value. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule	Specifies whether to enable the cost-based evaluation policy for a specified query rewriting rule. Some query rewriting rules support cost-based evaluation of whether to rewrite SQL statements. In this way, the kernel can generate better query plans and improve SQL execution efficiency. This parameter is mutually exclusive with the GUC parameter rewrite_rule . If both costbased_rewrite_rule and rewrite_rule are set, the rule configuration related to rewrite_rule does not take effect. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:costbased_rewrite_rule	Specifies whether to enable the cost-based evaluation policy for a specified query rewriting rule. Some query rewriting rules support cost-based evaluation of whether to rewrite SQL statements. In this way, the kernel can generate better query plans and improve SQL execution efficiency. This parameter is mutually exclusive with the GUC parameter rewrite_rule . If both costbased_rewrite_rule and rewrite_rule are set, the rule configuration related to rewrite_rule does not take effect. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:costbased_rewrite_rule_max_iterations	During plan generation for the SQL execution, if the number of conditions met for cost evaluation in the same rule exceeds the value of this parameter, the cost-based evaluation policy will be disabled for the extra conditions in the current request and converted to the rule-based rewriting policy. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:costbased_rewrite_rule_max_iterations	During plan generation for the SQL execution, if the number of conditions met for cost evaluation in the same rule exceeds the value of this parameter, the cost-based evaluation policy will be disabled for the extra conditions in the current request and converted to the rule-based rewriting policy. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule_timeout	When generating a plan for SQL execution, if the overall cost evaluation time using each rule exceeds the timeout interval (in milliseconds) specified by this parameter, the cost-based evaluation policy will be disabled for subsequent processes of the current request and the rule-based rewriting policy is used. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:costbased_rewrite_rule_timeout	When generating a plan for SQL execution, if the overall cost evaluation time using each rule exceeds the timeout interval (in milliseconds) specified by this parameter, the cost-based evaluation policy will be disabled for subsequent processes of the current request and the rule-based rewriting policy is used. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_ecpg_cursor_duplicate_operation	Specifies whether a cursor can be opened or closed repeatedly in A-compatible mode. This parameter takes effect only when ECPG is connected to A-compatible databases. If the database of this version is directly installed, set this parameter to on . If an earlier version is upgraded to the current version, set this parameter to off . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_ecpg_cursor_duplicate_operation	Specifies whether a cursor can be opened or closed repeatedly in ORA-compatible mode. This parameter takes effect only when ECPG is connected to ORA-compatible databases. If the database of this version is directly installed, set this parameter to on . If an earlier version is upgraded to the current version, set this parameter to off . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_remote_param	Specifies whether to enable the XC parameterized path. If this parameter is enabled, an XC parameterized path plan can be generated. In related sublink and nested loop join scenarios, if the outer table is small and the inner table is large, and the inner table has the leftmost matching index in the join condition, set this parameter to on to improve performance. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_remote_param	Specifies whether to enable the XC parameterized path. If this parameter is enabled, an XC parameterized path plan can be generated. In related sublink and nested loop join scenarios, if the outer table is small and the inner table is large, and the inner table has the leftmost matching index in the join condition, set this parameter to on to improve performance. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:standby_read_use_az_info	Specifies whether AZ information is used for the read on standby feature of GTM-Lite, so that the standby node in the same AZ as the CN is preferentially selected. This parameter is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_read_use_az_info	Specifies whether AZ information is used for the read on standby feature of GTM-Lite, so that the standby node in the same AZ as the CN is preferentially selected. This parameter is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_cached_operator	Specifies whether to enable operator execution state caching. If this parameter is disabled, the operator execution status cannot be cached, affecting query performance. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_cached_operator	Specifies whether to enable operator execution state caching. If this parameter is disabled, the operator execution status cannot be cached, affecting query performance. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:standby_read_use_load_balance	Specifies whether the read on standby feature of GTM-Lite uses load balancing to randomly select DN nodes that meet the consistency point (and AZ information). This function is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_read_use_load_balance	Specifies whether the read on standby feature of GTM-Lite uses load balancing to randomly select DN nodes that meet the consistency point (and AZ information). This function is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:m_format_dev_version	Specifies the compatibility configuration item of database platform minor versions. m_format_dev_version is only applied when sql_compatibility is set to M . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:m_format_dev_version	Specifies the compatibility configuration item of database platform minor versions. m_format_dev_version is only applied when sql_compatibility is set to M . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_smp_partitionwise	Specifies whether to enable partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_smp_partitionwise	Specifies whether to enable partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:force_smp_partitionwise_scan	Specifies whether to enable partition-wise parallel scan globally after enable_smp_partitionwise is enabled for using partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:force_smp_partitionwise_scan	Specifies whether to enable partition-wise parallel scan globally after enable_smp_partitionwise is enabled for using partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_fast_numeric_agg	Specifies whether to enable aggregation optimization for the numeric data type. If this parameter is disabled, the query performance of the corresponding scenarios may deteriorate. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_fast_numeric_agg	Specifies whether to enable aggregation optimization for the numeric data type. If this parameter is disabled, the query performance of the corresponding scenarios may deteriorate. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_err_stack	Specifies whether to print stack information for errors whose ACTION is "Contact Huawei technical support." This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_err_stack	Specifies whether to print stack information for errors whose ACTION is "Contact Huawei technical support." This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_show_err_code	Specifies whether to display error codes for front-end connections such as gsql. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_show_err_code	Specifies whether to display error codes for front-end connections such as gsql. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:blackbox_number	Specifies the maximum number of black box files that can be retained in the \$GAUSSLOG/gs_blackbox/{nodename} directory. If this parameter is set to 0 , the function is disabled and black box files are not retained, which may affect performance diagnosis. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:blackbox_number	Specifies the maximum number of black box files that can be retained in the \$GAUSSLOG/gs_blackbox/{nodename} directory. If this parameter is set to 0 , the function is disabled and black box files are not retained, which may affect performance diagnosis. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_fast_vecop	Specifies whether to enable vectorized calculation optimization for column-store tables. If it is enabled, column-store calculation of the Timestamp, int32, int64, and float8 data types is optimized, the row-store calling cost is reduced, and automatic vectorization calculation is supported. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_fast_vecop	Specifies whether to enable vectorized calculation optimization for column-store tables. If it is enabled, column-store calculation of the Timestamp, int32, int64, and float8 data types is optimized, the row-store calling cost is reduced, and automatic vectorization calculation is supported. This parameter is supported in GaussDB V2.0-8.201 or later.

Configurable Parameters for Distributed Instances of Version V2.0-8.x in Combined Deployment Mode

Table 14-2 Parameters for distributed instances (combined deployment)

Parameter	Description
recyclebin_retention_time	Specifies how long files will be kept in the recycle bin. Files in the recycle bin will be automatically deleted after this length of time. Unit: second
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. The value 0 indicates that there is no timeout limit. Unit: second
cn:page_version_partitions	Specifies the number of hash table partitions used for caching page version information in memory. This parameter directly affects the hash query efficiency and hash collision probability, and is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
datestyle	Specifies the display format for date and time.
max_replication_slots	Specifies the number of log replication slots on the primary node.
timezone	Specifies the time zone for displaying and interpreting time stamps.
cn:auto_increment_offset	Specifies the initial value of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + <i>N</i> × auto_increment_increment . <i>N</i> is a positive integer. If the value of this parameter is greater than that of auto_increment_increment , there will be an error when the values in the auto-increment column automatically increase.

Parameter	Description
dn:enable_xid_abort_check	Specifies whether to check the status of transaction ID rollback when a transaction is committed.
cn:audit_internal_event	Specifies whether to audit the connections and operations of internal tools cm_agent, gs_clean, and WDRXdb, and whether to audit the logins and logouts from CNs on DN.
enable_default_ustore_table	Specifies whether to enable the Ustore storage engine by default. If this parameter is set to on , all created tables are Ustore tables.
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots. Currently, only archive slots and backup slots are involved. Set this parameter to on only in cloud scenarios where logical replication is enabled. In other scenarios, set this parameter to off .
cn:enable_convert_illegal_character	Specifies whether the database supports characters not included the character sets.
cn:codegen_compile_thread_num	Specifies the number of Codegen compilation threads.
cn:enable_workload_rule	Specifies whether to enable SQL throttling.
dn:enable_enhance_toast_table	Specifies whether to use the enhanced TOAST out-of-line storage table. The value on indicates that the enhanced TOAST out-of-line storage table is used. The value off indicates that the TOAST out-of-line storage table is used.
dn:static_thread_pool_num	Specifies the number of threads used to create a static thread pool (static pool). This parameter takes effect only on CNs of a distributed instance after enable_thread_pool is enabled.
enable_stream_operator	Specifies the query optimizer's use of streams. If enable_stream_operator is disabled, a large number of logs indicating that the plans cannot be pushed down are recorded. If you do not need these logs, you are advised to disable both enable_unshipping_log and enable_stream_operator . The default value is recommended.

Parameter	Description
cn:auto_increment_increment	Specifies the auto-increment step of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is smaller than that of auto_increment_offset , there will be an error when the values in the auto-increment column automatically increase.
dn:support_binary_copy_version	Specifies whether to verify the binary file encoding information when data is imported using COPY FROM in BINARY mode. If forward compatibility is required, leave this parameter empty. Otherwise, retain the default value.
cn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.
cn:gs_perf_retention_days	Specifies how many days the flame graph files in HTML format are retained. Unit: day
dn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
cn:sql_mode	Specifies the SQL behavior control configuration item in M-compatible mode.
dn:enable_rls_match_index	Specifies whether indexes of a base table can be scanned based on target predicate conditions in row-level security scenarios. Target scenario: Row level security (RLS) policies are set and enabled in the base table, and the query predicate contains the unleakproof system function or like operator.

Parameter	Description
dn:audit_internal_event	Specifies whether to audit the connections and operations of internal tools cm_agent, gs_clean, and WDRXdb, and whether to audit the logins and logouts from CNs on DNs.
cn:llvm_max_memory	Specifies the upper limit of the memory used by IRs (including cached and in-use IRs) generated during compilation in Codegen execution mode. The memory used by Codegen is not applied for by preoccupation. It is a part of max_dynamic_memory and is restricted by the llvm_max_memory parameter. Unit: KB
dn:fix_func_selection	Specifies whether to optimize the function matching policy. The value catlist indicates the catlist sequence is optimized. (The non-B-compatible mode has been optimized. In non-B-compatible mode, system functions are always preferentially selected and executed. The policy in B-compatible mode is the same as that in versions earlier than 505.1.0. An error message indicating that the function is not unique may be displayed, or a system function may be selected for execution.)
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.
comm_no_delay	Specifies whether to use the no_delay attribute of a communication library connection.
enable_recyclebin	Enables or disables the recycle bin in real time.
dn:sql_mode	Specifies the SQL behavior control configuration item in M-compatible mode.
cn:instr_unique_sql_combination_options	Specifies the configuration items of combining unique SQL statements of the same type.
dn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.

Parameter	Description
dn:verify_log_buffers	Specifies the size or pages of verifyLog buffers in memory mode. The unit is 8 KB. For example, if the value of this parameter is 4, the requested memory is 4 x 8 KB = 32 KB. This parameter is valid only when page_version_check is set to persistence . If page_version_check is set to another value, the parameter value will still be sent to the kernel, but the relevant function does not take effect until after page_version_check is set to persistence .
cn:convert_illegal_char_mode	Specifies the placeholders of invalid characters that can be displayed on the client when enable_convert_illegal_char is enabled. Value range: 95 characters whose decimal codes range from 32 to 126 in the ASCII coding table.
dn:llvm_max_memory	Specifies the upper limit of the memory used by IRs (including cached and in-use IRs) generated during compilation in Codegen execution mode. The memory used by Codegen is not applied for by preoccupation. It is a part of max_dynamic_memory and is restricted by the llvm_max_memory parameter. Unit: KB
cn:page_version_check	Specifies the type of page version verification. off indicates that page version verification is disabled. memory indicates that page version verification in pure memory mode is enabled. The page version information will be lost after a restart. persistence indicates that persistent page version verification is enabled. The page version information will not be lost after a restart.
undo_space_limit_size	Specifies the threshold for forcibly recycling undo space. When the undo space usage reaches 80% of the threshold, forcible recycling starts. It is recommended that the value of this parameter be greater than or equal to the value of undo_limit_size_per_transaction . Unit: 8 KB
cn:tde_index_default_encrypt	When tde_index_default_encrypt is set to on and an index is created based on an encrypted table, the database automatically converts the index to an encrypted index.
recovery_time_target	Specifies the time for the standby node to write and replay logs. Unit: second
dn:enable_convert_illegal_char	Specifies whether the database supports characters not included the character sets.

Parameter	Description
dn:system_view_version	Determines the version of the system view. All versions are backward compatible. For example, when system_view_version is set to 3 , all features of version 2 and version 1 are also supported. For details, see the product documentation.
dn:enable_workload_rule	Specifies whether to enable SQL throttling.
dn:archive_interval	Specifies the archiving interval. Log files are forcibly archived when the period specified by this parameter has elapsed. A large value of this parameter affects the RPO of PITR. The default value is recommended. Unit: second
cn:enable_codegen	Specifies whether code optimization can be enabled. Currently, code optimization uses the LLVM optimization.
dn:codegen_compile_thread_num	Specifies the number of Codegen compilation threads.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DN connects to primary DN to obtain physical logs, and logical replication tools connect to primary DN to obtain logical logs. If this parameter is set to a value smaller than 20, scale-out may fail. The value of this parameter must be smaller than that of max_connections .
dn:resilience_ctrlstmt_detect_timelimit	Specifies the execution time of a normal SQL statement that will be marked as a slow SQL statement. The value 0 indicates that slow SQL statements are not identified. A value greater than 0 indicates that a normal SQL statement whose execution time exceeds the value of this parameter is marked as a slow SQL statement. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. Unit: millisecond
cn:a_format_date_timestamp	Specifies whether to enable current_date , current_timestamp , and localtimestamp to return the system time, instead of the transaction start time, when a transaction starts.

Parameter	Description
cn:page_missing_dirty_check	Specifies whether to enable the verification for pages not marked as dirty. The verification checks whether the modified pages are not marked as dirty. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
cn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.
dn:max_compile_functions	Specifies the maximum number of function compilation results stored in the server. Excessive functions and compilation results of stored procedures may occupy large memory space. Setting this parameter to an appropriate value can reduce the memory usage and improve system performance. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
dn:tde_index_default_encrypt	When tde_index_default_encrypt is set to on and an index is created based on an encrypted table, the database automatically converts the index to an encrypted index.

Parameter	Description
dn:recovery_parse_workers	Specifies the number of ParseRedoRecord threads in the ultimate RTO feature. This parameter must be used together with recovery_redo_workers . If both recovery_parse_workers and recovery_redo_workers are greater than 1, ultimate RTO is enabled. If you do not want to enable ultimate RTO, retain the default value 1 for recovery_parse_workers . When enabling ultimate RTO, ensure that replication_type is set to 1. If both ultimate RTO and parallel replay are enabled, only ultimate RTO takes effect, and parallel replay is ineffective. Ultimate RTO does not support column-store tables. Therefore, disable ultimate RTO in a system where column-store tables are used or are to be used. Ultimate RTO also does not support flow control. Flow control is determined by the recovery_time_target parameter.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
undo_retention_time	Specifies how long undo logs are kept. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content. Unit: second
cn:wdr_snapshot_space_threshold	Specifies the threshold for controlling the space used by snapshots. When the space used by snapshots reaches 80% of the value of this parameter, the control logic of the database is enabled to stabilize the space usage. Unit: KB
dn:page_version_check	Specifies the type of page version verification. off indicates that page version verification is disabled. memory indicates that page version verification in pure memory mode is enabled. The page version information will be lost after a restart. persistence indicates that persistent page version verification is enabled. The page version information will not be lost after a restart.

Parameter	Description
cn:max_compile_functions	Specifies the maximum number of function compilation results stored in the server. Excessive functions and compilation results of stored procedures may occupy large memory space. Setting this parameter to an appropriate value can reduce the memory usage and improve system performance. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
cn:page_version_max_num	Specifies the maximum number of page version information entries that can be cached in memory. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
dn:auto_increment_offset	Specifies the initial value of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is greater than that of auto_increment_increment , there will be an error when the values in the auto-increment column automatically increase.

Parameter	Description
dn:auto_increment_cache	Specifies the number of reserved auto-increment cache values when auto-increment is triggered by batch insertion or import of auto-increment columns. When auto-increment values are reserved, the auto-increment counter value is updated to the maximum auto-increment cache value. Before the cache values are used up, the auto-increment counter value remains unchanged, and the triggered auto-increment uses the cache values. If this parameter is set to 0, the auto-increment cache values are automatically reserved. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, 2^n values are reserved each time. For example, one value is reserved in the first auto-increment, two values are reserved in the second auto-increment, four values are reserved in the third auto-increment, and eight values are reserved in the fourth auto-increment. However, if the number of reserved values exceeds 65,535, 65,535 values are reserved. If this parameter is not set to 0, the number of reserved cache values is the value of this parameter. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, the value of auto_increment_cache is the number of auto-increment values reserved each time. The reserved cache values are valid only in the statement. If the reserved auto-increment cache values are used up and subsequent INSERT statements trigger auto-increment based on the auto-increment counter, the values in the auto-increment column in the table are discontinuous. This parameter does not affect the auto-increment column in the local temporary table.
dn:enable_codegen	Specifies whether code optimization can be enabled. Currently, code optimization uses the LLVM optimization.
dn:instr_unique_sql_combination_options	Specifies the configuration items of combining unique SQL statements of the same type. If this feature is enabled, the IDs of unique SQL statements of the same type are normalized, and the generated unique SQL strings are normalized.
rewrite_rule	Sets query rewriting rules.

Parameter	Description
dn:page_missing_dirty_check	Specifies whether to enable the verification for pages not marked as dirty. The verification checks whether the modified pages are not marked as dirty. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
dn:wdr_snapshot_full_backup_interval	Specifies the interval at which a full WDR snapshot is created. The interval specified by this parameter is about a number instead of time. For example, if the parameter is set to 12 , a full snapshot and then 11 incremental snapshots are generated for each group. If the parameter is set to 1 , all snapshots generated are full snapshots.
dn:wdr_snapshot_space_threshold	Specifies the threshold for controlling the space used by snapshots. When the space used by snapshots reaches 80% of the value of this parameter, the control logic of the database is enabled to stabilize the space usage. Unit: KB
cn:gs_perf_interval	Specifies the automatic perf data collection interval. The value 0 indicates that the collection is stopped. If the value is greater than 0 and less than 5, the value 5 is used. Unit: minute
dn:gs_perf_interval	Specifies the automatic perf data collection interval. The value 0 indicates that the collection is stopped. If the value is greater than 0 and less than 5, the value 5 is used. Unit: minute
cn:disable_keyword_options	Specifies the configuration items for database compatibility. Multiple items are separated by commas (,). An identifier with this parameter set will not be used as a keyword.
track_stmt_stat_level	Controls the level of statement execution tracking.
cn:b_format_dev_version	Specifies the compatibility configuration item of database platform minor versions.
dn:copy_special_character_version	Specifies whether to report an error when there are invalid characters during data import and export using COPY FROM.

Parameter	Description
cn:page_version_recycler_thread_num	Specifies the number of threads for recycling and verifying page version information. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
cn:wdr_snapshot_full_backup_interval	Specifies the interval at which a full WDR snapshot is created. The interval specified by this parameter is about a number instead of time. For example, if the parameter is set to 12 , a full snapshot and then 11 incremental snapshots are generated for each group. If the parameter is set to 1 , all snapshots generated are full snapshots.
cn:system_view_version	Determines the version of the system view. All versions are backward compatible. For example, when system_view_version is set to 3 , all features of version 2 and version 1 are also supported. For details, see the product documentation.
support_batch_bind	Specifies whether to bind and execute PBE (Parse, Bind, Execute) statements in batches through interfaces such as JDBC, ODBC, and libpq.
cn:check_disconnect_query	Specifies whether to enable the function of terminating statement execution on the server after the client is disconnected due to timeout.
password_effect_time	Specifies the validity period of an account password. Unit: day
cn:verify_log_buffers	Specifies the size or pages of verifyLog buffers in memory mode. The unit is 8 KB. For example, if the value of this parameter is 4, the requested memory is 4 x 8 KB = 32 KB. This parameter is valid only when page_version_check is set to persistence . If page_version_check is set to another value, the parameter value will still be sent to the kernel, but the relevant function does not take effect until after page_version_check is set to persistence .
cn:enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots. Currently, only archive slots and backup slots are involved. Set this parameter to on only in cloud scenarios where logical replication is enabled. In other scenarios, set this parameter to off .

Parameter	Description
undo_limit_size_per_transaction	Specifies the threshold for forcibly recycling undo space. When the undo space usage reaches 80% of the threshold, forcible recycling starts. It is recommended that the value of this parameter be greater than or equal to the value of undo_limit_size_per_transaction . Unit: 8 KB
dn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
cn:enable_rls_match_index	Specifies whether indexes of a base table can be scanned based on target predicate conditions in row-level security scenarios. Target scenario: Row level security (RLS) policies are set and enabled in the base table, and the query predicate contains the unleakproof system function or like operator.
dn:auto_increment_increment	Specifies the auto-increment step of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is smaller than that of auto_increment_offset , there will be an error when the values in the auto-increment column automatically increase.
cn:copy_special_character_version	Specifies whether to report an error when there are invalid characters during data import and export using COPY FROM.
behavior_compat_options	Specifies database compatibility configuration items. After the value of proc_outparam_override is changed, the database must be connected again or the instance must be rebooted. Otherwise, stored procedures and functions cannot be correctly called.
dn:gs_format_behavior_compat_options	Specifies the configuration items of GaussDB internal system functions.
dn:b_format_dev_version	Specifies the compatibility configuration item of database platform minor versions.
dn:check_disconnect_query	Specifies whether to enable the function of terminating statement execution on the server after the client is disconnected due to timeout.
cn:gs_format_behavior_compat_options	Specifies the configuration items of GaussDB internal system functions.

Parameter	Description
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.
cn:static_thread_pool_num	Specifies the number of threads used to create a static thread pool (static pool). This parameter takes effect only on CNs of a distributed instance after enable_thread_pool is enabled.
audit_system_object	Specifies whether to audit the CREATE, DROP, and ALTER operations on GaussDB database objects. GaussDB database objects include databases, users, schemas, and tables. You can change the value of this parameter to audit only the operations on required database objects. In the scenario where the leader node is forcibly selected, you are advised to set audit_system_object to the maximum value and audit all DDL objects. For details about the value range, see the product documentation.
cn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
dn:enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots. Currently, only archive slots and backup slots are involved. Set this parameter to on only in cloud scenarios where logical replication is enabled. In other scenarios, set this parameter to off .
dn:page_version_partitions	Specifies the number of hash table partitions used for caching page version information in memory. This parameter directly affects the hash query efficiency and hash collision probability, and is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.

Parameter	Description
dn:page_version_recycler_thread_num	Specifies the number of threads for recycling and verifying page version information. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
dn:dcf_thread_effective_time	Specifies the effective time of the DCF flushing thread. This parameter is used to determine whether the disk I/O hangs. If the DCF cannot access I/O resources within the period specified by this parameter, the DCF considers that the thread I/O hangs and a primary/standby switchover is triggered. If this parameter is set to 0 , I/O hang detection is disabled. Unit: second
dn:a_format_date_timestamp	Specifies whether to enable current_date , current_timestamp , and localtimestamp to return the system time, instead of the transaction start time, when a transaction starts.
cn:enable_xid_abort_check	Specifies whether to check the status of transaction ID rollback when a transaction is committed.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second
dn:disable_keyword_options	Specifies the configuration items for database compatibility. Multiple items are separated by commas (,). An identifier with this parameter set will not be used as a keyword.
cn:fix_func_selection	Specifies whether to optimize the function matching policy. The value catlist indicates the catlist sequence is optimized. (The non-B-compatible mode has been optimized. In non-B-compatible mode, system functions are always preferentially selected and executed. The policy in B-compatible mode is the same as that in versions earlier than 505.1.0. An error message indicating that the function is not unique may be displayed, or a system function may be selected for execution.)
autoanalyze	Specifies whether to automatically collect statistics on tables that have no statistics when a plan is generated.

Parameter	Description
password_lock_time	Specifies the lockout duration (in seconds) of an account after it exceeds the maximum number of failed login attempts specified by failed_login_attempts . The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
cn:support_binary_copy_version	Specifies whether to verify the binary file encoding information when data is imported using COPY FROM in BINARY mode. If forward compatibility is required, leave this parameter empty. Otherwise, retain the default value.
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond
cn:recovery_parse_workers	Specifies the number of ParseRedoRecord threads in the ultimate RTO feature. This parameter must be used together with recovery_redo_workers . If both recovery_parse_workers and recovery_redo_workers are greater than 1, ultimate RTO is enabled. If you do not want to enable ultimate RTO, retain the default value 1 for recovery_parse_workers . When enabling ultimate RTO, ensure that replication_type is set to 1. If both ultimate RTO and parallel replay are enabled, only ultimate RTO takes effect, and parallel replay is ineffective. Ultimate RTO does not support column-store tables. Therefore, disable ultimate RTO in a system where column-store tables are used or are to be used. Ultimate RTO also does not support flow control. Flow control is determined by the recovery_time_target parameter.

Parameter	Description
cn:auto_increment_cache	Specifies the number of reserved auto-increment cache values when auto-increment is triggered by batch insertion or import of auto-increment columns. When auto-increment values are reserved, the auto-increment counter value is updated to the maximum auto-increment cache value. Before the cache values are used up, the auto-increment counter value remains unchanged, and the triggered auto-increment uses the cache values. If this parameter is set to 0, the auto-increment cache values are automatically reserved. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, 2^n values are reserved each time. For example, one value is reserved in the first auto-increment, two values are reserved in the second auto-increment, four values are reserved in the third auto-increment, and eight values are reserved in the fourth auto-increment. However, if the number of reserved values exceeds 65,535, 65,535 values are reserved. If this parameter is not set to 0, the number of reserved cache values is the value of this parameter. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, the value of auto_increment_cache is the number of auto-increment values reserved each time. The reserved cache values are valid only in the statement. If the reserved auto-increment cache values are used up and subsequent INSERT statements trigger auto-increment based on the auto-increment counter, the values in the auto-increment column in the table are discontinuous. This parameter does not affect the auto-increment column in the local temporary table.
cn:enable_enhance_toast_table	Specifies whether to use the enhanced TOAST out-of-line storage table. The value on indicates that the enhanced TOAST out-of-line storage table is used. The value off indicates that the TOAST out-of-line storage table is used.
cn:archive_interval	Specifies the archiving interval. Log files are forcibly archived when the period specified by this parameter has elapsed. A large value of this parameter affects the RPO of PITR. The default value is recommended. Unit: second

