Data Warehouse Service

User Guide

Issue	01
Date	2024-12-27

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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.

Contents

1 Using GaussDB(DWS)

GaussDB(DWS) is an online data processing database that uses the Huawei Cloud infrastructure to provide scalable, fully-managed, and out-of-the-box analytic database service, freeing you from complex database management and monitoring. It is a native cloud service based on the Huawei converged data warehouse GaussDB, and is fully compatible with the standard ANSI SQL 99 and SQL 2003, as well as the PostgreSQL and Oracle ecosystems. GaussDB(DWS) provides competitive solutions for PB-level big data analysis in various industries.

GaussDB(DWS) provides an easy-to-use management console, allowing you to quickly create clusters and easily manage data warehouses.

Procedure Description

Figure 1-1 Procedure for using GaussDB(DWS)

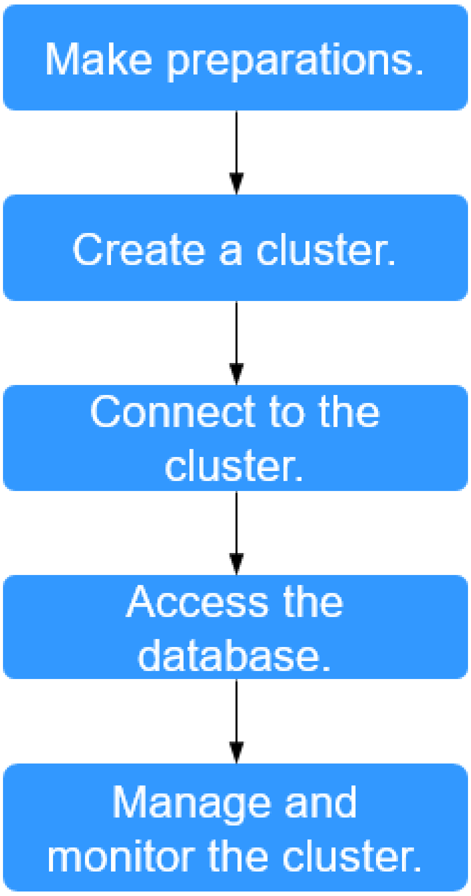


Table 1-1 Procedure description

Process	Task	Description	Operation Instruction
Making Preparations	-	Before using GaussDB(DWS), you need to apply for a Huawei Cloud account.	**Preparations**
Create a cluster.	-	Create a cluster before using GaussDB(DWS) to execute data analysis tasks. A GaussDB(DWS) cluster contains nodes in the same subnet. These nodes jointly provide services. During cluster creation, the system creates a default database.	• **Creating a GaussDB(DWS) Storage-Compute Coupled Cluster** • **Creating a GaussDB(DWS) Storage-Compute Decoupled Cluster** • **Creating a Yearly/ Monthly Cluster**

Process	Task	Description	Operation Instruction
Connect to the cluster.	-	After the GaussDB(DWS) cluster is created, use the SQL client tool or a third-party driver such as JDBC or ODBC to connect to the database in the cluster. You can download the SQL client tool and JDBC/ODBC driver on the **Client Connections** page of the GaussDB(DWS) console.	**Connecting to a GaussDB(DWS) Cluster**
Access the database.	-	After connecting to the cluster, you can create and manage databases, manage users and permissions, import and export data, and query and analyze data.	***Data Warehouse Service (DWS) Developer Guide***
Manage and monitor the cluster.	Cluster management	View the cluster status, modify cluster configurations, add cluster tags, and scale out, restart, and delete the cluster.	**GaussDB(DWS) Cluster Management**
	Snapshot management	Create snapshots to back up and restore the cluster.	**Backing Up and Restoring a GaussDB(DWS) Cluster**
	O&M and monitoring	View the running status and performance of the cluster through monitoring, log auditing, event notification, and resource load management.	• **Viewing GaussDB(DWS) Cluster Monitoring Information on Cloud Eye** • **Event Notifications Overview** • **GaussDB(DWS) Cluster Log Management** • **GaussDB(DWS) Resource Load Management**

Process	Task	Description	Operation Instruction
	Scaling and specification change	● Expand the capacity of an existing cluster on the console if your service requires additional compute or storage resources. ● Change the specifications of created clusters on the console.	● **Scaling Out a Cluster** ● **Changing GaussDB(DWS) Cluster Specifications**
	Cluster upgrade	Cluster 8.1.1 and later versions allow users to deliver cluster upgrade operations on the console.	**Upgrading a GaussDB(DWS) Cluster**
	Resource load management	GaussDB(DWS) provides the resource management function. You can put resources (CPU, memory, I/O, and storage space) into different resource pools, which are isolated from each other.	**GaussDB(DWS) Resource Load Management**

2 Preparations

2.1 Creating a User and Granting GaussDB(DWS) Permissions

Before using GaussDB(DWS), register a Huawei Cloud account. If you need to manage account permissions more precisely, use Identity and Access Management (IAM).

Registering a Public Cloud Account

If you do not have a Huawei Cloud account, register one.

1. Open the official public cloud website (**https://www.huaweicloud.com/intl/en-us/**) and click **Register** in the upper right corner. The registration page is displayed.
2. Enter registration information as prompted..
3. After the registration is successful, you can be automatically logged in to Huawei Cloud.

Using GaussDB(DWS) with IAM

This section describes how to use **IAM** to implement fine-grained permissions control for your GaussDB(DWS) resources. With IAM, you can:

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials, providing access to GaussDB(DWS) resources.
- Grant only the permissions required for users to perform specific tasks.
- Entrust a Huawei Cloud account or service to perform professional and efficient O&M on your GaussDB(DWS) resources.

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

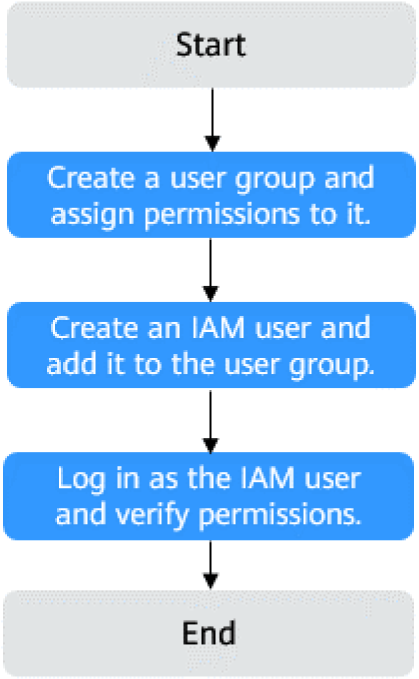
This section describes the procedure for granting permissions (see **IAM usage process**).

Prerequisites for Using IAM

Before assigning permission policies to a user group, you need to understand the GaussDB(DWS) permission policies. For details about the system policies supported by GaussDB(DWS), see **Supported System Policies**. For the system policies of other services, see **System Permissions**.

IAM usage process

Figure 2-1 Procedure


Start
Create a user group and assign permissions to it.
Create an IAM user and add it to the user group.
Log in as the IAM user and verify permissions.
End

1. **Create a user group and assign permissions**.

   Use the Huawei Cloud account to log in to the **IAM console**, create a user group, and attach the **DWS ReadOnlyAccess** policy to the group.

2. **Create a user and add it to a user group** .

   Create a user on the IAM console and add the user to the group created in Step **1**.

3. **Log in** and verify the permissions.

   Log in to the management console by using the user created and verify the user permissions.

   – Choose **Service List** > **Data Warehouse Service** to enter the GaussDB(DWS) management console, and click **Create DWS Cluster** to create a data warehouse cluster. If you cannot create one, the **DWS ReadOnlyAccess** policy has taken effect.

   – Choose any other service in **Service List**. If only the **DWS ReadOnlyAccess** policy is added and a message is displayed indicating that you have insufficient permission to access the service, **DWS ReadOnlyAccess** has taken effect.

2.2 Creating a GaussDB(DWS) Custom Policy

Custom policies can be created as a supplement to the system policies of GaussDB(DWS). For details about the custom policy actions, see **Permissions Policies and Supported Actions**.

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

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. This section provides examples of GaussDB(DWS) custom policies.

Custom Policy Examples

- Example 1: allowing users to create/restore, restart, and delete a cluster, configure security parameters, and reset passwords

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dws:cluster:create",
                "dws:cluster:restart",
                "dws:cluster:delete",
                "dws:cluster:setSecuritySettings",
                "dws:cluster:resetPassword",
                "dws:*:list*",
                "dws:*:get*",
                "tms:predefineTags:list"
                "ecs:*:get*",
                "ecs:*:list*",
                "elb:*:list*",
                "ecs:*:create*",
                "ecs:*:delete*",
                "vpc:*:get*",
                "vpc:*:list*",
                "vpc:*:create*",
                "vpc:*:delete*",
                "evs:*:get*",
                "evs:*:list*",
                "evs:*:create*",
                "evs:*:delete*"
            ]
        }
    ]
}
```

- Example 2: using wildcard character (*)

  For example, the following policy has all operation permissions on GaussDB(DWS) snapshots.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
                "Action": [
                    "dws:snapshot:*",
                    "dws:cluster:list",
                    "dws:openAPISnapshot:detail",
                    "dws:cluster:getDetail",
                    "ecs:*:get*",
                    "ecs:*:list*",
                    "vpc:*:get*",
                    "vpc:*:list*"
                ]
            }
    ]
}
```

- Example 3: denying cluster deletion

  A deny policy must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain 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(DWS) FullAccess** policy to a user but also forbid the user from deleting clusters. Create a custom policy for denying cluster deletion, and assign both policies to the group the user belongs to. Then the user can perform all operations on GaussDB(DWS) except deleting clusters. The following is an example of a deny policy:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dws:*:list*",
                "dws:*:get*"
            ]
         },
         {
        "Effect": "Deny",
                "Action": [
                    "dws:cluster:delete"
                ]
         }
    ]
}
```

- Example 4: defining multiple actions in a policy

  A custom policy can contain actions of multiple services that are all of the global or project-level type. The following is an example policy containing actions of multiple services:

```
{
        "Version": "1.1",
        "Statement": [
                {
                "Effect": "Allow",
                "Action": [
                        "dws:cluster:create",
                        "dws:cluster:restart",
                        "dws:cluster:setSecuritySettings",
                        "dws:*:get*",
                        "dws:*:list*",
                        "tms:predefineTags:list",
                        "elb:*:list*",
                        "ecs:*:get*",
                        "ecs:*:list*",
                        "ecs:*:create*",
                        "vpc:*:get*",
                        "vpc:*:list*",
                        "vpc:*:create*",
                        "evs:*:get*",
                        "evs:*:list*",
                        "evs:*:create*"
                    ]
                },
               {
           "Effect": "Deny",
                    "Action": [
                        "dws:cluster:delete"
                    ]
               }

        ]
  }
```

2.3 Allowing GaussDB(DWS) to Manage Resources

Huawei Cloud services interwork with each other, and certain operations require cooperation with other cloud services. To do so, you need to create a cloud service delegation and give GaussDB(DWS) permission to perform certain resource management tasks on your behalf by authorizing it to operate other cloud services.

NOTE

- We are currently rectifying agency permissions. Previously, agencies relied on IAM permissions. Now, they are being migrated to a new system. To switch to the new, lower-permission agency for better resource protection, contact O&M personnel.
- By default, only Huawei Cloud accounts or users with **Security Administrator** permissions can query and create agencies. By default, the IAM user does not have permission to query or create agencies. If you lack these permissions, contact an authorized user to grant access.
- The agency permission is obtained from the cache. The cache is updated once an hour. If you update an agency, the update will take effect one hour later.

GaussDB(DWS) Agency Permissions (New)

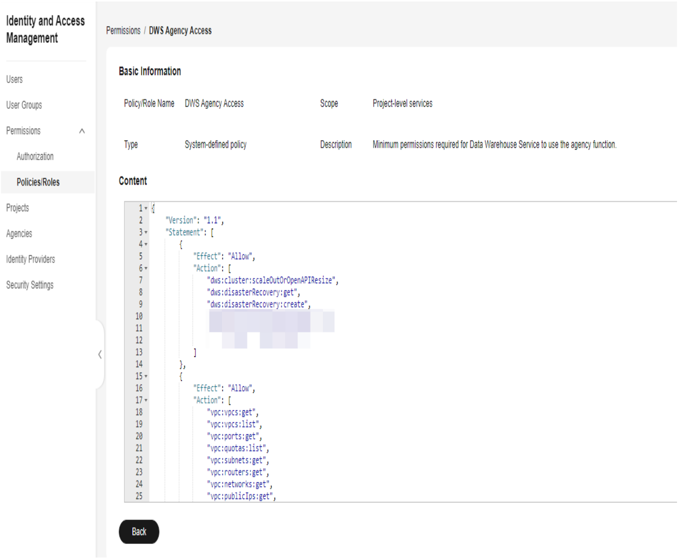
GaussDB(DWS) provides the following agency permissions based on the APIs on which the agency operation depends:

Table 2-1 GaussDB(DWS) agency permissions

Agency	Agency Permission	Scenario
DWSAgencyAccess	DWS Agency Access	Minimum permissions on which GaussDB(DWS) depends when using the agency function. For example, LTS depends only on **lts:groups:put**, and the system policy has only the operation permission on LTS.

You can log in to the IAM management console, choose Permission Management > Permissions, and click the **"DWS Agency Access"** permission to view the complete dependency information.

Figure 2-2 DWS Agency Access permission information



Basic Information

Policy/Role Name: DWS Agency Access

Scope: Project-level services

Type: System-defined policy

Description: Minimum permissions required for Data Warehouse Service to use the agency function.

Content

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dws:cluster:scaleOutOrOpenAPIResize",
                "dws:disasterRecovery:get",
                "dws:disasterRecovery:create",
                [redacted]
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "vpc:vpcs:get",
                "vpc:vpcs:list",
                "vpc:ports:get",
                "vpc:quotas:list",
                "vpc:subnets:get",
                "vpc:routers:get",
                "vpc:networks:get",
                "vpc:publicIps:get",
```

GaussDB(DWS) Agency Permissions (Old)

The following table describes the dependency scenarios of the old agency permissions.

Table 2-2 Agency and permission usage

Agency	Agency Permission	Scenario
DWSAccessLTS	LTS FullAccess	LTS collects and reports logs to LTS.
DWSAccessOBS	OBS Administrator	Audit log dump: reports audit logs to OBS buckets.
DWSAccessKMS	KMS Administrator	Used to query and rotate keys in a KMS encrypted cluster.
DWSAccessVPC	Server Administrator	If a node is faulty, the EIP is automatically migrated from the faulty node to a normal node.

Agency	Agency Permission	Scenario
DWSAccessDWS	Tenant Administrator	● In the DWS 3.0 scenario, the read-only logical cluster is scaled out or in periodically based on the automatic addition or deletion plan. ● In the scale-in scenario, clear user NICs and configure security group rules. ● When a node is faulty, ELB adds or deletes a listener instance.

2.4 Syntax of Fine-Grained Permissions Policies

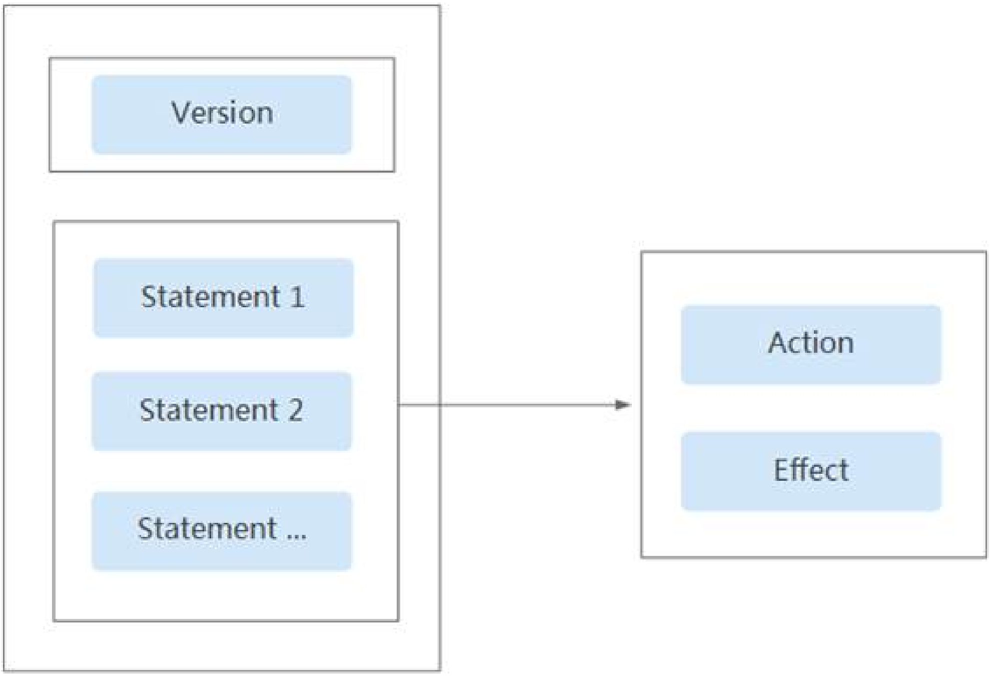
In actual services, you may need to grant different operation permissions on resources to users of different roles. The IAM service provides fine-grained access control. An IAM administrator (a user in the **admin** group) can create a custom policy containing required permissions. After a policy is granted to a user group, users in the group can obtain all permissions defined by the policy. In this way, IAM implements fine-grained permission management.

To control the GaussDB(DWS) operations on resources more precisely, you can use the user management function of IAM to grant different operation permissions to users of different roles for fine-grained permission control.

Policy Structure

A fine-grained policy consists of a Version and a Statement. Each policy can have multiple statements.

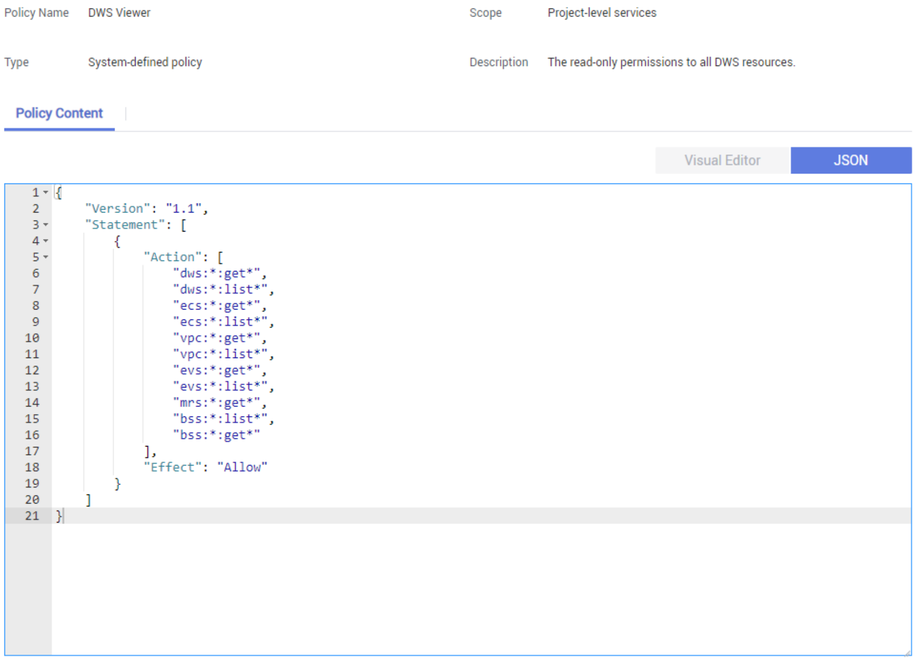
Figure 2-3 Policy structure



Policy Syntax

In the navigation pane on the IAM console, click **Policies** and then click the name of a policy to view its details. The **DWS ReadOnlyAccess** policy is used as an example to describe the syntax of fine-grained policies.

Figure 2-4 Setting the policy



Policy Name DWS Viewer

Scope Project-level services

Type System-defined policy

Description The read-only permissions to all DWS resources.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "dws:*:get*",
                "dws:*:list*",
                "ecs:*:get*",
                "ecs:*:list*",
                "vpc:*:get*",
                "vpc:*:list*",
                "evs:*:get*",
                "evs:*:list*",
                "mrs:*:get*",
                "bss:*:list*",
                "bss:*:get*"
            ],
            "Effect": "Allow"
        }
    ]
}
```