Parameter	Description
dn:convert_illegal_char_mode	Specifies the placeholders of invalid characters that can be displayed on the client when enable_convert_illegal_char is enabled. Value range: 95 characters whose decimal codes range from 32 to 126 in the ASCII coding table.
dn:use_workload_manager	Specifies whether to enable resource management. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:use_workload_manager	Specifies whether to enable resource management. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_save_datachanged_timestamp	Specifies whether to record the time when INSERT, UPDATE, DELETE, or EXCHANGE/TRUNCATE/DROP PARTITION is performed on table data. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_save_datachanged_timestamp	Specifies whether to record the time when INSERT, UPDATE, DELETE, or EXCHANGE/TRUNCATE/DROP PARTITION is performed on table data. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:max_process_memory	Specifies the maximum physical memory allowed for a DN, in KB. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:max_process_memory	Specifies the maximum physical memory allowed for a CN, in KB. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_shared_buffers_fraction	Specifies the shared_buffers proportion used on the server where a standby instance is deployed. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:standby_shared_buffers_fraction	Specifies the shared_buffers proportion used on the server where a standby instance is deployed. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:pagewriter_sleep	Specifies the interval (in milliseconds) for the page writer thread to flush dirty pages to disks after the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:pagewriter_sleep	Specifies the interval (in milliseconds) for the page writer thread to flush dirty pages to disks after the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:xloginsert_locks	Specifies the number of locks on concurrent write-ahead logging. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:xloginsert_locks	Specifies the number of locks on concurrent write-ahead logging. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_light_proxy	Specifies whether the optimizer optimizes the execution of simple queries on CNs. This parameter does not take effect if the character set of the application side does not match that of the kernel side. You are advised to set the character set to UTF-8 when creating a database. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_light_proxy	Specifies whether the optimizer optimizes the execution of simple queries on CNs. This parameter does not take effect if the character set of the application side does not match that of the kernel side. You are advised to set the character set to UTF-8 when creating a database. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:synchronous_commit	Specifies the synchronization mode of the current transaction. Generally, logs generated by a transaction are synchronized in the following sequence: The primary node writes the logs to the local memory. The primary node writes logs in the local memory to the local file system. The primary node flushes the logs in the local file system to disks. The primary node sends the logs to the standby node. The standby node receives the logs and saves them to its local memory. The standby node writes the logs in the local memory to the local file system. The standby node flushes the logs in the local file system to disks. The standby node replays the logs to complete the incremental update of data files. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:synchronous_commit	Specifies the synchronization mode of the current transaction. Generally, logs generated by a transaction are synchronized in the following sequence: The primary node writes the logs to the local memory. The primary node writes logs in the local memory to the local file system. The primary node flushes the logs in the local file system to disks. The primary node sends the logs to the standby node. The standby node receives the logs and saves them to its local memory. The standby node writes the logs in the local memory to the local file system. The standby node flushes the logs in the local file system to disks. The standby node replays the logs to complete the incremental update of data files. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_incremental_checkpoint	Specifies whether to enable incremental checkpoint. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_incremental_checkpoint	Specifies whether to enable incremental checkpoint. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_gsstat_cache	Specifies whether to enable cache acceleration for PgStat functions. PgStat functions include pg_stat_get_numscans, pg_stat_get_tuples_returned, pg_stat_get_tuples_fetched, pg_stat_get_tuples_inserted, pg_stat_get_tuples_updated, pg_stat_get_tuples_deleted, pg_stat_get_tuples_hot_updated, pg_stat_get_live_tuples, pg_stat_get_dead_tuples, pg_stat_get_autovacuum_count, pg_stat_get_autoanalyze_count, pg_stat_get_blocks_fetched, pg_stat_get_blocks_hit, pg_timestamp stat_get_last_autoanalyze_time, pg_timestamp stat_get_last_autovacuum_time, and pg_timestamp stat_get_last_updated. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:bypass_workload_manager	Enables or disables independent I/O control. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bypass_workload_manager	Enables or disables independent I/O control. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_resource_track	Specifies whether to enable real-time resource monitoring. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_resource_track	Specifies whether to enable real-time resource monitoring. This parameter must be applied on both CNs and DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_rt_percentile	Specifies whether to enable the function of calculating the response time of 80% and 95% of the SQL statements in the system. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_instr_rt_percentile	Specifies whether to enable the function of calculating the response time of 80% and 95% of the SQL statements in the system. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_cpu_timer	Specifies whether to capture the CPU time consumed during SQL statement execution. In the x86-based centralized deployment scenario where 32 vCPUs and 256 GB of memory are configured, the performance tested by BenchmarkSQL 5.0 fluctuates by about 3.5% by enabling or disabling this parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_instr_cpu_timer	Specifies whether to capture the CPU time consumed during SQL statement execution. In the x86-based centralized deployment scenario where 32 vCPUs and 256 GB of memory are configured, the performance tested by BenchmarkSQL 5.0 fluctuates by about 3.5% by enabling or disabling this parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:track_counts	Determines whether to enable collection of statistics on database activities. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:track_counts	Determines whether to enable collection of statistics on database activities. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_user_metric_persistent	Specifies whether to dump the historical monitoring data of user resources. If this parameter is set to on , data in the PG_TOTAL_USER_RESOURCE_INFO view is periodically sampled and saved to the system catalog GS_WLM_USER_RESOURCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_user_metric_persistent	Specifies whether to dump the historical monitoring data of user resources. If this parameter is set to on , data in the PG_TOTAL_USER_RESOURCE_INFO view is periodically sampled and saved to the system catalog GS_WLM_USER_RESOURCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:log_hostname	By default, connection log messages only show the IP address of the connecting host. The host name can be recorded when this parameter is set to on . It may take some time to parse the host name. Therefore, the database performance may be affected. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:log_hostname	By default, connection log messages only show the IP address of the connecting host. The host name can be recorded when this parameter is set to on . It may take some time to parse the host name. Therefore, the database performance may be affected. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_track_wait	Specifies whether to enable real-time collection of wait event information. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_instr_track_wait	Specifies whether to enable real-time collection of wait event information. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instance_metric_persistent	Specifies whether to dump instance resource monitoring data. When this parameter is set to on , the instance monitoring data is saved to the system catalog GS_WLM_INSTANCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_instance_metric_persistent	Specifies whether to dump instance resource monitoring data. When this parameter is set to on , the instance monitoring data is saved to the system catalog GS_WLM_INSTANCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_logical_io_statistics	Specifies whether to collect logical I/O statistics during resource monitoring. If this function is enabled, the read_kbytes , write_kbytes , read_counts , write_counts , read_speed , and write_speed columns in the PG_TOTAL_USER_RESOURCE_INFO view will collect statistics on the byte count, number of times, and speed of logical read and write. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_logical_io_statistics	Specifies whether to collect logical I/O statistics during resource monitoring. If this function is enabled, the read_kbytes , write_kbytes , read_counts , write_counts , read_speed , and write_speed columns in the PG_TOTAL_USER_RESOURCE_INFO view will collect statistics on the byte count, number of times, and speed of logical read and write. Columns related to logical read and write in the system catalogs GS_WLM_USER_RESOURCE_HISTORY and GS_WLM_INSTANCE_HISTORY will collect statistics on the logical read and write of related users and instances. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_sort	Specifies whether the query optimizer uses sort methods. It is impossible to suppress explicit sorts entirely, but setting this variable to off encourages the optimizer to choose other methods if available. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_sort	Specifies whether the query optimizer uses sort methods. It is impossible to suppress explicit sorts entirely, but setting this variable to off encourages the optimizer to choose other methods if available. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_material	Specifies whether the query optimizer uses materialization. It is impossible to suppress materialization entirely, but setting this variable to off prevents the optimizer from inserting materialized nodes. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_material	Specifies whether the query optimizer uses materialization. It is impossible to suppress materialization entirely, but setting this variable to off prevents the optimizer from inserting materialized nodes. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
gtm:gtm_max_trans	Specifies the maximum number of connections accepted by the GTM. You are advised not to change the value. This parameter is supported in GaussDB V2.0-8.201 or later.
gtm:csn_sync_interval	Specifies the interval for synchronizing CSNs between the primary and standby GTMs, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
gtm:restore_duration	Specifies the maximum number of transactions (identified by XIDs or CSNs) that the GTM will process in a single restoration cycle. This parameter is supported in GaussDB V2.0-8.201 or later.
cms:coordinator_heartbeat_timeout	Specifies the heartbeat timeout (in seconds) that triggers the automatic removal of faulty CNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:client_min_messages	Specifies which level of messages will be sent to the client. A higher level covers the messages of all the lower levels. The lower the level is, the fewer messages are sent. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:client_min_messages	Specifies which level of messages will be sent to the client. A higher level covers the messages of all the lower levels. The lower the level is, the fewer messages are sent. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:pagewriter_thread_num	Specifies the number of threads for background page flushing after the incremental checkpoint is enabled. Dirty pages are flushed in sequence to disks, updating recovery points. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:pagewriter_thread_num	Specifies the number of threads for background page flushing after the incremental checkpoint is enabled. Dirty pages are flushed in sequence to disks, updating recovery points. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:bgwriter_delay	Specifies the interval at which the background writer writes dirty shared buffers, in milliseconds. Each time, the background writer initiates write operations for some dirty buffers. In full checkpoint mode, the bgwriter_lru_maxpages parameter is used to control the amount of data to be written each time, and the process is restarted after the period of hibernation specified by bgwriter_delay (in milliseconds). In incremental checkpoint mode, the number of target idle buffer pages is calculated based on the value of candidate_buf_percent_target . If the number of idle buffer pages is insufficient, a batch of pages are flushed to disks at the interval specified by bgwriter_delay (in milliseconds). The number of flushed pages is calculated based on the target difference percentage. The maximum number of flushed pages is limited by max_io_capacity . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:bgwriter_delay	Specifies the interval at which the background writer writes dirty shared buffers, in milliseconds. Each time, the background writer initiates write operations for some dirty buffers. In full checkpoint mode, the bgwriter_lru_maxpages parameter is used to control the amount of data to be written each time, and the process is restarted after the period of hibernation specified by bgwriter_delay (in milliseconds). In incremental checkpoint mode, the number of target idle buffer pages is calculated based on the value of candidate_buf_percent_target . If the number of idle buffer pages is insufficient, a batch of pages are flushed to disks at the interval specified by bgwriter_delay (in milliseconds). The number of flushed pages is calculated based on the target difference percentage. The maximum number of flushed pages is limited by max_io_capacity . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:max_files_per_process	Specifies the maximum number of files that can be opened simultaneously by each server process. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:candidate_buf_percent_target	Specifies the proportion (%) of available buffers in shared_buffer in the candidate buffer chain when the incremental checkpoint is enabled. If the percentage is less than the value of this parameter, the bgwriter thread starts to flush dirty pages based on the max_io_capacity parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:candidate_buf_percent_target	Specifies the proportion (%) of available buffers in shared_buffer in the candidate buffer chain when the incremental checkpoint is enabled. If the percentage is less than the value of this parameter, the bgwriter thread starts to flush dirty pages based on the max_io_capacity parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:checkpoint_segments	Specifies the minimum number of WAL segment files in the period specified by checkpoint_timeout . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:checkpoint_segments	Specifies the minimum number of WAL segment files in the period specified by checkpoint_timeout . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:checkpoint_timeout	Specifies the maximum time between automatic WAL checkpoints, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:checkpoint_timeout	Specifies the maximum time between automatic WAL checkpoints, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:incremental_checkpoint_timeout	Specifies the maximum interval (in seconds) between automatic WAL checkpoints when the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:incremental_checkpoint_timeout	Specifies the maximum interval (in seconds) between automatic WAL checkpoints when the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:wal_writer_delay	Specifies the delay (in milliseconds) between activity rounds for the WAL writer. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:wal_writer_delay	Specifies the delay (in milliseconds) between activity rounds for the WAL writer. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:wal_buffers	Specifies the shared memory for storing WALs, that is, multiples of XLOG_BLCKSZ , in increments of 8 KB. For details about the value range, see wal_buffers . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:wal_buffers	Specifies the shared memory for storing WALs, that is, multiples of XLOG_BLCKSZ , in increments of 8 KB. For details about the value range, see wal_buffers . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bulk_write_ring_size	Specifies the size of the ring buffer used by the operation when a large amount of data is written (for example, the copy operation). This parameter is supported in GaussDB V2.0-8.201 or later.
cn:bulk_write_ring_size	Specifies the size of a ring buffer used for parallel data import. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:recovery_redo_workers	Specifies the number of PageRedoWorker threads corresponding to each ParseRedoRecord thread when ultimate RTO is enabled. recovery_redo_workers must be used together with recovery_parse_workers . The value of recovery_redo_workers takes effect only when recovery_parse_workers is greater than 1. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:recovery_redo_workers	Specifies the number of PageRedoWorker threads corresponding to each ParseRedoRecord thread when ultimate RTO is enabled. recovery_redo_workers must be used together with recovery_parse_workers . The value of recovery_redo_workers takes effect only when recovery_parse_workers is greater than 1. This parameter is supported in GaussDB V2.0-8.201 or later.
cms:phony_dead_effective_time	Specifies the maximum number of times DN processes are detected as zombie. If the number of times a process is detected as zombie is greater than the specified value, the process is considered to be a zombie process and will be restarted. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_ai_watchdog	Enables or disables the AI watchdog function. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_ai_watchdog	Enables or disables the AI watchdog function. This parameter is supported in GaussDB V2.0-8.201 or later.
enable_unshipping_log	Specifies whether to log statements that are not pushed down. The logs help locate performance issues that may be caused by statements not pushed down. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:max_connections	Specifies the maximum number of concurrent connections. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:max_connections	Specifies the maximum number of concurrent connections. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:resilience_memory_reject_percent	Specifies the dynamic memory usage for escape from memory overload. This parameter is only applied when use_workload_manager and enable_memory_limit are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:resilience_memory_reject_percent	Specifies the dynamic memory usage for escape from memory overload. This parameter is only applied when use_workload_manager and enable_memory_limit are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:resilience_threadpool_reject_cond	Specifies the proportion of accumulated sessions in the thread pool for escape from overload. This parameter is only applied when enable_thread_pool and use_workload_manager are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:resilience_threadpool_reject_cond	Specifies the proportion of accumulated sessions in the thread pool for escape from overload. This parameter is only applied when enable_thread_pool and use_workload_manager are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:vacuum_cost_delay	Specifies how long the process sleeps when vacuum_cost_limit has been exceeded, in milliseconds. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:vacuum_cost_delay	Specifies how long the process sleeps when vacuum_cost_limit has been exceeded, in milliseconds. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:b_format_version	Specifies the SQL behavior control configuration item in B-compatible mode. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:b_format_version	Specifies the forward compatibility in MySQL-compatible mode. This is a configuration item for controlling database platform compatibility behavior. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_control_group	Specifies whether to enable the Cgroups. This parameter must be applied on both CNs and DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_control_group	Specifies whether to enable the Cgroups. This parameter must be applied on both CNs and DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:track_stmt_parameter	After track_stmt_parameter is enabled, the executed statements recorded in statement_history are not normalized. The complete SQL statement information can be displayed to help the database administrator locate faults. For a simple query, the complete statement information is displayed. For a PBE statement, the complete statement information and information about each variable value are displayed. The format is "query string; parameters: \$1=value1,\$2=value2, ...". This parameter is used to display full SQL information for users and is not controlled by the track_activity_query_size parameter. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:track_stmt_parameter	After track_stmt_parameter is enabled, the executed statements recorded in statement_history are not normalized. The complete SQL statement information can be displayed to help the database administrator locate faults. For a simple query, the complete statement information is displayed. For a PBE statement, the complete statement information and information about each variable value are displayed. The format is "query string; parameters: \$1=value1,\$2=value2, ...". This parameter is used to display full SQL information for users and is not controlled by the track_activity_query_size parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:track_activity_query_size	Specifies byte counts of the current running commands used to trace each active session. If the number of bytes in a command is greater than the value of this parameter, the command will be truncated. This parameter affects the memory usage of multiple system views. Adjusting the value of this parameter has a great impact on the memory usage. You can estimate the memory usage based on the value calculated from instr_unique_sql_count times track_activity_query_size . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:track_activity_query_size	Specifies byte counts of the current running commands used to trace each active session. If the number of bytes in a command is greater than the value of this parameter, the command will be truncated. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_lock_account	Specifies whether to automatically lock an account when the number of consecutive login failures using incorrect passwords reaches the value of failed_login_attempts . Manual locking by administrators is not controlled by this parameter. Setting this parameter to off reduces account security. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_lock_account	Specifies whether to automatically lock an account when the number of consecutive login failures using incorrect passwords reaches the value of failed_login_attempts . Manual locking by administrators is not controlled by this parameter. Setting this parameter to off reduces account security. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:instr_unique_sql_track_type	Specifies how unique SQL statements are recorded. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:instr_unique_sql_track_type	Specifies how unique SQL statements are recorded. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:idle_in_transaction_timeout	Specifies the maximum idle duration (in seconds) of a transaction after the connection to the server is established and the transaction is started. The value 0 indicates that the timeout setting is disabled. A positive number indicates the maximum idle duration of a transaction after the connection to the server is established and the transaction is started. If the idle duration exceeds the value of this parameter, the client will be disconnected from the server. Set this parameter based on service security requirements. A smaller value indicates higher security but more inconvenient use. If this parameter is set to a small value, the client application may not send new requests in a timely manner after a transaction is started. As a result, the session times out and exits, affecting user experience. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:idle_in_transaction_timeout	Specifies the maximum idle duration (in seconds) of a transaction after the connection to the server is established and the transaction is started. The value 0 indicates that the timeout setting is disabled. A positive number indicates the maximum idle duration of a transaction after the connection to the server is established and the transaction is started. If the idle duration exceeds the value of this parameter, the client will be disconnected from the server. Set this parameter based on service security requirements. A smaller value indicates higher security but more inconvenient use. If this parameter is set to a small value, the client application may not send new requests in a timely manner after a transaction is started. As a result, the session times out and exits, affecting user experience. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_asp_standby	Specifies whether a standby node is enabled for ASP. If this function is enabled, the collected data of the standby node will also be sampled for generating an ASP report. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_asp_standby	Specifies whether a standby node is enabled for ASP. If this function is enabled, the collected data of the standby node will also be sampled for generating an ASP report. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:asp_flush_interval	Specifies the interval (in seconds) at which ASP flushes data. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:asp_flush_interval	Specifies the interval (in seconds) at which ASP flushes data. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:float_shortest_precision	Specifies whether to use the minimum precision method, instead of sprintf(), to serialize floating-point numbers. This minimum precision method is designed to enhance the performance of floating-point serialization system functions, specifically by reducing the single-threaded execution time of float4out() and float8out(). You are advised to set this parameter for new sites or sites that have performance requirements for floating-point number serialization. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:float_shortest_precision	Specifies whether to use the minimum precision method, instead of sprintf(), to serialize floating-point numbers. This minimum precision method is designed to enhance the performance of floating-point serialization system functions, specifically by reducing the single-threaded execution time of float4out() and float8out(). You are advised to set this parameter for new sites or sites that have performance requirements for floating-point number serialization. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_stream_pbe	Specifies whether the optimizer uses cache to optimize the STREAM distributed plan on DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_stream_pbe	Specifies whether the optimizer uses cache to optimize the STREAM distributed plan on DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:gsi_fastpath_level	Specifies whether the optimizer executes the determination system of simple query rules of GSI. The simple query rules of GSI are as follows: A single table, no subquery, no sublink, or the target table is not a view. Query statements cannot generate an execution plan for a common index. Query statements cannot generate the IndexOnlyScan execution plan of GSI. Query statements do not contain scan hints, such as TableScan and IndexScan. The WHERE condition contains an equivalent condition that contains the index key of the candidate GSI (unique and not a partial GSI index). This parameter is supported in GaussDB V2.0-8.201 or later.
dn:gsi_fastpath_level	Specifies whether the optimizer executes the determination system of simple query rules of GSI. The simple query rules of GSI are as follows: A single table, no subquery, no sublink, or the target table is not a view. Query statements cannot generate an execution plan for a common index. Query statements cannot generate the IndexOnlyScan execution plan of GSI. Query statements do not contain scan hints, such as TableScan and IndexScan. The WHERE condition contains an equivalent condition that contains the index key of the candidate GSI (unique and not a partial GSI index). This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_case_intervalton umeric	Specifies whether INTERVAL can be converted to NUMERIC in CASE statements. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_case_intervalton umeric	Specifies whether INTERVAL can be converted to NUMERIC in CASE statements. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_wdr_cache	Adjusts the status of the GUC parameter enable_gsstat_cache for WDR backend threads. If enable_wdr_cache is set to on , enable_gsstat_cache will be set to on for WDR backend threads. Otherwise, enable_gsstat_cache is set to off . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_wdr_cache	Adjusts the status of the GUC parameter enable_gsstat_cache for WDR backend threads. If enable_wdr_cache is set to on , enable_gsstat_cache will be set to on for WDR backend threads. Otherwise, enable_gsstat_cache is set to off . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_uniq_idx_a_com pat	Specifies whether composite unique indexes are compatible with A databases for NULL values. This parameter is only applied in A-compatible databases. You are advised to retain the default value. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_uniq_idx_a_com pat	Specifies whether composite unique indexes are compatible with ORA databases for NULL values. This parameter is only applied in ORA-compatible databases. You are advised to retain the default value. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule	Specifies whether to enable the cost-based evaluation policy for a specified query rewriting rule. Some query rewriting rules support cost-based evaluation of whether to rewrite SQL statements. In this way, the kernel can generate better query plans and improve SQL execution efficiency. This parameter is mutually exclusive with the GUC parameter rewrite_rule . If both costbased_rewrite_rule and rewrite_rule are set, the rule configuration related to rewrite_rule does not take effect. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:costbased_rewrite_rule	Specifies whether to enable the cost-based evaluation policy for a specified query rewriting rule. Some query rewriting rules support cost-based evaluation of whether to rewrite SQL statements. In this way, the kernel can generate better query plans and improve SQL execution efficiency. This parameter is mutually exclusive with the GUC parameter rewrite_rule . If both costbased_rewrite_rule and rewrite_rule are set, the rule configuration related to rewrite_rule does not take effect. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule_max_iterations	During plan generation for the SQL execution, if the number of conditions met for cost evaluation in the same rule exceeds the value of this parameter, the cost-based evaluation policy will be disabled for the extra conditions in the current request and converted to the rule-based rewriting policy. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:costbased_rewrite_rule_max_iterations	During plan generation for the SQL execution, if the number of conditions met for cost evaluation in the same rule exceeds the value of this parameter, the cost-based evaluation policy will be disabled for the extra conditions in the current request and converted to the rule-based rewriting policy. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:costbased_rewrite_rule_timeout	When generating a plan for SQL execution, if the overall cost evaluation time using each rule exceeds the timeout interval (in milliseconds) specified by this parameter, the cost-based evaluation policy will be disabled for subsequent processes of the current request and the rule-based rewriting policy is used. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:costbased_rewrite_rule_timeout	When generating a plan for SQL execution, if the overall cost evaluation time using each rule exceeds the timeout interval (in milliseconds) specified by this parameter, the cost-based evaluation policy will be disabled for subsequent processes of the current request and the rule-based rewriting policy is used. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_ecpg_cursor_duplicate_operation	Specifies whether a cursor can be opened or closed repeatedly in A-compatible mode. This parameter takes effect only when ECPG is connected to A-compatible databases. If the database of this version is directly installed, set this parameter to on . If an earlier version is upgraded to the current version, set this parameter to off . This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_ecpg_cursor_duplicate_operation	Specifies whether a cursor can be opened or closed repeatedly in ORA-compatible mode. This parameter takes effect only when ECPG is connected to ORA-compatible databases. If the database of this version is directly installed, set this parameter to on . If an earlier version is upgraded to the current version, set this parameter to off . This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_remote_param	Specifies whether to enable the XC parameterized path. If this parameter is enabled, an XC parameterized path plan can be generated. In related sublink and nested loop join scenarios, if the outer table is small and the inner table is large, and the inner table has the leftmost matching index in the join condition, set this parameter to on to improve performance. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_remote_param	Specifies whether to enable the XC parameterized path. If this parameter is enabled, an XC parameterized path plan can be generated. In related sublink and nested loop join scenarios, if the outer table is small and the inner table is large, and the inner table has the leftmost matching index in the join condition, set this parameter to on to improve performance. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:standby_read_use_az_info	Specifies whether AZ information is used for the read on standby feature of GTM-Lite, so that the standby node in the same AZ as the CN is preferentially selected. This parameter is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_read_use_az_info	Specifies whether AZ information is used for the read on standby feature of GTM-Lite, so that the standby node in the same AZ as the CN is preferentially selected. This parameter is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_cached_operator	Specifies whether to enable operator execution state caching. If this parameter is disabled, the operator execution status cannot be cached, affecting query performance. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_cached_operator	Specifies whether to enable operator execution state caching. If this parameter is disabled, the operator execution status cannot be cached, affecting query performance. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:standby_read_use_load_balance	Specifies whether the read on standby feature of GTM-Lite uses load balancing to randomly select DN's that meet the consistency point (and AZ information). This function is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_read_use_load_balance	Specifies whether the read on standby feature of GTM-Lite uses load balancing to randomly select DN's that meet the consistency point (and AZ information). This function is disabled by default. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:m_format_dev_version	Specifies the compatibility configuration item of database platform minor versions. m_format_dev_version is only applied when sql_compatibility is set to M . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:m_format_dev_version	Specifies the compatibility configuration item of database platform minor versions. m_format_dev_version is only applied when sql_compatibility is set to M . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_smp_partitionwise	Specifies whether to enable partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_smp_partitionwise	Specifies whether to enable partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:force_smp_partitionwise_scan	Specifies whether to enable partition-wise parallel scan globally after enable_smp_partitionwise is enabled for using partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:force_smp_partitionwise_scan	Specifies whether to enable partition-wise parallel scan globally after enable_smp_partitionwise is enabled for using partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_fast_numeric_agg	Specifies whether to enable aggregation optimization for the numeric data type. If this parameter is disabled, the query performance of the corresponding scenarios may deteriorate. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_fast_numeric_agg	Specifies whether to enable aggregation optimization for the numeric data type. If this parameter is disabled, the query performance of the corresponding scenarios may deteriorate. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_err_stack	Specifies whether to print stack information for errors whose ACTION is "Contact Huawei technical support." This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_err_stack	Specifies whether to print stack information for errors whose ACTION is "Contact Huawei technical support." This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_show_err_code	Specifies whether to display error codes for front-end connections such as gsql. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:enable_show_err_code	Specifies whether to display error codes for front-end connections such as gsql. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:blackbox_number	Specifies the maximum number of black box files that can be retained in the \$GAUSSLOG/gs_blackbox/{nodename} directory. If this parameter is set to 0 , the function is disabled and black box files are not retained, which may affect performance diagnosis. This parameter is supported in GaussDB V2.0-8.201 or later.
cn:blackbox_number	Specifies the maximum number of black box files that can be retained in the \$GAUSSLOG/gs_blackbox/{nodename} directory. If this parameter is set to 0 , the function is disabled and black box files are not retained, which may affect performance diagnosis. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_fast_vecop	Specifies whether to enable vectorized calculation optimization for column-store tables. If it is enabled, column-store calculation of the Timestamp, int32, int64, and float8 data types is optimized, the row-store calling cost is reduced, and automatic vectorization calculation is supported. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
cn:enable_fast_vecop	Specifies whether to enable vectorized calculation optimization for column-store tables. If it is enabled, column-store calculation of the Timestamp, int32, int64, and float8 data types is optimized, the row-store calling cost is reduced, and automatic vectorization calculation is supported. This parameter is supported in GaussDB V2.0-8.201 or later.

Configurable Parameters for Centralized Instances of Version V2.0-8.x

Table 14-3 Parameters for centralized instances

Parameter	Description
dn:qrw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:recovery_max_workers	Specifies the number of concurrent replayer threads.
dn:enable_auto_clean_unique_sql	Specifies whether to enable automatic cleaning of unique SQL statements in hash tables.
dn:gs_perf_retention_days	Specifies how many days the flame graph files in HTML format are retained. Unit: day
dn:page_version_max_num	Specifies the maximum number of page version information entries that can be cached in memory. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
log_autovacuum_min_duration	Specifies the interval which should elapse before autovacuum operations are logged. Autovacuum operations equal to or beyond the specified interval will be logged. If it is set to 0 , all autovacuum operations will be logged. If it is set to -1 , no autovacuum operations will be logged.

Parameter	Description
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
dn:max_connections	Specifies the maximum number of concurrent connections to DNs. The value of this parameter must be greater than that of max_wal_senders .
datestyle	Specifies the display format for date and time.
max_replication_slots	Specifies the number of log replication slots on the primary node.
timezone	Specifies the time zone for displaying and interpreting time stamps.
dn:enable_xid_abort_check	Specifies whether to check the status of transaction ID rollback when a transaction is committed.
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots. Currently, only archive slots and backup slots are involved. Set this parameter to on only in cloud scenarios where logical replication is enabled. In other scenarios, set this parameter to off .
dn:enable_enhance_toast_table	Specifies whether to use the enhanced TOAST out-of-line storage table. The value on indicates that the enhanced TOAST out-of-line storage table is used. The value off indicates that the TOAST out-of-line storage table is used.
dn:support_binary_copy_verification	Specifies whether to verify the binary file encoding information when data is imported using COPY FROM in BINARY mode.
dn:recovery_time_target	Specifies the time for the standby node to write and replay logs. Unit: second
plat_compat_allow_public_key_retrieval	Specifies the database configuration item in M-compatible mode. This parameter specifies whether a client can request the RSA public key. on : The kernel allows the client to request the RSA public key for password transmission encryption. off : The client is not allowed to request the RSA public key.

Parameter	Description
dn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
dn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
dn:enable_rls_match_index	Specifies whether indexes of a base table can be scanned based on target predicate conditions in row-level security scenarios. Target scenario: Row level security (RLS) policies are set and enabled in the base table, and the query predicate contains the unleakproof system function or like operator.
dn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
dn:audit_internal_event	Specifies whether to audit the connections and operations of internal tools cm_agent, gs_clean, and WDRXdb, and whether to audit the logins and logouts from CNs on DN.
dn:numa_distribute_mode	Specifies the distribution of some shared data and threads among NUMA nodes. This parameter is used to optimize the performance of large-scale Arm servers with multiple NUMA nodes. Generally, you do not need to set this parameter. This parameter cannot be set to all for instances deployed on x86-based servers.
dn:max_compile_packages	Specifies the maximum number of package compilation results stored in the server. Default values for different specifications are calculated by rounding down (max_process_memory × 2%)/4.4, in MB.

Parameter	Description
dn:fix_func_selection	Specifies whether to optimize the function matching policy. The value catlist indicates the catlist sequence is optimized. (The non-B-compatible mode has been optimized. In non-B-compatible mode, system functions are always preferentially selected and executed. The policy in B-compatible mode is the same as that in versions earlier than 505.1.0. An error message indicating that the function is not unique may be displayed, or a system function may be selected for execution.)
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.
dn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
dn:sql_mode	Specifies the SQL behavior control configuration item in M-compatible mode.
dn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.
dn:undo_space_limit_size	Specifies the threshold for forcibly recycling undo space. When the undo space usage reaches 80% of the threshold, forcible recycling starts. It is recommended that the value of this parameter be greater than or equal to the value of undo_limit_size_per_transaction . Unit: 8 KB
dn:umdk_enabled	Specifies whether UMDK is enabled for the primary and standby DNs of the current instance. If the UMDK protocol is used for communication between the primary and standby DNs, the related log keyword on DNs is umdk . If the TCP protocol is used for communication between the primary and standby DNs, logs are recorded.

Parameter	Description
dn:verify_log_buffers	Specifies the size or pages of verifyLog buffers in memory mode. The unit is 8 KB. For example, if the value of this parameter is 4, the requested memory is 4 x 8 KB = 32 KB. This parameter is valid only when page_version_check is set to persistence . If page_version_check is set to another value, the parameter value will still be sent to the kernel, but the relevant function does not take effect until after page_version_check is set to persistence .
dn:llvm_max_memory	Specifies the upper limit of the memory used by IRs (including cached and in-use IRs) generated during compilation in Codegen execution mode. The memory used by Codegen is not applied for by preoccupation. It is a part of max_dynamic_memory and is restricted by the llvm_max_memory parameter. Unit: KB
dn:enable_control_group	Specifies whether to enable the Cgroups.
dn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
dn:enable_convert_illegal_char	Specifies whether the database supports characters not included the character sets.
dn:system_view_version	Determines the version of the system view. All versions are backward compatible. For example, when system_view_version is set to 3 , all features of version 2 and version 1 are also supported. For details, see the product documentation.
track_activity_query_size	Specifies the maximum number of bytes to be logged for each SQL statement. If the number of bytes of a SQL statement exceeds the specified parameter value, the SQL statement will be truncated. Setting this parameter to a large value can lead to excessive memory usage, potentially causing the system to run out of memory. You are advised to set this parameter to a value no more than 4096.
dn:enable_workload_rule	Specifies whether to enable SQL throttling.
dn:archive_interval	Specifies the archiving interval. Log files are forcibly archived when the period specified by this parameter has elapsed. A large value of this parameter affects the RPO of PITR. The default value is recommended. Unit: second

Parameter	Description
div_precision_increment	Specifies the database configuration item in M-compatible mode. This is a session-level parameter, which is used to set the value of precision that the division result can improve. The final precision is the precision of the first operation parameter added by the value of this parameter.
lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond
dn:adaptive_cost_min_time	Specifies the execution duration threshold of SQL statements for cardinality feedback collection. Only the feedback of statements whose execution duration is greater than the value of this parameter is collected. Unit: millisecond
dn:audit_dml_state_select	Determines whether to audit the SELECT operation.
dn:codegen_compile_thread_num	Specifies the number of Codegen compilation threads.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DNs connect to primary DNs to obtain physical logs, and logical replication tools connect to primary DNs to obtain logical logs. If this parameter is set to a value smaller than 20, scale-out may fail. The value of this parameter must be smaller than that of max_connections .
dn:max_concurrent_autonomous_transactions	Specifies the maximum number of autonomous transaction connections, that is, the maximum number of concurrent autonomous transactions executed at the same time. If this parameter is set to 0 , autonomous transactions cannot be executed. The theoretical maximum value is 10000 . Set this parameter based on workload requirements and hardware configurations. It is recommended that this parameter be set to a value less than or equal to 1/10 of max_connections .
dn:enable_extension	Specifies whether database extensions can be created. This parameter can be used only in specific scenarios after evaluation. Generally, this parameter does not need to be adjusted.

Parameter	Description
dn:max_compile_functions	Specifies the maximum number of function compilation results stored in the server. Excessive functions and compilation results of stored procedures may occupy large memory space. Setting this parameter to an appropriate value can reduce the memory usage and improve system performance. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
dn:auto_explain_log_min_duration	Specifies the minimum duration of execution plans that are automatically printed. Only execution plans whose duration is greater than the value of auto_explain_log_min_duration will be printed. Unit: millisecond
dn:enable_default_ustore_table	Specifies whether to enable the Ustore storage engine by default. If this parameter is set to on , all created tables are Ustore tables.
dn:enable_vacuum_control	Specifies whether to move the database permanent thread autoVacuumWorker to the Vacuum Cgroup.
dn:recovery_parse_workers	Specifies the number of ParseRedoRecord threads in the ultimate RTO feature. This parameter must be used together with recovery_redo_workers . If both recovery_parse_workers and recovery_redo_workers are greater than 1 , ultimate RTO is enabled. If you do not want to enable ultimate RTO, retain the default value 1 for recovery_parse_workers . When enabling ultimate RTO, ensure that replication_type is set to 1 . If both ultimate RTO and parallel replay are enabled, only ultimate RTO takes effect, and parallel replay is ineffective. Ultimate RTO does not support column-store tables. Therefore, disable ultimate RTO in a system where column-store tables are used or are to be used. Ultimate RTO also does not support flow control. Flow control is determined by the recovery_time_target parameter.
dn:page_version_check	Specifies the type of page version verification. off indicates that page version verification is disabled. memory indicates that page version verification in pure memory mode is enabled. The page version information will be lost after a restart. persistence indicates that persistent page version verification is enabled. The page version information will not be lost after a restart.

Parameter	Description
dn:num_internal_lock_partitions	Specifies the number of internal lightweight lock partitions. Changing the value of this parameter affects performance and memory usage. Before modifying this parameter, determine your use case and fully verify the change in a test environment. For details, see the reference document.
dn:enable_auto_explain	Specifies whether to automatically print execution plans. This parameter is used to locate slow stored procedures or slow queries.
dn:audit_function_exec	Specifies whether to record the audit information during the execution of the stored procedures, anonymous blocks, or user-defined functions (excluding system functions). The value 0 means to disable the function, and 1 means to enable it.

Parameter	Description
dn:auto_increment_cache	Specifies the number of reserved auto-increment cache values when auto-increment is triggered by batch insertion or import of auto-increment columns. When auto-increment values are reserved, the auto-increment counter value is updated to the maximum auto-increment cache value. Before the cache values are used up, the auto-increment counter value remains unchanged, and the triggered auto-increment uses the cache values. If this parameter is set to 0, the auto-increment cache values are automatically reserved. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, 2^n values are reserved each time. For example, one value is reserved in the first auto-increment, two values are reserved in the second auto-increment, four values are reserved in the third auto-increment, and eight values are reserved in the fourth auto-increment. However, if the number of reserved values exceeds 65,535, 65,535 values are reserved. If this parameter is not set to 0, the number of reserved cache values is the value of this parameter. When auto-increment is triggered for the first time, if the number of rows to be inserted into the auto-increment column is known, the number is the reserved value. If the number of rows is unknown, the value of auto_increment_cache is the number of auto-increment values reserved each time. The reserved cache values are valid only in the statement. If the reserved auto-increment cache values are used up and subsequent INSERT statements trigger auto-increment based on the auto-increment counter, the values in the auto-increment column in the table are discontinuous. This parameter does not affect the auto-increment column in the local temporary table.
dn:enable_codegen	Specifies whether code optimization can be enabled. Currently, code optimization uses the LLVM optimization.
dn:instr_unique_sql_combination_options	Specifies the configuration items of combining unique SQL statements of the same type. If this feature is enabled, the IDs of unique SQL statements of the same type are normalized, and the generated unique SQL strings are normalized.

Parameter	Description
plan_cache_mode	Specifies the policy of generating and caching execution plans of prepared statements.
dn:group_concat_max_len	Specifies the maximum permitted result length in bytes for the GROUP_CONCAT() function.
dn:hadr_recovery_point_target	Specifies the time allowed for the standby instance to flush logs to disks in streaming DR. This ensures that the log difference between the primary and standby instances is controlled within the period specified by hadr_recovery_point_target during the switchover and the standby instance can be promoted to primary. If this parameter is set to a small value, the performance of the primary node is affected. If it is set to a large value, the log flow is not effectively controlled. The value 0 indicates that log flow control is disabled. Unit: second
dn:behavior_compat_options	Specifies database compatibility configuration items. After the value of proc_outparam_override is changed, the database must be connected again or the instance must be rebooted. Otherwise, stored procedures and functions cannot be correctly called.
dn:page_missing_dirty_check	Specifies whether to enable the verification for pages not marked as dirty. The verification checks whether the modified pages are not marked as dirty. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
dn:wdr_snapshot_full_backup_interval	Specifies the interval at which a full WDR snapshot is created. The interval specified by this parameter is about a number instead of time. For example, if the parameter is set to 12 , a full snapshot and then 11 incremental snapshots are generated for each group. If the parameter is set to 1 , all snapshots generated are full snapshots.
dn:vacuum_defer_cleanup_age	Specifies the number of transactions used by VACUUM.
cms:datastorage_threshold_value_check	Specifies the disk usage threshold to put a database node into read-only mode. If the disk usage of a data directory exceeds this threshold, the database node is automatically changed to read-only. Unit: percentage (%)

Parameter	Description
dn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48.
dn:wdr_snapshot_space_threshold	Specifies the threshold for controlling the space used by snapshots. When the space used by snapshots reaches 80% of the value of this parameter, the control logic of the database is enabled to stabilize the space usage. Unit: KB
dn:m_format_behavior_compat_options	Specifies the configuration items for the M-compatible mode.
dn:gs_perf_interval	Specifies the automatic perf data collection interval. The value 0 indicates that the collection is stopped. If the value is greater than 0 and less than 5, the value 5 is used. Unit: minute
dn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
max_allowed_packet	Specifies the database configuration item in M-compatible mode. This parameter indicates the upper limit of the return value of a string function. The value must be a multiple of 1,024. Unit: KB
track_stmt_stat_level	Controls the level of statement execution tracking.
dn:max_standby_archive_delay	Specifies the wait period before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode. -1 indicates that the standby node waits until the conflicting queries are complete. Unit: millisecond
dn:copy_special_character_version	Specifies whether to report an error when there are invalid characters during data import and export using COPY FROM.
log_temp_files	Specifies whether to log temporary file details when they are deleted. A positive value indicates that the deletion information of temporary files whose size is larger than the value specified by log_temp_files is recorded. 0 indicates that the deletion information of all temporary files is recorded. -1 indicates that the deletion information of any temporary files is not recorded. Unit: KB
enable_wdr_snapshot	Specifies whether to enable WDR snapshots. If this parameter is enabled, the kernel periodically takes performance snapshots.

Parameter	Description
dn:m_format_dev_version	Specifies the compatibility configuration item of database platform minor versions.
password_effect_time	Specifies the validity period of an account password. Unit: day
dn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
dn:immediate_analyze_threshold	Specifies the threshold for automatically analyzing inserted data. When the amount of inserted data reaches the original data amount multiplied by the value of immediate_analyze_threshold , and the total number of rows exceeds 100, ANALYZE is automatically triggered.
dn:auto_increment_increment	Specifies the auto-increment step of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is smaller than that of auto_increment_offset , there will be an error when the values in the auto-increment column automatically increase.
dn:gs_format_behavior_compat_options	Specifies the configuration items of GaussDB internal system functions.
enable_nestloop	Controls whether the query optimizer uses the nested-loop join plan type to fully scan inner tables.
dn:b_format_dev_version	Specifies the compatibility configuration item of database platform minor versions.
dn:check_disconnect_query	Specifies whether to enable the function of terminating statement execution on the server after the client is disconnected due to timeout.
dn:wal_keep_segments	Specifies the minimum number of transaction log files stored in the pg_xlog directory. Standby nodes obtain the logs from the primary node to perform streaming replication.
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.
dn:max_standby_streaming_delay	Specifies the wait period before queries on the standby node are canceled when the queries conflict with WAL data receiving through streaming replication in hot standby mode. -1 indicates that the standby node waits until the conflicting queries are complete. Unit: millisecond

Parameter	Description
audit_system_object	Specifies whether to audit the CREATE, DROP, and ALTER operations on GaussDB database objects. GaussDB database objects include databases, users, schemas, and tables. You can change the value of this parameter to audit only the operations on required database objects. In the scenario where the leader node is forcibly selected, you are advised to set audit_system_object to the maximum value and audit all DDL objects. For details about the value range, see the product documentation.
dn:enable_copy_server_files	Specifies whether to enable the privilege to copy server files. If this parameter is set to on , users with the SYSADMIN privilege or users who inherit the built-in role privilege gs_role_copy_files are allowed to run the COPY FROM FILENAME or COPY TO FILENAME command. If it is set to off , only the initial user is allowed to run the COPY FROM FILENAME or COPY TO FILENAME statement. This parameter is supported in GaussDB V2.0-8.201 or later.
checkpoint_segments	Specifies the minimum number of WAL segment files in the period specified by checkpoint_timeout .
dn:page_version_partitions	Specifies the number of hash table partitions used for caching page version information in memory. This parameter directly affects the hash query efficiency and hash collision probability, and is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off . For details about the value range, see the product documentation.
dn:password_encryption_type	Specifies how user passwords are encrypted. 0 : Passwords are encrypted using MD5. 1 : Passwords are encrypted using SHA-256 and MD5. 2 : Passwords are encrypted using SHA-256. 3 : Passwords are encrypted using SM3. MD5 is not recommended because it is a weak encryption algorithm.
dn:enable_plsql_opfusion	Specifies whether to enable SQLBypass for stored procedures.

Parameter	Description
dn:page_version_recycler_thread_num	Specifies the number of threads for recycling and verifying page version information. This parameter is only valid when page_version_check is not set to off . If page_version_check is set to off , the parameter value will still be sent to the kernel, but the relevant function is not enabled until after page_version_check is set to a value other than off .
dn:dcf_thread_effective_time	Specifies the effective time of the DCF flushing thread. This parameter is used to determine whether the disk I/O hangs. If the DCF cannot access I/O resources within the period specified by this parameter, the DCF considers that the thread I/O hangs and a primary/standby switchover is triggered. If this parameter is set to 0 , I/O hang detection is disabled. Unit: second
dn:audit_login_logout	Specifies whether to audit users' logins (including successful and failed logins) and logouts. 0 : Disable user login and logout auditing. 1 : Audit only successful logins. 2 : Audit only failed logins. 3 : Audit both successful and failed logins. 4 : Audit only user logouts. 5 : Audit only user logouts and successful logins. 6 : Audit only user logouts and failed logins. 7 : Audit successful and failed logins, as well as user logouts.
dn:a_format_date_timestamp	Specifies whether to enable current_date , current_timestamp , and localtimestamp to return the system time, instead of the transaction start time, when a transaction starts.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second
temp_file_limit	Specifies the maximum amount of disk space that a session can use for temporary files. The value -1 indicates there is no limit. Unit: KB
dn:disable_keyword_options	Specifies the configuration items for database compatibility. Multiple items are separated by commas (,). An identifier with this parameter set will not be used as a keyword.

Parameter	Description
dn:hadr_recovery_time_target	Specifies whether the standby DB instance completes log writing and replay in streaming DR mode. If this parameter is set to a small value, the performance of the primary node is affected. If it is set to a large value, the log flow is not effectively controlled. The value 0 indicates that log flow control is disabled. Unit: second
password_lock_time	Specifies the duration for a locked account to be automatically unlocked. Unit: day
wdr_snapshot_interval	Specifies the interval at which the background thread Snapshot automatically takes snapshots of the database monitoring data. Unit: minute
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond
dn:wal_file_preinit_bounds	Specifies the maximum number of WAL segment files that can be pre-expanded by the WAL writer auxiliary thread per second during service running. The WAL segment file size is 16 MB. If this parameter is set to 0 , there is no restriction.
dn:enable_feedback_cardes t	Specifies whether to enable the feedback-based optimizer cardinality and cost correction functions. This parameter is used by developers to diagnose model-related problems. If enable_adaptive_cost is set to off and this parameter is set to on , the operator information is still collected and the API of cardinality estimation feedback is still called. However, in this case, the thread for background automatic model maintenance will not be enabled. You can use the gs_acm_analyze_workload_manual() function to manually train models for diagnosing problems.
dn:enable_global_plsqlcach e	Specifies whether to globally cache compilation products of packages, stored procedures, and functions, and cache execution products at the session level. Enabling this function can reduce the memory usage of database nodes in high concurrency scenarios.
dn:convert_illegal_char_mo de	Specifies the placeholders of invalid characters that can be displayed on the client when enable_convert_illegal_char is enabled. Value range: 95 characters whose decimal codes range from 32 to 126 in the ASCII coding table.

Parameter	Description
dn:advance_xlog_file_num	Specifies the number of Xlog files that are periodically initialized in advance in the background. This parameter is used to prevent the Xlog file initialization from affecting the performance during transaction commit. However, such a fault may occur only when the system is overloaded. Therefore, you do not need to set this parameter.
enable_mergejoin	Controls whether the query optimizer uses the merge-join plan type.
dn:tde_index_default_encrypt	When tde_index_default_encrypt is set to on and an index is created based on an encrypted table, the database automatically converts the index to an encrypted index.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
undo_retention_time	Specifies how long undo logs are kept. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content. Unit: second
enable_global_syscache	Determines whether to enable global system cache.
dn:max_io_capacity	Specifies the maximum I/O per second for the background writer process to flush pages in batches. Set this parameter based on the service scenario and the disk I/O capability. If the RTO is short or the data volume is many times that of the shared memory and the service access data volume is random, the value of this parameter cannot be too small. A small value of max_io_capacity reduces the number of pages flushed by the background writer. If a large number of pages are evicted due to service triggering, the services are affected. Unit: KB
dn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.