```
{
       "Version": "1.1",
    "Depends": [],
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dws:*:get*",
                "dws:*:list*",
                "ecs:*:get*",
                "ecs:*:list*",
                "vpc:*:get*",
                "vpc:*:list*",
                "evs:*:get*",
                "evs:*:list*",
                "mrs:*:get*",
                "bss:*:list*",
                "bss:*:get*"
            ]
        }
    ]
}
```

- **Version**: Distinguishes between role-based access control (RBAC) and fine-grained policies.
    - **1.0**: RBAC policies. An RBAC policy consists of permissions for an entire service. Users in a group with such a policy assigned are granted all of the permissions required for that service.
    - **1.1**: Fine-grained policies. A fine-grained policy consists of API-based permissions for operations on specific resource types. Fine-grained policies, as the name suggests, allow for more fine-grained control than RBAC policies. Users granted permissions of such a policy can only perform specific operations on the corresponding service. Fine-grained policies include system and custom policies.
- **Depends**: dependency item.
- **Statement**: Permissions defined by a policy, including Effect and Action.
    - Effect

      The value of **Effect** can be **Allow** or **Deny**. System policies contain only **Allow** statements. For custom policies containing both **Allow** and **Deny** statements, **Deny** statements take precedence over **Allow** statements.
    - Action

      Actions allowed on resources. An action is in the format of *Service name*:*Resource type*:*Action*. A policy can contain one or more actions. You can use a wildcard (*) to indicate all services, resource types, or actions.

      Example: **dws:cluster:create**, permissions for create data warehouse clusters.

List of Supported Actions

When creating a custom policy on IAM, you can add the operations on GaussDB(DWS) resources or the permissions corresponding to RESTful APIs to the action list of the policy authorization statement so that the policy contains the operation permissions. The following table lists the GaussDB(DWS) permissions.

- **REST API**

  For details about REST API actions supported by GaussDB(DWS), see **Permissions Policies and Supported Actions**.

- **Management console operations**

  Table 2-3 describes the GaussDB(DWS) operations on resources and corresponding permissions.

  NOTE

  - Some GaussDB(DWS) permissions depend on the actions of ECS, VPC, EVS, ELB, MRS, and OBS. Grant GaussDB(DWS) the required service admin permissions.
  - The table shows frequently used GaussDB(DWS) APIs, but some only allow project-based authentication (IAM authentication) and not enterprise project authentication. To use these APIs, they must be configured on the IAM authentication page.

Table 2-3 GaussDB(DWS) permissions

Operation	Permission	Dependent Permission	Scope
Creating a cluster	"dws:cluster:create"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "vpc:securityGroupRules:delete", "vpc:ports:update", "evs:*:get*", "evs:*:list*", "evs:*:create*",	● Scope: – Project – Enterprise Project
Obtaining the cluster list	"dws:cluster:list"	"dws:*:get*", "dws:*:list*",	● Not supported – Enterprise Project ● Scope: – Project
Obtaining the details of a cluster	"dws:cluster:getDetail"	"dws:*:get*", "dws:*:list*", "vpc:vpcs:list", "vpc:securityGroups:get"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Setting automated snapshot policy	"dws:cluster:setAutomatedSnapshot"	"dws:backupPolicy:list"	• Scope: – Project – Enterprise Project
Setting security parameters/ parameter groups	"dws:cluster:setSecuritySettings"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise Project
Restarting a Cluster	"dws:cluster:restart"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise Project
Scaling out clusters	"dws:cluster:scaleOut"	"dws:*:get*", "dws:*:list*", "dws:cluster:scaleOutOrOpenAPIResize", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "vpc:*:update*", "evs:*:get*", "evs:*:list*", "evs:*:create*",	• Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Scaling out or resizing a cluster via API	"dws:cluster:scaleOutOrOpenAPIResize"	"dws:*:get*", "dws:*:list*", "vpc:vpcs:list", "vpc:ports:create", "vpc:ports:get", "vpc:ports:update", "vpc:subnets:get", "vpc:subnets:update", "vpc:subnets:create", "vpc:routers:get", "vpc:routers:update", "vpc:networks:create", "vpc:networks:get", "vpc:networks:update", "ecs:serverInterfaces:use", "ecs:serverInterfaces:get", "ecs:cloudServerFlavors:get"	● Scope: – Project – Enterprise Project
Resetting Your Password	"dws:cluster:resetPassword"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Deleting a cluster	"dws:cluster:delete"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:delete*", "vpc:*:get*", "vpc:*:list*", "vpc:*:delete*", "evs:*:get*", "evs:*:list*", "evs:*:delete*",	• Scope: – Project – Enterprise Project
Configuring maintenance windows	"dws:cluster:setMaintainceWindow"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise Project
Binding EIPs	"dws:eip:operate"	"dws:*:get*", "dws:*:list*", "eip:*:get*", "eip:*:list*"	• Scope: – Project – Enterprise Project
Unbinding EIPs	"dws:eip:operate"	"dws:*:get*", "dws:*:list*", "eip:*:get*", "eip:*:list*"	• Scope: – Project – Enterprise Project
Creating DNS domain names	"dws:dns:create"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise Project
Releasing DNS domain names	"dws:dns:release"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Modifying DNS domain names	"dws:dns:edit"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project
Creating MRS connections	"dws:MRSConnection:create"	"dws:*:get*", "dws:*:list*", "mrs:*:get*", "mrs:*:list*", "mrs:cluster:create", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*"	● Scope: – Project – Enterprise Project
Updating MRS connections	"dws:MRSConnection:update"	"dws:*:get*", "dws:*:list*", "mrs:*:get*", "mrs:*:list*", "mrs:cluster:create", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Deleting MRS connections	"dws:MRSConnection:delete"	"dws:*:get*", "dws:*:list*", "mrs:*:get*", "mrs:*:list*", "mrs:cluster:create" "ecs:*:get*", "ecs:*:list*", "ecs:*:delete*", "vpc:*:get*", "vpc:*:list*", "vpc:*:delete*", "evs:*:get*", "evs:*:list*", "evs:*:delete*",	● Scope: – Project – Enterprise Project
MRS data source list	"dws:MRSSource:list"	"mrs:cluster:list", "mrs:tag:listResource", "mrs:tag:list", "dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Adding/Deleting tags	"dws:tag:addAndDelete"	"dws:*:get*", "dws:*:list*", "dws:openAPITag:update", "dws:openAPITag:getResourceTag",	● Scope: – Project – Enterprise Project
Editing tags	"dws:tag:edit"	"dws:*:get*", "dws:*:list*", "dws:openAPITag:update", "dws:openAPITag:getResourceTag",	● Scope: – Project – Enterprise Project
Creating a snapshot	"dws:snapshot:create"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Obtaining the snapshot list	"dws:snapshot:list"	--	● Scope: – Project – Enterprise Project
Viewing the snapshot list of a cluster	"dws:clusterSnapshot:list"	"dws:cluster:list", "dws:openAPICluster:getDetail"	● Scope: – Project – Enterprise Project
Deleting snapshots	"dws:snapshot:delete"	"dws:snapshot:list"	● Scope: – Project – Enterprise Project
Copying snapshots	"dws:snapshot:copy"	"dws:snapshot:list", "dws:snapshot:create"	● Scope: – Project – Enterprise Project
Restoring data to a new cluster	"dws:cluster:restore"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Resizing a cluster	"dws:cluster:resize"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "ecs:*:delete*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "vpc:*:delete*", "evs:*:get*", "evs:*:list*", "evs:*:create*", "evs:*:delete*"	● Scope: – Project – Enterprise Project
Switchback	"dws:cluster:switchover"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Querying the ELB list	"dws:elb:list"	"dws:*:get*", "dws:*:list*", "elb:*:get*", "elb:*:list*",	● Not supported – Enterprise Project ● Scope: – Project
Associating ELB	"dws:elb:bind"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "vpc:*:get*", "vpc:*:list*", "evs:*:get*", "evs:*:list*", "elb:*:get*", "elb:*:list*", "elb:*:delete*", "elb:*:create*",	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Disassociating ELB	"dws:elb:unbind"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "vpc:*:get*", "vpc:*:list*", "evs:*:get*", "evs:*:list*", "elb:*:get*", "elb:*:list*", "elb:*:delete*",	● Scope: – Project – Enterprise Project
Querying snapshot configurations	"dws:snapshotConfig:list"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project
Updating a snapshot policy	"dws:backupPolicyDetail:update"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project
Deleting a snapshot policy	"dws:backupPolicy:delete"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project
Querying a snapshot policy	"dws:backupPolicy:list"	"dws:cluster:list"	● Scope: – Project – Enterprise Project
Querying cluster encryption information	"dws:clusterEncryptInfo:list"	"dws:*:get*", "dws:*:list*", "KMS Administrator"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Creating an agent	"dws:createAgency:create"	"dws:*:get*", "dws:*:list*", "security administrator"	● Scope: – Project – Enterprise Project
Querying OBS bucket information	"dws:queryBuckets:list"	"dws:*:get*", "dws:*:list*",	● Scope: – Project – Enterprise Project
Adding a node	"dws:expandWithExistedNodes:update"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "vpc:*:update*", "evs:*:get*", "evs:*:list*", "evs:*:create*",	● Scope: – Project – Enterprise Project
Deleting a DR backup	"dws:disasterRecovery:delete"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:delete*", "vpc:*:get*", "vpc:*:list*", "vpc:*:delete*", "evs:*:get*", "evs:*:list*", "evs:*:delete*"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Creating a DR backup	"dws:disasterRecovery:create"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*",	• Scope: – Project – Enterprise Project
Other DR and backup operations	"dws:disasterRecovery:otherOperate"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*"	• Scope: – Project – Enterprise Project
Querying DR and backup operations	"dws:disasterRecovery:get"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "vpc:*:get*", "vpc:*:list*", "evs:*:get*", "evs:*:list*"	• Scope: – Project – Enterprise Project
Adding a CN	"dws:module:install"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Deleting a CN	"dws:module:uninstall"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Removing nodes	"dws:clusterNodes:operate"	"dws:*:get*", "dws:*:list*"	- Scope: – Project – Enterprise Project
Updating the node alias	dws:instanceAliasName:update	dws:cluster:list	- Scope: – Project – Enterprise Project
Redistributing data	"dws:redistribution:operate"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Querying redistribution	"dws:redistributionInfo:list"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Stopping redistribution	"dws:redistribution:suspend"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Resuming redistribution	"dws:redistribution:recover"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Adding disk capacity	"dws:disk:expand"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*",	● Scope: – Project – Enterprise Project
Scaling in a cluster	"dws:cluster:shrink"	"dws:*:get*", "dws:*:list*", "dws:createAgency:create", "ecs:*:get*", "ecs:*:list*", "ecs:*:delete*", "vpc:*:get*", "vpc:*:list*", "vpc:*:delete*", "evs:*:get*", "evs:*:list*", "evs:*:delete*"	● Scope: – Project – Enterprise Project
Querying product specifications	"dws:specProduct:list"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Changing from pay-per-use to yearly/monthly	"dws:ondemandToPeriod:operate"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "vpc:securityGroupRules:delete", "evs:*:get*", "evs:*:list*", "evs:*:create*", "bss:coupon:view", "bss:order:pay", "bss:order:view", "bss:contract:update", "bss:balance:view", "bss:renewal:view", "bss:unsubscribe:update", "bss:renewal:update", "bss:order:update"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Modifying a yearly/monthly cluster	"dws:periodCluster:modify"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:delete*", "vpc:*:get*", "vpc:*:list*", "vpc:*:delete*", "evs:*:get*", "evs:*:list*", "evs:*:delete*", "bss:coupon:view", "bss:order:pay", "bss:order:view", "bss:contract:update", "bss:balance:view", "bss:renewal:view", "bss:unsubscribe:update", "bss:renewal:update", "bss:order:update"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Creating a yearly/monthly cluster	"dws:periodCluster:create"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*", "bss:coupon:view", "bss:order:pay", "bss:order:view", "bss:contract:update", "bss:balance:view", "bss:renewal:view", "bss:unsubscribe:update", "bss:renewal:update", "bss:order:update"	● Scope: – Project – Enterprise Project
Performing a check before cluster creation	"dws:checkCluster:create"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*",	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Performing a check before adding disk capacity to a yearly/monthly cluster	"dws:periodExpandPre-check:operate"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*", "vpc:*:get*", "vpc:*:list*", "vpc:*:create*", "evs:*:get*", "evs:*:list*", "evs:*:create*",	● Scope: – Project – Enterprise Project
Binding the management plane IP address	"dws:bindManageIp:operate"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Obtaining user authorization	"dws:checkAuthorize:operate"	"dws:*:get*", "dws:*:list*", "dws:checkSupport:operate"	● Scope: – Project – Enterprise Project
Authorizing a user	"dws:authorize:operate"	"dws:*:get*", "dws:*:list*", "dws:checkSupport:operate"	● Scope: – Project – Enterprise Project
Querying user databases	"dws:userDatabase:list"	"dws:*:get*", "dws:*:list*", "dws:checkSupport:operate"	● Scope: – Project – Enterprise Project
Querying user schemas	"dws:schemas:list"	"dws:*:get*", "dws:*:list*", "dws:checkSupport:operate"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Querying user tables	"dws:tables:list"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Restoring tables	"dws:tableRestore:operate"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Checking the name of the table to be restored	"dws:tableRestoreCheck:operate"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Checking whether a cluster supports fine-grained backup	"dws:checkSupport:operate"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Querying the list of flavors that can be changed	"dws:supportFlavors:list"	"dws:*:get*", "dws:*:list*",	- Scope: – Project – Enterprise Project
Changing the node flavor	"dws:specResize:operate"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:create*"	- Scope: – Project – Enterprise Project
Stopping snapshot creation	"dws:snapshot:stop"	"dws:snapshot:list"	- Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Terminating a session	"dws:dmsSession:terminate"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Workload report operations	"dws:dmsWorkloadDiagnosisReport:create"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Modifying an alarm rule	"dws:dmsAlarmRule:update"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Enabling an alarm rule	"dws:dmsAlarmRule:enable"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Enabling a cluster alarm	"dws:dmsClusterAlarm:enable"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Disabling a cluster alarm	"dws:dmsClusterAlarm:disable"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
gRPC external service	"dws:dmsGrpcOuter:operation"	"dws:dmsQuery:list", "dws:cluster:setSecuritySettings", "obs:bucket:ListAllMyBuckets"	- Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Adding a SQL probe	"dws:dmsProbe:add"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise Project
Modifying a SQL probe	"dws:dmsProbe:update"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise Project
Deleting a SQL probe	"dws:dmsProbe:delete"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise Project
Enabling or disabling a SQL probe	"dws:dmsProbe:enable"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise Project
Creating a User panel	"dws:dmsUserBoard:create"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project
Modifying a user panel	"dws:dmsUserBoard:update"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project
Deleting a user panel	"dws:dmsUserBoard:delete"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Terminating a query	"dws:dmsQuery:terminate"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Enabling or disabling DMS	"dws:dmsService:enableOrDisable"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Modifying DMS storage configurations	"dws:dmsStorageConfig:modify"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Obtaining, or creating a DDL review	"dws:dmsDdlExamine:getOrCreate"	"dws:dmsGrpcOuter:operation"	- Not supported – Enterprise Project - Scope: – Project
Workload snapshot operations	"dws:dmsWorkloadDiagnosisSnapshot:create"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Creating an alarm rule	"dws:dmsAlarmRule:add"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Deleting an alarm rule	"dws:dmsAlarmRule:delete"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Executing a SQL probe	"dws:dmsProbe:execute"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Deleting a monitoring item	"dws:dmsPerformanceMonitor:delete"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project
Enabling or disabling DMS monitoring metrics	"dws:dmsCollectItem:enableOrDisable"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Modifying DMS monitoring configurations	"dws:dmsCollectConfig:modify"	"dws:dmsGrpcOuter:operation"	- Scope: – Project – Enterprise Project
Conditional query	"dws:dmsQuery:list"	"dws:cluster:list"	- Scope: – Project – Enterprise Project
OpenAPI Conditional Query	"dws:dmsOpenapiQuery:list"	"dws:cluster:list"	- Scope: – Project – Enterprise Project
Disabling an alarm rule	"dws:dmsAlarmRule:disable"	"dws:dmsQuery:list"	- Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Deleting an alarm record	"dws:dmsAlarmRecord:delete"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project
Checking SQL probes	"dws:dmsProbe:check"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise Project
Adding a monitoring item	"dws:dmsPerformanceMonitor:add"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project
Modifying monitoring metrics	"dws:dmsPerformanceMonitor:update"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project
Downloading historical monitoring trend	"dws:dmsTrendHistory:down"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise Project