Parameter	Description
index_txntype	Sets the index page format to PCR or RCR. This parameter is left unconfigured during system initialization. By default, the created indexes are compatible with the index type (RCR) of earlier versions. Once this parameter is specified, it cannot be left unconfigured again.
dn:auto_increment_offset	Specifies the initial value of an auto-increment column. The auto-increment value is calculated by the following formula: auto_increment_offset + $N \times$ auto_increment_increment . N is a positive integer. If the value of this parameter is greater than that of auto_increment_increment , there will be an error when the values in the auto-increment column automatically increase.
autovacuum_naptime	Specifies the minimum delay between autovacuum runs on any given database. If this parameter is set to a smaller value, the load is more stable but the I/O increases. If this parameter is set to a larger value, the load may fluctuate more but the I/O decreases. Unit: second
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. The value 0 indicates that there is no timeout limit. Unit: second
dn:use_workload_manager	Specifies whether to enable resource management. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_save_datachanged_timestamp	Specifies whether to record the time when INSERT, UPDATE, DELETE, or EXCHANGE/TRUNCATE/DROP PARTITION is performed on table data. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:max_process_memory	Specifies the maximum physical memory allowed for a DN, in KB. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:standby_shared_buffers_fraction	Specifies the shared_buffers proportion used on the server where a standby instance is deployed. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:pagewriter_sleep	Specifies the interval (in milliseconds) for the page writer thread to flush dirty pages to disks after the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:xloginsert_locks	Specifies the number of locks on concurrent write-ahead logging. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:synchronous_commit	Specifies the synchronization mode of the current transaction. Generally, logs generated by a transaction are synchronized in the following sequence: The primary node writes the logs to the local memory. The primary node writes logs in the local memory to the local file system. The primary node flushes the logs in the local file system to disks. The primary node sends the logs to the standby node. The standby node receives the logs and saves them to its local memory. The standby node writes the logs in the local memory to the local file system. The standby node flushes the logs in the local file system to disks. The standby node replays the logs to complete the incremental update of data files. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_incremental_checkpoint	Specifies whether to enable incremental checkpoint. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_rt_percentile	Specifies whether to enable the function of calculating the response time of 80% and 95% of the SQL statements in the system. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_cpu_timer	Specifies whether to capture the CPU time consumed during SQL statement execution. In the x86-based centralized deployment scenario where 32 vCPUs and 256 GB of memory are configured, the performance tested by BenchmarkSQL 5.0 fluctuates by about 3.5% by enabling or disabling this parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_user_metric_persistent	Specifies whether to dump the historical monitoring data of user resources. If this parameter is set to on , data in the PG_TOTAL_USER_RESOURCE_INFO view is periodically sampled and saved to the system catalog GS_WLM_USER_RESOURCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:log_hostname	By default, connection log messages only show the IP address of the connecting host. The host name can be recorded when this parameter is set to on . It may take some time to parse the host name. Therefore, the database performance may be affected. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instr_track_wait	Specifies whether to enable real-time collection of wait event information. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_instance_metric_persistent	Specifies whether to dump instance resource monitoring data. When this parameter is set to on , the instance monitoring data is saved to the system catalog GS_WLM_INSTANCE_HISTORY . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_logical_io_statistics	Specifies whether to collect logical I/O statistics during resource monitoring. If this function is enabled, the read_kbytes , write_kbytes , read_counts , write_counts , read_speed , and write_speed columns in the PG_TOTAL_USER_RESOURCE_INFO view will collect statistics on the byte count, number of times, and speed of logical read and write. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:client_min_messages	Specifies which level of messages will be sent to the client. A higher level covers the messages of all the lower levels. The lower the level is, the fewer messages are sent. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:pagewriter_thread_num	Specifies the number of threads for background page flushing after the incremental checkpoint is enabled. Dirty pages are flushed in sequence to disks, updating recovery points. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:bgwriter_delay	Specifies the interval at which the background writer writes dirty shared buffers, in milliseconds. Each time, the background writer initiates write operations for some dirty buffers. In full checkpoint mode, the bgwriter_lru_maxpages parameter is used to control the amount of data to be written each time, and the process is restarted after the period of hibernation specified by bgwriter_delay (in milliseconds). In incremental checkpoint mode, the number of target idle buffer pages is calculated based on the value of candidate_buf_percent_target . If the number of idle buffer pages is insufficient, a batch of pages are flushed to disks at the interval specified by bgwriter_delay (in milliseconds). The number of flushed pages is calculated based on the target difference percentage. The maximum number of flushed pages is limited by max_io_capacity . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:candidate_buf_percent_target	Specifies the proportion (%) of available buffers in shared_buffer in the candidate buffer chain when the incremental checkpoint is enabled. If the percentage is less than the value of this parameter, the bgwriter thread starts to flush dirty pages based on the max_io_capacity parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
checkpoint_segments	Specifies the minimum number of WAL segment files in the period specified by checkpoint_timeout . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:checkpoint_timeout	Specifies the maximum time between automatic WAL checkpoints, in seconds. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:incremental_checkpoint_timeout	Specifies the maximum interval (in seconds) between automatic WAL checkpoints when the incremental checkpoint is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:wal_writer_delay	Specifies the delay (in milliseconds) between activity rounds for the WAL writer. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:wal_buffers	Specifies the shared memory for storing WALs, that is, multiples of XLOG_BLCKSZ , in increments of 8 KB. For details about the value range, see wal_buffers . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:walwriter_cpu_bind	Specifies the CPU core bound to the WAL write thread. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bulk_write_ring_size	Specifies the size of the ring buffer used by the operation when a large amount of data is written (for example, the copy operation). This parameter is supported in GaussDB V2.0-8.201 or later.
dn:recovery_redo_workers	Specifies the number of PageRedoWorker threads corresponding to each ParseRedoRecord thread when ultimate RTO is enabled. recovery_redo_workers must be used together with recovery_parse_workers . The value of recovery_redo_workers takes effect only when recovery_parse_workers is greater than 1. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_ai_watchdog	Enables or disables the AI watchdog function. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:max_connections	Specifies the maximum number of concurrent connections. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:resilience_memory_reject_percent	Specifies the dynamic memory usage for escape from memory overload. This parameter is only applied when use_workload_manager and enable_memory_limit are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:resilience_threadpool_reject_cond	Specifies the proportion of accumulated sessions in the thread pool for escape from overload. This parameter is only applied when enable_thread_pool and use_workload_manager are enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:vacuum_cost_delay	Specifies how long the process sleeps when vacuum_cost_limit has been exceeded, in milliseconds. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:b_format_version	Specifies the SQL behavior control configuration item in B-compatible mode. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_control_group	Specifies whether to enable the Cgroups. This parameter must be applied on both CNs and DNs. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:track_stmt_parameter	After track_stmt_parameter is enabled, the executed statements recorded in statement_history are not normalized. The complete SQL statement information can be displayed to help the database administrator locate faults. For a simple query, the complete statement information is displayed. For a PBE statement, the complete statement information and information about each variable value are displayed. The format is "query string; parameters: \$1=value1,\$2=value2, ...". This parameter is used to display full SQL information for users and is not controlled by the track_activity_query_size parameter. This parameter is supported in GaussDB V2.0-8.201 or later.
track_activity_query_size	Specifies byte counts of the current running commands used to trace each active session. If the number of bytes in a command is greater than the value of this parameter, the command will be truncated. This parameter affects the memory usage of multiple system views. Adjusting the value of this parameter has a great impact on the memory usage. You can estimate the memory usage based on the value calculated from instr_unique_sql_count times track_activity_query_size . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:auto_explain_level	Specifies the log level for automatically printing execution plans.
cma:log_saved_days	Specifies the maximum number of days to store logs for.
cma:log_threshold_check_interval	Specifies the interval for compressing and clearing logs, in seconds.
dn:wdr_snapshot_query_timeout	Specifies the execution timeout, in seconds, for the SQL statements associated with database monitoring snapshot operations.
dn:enable_wdr_snapshot_standby	Specifies whether to enable database monitoring snapshots on the standby node.

Parameter	Description
dn:backtrace_min_messages	Prints the function's stack information to the server's log file if the information generated is at least the level specified by this parameter.
dn:log_statement	Specifies whether to record SQL statements. For clients using extended query protocols, logging occurs when an execute message and values (enclosed by a pair of single quotation marks) of bound parameters are received.
dn:asp_flush_rate	Determines how many samples are flushed from memory to disk. When the number of samples in memory reaches the value of asp_sample_num , a portion of them will be flushed to disk based on the ratio specified by this parameter. If this parameter is set to 10 , the data flushing ratio is 10:1 (10%).
dn:asp_sample_interval	Specifies the sampling interval, in seconds.
dn:audit_space_limit	Specifies the maximum storage space occupied by audit files, in KB.
dn:enable_defer_calculate_snapshot	Determines if snapshot calculations for xmin and oldestxmin are deferred. If this parameter is set to on , calculations are triggered every 1,000 transactions or once per second. This reduces overhead under heavy load but slows down the advancement of oldestxmin , potentially delaying garbage collection. If this parameter is set to off , xmin and oldestxmin update in real time, improving tuple cleanup but with higher processing overhead.
dn:enable_global_plancache	Specifies whether to share the execution plan cache for statements in PBE queries and stored procedures. Enabling it can help reduce the memory needed on database nodes during high-concurrency workloads. With enable_global_plancache enabled, local_syscache_threshold is automatically set to a minimum of 16 MB for GPC to take effect. If the value of local_syscache_threshold is configured below 16 MB, it defaults to 16 MB. If set above, the specified value is used.
dn:enable_prevent_job_task_startup	Determines whether to start the job thread.

Parameter	Description
dn:log_min_messages	Specifies which level of messages will be written into the server log. A higher level covers the messages of all the lower levels. The lower the level is, the fewer messages will be written into the run log on the server. A same value for client_min_messages and log_min_messages does not indicate the same level. For some log information, after this parameter is enabled, you also need to set logging_module to enable log printing for the corresponding module.
dn:max_recursive_times	Specifies the maximum number of WITH RECURSIVE iterations.
dn:vacuum_freeze_min_age	Specifies whether VACUUM replaces the xmin column of a record with FrozenXID when scanning a table (in the same transaction). Although you can set this parameter to any value, VACUUM will limit the effective value to 50% of autovacuum_freeze_max_age by default.
dn:vacuum_freeze_table_age	Specifies when VACUUM scans the whole table and freezes old tuples. VACUUM performs a full table scan if the difference between the current transaction ID and the value of pg_class.relfrozenxid64 is greater than the specified time. Although you can set this parameter to any value, VACUUM will limit the effective value to 95% of autovacuum_freeze_max_age by default.
dn:allow_create_sysobject	Specifies whether objects such as functions, stored procedures, synonyms, aggregate functions, and operators can be created or modified in the system schema. The system schema refers to the schema provided by the database after initialization, excluding the public schema. The OID of the system schema is usually less than 16384.
dn:standby_max_query_time	Specifies the maximum allowed query time (in seconds) on the standby node when ultimate RTO is enabled. If a query exceeds this limit, it will be canceled. The exact cancellation timing depends on the standby_recycle_interval for the recycling thread and the snapshot timing, so the actual query duration on the standby node may exceed this limit.

Parameter	Description
dn:exrto_standby_read_opt	Specifies whether to enable read optimization on standby nodes with ultimate RTO. This parameter is enabled by default. The setting of this parameter is not synchronized between primary and standby nodes.
dn:dirty_page_percent_max	Specifies the percentage of dirty pages relative to the value of shared_buffers after the incremental checkpoint is enabled. When the value of this parameter is reached, the background page flush thread flushes dirty pages based on the value of max_io_capacity .
dn:audit_enabled	Determines whether to enable or disable the audit process. After the audit process is enabled, it can read the auditing information written by the background process from the pipe and save it into audit files.
dn:enable_bitmapscan	Specifies whether the optimizer uses bitmap scan.
dn:autovacuum	Specifies whether to enable the autovacuum process. Before enabling the process, make sure the track_counts parameter is set to on .
dn:autovacuum_analyze_scale_factor	Specifies a fraction of the table size added to the autovacuum_analyze_threshold parameter when deciding whether to trigger an ANALYZE operation.
dn:autovacuum_analyze_threshold	Specifies the threshold for triggering ANALYZE. When the number of deleted, inserted, or updated rows in a table exceeds the specified threshold, the ANALYZE operation is executed on this table.
dn:autovacuum_freeze_max_age	Specifies the maximum age (in transactions) that a table's pg_class.relfrozenxid column can attain before a VACUUM operation is performed.
dn:autovacuum_io_limits	Determines the maximum number of input/output operations that the autovacuum process can execute per second.
dn:autovacuum_vacuum_threshold	Specifies the threshold for triggering VACUUM. When the number of deleted or updated rows in a table exceeds the specified threshold, the VACUUM operation is executed on this table.
dn:default_transaction_read_only	Specifies whether each new transaction is in read-only state. If this parameter is set to on , DML operations and write transactions cannot be executed.

Parameter	Description
dn:autovacuum_vacuum_cost_delay	Sets the amount of time, in milliseconds, that the autovacuum process will sleep when the cost limit has been exceeded.
dn:autovacuum_vacuum_cost_limit	Sets the accumulated cost that will cause the autovacuum process to sleep.
dn:enable_lock_account	Specifies whether to automatically lock an account when the number of consecutive login failures using incorrect passwords reaches the value of failed_login_attempts . Manual locking by administrators is not controlled by this parameter. Setting this parameter to off reduces account security. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_force_create_obj	Specifies whether to enable one-off import. If this parameter is enabled and there is an undefined object when functions or packages are created or rebuilt, a virtual object will be created for compilation. During function body compilation, try-catch blocks will be used to capture exceptions so that the creation or rebuilding process can run successfully. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:ddl_invalid_mode	Determines how the database handles the objects (functions or packages) that are dependent on a function or package when the function or package is being dropped with a DROP CASCADE operation. During cascade invalidation, if an object is already invalid, the invalidation process will stop. • If this parameter is set to delete, dependent objects will be automatically deleted when a DROP CASCADE operation is performed on a function or package. • If this parameter is set to invalid, dependent objects will be invalidated when a DROP CASCADE operation is performed on a function or package. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:foreign_key_checks	Specifies whether to enable integrity check for foreign key constraints. This parameter is only applied in M-compatible mode. To improve the efficiency of data import, you are advised to set this parameter to off only when you are sure that the data to import meets the integrity requirements of foreign key constraints. If this parameter is set to off , DML operations on referencing and referenced tables will not trigger integrity check on foreign keys. In this case, after a DML operation is performed on a table that may contain data that conflicts with foreign key integrity, there will be no validation of existing data against foreign key constraint integrity, even if this parameter is re-enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:instr_unique_sql_track_type	Specifies how unique SQL statements are recorded. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:idle_in_transaction_timeout	Specifies the maximum idle duration (in seconds) of a transaction after the connection to the server is established and the transaction is started. The value 0 indicates that the timeout setting is disabled. A positive number indicates the maximum idle duration of a transaction after the connection to the server is established and the transaction is started. If the idle duration exceeds the value of this parameter, the client will be disconnected from the server. Set this parameter based on service security requirements. A smaller value indicates higher security but more inconvenient use. If this parameter is set to a small value, the client application may not send new requests in a timely manner after a transaction is started. As a result, the session times out and exits, affecting user experience. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_asp_standby	Specifies whether a standby node is enabled for ASP. If this function is enabled, the collected data of the standby node will also be sampled for generating an ASP report. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:asp_flush_interval	Specifies the interval (in seconds) at which ASP flushes data. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_htap	Specifies whether to enable HTAP. If HTAP is enabled, the inventory row-store data of the column specified by the COLVIEW keyword will be loaded to the in-memory column view (IMCV). It is used to improve the query efficiency of analytical processing (AP). This parameter is disabled by default. You can determine whether to enable it by considering the use case of your database. If there are a large number of AP requests, enabling this parameter improves query efficiency. You are advised not to enable this parameter in TP-only query scenarios. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_parallel_populate	Specifies whether to enable parallel loading for converting inventory row-store data to IMCV to accelerate IMCV table creation. This parameter takes effect only when HTAP is enabled. If this parameter is enabled, loading row-store data to IMCV tables may affect the performance of row-store tables. Set this parameter based on application requirements. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:htap_router_mode	Specifies the HTAP transparent routing mode. This parameter takes effect only when HTAP is enabled. ● row: The query workload is forced to select a row-store plan. ● column: When columns in a query request are loaded into the IMCV, the column-store plan (IMCVScan) is forcibly executed regardless of the cost. If some columns are not loaded to the IMCV, the row-store plan is executed. ● auto: The system automatically selects a row-store, column-store, or a hybrid plan based on whether the columns in the query request are loaded into the IMCV and the cost comparison between row-store and column-store plans. If there are hybrid workloads (scenarios involving both transactional and analytical queries), you are advised to set this parameter to auto. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_imcvscan	Specifies whether to enable the HTAP in-memory column scanning operator IMCVScan. This parameter takes effect only when HTAP is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:htap_max_mem_size	Specifies the upper limit of the available memory (in KB) for the HTAP feature. This parameter takes effect only when HTAP is enabled. The value of this parameter cannot exceed the value of max_process_memory . This parameter should be configured based on the available free memory and the specific use. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:batch_error_mode_savepoint_name	Specifies the name of a savepoint in BatchErrorMode mode. This parameter is only applied to centralized instances. Value range: The value can contain 0 to 1,023 characters. Only letters, digits, and underscores (_) are allowed. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:float_shortest_precision	Specifies whether to use the minimum precision method, instead of sprintf(), to serialize floating-point numbers. This minimum precision method is designed to enhance the performance of floating-point serialization system functions, specifically by reducing the single-threaded execution time of float4out() and float8out(). You are advised to set this parameter for new sites or sites that have performance requirements for floating-point number serialization. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_case_intervaltonumeric	Specifies whether INTERVAL can be converted to NUMERIC in CASE statements. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_logical_replication_dictionary	Specifies whether to enable the function of creating logical replication slots for multi-version dictionary tables. A logical replication slot of the dictionary type can be created only after the system catalog of the dictionary type is baselined. Otherwise, the logical replication slot created will still be of the online catalog type, even if this parameter is enabled. The default value is on . If you need to create a logical replication slot of the online catalog type, set this parameter to off . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:logical_replication_dictionary_retention_time	Specifies the retention period (in days) of data in the system catalogs related to the data dictionary. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_uniq_idx_a_com pat	Specifies whether composite unique indexes are compatible with A databases for NULL values. This parameter is only applied in A-compatible databases. You are advised to retain the default value. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule	Specifies whether to enable the cost-based evaluation policy for a specified query rewriting rule. Some query rewriting rules support cost-based evaluation of whether to rewrite SQL statements. In this way, the kernel can generate better query plans and improve SQL execution efficiency. This parameter is mutually exclusive with the GUC parameter rewrite_rule . If both costbased_rewrite_rule and rewrite_rule are set, the rule configuration related to rewrite_rule does not take effect. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule_ max_iterations	During plan generation for the SQL execution, if the number of conditions met for cost evaluation in the same rule exceeds the value of this parameter, the cost-based evaluation policy will be disabled for the extra conditions in the current request and converted to the rule-based rewriting policy. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:costbased_rewrite_rule_ timeout	When generating a plan for SQL execution, if the overall cost evaluation time using each rule exceeds the timeout interval (in milliseconds) specified by this parameter, the cost-based evaluation policy will be disabled for subsequent processes of the current request and the rule-based rewriting policy is used. This parameter is applied when the cost-based evaluation policy is enabled in the query rewriting phase. Additionally, it is only applied when the GUC parameter costbased_rewrite_rule is set to a value other than none . If this parameter is set to 0 , the cost-based rewriting policy will be disabled for the current SQL statement. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_ecpg_cursor_duplicate_operation	Specifies whether a cursor can be opened or closed repeatedly in A-compatible mode. This parameter takes effect only when ECPG is connected to A-compatible databases. If the database of this version is directly installed, set this parameter to on . If an earlier version is upgraded to the current version, set this parameter to off . This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bm25_ncandidates	Specifies the number of top documents to be prepared when the BM25 index is used to sort and retrieve similar documents. This parameter is used to accelerate the retrieval of similar documents with the WHERE filter criterion. Set this parameter based on the value of LIMIT and the probability of documents satisfying the filter condition. The value of this parameter = Value of LIMIT/ Probability of documents satisfying the filter condition. If the value is too large or too small, the BM25 index retrieval latency is prolonged. In severe cases, the latency may increase by three to four times. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bm25_ranking_metric	Specifies the algorithm used by the BM25 index to rank and retrieve similar documents during a search. The mappings between the values and algorithms are as follows: • 0: BM25_OKAPI (default value) • 1: BM25_ATIRE • 2: BM25_L • 3: BM25_PLUS • 4: BM25_ADPT • 5: BM25_T • 6: BM25_C • 7: BM25_Q • 8: TF_IDF • 9: TF_LDP_IDF • > 9: BM25_OKAPI This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:enable_interp_reuse_transaction	Specifies whether subtransactions of nested exception blocks can be reused in stored procedures. This parameter takes effect only when the plstmt_implicit_savepoint option in the behavior_compat_options parameter is enabled. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_smp_dml	Specifies whether DML statements can be executed in parallel. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_cached_operator	Specifies whether to enable operator execution state caching. If this parameter is disabled, the operator execution status cannot be cached, affecting query performance. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_smp_partitionwise	Specifies whether to enable partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:force_smp_partitionwise_scan	Specifies whether to enable partition-wise parallel scan globally after enable_smp_partitionwise is enabled for using partition-wise joins in the SMP scenario. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_fast_numeric_agg	Specifies whether to enable aggregation optimization for the numeric data type. If this parameter is disabled, the query performance of the corresponding scenarios may deteriorate. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_err_stack	Specifies whether to print stack information for errors whose ACTION is "Contact Huawei technical support." This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_show_err_code	Specifies whether to display error codes for front-end connections such as gsql. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:sqlapply_preserve_commit_order	Specifies whether the logical standby node replays and commits transactions based on the transaction commit sequence of the remote primary node. This parameter is supported in GaussDB V2.0-8.201 or later.

Parameter	Description
dn:blackbox_number	Specifies the maximum number of black box files that can be retained in the \$GAUSSLOG/gs_blackbox/{nodename} directory. If this parameter is set to 0 , the function is disabled and black box files are not retained, which may affect performance diagnosis. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_bloom_filter	Specifies whether Bloom filter optimization can be used. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:bloom_filter_apply_threshold	Specifies the minimal data threshold on the hash join's apply side for creating a Bloom filter when Bloom filter optimization is enabled. If the data volume on the apply side is less than the value of this parameter, no Bloom filter is created. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:scan_wait_for_bloom_filter	Specifies whether the scan operator waits for the completion of Bloom filter creation. This parameter is supported in GaussDB V2.0-8.201 or later.
dn:enable_fast_vecop	Specifies whether to enable vectorized calculation optimization for column-store tables. If it is enabled, column-store calculation of the Timestamp, int32, int64, and float8 data types is optimized, the row-store calling cost is reduced, and automatic vectorization calculation is supported. This parameter is supported in GaussDB V2.0-8.201 or later.

Configurable Parameters for Version V2.0-3.x

The following table describes the parameters that can be modified.

Configurable Parameters for Distributed Instances of Version V2.0-3.x

Table 14-4 Parameters for distributed instances

Parameter	Description
audit_system_object	Determines whether to audit the CREATE, DROP, and ALTER operations on GaussDB Kernel database objects. GaussDB Kernel database objects include databases, users, schemas, and tables. You can change the parameter value to audit only the operations on required database objects. During a forcible primary/standby failover, set audit_system_object to the maximum value and audit all DDL objects. If the parameter value is incorrectly changed, DDL audit logs will be lost. Contact technical support to change it.
autoanalyze	Specifies whether to automatically collect statistics on tables that have no statistics when a plan is generated.
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. 0 indicates there is no timeout. Unit: second
cn:effective_cache_size	Specifies the size of the disk buffer available to the CN optimizer in a single query. Unit: 8 KB
cn:enable_hotkeys_collection	Specifies whether to collect statistics on accessed key values in databases.
cn:track_stmt_session_slot	Specifies the maximum number of full or slow SQL statements that can be cached in a CN session.
datestyle	Specifies the display format for date and time.
dn:effective_cache_size	Specifies the size of the disk buffer available to the DN optimizer in a single query. Unit: 8 KB
dn:enable_hotkeys_collection	Specifies whether to collect statistics on accessed key values in databases.
dn:track_stmt_session_slot	Specifies the maximum number of full or slow SQL statements that can be cached in a DN session.
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.

Parameter	Description
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots.
enable_stream_operator	Specifies the query optimizer's use of streams. When this parameter is set to off , a large number of logs indicating that the stream plans cannot be pushed down are recorded.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
max_replication_slots	Specifies the number of log replication slots on the primary node.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DNs connect to primary DNs to obtain physical logs, and logical replication tools connect to primary DNs to obtain logical logs.
password_effect_time	Specifies the validity period of the password, in days.
password_lock_time	Specifies the duration for a locked account to be automatically unlocked, in days.
recovery_time_target	Specifies the time for the standby node to write and replay logs, in seconds.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second
timezone	Specifies the time zone for displaying and interpreting time stamps.
track_stmt_stat_level	Controls the level of statement execution tracking.

Parameter	Description
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
cn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48.
dn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48.
cn:grw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:grw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
cn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
dn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
cn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
dn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.

Parameter	Description
cn:audit_dml_state_select	Determines whether to audit the SELECT operation.
dn:audit_dml_state_select	Determines whether to audit the SELECT operation.
cn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
dn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
cn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
dn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
cn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
dn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
cn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
dn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
wdr_snapshot_interval	Specifies the interval (in minutes) at which the background thread Snapshot automatically performs snapshot operations on the database monitoring data.
cn:enable_auto_explain	Specifies whether to automatically print execution plans. This parameter is used to locate slow stored procedures or slow queries.
dn:enable_auto_explain	Specifies whether to automatically print execution plans. This parameter is used to locate slow stored procedures or slow queries.
enable_wdr_snapshot	Specifies whether to enable WDR snapshots. If this parameter is enabled, the kernel periodically takes performance snapshots.

Parameter	Description
cn:max_concurrent_autonomous_transactions	Specifies the maximum number of autonomous transaction connections, that is, the maximum number of concurrent autonomous transactions executed at the same time. If this parameter is set to 0 , autonomous transactions cannot be executed.
dn:max_concurrent_autonomous_transactions	Specifies the maximum number of autonomous transaction connections, that is, the maximum number of concurrent autonomous transactions executed at the same time. If this parameter is set to 0 , autonomous transactions cannot be executed.
cn:max_standby_archive_delay	Specifies the wait period (in milliseconds) before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode.
dn:max_standby_archive_delay	Specifies the wait period (in milliseconds) before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode.
cn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
dn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
cn:recovery_max_workers	Specifies the number of concurrent replayer threads.
dn:recovery_max_workers	Specifies the number of concurrent replayer threads.
cn:auto_explain_log_min_duration	Specifies the minimum duration of execution plans that are automatically printed. Only execution plans whose duration is greater than the value of auto_explain_log_min_duration will be printed. Unit: ms
dn:auto_explain_log_min_duration	Specifies the minimum duration of execution plans that are automatically printed. Only execution plans whose duration is greater than the value of auto_explain_log_min_duration will be printed. Unit: ms
cn:audit_function_exec	Specifies whether to record the audit information during the execution of the stored procedures, anonymous blocks, or user-defined functions (excluding system functions).

Parameter	Description
dn:audit_function_exec	Specifies whether to record the audit information during the execution of the stored procedures, anonymous blocks, or user-defined functions (excluding system functions).
cn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
dn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
cms:datastorage_threshold_value_check	Specifies the disk usage threshold to put a database node into read-only mode. If the disk usage of a data directory exceeds this threshold, the database node is automatically changed to read-only. Unit: percentage (%)
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.
cn:enable_default_ustore_table	Specifies whether to enable the Ustore storage engine by default. If this parameter is set to on , all created tables are Ustore tables.
dn:enable_default_ustore_table	Specifies whether to enable the Ustore storage engine by default. If this parameter is set to on , all created tables are Ustore tables.
cn:undo_space_limit_size	Specifies the undo forcible reclamation threshold. If 80% of the specified parameter value is reached, forcible reclamation is triggered. The unit is 8 KB. It is recommended that the value be at least the value of undo_limit_size_per_transaction .
dn:undo_space_limit_size	Specifies the undo forcible reclamation threshold. If 80% of the specified parameter value is reached, forcible reclamation is triggered. The unit is 8 KB. It is recommended that the value be at least the value of undo_limit_size_per_transaction .
cn:undo_limit_size_per_transaction	Specifies the maximum undo space for a single transaction. The unit is 8 KB. If the undo space of a transaction exceeds this parameter value, the transaction is rolled back due to an error. It is recommended that this parameter value be smaller than the value of undo_space_limit_size . If this parameter value is larger, the value of undo_space_limit_size will be used as the maximum undo space for a single transaction.

Parameter	Description
dn:undo_limit_size_per_transaction	Specifies the maximum undo space for a single transaction. The unit is 8 KB. If the undo space of a transaction exceeds this parameter value, the transaction is rolled back due to an error. It is recommended that this parameter value be smaller than the value of undo_space_limit_size . If this parameter value is larger, the value of undo_space_limit_size will be used as the maximum undo space for a single transaction.
cn:enable_recyclebin	Enables or disables the recycle bin in real time.
dn:enable_recyclebin	Enables or disables the recycle bin in real time.
cn:recyclebin_retention_time	Specifies how long files will be kept in the recycle bin, in seconds. Files in the recycle bin will be automatically deleted after this length of time.
dn:recyclebin_retention_time	Specifies how long files will be kept in the recycle bin, in seconds. Files in the recycle bin will be automatically deleted after this length of time.
cn:undo_retention_time	Specifies how long undo logs are kept, in seconds. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content.
dn:undo_retention_time	Specifies how long undo logs are kept, in seconds. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content.
cn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.
dn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.

Parameter	Description
cn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.
dn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.
cn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
dn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
cn:resilience_ctrlstmt_control_iopslimit	Specifies the maximum IOPS that can be used by slow SQL statements after normal SQL statements are marked as slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. 0(None) : The IOPS is not limited. 10(LOW) : The limit level for IOPS is LOW . 20(MEDIUM) : The limit level for IOPS is MEDIUM . 50(HIGH) : The limit level for IOPS is HIGH .
dn:resilience_ctrlstmt_control_iopslimit	Specifies the maximum IOPS that can be used by slow SQL statements after normal SQL statements are marked as slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. 0(None) : The IOPS is not limited. 10(LOW) : The limit level for IOPS is LOW . 20(MEDIUM) : The limit level for IOPS is MEDIUM . 50(HIGH) : The limit level for IOPS is HIGH .

Parameter	Description
dn:resilience_ctrlstmt_detect_timelimit	Specifies the execution time of a normal SQL statement that will be marked as a slow SQL statement. The value 0 indicates that slow SQL statements are not identified. A value greater than 0 indicates that a normal SQL statement whose execution time exceeds the value of this parameter is marked as a slow SQL statement. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users. Unit: ms

Configurable Parameters for Centralized Instances of Version V2.0-3.x

Table 14-5 Parameters for centralized instances

Parameter	Description
audit_system_object	Determines whether to audit the CREATE, DROP, and ALTER operations on GaussDB Kernel database objects. GaussDB Kernel database objects include databases, users, schemas, and tables. You can change the parameter value to audit only the operations on required database objects. During a forcible primary/standby failover, set audit_system_object to the maximum value and audit all DDL objects. If the parameter value is incorrectly changed, DDL audit logs will be lost. Contact technical support to change it.
autoanalyze	Specifies whether to automatically collect statistics on tables that have no statistics when a plan is generated.
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. 0 indicates there is no timeout. Unit: second
datestyle	Specifies the display format for date and time.
dn:wal_keep_segments	Specifies the minimum number of transaction log files stored in the pg_xlog directory. Standby nodes obtain the logs from the primary node to perform streaming replication.

Parameter	Description
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
max_replication_slots	Specifies the number of log replication slots on the primary node.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DN's connect to primary DN's to obtain physical logs, and logical replication tools connect to primary DN's to obtain logical logs.
password_effect_time	Specifies the validity period of the password, in days.
password_lock_time	Specifies the duration for a locked account to be automatically unlocked, in days.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second
timezone	Specifies the time zone for displaying and interpreting time stamps.
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond

Parameter	Description
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
dn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48 .
dn:qrw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
dn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
dn:audit_dml_state_select	Determines whether to audit the SELECT operation.
dn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
dn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
dn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
dn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
dn:vacuum_defer_cleanup_age	Specifies the number of transactions used by VACUUM.
dn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.

Parameter	Description
wdr_snapshot_interval	Specifies the interval (in minutes) at which the background thread Snapshot automatically performs snapshot operations on the database monitoring data.
undo_retention_time	Specifies how long undo logs are kept, in seconds. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content.
track_stmt_stat_level	Controls the level of statement execution tracking.
dn:enable_auto_explain	Specifies whether to automatically print execution plans. This parameter is used to locate slow stored procedures or slow queries.
enable_wdr_snapshot	Specifies whether to enable WDR snapshots. If this parameter is enabled, the kernel periodically takes performance snapshots.
dn:max_concurrent_autonomous_transactions	Specifies the maximum number of autonomous transaction connections, that is, the maximum number of concurrent autonomous transactions executed at the same time. If this parameter is set to 0 , autonomous transactions cannot be executed.
dn:max_standby_archive_delay	Specifies the wait period (in milliseconds) before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode.
dn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
dn:recovery_max_workers	Specifies the number of concurrent replayer threads.
dn:track_stmt_parameter	If this parameter is enabled, the executed statements recorded in statement_history are not normalized. The complete SQL statement information can be displayed to help the database administrator locate faults. For a simple query, the complete statement information is displayed. For a PBE statement, the complete statement information and information about each variable value are displayed. The format is "querystring; parameters:\$1=value1,\$2=value2,...". This parameter is used to display full SQL information for users and is not controlled by the track_activity_query_size parameter.

Parameter	Description
dn:auto_explain_log_min_duration	Specifies the minimum duration of execution plans that are automatically printed. Only execution plans whose duration is greater than the value of auto_explain_log_min_duration will be printed. Unit: ms
dn:recovery_time_target	Specifies the time for the standby node to write and replay logs. Unit: second
dn:audit_function_exec	Specifies whether to record the audit information during the execution of the stored procedures, anonymous blocks, or user-defined functions (excluding system functions).
dn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
cms:datastorage_threshold_value_check	Specifies the disk usage threshold to put a database node into read-only mode. If the disk usage of a data directory exceeds this threshold, the database node is automatically changed to read-only. Unit: percentage (%)
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.
dn:undo_space_limit_size	Specifies the undo forcible reclamation threshold. If 80% of the specified parameter value is reached, forcible reclamation is triggered. The unit is 8 KB. It is recommended that the value be at least the value of undo_limit_size_per_transaction .
dn:group_concat_max_len	Specifies the maximum permitted result length in bytes for the GROUP_CONCAT() function.
dn:enable_extension	Specifies whether database extensions can be created. The extension installation is a lab feature and is not recommended.
dn:cost_model_version	Specifies the version of the optimizer cost model. It is a protective parameter. It prevents new optimizer cost models from being applied, so you can keep the current model consistent with the plan of an existing version. If the value of this parameter is changed, many SQL plans may be changed. Exercise caution when modifying this parameter.

Parameter	Description
dn:immediate_analyze_threshold	Specifies the threshold for automatically analyzing inserted data. When the amount of inserted data reaches the original data amount multiplied by the value of immediate_analyze_threshold , and the total number of rows exceeds 100, ANALYZE is automatically triggered.
dn:enable_dynamic_sample_size	Specifies whether to dynamically adjust the number of sampled rows. For a large table with more than one million rows, the number of sampled rows is dynamically adjusted during statistics collection to improve statistics accuracy.
dn:max_io_capacity	Specifies the maximum I/O per second for the background writer process to flush pages in batches. Set this parameter based on the service scenario and the disk I/O capability. If the RTO is short or the data volume is many times that of the shared memory and the service access data volume is random, the value of this parameter cannot be too small. A small value of max_io_capacity reduces the number of pages flushed by the background writer. If a large number of pages are evicted due to service triggering, the services are affected. Unit: KB
dn:max_connections	Specifies the maximum number of concurrent connections to DNs.
log_autovacuum_min_duration	Specifies the interval which should elapse before autovacuum operations are logged. Autovacuum operations equal to or beyond the specified interval will be logged. If it is set to 0 , all autovacuum operations will be logged. If it is set to -1 , no autovacuum operations will be logged.
dn:sql_beta_feature	Specifies which SQL engine optional beta features are enabled, including optimization of row count estimation and query equivalence estimation. These optional features can improve performance in certain situations, but they may also degrade performance if not thoroughly tested. In certain scenarios, you can set the query rewriting rules through this GUC parameter to improve query efficiency.
dn:ai_watchdog_max_cpu_usage	Specifies the expected upper limit of the database CPU usage. The value is normalized based on the multi-core situation.
dn:enable_control_group	Specifies whether to enable the Cgroups.
dn:explain_perf_mode	Specifies the display format of EXPLAIN.

Parameter	Description
dn:enable_ai_watchdog	Enables or disables the AI watchdog function.
dn:resilience_ctrlstmt_detect_timelimit	Specifies the execution time of a normal SQL statement that will be marked as a slow SQL statement. The value 0 indicates that slow SQL statements are not identified. A value greater than 0 indicates that a normal SQL statement whose execution time exceeds the value of this parameter is marked as a slow SQL statement. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
dn:resilience_ctrlslot_available_maxpercent	Specifies the maximum percentage of threads in the thread pool that can be occupied by slow SQL statements. This parameter is only valid for SELECT statements that are executed by non-sysadmin/monitoradmin users.
dn:require_ssl	Controls whether the server requires SSL connections. This parameter is only valid when ssl is set to on .
dn:enable_gpi_auto_update	Determines whether global indexes are updated by default in partition DDL commands.
dn:maintenance_work_memory	Specifies the maximum memory (in bytes) to be used for maintenance.
dn:work_mem	Specifies the amount of memory required by internal sort operations and hash tables before they write data to temporary disk files.
dn:enable_cachedplan_mgr	Specifies whether to enable the adaptive plan selection function.
dn:recovery_parse_workers	Specifies the number of ParseRedoRecord threads in the ultimate RTO feature.
dn:recovery_redo_workers	Specifies the number of PageRedoWorker threads corresponding to each ParseRedoRecord thread when ultimate RTO is enabled.
dn:max_standby_base_page_size	Sets the maximum storage space of base page files on the standby node after ultimate RTO is enabled.
dn:pagewriter_sleep	Specifies the interval for the page writer thread to flush dirty pages to disks after the incremental checkpoint is enabled.

Parameter	Description
dn:enable_double_write	Specifies whether to enable double-write. If both enable_incremental_checkpoint and enable_double_write are enabled, double-write will be used for protection and full_page_writes is not used to prevent half-page write.
dn:bgwriter_delay	Specifies the interval at which the background writer writes dirty shared buffers. Each time, the background writer initiates write operations for some dirty buffers. In full checkpoint mode, the bgwriter_lru_maxpages parameter is used to control the amount of data to be written each time, and the process is restarted after the period of hibernation specified by bgwriter_delay (in milliseconds). In incremental checkpoint mode, the number of target idle buffer pages is calculated based on the value of candidate_buf_percent_target . If the number of idle buffer pages is insufficient, a batch of pages are flushed to disks at the interval specified by bgwriter_delay (in milliseconds). The number of flushed pages is calculated based on the target difference percentage. The maximum number of flushed pages is limited by max_io_capacity .
dn:wal_receiver_buffer_size	Sets the size of the memory buffer that stores the Xlogs received by the standby and secondary nodes. Currently, primary/standby/secondary deployment is not supported by default.
dn:enable_incremental_checkpoint	Specifies whether to enable incremental checkpoint.

Configurable Parameters for Version V2.0-2.x

The following table describes the parameters that can be modified.

Configurable Parameters for Distributed Instances of Version V2.0-2.x

Table 14-6 Parameters for distributed instances

Parameter	Description
audit_system_object	Determines whether to audit the CREATE, DROP, and ALTER operations on GaussDB Kernel database objects. GaussDB Kernel database objects include databases, users, schemas, and tables. You can change the parameter value to audit only the operations on required database objects. During a forcible primary/standby failover, set audit_system_object to the maximum value and audit all DDL objects. If the parameter value is incorrectly changed, DDL audit logs will be lost. Contact technical support to change it.
autoanalyze	Specifies whether to automatically collect statistics on tables that have no statistics when a plan is generated.
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. 0 indicates there is no timeout. Unit: second
cn:effective_cache_size	Specifies the size of the disk buffer available to the CN optimizer in a single query. Unit: 8 KB
cn:enable_hotkeys_collection	Specifies whether to collect statistics on accessed key values in databases.
cn:track_stmt_session_slot	Specifies the maximum number of full or slow SQL statements that can be cached in a CN session.
datestyle	Specifies the display format for date and time.
dn:effective_cache_size	Specifies the size of the disk buffer available to the DN optimizer in a single query. Unit: 8 KB
dn:enable_hotkeys_collection	Specifies whether to collect statistics on accessed key values in databases.
dn:track_stmt_session_slot	Specifies the maximum number of full or slow SQL statements that can be cached in a DN session.
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.

Parameter	Description
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots.
enable_stream_operator	Specifies the query optimizer's use of streams. When this parameter is set to off , a large number of logs indicating that the stream plans cannot be pushed down are recorded.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
max_replication_slots	Specifies the number of log replication slots on the primary node.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DNs connect to primary DNs to obtain physical logs, and logical replication tools connect to primary DNs to obtain logical logs.
password_effect_time	Specifies the validity period of the password, in days.
password_lock_time	Specifies the duration for a locked account to be automatically unlocked, in days.
recovery_time_target	Specifies the time for the standby node to write and replay logs, in seconds.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second
timezone	Specifies the time zone for displaying and interpreting time stamps.
track_stmt_stat_level	Controls the level of statement execution tracking.

Parameter	Description
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
cn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48 .
dn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48 .
cn:grw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:grw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
cn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
dn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
cn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
dn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.

Parameter	Description
cn:audit_dml_state_select	Determines whether to audit the SELECT operation.
dn:audit_dml_state_select	Determines whether to audit the SELECT operation.
cn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
dn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
cn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
dn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.
cn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
dn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
cn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
dn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
wdr_snapshot_interval	Specifies the interval (in minutes) at which the background thread Snapshot automatically performs snapshot operations on the database monitoring data.
enable_wdr_snapshot	Specifies whether to enable WDR snapshots. If this parameter is enabled, the kernel periodically takes performance snapshots.
cn:max_standby_archive_delay	Specifies the wait period (in milliseconds) before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode.

Parameter	Description
dn:max_standby_archive_delay	Specifies the wait period (in milliseconds) before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode.
cn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
dn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
cn:recovery_max_workers	Specifies the number of concurrent replayer threads.
dn:recovery_max_workers	Specifies the number of concurrent replayer threads.
cn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
dn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB
cms:datastorage_threshold_value_check	Specifies the disk usage threshold to put a database node into read-only mode. If the disk usage of a data directory exceeds this threshold, the database node is automatically changed to read-only. Unit: percentage (%)
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.