Obtaining cluster ring information	"dws:ring:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project – Enterprise Project
Obtaining the cluster process topology	"dws:processTopo:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Querying intelligent O&M information	"dws:operationalTask:get"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Intelligent O&M Operations	"dws:operationalTask:operate"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Adding, deleting, and modifying a logical cluster	"dws:logicalCluster:operate"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Querying a logical cluster	"dws:logicalCluster:get"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Elastic logical cluster planning	"dws:logicalClusterPlan:operate"	"dws:*:get*", "dws:*:list*", "dws:logicalCluster:*", "dws:cluster:scaleOut", "iam:agencies:*", "iam:permissions:*Agency*"	● Scope: – Project – Enterprise Project
Creating an endpoint service	"dws:vpcEndpointService:create"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project
Querying the resource management list	"dws:workLoadManager:get"	"dws:*:get*", "dws:*:list*"	● Scope: – Project – Enterprise Project

Operation	Permission	Dependent Permission	Scope
Resource management operations	"dws:workLoadManager:operate"	"dws:*:get*", "dws:*:list*"	- Scope: – Project – Enterprise Project
LTS operations	"dws:ltsAccess:operate"	"dws:*:get*", "dws:*:list*"	- Scope: – Project – Enterprise Project
Querying LTS Information	"dws:ltsAccess:get"	"dws:*:get*", "dws:*:list*"	- Scope: – Project – Enterprise Project
Querying events	"dws:event:list"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project
Querying event specifications	"dws:event:list"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project
Querying event subscriptions	"dws:eventSub:list"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project

Operation	Permission	Dependent Permission	Scope
Creating an event subscription	"dws:eventSub:create"	"dws:*:get*", "dws:*:list*",	● Not supported – Enterprise Project ● Scope: – Project
Updating an event subscription	"dws:eventSub:update"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Deleting an event subscription	"dws:eventSub:delete"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Querying alarm statistics	"dws:alarmStatistic:list"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Querying alarm details	"dws:alarmDetail:list"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project

Operation	Permission	Dependent Permission	Scope
Querying alarm configurations	"dws:alarmConfig:list"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project
Querying alarm subscriptions	"dws:alarmSub:list"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project
Creating an alarm subscription	"dws:alarmSub:create"	"dws:*:get*", "dws:*:list*",	- Not supported – Enterprise Project - Scope: – Project
Updating an alarm subscription	"dws:alarmSub:update"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project
Deleting an alarm subscription	"dws:alarmSub:delete"	"dws:*:get*", "dws:*:list*"	- Not supported – Enterprise Project - Scope: – Project

Operation	Permission	Dependent Permission	Scope
Delivering cluster upgrade operations (upgrade, rollback, submission, and retry)	"dws:cluster:doUpdate"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Querying the available upgrade paths of a cluster	"dws:cluster:getUpgradePaths"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Querying cluster upgrade records	"dws:cluster:getUpgradeRecords"	"dws:*:get*", "dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Starting a cluster	"dws:cluster:startCluster"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:start", "ecs:*:stop"	● Not supported – Enterprise Project ● Scope: – Project
Stopping a cluster	"dws:cluster:stopCluster"	"dws:*:get*", "dws:*:list*", "ecs:*:get*", "ecs:*:list*", "ecs:*:start", "ecs:*:stop"	● Not supported – Enterprise Project ● Scope: – Project

Operation	Permission	Dependent Permission	Scope
Obtaining discount nodes of a cluster	"dws:cluster:listDiscountNode"	"dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Obtaining tags	"dws:openAPItag:list"	"dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Service EPS list	"dws:service:listEps"	"dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Obtaining the DR information	"dws:disasterRecovery:get"	"dws:*:*"	● Not supported – Enterprise Project ● Scope: – Project
Cluster restoration check	"dws:cluster:checkRestore"	"dws:*:*"	● Not supported – Enterprise Project ● Scope: – Project

Operation	Permission	Dependent Permission	Scope
Static alarm list	"dws:alarmStatistic:list"	"dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Obtaining static resource information	"dws:service:getResourceStatistics"	"dws:*:*"	● Not supported – Enterprise Project ● Scope: – Project
Alarm details list	"dws:alarmDetail:list"	"dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project
Obtaining the cluster details	"dws:openAPICluster:getDetail"	"dws:*:*"	● Not supported – Enterprise Project ● Scope: – Project
Cluster event specifications	"dws:eventSpec:list"	"dws:*:list*"	● Not supported – Enterprise Project ● Scope: – Project

Operation	Permission	Dependent Permission	Scope
Cluster DR list	"dws:cluster:listDisasterRecovery"	"dws:*:list*",	- Not supported – Enterprise Project - Scope: – Project

Authorization Using the Fine-Grained Permission Policy

Step 1 Log in to the IAM console as and create a user-defined policy.

For details, see **Creating Custom Policies** in the *Identity and Access Management User Guide*.

Refer to the following to create the policy:

- Use the IAM administrator account, that is, the user in the admin user group, because only the IAM administrator has the permissions to create users and user groups and modify user group permissions.
- GaussDB(DWS) is a project-level service, so its **Scope** must be set to **Project-level service**. If this policy is required to take effect for multiple projects, authorization is required to each project.
- Two GaussDB(DWS) policy templates are preconfigured on IAM. When creating a custom policy, you can select either of the following templates and modify the policy authorization statement based on the template:
  - **DWS Admin**: has all execution permissions on GaussDB(DWS).
  - **DWS Viewer**: has the read-only permission on GaussDB(DWS).
- You can add permissions corresponding to GaussDB(DWS) operations or RESTful APIs listed in **List of Supported Actions** to the action list in the policy authorization statement, so that the policy can obtain the permissions.

  For example, if **dws:cluster:create** is added to the action list of a policy statement, the policy has the permission to create or restore clusters.
- If you want to use other services, grant related operation permissions on these services. For details, see the help documents of related services.

  For example, when creating a data warehouse cluster, you need to configure the VPC to which the cluster belongs. To obtain the VPC list, add permission **vpc:*:get*** to the policy statement.

Step 2 Create a user group.

For details, see **Creating a User Group** in the *Identity and Access Management User Guide*.

Step 3 Add users to the user group and grant the new custom policy to the user group so that users in it can obtain the permissions defined by the policy.

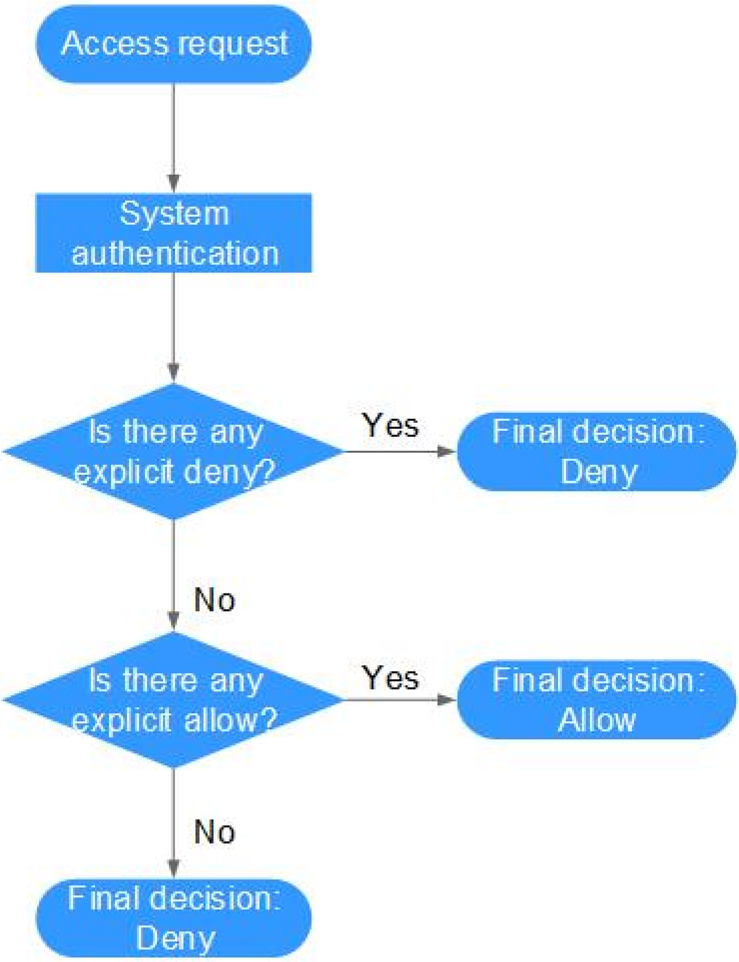
For details, see **Viewing and Modifying User Group Information** in the *Identity and Access Management User Guide*.

----End

Authentication Logic

If a user is granted permissions of multiple policies or of only one policy containing both Allow and Deny statements, then authentication starts from the Deny statements. The following figure shows the authentication logic for resource access.

Figure 2-5 Authentication logic



NOTE

The actions in each policy bear the OR relationship.

1. A user accesses the system and makes an operation request.
2. The system evaluates all the permissions policies assigned to the user.
3. In these policies, the system looks for explicit deny permissions. If the system finds an explicit deny that applies, it returns a decision of Deny, and the authentication ends.

4. If no explicit deny is found, the system looks for allow permissions that would apply to the request. If the system finds an explicit allow permission that applies, it returns a decision of Allow, and the authentication ends.
5. If no explicit allow permission is found, IAM returns a decision of Deny, and the authentication ends.

2.5 RBAC Syntax of RBAC Policies

Policy Structure

An RBAC policy consists of a Version, a Statement, and Depends.

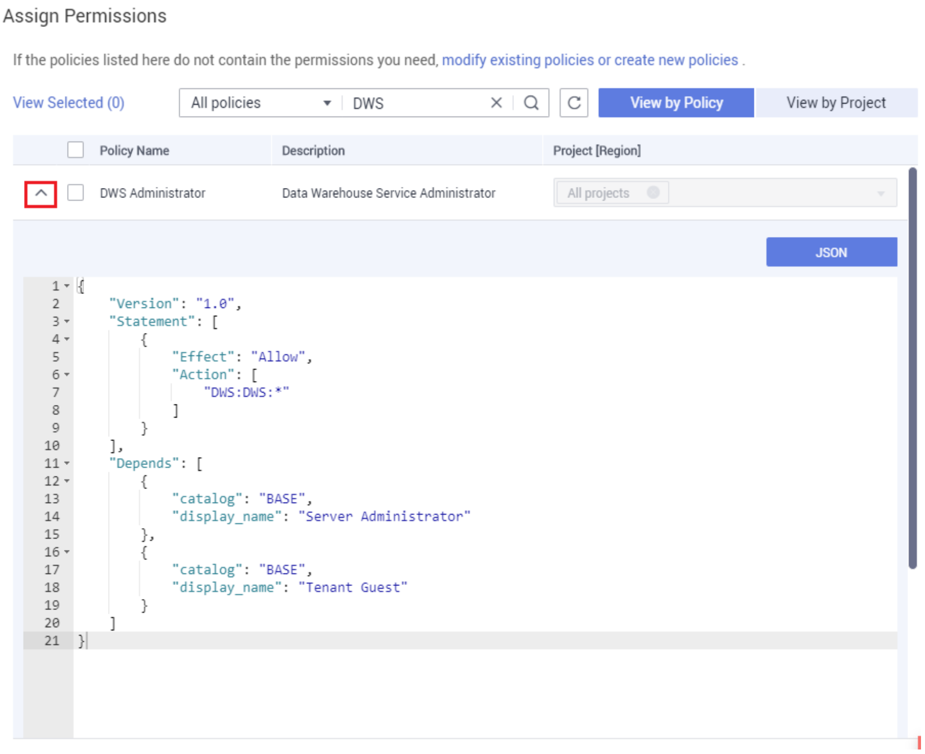
Figure 2-6 RBAC policy structure

Version
Statement
Depends

Policy Syntax

When selecting a policy for a user group, click ⌄ below the policy to view the details of the policy. The **DWS Administrator** policy is used as an example to describe the syntax of RBAC policies.

Figure 2-7 Syntax of RBAC Policies



Assign Permissions

If the policies listed here do not contain the permissions you need, modify existing policies or create new policies .

Policy Name	Description	Project [Region]
DWS Administrator	Data Warehouse Service Administrator	All projects

```
{
    "Version": "1.0",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "DWS:DWS:*"
            ]
        }
    ],
    "Depends": [
        {
            "catalog": "BASE",
            "display_name": "Server Administrator"
        },
        {
            "catalog": "BASE",
            "display_name": "Tenant Guest"
        }
    ]
}
```

```
{
        "Version": "1.0",
        "Statement": [
                {
                        "Effect": "Allow",
                        "Action": [
                                "dws:dws:*"
                        ]
                }
        ],
        "Depends": [
                {
                        "catalog": "BASE",
                        "display_name": "Server Administrator"
                },
                {
                        "catalog": "BASE",
                        "display_name": "Tenant Guest"
                }
        ]
}
```

Parameter	Meaning	Value
Version	Policy version	The value is fixed to **1.0**.

Parameter		Meaning	Value
Statement	Action	Operations to be performed on GaussDB(DWS)	Format: *Service name:Resource type:Operation*. **dws:dws:***: Permissions for performing all operations on all resource types in GaussDB(DWS).
	Effect	Whether the operation defined in an action is allowed	● Allow ● Deny
Depends	catalog	Name of the service to which dependencies of a policy belong	Service name Example: **BASE**
	display_name	Name of a dependent policy	Policy name Example: **Server Administrator**

NOTE

When using RBAC for authentication, pay attention to the **Depends** parameter and grant other dependent permissions at the same time.

For example, the **DWS Administrator** permission depends on the **Server Administrator** and **Tenant Guest** permissions. When granting the **DWS Administrator** permission to users, you also need to grant the two dependent permissions to the users.

3 Creating a GaussDB(DWS) Cluster

3.1 Creating a Dedicated GaussDB(DWS) Cluster

3.1.1 Creating a GaussDB(DWS) Storage-Compute Coupled Cluster

To use Huawei Cloud GaussDB(DWS), create a data warehouse cluster first. When you create a data warehouse cluster, the yearly/monthly billing mode is used by default, which is more favorable than the pay-per-use billing mode. You can customize the computing resources and storage space of the cluster. If you select the pay-per-use mode, nodes will be billed by actual duration of use, with a billing cycle of one hour. This mode is flexible. You can enable or disable the service whenever you like.

This section describes how to create a data warehouse cluster on the GaussDB(DWS) console.

⚠ WARNING

- You are advised not to use clusters with low specifications, such as clusters with 16 GB memory and 4-core vCPUs, in the production environment. Otherwise, resource overload may occur.
- Configure **load balancing** to balance the connections to each CN and high availability of the cluster and prevent service interruptions. You are not advised to directly connect services to a single CN.
- The GaussDB(DWS) clusters under the same account are physically isolated and cannot share data. You can import data from a remote GaussDB(DWS) cluster to a local one by using a foreign table. For details, see **Tutorial: Importing Remote GaussDB(DWS) Data Sources**.
- To ensure stable service running, read **Before You Start: Performance Management Requirements** and **Before You Start: High Availability and Reliability Requirements** after creating a cluster.

Preparations Before Creating a Cluster

- You have evaluated the flavor of cluster nodes.

  You can select the number of nodes by data volume, service load, and performance. More nodes bring you stronger storage and compute capabilities.

  When first using GaussDB(DWS), you can create a cluster with a smaller flavor. Then, you can adjust the cluster scale and node flavor based on the data volume and service load changes without interrupting services. For details, see **Scaling Out a Cluster**.
- Determine the number of nodes that can be used by users.

  The number of nodes that can be used by users must meet the following requirements. Otherwise, the system displays a message indicating that the cluster cannot be created.

  The number of available nodes depends on the product type you choose. A storage-compute coupled data warehouse (standalone) has only one node. For other types of clusters, the number of nodes can be greater than or equal to 3. You can view the number of available nodes on the **Clusters** > **Dedicated Clusters** page.

Creating a Cluster

Step 1 Go to the **page for creating a GaussDB(DWS) cluster**.

Step 2 Choose **Region** and select the actual working region of the cluster node.

For more information about regions, visit **Regions and Endpoints**.

Step 3 Select a billing mode. For more information, see **Pricing Details**.

- Yearly/Monthly: If you select **Yearly/Monthly**, you need to set the required duration in **Step 13** before proceeding with the following steps.

  NOTE

  If the current console does not support this billing mode, contact technical support.
- **Pay-per-use** (hourly): If you select this billing mode, go to the next step.

Step 4 Select an AZ. You can select **Single AZ** or **Multi-AZ** as required.

For more information, see **Regions and AZs**.

NOTE

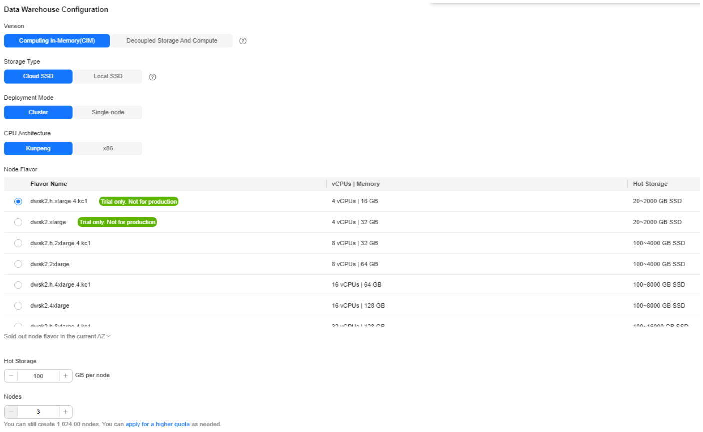
- Multi-AZ clusters are supported only by clusters of version 8.2.0.100 or later.
- The **Multi-AZ** option is displayed only if the number of AZs in the selected region is greater than or equal to 3. If this condition is not met, only a single-AZ cluster can be created.
- For a multi-AZ cluster, only three AZs can be selected at a time so far. Server nodes are evenly distributed among the three AZs.
- The numbers of nodes in a multi-AZ cluster must be a multiple of 3.
- In a multi-AZ cluster, the number of DNs must be less than or equal to 2.

Step 5 Configure **Resource**, **CPU Architecture**, and **Node Flavor**.

NOTE

- The number of nodes in a new cluster cannot exceed the quota that can be used by a user or 256. If the node quota is insufficient, click **Increase quota** to submit a service ticket and apply for higher node quota.
- If you have yearly/monthly nodes that meet service requirements, you are advised to use these nodes first to save costs. You can select **Yearly/Monthly** for **Billing Mode**.
- After a cluster is created, its type cannot be changed. For details about the differences between product types, see **Data Warehouse Types**.

Figure 3-1 Configuring node parameters



Data Warehouse Configuration

Version

[✓ Computing In-Memory(CIM)] Decoupled Storage And Compute

Storage Type

[✓ Cloud SSD] Local SSD

Deployment Mode

[✓ Cluster] Single-node

CPU Architecture

[✓ Kunpeng] x86

Node Flavor

	Flavor Name	vCPUs | Memory	Hot Storage
●	dwsk2.h.xlarge.4.kc1 Trial only. Not for production	4 vCPUs | 16 GB	20~2000 GB SSD
○	dwsk2.xlarge Trial only. Not for production	4 vCPUs | 32 GB	20~2000 GB SSD
○	dwsk2.h.2xlarge.4.kc1	8 vCPUs | 32 GB	100~4000 GB SSD
○	dwsk2.2xlarge	8 vCPUs | 64 GB	100~4000 GB SSD
○	dwsk2.h.4xlarge.4.kc1	16 vCPUs | 64 GB	100~8000 GB SSD
○	dwsk2.4xlarge	16 vCPUs | 128 GB	100~8000 GB SSD
○	dwsk2.h.8xlarge.4.kc1	32 vCPUs | 128 GB	100~16000 GB SSD

Sold-out node flavor in the current AZ

Hot Storage

100 GB per node

Nodes

3

You can still create 1,024.00 nodes. You can apply for a higher quota as needed.

Table 3-1 Node configuration parameters

Parameter	Description	Example Value
Resource	The options are as follows: ● **Computing In-Memory(CIM):** The storage-compute coupled data warehouse provides enterprise-level data warehouse services with high performance, high scalability, high reliability, high security, low latency, and easy O&M. It is capable of data analysis at a scale of 2,048 nodes and 20 petabytes of data and is suitable for converged analysis services that integrate databases, warehouses, marts, and lakes. ● **Decoupled Storage and Compute**: The storage-compute decoupled data warehouse is designed with a cloud native architecture that separates storage and compute. It also features hierarchical auto scaling for computing and storage, as well as multi-logical cluster shared storage technology (Virtual Warehouse or VW). These capabilities allow for computing isolation and concurrent expansion to handle varying loads, making it an ideal choice for OLAP analysis scenarios.	-
Storage Type	It can be: ● **Cloud SSD** ● **Extreme SSD**: suitable for workloads that demand super-high bandwidth and super-low latency. ● **Extreme SSD V2**: ultra-high-performance SSD EVS disks dedicated for latency-sensitive mission-critical applications. ● Local SSD **NOTE** Local SSD disks do not support disk scale-out. For more information, see **Disk Types and Performance**.	-

Parameter	Description	Example Value
Deployment Mode (storage-compute coupling)	The options are: ● **Cluster**: The storage-compute coupled data warehouse is capable of data analysis at a scale of 2,048 nodes and 20 petabytes of data. It is suitable for converged analysis services that integrate databases, warehouses, marts, and lakes. ● **Single-node**: A standalone data warehouse does not provide HA capabilities. It can be restored by the automatic reconstruction of ECS, and its data reliability is ensured by the EVS multi-copy mechanism. It is less expensive than other specifications. It is a good choice for lightweight services.	Cluster
CPU Architecture	The following CPU architectures can be selected: ● **x86** ● **Kunpeng** **NOTE** The x86 and Kunpeng architectures differ only in their underlying structure, which is not sensible to the application layer. Both architectures use the same SQL syntax. If you need to create a cluster and find that x86 servers are not enough, you can opt for the Kunpeng architecture.	-
Node Flavor	Select a node flavor. Each node flavor shows the vCPU, memory, and recommended application scenario. For more information about the node flavors supported by GaussDB(DWS) and their prices, see the **GaussDB(DWS) pricing details**. For details about the node flavors supported by GaussDB(DWS), see **Data Warehouse Specifications**.	dws.dc.4xlarge

Parameter	Description	Example Value
Hot storage	Available storage capacity of each node. **NOTE** ● The storage capacity you apply for has the necessary file system overhead, which includes index nodes and the space required for database running. The storage space must be an integer multiple of 100. ● 200 GB per node is the actual storage capacity for service data. For example, if the number of nodes is set to 3, the total resource capacity is 600 GB. ● By default, tablespaces are automatically created when you configure cold and hot data storage. You do not need to manually create tablespaces. This feature is supported only in clusters of 8.1.3 and later versions.	-
Nodes	Specify the number of nodes in the cluster. The number of nodes ranges from 3 to 256.	3
Total	Display the cluster's total capacity. The storage capacity of each flavor is the actual database space used for storing data. The displayed storage capacity has deducted the disk space consumed by backups and RAIDs.	-

Step 6 Click **Next: Configure Network**.

Step 7 Configure the network.

Table 3-2 Network parameters

Parameter	Description	Example Value
VPC	Specify a VPC to isolate the cluster's network. If you create a data warehouse cluster for the first time and have not configured the VPC, click **View VPC**. On the VPC management console that is displayed, create a VPC as needed. For details about how to create a VPC, see **Creating a VPC** in the *Virtual Private Cloud User Guide*. After selecting a VPC from the drop-down list, click **View VPC** to enter the VPC management console and view the detailed information about the VPC. You can click ↻ to refresh the options in the **VPC** drop-down list. **NOTE** ● You can**create a share** to share VPC resources with other members. After **responding to a resource sharing invitation**, the members can select the shared VPC resources. For details, see **How Do I Use VPC Sharing to Process GaussDB(DWS) Resources?**	vpc-dws
Subnet	Specify a VPC subnet. A subnet provides dedicated network resources that are isolated from other networks, improving network security. **NOTE** After a cluster is created, the subnet cannot be modified. If you need to modify the subnet, you can restore the snapshot of the cluster to a new cluster. The data of the new cluster is the same as that of the old cluster, and the subnet can be modified when the new cluster is created.	subnet-dws

Parameter	Description	Example Value
Security Group	Specify a VPC security group. A security group restricts access rules to enhance security when GaussDB(DWS) and other services access each other. ● Automatic creation If **Automatic creation** is selected, the system automatically creates a default security group. This option is selected by default. The rule of the default security group is as follows: The outbound allows all access requests, while the inbound is open only to the database port that you set to connect to the GaussDB(DWS) cluster. The format of the default security group name is dws-<*Cluster_name*>-<*Cluster_database_port*>, for example, **dws-dws-demo-8000**. **NOTE** If the quotas of the security group and the security group rule are insufficient, an error message will be displayed after you submit the cluster creation application. Select an existing group and retry. ● Manual creation You can also log in to the **VPC management console** to manually create a security group. Then, go back to the page for creating data warehouse clusters, click ↻ next to the **Security Group** drop-down list to refresh the page, and select the new security group. To enable the GaussDB(DWS) client to connect to the cluster, you need to add an inbound rule to the new security group to grant the access permission to the database port of the GaussDB(DWS) cluster. The following is an example of an inbound rule. For details, see **Adding an Inbound Rule**. – **Protocol**: **TCP**. – **Port**: **8000**. Use the database port set when creating the GaussDB(DWS) cluster. This port is used for receiving client connections to GaussDB(DWS). – **Source**: Select **IP address** and use the host IP address of the client host, for example, **192.168.0.10/32**.	Automatic creation