Configurable Parameters for Centralized Instances of Version V2.0-2.x

Table 14-7 Parameters for centralized instances

Parameter	Description
audit_system_object	Determines whether to audit the CREATE, DROP, and ALTER operations on GaussDB Kernel database objects. GaussDB Kernel database objects include databases, users, schemas, and tables. You can change the parameter value to audit only the operations on required database objects. During a forcible primary/standby failover, set audit_system_object to the maximum value and audit all DDL objects. If the parameter value is incorrectly changed, DDL audit logs will be lost. Contact technical support to change it.

Parameter	Description
autoanalyze	Specifies whether to automatically collect statistics on tables that have no statistics when a plan is generated.
autoanalyze_timeout	Specifies the autoanalyze timeout period. If the duration of autoanalyze on a table exceeds the value of autoanalyze_timeout , the autoanalyze operation is automatically canceled. 0 indicates there is no timeout. Unit: second
datestyle	Specifies the display format for date and time.
dn:wal_keep_segments	Specifies the minimum number of transaction log files stored in the pg_xlog directory. Standby nodes obtain the logs from the primary node to perform streaming replication.
enable_seqscan	Specifies whether to enable the optimizer's use of sequential scan plan types. It is impossible to completely suppress sequential scans, but setting this parameter to off allows the optimizer to choose other methods if available.
enable_slot_log	Specifies whether to enable primary/standby synchronization for logical replication slots.
failed_login_attempts	Specifies the maximum number of incorrect password attempts before an account is locked. The account will be automatically unlocked after the time specified in password_lock_time elapses. Only the sysadmin user can set this parameter.
log_min_duration_statement	Specifies the threshold for logging the duration of a completed statement. If a statement runs for a period greater than or equal to the specified value, its duration will be logged. The value -1 disables logging statement durations. If this parameter is set to a small value, the load throughput may be affected. Unit: millisecond
max_replication_slots	Specifies the number of log replication slots on the primary node.
max_wal_senders	Specifies the maximum number of WAL sender threads that can be created. Processes created for the following purposes occupy WAL sender threads: Standby DNs connect to primary DNs to obtain physical logs, and logical replication tools connect to primary DNs to obtain logical logs.
password_effect_time	Specifies the validity period of the password, in days.

Parameter	Description
password_lock_time	Specifies the duration for a locked account to be automatically unlocked, in days.
session_timeout	Specifies how long to wait before a server connection is disconnected due to inactivity. The value 0 indicates there is no time limit. Unit: second
timezone	Specifies the time zone for displaying and interpreting time stamps.
update_lockwait_timeout	Specifies the maximum duration that a lock waits for concurrent updates on a row to complete when the concurrent update feature is enabled. If the lock wait time exceeds this value, the system will report an error. Unit: millisecond
wal_level	Specifies the level of information to be written to the WAL. This is a required value and cannot be commented out. Determines how much information is written to the WAL. When this parameter is set to logical , logical logs are extracted and primary key information is recorded in Xlogs.
dn:audit_thread_num	Specifies the number of audit threads. Value range: 1 to 48 .
dn:qrw_inlist2join_optmode	Specifies whether to enable inlist-to-join query rewriting.
dn:audit_xid_info	Determines whether to record the transaction IDs of SQL statements in the detail_info column of audit logs. 0 : The transaction IDs are not recorded. 1 : The transaction IDs are recorded.
dn:default_limit_rows	Specifies the estimated number of rows to return by default for generating a generic plan, that is, the default value for the LIMIT clause. If this parameter is set to a negative number, the value is converted to a percentage, for example, -5 is equivalent to 5%, indicating that 5% of the total rows will be returned.
dn:audit_dml_state_select	Determines whether to audit the SELECT operation.
dn:audit_dml_state	Determines whether to audit the INSERT, UPDATE, and DELETE operations on a specific table. 0 : These operations are not audited. 1 : These operations are audited.
dn:random_page_cost	Specifies the estimated cost for the optimizer to fetch an out-of-sequence disk page.

Parameter	Description
dn:audit_set_parameter	Determines whether to audit the SET operation. 0 : The SET operation is not audited. 1 : The SET operation is audited.
dn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
dn:vacuum_defer_cleanup_age	Specifies the number of transactions used by VACUUM.
dn:enable_pbe_optimization	Specifies whether the optimizer optimizes the query plan for statements executed in Parse Bind Execute (PBE) mode.
wdr_snapshot_interval	Specifies the interval (in minutes) at which the background thread Snapshot automatically performs snapshot operations on the database monitoring data.
undo_retention_time	Specifies how long undo logs are kept, in seconds. This parameter is only used for flashback query. Note: 1. The undo space of the local disk increases. 2. In subsequent incremental backups, the size of the backup set increases due to the retention of additional undo content.
track_stmt_stat_level	Controls the level of statement execution tracking.
enable_wdr_snapshot	Specifies whether to enable WDR snapshots. If this parameter is enabled, the kernel periodically takes performance snapshots.
dn:max_standby_archive_delay	Specifies the wait period (in milliseconds) before queries on a standby node are canceled when the queries conflict with WAL processing and archiving in hot standby mode.
dn:max_standby_streaming_delay	Specifies how long a standby node waits before canceling queries, in milliseconds.
dn:recovery_max_workers	Specifies the number of concurrent replayer threads.
dn:recovery_time_target	Specifies the time for the standby node to write and replay logs. Unit: second
dn:local_syscache_threshold	Specifies the size of system catalog cache in a session. Unit: KB

Parameter	Description
cms:datastorage_threshold_value_check	Specifies the disk usage threshold to put a database node into read-only mode. If the disk usage of a data directory exceeds this threshold, the database node is automatically changed to read-only. Unit: percentage (%)
wdr_snapshot_retention_days	Specifies how many days database monitoring snapshots are saved for.
log_autovacuum_min_duration	Specifies the interval which should elapse before autovacuum operations are logged. Autovacuum operations equal to or beyond the specified interval will be logged. If it is set to 0 , all autovacuum operations will be logged. If it is set to -1 , no autovacuum operations will be logged.
dn:max_connections	Specifies the maximum number of concurrent connections to DNs.

14.2 Modifying GaussDB Instance Parameters

You can modify parameters of a GaussDB instance to bring out the best possible performance of the instance. You can also check the parameter values of an instance.

GaussDB provides the following types of parameters:

- Public parameters: GaussDB uses a set of default running parameters after it is installed. You can modify the parameters to better fit your application scenarios and data volume.
- Parameters for data redistribution: These parameters are used to control the data redistribution policy during database scale-out.

Precautions

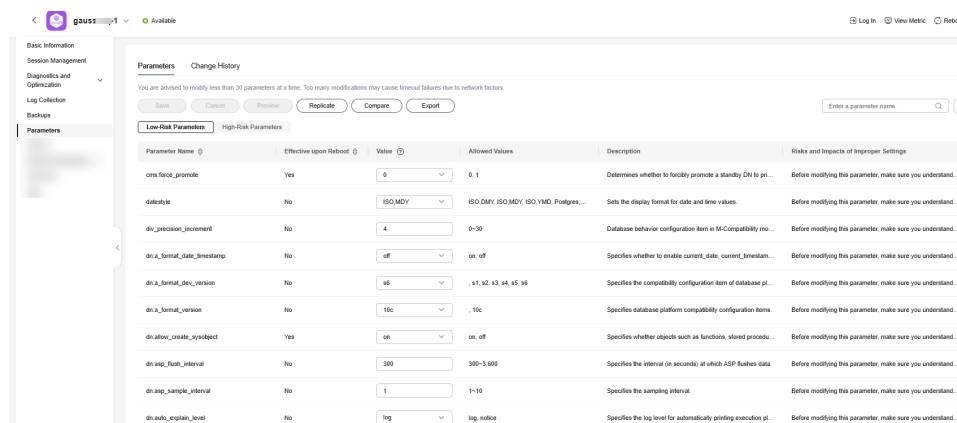
- Parameters for data redistribution can be modified only for distributed instances of version V2.0-2.6 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- Parameters of read replicas can be modified only for centralized (1 primary + 2 standby) instances of version V2.0-2.7.1 or later.
- Before modifying parameters, make sure you understand their meanings and fully verify the changes in a test environment to avoid instance or service exceptions caused by inappropriate parameter settings.

Modifying Common Parameters of the Current Instance

Step 1 [Log in to the management console](#).

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

- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane on the left, choose **Parameters**.

Figure 14-1 Parameters

- You can modify and query the parameters applied to the instance on this page. After modifying parameters, you can preview the changes or cancel the modification. After confirming that all changes are correct, click **Save**.
The modification of some parameters is only applied after the instance is rebooted. After you modify a parameter value, view the value in the **Effective upon Reboot** column.
 - If the value is **Yes** and the instance status on the **Instances** page is **Parameter change. Pending reboot**, you must reboot the instance for the modifications to take effect.
 - If the value is **No**, the modifications take effect immediately for the instance.
- You can click **Replicate** to save the parameters of the instance as a parameter template. You can view the parameter template under the **Custom Templates** tab of the **Parameter Templates** page. For details, see [Managing Parameter Templates for GaussDB Instances](#).
- You can click **Export** to download the parameters of the instance to your local PC.
- You can click **Compare** to compare the parameter template applied to the current instance with an existing parameter template.

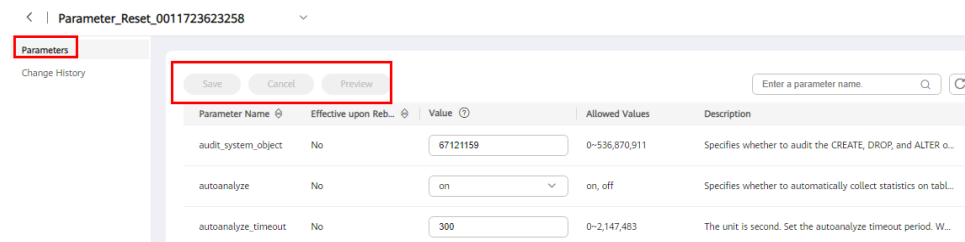
----End

Modifying Common Parameters of Multiple Instances at a Time

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.

- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** In the navigation pane on the left, choose **Parameter Templates**. Click the **Custom Templates** tab, and click the name of the target parameter template.
- Step 5** Modify parameters as needed.

Figure 14-2 Modifying parameters



- Click **Save**. In the displayed dialog box, click **Yes** to save the modification.
 - To cancel your changes, click **Cancel**.
 - To preview your changes, click **Preview**.
- Step 6** After the parameters are modified, click **Change History** to view what changes have been made.
- The changes take effect only after you apply the parameter template to instances. For details, see [Applying a Parameter Template](#).

----End

Modifying Data Redistribution Parameters of the Current Instance

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Change Parameters for Scale-out** or **Change Parameters for Redistribution**.

Figure 14-3 Parameters

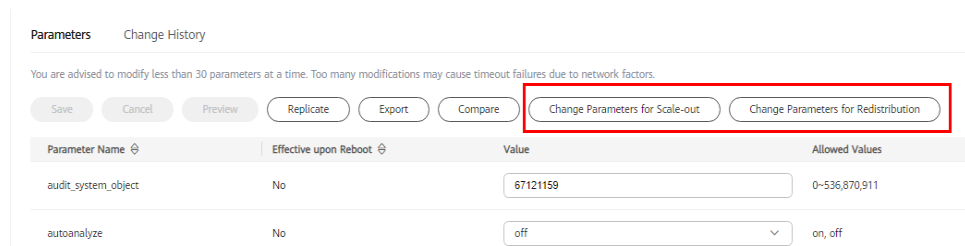


Figure 14-4 Changing parameters for scale-out

Change Parameters for Scale-out ✕

Parameter Name	Value	Allowed Values	Description
lockwait_timeout		1~1,200,000	Specifies the lock timeout interval, in ms. If a thread doe...
lockwait_interval		0~300	Specifies the maximum time in seconds that a thread w...
trylock_threshold		1~2,147,483,647;-1	Specifies the maximum number of attempts to obtain a ...
enable_cancel	false	false>true	Enables or disables fast service failure. Enable this funct...
last_catchup_threshold		1,000~60,000	Specifies the time required for DELETE and INSERT oper...
catchup_times		1~2,147,483,647;-1	Specifies the maximum number of catchups. Extra catch...
write_error_mode	false	false>true	Specifies whether to use write error mode. 'true' write er...
catchup_query_dop		1~32	Specifies how many operations can be simultaneously ex...
parallel_catchup_threshold		1,000~1,800,000	Specifies the time threshold for enabling parallel catchu...
parallel_reindex_jobs		1~64	Specifies the number of indexes that are created in paral...

★ Enter confirm

★ Confirm

☐ I understand the consequences.

OK

Cancel

Figure 14-5 Changing parameters for redistribution

Change Parameters for Redistribution ✕

Parameter Name	Value	Allowed Values	Description
redis_parallel_jobs		1~8	Specifies the number of concurrent tasks during data re...
redis_resource_level	l m h f	l m h f	Specifies the resource level during data redistribution. V...
redis_join_tables	 0/1,000	ex: ["database1","schema1","ta...	Whether to enable concurrent scale-out of multiple tabl...

★ Enter confirm

★ Confirm

☐ I understand the consequences.

OK

Cancel

Step 6 Enter required parameter values, enter **confirm** in the text box, select the confirmation check box, and click **OK**.

----End

FAQ

- [How Can I Change the Time Zone of a GaussDB Instance?](#)
- [How Can I Configure a Password Expiration Policy in GaussDB?](#)
- [Can I Use SQL Commands in GaussDB to Modify Global Parameters?](#)
- [Can I Enable SELECT Operation Auditing in GaussDB?](#)

- [How Do I Set the Disk Usage Threshold for Putting a Database Node into Read-Only Mode in GaussDB?](#)
- [How Do I Set the Date and Time Display Format in GaussDB?](#)
- [How Do I Set the Time Zone Used for Displaying and Interpreting Timestamps in GaussDB?](#)
- [How Do I Change the Maximum Number of DN Connections Allowed in GaussDB?](#)
- [How Do I Enable the Audit of INSERT, UPDATE, and DELETE Operations in GaussDB?](#)
- [How Do I Set the Maximum Number of WAL Sender Threads That Can Be Created in GaussDB?](#)
- [How Do I Set the Maximum Number of Bytes to Be Logged for Each SQL Statement in GaussDB?](#)
- [How Do I Set the Connection Timeout Interval for GaussDB?](#)
- [How Do I Set the Automatic Unlocking Time After an Account Is Locked in GaussDB?](#)
- [How Do I Set the Maximum Number of Failed Login Attempts for Accounts in GaussDB?](#)

14.3 Viewing Parameter Change History of a GaussDB Instance

Scenarios

After modifying parameters of a GaussDB instance or those in a custom template, you can check their change history.

Precautions

- The change history of a newly replicated or created parameter template is initially blank.
- The change history of the last seven days is displayed.
- The parameter change history of read replicas is only available for centralized (1 primary + 2 standby) instances of version V2.0-2.7.1 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).

Viewing Change History of DB Instance Parameters

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

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

Figure 14-6 Viewing the parameter change history

Parameters

Change History

The parameter change history of the last seven days is displayed.

Enter a parameter name.

🔍

↺

Parameter Name ...	Original Value	New Value	Modification Status	Modification Time ...	Application Status	Application Time
autovacuum_napt...	600	550	✔️ Successful	Aug 16, 2024 10:09...	✔️ Applied	Aug 16, 2024 10:0...

Step 6 On the displayed page, click **Change History**.

You can view the parameter name, original parameter value, new parameter value, modification status, modification time, application status, and application time.

You can apply the parameter history template to instances as required by referring to [Applying a Parameter Template](#).

----End

Viewing Change History of a Parameter Template

Step 1 [Log in to the management console](#).

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

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

Step 4 Choose **Parameter Templates** in the navigation pane on the left. On the **Custom Templates** page, click the parameter template name.

Step 5 On the displayed page, choose **Change History** in the navigation pane on the left.

Figure 14-7 Viewing the change history of a parameter template

Parameters

Change History

The parameter change history of the last seven days is displayed.

Enter a parameter name.

Parameter Name	Original Value	New Value	Modification Status	Modification Time
autovacuum_naptime	600	550	Successful	Aug 16, 2024 10:47:37 GMT...

You can view the parameter name, original parameter value, new parameter value, modification status, and modification time.

----End

14.4 Exporting Parameters of a GaussDB Instance

Scenarios

You can export the parameter template information (parameter names, values, and descriptions) of a DB instance to a CSV file for analysis.

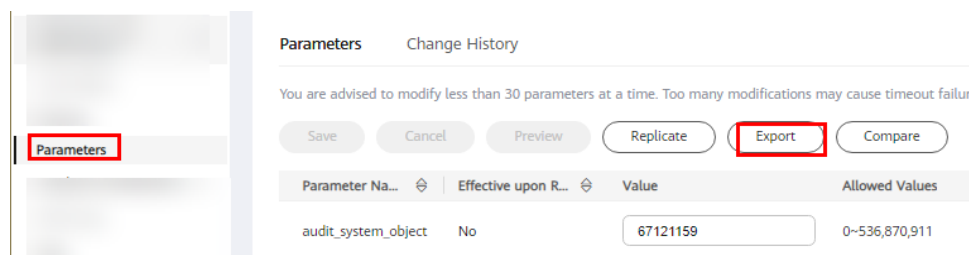
Precautions

The parameters of read replicas can be exported only for centralized (1 primary + 2 standby) instances of version V2.0-2.7.1 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).

Procedure

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane on the left, choose **Parameters**. On the displayed page, click **Export** above the parameter list.

Figure 14-8 Exporting parameters



Exporting to a file: You can export the parameter template information (parameter names, parameter values, and descriptions) of an instance to a CSV file for analysis.

- Step 6** In the displayed dialog box, enter the file name and click **OK**.

The file name must start with a letter and can contain 4 to 81 characters. Only letters, digits, hyphens (-), and underscores (_) are allowed.

----End

14.5 Creating a Custom Parameter Template for GaussDB Instances

Scenarios

Parameter templates store engine configurations that can be applied to one or more DB instances. In GaussDB, there are two types: default and custom parameter templates.

If you do not specify a custom parameter template when creating an instance or restoring data to a new instance, the system defaults to applying the default parameter template.

Note that its parameters are immutable. For tailored needs, you can create a custom parameter template and select it during instance creation. You can also apply a custom parameter template to existing instances. For details, see [Applying a Parameter Template](#).

If you have already created a parameter template and want to provide most of its custom parameters and values in a new parameter template, you can replicate the template you created following the instructions provided in [Replicating a Parameter Template](#).

Precautions

- Not all of the DB engine parameters in a custom parameter template can be changed. The parameters related to compute specifications (that is, the parameters whose default values are **default**) in the parameter template cannot be modified.
- GaussDB parameter template quotas are not shared by Document Database Service (DDS).
- A maximum of 100 GaussDB database parameter templates can be created for each project. All GaussDB engines share the parameter template quota.
- When you change a parameter value on the **Parameters** page of an instance, the change is only applied to the current instance and does not affect other instances.
- Certain parameter changes require a manual restart of the instance to take effect.
- Improper parameter settings may have unintended consequences, including reduced performance and system instability. Exercise caution when changing database parameters; always back up your data beforehand. Do not perform boundary tests within the parameter template to prevent system instability. Before applying parameter template changes to a production DB instance, you should try out these changes on a test DB instance.

Procedure

Step 1 [Log in to the management console](#).

- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** In the navigation pane on the left, choose **Parameter Templates**.
- Step 5** On the **Parameter Templates** page, click **Create Parameter Template**.
- Step 6** In the displayed dialog box, configure required information and click **OK**.
- Select a DB engine for the parameter template.
 - The template name is case-sensitive and can contain 1 to 64 characters. Only uppercase letters, lowercase letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.
 - The template description can contain a maximum of 256 characters and cannot include carriage return characters and the following special characters: `>!<"&'=`
- End

14.6 Managing Parameter Templates for GaussDB Instances

You can perform the following operations to manage GaussDB parameter templates:

- **Applying a parameter template:** Modifications to parameters in a parameter template take effect for instances only after you apply this parameter template to target instances. A parameter template can be applied only to instances of the same version.
- **Replicating a parameter template:** You can replicate a custom parameter template that you have created. When you have already created a parameter template and want to include most of the custom parameters and values from that template in a new parameter template, you can replicate that parameter template. You can also replicate the parameter template used by an existing instance to generate a new one for future use.
- **Comparing instance parameters with a parameter template:** You can compare instance parameters with a parameter template to see the differences of parameter settings.
- **Comparing parameter templates:** You can compare two default GaussDB parameter templates to see the differences between them. You can also compare two custom parameter templates.
- **Resetting a parameter template:** You can reset all parameters in a parameter template you have created to their default settings.
- **Modifying the description of a parameter template:** You can edit the description of a parameter template you have created.
- **Deleting a parameter template:** You can create up to 100 parameter templates and delete parameter templates that are no longer used.

Applying a Parameter Template

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Parameter Templates** page, perform the following operations based on the type of the parameter template to be applied:

- If you intend to apply a default parameter template to instances, click the **Default Templates** tab and click **Apply** in the **Operation** column of the target parameter template.
- If you intend to apply a custom parameter template to instances, click **Custom Templates** and choose **More > Apply** in the **Operation** column of the target parameter template.

A parameter template can be applied to one or more instances.

Step 5 In the displayed dialog box, select one or more instances to which the parameter template will be applied and click **OK**.

Step 6 After the parameter template is applied, check its application records.

- If you intend to check the application records of a default parameter template, click the **Default Templates** tab and click **View Application Record** in the **Operation** column of the target parameter template.
- If you intend to check the application records of a custom parameter template, click **Custom Templates** and choose **More > View Application Record** in the **Operation** column of the target parameter template.

If the application status of an instance is **Applying**, it will not be displayed in the instance list when you apply a parameter template again. If you want to apply the parameter template to the same instance again, ensure that the application status is **Successful**.

After the parameter template is successfully applied, if you modify parameters in the parameter template and the instance status is **Parameter change. Pending reboot**, you must reboot the instance for the modifications to take effect. If no parameter that requires an instance reboot is modified, the instance status remains unchanged.

-----End

Replicating a Parameter Template

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Parameter Templates** page, click the **Custom Templates** tab. Locate the parameter template to be replicated and click **Replicate** in the **Operation** column.

Alternatively, on the **Instances** page, click the instance name to go to the **Basic Information** page. In the navigation pane, choose **Parameters**. On the displayed page, click **Replicate** to generate a new parameter template for future use.

Step 5 In the displayed dialog box, configure required details and click **OK**.

- The template name is case-sensitive and can contain 1 to 64 characters. Only letters, digits, hyphens (-), underscores (_), and periods (.) are allowed.
- The template description can contain up to 256 characters, but cannot contain carriage returns and the following special characters: > ! < " & ' =

After the parameter template is replicated, a new template is generated in the list on the **Custom Templates** tab of the **Parameter Templates** page. The new parameter template may not be displayed immediately. You are advised to wait for at least 5 minutes before using the new template.

Default parameter templates cannot be replicated, but you can create parameter templates based on them.

----End

Comparing Instance Parameters with a Parameter Template

CAUTION

Parameters of read replicas can be compared only for centralized (1 primary + 2 standby) instances of version V2.0-2.7.1 or later.

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the instance name to go to the **Basic Information** page.

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

Step 6 On the displayed page, click **Compare** to compare the parameters of the current instance.

Step 7 In the displayed dialog box, select a parameter template that you want to compare with parameters of the current instance and click **OK**.

- If their settings are different, the parameter names and values of both parameter templates are displayed.
- If their settings are the same, no data is displayed.

----End

Comparing Parameter Templates

Step 1 [Log in to the management console](#).

- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Parameter Templates** page, click **Default Templates** or **Custom Templates**. Locate a parameter template and click **Compare** in the **Operation** column.
- Step 5** In the displayed dialog box, select a parameter template that uses the same DB engine as the target template and click **OK**.
- If their settings are different, the parameter names and values of both parameter templates are displayed.
 - If their settings are the same, no data is displayed.
- End

Resetting a Parameter Template

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Parameter Templates** page, click the **Custom Templates** tab. Locate the parameter template and choose **More > Reset** in the **Operation** column.
- Step 5** Click **Yes** to reset all parameters to their default values.
- For details about the parameter template statuses, see parameter template-related statuses in [Instance Statuses](#).
 - After you reset a parameter template, you need to view the status of the instance that the parameter template applies to in the instance list. If its status is **Parameter change. Pending reboot**, you must reboot the instance for the reset to take effect.
- End

Modifying the Description of a Parameter Template

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Parameter Templates** page, click the **Custom Templates** tab. Locate the parameter template and click  in the **Description** column.
- Step 5** Enter a new description. You can click  to submit or  to cancel the modification.

- After you submit the modification, you can view the new description in the **Description** column.
- The description can contain up to 256 characters, but cannot contain the following special characters: >!<"&'=

 NOTE

You cannot modify the description of any default parameter template.

----End

Deleting a Parameter Template

 CAUTION

- Deleted parameter templates cannot be recovered. Exercise caution when performing this operation.
 - Default parameter templates cannot be deleted.
-

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Parameter Templates** page, click the **Custom Templates** tab. Locate the parameter template to be deleted and choose **More > Delete** in the **Operation** column.

Step 5 Click **Yes** to delete it.

----End

15 Monitoring and Alarming

15.1 Supported Metrics of GaussDB

Description

This section describes metrics reported by GaussDB as well as their namespaces and dimensions.

Namespace

SYS.GAUSSDBV5

Metric Collection Constraints

- Standby DN of distributed instances: Metric data can be collected only when the instance version is V2.0-3.100.0 or later, and the transaction consistency must be eventual consistency.
- Standby DN of centralized instances: Metric data can be collected only when the instance version is V2.0-2.0.10 or later

For details about how to view the instance version, see [Checking the DB Engine Version](#).

Supported Metrics

The following table lists the performance metrics of GaussDB.

Table 15-1 Monitoring metrics supported by GaussDB

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds001_cpu_util	CPU Usage	CPU usage of the monitored object	Current node	%	N/A	Node	60s
rds002_mem_util	Memory Usage	Memory usage of the monitored object	Current node	%	N/A	Node	60s
rds003_bytes_in	Data Write Volume	Average number of bytes sent by the VM of the monitored object in a measurement period	Current node	Byte/s	1024 (IEC)	Node	60s
rds004_bytes_out	Outgoing Data Volume	Average number of bytes received by the VM of the monitored object in a measurement period	Current node	Byte/s	1024 (IEC)	Node	60s
rds014_iops	Disk IOPS	Real-time value of data disk reads and writes per second of the monitored node	Current node	Count/s	N/A	Node	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds016_disk_write_throughput	Disk Write Throughput	Real-time write throughput per second of the data disk on the monitored node	Current node	Byte/s	1024 (IEC)	Node	60s
rds017_disk_read_throughput	Disk Read Throughput	Real-time read throughput per second of the data disk on the monitored node	Current node	Byte/s	1024 (IEC)	Node	60s
rds020_avg_disk_ms_per_write	Time Required for per Data Disk Write	Average time required for a data disk write on the monitored node in a measurement period	Current node	ms	N/A	Node	60s
rds021_avg_disk_ms_per_read	Time Required for per Data Disk Read	Average time required for a data disk read on the monitored node in a measurement period	Current node	ms	N/A	Node	60s
io_bandwidth_usage	Disk I/O Bandwidth Usage	Percentage of current disk I/O bandwidth	Current node	%	N/A	Node	60s
iops_usage	IOPS Usage	Percentage of used IOPS in the total IOPS	Current node	%	N/A	Node	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds005_instance_disk_used_size	Used Instance Disk Size	Real-time used data disk size of the monitored instance	Instance	GB	N/A	Instance	60s
rds006_instance_disk_total_size	Total Instance Disk Size	Real-time total data disk size of the monitored instance	Instance	GB	N/A	Instance	60s
rds007_instance_disk_usage	Instance Disk Usage	Real-time data disk usage of the monitored instance	Instance	%	N/A	Instance	60s
rds035_buffer_hit_ratio	Buffer Hit Rate	Buffer hit rate of the database	Instance	%	N/A	Instance	60s
rds036_deadlocks	Deadlocks	Incremental number of database transaction deadlocks in a measurement period	Instance	Count	N/A	Instance	60s
rds048_P80	Response Time of 80% SQL Statements	Real-time response time of 80% of database SQL statements	Instance	us	N/A	Instance	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds049_P95	Response Time of 95% SQL Statements	Real-time response time of 95% of database SQL statements	Instance	us	N/A	Instance	60s
rds008_disk_used_size	Used Disk Size	Real-time used data disk size of the monitored node	Current node	GB	N/A	Component	60s
rds009_disk_total_size	Total Disk Size	Real-time total data disk size of the monitored node	Current node	GB	N/A	Component	60s
rds010_disk_usage	Disk Usage	Real-time data disk usage of the monitored node	Current node	%	N/A	Component	60s
rds024_current_sleep_time	Primary Node Flow Control Duration	Real-time primary node flow control duration on the monitored node	Distributed: standby DNs Centralized: standby DNs	us	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds025_current_rto	Standby Node RTO	Real-time Recovery Time Objective (RTO) of the primary/standby replication of the monitored node	Distributed: standby DNs Centralized: standby DNs	s	N/A	Component	60s
rds026_login_counter	User Logins per Second	Average number of logins per second in a measurement period	Distributed: all CNs Centralized: primary DNs	Count/s	N/A	Component	60s
rds027_logout_counter	User Logouts per Second	Average number of logouts per second in a measurement period	Distributed: all CNs Centralized: primary DNs	Count/s	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds028_standby_delay	Standby Node Redo Progress	Real-time redo progress of the standby node in a shard. It indicates the difference of the redo progress between the primary and standby nodes.	Distributed: standby DN Centralized: primary DN	Byte	1024 (IEC)	Component	60s
rds030_wait_ratio	Lock-Waiting Session Rate	Real-time rate of lock waiting sessions to active sessions	Distributed: all CNs + primary DN Centralized: all DN	%	N/A	Component	60s
rds031_active_ratio	Active Session Rate	Real-time rate of active sessions to all sessions	Distributed: all CNs + primary DN Centralized: all DN	%	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds034_inuse_counter	CN Connections	Real-time number of in-use connections in the CN connection pool	Distributed: all CNs Centralized: N/A	Count	N/A	Component	60s
rds037_commit_counter	User Committed Transactions per Second	Average number of transactions committed by users per second in a measurement period	Distributed: all CNs Centralized: primary DN	Count /s	N/A	Component	60s
rds038_rollback_counter	User Rollback Transactions per Second	Average number of transactions rolled back by users per second in a measurement period	Distributed: all CNs Centralized: primary DN	Count /s	N/A	Component	60s
rds039_bg_commit_counter	Background Committed Transactions per Second	Average number of transactions committed by the background per second in a measurement period	Distributed: all CNs Centralized: primary DN	Count /s	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds040_bg_rollback_counter	Background Rollback Transactions per Second	Average number of transactions rolled back by the background per second in a measurement period	Distributed: all CNs Centralized: primary DN	Count /s	N/A	Component	60s
rds041_resp_avg	Average Response Time of User Transactions	Average response time of user transactions	Distributed: all CNs Centralized: primary DN	us	N/A	Component	60s
rds042_rollback_ratio	User Transaction Rollback Rate	Average rate of user rollback transactions to all user committed and rolled back transactions in a measurement period	Distributed: all CNs Centralized: primary DN	%	N/A	Component	60s
rds043_bg_rollback_ratio	Background Transaction Rollback Rate	Average rate of background rollback transactions to all user committed and rolled back transactions in a measurement period	Distributed: all CNs Centralized: primary DN	%	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds044_ddl_count	Data Definition Language/s	Average number of DDL statements in user load at the query layer in a measurement period	Distributed: all CNs + all DN Centralized: all DN	Count/s	N/A	Component	60s
rds045_dml_count	Data Manipulation Language/s	Average number of DML statements in user load at the query layer in a measurement period	Distributed: all CNs + all DN Centralized: all DN	Count/s	N/A	Component	60s
rds046_dcl_count	Data Control Language/s	Average number of DCL statements in user load at the query layer in a measurement period	Distributed: all CNs + all DN Centralized: all DN	Count/s	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds047_ddl_dcl_ratio	DDL and DCL Rate	Average rate of DDL and DCL statements to DDL, DCL, and DML statements in user load at the query layer in a measurement period	Distributed: all CNs + all DN Centralized: all DN	%	N/A	Component	60s
rds050_ckpt_delay	Data Volume to Be Flushed to Disks	Real-time amount of data to be flushed to disks during synchronization	Distributed: all CNs + primary DN Centralized: primary DN	Byte	1024 (IEC)	Component	60s
rds051_phyrds	Physical Reads per Second	Average number of physical reads per second in a measurement period	Distributed: all CNs + primary DN Centralized: all DN	Count/s	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds052_phy wrts	Physical Writes per Second	Average number of physical writes per second in a measurement period	Distributed: all CNs + primary DN Centralized: all DNs	Count /s	N/A	Component	60s
rds053_online_session	Online Sessions	Real-time number of online sessions	Distributed: all CNs + all DNs Centralized: all DNs	Count	N/A	Component	60s
rds054_active_session	Active Sessions	Real-time number of active sessions	Distributed: all CNs + primary DN Centralized: primary DN	Count	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds055_online_ratio	Online Session Rate	Real-time percentage of online sessions on a CN (of a distributed instance) or a primary DN (of a centralized instance)	Distributed: all CNs + primary DNs Centralized: all DNs	%	N/A	Component	60s
rds060_long_running_transaction_exectime	Maximum Execution Duration of Database Transactions	Maximum execution duration of database transactions	Distributed: all CNs + primary DNs Centralized: all DNs	s	N/A	Component	60s
rds066_replication_slot_wal_log_size	WAL Log Size in the Replication Slot	Real-time size of WAL logs reserved in the replication slot of a primary DN	Distributed: primary DNs Centralized: all DNs	Byte	1024 (IEC)	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds067_xlog_lsn	Xlog Rate	Real-time rate of Xlogs on CNs or primary DNs	Distributed: all CNs + primary DNs Centralized: primary DNs	Byte/s	1024 (IEC)	Component	60s
rds068_swap_used_ratio	Swap Memory Usage	Real-time swap memory usage of the OS	Current node	%	N/A	Node	60s
rds069_swap_total_size	Total Swap Memory	Real-time total swap memory size of the OS	Current node	MB	N/A	Node	60s
rds070_thread_pool	Thread Pool Usage	Real-time thread pool usage on a CN and DN	Distributed: all CNs + primary DNs Centralized: all DNs	%	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds071_locks_session	Sessions Waiting for Locks	Number of sessions waiting for locks on a CN or primary DN. This metric is updated in real time.	Distributed: all CNs + primary DNs Centralized: all DNs	Count	N/A	Component	60s
rds072_streaming_dr_xlog_gap	Shard Log Gap of DR Cluster	Log difference between shards in the DR cluster and shards in the production cluster when streaming DR is enabled	Distributed: all CNs + primary DNs Centralized: primary DNs	Byte	1024 (IEC)	Component	60s
rds073_streaming_dr_xlog_to_be_replayed	Size of Shard Logs to Be Replayed in DR Cluster	Size of the logs to be replayed of each shard in the DR cluster when streaming DR is enabled	Distributed: all CNs + primary DNs Centralized: primary DNs	Byte	1024 (IEC)	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds074_streaming_dr_xlog_flushing_rate	Flushing Rate of Shard Logs in DR Cluster	Rate at which logs of each shard in the DR cluster are flushed to disk when streaming DR is enabled	Distributed: all CNs + primary DN Centralized: primary DN	Byte/s	1024 (IEC)	Component	60s
rds075_streaming_dr_xlog_replay_rate	Replay Rate of Shard Logs in DR Cluster	Rate at which logs of each shard in the DR cluster are replayed when streaming DR is enabled	Distributed: all CNs + primary DN Centralized: primary DN	Byte/s	1024 (IEC)	Component	60s
rds076_streaming_dr_rpo	Shard RPO	Real-time RPO of each shard when streaming DR is enabled	Distributed: all CNs + primary DN Centralized: primary DN	s	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds077_streaming_dr_rto	Shard RTO	Real-time RTO of each shard when streaming DR is enabled	Distributed: all CNs + primary DN Centralized: primary DN	s	N/A	Component	60s
rds078_inactive_replication_slot	Inactive Replication Slots	Number of physical and logical replication slots that are inactive	Distributed: all CNs + primary DN Centralized: all DN	Count	N/A	Component	60s
rds079_standby_not_replayed_log	Size of Read Replica Logs Not Replayed	Difference between the number of replayed read replica logs and the number of received read replica logs	Distributed: standby DN Centralized: standby DN	Byte	1024 (IEC)	Component	60s
cpu_user_usage	User-mode CPU Time Percentage	Percentage of the CPU time devoted to processing application code	Current node	%	N/A	Node	10s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
cpu_sys_usage	Kernel-mode CPU Time Percentage	Percentage of the CPU time devoted to processing kernel code	Current node	%	N/A	Node	10s
cpu_wait_usage	Disk I/O Wait Time Percentage	Percentage of time that the CPU is waiting for disk I/O operations to complete	Current node	%	N/A	Node	10s
rds080_xlog_num	Xlogs	Real-time number of Xlogs in the data directory on a CN or DN	Distributed: all CNs + all DNs Centralized: all DNs	Count	N/A	Component	60s
rds081_xlog_size	Xlog Size	Real-time size of Xlogs in the data directory on a CN or DN	Distributed: all CNs + all DNs Centralized: all DNs	MB	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds064_dynamic_used_memory	Used Dynamic Memory	Real-time used dynamic memory of the monitored object	Distributed: all CNs + all DNs Centralized: all DNs	MB	N/A	Component	60s
rds065_dynamic_used_memory_usage	Dynamic Memory Usage	Real-time dynamic memory usage of the monitored object	Distributed: all CNs + all DNs Centralized: all DNs	%	N/A	Component	60s
rds061_idle_in_transaction_num	Idle Transactions	Real-time reporting of how many idle transactions there are for the monitored object	Distributed: all CNs + all DNs Centralized: all DNs	Count	N/A	Component	60s
rds062_slowquery_sys	Slow SQL Statements in the System Database	Real-time number of slow SQL statements in the system database on the primary DN or CN in a measurement period	Distributed: all CNs Centralized: primary DNs	Count	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds063_slowquery_user	Slow SQL Statements in the User Database	Real-time number of slow SQL statements in the user database on the primary DN or CN in a measurement period	Distributed: all CNs Centralized: primary DNs	Count	N/A	Component	60s
rds082_gaussv5_wait_session	Waiting Sessions	Real-time number of waiting sessions	Distributed: all CNs + standby DNs Centralized: all DNs	Count	N/A	Component	60s
rds083_cn_temp_dir_size	CN Temporary Directory Size	Real-time size of the temporary directories under the data directory on a CN	Distributed: all CNs	MB	N/A	Component	60s
rds084_sys_database_size	System Database Size	Real-time postgres database size on the monitored instance	Current node	Byte	1024 (IEC)	Node	60s
rds085_user_database_size	User Database Total Size	Real-time user database size on the monitored instance	Current node	Byte	1024 (IEC)	Node	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds086_select_distribution	SELECT Distribution	Real-time percentage of SELECT statements	Distributed: all CNs + all DN Centralized: all DN	%	N/A	Component	60s
rds087_update_distribution	UPDATE Distribution	Real-time percentage of UPDATE statements	Distributed: all CNs + all DN Centralized: all DN	%	N/A	Component	60s
rds088_insert_distribution	INSERT Distribution	Real-time percentage of INSERT statements	Distributed: all CNs + all DN Centralized: all DN	%	N/A	Component	60s
rds089_delete_distribution	DELETE Distribution	Real-time percentage of DELETE statements	Distributed: all CNs + all DN Centralized: all DN	%	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds090_gaussdbv5_statement	SQL Statements	Real-time number of unique SQL statements	Distributed: all CNs Centralized: primary DN	Count	N/A	Component	1 hour
rds091_gaussv5_qps	Read Requests	Average number of read requests per second of a tenant in a specified period	Distributed: all CNs Centralized: all DN	Count	N/A	Component	60s
rds092_gaussv5_tps_rt_insert	INSERT Request Response Time	Average response time for INSERT requests of a tenant in a specified period	Distributed: all CNs Centralized: all DN	ms	N/A	Component	60s
rds093_gaussv5_tps_rt_update	UPDATE Request Response Time	Average response time for UPDATE requests of a tenant in a specified period	Distributed: all CNs Centralized: all DN	ms	N/A	Component	60s
rds094_gaussv5_tps_rt_delete	DELETE Request Response Time	Average response time for DELETE requests of a tenant in a specified period	Distributed: all CNs Centralized: all DN	ms	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds095_gaussv5_qps_rt	Read Request Response Time	Average response time for read requests of a tenant in a specified period	Distributed: all CNs Centralized: all DN	ms	N/A	Component	60s
retrans_rate	Retransmission Ratio	Real-time retransmission ratio of TCP packets	Current node	%	N/A	Node	60s
rds096_process_used_memory	Process Used Memory	Real-time used memory by a CN or DN	Distributed: all CNs + all DN Centralized: all DN	MB	N/A	Component	60s
rds097_2pc_transaction_prepare	Oldest Two-Phase Commit Transaction Duration	Maximum duration of uncommitted transactions using two-phase commit	Centralized: primary DN	s	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds098_dn_instance_status	DN Status	Real-time status of a DN. 1 : a normal primary DN; 2 : a normal standby DN; 3 : a normal main standby DN; 4 : a normal cascaded standby DN; 10 : standby DN catching up with primary DN using Xlog files; 20 : a properly connected standby DN with abnormal replication status; 21 : a disconnected DN	Centralized: all DNs	N/A	N/A	Component	60s
rds099_replication_slot_dir_size	Replication Slot Directory Size	Real-time size of the replication slot directory	Centralized: all DNs	KB	N/A	Component	300s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds100_standby_diff_redo_and_receive	Difference Between Redo and Receipt Positions on Standby Node	The difference (in bytes) between the redo position and data receipt position on the standby node. This metric is used to determine whether data inconsistency is caused by slow redo rate on the standby node or because the primary node has not sent redo data.	Distributed: standby DNs Centralized: standby DNs	Byte	1024 (IEC)	Component	60s
rds101_online_distinct_client_addr_count	Online Clients	Number of online clients on each CN	Distributed: all CNs	Count	N/A	Component	60s
rds102_working_distinct_client_addr_count	Active Clients	Number of active client connections on each CN	Distributed: all CNs	Count	N/A	Component	60s

Metric ID	Metric	Description	Display Object	Unit	Conversion Rule	Monitored Object	Monitoring Period (Raw Data)
rds103_shard_min_rto	Shard RTO	Shortest possible RTO among multiple standby DN's within the current shard.	Centralized: primary DN's Distributed: primary DN's	s	N/A	Component	60s
rds104_invalid_usr_pwd_login_denied_count	Login Attempts with Incorrect Usernames or Passwords	The number of failed login attempts due to incorrect usernames or passwords in kernel logs. The value is the difference between two collected values (incremental value).	Centralized: all DN's Distributed: all CN's	Count	N/A	Component	300s
rds114_inconsistent_data_num	Heap and Index Tuple Inconsistencies	The number of inconsistencies between heap tuples and index tuples.	Distributed: all CN's + all DN's	Count	N/A	Component	60s

Dimensions

Table 15-2 Dimensions involved in GaussDB

Key	Value
gaussdbv5_instance_id	GaussDB instance
gaussdbv5_node_id	GaussDB node
gaussdbv5_component_id	GaussDB component

15.2 Querying GaussDB Monitoring Metrics

Scenarios

Cloud Eye monitors operating statuses of DB instances. You can view the DB instance monitoring metrics on the management console. For details, see [Viewing Metrics of DB Instances](#).