Parameter	Description	Example Value
	After a GaussDB(DWS) cluster is created, you can change the security group. You can also add, delete, or modify security group rules in the current security group. For details, see **Modifying a Security Group**. Changing the security group of a cluster may cause brief service disruption. Exercise caution when performing this operation. For better network performance, do not select more than five security groups.	

Parameter	Description	Example Value
EIP	Specify whether users can use a client to connect to a cluster's database over the Internet. The following methods are supported: ● **Do not use**: Do not specify any EIPs here. If GaussDB(DWS) is used in the production environment, first bind it to ELB, and then bind it to an EIP on the ELB page. ● **Buy now**: Specify the EIP bandwidth, and an EIP with dedicated bandwidth will be bound to the cluster. The EIP can be used to access the cluster over the Internet. The bandwidth name of an automatically assigned EIP starts with the cluster name. ● **Specify**: A specified EIP is bound to the cluster. If no available EIPs are displayed in the drop-down list, click **View EIP** to go to the EIP page and create one that meets your needs. You can set the IP address type and bandwidth as required. **NOTE** ● In yearly/monthly billing mode, you cannot buy an EIP during cluster creation. ● If you use the EIP binding function for the first time in each project of each region, the system prompts you to create the **DWSAccessVPC** agency to authorize GaussDB(DWS) to access VPC. After the authorization is successful, GaussDB(DWS) can switch to a healthy VM when the VM bound with the EIP becomes faulty. ● By default, only Huawei Cloud accounts or users with Security Administrator permissions can query and create agencies. By default, the IAM users in those accounts cannot query or create agencies. When the users use the EIP, the system makes the binding function unavailable. Contact a user with the **DWS Administrator** permissions to authorize the agency on the current page. ● **Do not use** indicates disabling access to the cluster over the public network. After a cluster is created, if you want to access it over the public network, bind an EIP to the cluster and create a public network domain name. For details, see **Creating a Public Network Domain Name**. ● If GaussDB(DWS) is used for the production environment, the new GaussDB(DWS) cluster needs to be bound to ELB and then to EIP. Select **Do not use** here.	Buy now
Bandwidth	When **EIP** is set to **Buy now**, you need to specify the bandwidth of the EIP. The value ranges from 1 Mbit/s to 100 Mbit/s.	50 Mbit/s

Parameter	Description	Example Value
ELB	Specifies whether ELB is bound. With ELB health checks, CN requests of a cluster can be quickly forwarded to normal CNs. If a CN is faulty, the workload can be immediately shifted to a healthy node, minimizing cluster access faults. Currently, ELBs can be bound in the same VPC or across VPCs. ● **Do not use**: The load balancer is not used. If GaussDB(DWS) is used in the production environment, first bind it to ELB, and then bind it to an EIP on the ELB page. ● **Specify**: Specify an ELB to be bound to the cluster. If no available ELBs are displayed in the drop-down list, click **Create ELB** to go to the ELB page and create one as needed. **WARNING** Configure load balancing to ensure load balancing and high availability of the cluster and prevent service interruptions. You are not advised to directly connect services to a single CN.	Specify

Step 8 Click **Next: Configure Advanced Settings**.

Step 9 Configure cluster parameters.

Table 3-3 Cluster parameters

Parameter	Description	Example Value
Cluster Name	Name of the data warehouse cluster. Enter 4 to 64 characters. Only letters (case-insensitive), digits, hyphens (-), and underscores (_) are allowed. The name must start with a letter. **NOTE** If the cluster name cannot be changed on the console, contact technical support.	DWS-demo
Cluster Version	Version of the database instance installed in the cluster. The example version number is for reference only.	-
Default Database	The default database name of the cluster is **gaussdb**. **NOTE** This name cannot be changed.	gaussdb

Parameter	Description	Example Value
Administrator Account	Database administrator name. The name must: ● Consist of lowercase letters, digits, or underscores. ● Start with a lowercase letter or an underscore. ● Contain 6 to 64 characters. ● Cannot be a keyword of the GaussDB(DWS) database. For details about the keywords of the GaussDB(DWS) database, see **Keyword** in the *Data Warehouse Service (DWS) Developer Guide*.	dbadmin
Administrator Password	Password of the database administrator account. The password must: ● Contain 12 to 32 characters. ● Cannot be the username or the username spelled backwards. ● Contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-) ● Pass the weak password check. **NOTE** Change the password regularly and keep it secure.	-
Confirm Password	Enter the database administrator password again.	-
Database Port	Set the port used when the client or application connects to the database in the cluster. The port number ranges from 8000 to 30000. **NOTE** The database port of a created cluster cannot be changed. You can specify the database port only when creating a cluster.	8000

Parameter	Description	Example Value
IPv6	Specify whether to enable the IPv6 dual stack for the cluster. If this function is enabled, a client or application can connect to the database using an IPv6 address. **NOTE** To enable IPv6, the following conditions must be met: ● The subnet configured in **Step 7** is an IPv6 dual-stack subnet. ● The cluster supports IPv6 addresses and a maximum of three NICs. ● The cluster version must be 8.2.1.210 or later.	-
Time Zone	You can set the time zone for the tenant cluster, including the system OS time zone and cluster data warehouse time zone.	-

Step 10 Select the enterprise project of the cluster. You can configure this parameter only when the Enterprise Project Management service is enabled. The default value is **default**.

An enterprise project facilitates project-level management and grouping of cloud resources and users.

You can select the default enterprise project **default** or other existing enterprise projects. To create an enterprise project, log in to the Enterprise Management console. For details, see the *Enterprise Management User Guide*.

Step 11 Configure advanced parameters. Select **Default** to keep the default values of the advanced parameters. You can also select **Custom** to modify the values.

- **Backup Device**

  Set the backup device used by the current cluster. For details about the parameter configuration principles, see **Table 3-4**.

Table 3-4 Automated snapshot parameters

Parameter	Description
Backup Device	Select **OBS** or **NFS** from the drop-down list.
NFS Backup File System Address (NFS)	NFS shared IP address. To mount the SFS shared path, enter its IP address. If successful, a mount directory will be created in the **/var/chroot/nfsbackup** directory of the cluster instance.

- **CNs**

CNs, or Coordinators, receive access requests from the clients and return the execution results. They also split and distribute tasks to the Datanodes (DNs) for parallel execution.

The value ranges from 3 to the number of cluster nodes. The maximum value is **20** and the default value is **3**. In a large-scale cluster, you are advised to deploy multiple CNs.

- **Tag**

  A tag is a key-value pair used to identify a cluster. For details about the keys and values, see **Table 3-5**. By default, no tag is added to the cluster.

  If your organization has configured GaussDB(DWS) tag policies, you need to add tags to clusters based on the tag policies. If a tag does not comply with the tag policies, cluster creation may fail. Contact your organization administrator to learn more about tag policies.

  For details about tags, see **Overview**.

Table 3-5 Tag parameters

Parameter	Description	Example Value
Tag key	The options are as follows: – Select a predefined tag key or an existing resource tag key from the drop-down list of the text box. NOTE To add a predefined tag, you need to create one on TMS and select it from the drop-down list of **Tag key**. You can click **View predefined tags** to enter the **Predefined Tags** page of TMS. Then, click **Create Tag** to create a predefined tag. For more information, see section **Creating Predefined Tags** in the *Tag Management Service User Guide*. – Enter a tag key in the text box. A tag key can contain a maximum of 128 characters. It cannot be an empty string or start or end with a space. The value cannot contain the following characters: *<> \,|/ NOTE A key must be unique in a given cluster.	key01
Value	You can: – Select a predefined tag value or resource tag value from the drop-down list of the text box. – Enter a tag value in the text box. A tag value can contain a maximum of 255 characters, which can be an empty string. It cannot start or end with a space. The value cannot contain the following characters: *<> \,|/	value01

- **Encrypt DataStore**

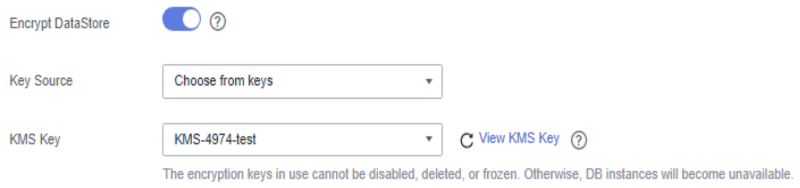
  If this function is enabled, Key Management Service (KMS) encrypts the cluster and the cluster's snapshot data.

When you enable database encryption for each project in each region for the first time, the system displays a **Create Agency** dialog box. Click **Yes** to create **DWSAccessKMS** to authorize GaussDB(DWS) to access KMS. If you click **No**, the encryption function is not enabled. Select the created KMS key from the **KMS Key Name** drop-down list. If no key is available, you can log in to the KMS console to create one. For details, see **Data Encryption Workshop User Guide**.

NOTICE

- Only users with the Tenant Admin permission can view and toggle the **Encrypt DataStore** switch.
- By default, only Huawei Cloud accounts or users with **Security Administrator** permissions can query and create agencies. IAM users under an account do not have the permission to query or create agencies by default. Contact a user with that permission and complete the authorization on the current page.
- The database encryption function cannot be disabled once it is enabled.
- After **Encrypt DataStore** is enabled, the key cannot be disabled, deleted, or frozen when being used. Otherwise, the cluster becomes abnormal and the database becomes unavailable.
- After database encryption is enabled, you cannot use open APIs to restore created snapshots.

- Method 1: Select a key name. You can **create a resource share** to share KMS resources with other members. After **accepting the sharing invitation**, members can select the shared KMS resource from the key source.

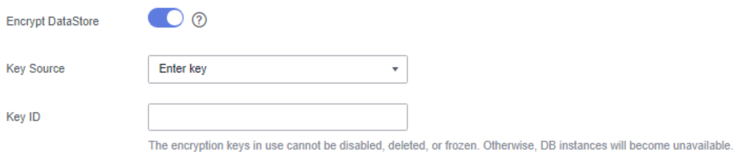


Encrypt DataStore: enabled
Key Source: Choose from keys
KMS Key: KMS-4974-test

The encryption keys in use cannot be disabled, deleted, or frozen. Otherwise, DB instances will become unavailable.

- Method 2: Enter the key ID. Enter the key ID used for authorizing the current tenant. For details, see **Viewing a CMK**.

  When you grant permissions on the **Creating a Grant** page, the authorized object must be an account instead of a user. The authorized operations must at least contain **Querying key details**, **Encrypting data**, and **Decrypting data**.



Encrypt DataStore: enabled
Key Source: Enter key
Key ID:
The encryption keys in use cannot be disabled, deleted, or frozen. Otherwise, DB instances will become unavailable.

Step 12 Click **Next: Confirm**.

NOTE

If the number of requested nodes, vCPU (cores), or memory (GB) exceed the user's remaining quota, a warning dialog box is displayed, indicating that the quota is insufficient and displaying the detailed remaining quota and the current quota application. You can click **Increase quota** in the warning dialog box to submit a service ticket and apply for higher node quota. Once approved, we will update your resource quota accordingly and send you a notification. For details about quota operations, see **Quotas**.

Step 13 Select a billing mode. If you select the yearly/monthly mode, you also need to configure the service duration.

Table 3-6 Duration

Parameter	Description
Required Duration (Yearly/Monthly)	Configure the required duration. You get a greater discount if you purchase a longer period. **Price** is displayed at the bottom of the page for your reference. You can click **Pricing details** to view the detailed price.
Auto-renewal (Yearly/Monthly)	● By default, this option is not selected. ● Renewal rules: – Monthly subscriptions are renewed for a month each time. – Yearly subscriptions are renewed for a year each time. Example: Customer A purchases a cluster in yearly/monthly mode and select enables auto-renewal. If the cluster is subscribed to for eight months, it will be automatically renewed each month. If the cluster is subscribed to for two years, it will be automatically renewed each year. For details about the renewal fee deduction, see **Fee Deduction Rules**.