Because monitoring data requires time for transmission and display, the DB instance status shown on the Cloud Eye console reflects its state from 5 to 10 minutes earlier. If your DB instance is newly created, wait for 5 to 10 minutes and then view the monitoring data.

You can also view database metrics on the [Monitoring Dashboards](#) page of the GaussDB console. On the **Monitoring Dashboards** page, you can select all nodes or components of an instance at the same time to check their metrics. In this way, you can easily compare metric differences between components or nodes. In contrast, CES allows you to check metrics of only one node or component at a time.

Prerequisites

- A DB instance is running properly.
Monitoring metrics of the DB instances that are faulty or have been deleted cannot be displayed on the Cloud Eye console. You can view their monitoring metrics after they are rebooted or restored to be normal.

NOTE

If a DB instance has been faulty for 24 hours, Cloud Eye considers that it does not exist and deletes it from the monitoring object list. You need to manually clear the alarm rules created for the DB instance.

- The DB instance keeps running properly for about 10 minutes.
For a newly created DB instance, you need to wait for a while before viewing the monitoring metrics.

Viewing Metrics of DB Instances

Step 1 [Log in to the management console.](#)

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

Step 3 Under **Management & Governance** of the service list, click **Cloud Eye**.

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

Step 5 Click the target instance name to view its monitoring information.

Cloud Eye can monitor performance metrics in the last 1 hour, last 3 hours, last 12 hours, last 24 hours, or last 7 days.

You can also click  in the upper left corner of the page and choose **Databases > GaussDB**. On the **Instances** page, click **View Metric** in the **Operation** column of the row containing the target instance to go to the Cloud Eye console. Alternatively, click the name of the target DB instance on the **Instances** page. On the displayed page, click **View Metric** in the upper right corner to go to the Cloud Eye console.

----End

15.3 Checking GaussDB Monitoring Dashboards

Scenarios

On the GaussDB console, you can see real-time performance metrics for your instances and historical performance metrics on different components of the specified instance.

Precautions

- The instance and nodes must be specified.
- You can select a maximum of nine nodes at a time.
- You can select a maximum of nine components at a time.

Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 Choose **Monitoring and Inspection > Monitoring Dashboards**.

Step 5 On the **Monitoring Dashboards** page, select the specified instance and nodes.

Figure 15-1 Selecting an instance

Instances

Search by DB Instance Name by default.

Name/ID	DB Instance ...	DB Engine Version	Status
	Distributed	GaussDB V2.0-8.201.0	Available
	Centralized	GaussDB V2.0-8.201.0	Available

Figure 15-2 Selecting nodes

Nodes ?

Search by Node Name by default.

☒ All ☒ gauss-f954_root_2 (slave) ☒ gauss-f954_root_0 (master) ☒ gauss-f954_root_1 (slave)

Step 6 Click **View Metrics**.

Step 7 Select a time segment and click to view corresponding metric data.

Figure 15-3 Viewing metric data

Performance Metrics

Last hour Last 6 hours Last 12 hours Last day Last 7 days Start Date - End Date

☐ CPU/Memory

☐ Disk/Storage

☐ Network

☐ Connections

☐ Workloads

☐ Locks

☐ Synchronization Status

☐ Process Resources

----End

15.4 Creating Alarm Rules for a GaussDB Instance

Scenarios

You can set alarm rules to customize the monitored objects and notification policies and to stay aware of the operating status.

The alarm rules include alarm rule names, resource type, dimensions, monitored objects, metrics, alarm thresholds, monitoring period, and whether to send notifications.

Procedure

- Step 1** [Log in to the management console.](#)
- Step 2** Under **Management & Governance** of the service list, click **Cloud Eye**.
- Step 3** In the navigation pane on the left, choose **Cloud Service Monitoring**.
- Step 4** Click **GaussDB** in the list.
- Step 5** Locate the instance for which you want to create an alarm rule, click **More** in the **Operation** column, and choose **Create Alarm Rule**.
- Step 6** On the displayed page, set parameters as required.

Table 15-3 Alarm rule information

Parameter	Description
Name	Alarm rule name. The system generates a random name, and you can change it if needed. The value can contain only letters, digits, underscores (_), and hyphens (-), and cannot exceed 128 characters.
Description	Description of the alarm rule. The value can contain a maximum of 256 characters. This parameter is optional.
Method	Mode for configuring an alarm policy. You can select Associate template or Configure manually . • Configure manually: You can create a custom alarm policy as needed. • Associate template: If the same alarm rule needs to be configured for multiple GaussDB instances, you can use an alarm template to simplify operation.
Template	This parameter is only available if you select Associate template for Method . You can select a default alarm template or create a custom one. After an associated template is modified, the policies contained in this alarm rule to be created will be updated accordingly.
Alarm Policy	This parameter is only available if you select Configure manually for Method . An alarm is triggered when the metric configured for this alarm reaches the preset threshold in consecutive periods. For example, an alarm is triggered if the average CPU usage is 80% or higher for three consecutive 5-minute periods. A maximum of 50 alarm policies can be added to an alarm rule. If any of these alarm policies is met, an alarm will be triggered.

Table 15-4 Alarm notification

Parameter	Description
Alarm Notifications	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text messages, or through HTTP/HTTPS requests to servers. This function is enabled by default.
Notified By	The following three options are available: • Notification policies: Flexible alarm notifications by severity and more notification channels are provided. • Notification groups: Configure notification templates on the Cloud Eye console. • Topic subscriptions: Configure notification templates on the Simple Message Notification (SMN) console.
Notification Policies	This parameter is only available if you select Notification policies for Notified By . Select one or more notification policies. You can specify the notification group, window, template, and other parameters in a notification policy. For how to create a notification policy, see Creating, Modifying, or Deleting a Notification Policy .
Notification Group	This parameter is only available if you select Notification groups for Notified By . Select the notification groups to which alarm notifications will be sent. For details about how to create a notification group, see Creating a Recipient and Notification Group .
Recipient	This parameter is only available if you select Topic subscriptions for Notified By . You can select the account contact or a topic as the object to which alarm notifications will be sent. • The account contact is the mobile phone number and email address of the registered account. • A topic is a specific event type for publishing messages or subscribing to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see Creating a Topic and Adding Subscriptions.
Notification Template	This parameter is only available if you select Notification groups or Topic subscriptions for Notified By . You can select an existing template or create a new one to send alarm notifications.
Notification Window	This parameter is only available if you select Notification groups or Topic subscriptions for Notified By . Cloud Eye sends notifications only within the notification window you specified. If Notification Window is set to 08:00-20:00 , alarm notifications are sent only from 08:00 to 20:00.

Parameter	Description
Trigger Condition	This parameter is only available if you select Notification groups or Topic subscriptions for Notified By . You can select either Generated alarm or Cleared alarm , or both.
Enterprise Project	Enterprise project that the alarm rule belongs to. Only users who have the permissions of the enterprise project can view and manage this alarm rule.
Tags	Key-value pairs that you can use to easily categorize and search for cloud resources. You are advised to create predefined tags in Tag Management Service (TMS). If your organization has configured tag policies for Cloud Eye, you need to add tags to alarm rules based on tag policies. If a tag does not comply with the policies, an alarm rule may fail to be created. Contact your organization administrator to learn more about tag policies. • A key can contain up to 128 characters, and a value can contain up to 225 characters. • You can add up to 20 tags.

Step 7 Click **Create**. The alarm rule is created.

For details about how to create alarm rules, see [Creating an Alarm Rule](#).

----End

15.5 Event Monitoring

15.5.1 Supported Events of GaussDB

Event monitoring provides event data reporting, query, and alarm reporting. You can create alarm rules for both system and custom events. When specific events occur, Cloud Eye generates alarms for you.

Events are key operations on database resources that are stored and monitored by Cloud Eye. You can view events to see operations performed by specific users on specific resources, for example, changing instance specifications.

Event monitoring provides an API for reporting custom events, which helps you collect and report abnormal events or important change events generated by services to Cloud Eye.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events. For details about system events, see [Table 15-5](#).

Table 15-5 Events supported by GaussDB

Source	Name	ID	Severity	Description	Handling Suggestion	Impact
GaussDB	Process status alarm	ProcessStatusAlarm	Major	Key processes exit, including CMS/CMA, ETCD, GTM, CN, and DN processes.	Wait until the process is automatically recovered or a primary/standby failover is automatically performed. Check whether services are recovered. If services are not recovered, submit a service ticket .	If processes on primary nodes are faulty, services are interrupted and then rolled back. If processes on standby nodes are faulty, services are not affected.
	Component status alarm	ComponentStatusAlarm	Major	Key components do not respond, including CMA, ETCD, GTM, CN, and DN components.	Wait until the process is automatically recovered or a primary/standby failover is automatically performed. Check whether services are recovered. If services are not recovered, submit a service ticket .	If processes on primary nodes do not respond, neither do the services. If processes on standby nodes are faulty, services are not affected.

Source	Name	ID	Severity	Description	Handling Suggestion	Impact
	Cluster status alarm	ClusterStatusAlarm	Major	The cluster is abnormal, including the following faults: The cluster is read-only. The majority of ETCD members are faulty. The cluster resources are unevenly distributed.	To handle the event, submit a service ticket .	If the cluster status is read-only, only read requests are processed. If the majority of ETCD members are faulty, the cluster is unavailable. If resources are unevenly distributed, the instance performance and reliability deteriorate.
	Hardware resource alarm	HardwareResourceAlarm	Major	A major hardware fault occurs in the instance, such as disk damage or GTM network fault.	To handle the event, submit a service ticket .	Some or all services are affected.
	Status transition alarm	StateTransitionAlarm	Major	The following events occur in the instance: DN build attempt, DN build failure, forcible DN promotion, primary/standby DN switchover/failover, or primary/standby GTM switchover/failover.	Wait until the fault is automatically rectified and check whether services are recovered. If services are not recovered, submit a service ticket .	Some services are interrupted.

Source	Name	ID	Severity	Description	Handling Suggestion	Impact
	Other abnormal alarm	Other AbnormalAlarm	Major	A kernel alarm that does not fall under process status, component status, cluster status, or status transition alarms is reported. For example, a CN removal, CN addition, archive media exception, consecutive archiving failures, or disk usage threshold alarm is reported.	To handle the event, submit a service ticket .	Some services are interrupted.
	Instance running status abnormal	GaussDBV5FaultyDBInstance	Major	This event is a key alarm event and is reported when an instance is faulty due to a disaster or a server failure.	To handle the event, submit a service ticket .	The database service may be unavailable.
	Instance running status recovery	GaussDBV5InstanceRecovered	Major	If a disaster occurs, GaussDB provides an HA tool to automatically or manually rectify the fault. After the fault is rectified, this event is reported.	No further action is required.	None

Source	Name	ID	Severity	Description	Handling Suggestion	Impact
	Faulty DB node	Gauss DBV5 Faulty DBNode	Major	This event is a key alarm event and is reported when a database node is faulty due to a disaster or a server failure.	Check whether the database service is available and submit a service ticket .	The database service may be unavailable.
	DB node recovered	Gauss DBV5 Faulty DBNodeRecovered	Major	If a disaster occurs, GaussDB provides an HA tool to automatically or manually rectify the fault. After the fault is rectified, this event is reported.	No further action is required.	None
	Instance creation failure	Gauss DBV5 Create InstanceFailed	Major	Instances fail to be created because the quota is insufficient or underlying resources are exhausted.	Delete the instances that are no longer used and try to provision new instances again, or submit a service ticket to adjust the resource quota.	Instances fail to be created.

Source	Name	ID	Severity	Description	Handling Suggestion	Impact
	Node adding failure	Gauss DBV5 ExpandClusterFailed	Major	The underlying resources are insufficient.	To handle the event, submit a service ticket to coordinate resources. Then, delete the node that failed to be added and add a new one.	None
	Storage scale-up failure	Gauss DBV5EnlargeVolumeFailed	Major	The underlying resources are insufficient.	To handle the event, submit a service ticket to coordinate resources. Then, scale up the storage again.	Services may be interrupted.
	Reboot failure	Gauss DBV5RestartInstanceFailed	Major	The network is abnormal.	Retry the reboot operation or submit a service ticket .	The database service may be unavailable.
	Full backup failure	Gauss DBV5FullBackupFailed	Major	The backup files fail to be exported or uploaded.	To handle the event, submit a service ticket .	Data cannot be backed up.
	Differential backup failure	Gauss DBV5DifferentialBackupFailed	Major	The backup files fail to be exported or uploaded.	To handle the event, submit a service ticket .	Data cannot be backed up.

Source	Name	ID	Severity	Description	Handling Suggestion	Impact
	Backup deletion failure	Gauss DBV5 Delete BackupFailed	Major	Backup files fail to be cleared.	To handle the event, submit a service ticket .	There may be residual OBS files.
	EIP binding failure	Gauss DBV5 BindEIPFailed	Major	The EIP has been used or EIP resources are insufficient.	To handle the event, submit a service ticket .	The instance cannot be accessed from the public network.
	EIP unbinding failure	Gauss DBV5 UnbindEIPFailed	Major	The network or the EIP service is faulty.	Unbind the IP address again or submit a service ticket .	There may be residual IP resources.
	Parameter template application failure	Gauss DBV5 UpdateInstanceParameterGroupFailed	Major	Changing a parameter group times out.	Change the parameter group again.	None
	Parameter modification failure	Gauss DBV5 ApplyParameterFailed	Major	Changing a parameter group times out.	Change the parameter group again.	None
	Backup and restoration failure	Gauss DBV5 RestoreFromBackupFailed	Major	The underlying resources are insufficient or backup files fail to be downloaded.	To handle the event, submit a service ticket .	The database service may be unavailable during the restoration failure.

15.5.2 Checking GaussDB Event Monitoring Data

Scenarios

Event monitoring provides event data reporting, query, and alarm reporting. When there are specified events, you will receive alarm notifications from Cloud Eye.

Event monitoring is enabled by default. You can view monitoring details about system events and custom events.

This section describes how to view the event monitoring data.

Procedure

Step 1 [Log in to the management console.](#)

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

Step 3 Click  in the upper left corner of the page. Under **Management & Governance**, click **Cloud Eye**.

You can also click  in the upper left corner of the page and choose **Databases > GaussDB**. On the **Instances** page, click **View Metric** in the **Operation** column of the row containing the target instance to go to the Cloud Eye console. Alternatively, click the name of the target instance on the **Instances** page. On the displayed page, click **View Metric** in the upper right corner to go to the Cloud Eye console. Then, click  to return to the main page of Cloud Eye.

Step 4 In the navigation pane on the left, choose **Event Monitoring**.

Step 5 On the displayed page, check all system events of the last 24 hours that are displayed by default.

You can also click **1h**, **3h**, **12h**, **1d**, **7d**, or **30d** to view the events generated in different periods.

Step 6 Expand an event, and click **View Event** in the **Operation** column to view details about a specific event.

----End

15.5.3 Creating an Alarm Rule to Monitor a GaussDB Event

Scenarios

You can create alarm rules for event monitoring.

Procedure

Step 1 [Log in to the management console.](#)

Step 2 Click  in the upper left corner of the page, and choose **Management & Governance > Cloud Eye**.

Step 3 In the navigation pane, choose **Event Monitoring**. On the **Event Monitoring** page, click **Create Alarm Rule**.

Step 4 On the **Create Alarm Rule** page, configure required parameters.

Table 15-6 Parameters for creating an alarm rule

Parameter	Description
Name	Name of the alarm rule. The system generates a random name, and you can change it if needed.
Description	Supplementary information about the alarm rule. This parameter is optional.
Alarm Type	Alarm type corresponding to the alarm rule.
Event Type	Event type of the metric corresponding to the alarm rule.
Event Source	Service the event is generated for. Select GaussDB .
Monitoring Scope	Monitoring scope for event monitoring.
Method	Method you use to create the alarm rule. You can select Configure manually .
Event Name	Instantaneous operations users performed on system resources, such as login and logout.
Triggering mode	Select Immediate trigger or Accumulative trigger based on the operation severity.
Alarm Policy	Policy that triggers an alarm. For example, an alarm is triggered if the event occurred for three consecutive periods of 5 minutes. This parameter is mandatory when the triggering mode is set to Accumulative trigger .
Alarm Severity	Alarm severity, which can be Critical , Major , Minor , or Informational .
Operation	You can click Delete to delete an alarm policy.

Toggle on  next to the **Alarm Notification** field to enable alarm notification. The notification window is 24 hours by default. If the topics you need are not displayed in the **Notification Object** drop-down list, click **Create an SMN topic** first. Then, select **Generated alarm** and **Cleared alarm** for **Trigger Condition**.

 NOTE

Cloud Eye sends notifications only within the notification window specified in the alarm rule.

Table 15-7 Parameters for setting alarm notifications

Parameter	Description
Alarm Notification	Specifies whether to notify users when alarms are triggered. Notifications can be sent by email or text message, or through HTTP/HTTPS request to servers.
Notification Recipient	There are two options: Notification group and Topic subscription .
Notification Group	Notification group the alarm notification is to be sent to.
Notification Object	Object that receives alarm notifications. You can select the account contact or a topic. - Account contact is the mobile phone number and email address provided for registration. - A topic is used to publish messages and subscribe to notifications. If the required topic is unavailable, create one first and add subscriptions to it. For details, see Creating a Topic and Adding Subscriptions .
Notification Window	Cloud Eye sends notifications only within the validity period specified in the alarm rule. If Notification Window is set to 08:00-20:00 , Cloud Eye sends notifications only within 08:00-20:00.
Trigger Condition	Condition for triggering an alarm notification. You can select Generated alarm (when an alarm is generated), Cleared alarm (when an alarm is cleared), or both.

Step 5 Click **Create**. The alarm rule is created.

For details about alarm rule parameters, see [Creating an Alarm Rule](#) in *Cloud Eye User Guide*.

----End

16 Logs and Auditing

16.1 Downloading Error Logs and Slow Query Logs of a GaussDB Instance

GaussDB allows you to download slow query logs and error logs. Slow query logs help you locate slow SQL statement execution problems. Error logs help you locate instance problems.

Precautions

- CNs and DN of the instance are normal.
- The IaaS network is normal.
- Logs generated in the last 12 hours are collected for the analysis of slow query logs.

Slow Query Logs

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Log Analysis**.

Step 6 The system checks whether there has been a slow query log task in the last 5 minutes and, if there is not, generates one. Click the **Slow Query Logs** tab. Click **Download** in the **Operation** column of the record whose status is **Preparation completed**.

After the log is downloaded, you can analyze the log on the local PC.

Figure 16-1 Downloading a slow query log

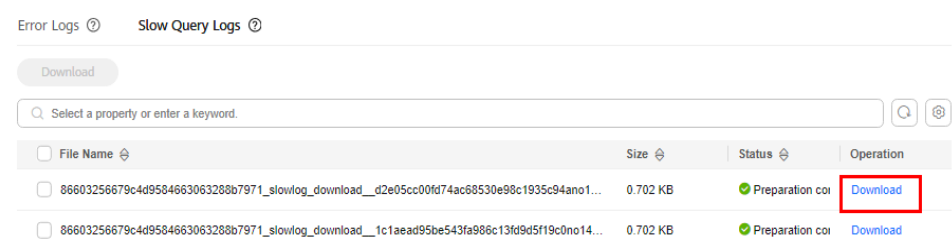


Table 16-1 describes the fields contained in slow query logs.

Table 16-1 Fields in slow query logs

Field	Type	Description
message_version	String	Log format version. The value is fixed at V1 .
db_name	name	Database name.
schema_name	name	Schema name.
origin_node	integer	Node name.
user_name	name	Username.
application_name	text	Name of the application that sends a request.
client_addr	text	IP address of the client that sends a request.
client_port	integer	Port number of the client that sends a request.
unique_query_id	bigint	ID of the normalized SQL statement.
debug_query_id	bigint	ID of the unique SQL statement. Some statements are not unique. For example, the value of debug_query_id in the Parse packet, DCL statements, and TCL statements is 0 .
query	text	Normalized SQL (available only on CNs). When track_stmt_parameter is enabled, complete SQL statements are displayed.
start_time	timestamp with time zone	Time when a statement starts.
finish_time	timestamp with time zone	Time when a statement ends.
slow_sql_threshold	bigint	Standard for slow SQL statement execution.

Field	Type	Description
transaction_id	bigint	Transaction ID.
thread_id	bigint	ID of an execution thread.
session_id	bigint	Session ID of a user.
n_soft_parse	bigint	Number of soft parses. The value of n_soft_parse plus the value of n_hard_parse may be greater than the value of n_calls because the number of subqueries is not counted in the value of n_calls .
n_hard_parse	bigint	Number of hard parses. The value of n_soft_parse plus the value of n_hard_parse may be greater than the value of n_calls because the number of subqueries is not counted in the value of n_calls .
query_plan	text	Statement execution plan.
n_returned_rows	bigint	Number of rows in the result set returned by the SELECT statement.
n_tuples_fetched	bigint	Number of rows randomly scanned.
n_tuples_returned	bigint	Number of rows sequentially scanned.
n_tuples_inserted	bigint	Number of rows inserted.
n_tuples_updated	bigint	Number of rows updated.
n_tuples_deleted	bigint	Number of rows deleted.
n_blocks_fetched	bigint	Number of buffer block access times.
n_blocks_hit	bigint	Number of buffer block hits.
db_time	bigint	Valid DB time, which is accumulated if multiple threads are involved (unit: microsecond).
cpu_time	bigint	CPU time (unit: microsecond).
execution_time	bigint	Execution time in the executor (unit: microsecond).
parse_time	bigint	SQL parsing time (unit: microsecond).

Field	Type	Description
plan_time	bigint	SQL plan generation time (unit: microsecond).
rewrite_time	bigint	SQL rewriting time (unit: microsecond).
pl_execution_time	bigint	Execution time of PL/pgSQL (unit: microsecond).
pl_compilation_time	bigint	Compilation time of PL/pgSQL (unit: microsecond).
data_io_time	bigint	I/O time (unit: microsecond).
net_send_info	text	Network status of messages sent through a physical connection, including the time (in microseconds), number of calls, and throughput (in bytes). In a distributed database, CNs communicate with each other, CNs communicate with the client, and CNs communicate with DN through physical connections. This column can be used to analyze the network overhead of SQL statements in a distributed system. Example: {"time":xxx, "n_calls":xxx, "size":xxx}.
net_rcv_info	text	Network status of messages received through a physical connection, including the time (in microseconds), number of calls, and throughput (in bytes). In a distributed database, CNs communicate with each other, CNs communicate with the client, and CNs communicate with DN through physical connections. This column can be used to analyze the network overhead of SQL statements in a distributed system. Example: {"time":xxx, "n_calls":xxx, "size":xxx}.
net_stream_send_info	text	Network status of messages sent through a logical connection, including the time (in microseconds), number of calls, and throughput (in bytes). In a distributed database, DN of different shards communicate with each other through logical connections. This column can be used to analyze the network overhead of SQL statements in a distributed system. Example: {"time":xxx, "n_calls":xxx, "size":xxx}.

Field	Type	Description
net_stream_recv_info	text	Network status of messages received through a logical connection, including the time (in microseconds), number of calls, and throughput (in bytes). In a distributed database, DNs of different shards communicate with each other through logical connections. This column can be used to analyze the network overhead of SQL statements in a distributed system. Example: {"time":xxx, "n_calls":xxx, "size":xxx}.
lock_count	bigint	Number of locks.
lock_time	bigint	Time required for locking.
lock_wait_count	bigint	Number of lock waits.
lock_wait_time	bigint	Time required for lock waiting.
lock_max_count	bigint	Maximum number of locks.
lwlock_count	bigint	Number of lightweight locks (reserved).
lwlock_wait_count	bigint	Number of lightweight lock waits.
lwlock_time	bigint	Time required for lightweight locking (reserved).

Field	Type	Description
details	bytea	List of wait events and statement lock events. When the value of the record level is greater than or equal to L0, the list of waiting events starts to be recorded. It displays statistics about wait events on the current node. For details about key events, see Table: Waiting State List, Table: List of Wait Events Corresponding to Lightweight Locks, Table: List of I/O Wait Events, and Table: List of Wait Events Corresponding to Transaction Locks. You can also view the list of all events in the system in the wait_event_info view. For details about the impact of each transaction lock on services, see LOCK. When the value of the record level is L2, the list of statement lock events is recorded. The list records events in chronological order. The number of records is affected by the value of the track_stmt_details_size parameter. This field is in binary format and needs to be read by using the parsing function pg_catalog.statement_detail_decode. For details, see Table: statement_detail_decode Parameter Description. Events include: • Start locking. • Complete locking. • Start lock waiting. • Complete lock waiting. • Start unlocking. • Complete unlocking. • Start lightweight lock waiting. • Complete lightweight lock waiting.
is_slow_sql	boolean	Specifies whether the SQL statement is a slow SQL statement. • t (true): yes. • f (false): no.
lwlock_wait_time	bigint	Time required for lightweight lock waiting.

----End

Error Logs

- Step 1** [Log in to the management console](#).
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane, choose **Log Analysis**.
- Step 6** On the displayed page, click the **Error Logs** page, enable **Error Log Collection**, and click **Download** in the **Operation** column of the log whose status is **Preparation completed** to download the error log file. After the log is downloaded, you can analyze the log on the local PC.

Error logs are stored in the **gs_log** directory and are named in the format **gaussdb-creation_time.log**. The default format of each row of logs in the log file is as follows: *Date+Time+Node name+Username+Database name+IP address+Session ID+Transaction ID+Application name+Log level+Log content*

Table 16-2 Parameters of error logs

Field	Description
Date	Date when a log is generated. The format is yyyy-mm-dd.
Time	Time when a log is generated. The format is hh:mm:ss.ms.
Node name	The node to which an error is reported.
Username	Username of the database user who triggers log generation.
Database name	Name of the database that triggers log generation.
IP address	IP address of the client that triggers log generation.
Thread ID	Thread ID.
Session ID	ID of the session that triggers log generation.
Transaction ID	Transaction ID (0 indicates that no transaction ID is assigned).
Thread name	Thread name.

Field	Description
Query ID	ID of a query initiated by a user, which is recorded in the background.
Module name	Module name.
Log level	Log level, such as FATAL , ERROR , or LOG . Different log levels indicate different severities.
Log content	Log content.

-----End

FAQ

How Do I View Deadlock Logs of GaussDB?

16.2 Downloading Switchover/Failover Logs of a GaussDB Instance

You can download switchover/failover logs of a GaussDB instance. If switchover/failover log collection is enabled for an instance whose **Failover Priority** is **Availability**, GaussDB can collect Xlogs that cannot be replayed on the standby node in time when a switchover or failover occurs and convert the Xlogs into a SQL file. You can download the SQL file and run SQL statements to replay the data in the SQL file as required.

Precautions

- CNs and DN of the instance are normal.
- The IaaS network is normal.
- Switchover/Failover logs are available only for distributed instances whose **Failover Priority** is **Availability**.

Switchover/Failover Logs

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Log Analysis**.

Step 6 On the displayed page, click the **Switchover/Failover Logs** page, enable **Switchover/Failover Log Collection**, and click **Download** in the **Operation**

column of the record whose status is **Preparation completed** to download the switchover/failover log file.

----End

16.3 Viewing GaussDB Operation Logs on CTS

With Cloud Trace Service (CTS), you can record operations associated with GaussDB for future query, audit, and backtracking.

GaussDB Operations That Can Be Recorded by CTS

Table 16-3 Operations supported by CTS

Operation	Resource Type	Trace Name
Creating a DB instance or restoring data to a new DB instance	instance	createInstance
Deleting an instance	instance	deleteInstance
Changing instance specifications	instance	resizeFlavor
Upgrading the instance version	instance	upgradeVersion
Resetting a password	instance	resetPassword
Rebooting a DB instance	instance	instanceRestart
Binding an EIP	instance	setOrResetPublicIP
Unbinding an EIP	instance	setOrResetPublicIP
Modifying resource tags	instance	modifyTag
Deleting resource tags	instance	deleteTag
Adding resource tags	instance	createTag
Renaming a DB instance	instance	renameInstance
Adding nodes	instance	instanceAction
Deleting task records	workflowTask	deleteTaskRecord

Operation	Resource Type	Trace Name
Reducing the number of replicas	instance	reduceReplica
Deleting coordinator nodes	instance	reduceCoordinator-Node
Modifying the recycling policy	backup	setRecyclePolicy
Creating a manual backup	backup	createManualSnapshot
Deleting a manual backup	backup	deleteManualSnapshot
Modifying the backup policy	backup	setBackupPolicy
Restoring a DB instance	backup	restoreInstance
Restoring data of an instance using a backup	instance	restoreInstance
Changing the retention period of automated backups	instance	setBackupPolicy
Creating a parameter group	parameterGroup	createParameterGroup
Applying a parameter group	parameterGroup	applyParameterGroup
Replicating a parameter group	parameterGroup	copyParameterGroup
Deleting a parameter group	parameterGroup	deleteParameterGroup
Resetting a parameter group	parameterGroup	resetParameterGroup
Updating a parameter group	parameterGroup	updateParameterGroup
Changing the port	instance	modifyPort
Creating slow query log download tasks	instance	createSlowLogDownload
Enabling or disabling switchover/failover logs	instance	switchErrorLog

Operation	Resource Type	Trace Name
Scaling up storage for shards	instance	resizeVolume
Modifying storage autoscaling policies	instance	autoEnlargeVolume
Deleting shards	instance	reduceShard
Changing standby data nodes to log nodes	instance	switchReplica
Performing a primary/standby switchover	instance	switchShard
Changing the disk type	instance	changeVolumeType
Starting an instance or node	instance	startInstance
Stopping an instance or node	instance	stopInstance
Changing a single-replica instance to an instance with primary and standby nodes	instance	changeDeployment-Solution
Enabling advanced features	instance	updateFeatures
Updating the policy of agency permissions	feature	updateAgency

Querying Audit Logs

You can query GaussDB traces (audit logs) on the CTS console. For details, see [Querying Real-Time Traces](#).

16.4 Interconnecting with LTS and Querying Database Audit Logs

Scenarios

Audit logs capture database start, stop, and connection events, along with DDL, DML, and DCL operations. Each log includes the event time, type, result,

username, database, connection details, affected object, instance name, and port number. Security administrators can use these logs to reconstruct events leading to the current database state and identify unauthorized operations, including who performed them, when, and what was done, strengthening traceability and accountability.

If **Upload Audit Logs to LTS** is enabled for a GaussDB instance, new audit logs of it are automatically uploaded to Log Tank Service (LTS) and you can search for logs, monitor logs, download logs, and view real-time logs on the LTS console.

Precautions

- The DB engine version must be V2.0-2.1.0 or later.
- Audit logs record all requests sent to your DB instance and are stored in LTS.
- Toggling on or off this function will not be applied immediately. There is a delay of about 10 minutes.
- For details about how to enable or disable the audit log function, configure [audit_enabled](#).
- For details about the parameters for controlling audit logs, see [Audit Items](#).
- You will be billed for this function. For details, see [LTS Pricing Details](#).
- After this function is enabled, audit policies you configured are reported to LTS by default.

Enabling Upload Audit Logs to LTS

Step 1 [Log in to the management console](#).

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

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

Step 4 In the navigation pane on the left, choose **Instances**.

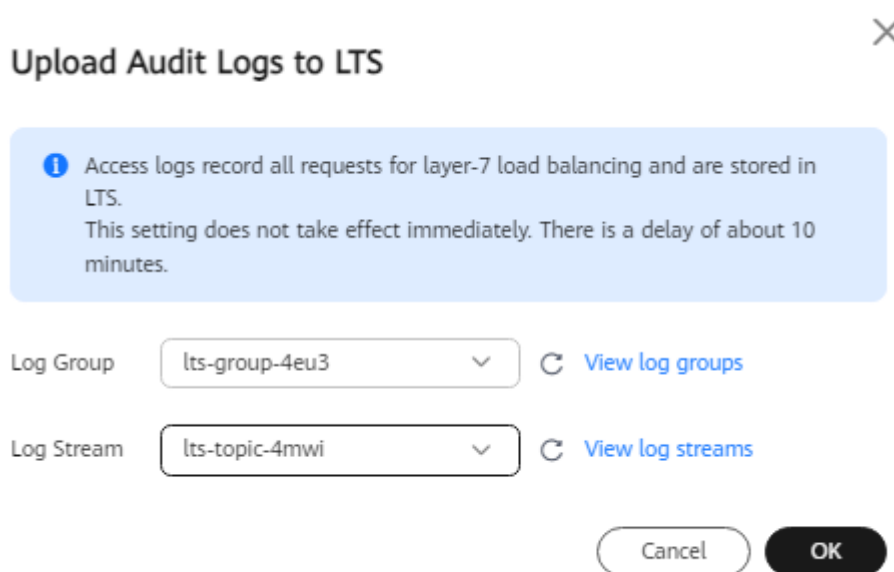
Step 5 Click the instance name to go to the **Basic Information** page.

Step 6 In the navigation pane on the left, click **Audit Logs**.

Step 7 Click  next to **Upload Audit Logs to LTS**.

Step 8 In the displayed dialog box, configure **Log Group** and **Log Stream**.

If you enable log reporting to LTS for the first time, click **View Log Groups** to log in to the LTS console and configure log groups and log streams. For details, see [Managing Log Groups](#) and [Managing Log Streams](#).

Figure 16-2 Enabling Upload Audit Logs to LTS

Step 9 Click **OK**.

After this function is enabled, audit logs will not be uploaded immediately to LTS. There is a delay of about 10 minutes. For details, see [Viewing Real-Time Logs](#).

----End

Disabling Upload Audit Logs to LTS

Step 1 [Log in to the management console](#).

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

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

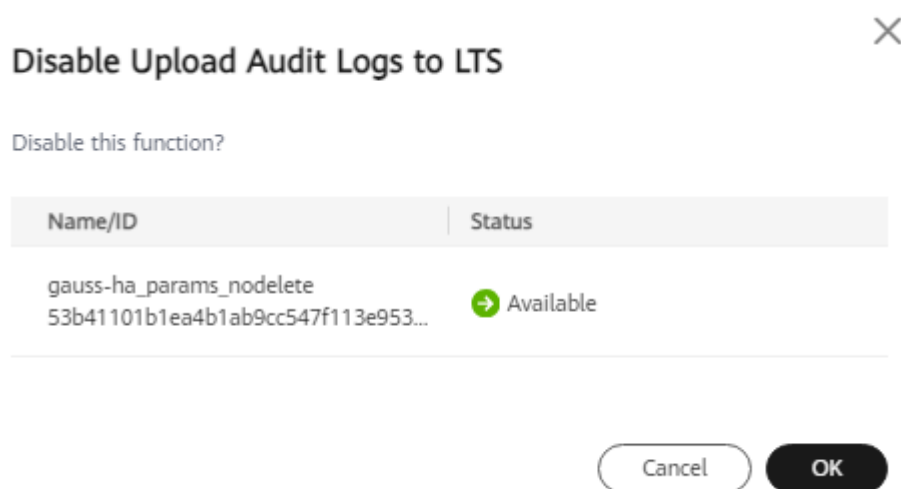
Step 4 In the navigation pane on the left, choose **Instances**.

Step 5 Click the instance name to go to the **Basic Information** page.

Step 6 In the navigation pane on the left, click **Audit Logs**.

Step 7 Click  next to **Upload Audit Logs to LTS**.

Step 8 In the displayed dialog box, confirm the information.

Figure 16-3 Disabling Upload Audit Logs to LTS

Step 9 In the displayed dialog box, click **OK**.

-----End

17 Quota Adjustment

17.1 Adjusting Cloud Service Resource Quotas of GaussDB

What Is a Quota?

A quota is a limit on the quantity or capacity of a certain type of service resources available to you. Examples of GaussDB quotas include the maximum number of GaussDB instances that you can create. Quotas are put in place to prevent excessive resource usage.

If the existing resource quotas cannot meet your service requirements, you can request higher quotas.

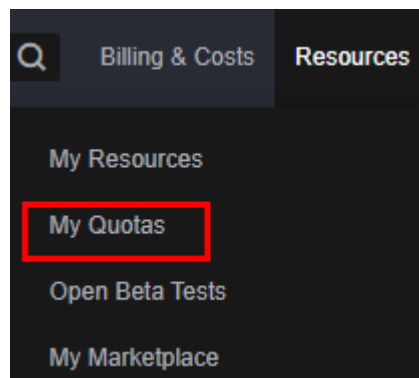
How Do I View My Quotas?

Step 1 [Log in to the management console.](#)

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

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

Figure 17-1 My Quotas



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

----End

How Do I Apply for a Higher Quota?

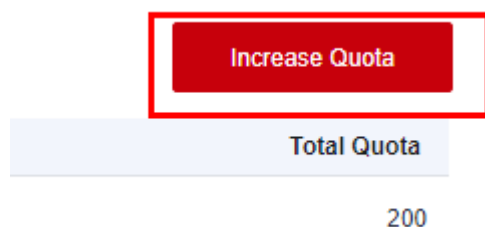
Step 1 [Log in to the management console.](#)

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

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

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

Figure 17-2 Increase Quota



Step 5 On the **Create Service Ticket** page, configure parameters as required.

In the **Problem Description** area, fill in the content and reason for quota adjustment.

Step 6 Read the agreements and confirm that you agree to them, and then click **Submit**.

----End

17.2 Adjusting GaussDB Resource Quotas of an Enterprise Project

The GaussDB management console on Huawei Cloud provides quota management for enterprise projects of tenants.

Quota management is available only for enterprise accounts configured in the whitelist. To apply for the permissions required, [submit a service ticket](#).

Managing Quotas

Step 1 [Log in to the management console.](#)

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

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

Step 4 In the navigation pane on the left, click **Quotas**.

Figure 17-3 Quotas

GaussDB

Overview

Instances

Monitoring and Inspection

Backups

Parameter Templates

Task Center

Recycle Bin

Quotas

Quotas

Q. Enter an enterprise project name or select another filter from the drop-down list to search.

Enterprise Project	Instances (Used/T...	vCPUs (Used/Total)	GB of Memory (Us...	GB of Storage (Us...	Operation
default	39/Unlimited	418/Unlimited	2240/Unlimited	16096/Unlimited	Edit
	1/Unlimited	48/Unlimited	192/Unlimited	360/Unlimited	Edit
	0/Unlimited	0/Unlimited	0/Unlimited	400/Unlimited	Edit
	0/Unlimited	0/Unlimited	0/Unlimited	0/Unlimited	Edit
	0/Unlimited	0/Unlimited	0/Unlimited	0/Unlimited	Edit

On this page, you can view the usage of instances, vCPUs, memory, and storage under each project.

Step 5 Locate the enterprise project to be managed, and click **Edit** in the **Operation** column.

Figure 17-4 Modifying quotas

Edit Quota

The text box in gray indicates the used quota. The quota you enter cannot be less than the used quota.

Enterprise Project

default

Instances (Used/Total)

39

/

-1

vCPUs (Used/Total)

418

/

-1

Memory (Used/Total)

2240

/

-1

GB

Storage (Used/Total)

16096

/

-1

GB

Cancel

OK

Table 17-1 Parameter description

Category	Description
Instances	• The first number indicates the number of existing instances in the enterprise project. • The second number indicates the maximum number of instances that can be created in the enterprise project. The minimum value must be greater than or equal to the number of existing instances. The maximum value is 100000. If this parameter is set to -1, the number is not limited.
vCPUs	• The first number indicates the number of vCPUs used by existing instances in the enterprise project. • The second number indicates the maximum number of vCPUs that can be used by instances in the enterprise project. The minimum value must be greater than or equal to the number of vCPUs used by existing instances. The maximum value is 2147483646. If this parameter is set to -1, the number is not limited.
Memory (GB)	• The first number indicates the size of memory used by existing instances in the enterprise project. • The second number indicates the maximum size of memory that can be used by instances in the enterprise project. The minimum value must be greater than or equal to the size of memory used by existing instances. The maximum value is 2147483646. If this parameter is set to -1, the number is not limited.
Storage (GB)	• The first number indicates the storage space used by existing instances in the enterprise project. • The second number indicates the maximum storage space that can be used by instances in the enterprise project. The minimum value must be greater than or equal to the storage space used by existing instances. The maximum value is 2147483646. If this parameter is set to -1, the number is not limited.

 NOTE

When you access the **Quotas** page for the first time, **Settings** is displayed.

Step 6 In the displayed dialog box, enter a new quota. Click **OK**.

Step 7 On the **Quotas** page, check whether the quotas are changed.

-----**End**

18 Diagnostics and Optimization

18.1 Viewing Slow SQL Statements

Scenarios

When your GaussDB instance encounters slow SQL queries, you can use the following features for optimization:

- SQL throttling: Control the access frequency or concurrency of abnormal SQL statements to ensure service availability.
- SQL plan: Bind or unbind specific execution plans to influence query behavior.
- Table definition query: Extract table names from SQL texts and retrieve corresponding table definitions based on table name, schema name, and database name.

Constraints

- The database kernel does not permanently store slow SQL statement data. Slow SQL statement data for a DB instance will be cleared after any of the following operations is performed on the instance: rebooting or upgrading an instance, changing the instance specifications, starting or stopping a node, performing a primary/standby switchover, or changing the deployment model.
- The GaussDB instances must run on HCEOS.
- SQL plans are only available for centralized instances.
- Table definition query is available for instances whose DB engine version is V2.0-3.208.0 or later and kernel version is 503.1.0.SPC1200 or later. For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).
- The maximum time span of slow SQL statement query is one day.
- SQL throttling cannot be performed for system-level users.

Viewing Slow SQL Statements

Step 1 [Log in to the management console](#).

- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** Choose **Diagnostics and Optimization > SQL Views**.
- Step 6** On the **Slow SQL** tab, select one or more nodes, set the time range, apply combined filters using SQL text, normalized SQL ID, and other extended fields, and click **Query** to obtain the slow SQL statement list.
- Data of a maximum of one day can be queried. You can select a time range as required and view data by node.
 - **Filter System Users:** Toggle this option on if you want to ignore all SQL statements executed by system users. It is toggled on by default.
 - You can select **OR** or **AND** to define how multiple SQL statements are combined during filtering. In the filter box, you may enter up to five SQL text snippets for combined filtering.
 - **Execution Longer Than:** Set this parameter to a value greater than or equal to **1s** to query the execution information of the SQL statements whose execution time is greater than or equal to the preset value. This threshold is controlled by the **log_min_duration_statement** parameter. The default minimum execution duration for determining a slow SQL statement is 3s. You can adjust the threshold on the **Parameters** page. For details about how to modify parameters, see [Modifying Instance Parameters](#).

Table 18-1 Key slow SQL parameters

Parameter	Description
Normalized SQL ID	Normalized ID of the slow SQL statement
Node Name/ID	Name and ID of the node where the SQL statement is executed
Username	Name of the user who executed the SQL statement
SQL Text	Normalized template of the slow SQL statement
SQL Plan	SQL execution plan
Slow SQL Queries	Number of times that the slow SQL statement is executed within a specified time range
Avg. Execution Duration (s)	Average execution time (in seconds) of a slow SQL template within a specified time range
Avg. CPU Duration (s)	CPU time, in seconds
Avg. I/O Duration (s)	I/O time, in seconds
Avg. Rows Returned	Number of rows in the result set returned by the SELECT statement

Parameter	Description
Avg. Rows Scanned	Number of rows scanned by the SQL statement
Buffer Hit Rate	Buffer cache efficiency

Step 7 Click **View SQL** in the **Operation** column to view the collected SQL text and the parsed original SQL text.

When **track_stmt_parameter** is enabled, variables in the collected SQL text are replaced with their actual values, so you can see the original SQL statement. However, for SQL text collected when **track_stmt_parameter** is disabled, the original values of the variables cannot be retrieved, and no content will be displayed here. For details about how to modify parameters, see [Modifying Instance Parameters](#).

Step 8 Click **SQL Throttling** in the **Operation** column to apply concurrency control to the normalized SQL ID of the selected slow SQL statement. For details about how to create a SQL throttling task, see [Configuring SQL Throttling Rules](#).

----End

SQL Plan

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Views**.

Step 6 On the **Slow SQL** tab, obtain the slow SQL statements.

Step 7 In the **Operation** column of a specified record, click **More** and choose **SQL Plan** to bind or unbind execution plans.

Table 18-2 Columns in the Bind or Unbind Execution Plans area

Column Name	Description
Execution Plan	SQL execution plan.
Unique SQL ID	Unique ID of a SQL statement in the database.
Plan ID	Unique ID of the plan used by the current SQL statement.
Cost	Total plan cost.

Column Name	Description
Binding Status	Plan status. Plans in the Binding state can be unbound, and those in the Unbound state can be bound. • Binding • Unbound

Step 8 In the displayed dialog box, locate an execution plan, click  in the **Binding Status** column of the **Bind or Unbind Execution Plans** area to bind or unbind the plan, and click **OK**.

----End

Creating a SQL Patch

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Views**.

Step 6 On the **Slow SQL** tab, obtain the slow SQL statements.

Step 7 In the **Operation** column of a specified record, click **More** and choose **SQL Patch** to view SQL patch details.

- If no SQL patch is created, enter the patch name and content, and click **OK** to create one.

Figure 18-1 SQL patch

SQL Patch Details

Tips:

- For a distributed instance, the SQL Patch is only applied for the current CN. If a new node is added, bind the SQL Patch again to apply it to the new node.

Patch Name

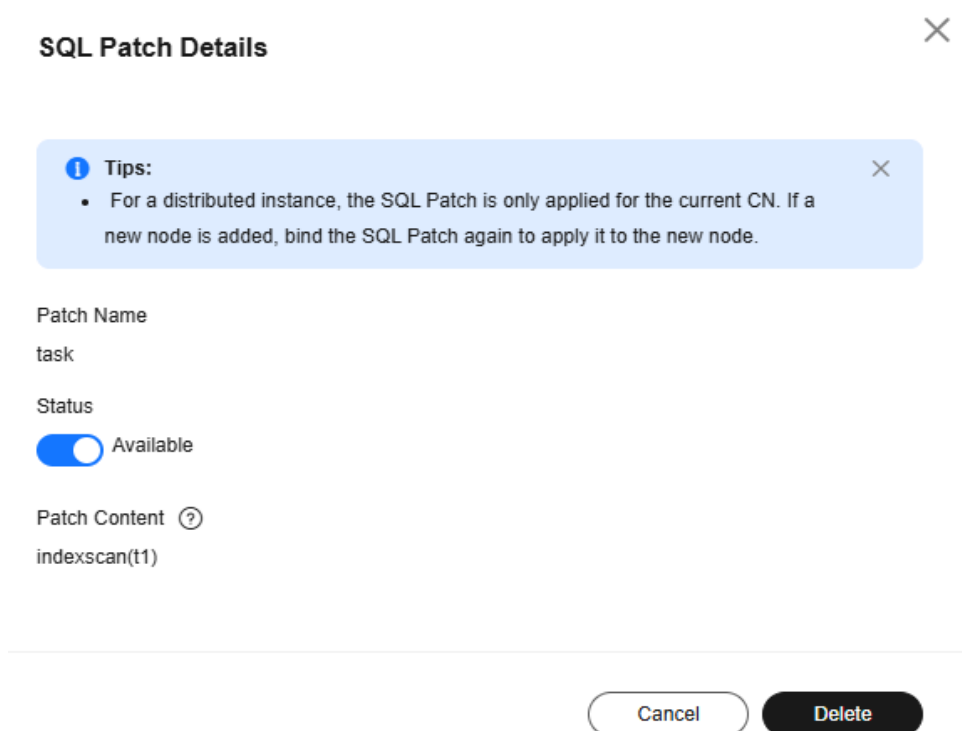
Patch Content ?

0/1,024

Cancel

OK

- If a SQL patch has been created, the SQL patch information is displayed.

Figure 18-2 Viewing SQL patch details

- **Status:** Click ☐ to enable or disable the SQL patch. If a SQL patch is disabled, the status is **Disabled**. If it is enabled, the status is **Available**.
- To delete a SQL patch, click **Delete**.

-----End

Querying Table Definitions

Step 1 Log in to the management console.

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Views**.

Step 6 On the **Slow SQL** tab, select one or more nodes, specify the time range, and click **Query** to obtain the slow SQL statement list.

Step 7 In the **Operation** column of a specified record, click **More** and choose **Table Definition**. On the **Table Definition** page, view information in the **Detected Tables** area.

If no table is detected, click **Add Table** and enter a table name manually. If any detection result is incorrect, click **Edit** to modify, and then save the result.

Step 8 Click  next to a table name to view its table definition.

Figure 18-3 Table Definition

Table Definition

SQL Text

Replicate

```
CREATE TABLE sales_management.customers ( customer_id SERIAL PRIMARY KEY,
customer_name VARCHAR(100) NOT NULL, email VARCHAR(100) UNIQUE, phone
VARCHAR(20), address TEXT, created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP);
```

Detected Tables

If no table is detected, click "Add Table" and enter a table name manually. If any detection result is incorrect, click "Edit" to modify, and then save the result. Only regular user tables are supported. Definitions of views and system catalogs are unavailable.

Add Table

Table Name	Operation
customers	Edit Delete

Schema	Table Definition
sales_management	SET search_path = sales_management; CREATE TAB...

Close

Step 9 Click the SQL statement in the table definition to view the definition details.

Figure 18-4 Table definition details

----End

18.2 Viewing Top SQL Statements

Scenarios

GaussDB allows you to use workload diagnosis report (WDR) snapshots to query top SQL statements in a specified period and monitor SQL statement execution from multiple dimensions of resources.

Constraints

- The system retrieves top SQL data from a performance report generated based on two valid snapshots within your specified time range. Make sure there are no kernel reset events (such as node restarts, primary/standby switchovers, and DROP DATABASE operations) between the two snapshots.

- Top SQL data can be displayed only when at least two snapshots are available for comparison in the specified time range. By default, a snapshot is generated every hour.

Viewing Top SQL Statements

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Views**.

Step 6 On the **Top SQL** tab, select one or more nodes, set the time range, apply combined filters using SQL text, normalized SQL ID, and other extended fields, and click **Query** to obtain the top SQL statement list.

- Data of a maximum of one day can be queried. You can select a time range as required and view data by node.
- **Filter System Users:** Toggle this option on if you want to ignore all SQL statements executed by system users. It is toggled on by default.
- You can select **OR** or **AND** to define how multiple SQL statements are combined during filtering. In the filter box, you may enter up to five SQL text snippets for combined filtering.

Table 18-3 Top SQL statement fields

Field	Description
Normalized SQL ID	ID of the normalized form of a SQL statement
Node Name/ID	ID and name of the SQL execution node
Username	Name of the user who executed the SQL statement
SQL Text	Normalized SQL text
Invocation Frequency	The proportion of calls for a specific SQL statement relative to the total number of calls during a given time period
CPU Overhead Rate	The proportion of the CPU usage for a specific SQL statement relative to the total CPU usage during a given time period
I/O Overhead Rate	The proportion of the I/O usage for a specific SQL statement relative to the total I/O usage during a given time period
Invocation Times	The number of times that a specific SQL statement is executed in a given time period

Field	Description
Returned Tuples	The difference in the number of rows returned by the SELECT query of a specific SQL statement within a given time period
Read Tuples	The difference in the number of rows scanned sequentially by a specific SQL statement within a given time period
Avg. Execution Time	The average time taken to execute a specific SQL statement within a given time period Unit: ms
Total Execution Time	The total time taken to execute a specific SQL statement within a given time period Unit: ms
CPU Overhead	The CPU time consumed by a specific SQL statement within a given time period Unit: ms
I/O Overhead	The I/O overhead consumed by a specific SQL statement within a given time period Unit: ms
Min. Execution Duration	The minimum execution time taken to execute a specific SQL statement within a given time period Unit: ms
Max. Execution Duration	The maximum execution time taken to execute a specific SQL statement within a given time period Unit: ms
SQL Hit Ratio	The ratio of rows returned (hit) to the rows scanned by a specific SQL statement within a given time period

----End

Creating a SQL Patch

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Views**.

Step 6 On the **Top SQL** tab, click **SQL Patch** in the **Operation** column to view SQL patch details.

- If no SQL patch is created, enter the patch name and content, and click **OK** to create one.

Figure 18-5 SQL patch

SQL Patch Details

Tips:

- For a distributed instance, the SQL Patch is only applied for the current CN. If a new node is added, bind the SQL Patch again to apply it to the new node.

Patch Name

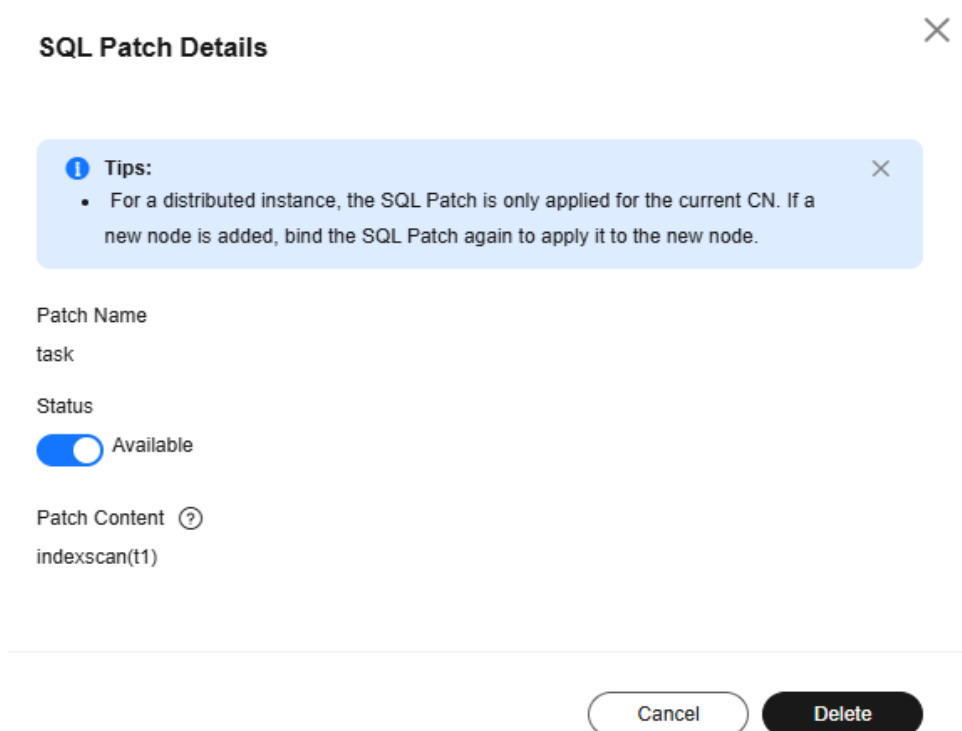
Patch Content ?

0/1,024

Cancel

OK

- If a SQL patch has been created, the SQL patch information is displayed.

Figure 18-6 Viewing SQL patch details

- **Status:** Click  to enable or disable the SQL patch. If a SQL patch is disabled, the status is **Disabled**. If it is enabled, the status is **Available**.
- To delete a SQL patch, click **Delete**.

-----End

18.3 Viewing Full SQL Statistics Using SQL Explorer

Scenarios

SQL Explorer collects and stores all SQL statements executed on your GaussDB instance. This allows you to review SQL statement metrics, execution details, and other related information.

You can configure SQL Explorer to collect data at the instance level, for example, determine whether to filter out system users, specify the default data retention period, and select the types of SQL statements to be collected.

You can also identify table names in SQL text using table definition query or query the table definition based on the table name, schema name, and database name.

Prerequisites

- You have logged in using an account assigned the **gaussdb:instance:listFullSql** and **gaussdb:instance:operateFullSql** actions.

- The login account must be assigned the API actions **lts:groups:***, **lts:topics:***, and **lts:logs:list** of Log Tank Server (LTS).
- The full SQL statistics and table definition functions are only available for authorized users.
- LTS is running properly.
- The CCMS service has been connected for managing built-in accounts of LTS.

Constraints

- To enable SQL Explorer, the DB engine version of GaussDB instances must be V2.0-8.0.1 or later, and the kernel version must be 505.0.0 or later.
- The GaussDB instance and all its nodes to be collected must be in the available state.
- Batch collection and reporting of full SQL data to LTS may be delayed by up to 2 minutes and is not real-time. High concurrency may result in some minor data loss.
- You can query full SQL data generated in the past 30 days.
- Collected data will be automatically deleted upon expiration. You can set the retention period according to your business needs, from a minimum of 1 day to a maximum of 30 days. Expired data cannot be queried.
- Enabling SQL Explorer may temporarily affect database performance.
- Enabling SQL Explorer will automatically connect your instance to LTS. All collected SQL records are continuously stored in LTS (100% indexed fields), which will continuously incur costs. For pricing details of LTS, see [Product Pricing Details](#).
- Ensure that the time (the timestamp of UTC-0) of the management plane node is synchronized with that of each instance node. The time difference cannot exceed 10 seconds. Otherwise, the functionality will be affected.
- Before deleting a GaussDB instance, manually disable SQL Explorer. If you do not need the collected full SQL data, log in to LTS and manually delete the corresponding log group or log stream.
- Table definition query is available for instances whose DB engine version is V2.0-3.208.0 or later and kernel version is 503.1.0.SPC1200 or later. For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).

Enabling SQL Explorer

Step 1 [Log in to the management console](#).

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

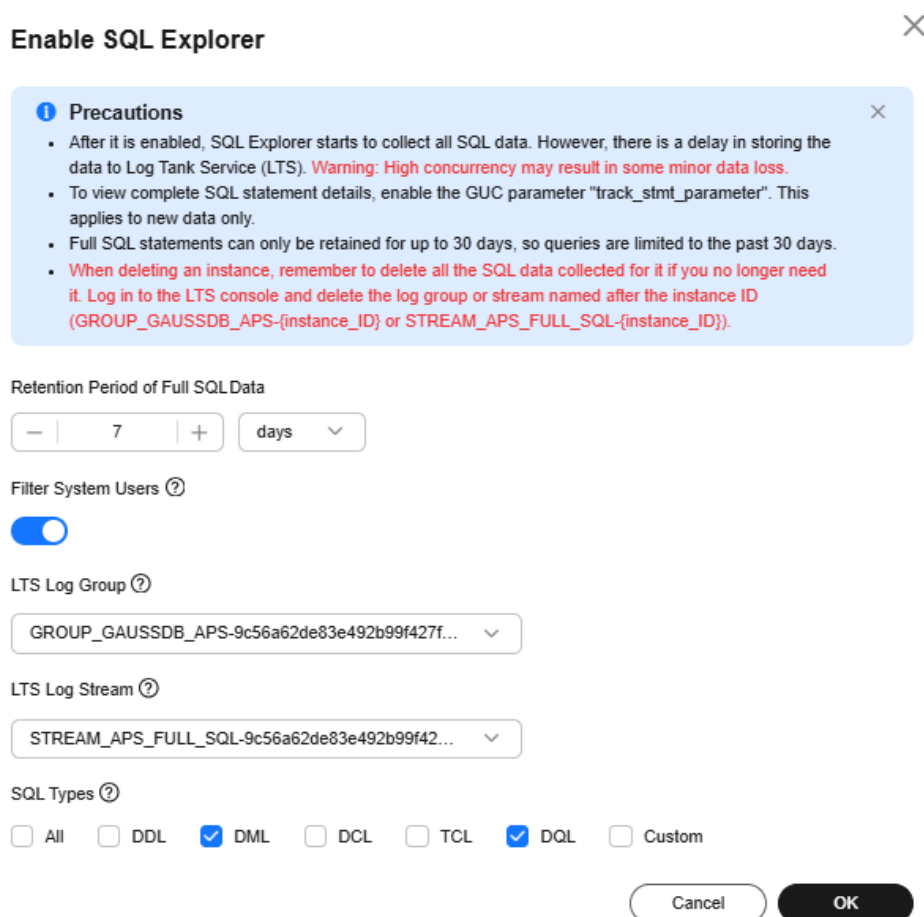
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

- Step 6** Click the **SQL Explorer** tab. On the displayed page, click  next to **Enable SQL Explorer**. In the displayed dialog box, configure the parameters for collecting full SQL data.

Figure 18-7 Enabling SQL Explorer



Enable SQL Explorer

Precautions

- After it is enabled, SQL Explorer starts to collect all SQL data. However, there is a delay in storing the data to Log Tank Service (LTS). **Warning: High concurrency may result in some minor data loss.**
- To view complete SQL statement details, enable the GUC parameter "track_stmt_parameter". This applies to new data only.
- Full SQL statements can only be retained for up to 30 days, so queries are limited to the past 30 days.
- When deleting an instance, remember to delete all the SQL data collected for it if you no longer need it. Log in to the LTS console and delete the log group or stream named after the instance ID (GROUP_GAUSSDB_APS-{instance_ID} or STREAM_APS_FULL_SQL-{instance_ID}).

Retention Period of Full SQL Data

7 days

Filter System Users 



LTS Log Group 

GROUP_GAUSSDB_APS-9c56a62de83e492b99f427f...

LTS Log Stream 

STREAM_APS_FULL_SQL-9c56a62de83e492b99f42...

SQL Types 

☐ All ☐ DDL ☒ DML ☐ DCL ☐ TCL ☒ DQL ☐ Custom

Cancel OK

- Retention Period of Full SQL Data:** Set the number of days for storing collected full SQL data in LTS. Expired data will be automatically deleted. The default value is 7 days. You can set a value from 1 to 30.
- Filter System Users:** Toggle this option on if you want to ignore all SQL statements executed by system users during data collection. It is toggled on by default.
- LTS Log Group:** Select the log group dedicated to intelligent O&M. Each instance is configured independently as required. When SQL Explorer is enabled, a log group is created automatically. After an instance is deleted, if the log group is no longer needed, you can manually delete it on the LTS console. For details, see [Deleting a Log Group](#).
- LTS Log Stream:** Select the log stream dedicated to SQL Explorer. Each instance is configured independently as required. When SQL Explorer is enabled, a log stream is created and configured automatically.

- **SQL Types:** Select the types of SQL statements to be collected. You can add custom types with the **Custom** option.

Step 7 Click **OK**.

Wait until the system displays a message indicating that SQL Explorer is enabled.

----End

Viewing Full SQL Statistics

Step 1 [Log in to the management console](#).

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

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

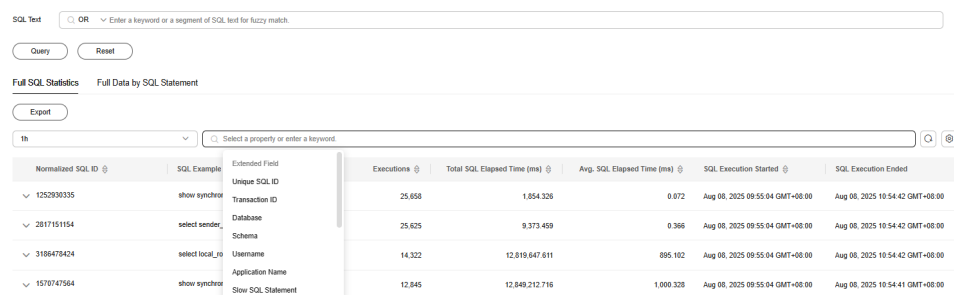
Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

Step 6 On the **Full SQL Statistics** tab of the **SQL Explorer** page, click  to query statistics of normalized SQL records. By default, data from the last hour is displayed.

You can adjust search criteria to narrow down results and precisely target specific records. For example, you can modify the time range and combine filters such as SQL text (up to 5 entries), normalized SQL ID, and other extended fields.

Figure 18-8 Adding search criteria



The screenshot shows the SQL Explorer interface. At the top, there is a search bar for 'SQL Text' with a dropdown menu and a 'Query' button. Below this, there are tabs for 'Full SQL Statistics' and 'Full Data by SQL Statement'. The 'Full SQL Statistics' tab is active, and there is an 'Export' button. Below the tabs, there is a table with columns: 'Normalized SQL ID', 'SQL Example', 'Extended Field', 'Executions', 'Total SQL Elapsed Time (ms)', 'Avg. SQL Elapsed Time (ms)', 'SQL Execution Started', and 'SQL Execution Ended'. The table contains four rows of data, each representing a different SQL statement. The first row has a 'Normalized SQL ID' of 1252830335 and a 'SQL Example' of 'show synchron'. The second row has a 'Normalized SQL ID' of 2817151154 and a 'SQL Example' of 'select sender'. The third row has a 'Normalized SQL ID' of 3186478424 and a 'SQL Example' of 'select local_ro'. The fourth row has a 'Normalized SQL ID' of 1570747564 and a 'SQL Example' of 'show synchron'. The 'Extended Field' column shows 'Unique SQL ID', 'Transaction ID', 'Database', 'Schema', 'Username', 'Application Name', and 'Slow SQL Statement' for the respective rows. The 'Executions' column shows values 25,658, 25,625, 14,322, and 12,845. The 'Total SQL Elapsed Time (ms)' column shows values 1,854,326, 9,373,459, 12,819,647,611, and 12,849,212,716. The 'Avg. SQL Elapsed Time (ms)' column shows values 0.072, 0.366, 895.102, and 1,000.328. The 'SQL Execution Started' column shows values 'Aug 08, 2025 09:55:04 GMT+08:00' for all rows. The 'SQL Execution Ended' column shows values 'Aug 08, 2025 10:54:42 GMT+08:00' for all rows.

Normalized SQL ID	SQL Example	Extended Field	Executions	Total SQL Elapsed Time (ms)	Avg. SQL Elapsed Time (ms)	SQL Execution Started	SQL Execution Ended
1252830335	show synchron	Unique SQL ID	25,658	1,854,326	0.072	Aug 08, 2025 09:55:04 GMT+08:00	Aug 08, 2025 10:54:42 GMT+08:00
2817151154	select sender	Transaction ID	25,625	9,373,459	0.366	Aug 08, 2025 09:55:04 GMT+08:00	Aug 08, 2025 10:54:42 GMT+08:00
3186478424	select local_ro	Database	14,322	12,819,647,611	895.102	Aug 08, 2025 09:55:04 GMT+08:00	Aug 08, 2025 10:54:42 GMT+08:00
1570747564	show synchron	Schema	12,845	12,849,212,716	1,000.328	Aug 08, 2025 09:55:04 GMT+08:00	Aug 08, 2025 10:54:41 GMT+08:00

Step 7 Expand a row to view the aggregated SQL execution records of the specified normalized SQL ID.

Figure 18-9 Querying all SQL execution records of a normalized SQL ID

Normalized SQL ID	SQL Example	Executions	Total SQL Elapsed Time (ms)	Avg. SQL Elapsed Time (ms)	SQL Execution Started	SQL Execution Ended
125293035	show synchronous_commit	21,862	1,582,621	0.072	Aug 07, 2025 18:31:29 GMT+08:00	Aug 07, 2025 19:23:54 GMT+08:00
281715154	select sender_pid,local_role...	21,838	7,977,152	0.365	Aug 07, 2025 18:31:29 GMT+08:00	Aug 07, 2025 19:23:53 GMT+08:00

Unique S...	SQL Text	Trans...	Database	Schema	Users...	Node ID	Comp...	Appli...	Started	Ended	To	Operation
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More
10810327955	select send...	0	postgres	"user"/public	rdAdmin	feid...	db...	cm_agent	Aug 07, 2...	Aug 07, 2...		View SQL SQL Patch More

- You can apply secondary filtering in the expanded list to narrow down and accurately locate specific records.
- A maximum of 1,000 records can be displayed on this page. If the actual number exceeds this limit, narrow your search criteria to locate target records.
- The current sorting function is based on cached sorting of the records displayed on the page (up to 1,000 records).

Step 8 Click **View SQL** in the **Operation** column to view the collected SQL text and the parsed original SQL text.

When **track_stmt_parameter** is enabled, variables in the collected SQL text are replaced with their actual values, so you can see the original SQL statement. However, for SQL text collected when **track_stmt_parameter** is disabled, the original values of the variables cannot be retrieved, and no content will be displayed here. For details about how to modify parameters, see [Modifying Instance Parameters](#).

Step 9 Click **More** and choose **View Details** in the **Operation** column to view other parameter details of full SQL records, including the details about SQL execution on components.

Figure 18-10 Querying SQL execution details

Other Parameter Details

^ dn_6003

Database (db_name)	postgres
Schema (schema_name)	"\$user",public
Original Node (origin_node)	0
Username (username)	rdsAdmin
-- (application_name)	cm_agent
Client IP Address (client_addr)	172.16.41.24
Client Port (client_port)	56624
Normalized SQL ID of outer SQL statement (parent_sql_id)	0
Normalized SQL ID (sql_id)	<HighLightTag>2817151154</HighLightTag>
Unique SQL ID (sql_exec_id)	<HighLightTag>10810327955550921657</HighLightTag>
Transaction ID (transaction_id)	0
SQL Text (query)	select sender_pid,local_role,peer_role,peer_state,state, sender_sent_location,sender_write_location,sender_flush_location,sender_replay_location,receiver_received_location,receiver_write_location,receiver_flush_location,receiver_replay_location,sync_percent,sync_state,sync_priority,sync_most_available,channel from pg_stat_get_wal_senders() where peer_role=? order by TRIM(BOTH FROM REPLACE(sync_percent, ?, ?))::numeric desc;
Thread ID (thread_id)	140203372049984
Session ID (session_id)	140203372049984
Time when the statement started (start_time)	2025-08-07 11:23:53 +0000

Step 10 Click **More** in the **Operation** and choose **Link** or **End-to-End Trace** to view link information of the SQL statement. For more link information, see [Performing Full-Link Analysis for a SQL Record](#).

----End

Viewing Full Data by SQL Statement

- Step 1 [Log in to the management console](#).
- Step 2 Click  in the upper left corner and select a region and project.

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

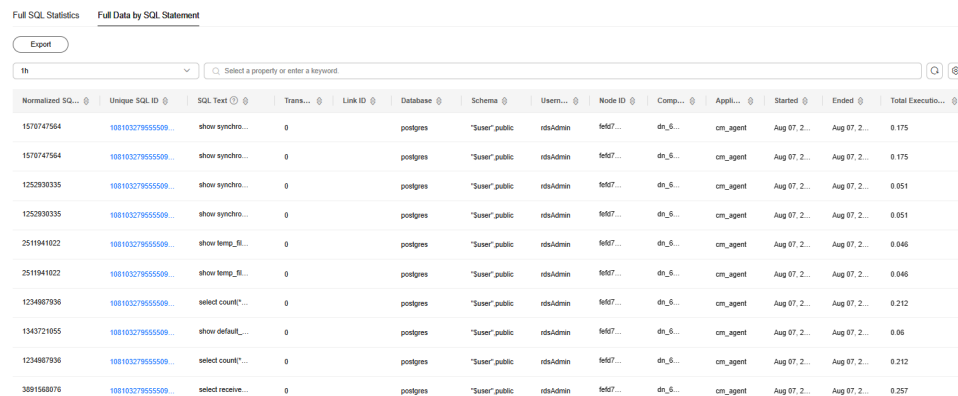
Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

Step 6 Click the **Full Data by SQL Statement** tab on the **SQL Explorer** page. On the

displayed tab, click  to query all SQL records in a specified period. By default, data from the last hour is displayed.

Figure 18-11 Querying full data by SQL statement

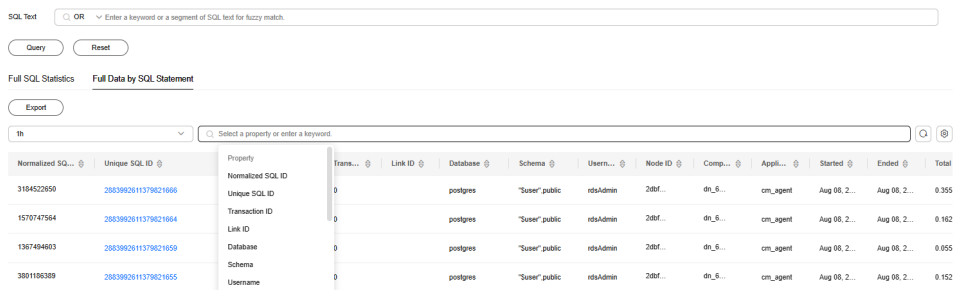


Normalized SQL...	Unique SQL ID	SQL Text	Trans...	Link ID	Database	Schema	Users...	Node ID	Comp...	Appli...	Started	Ended	Total Execution...
1570747564	108103279555509...	show synchro...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.175
1570747564	108103279555509...	show synchro...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.175
1252030335	108103279555509...	show synchro...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.051
1252030335	108103279555509...	show synchro...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.051
2511941022	108103279555509...	show temp_fi...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.046
2511941022	108103279555509...	show temp_fi...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.046
1234807936	108103279555509...	select count("...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.212
1343721055	108103279555509...	show default...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.06
1234807936	108103279555509...	select count("...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.212
3691568076	108103279555509...	select receive...	0		postgres	"Suser" public	rdsAdmin	fe67...	dn_s...	cm_agent	Aug 07, 2...	Aug 07, 2...	0.257

You can adjust search criteria to narrow down results and precisely target specific records. For example, you can modify the time range or apply combined filters such as SQL text and valid DB time.

- SQL text: You can enter up to five SQL statement snippets at a time.
- Valid DB time, CPU time, I/O time, and executor execution time: You can specify a time range to filter results.

Figure 18-12 Adding search criteria



SQL Text	OR	Enter a keyword or a segment of SQL text for fuzzy match.
Query	Reset	

Normalized SQL...	Unique SQL ID	Property	Items...	Link ID	Database	Schema	Users...	Node ID	Comp...	Appli...	Started	Ended	Total
3184522650	2083962611379621666	Normalized SQL ID	0		postgres	"Suser" public	rdsAdmin	208f...	dn_s...	cm_agent	Aug 08, 2...	Aug 08, 2...	0.355
1570747564	2083962611379621664	Transaction ID	0		postgres	"Suser" public	rdsAdmin	208f...	dn_s...	cm_agent	Aug 08, 2...	Aug 08, 2...	0.162
1367494603	2083962611379621659	Link ID	0		postgres	"Suser" public	rdsAdmin	208f...	dn_s...	cm_agent	Aug 08, 2...	Aug 08, 2...	0.055
3801196389	2083962611379621655	Database	0		postgres	"Suser" public	rdsAdmin	208f...	dn_s...	cm_agent	Aug 08, 2...	Aug 08, 2...	0.152

Step 7 Click **View SQL** in the **Operation** column to view the collected SQL text and the parsed original SQL text.

When **track_stmt_parameter** is enabled, variables in the collected SQL text are replaced with their actual values, so you can see the original SQL statement. However, for SQL text collected when **track_stmt_parameter** is disabled, the

original values of the variables cannot be retrieved, and no content will be displayed here. For details about how to modify parameters, see [Modifying Instance Parameters](#).

Step 8 Click **More** and choose **View Details** in the **Operation** column to view other parameter details of full SQL records, including the details about SQL execution on components.

Figure 18-13 Querying SQL execution details

Other Parameter Details ✕

^ dn_6003

Database (db_name)	postgres
Schema (schema_name)	"\$user",public
Original Node (origin_node)	0
Username (username)	rdsAdmin
-- (application_name)	cm_agent
Client IP Address (client_addr)	172.16.41.24
Client Port (client_port)	56624
Normalized SQL ID of outer SQL statement (parent_sql_id)	0
Normalized SQL ID (sql_id)	<HighLightTag>2817151154</HighLightTag>
Unique SQL ID (sql_exec_id)	<HighLightTag>10810327955550921657</HighLightTag>
Transaction ID (transaction_id)	0
SQL Text (query)	select sender_pid,local_role,peer_role,peer_state,state, sender_sent_location,sender_write_location,sender_flush_location,sender_replay_location,receiver_received_location,receiver_write_location,receiver_flush_location,receiver_replay_location,sync_percent,sync_state,sync_priority,sync_most_available,channel from pg_stat_get_wal_senders() where peer_role=? order by TRIM(BOTH FROM REPLACE(sync_percent, ?, ?))::numeric desc;
Thread ID (thread_id)	140203372049984
Session ID (session_id)	140203372049984
Time when the statement started (start_time)	2025-08-07 11:23:53 +0000

Step 9 Click **More** in the **Operation** and choose **Link** or **End-to-End Trace** to view link information of the SQL statement. For details, see [Performing Full-Link Analysis for a SQL Record](#).

----End

Modifying Default Settings

Step 1 Log in to the management console.

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

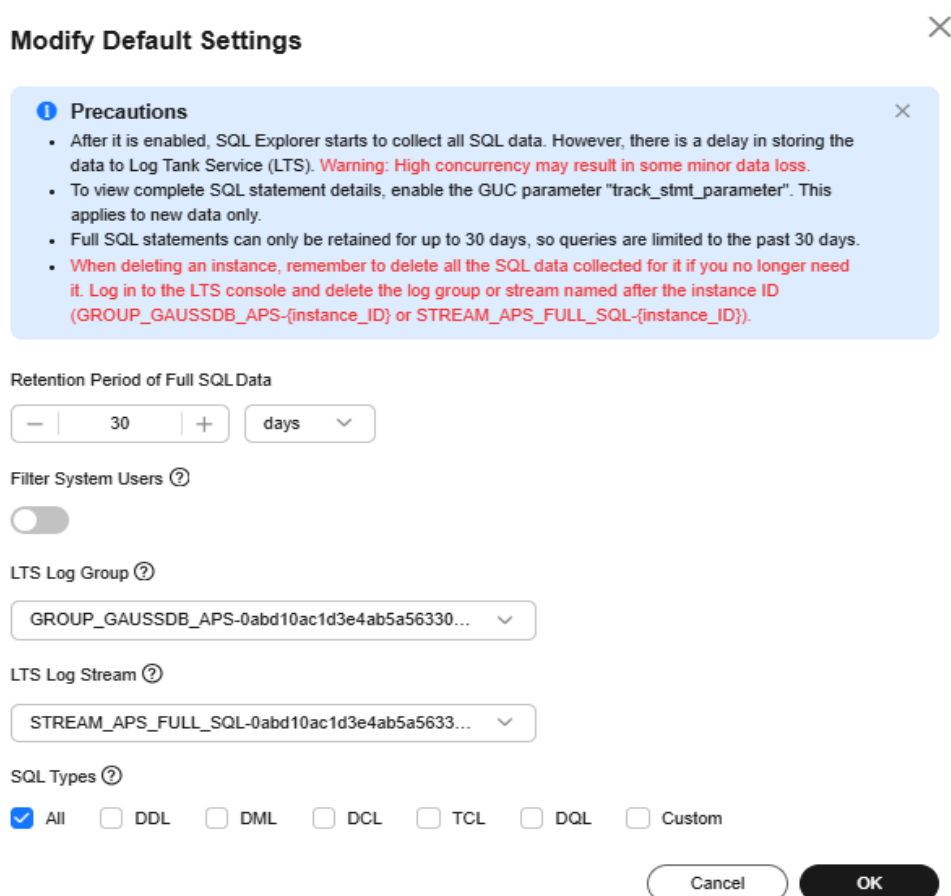
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

Step 6 On the **SQL Explorer** page, click **Modify Default Settings**. In the displayed dialog box, modify the SQL Explorer configuration parameters.

Figure 18-14 Modifying the default settings of SQL Explorer



Step 7 Click **OK**.

Step 8 Wait until the system displays a message indicating that the modification is successful.

----End

Disabling SQL Explorer

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

Step 6 On the **SQL Explorer** page, click . In the displayed dialog box, click **OK**.

Step 7 Wait until the system displays a message indicating that SQL Explorer is disabled.

----End

Creating a SQL Patch

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

Step 6 On the **Full SQL Statistics** tab of the **SQL Explorer** page, click  to expand the target row or click the **Full Data by SQL Statement** tab.

Step 7 Locate a target SQL statement and click **SQL Patch** in the **Operation** column. The **SQL Patch Details** page is displayed.

- If no SQL patch is created, enter the patch name and content, and click **OK** to create one.

Figure 18-15 SQL patch

SQL Patch Details

Tips:

- For a distributed instance, the SQL Patch is only applied for the current CN. If a new node is added, bind the SQL Patch again to apply it to the new node.

Patch Name

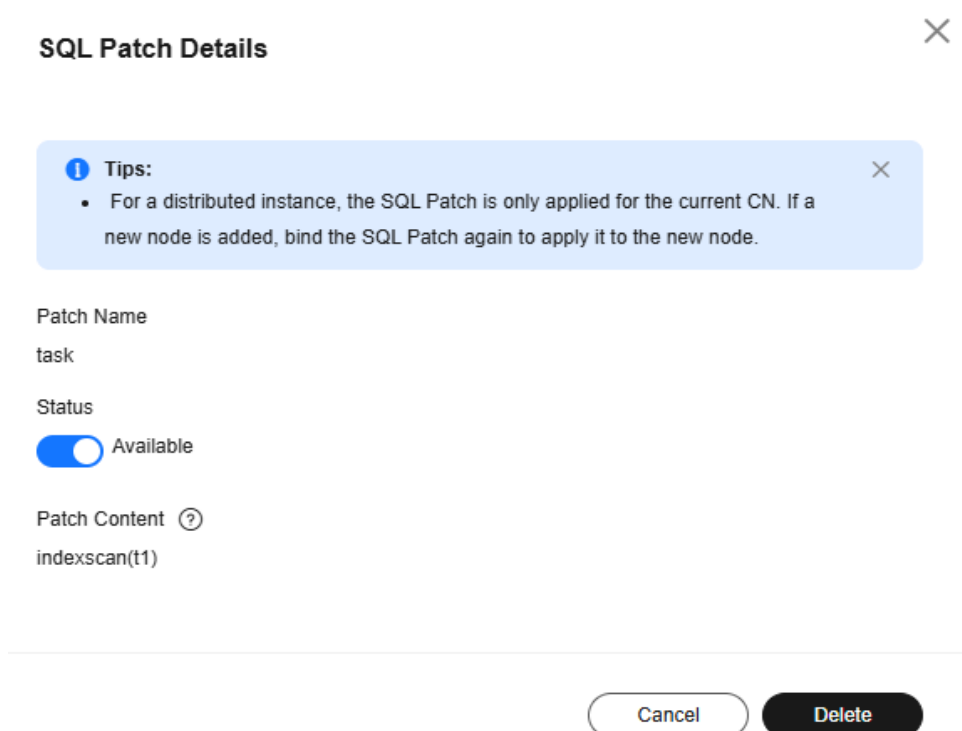
Patch Content ?

0/1,024

Cancel

OK

- If a SQL patch has been created, the SQL patch information is displayed.

Figure 18-16 Viewing SQL patch details

- **Status:** Click ☐ to enable or disable the SQL patch. If a SQL patch is disabled, the status is **Disabled**. If it is enabled, the status is **Available**.
- To delete a SQL patch, click **Delete**.

----End

Querying Table Definitions

Step 1 Log in to the management console.

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis**.

Step 6 On the **Full SQL Statistics** tab of the **SQL Explorer** page, click  to expand the target row or click the **Full Data by SQL Statement** tab.

Step 7 In the **Operation** column of a specified record, click **More** and choose **Table Definition**. On the **Table Definition** page, view information in the **Detected Tables** area.

If no table is detected, click **Add Table** and enter a table name manually. If any detection result is incorrect, click **Edit** to modify, and then save the result.

Step 8 Click  next to a table name to view its table definition.

Figure 18-17 Table Definition

Table Definition

SQL Text

Replicate

```
CREATE TABLE sales_management.customers ( customer_id SERIAL PRIMARY KEY,
customer_name VARCHAR(100) NOT NULL, email VARCHAR(100) UNIQUE, phone
VARCHAR(20), address TEXT, created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP);
```

Detected Tables

If no table is detected, click "Add Table" and enter a table name manually. If any detection result is incorrect, click "Edit" to modify, and then save the result. Only regular user tables are supported. Definitions of views and system catalogs are unavailable.

Add Table

Table Name	Operation
 customers	Edit Delete

Schema	Table Definition
sales_management	SET search_path = sales_management; CREATE TAB...

Close

Step 9 Click the SQL statement in the table definition to view the definition details.

Figure 18-18 Table definition details



----End

18.4 Performing Full-Link Analysis for a SQL Record

Scenarios

You can query details about SQL execution links using the unique SQL ID or transaction ID of an application SQL statement.

Prerequisites

- You have logged in using an account assigned the **gaussdb:instance:listSqlLink** action.
- SQL Explorer has been enabled, and valid data has been collected.

Constraints

- If the transaction ID is 0, it cannot be used to query link details.
- If the unique SQL ID is 0, no link information is available.
- The DB engine version of GaussDB instances must be V2.0-8.0.1 or later, and the kernel version must be 505.0.0 or later. For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > SQL Diagnosis** and click the **SQL Explorer** tab.

Step 6 On the **Full SQL Statistics** tab, click  to expand the target row. Alternatively, click the **Full Data by SQL Statement** tab.

Step 7 Click the transaction ID or click **More** in the **Operation** column and choose **End-to-End Trace** to go to the SQL full-link analysis page.

- If you click **More** in the **Operation** column and choose **End-to-End Trace**, the execution link details of the SQL statement are displayed.
- If you click a transaction ID (which is not 0), the execution link details of the transaction are displayed.

Figure 18-19 SQL link information

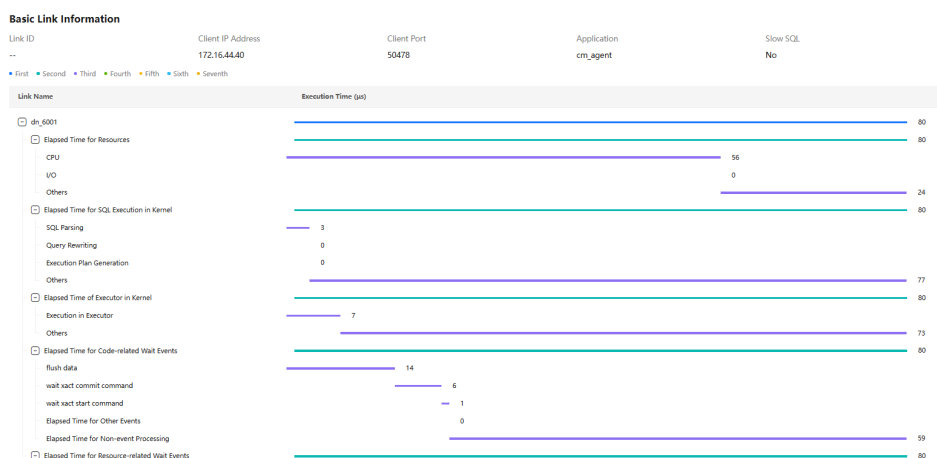


Table 18-4 Parameters for SQL link analysis

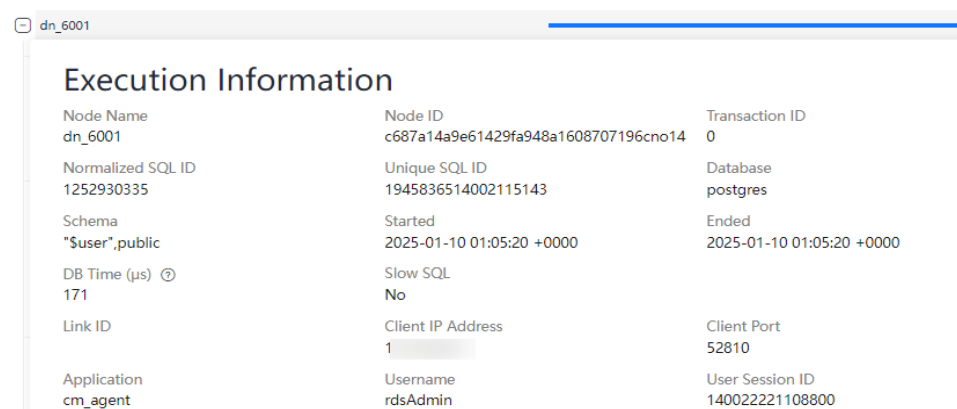
Module	Parameter	Description
Elapsed Time for Resources	CPU	Time spent on CPU processing
	I/O	Time spent on I/O operations
	Others	Valid DB time minus the CPU and I/O time
Elapsed Time for Kernel Execution	SQL Parsing	Time spent on parsing the SQL statement
	Query Rewriting	Time spent on rewriting the SQL statement
	Execution Plan Generation	Time spent on generating a SQL execution plan
	Execution in Executor	Time spent on executing the SQL statement
	Others	Total SQL execution time minus the time spent on SQL parsing, SQL rewriting, SQL plan generation, and execution in the executor
Elapsed Time for Code-related Wait Events	Top 5 Events by Elapsed Time	Top events with the longest elapsed time
	Elapsed Time for Other Events	Time consumed by all events minus the time consumed by top 5 events
	Others	Valid DB time minus the time consumed by all wait event code
Elapsed Time for Resource-related Wait Events - Elapsed Time for I/O	Top 5 Events by Elapsed Time	The most time-consuming I/O events
	Elapsed Time for Other Events	Time consumed by all I/O events minus the time consumed by top 5 I/O events
	Others	Valid DB time minus the time consumed by all I/O wait events
Elapsed Time for Resource-related Wait Events - Elapsed Time for Locks	Top 5 Events by Elapsed Time	The most time-consuming lock events
	Elapsed Time for Other Events	Time consumed by all lock events minus the time consumed by top 5 lock events
	Others	Valid DB time minus the time consumed by all lock events

Module	Parameter	Description
Elapsed Time for Resource-related Wait Events - Elapsed Time for Lightweight Locks	Top 5 Events by Elapsed Time	The most time-consuming lightweight lock events
	Elapsed Time for Other Events	Time consumed by all lightweight lock events minus the time consumed by top 5 lightweight lock events
	Others	Valid DB time minus the time consumed by all lightweight lock events

For details about wait event parameters, see the description of **wait_status** and **wait_event** in "Developer Guide" > "System Catalogs and System Views" > "System Views" > "Other System Views" > "PG_THREAD_WAIT_STATUS."

Step 8 Hover the pointer over a node to view the details about SQL execution on that node.

Figure 18-20 SQL execution information



Execution Information		
Node Name dn_6001	Node ID c687a14a9e61429fa948a1608707196cno14	Transaction ID 0
Normalized SQL ID 1252930335	Unique SQL ID 1945836514002115143	Database postgres
Schema "\$user",public	Started 2025-01-10 01:05:20 +0000	Ended 2025-01-10 01:05:20 +0000
DB Time (μs) ⓘ 171	Slow SQL No	
Link ID	Client IP Address 1	Client Port 52810
Application cm_agent	Username rdsAdmin	User Session ID 14002221108800

DB Time indicates the total DB time, in microseconds, which is accumulated if multiple threads are involved. The way for collecting DB time statistics is slightly different from that for collecting start/end time statistics, but the difference does not affect performance issue analysis.

----End

18.5 Configuring SQL Throttling Rules

Scenarios

GaussDB supports SQL throttling. When there are abnormal SQL statements (for example, poor indexing) or the number of concurrent SQL requests increases, you can use this function to control the number of access requests or concurrency of specific SQL statements to ensure service availability.

Prerequisites

You have logged in using an account assigned the following actions:

gaussdb:instance:listFlowlimit, gaussdb:instance:flowlimitAddOrUpdate, gaussdb:instance:createFlowLimit, gaussdb:instance:updateFlowLimit, gaussdb:instance:deleteFlowLimit, gaussdb:instance:getFlowLimitInfo, gaussdb:instance:getFlowLimitTemplate, gaussdb:instance:listFlowLimitInfo, gaussdb:instance:syncFlowLimit, and gaussdb:instance:flowlimitDelete

Constraints

- The DB engine version of GaussDB instances must be V2.0-8.0.1 or later, and the kernel version must be 505.0.0 or later.
- Auto throttling is unavailable to distributed instances.
- SQL throttling cannot be performed for system-level users (such as **root**).
- Currently, SQL throttling tasks cannot be deleted in batches.
- Each instance supports a maximum of 1,000 throttling tasks.
- For a keyword-based throttling task, keywords are separated by commas (,), and the sequence of keywords is not distinguished.
- If you used CLI commands to add, delete, or modify a SQL throttling task as a system user (for example, user **root**), the actual throttling task status will be different from that displayed on the web UI.
- When a throttling rule is being applied, CPU usage increases. To ensure system stability, do not create too many rules for a given instance.
- SQL statements that have been executed before a throttling rule is added are not counted.
- In DR scenarios, the throttling rules applied to the primary instance are automatically synchronized to the DR instance. After the DR instance is promoted to primary, the throttling rules cannot be added, deleted, modified, or queried. Throttling tasks can be managed again only after the original primary instance becomes primary.
- If data is lost when you restore a backup to a new DB instance or switch the original primary instance to primary, handle the data loss issue by following the instructions in *Troubleshooting*.
- You can still create, modify, and query a throttling rule if the instance status is abnormal or when instance specifications are being changed, but the operation may fail. If a failure or exception occurs, you can delete the throttling rule. Before deleting a throttling rule, ensure that the database is running properly.
- Throttling is not allowed for SQL statements involved in setting up database connections, for example, **select setting from pg_settings where name= 'delimiter_name'**, because new connections cannot be set up if such statements are blocked.
- When an instance node is being created or rebooted (for example, for adding replicas or shards), the node may fail to be connected. As a result, the SQL throttling task details cannot be queried.
- You may receive a kernel error message like **ERROR: The workload rule takes effect and this request will be cancelled. rule_id: x, rule_name: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"**. The rule name in such a message indicates the throttling task ID on this page and the task ID in audit logs.

- Do not create tables with the same name. If a table with the same name exists in different databases or schemas, the throttling rule may not take effect when you create a SQL ID-based throttling task related to the table.
- The auto throttling rules take effect only when the CPU usage, memory usage, and maximum concurrency (the maximum number of active sessions) all exceed respective thresholds.

Creating a Throttling Task

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and OptimizationSQL DiagnosisSQL Throttling**.

Step 6 On the **SQL Throttling Tasks** or **Auto Throttling Tasks** tab, click **Create Task**. In the right pane, set the throttling parameters described in [Table 18-5](#).

- Creating a SQL throttling task

Figure 18-21 Creating a SQL throttling task

Create SQL Throttling Task

⚠ Connections of system users like root cannot be throttled. Keyword-based throttling is unavailable to system databases.

Throttling Type

SQL ID

Keyword

Task name

Enter a keyword.

?

Throttled Databases

—Select—

▼

SQL Template

—Select—

▼

Max. Concurrency

—

0

+

Time Range

Start Date — End Date

📅

Cancel

OK

Table 18-5 Parameters for creating a SQL throttling task

Throttling Type	Parameter	Description
SQL ID	Task Name	Name of the SQL throttling task.
	SQL Template	The system records the SQL statement execution records of each node and converts the records into a template. You can select the template for performing SQL throttling on each node. NOTE When an instance is rebooted, the SQL template of the instance node will be cleared.

Throttling Type	Parameter	Description
	Max. Concurrency	Maximum number of active connections for executing the throttled SQL statement on the specified node at the same time.
	Time Range	Time period when the throttling takes effect.
Keyword NOTE – 2 to 100 keywords can be specified for a single task. Each keyword can contain 2 to 64 characters. – Keywords are case-insensitive. – The fewer the keywords, the larger the range of SQL statements to be throttled.	Task Name	Name of the SQL throttling task.
	SQL Statement Type	SQL statement type, which can be SELECT, INSERT, UPDATE, DELETE, or MERGE.
	Keyword	Keywords used for throttling. The throttling task throttles the statements that match the specified keywords and SQL statement type.
	Target Databases	Databases where the throttling takes effect.
	Max. Concurrency	Maximum number of active connections for executing the throttled SQL statement on the specified node at the same time.
	Time Range	Time period when the throttling takes effect.

- Creating an auto throttling task

Figure 18-22 Create Auto Throttling Task

Create Auto Throttling Task

Scenario

Scenario

Triggered by thresholds

Task name

Task name

Enter a keyword.

Triggers

Throttling takes effect if all specified threshold conditions are met.

CPU Usage

--

020406080100

Memory Usage

--

020406080100

Throttling Information

Max. Concurrency

0

Cancel

OK

Table 18-6 Parameters for creating an auto throttling task

Parameter	Description
Scenario	Condition for triggering throttling. Currently, only Triggered by thresholds is available.
Task Name	Name of the throttling task.

Parameter	Description
CPU Usage	CPU usage threshold for triggering throttling on the specified node. Throttling takes effect if both the CPU and memory usage threshold conditions are met. The CPU usage and memory usage thresholds cannot both be set to 0. If you only need one of them for throttling, set the other threshold to 0.
Memory Usage	Memory usage threshold for triggering throttling on the specified node. Throttling takes effect if both the CPU and memory usage threshold conditions are met. The CPU usage and memory usage thresholds cannot both be set to 0. If you only need one of them for throttling, set the other threshold to 0.
Max. Concurrency	Maximum number of active connections for executing SQL statements on the specified node at the same time.

- Step 7** Click **OK**.
- Step 8** After a throttling task is created, view throttling details or modify the task.
- To view task details, click **Details** in the **Operation** column of the task.

Figure 18-23 Viewing SQL throttling task details

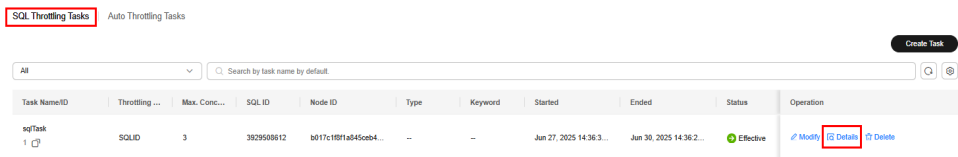
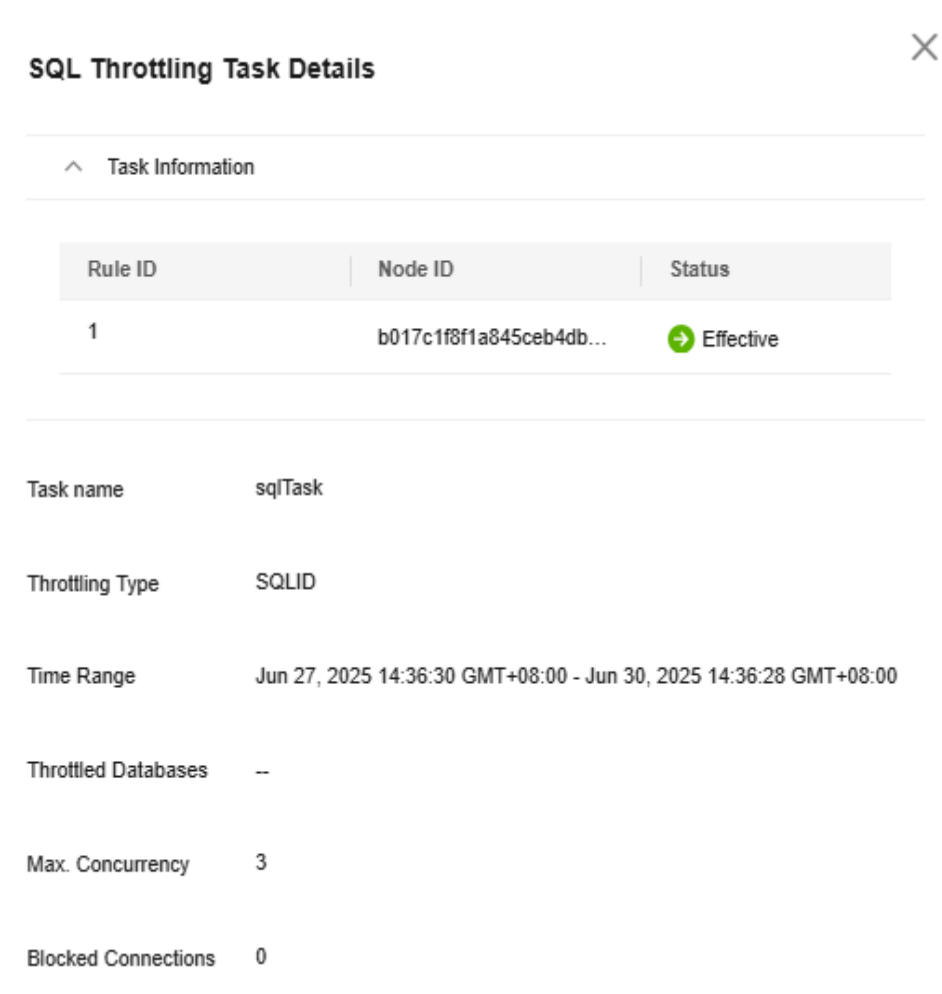
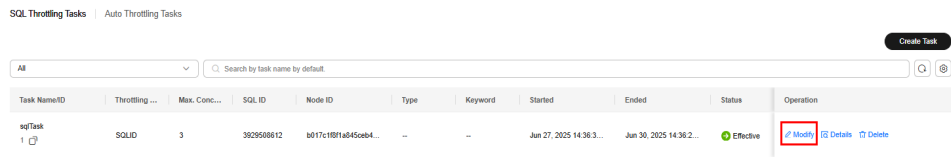


Figure 18-24 SQL throttling task details



- To modify a throttling task, click **Modify** in the **Operation** column of the task.

Figure 18-25 Modifying a throttling task



- To delete a throttling task, click **Delete** in the **Operation** column of the task.

NOTE

After the throttling rule of a task is modified, the instance reboots, or a primary/standby switchover is performed, the number of blocked connections of the task will be cleared.

----End

18.6 Data Storage Insights

Scenarios

GaussDB lets you monitor key storage metrics for your instance components, including used data disk space, disk usage rate, Xlog size, and detailed storage information for database tables.

Constraints

The DB engine version of GaussDB instances must be V2.0-3.208.0 or later.

Precautions

- Only one database can be selected at a time in the **All Databases** area to view table statistics.
- Statistics on tables of system users are not collected.
- Only schemas that contain at least one table are included in the collection.
- When you change the deployment model of an instance or add shards, data migration and redistribution will occur. Use this feature after those operations are fully completed.
- The value provided in the **Available days of storage** field is an estimate. Actual available days may vary due to workload changes.

Viewing the Instance Storage Overview

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > Data Storage Insights**.

Step 6 In the **Disk Usage** area, review the data disk usage of the instance.

Table 18-7 Metric description

Metric	Description
Used/Total	This metric displays the amount of data disk space currently used and the total available capacity for the nodes of the monitored object.
Average daily growth	This value indicates the average daily increase in stored data over the past 7 days.

Metric	Description
Available days of storage	The value provided here is an estimate. Actual available days may vary due to workload changes. Available days of storage = (Total instance storage – Used capacity)/Average daily growth

Step 7 Select the instance components, nodes, and time range to view the metric data of the corresponding components and nodes.

Figure 18-26 Trends



Table 18-8 Metric description

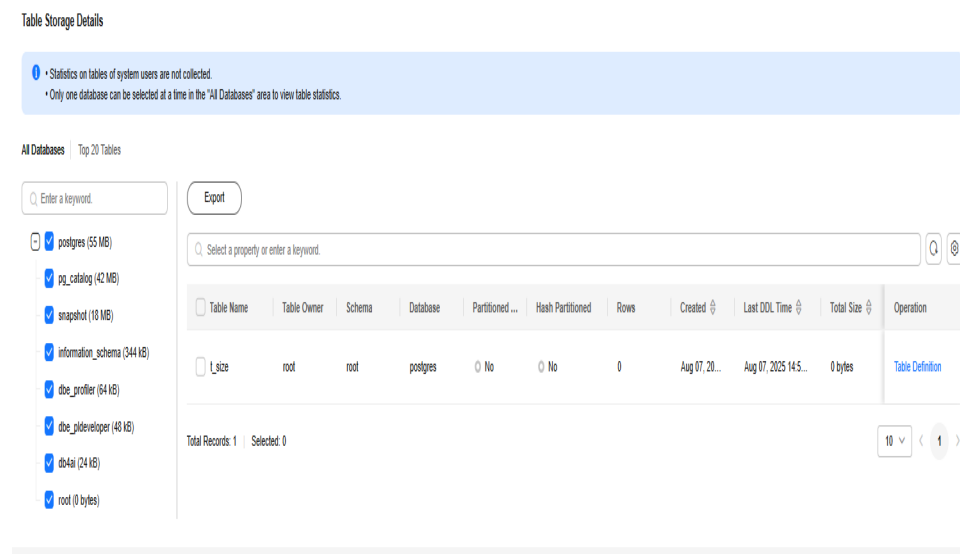
Metric	Description
Used Data Disk Space	Real-time used data disk size of the selected nodes
Data Disk Usage	Real-time data disk usage of the selected nodes
Xlog Size	Real-time size of Xlogs in the data directory on a CN or DN

----End

Viewing Database Table Data

- Step 1 [Log in to the management console.](#)
- Step 2 Click  in the upper left corner and select a region and project.
- Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB.**

- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** Choose **Diagnostics and Optimization > Data Storage Insights**.
- Step 6** Scroll down to the **Table Storage Details** area. On the **All Databases** tab, click  next to the target database to expand its schemas. Then, select one or more schemas to view corresponding table statistics on the right.

Figure 18-27 Database table information

You can also click **Top 20 Tables** to view statistics of the 20 largest tables among all databases.

Table 18-9 Field description

Field	Description
Table Name	Table name.
Table Owner	Owner of the table.
Schema	Name of the schema the table belongs to.
Database	Name of the database the table belongs to.
Partitioned Table	Whether the table or index has the property of a partitioned table.
Hash Partitioned	Whether the table contains a hash-partitioned column.
Rows	Number of rows in the table.
Created	Creation time.
Last DDL Time	Last time when a DDL operation is performed.

Field	Description
Total Size	Table size.
Avg. Size per DN	The ideal evenly-distributed size of a table on each DN. Avg. Size per DN = Total table size/Number of DNs This field is only available for distributed instances.
Max DN Size	Ratio of the maximum table size on a single DN to the total table size. Max DN Size = Maximum size occupied by a table on a single DN/Total table size This field is only available for distributed instances.
Min DN Size	Ratio of the minimum table size on a single DN to the total table size. Min DN Size = Minimum size occupied by a table on a single DN/Total table size This field is only available for distributed instances.
Total Data Skew	Table skew. Total Data Skew = Maximum size occupied by a table on a single DN – Minimum size occupied by a table on a single DN This field is only available for distributed instances.
Data Skew Rate	Table skew rate. Data Skew Rate = Total data skew/Total table size This field is only available for distributed instances.
Data Distribution Standard Deviation	Standard deviation of table distribution. (For two tables of the same size, a larger deviation indicates a more severe skew.) This field is only available for distributed instances.

Step 7 Click **Table Definition** in the **Operation** column to view the table definition.

Figure 18-28 Table definition



----End

18.7 Performance Reports

18.7.1 Generating a Workload Diagnosis Report

Scenarios

You can collect and download workload diagnosis reports (WDRs) on the GaussDB console.

Constraints

The DB engine version of GaussDB instances must be V2.0-3.100 or later, and the kernel version must be 503.0 or later. For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).

Precautions

The two snapshots used to generate a report should meet the following conditions:

- No node is rebooted between the two snapshots.
- No primary/standby switchover is performed between the two snapshots.
- No DROP operation is performed on the database between the two snapshots.
- The DB instance is not read-only.
- The DB instance is not a DR instance.

Operation tips:

- To generate a snapshot immediately without waiting for the specified generation interval, click **Create Snapshot**.
- By default, snapshots are generated every 60 minutes. Set the collection period based on the specified generation interval. If the specified period is shorter than the interval, the collection fails.
- If the number of generated snapshots exceeds the maximum number allowed within the retention period (24 x 8 = 192 by default), each time a new snapshot is generated, the system deletes the snapshot with the lowest snapshot ID.
- Snapshot collection may fail if the data is collected when a large number of DDL statements are executed.
- If a DB instance contains a large number of databases or tables, it takes a long time to collect snapshot data.
- The kernel collects data from DN nodes by shard. If you select one DN in a shard in a WDR report, data of all DN nodes in that shard will be displayed. Assume that **dn_6001**, **dn_6002**, and **dn_6003** belong to the same shard. If **dn_6001** is selected for data collection, **dn_6001_6002_6003** is displayed in the report.

Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Diagnostics and Optimization > Performance Reports**.

Step 6 Specify the snapshot collection period as required and click **Create Snapshot**.

This operation will trigger a full WDR snapshot query to retrieve all available completed snapshots within the specified period. Any time ranges that include kernel reset events (node restarts, primary/standby switchovers, RESET operations on performance metrics, or DROP DATABASE operations) will be excluded. The system then displays all available snapshot ranges.

Step 7 Specify **Collection Type**.

- In the **Collection Type** field, select a type as required.
Instance level and **Component level** are available. You can change selected components if **Component level** is selected.

Figure 18-29 WDR task configuration

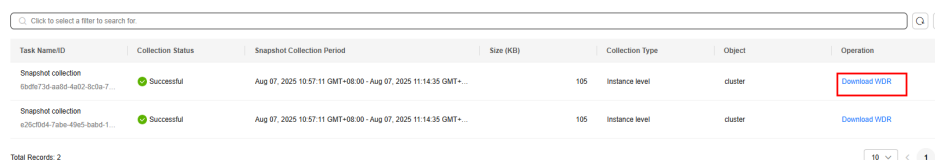


Step 8 Click **Collect** to submit the snapshot collection and WDR report creation task.

Step 9 View the WDR task list and wait until the task is created.

- If the collection status is **Successful**, click **Download WDR** to obtain the performance report of the specified period.

Figure 18-30 WDR task list



Task Name/ID	Collection Status	Snapshot Collection Period	Size (KB)	Collection Type	Object	Operation
Snapshot collection 6b09e736-ea09-4a02-8c3a-7...	Successful	Aug 07, 2025 10:57:11 GMT+08:00 - Aug 07, 2025 11:14:35 GMT+...	105	Instance level	cluster	Download WDR
Snapshot collection e25c5d44-7abe-49e5-bab6-1...	Successful	Aug 07, 2025 10:57:11 GMT+08:00 - Aug 07, 2025 11:14:35 GMT+...	105	Instance level	cluster	Download WDR

- If the collection status is **Failed**, contact Huawei Cloud customer service.

----End

18.7.2 Generating an Active Session Profile

Scenarios

You can collect and download active session profiles (ASPs) on the GaussDB console.

Constraints

The DB engine version of GaussDB instances must be V2.0-3.300 or later, and the kernel version must be 503.2 or later. For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).

Precautions

- After ASP Generation is enabled, ASP data is sampled every second.
- If ASP Generation is disabled, there may be no data available when the data source is set to template-collected data for SQL diagnosis and slow SQL data collection.
- The default collection period is the latest 2 hours and the maximum length is two days.
- The collected ASP reports are retained for three days by default.

Procedure

Step 1 [Log in to the management console](#).

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

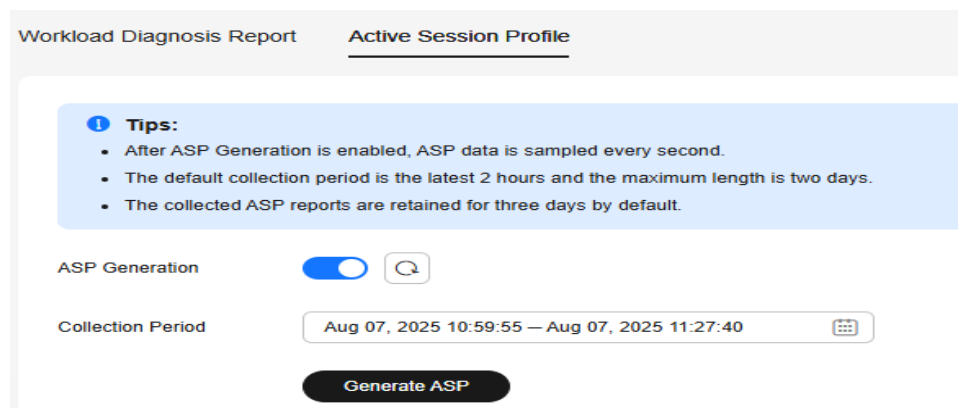
Step 3 Click  in the upper left corner of the page and choose **Databases > GaussDB**.

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Diagnostics and Optimization > Performance Reports**. Then, click the **Active Session Profile** tab.

Step 6 Select the collection period and click **Generate ASP**.

Figure 18-31 Generating an ASP



Step 7 Refresh the instance later and view the collection result.

- If the collection is successful, click **Download ASP** to obtain the ASP of the specified period.
- If the collection failed, contact Huawei Cloud customer service.

----End

19 Session Management

19.1 Managing Real-Time Sessions

Scenarios

GaussDB allows you to view all real-time component-level sessions of an instance and kill specified sessions. You can also identify table names in SQL text using table definition query or query the table definition based on the table name, schema name, and database name.

Constraints

- Table definition query is available for instances whose DB engine version is V2.0-3.208.0 or later and kernel version is 503.1.0.SPC1200 or later. This function is only available for authorized users. To apply for the permissions needed, [submit a service ticket](#).
- For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).
- The system limits queries to return up to 200 real-time session records, ensuring requests do not time out from excessive data.
- If the normalized SQL ID is 0 or the user is a system user, SQL throttling tasks cannot be created.
- For distributed instances, memory information is only available when querying sessions of a single CN or DN. This is unavailable when querying sessions by all CNs, all session statuses, or CN drill-down sessions.

Viewing Real-Time Sessions

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Diagnostics and Optimization > Session Management** to view real-time session information. In the search box, select the database, client IP address, username, or other filters to search for session data.

Table 19-1 Metrics for real-time sessions

Metric	Description
Normalized SQL ID	If the normalized SQL ID is 0, SQL patches and SQL throttling tasks cannot be created.
Waiting	If the session is currently waiting for a lock, the value is true . Otherwise, the value is false .
Blocked Session ID	ID of the session that is blocking the current session from obtaining a lock.
Wait Event	Event that the current thread is waiting for.
Status	Overall state of the session. Enumerated values: • active: The backend is executing a query. • idle: The backend is waiting for a new client command. • idle in transaction: The backend is in a transaction, but there is no statement being executed in the transaction. • fastpath function call: The backend is executing a fast-path function. • disabled: This state is displayed if track_activities is disabled in the backend.

- Select one or more sessions and click **Kill Selected Sessions**. In the displayed dialog box, click **OK**.
You can also click **Kill Session** in the **Operation** column of a specified session to kill it.
- Click **Kill All Idle Sessions**. In the displayed dialog box, click **OK**.
- You can click **Configure Auto Killing** to enable or disable automatic transaction killing and configure automatic killing rules for long transactions and large transactions.

Figure 19-1 Long transactions**Configure Auto Killing**

⚠ Before performing an operation that is expected to run for a long time, for example, data migration, disable auto killing or increase the threshold first. After the operation is complete, restore the settings as needed.

⚠ The system checks for and kills long or large transactions every 5 minutes. If a transaction meets the specified criteria, for example, its Xlog size exceeds 10 MB, but the auto killing task is waiting for its next run, the transaction will only be killed during the next check, not immediately.

Type

Long transactions

Large transactions

Long transactions severely affect system performance and may even cause the entire database to be unavailable.

Enable Auto Killing



Disabled

Execution Time
Threshold (optional)

1

day

0

hours

0

minutes

Target Databases
(optional)

--Select--



If no value is selected, this rule applies to all databases by default.

Target Users (optional)

Enter a keyword.

If no value is entered, this rule applies to all users by default.

High-Risk Operation

To confirm the configuration, enter "YES" below. [Auto Enter](#)

YES

Cancel

OK

Figure 19-2 Large Transactions

Configure Auto Killing

- ⚠** Before performing an operation that is expected to run for a long time, for example, data migration, disable auto killing or increase the threshold first. After the operation is complete, restore the settings as needed.
- ⚠** The system checks for and kills long or large transactions every 5 minutes. If a transaction meets the specified criteria, for example, its Xlog size exceeds 10 MB, but the auto killing task is waiting for its next run, the transaction will only be killed during the next check, not immediately.

Type: Long transactions | **Large transactions**

Large transactions drastically increase the log file size, which can quickly exhaust all available storage.

Enable Auto Killing: ☐ Disabled

Xlog Size Threshold (optional): 1 GB 0 MB 0 KB 0 Bytes

Target Databases (optional): --Select--
If no value is selected, this rule applies to all databases by default.

Target Users (optional): Enter a keyword.
If no value is entered, this rule applies to all users by default.

High-Risk Operation

To confirm the configuration, enter "YES" below. [Auto Enter](#)

YES

Cancel

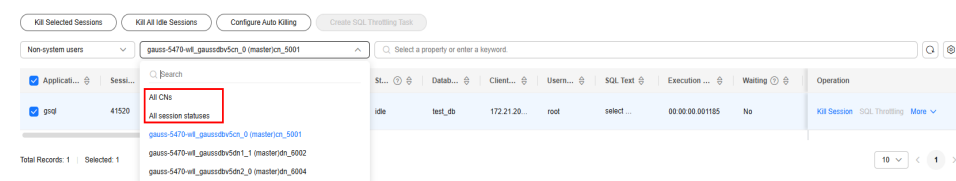
OK

- Select the sessions for which you want to control their concurrency, and click **Create SQL Throttling Task** to limit the concurrency of the normalized SQL IDs involved in the selected sessions.

You can also click **SQL Throttling** in the **Operation** column of a specified session to limit the concurrency of the specified normalized SQL ID. For details about how to create a SQL throttling task, see [Configuring SQL Throttling Rules](#).

- For a distributed instance, you can select **All CNs** or **All session statuses** from the drop-down list of the node filter. If you select **All CNs**, the sessions of all CNs are displayed. If you select **All session statuses**, the session thread statuses of all CNs and DN nodes are displayed.

Figure 19-3 All CNs/All session statuses



 NOTE

- An idle session is a session in the **idle** state. Sessions in the **idle in transactions** or **active** state are not regarded as idle sessions.
- Only sessions whose client IP addresses are not empty are displayed in the session list.
- Normalized SQL IDs cannot be queried for instances of V2.0-1.4 or earlier, so the normalized SQL ID column for such instances is empty.
- Real-time session statistics are only available for nodes with CNs or DN (primary or standby DN) deployed.

----End

Creating a SQL Patch

Step 1 [Log in to the management console](#).

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Diagnostics and Optimization > Session Management** to view real-time session information.

**CAUTION**

If the normalized SQL ID is 0, SQL patches cannot be created.

Step 6 Locate a target SQL statement, click **More**, and click **SQL Patch** in the **Operation** column to go to the **SQL Patch Details** page.

- If no SQL patch is created, enter the patch name, and click **OK**.

Figure 19-4 SQL patch

SQL Patch Details

Tips:

- For a distributed instance, the SQL Patch is only applied for the current CN. If a new node is added, bind the SQL Patch again to apply it to the new node.

Patch Name

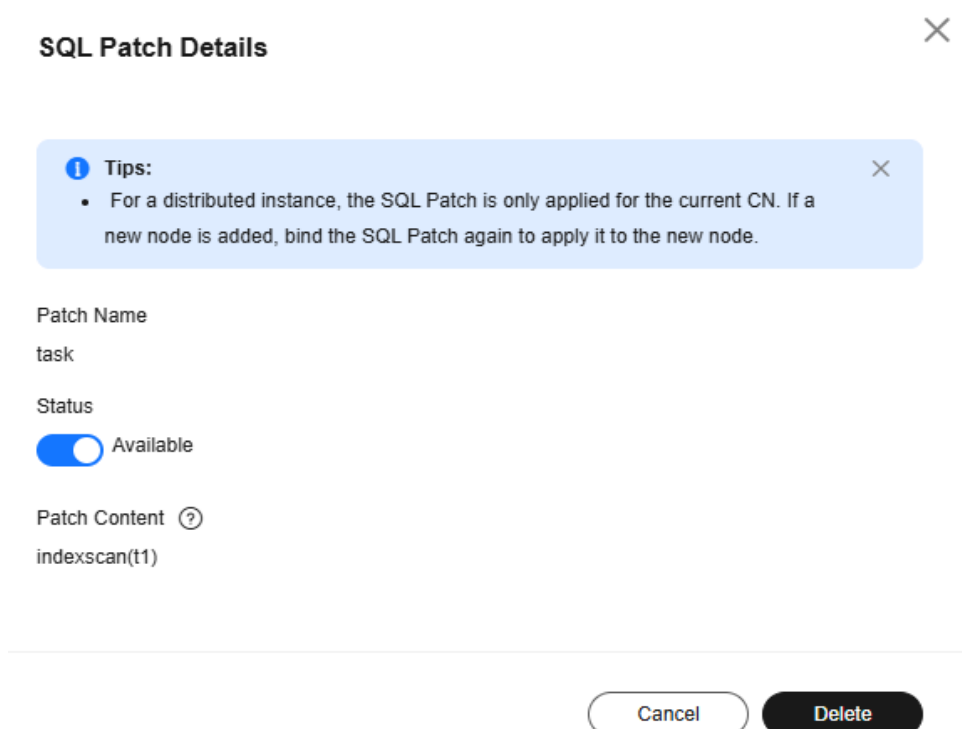
Patch Content ?

0/1,024

Cancel

OK

- If a SQL patch has been created, the SQL patch information is displayed.

Figure 19-5 Viewing SQL patch details

- **Status:** Click  to enable or disable the SQL patch. If a SQL patch is disabled, the status is **Disabled**. If it is enabled, the status is **Available**.
- To delete a SQL patch, click **Delete**.

----End

Querying Table Definitions

Step 1 Log in to the management console.

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 Choose **Diagnostics and Optimization > Session Management**.

Step 6 Click **Table Definition** in the **Operation** column. On the **Table Definition** page, view information in the **Detected Tables** area.

If no table is detected, click **Add Table** and enter a table name manually. If any detection result is incorrect, click **Edit** to modify, and then save the result.

Step 7 Click  next to a table name to view its table definition.

Figure 19-6 Table Definition

Table Definition

SQL Text

Replicate

```
CREATE TABLE sales_management.customers ( customer_id SERIAL PRIMARY KEY,
customer_name VARCHAR(100) NOT NULL, email VARCHAR(100) UNIQUE, phone
VARCHAR(20), address TEXT, created_at TIMESTAMP DEFAULT
CURRENT_TIMESTAMP);
```

Detected Tables

If no table is detected, click "Add Table" and enter a table name manually. If any detection result is incorrect, click "Edit" to modify, and then save the result. Only regular user tables are supported. Definitions of views and system catalogs are unavailable.

Add Table

Table Name	Operation
^ customers	<a>Edit <a>Delete

Schema	Table Definition
sales_management	<a>SET search_path = sales_management; CREATE TAB...

Close

Step 8 Click the SQL statement in the table definition to view the definition details.

Figure 19-7 Table definition details



----End

19.2 Viewing Session Statistics

Scenarios

On the GaussDB console, you can monitor both active and total session counts categorized by user, access source, and database.

Constraints

The DB engine version of GaussDB instances must be V2.0-3.100 or later, and the kernel version must be 503.0 or later. For details about how to query the version, see [Checking the DB Engine and Kernel Engine Version of a GaussDB Instance](#).

Procedure

- Step 1** [Log in to the management console.](#)
- Step 2** Click  in the upper left corner and select a region and project.
- Step 3** Click  in the upper left corner of the page and choose **Databases > GaussDB**.
- Step 4** On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.
- Step 5** In the navigation pane, choose **Diagnostics and Optimization > Session Management**. Then, click the **Session Statistics** tab.

Table 19-2 Metrics for real-time session statistics

Metric	Description
User	Database login username
Source	Origin of the session connection to the database
Database	Name of the database being accessed
Active Sessions	Number of active sessions
Total Sessions	Total number of sessions

----End

20 Disaster Recovery Management

20.1 Constraints

GaussDB provides the cross-region disaster recovery (DR) capability to improve your data security. If a disaster occurs in the primary data center, the DR data center takes over services quickly. The cross-region DR is suitable for data centers with a distance greater than 200 km.

Constraints

Constraints on DR relationship establishment:

- Currently, a primary instance can only establish a DR relationship with one DR instance. If you want to re-establish a new DR relationship, you must stop the DR relationship between the primary and original DR instances and then select an available instance to be the DR instance.
- To establish a DR relationship for distributed instances, the number of shards of the primary instance must be the same as that of the DR instance.
- The instances for which a DR relationship is established must use the same OS. DR relationships cannot be established between instances running on different OSs, for example, they run on EulerOS and HCEOS separately.

Constraints during DR relationship establishment:

- The DR instance is only used to synchronize data of the primary instance. It cannot read or write data.
- During DR relationship creation, storage of the primary and DR instances cannot be scaled up.
- For the primary instance, if the instance is forcibly started when a majority of AZs are faulty, data cannot be synchronized to the DR instance. In this case, you need to disconnect the DR relationship.
- If a minority of nodes for the DR instance are faulty, the DR instance still works properly.
- If a majority of nodes for the DR instance are faulty, the DR instance cannot be promoted to primary.

- When you restore the primary instance to a new instance, the DR user account cannot be automatically deleted. You need to delete the account manually.
- The DR instance in a DR relationship does not support adding replicas. After the DR instance is promoted to primary, replicas in the instance can be added.
- The resource types (virtualization types) of the primary and DR instances must be the same.
- It is recommended that the specifications of the DR instance be the same as those of the primary instance. Smaller specifications of the DR instance will slow down Xlog replay on the DR instance.
- Before setting up cross-cloud DR, you need to delegate the DR operation permissions to a system user and connect the two instances properly.
 - No action is required if the two instances are in the same VPC.
 - If the two instances are deployed in different VPCs, create a VPC peering connection on the **Virtual Private Cloud > VPC Peering Connections** page of the VPC Endpoint console and configure routes between subnets.
 - In cross-cloud scenarios, submit a request to Direct Connect for creating a direct connection between the VPCs in the two regions. Specifically, [submit a service ticket](#).
 - Ensure that the security group allows IPv4 inbound traffic from port 12016.
- Compatibility constraints:
 - In the cross-cloud DR scenario, the peer DR instance name is not displayed in the DR relationship list.
- If the pre-verification for a cross-cloud DR operation fails, you can deliver the same operation again on the page after the fault is rectified.

20.2 Creating a DR Relationship

This section describes how to create a disaster recovery (DR) relationship.

Precautions

- Before setting up cross-cloud DR, you need to delegate the DR operation permissions to a system user and connect the two instances properly.
 - No action is required if the two instances are in the same VPC.
 - If the two instances are deployed in different VPCs, create a VPC peering connection on the **Virtual Private Cloud > VPC Peering Connections** page of the VPC Endpoint console and configure routes between subnets.
 - In cross-cloud scenarios, submit a request to Direct Connect for creating a direct connection between the VPCs in the two regions. Specifically, [submit a service ticket](#).
 - Ensure that the security group allows IPv4 inbound traffic from port 12016.
- Before establishing the DR relationship, log in to the console and select the cloud where the primary or DR instance is deployed separately, go to the **Basic Information** page of the instance from the **Instances** page and click **Reset Configuration** for the **DR IP Address** field in the **Configuration** area.

Additionally, configure the CIDR block of the peer instance for both instances. (You can obtain the CIDR block information of an instance from the **Subnet** field of the **Network Information** area on its **Basic Information** page.)

- A DR relationship cannot be established between instances with different CPU architectures.
- The replica consistency protocol and DB engine kernel version of the primary instance must match those of the DR instance.
- After a DR task is created, all databases on the selected DR instance will be cleared. The backup policy will be disabled for the selected DR instance, but historical backup data will be retained. If you want to retain data of the selected DR instance, create a manual backup before the DR task is created.
- After a DR relationship is created, DBS will be automatically authorized to access VPC resource information and query IaaS APIs.
- After a DR relationship is created, the DR network settings and other configurations of the instance will be reset.
- The following table lists the supported DR modes.

Table 20-1 Dual-instance streaming DR

Deployment	Primary Instance	DR Instance
Distributed	Combined (Basic edition)	Combined (Basic edition)
Centralized	1 primary + 2 standby	1 primary + 2 standby or Single
	1 primary + 1 standby + 1 log (Paxos)	1 primary + 1 standby + 1 log (Paxos) or Single (Paxos)

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 In the navigation pane on the left, choose **Disaster Recovery**. On the displayed page, click **Create DR Task**.

Step 5 Configure required parameters described in [Table 20-2](#) and select the confirmation information.

Table 20-2 Parameter description

Parameter	Description
DR Type	Type of the disaster recovery. Streaming DR is supported.
Primary Instance	Primary instance in the DR relationship. - Only available instances that run V2.0-2.0 or later versions and have more than two replicas are displayed. - When Replica Consistency Protocol is set to Paxos, only 1 primary + 1 standby + 1 log instances can create DR relationships.
DR Instance IP Address	To obtain the IP address of the DR instance, log in to the cloud where the DR instance is deployed, go to the Basic Information page of the instance from the Instances page and view the value of DR IP Address in the Configuration area. Before establishing the DR relationship, log in to the console and select the cloud where the primary or DR instance is deployed separately, go to the Basic Information page of the instance from the Instances page and click Reset Configuration for the DR IP Address field in the Configuration area. Additionally, configure the CIDR block of the peer instance for both instances. (You can obtain the CIDR block information of an instance from the Subnet field of the Network Information area on its Basic Information page.)
DR Instance DB User	User account for DR, which is used to authenticate the primary and DR instances during DR. The default account name is user root of the DR instance.
DR Instance User Password	Administrator password of the DR instance.

Step 6 Click **OK** to create a DR relationship. You can view the task status on the **Disaster Recovery** page.

----End

20.3 Checking DR Task Statuses

After the DR relationship is established between the primary and DR instances, you can view the data synchronization status on the DR task details page.

Procedure

Step 1 [Log in to the management console.](#)

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

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

Step 4 Choose **Disaster Recovery** in the navigation pane on the left.

Step 5 Click the name of the target DR task to go to the task details page.

View the statuses of the primary and DR instances.

- Statuses of the primary instance
 - **Full synchronizing**: The primary instance is fully synchronizing data to the DR instance after the DR relationship is established.
 - **Full synchronization failed**: The primary instance fails to fully synchronize data to the DR instance after the DR relationship is established.
 - **Demoting to standby**: The roles of the primary and DR instances are being switched. In this case, the status of the primary instance is **Demoting to standby**.
 - **Incremental synchronization in progress**: In the DR relationship, the primary instance synchronizes incremental data to the DR instance.
 - **Incremental synchronization failed**: In the DR relationship, the primary instance fails to synchronize incremental data to the DR instance.
 - **RPO** indicates the time, in seconds, from when a transaction (generally the latest transaction) was submitted to when the transaction was sent to the DR instance.
Description of special RPO values:
 - If the RPO is 0, all transactions have been sent to the DR instance.
 - If the RPO is -1, there are transactions that were not synchronized to the DR instance database or the DR instance is stopped.
- Statuses of the DR instance
 - **Full restoration**: The DR instance is in the full restoration status after DR relationship is established.
 - **Full restoration failed**: The DR instance fails to be fully restored.
 - **Incremental restoration**: The DR instance is synchronizing incremental data from the primary instance.
 - **Incremental restoration failed**: The DR instance fails to synchronize incremental data from the primary instance.
 - **Promoting to primary**: The roles of the primary and DR instances are being switched, or the system is failing over to the DR instance in case the primary instance becomes unavailable. In this case, the status of the DR instance is **Promoting to primary**.
 - **Promotion failed**: The DR instance fails to be promoted to primary.
 - **RTO** measures the amount of data being transmitted.
Description of special RTO values:
 - If the RTO is 0, all transactions have been transmitted to and successfully executed on the DR instance.

- If the RTO is -1, there are transactions that were not restored to the DR instance or the DR instance is stopped.

----End

20.4 Promoting the DR Instance to Primary

If the primary instance is unavailable, you can manually promote the DR instance to primary.

Precautions

- If the DR instance is available and in the incremental restoration status, you can promote the DR instance to the primary instance.
- If the DR instance is promoted to the primary instance, it can process read and write requests.
- A DR instance can be promoted to primary regardless of whether the primary instance is available. In the streaming DR scenario, if the primary instance is available, you can stop the DR task on the primary instance after the DR instance is promoted to primary.
- If a DR instance node is faulty before the promotion, repair or replace the faulty node after the promotion is complete. Specifically, [submit a service ticket](#).
- In the streaming DR scenario, you can determine whether to enable **Re-establish DR relationship** when promoting the DR instance to primary. For details, see [Re-creating a DR Task After the Primary Instance Is Faulty](#).
- After a DR task is complete, the whitelist configuration set in the **Reset Configuration** dialog box will be automatically cleared.
- After the promotion is complete, the DR task cannot be retried. You need to create a new DR task. After the DR instance is promoted to primary, you need to manually stop the DR task on the original primary instance to clear the DR data.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 In the navigation pane on the left, choose **Disaster Recovery**. Locate a DR task, and click **Promote DR Instance to Primary** in the **Operation** column.

Step 5 Enable **Re-establish DR relationship**.

- If you enable **Re-establish DR relationship**, the system saves the DR relationship record. After the DR instance is promoted to primary, you can re-establish the DR relationship by one click.
- If you disable this option, the DR relationship will be disconnected and cannot be re-established by one click.

Step 6 In the displayed dialog box, select **Promote DR instance to primary** and click **OK**.

The DR instance becomes the primary instance.

----End

20.5 Stopping a DR Task

This section describes how to stop the DR relationship between the primary and DR instances.

Precautions

- After the DR task is stopped, you need to create a new DR task if needed.
- After the DR task is complete, the network configuration will be automatically cleared. If you need to set up a new DR relationship, go to the instance details page to reset the configuration.
- In the streaming DR scenario, if instance B is promoted to the primary instance and removes the DR relationship with instance A, instance A does not know that the relationship has been deleted and can successfully remove the DR relationship with instance B only when instance B is available. If instance B is abnormal, you need to manually skip the removal step related to instance B when deleting the relationship on the instance A side.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 In the navigation pane on the left, choose **Disaster Recovery**. Locate a DR task and click **Stop** in the **Operation** column.

Step 5 In the displayed dialog box, select **Stop DR Relationship** and click **OK**.

----End

20.6 Deleting a DR Task

After a DR task is stopped or fails to be created, you can delete the DR task record on the console.

Procedure

Step 1 [Log in to the management console](#).

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

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

Step 4 In the navigation pane on the left, choose **Disaster Recovery**. Locate a stopped DR task and click **Delete** in the **Operation** column.

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

----End

20.7 Re-creating a DR Task After the Primary Instance Is Faulty

Scenarios

A DR relationship has been established with instance A as the primary instance and instance B as the DR instance. If primary instance A is faulty, you need to manually promote instance B to primary.

After instance A is recovered, re-establish the DR relationship between instance B and instance A, and then perform a DR switchover to promote instance A to the primary instance.

Procedure

Step 1 If the primary instance A is faulty, log in to the console, select the region where the DR instance B is deployed, promote instance B to primary by referring to [Promoting the DR Instance to Primary](#), and enable **Re-establish DR Relationship**.

In this case, instance B takes over the role of instance A to provide services.

Step 2 Log in to the console, select the region where instance B is located, and re-establish a DR relationship by referring to [Re-establishing a DR Relationship](#).

Step 3 After the fault of instance A is rectified, switch roles of instance A and instance B by referring to [Switching Roles of Primary and DR Instances](#).

In this case, the services are taken over by instance A and the DR relationship is restored to the status before the fault occurs.

After the switchover, instance A provides services. You need to switch services to instance A.

----End

20.8 Switching Roles of Primary and DR Instances

Scenarios

If the primary and DR regions and primary and DR instances are normal, you can log in to either the primary instance or DR instance to switch their roles. This section describes how to switch roles of primary and DR instances.

Procedure

- Step 1** Log in to the console and select the region where the primary or DR instance is located.
- Step 2** In the navigation pane on the left, choose **Disaster Recovery**. Locate a DR task, and click **Switch Roles** in the **Operation** column.
- Step 3** In the dialog box that is displayed, confirm related information.
- Step 4** Select **Switch roles** and click **OK**.

----End

20.9 Re-establishing a DR Relationship

Scenarios

After the DR instance is promoted to primary and the original primary instance restores, you can re-establish the DR relationship between the two instances. If the DR relationship is re-established at the DR instance side, the original primary instance becomes the DR instance, and the original DR instance becomes primary.

Precautions

- Streaming DR is available only for instances whose DB engine version is V2.0-3.200 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).

Procedure

- Step 1** Log in to the console and select the region where the new primary instance is located.
- Step 2** In the navigation pane on the left, choose **Disaster Recovery**. Locate the DR task and click **Re-establish DR Relationship** in the **Operation** column.
- Step 3** In the displayed dialog box, select the confirmation information, and click **OK**.

If the DR relationship is re-established at the DR instance side, the original primary instance becomes the DR instance, and the original DR instance becomes primary.

----End

20.10 Performing a DR Drill

Scenarios

If the primary and DR regions and primary and DR instances are normal, you can log in to the DR instance and perform a DR drill. This section describes the precautions and how to perform a DR drill.

Precautions

- This function is supported only when streaming DR is used and the DB engine version is V2.0-3.200 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- After a DR drill starts, data synchronization between the primary and DR instances is suspended. Automated backups for the DR instance are created.
- When a DR drill is being performed on the DR instance, do not remove the DR relationship from the primary instance side.
- DR drills are supported for dual-instance streaming DR only.

Procedure

Step 1 Log in to the console and select the region where the DR instance is located.

Step 2 In the navigation pane on the left, choose **Disaster Recovery**. Click **More** in the **Operation** column of the target DR task, and choose **Enable DR Drill**.

Log Cache Space/Primary Instance Space: During the DR drill, the DR instance provides services and the incremental data is temporarily cached in the primary instance. After the drill is complete, the DR instance can perform incremental replay based on the temporarily stored incremental data. If the cached incremental data exceeds the value of this parameter, the DR instance will replay all data after the drill is complete.

Step 3 After the DR drill is enabled, perform DR drill operations on the DR instance.

Step 4 After the drill is complete, log in to the console and select the region where the DR instance is located. On the **Disaster Recovery** page, click **More** in the **Operation** column of the target DR task, and choose **Disable DR Drill**.

Step 5 After the DR drill is complete, view that the DR task is restored to the status before the DR drill is performed. The data generated by the primary instance during the DR drill is synchronized to the DR instance, and the data generated by the DR instance during the DR drill is cleared.

----End

20.11 Enabling or Disabling Log Cache

Scenarios

If the primary and standby regions are disconnected for a long time due to network problems, you can enable log cache to temporarily store incremental data in the primary instance storage space. When the network recovers, the incremental data, instead of full data, is synchronized to the DR instance.

Precautions

- This function is supported only when streaming DR is used and the DB engine version is V2.0-3.200 or later. For details about how to view the instance version, see [Checking the DB Engine Version](#).
- During the log caching period in the cross-cloud DR scenario, the DR relationship cannot be removed.

Procedure

- Step 1** Log in to the console and select the region where the DR instance is located.
- Step 2** In the navigation pane on the left, choose **Disaster Recovery**. Click **More** in the **Operation** column of the target DR task, and choose **Enable Log Cache**.
- Step 3** After the log cache is enabled, the task status of the primary instance changes to **Caching logs**. The incremental data temporarily stores in the primary instance storage space.
- Step 4** After the DR relationship is normal, log in to the console, select the region where the primary instance is located, and click **Disable Log Cache** in the **Operation** column.
- Step 5** After log cache is disabled, check that the DR task is restored to the status before the log cache is enabled. The data generated during the log cache on the primary instance is synchronized to the DR instance.

----End

20.12 Restrictions on the DR Instance

After a DB instance serves as a DR instance, some basic functions are unavailable and some monitoring metrics cannot be reported. This section describes the restrictions on the DR instance.

Restrictions on the Primary Instance

After a DR relationship is established, the DR instance only allows for storage scaling, specifications change, rebooting, and parameter modification. The basic functions of the primary instance have the following restrictions:

Function	Supported by the Primary Instance During DR
Rebooting an instance	Yes
Modifying parameters	Yes
Applying parameters	Yes
Resetting a password	Yes
Performing a full backup	Yes
Creating a differential backup	Yes
Deleting a backup	Yes
Modifying the backup policy	Yes
Restoring data to the original instance	No
Restoring data to a new instance	Yes
Scaling up storage space	Yes

Function	Supported by the Primary Instance During DR
Scaling down storage space	Yes
Adding nodes (only for distributed instances)	No
Changing instance specifications (only for centralized instances)	Yes
Upgrading a minor version	Yes
Viewing slow query logs	Yes
Viewing metrics	Yes
Deleting an instance	No
Deleting an automated backup	N/A
Rebuilding a deleted instance from the recycle bin	N/A
Backing up and restoring data in a single table	Yes
Querying the disk usage	Yes
Creating a database	Yes
Querying a database	Yes
Creating a schema and user	Yes
Deleting a schema and user	Yes
Performing database operations	Yes
Using DAS for database management	Yes
Repairing a node	Yes
Replacing a node	Yes
Forcibly starting a minority of AZs	No
Resuming AZs	No
Switching AZs	Yes
Resetting configurations (for DR)	Yes
Adding replicas	No
Enabling transparent data encryption	No
Managing tags	Yes
Binding an EIP	Yes

Restrictions on Monitoring Metrics

After the DR relationship is established, some monitoring metrics of the DR instance are unavailable. The following table lists the monitoring metrics of the DR instance.



CAUTION

If the deployment model of an instance is **1 primary + 1 standby + 1 log**, the log node does not have a data disk, and the following node-level metrics related to data disks cannot be displayed for the node:

- Data Disk IOPS
- Data Disk Write Throughput
- Data Disk Read Throughput
- Time Required for per Data Disk Write
- Time Required for per Data Disk Read

Category	Sub-category	Metric	Displayed for Distributed DR Instance	Displayed for Centralized DR Instance
Instance level	-	Used Instance Disk Size	Yes	No
		Total Instance Disk Size	Yes	No
		Instance Disk Usage	Yes	No
		Deadlocks	No	No
		Response Time of 80% SQL Statements	No	No
		Response Time of 95% SQL Statements	No	No
		Buffer Hit Rate	No	No
Node level	-	CPU Usage	Yes	Yes
		Memory Usage	Yes	Yes
		Data Write Volume	Yes	Yes
		Outgoing Data Volume	Yes	Yes
		Disk IOPS	Yes	Yes
		Disk Write Throughput	Yes	Yes

Category	Sub - category	Metric	Displayed for Distributed DR Instance	Displayed for Centralized DR Instance
		Disk Read Throughput	Yes	Yes
		Time Required for per Data Disk Write	Yes	Yes
		Time Required for per Data Disk Read	Yes	Yes
		Swap Memory Usage	Yes	Yes
		Total Swap Memory	Yes	Yes
		IOPS Usage (This metric applies only to nodes that use EVS disks.)	Yes	Yes
		Disk I/O Bandwidth Usage (This metric applies only to nodes that use EVS disks.)	Yes	Yes

Category	Sub-category	Metric	Displayed for Distributed DR Instance	Displayed for Centralized DR Instance
Component level	CN	User Logins, User Logouts, Lock Waiting Session Rate, Active Session Rate, CN Connections, User Committed Transactions, User Rollback Transactions, Background Committed Transactions, Background Rollback Transactions, Average Response Time of User Transactions, User Transaction Rollback Rate, Background Transaction Rollback Rate, Data Definition Language, Data Manipulation Language, Data Control Language, DDL and DCL Rate, Data Volume to Be Flushed to Disks, Physical Reads per Second, Physical Writes per Second, Online Sessions, Active Sessions, Online Session Rate, Used Dynamic Memory, Dynamic Memory Usage, Maximum Execution Duration of Database Transactions, Idle Transactions, Slow SQL Statements in the System Database, Slow SQL Statements in the User Database, and Xlog Rate	No	N/A
	Primary DN	Used Disk Size, Total Disk Size, Disk Usage, Data Volume to Be Flushed to Disks, Physical Reads per Second, and Physical Writes per Second	N/A	N/A

Category	Sub-category	Metric	Displayed for Distributed DR Instance	Displayed for Centralized DR Instance
		Standby Redo Progress, User Logins, User Logouts, Lock Waiting Session Rate, Active Session Rate, User Committed Transactions, User Rollback Transactions, Background Committed Transactions, Background Rollback Transactions, Average Response Time of User Transactions, User Transaction Rollback Rate, Background Transaction Rollback Rate, Data Definition Language, Data Manipulation Language, Data Control Language, DDL and DCL Rate, Online Sessions, Active Sessions, Online Session Rate, Used Dynamic Memory, Dynamic Memory Usage, Size of WALs Retained in the Replication Slot, and Xlog Rate	N/A	N/A
	Standby DN	Used Disk Size	Yes	Yes
		Total Disk Size	Yes	Yes
		Disk Usage	Yes	Yes
		Primary Node Flow Control Duration	No	No
		Standby Node RTO	No	No
		standby_xlog_replay_speed	No	No
		Difference Between Redo and Receipt Positions on Standby Node	No	No

Category	Sub-category	Metric	Displayed for Distributed DR Instance	Displayed for Centralized DR Instance
		Standby Node Redo Progress	No	N/A
		Lock-Waiting Session Rate	No	No
		Active Session Rate	No	No
		Online Session Rate	No	No
		Online Sessions	No	No
		Dynamic Memory Usage	No	No

Restrictions on Major Version Upgrade in the DR Mode

Only gray upgrade with the rolling upgrade method is supported.

- Step 1** Initiate the upgrade process for the primary instance. Do not commit the upgrade.
- Step 2** Initiate the upgrade process for the DR instance.
- Step 3** After the DR instance upgrade is complete, commit the upgrade of the DR instance first and then the primary instance.

-----End

Restrictions on Minor Version Upgrade in the DR Mode

CAUTION

- Before committing the upgrade of the DR instance, ensure that the upgrade process of the primary instance is complete.
 - Commit the upgrade of the DR instance first and then the primary instance.
 - If the DR instance upgrade has been committed, the upgrade of the primary instance cannot be rolled back.
 - During the upgrade of the primary and DR instances, do not perform a switchover.
-

Only gray upgrade with the rolling upgrade method is supported.

- Step 1** Initiate the upgrade process for the primary and DR instances at the same time.

Step 2 After the upgrade is complete, commit the upgrade of the DR instance first and then the primary instance.

----End

21 Managing GaussDB Tasks

You can view the progresses and results of tasks on the **Task Center** page. You can also delete tasks that are no longer required.

Precautions

Deleted task records cannot be recovered. Exercise caution when performing this operation.

Tasks That Can Be Viewed and Managed

Table 21-1 Supported task types

Category	Task
Instance creation	Creating a GaussDB instance
Instance deletion	Deleting a GaussDB instance
Instance modifications	• Adding shards • Adding coordinator nodes • Scaling up storage • Changing instance specifications • Changing the deployment model of an instance
Instance upgrade	• Rolling upgrade • Upgrade commit • Upgrade auto-commit • In-place upgrade • Upgrade rollback • Hot patch update • Hot patch rollback

Category	Task
Backup and restoration	• Creating a manual backup • Restoring data to a new instance • Restoring data to an existing instance • Restoring data to the original instance • Stopping a backup

Viewing a Task

Step 1 [Log in to the management console.](#)

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

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

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed page, view the task details.

- To identify a task, you can use the task name/ID or instance name/ID, or simply select a task name in the search box displayed in the upper pane of the page.
- You can view the progress and status of tasks in a specific period. The default period is seven days.
The task list can only show up to 30 days of past tasks.
- You can view tasks in the following statuses:
 - Running
 - Completed
 - Failed
- You can view the task creation and completion time.

----End

Deleting a Task Record

You can delete the task records that no longer need to be displayed. The deletion only deletes the task records, and does not delete the DB instances or terminate the tasks that are being executed.

Step 1 [Log in to the management console.](#)

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

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

Step 4 Choose **Task Center** in the navigation pane on the left. On the displayed page, locate the task record to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.

You can delete tasks in the following statuses:

- Completed
- Failed

----End

Why Does the Task Progress Percentage Remain Unchanged?

Task progress is not linear. It is dynamically calculated based on factors like task type, data volume, workload, and instance resources. If progress appears stuck for a while, it usually means time-consuming operations are in progress.

Why Can't I Find My Instance Operation Task in the Task Center?

1. **Unsupported task type:** Only tasks listed in [Table 21-1](#) can be displayed in the task center. If your task is not listed, it will not be displayed.
2. **Submission delay:** If your task is supported, it may have just been submitted. Wait for a while and refresh the page.

Can I Cancel an Ongoing Task in the Task Center?

No, tasks that are currently in progress cannot be canceled.

22 Managing GaussDB Tags

Scenarios

Tag Management Service (TMS) enables you to use tags on the management console to manage resources. TMS works with other cloud services to manage tags. TMS manages tags globally. Other cloud services manage only their own tags.

You can use unified tags to manage various resources and use custom tags to define different DB instance categories. This section describes how to edit and delete tags.

Precautions

- You are advised to configure predefined tags on the TMS console.
- A tag consists of a key and value. You can add only one value for each key.
- A maximum of 20 tags can be added for an instance.

Editing Tags

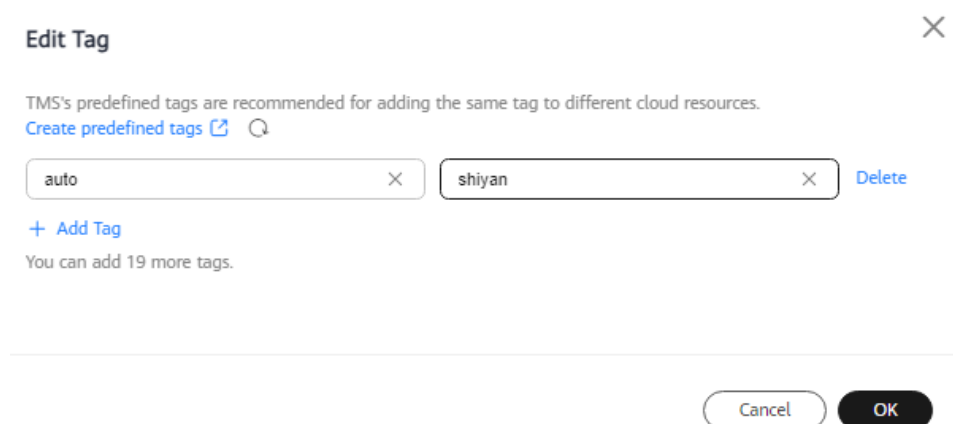
Step 1 [Log in to the management console.](#)

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

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane, choose **Tags**. On the displayed page, click **Edit Tag**. In the displayed dialog box, click **Add Tag**, enter a tag key and value, and click **OK**.

Figure 22-1 Editing a tag

- When you enter a tag key and value, the system automatically displays all tags (including predefined tags and resource tags) associated with all DB instances except the current one.
- A tag key can contain up to 128 characters. It cannot start with **_sys_** or a space, and cannot end with a space. Only letters, digits, spaces, and the following special characters are allowed: **_:/=+-@**
- A tag value can contain up to 255 characters. Only letters, digits, spaces, and the following special characters are allowed: **_:/=+-@**

Step 6 View and manage the tag on the **Tags** page.

----End

Deleting a Tag

Step 1 [Log in to the management console.](#)

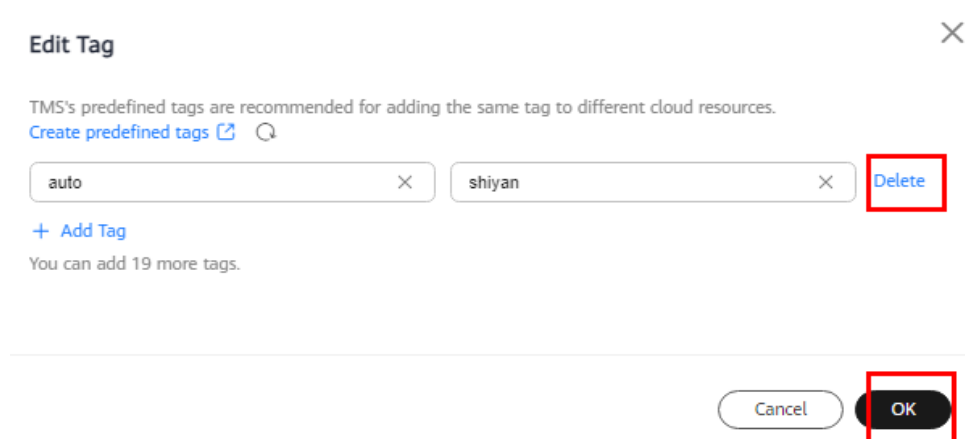
Step 2 Click  in the upper left corner and select a region and project.

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

Step 4 On the **Instances** page, click the name of the target instance to go to the **Basic Information** page.

Step 5 In the navigation pane on the left, choose **Tags**. On the displayed page, click **Edit Tag**, locate the tag to be deleted, and click **Delete**. Then, click **OK**.

Figure 22-2 Deleting a tag



Step 6 Check that the tag is no longer displayed on the **Tags** page.

-----End