Step 14 Click **Buy Now**. If the billing mode is yearly/monthly billing, click **Buy Now**. The payment page is displayed.

After the submission is successful, the creation starts. Click **Back to Cluster List**. The cluster management page is displayed. The initial status of the cluster is **Creating**. Cluster creation takes some time. Wait for a while. Clusters in the **Available** state are ready for use.

NOTE

- For load balancing and high availability purposes, and to prevent single CN failures, a cluster must be bound to ELB. For details, see **Binding and Unbinding ELBs for a GaussDB(DWS) Cluster**.

----End

Handling the Cluster Creation Failure

If a cluster fails to be created, you can go to the **Clusters** > **Dedicated Clusters** page of the GaussDB(DWS) console to view the cluster status and the cause of failure.

Checking the failure cause

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, locate the cluster whose **Cluster Status** is **Creation failed**.

Step 3 Click ? in the **Cluster Status** column to view the cause of the cluster creation failure.

If the fault persists, contact technical support.

----End

Deleting a cluster that fails to be created

You can delete a cluster that fails to be created if you do not need it. Before deletion, check the cause of creation failure.

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, locate the row containing the failed cluster to be deleted, and choose **More** > **Delete**.

Step 3 In the displayed dialog box, confirm the deletion. You can determine whether to perform the following operations:

- Create a snapshot for the cluster.

  If the cluster status is normal, click **Create Snapshot**. On the snapshot list page, click **Create Snapshot** to create a snapshot for the cluster to be deleted. For details, see **Manual Snapshots**. In the row of a cluster, choose **More** > **Delete**.

- Delete associated resources.
  - Release the EIP bound to the cluster.

    If an EIP is bound to the cluster, you are advised to select **EIP** to release the EIP. If you do not release the EIP, you can bind it to another cluster or cloud resource and it will be billed based on the EIP pricing rule of VPC.

  - Delete automated snapshots.
  - Delete manual snapshots.

    If you have created a manual snapshot, you can select **Manual Snapshot** to delete it.

Step 4 After confirming that the information is correct, enter **DELETE** or click **Auto Enter** and click **OK** to delete the cluster. The cluster status in the cluster list will change to **Deleting** and the cluster deletion progress will be displayed.

If the cluster to be deleted uses an automatically created security group that is not used by other clusters, the security group is automatically deleted when the cluster is deleted.

----End

3.1.2 Creating a GaussDB(DWS) Storage-Compute Decoupled Cluster

The storage-compute decoupled cluster uses the cloud-native and cost-effective architecture. It supports hot and cold data analysis, elastic scaling of storage and compute resources, unlimited computing power and capacity, and pay-per-use pricing. It is suitable for OLAP analysis scenarios.

This section describes how to create a storage-compute decoupled cluster on the GaussDB(DWS) console.

⚠ WARNING

- You are advised not to use clusters with low specifications, such as clusters with 16 GB memory and 4-core vCPUs, in the production environment. Otherwise, resource overload may occur.
- Configure **load balancing** to balance the connections to each CN and high availability of the cluster and prevent service interruptions. You are not advised to directly connect services to a single CN.

Preparations Before Creating a Cluster

- You have evaluated the flavor of cluster nodes.

  You can select the number of nodes by data volume, service load, and performance. More nodes bring you stronger storage and compute capabilities.

  When first using GaussDB(DWS), you can create a cluster with a smaller flavor. Then, you can adjust the cluster scale and node flavor based on the data volume and service load changes without interrupting services. For details, see **Scaling Out a Cluster**.
- Determine the number of nodes that can be used by users.

  Ensure that the number of available nodes is greater than or equal to 3. Otherwise, the system displays a message indicating that the cluster cannot be created. You can choose **Clusters** > **Dedicated Clusters** to view the number of available nodes.

Creating a Cluster

Step 1 Go to the **page for creating a GaussDB(DWS) cluster**.

Step 2 Select a billing mode. For more information, see **Pricing Details**.

- Yearly/Monthly: If you select **Yearly/Monthly**, you need to set the required duration in **12** before proceeding with the following steps.
- Pay-per-use (hourly): If you select this mode, you are charged based on the actual usage duration (accurate to minutes).

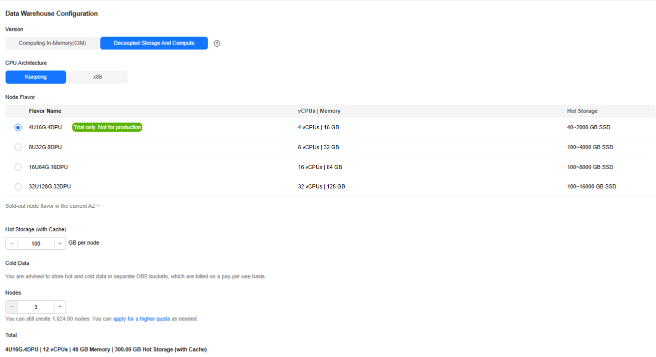
Step 3 Select a region and an AZ.

Step 4 Configure **Resource**, **CPU Architecture**, and **Node Flavor**.

NOTE

- The number of nodes in a new cluster cannot exceed the quota that can be used by a user or 256. If the node quota is insufficient, click **Increase quota** to submit a service ticket and apply for higher node quota.
- If you have yearly/monthly nodes that meet service requirements, you are advised to use these nodes first to save costs. You can select **Yearly/Monthly** for **Billing Mode**.

Figure 3-2 Configuring node parameters



Data Warehouse Configuration

Version

Version: Decoupled Storage And Compute

CPU Architecture

CPU Architecture: Kunpeng

Node Flavor

	Flavor Name	vCPUs | Memory	Hot Storage
●	4U16G.4DPU Trial only. Not for production	4 vCPUs | 16 GB	40~2000 GB SSD
○	8U32G.8DPU	8 vCPUs | 32 GB	100~4000 GB SSD
○	16U64G.16DPU	16 vCPUs | 64 GB	100~8000 GB SSD
○	32U128G.32DPU	32 vCPUs | 128 GB	100~16000 GB SSD

Sold-out node flavor in the current AZ

Hot Storage (with Cache): 100 GB per node

Cold Data

You are advised to store hot and cold data in separate OBS buckets, which are billed on a pay-per-use basis.

Nodes: 3

You can still create 1,024.00 nodes. You can apply for a higher quota as needed.

Total

4U16G.4DPU | 12 vCPUs | 48 GB Memory | 300.00 GB Hot Storage (with Cache)

Table 3-7 Node configuration parameters

Parameter	Description	Example Value
Resource	The options are as follows: **Decoupled Storage and Compute**: The storage-compute decoupled data warehouse is designed with a cloud native architecture that separates storage and compute. It also features hierarchical auto scaling for computing and storage, as well as multi-logical cluster shared storage technology (Virtual Warehouse or VW). These capabilities allow for computing isolation and concurrent expansion to handle varying loads, making it an ideal choice for OLAP analysis scenarios.	-
CPU Architecture	The following CPU architectures can be selected: ● **x86** ● **Kunpeng** NOTE The x86 and Kunpeng architectures differ only in their underlying structure, which is not sensible to the application layer. Both architectures use the same SQL syntax. If you need to create a cluster and find that x86 servers are not enough, you can opt for the Kunpeng architecture.	-

Parameter	Description	Example Value
Node Flavor	Select a node flavor. Each node flavor shows the vCPU, memory, and recommended application scenario. For more information about the node flavors supported by GaussDB(DWS) and their prices, see the **GaussDB(DWS) pricing details**. For details about the node flavors supported by GaussDB(DWS), see **Data Warehouse Specifications**.	-
Hot Storage (with Cache)	Available storage capacity of each node. **NOTE** ● The storage capacity you apply for has the necessary file system overhead, which includes index nodes and the space required for database running. ● The displayed **200GB/node** includes the storage for cache. For example, if you create 3 nodes, each having 200 GB capacity, the total resource capacity is 600 GB, and the actual storage space available to you is 300 GB.	-
Cold Data	Store data in separate OBS buckets, which are billed on a pay-per-use basis.	-
Nodes	Specify the number of nodes in the cluster. The number of nodes ranges from 3 to 256.	3
Total	Display the cluster's total capacity. The storage capacity of each flavor includes the storage for cache. The displayed storage capacity includes the disk space consumed by backups and RAIDs.	-

Step 5 Click **Next: Configure Network**.

Step 6 Configure the network.

Table 3-8 Network parameters

Parameter	Description	Example Value
VPC	Specify a VPC to isolate the cluster's network. If you create a data warehouse cluster for the first time and have not configured the VPC, click **View VPC**. On the VPC management console that is displayed, create a VPC as needed. For details about how to create a VPC, see **Creating a VPC** in the *Virtual Private Cloud User Guide*. After selecting a VPC from the drop-down list, click **View VPC** to enter the VPC management console and view the detailed information about the VPC. You can click ↻ to refresh the options in the **VPC** drop-down list. **NOTE** You can**create a share** to share VPC resources with other members. After **responding to a resource sharing invitation**, the members can select the shared VPC resources. For details, see **How Do I Use VPC Sharing to Process GaussDB(DWS) Resources?**	vpc-dws
Subnet	Specify a VPC subnet. A subnet provides dedicated network resources that are isolated from other networks, improving network security.	subnet-dws

Parameter	Description	Example Value
Security Group	Specify a VPC security group. A security group restricts access rules to enhance security when GaussDB(DWS) and other services access each other. ● Automatic creation If **Automatic creation** is selected, the system automatically creates a default security group. This option is selected by default. The rule of the default security group is as follows: The outbound allows all access requests, while the inbound is open only to the database port that you set to connect to the GaussDB(DWS) cluster. The format of the default security group's name is dws-<*cluster name*>-<*database port of the GaussDB(DWS) cluster*>, for example, **dws-dws-demo-8000**. **NOTE** If the quotas of the security group and the security group rule are insufficient, an error message will be displayed after you submit the cluster creation application. Select an existing group and retry. ● Manual creation You can also log in to the **VPC management console** to manually create a security group. Then, go back to the page for creating data warehouse clusters, click ↻ next to the **Security Group** drop-down list to refresh the page, and select the new security group. To enable the GaussDB(DWS) client to connect to the cluster, you need to add an inbound rule to the new security group to grant the access permission to the database port of the GaussDB(DWS) cluster. The following is an example of an inbound rule. For details, see **Adding an Inbound Rule**. – **Protocol**: **TCP**. – **Port**: **8000**. Use the database port set when creating the GaussDB(DWS) cluster. This port is used for receiving client connections to GaussDB(DWS). – **Source**: Select **IP address** and use the host IP address of the client host, for example, **192.168.0.10/32**.	Automatic creation

Parameter	Description	Example Value
	After a GaussDB(DWS) cluster is created, you can change the security group. You can also add, delete, or modify security group rules in the current security group. For details, see **Modifying a Security Group**.	

Parameter	Description	Example Value
EIP	Specify whether users can use a client to connect to a cluster's database over the Internet. The following methods are supported: ● **Do not use**: Do not specify any EIPs here. If GaussDB(DWS) is used in the production environment, first bind it to ELB, and then bind it to an EIP on the ELB page. ● **Buy now**: Specify the EIP bandwidth, and an EIP with dedicated bandwidth will be bound to the cluster. The EIP can be used to access the cluster over the Internet. The bandwidth name of an automatically assigned EIP starts with the cluster name. ● **Specify**: Specify an EIP to be bound to the cluster. If no available EIPs are displayed in the drop-down list, click **View EIP** to go to the EIP page and create one that meets your needs. You can set the IP address type and bandwidth as required. **NOTE** ● In yearly/monthly billing mode, you cannot buy an EIP during cluster creation. ● If you use the EIP binding function for the first time in each project of each region, the system prompts you to create the **DWSAccessVPC** agency to authorize GaussDB(DWS) to access VPC. After the authorization is successful, GaussDB(DWS) can switch to a healthy VM when the VM bound with the EIP becomes faulty. ● By default, only cloud accounts or users with **Security Administrator** permissions can query and create agencies. By default, the IAM users in those accounts cannot query or create agencies. When the users use the EIP, the system makes the binding function unavailable. Contact a user with the **DWS Administrator** permissions to authorize the agency on the current page. ● **Do not use** indicates disabling access to the cluster over the public network. After a cluster is created, if you want to access it over the public network, bind an EIP to the cluster and create a public network domain name. For details, see **Creating a Public Network Domain Name**. ● If GaussDB(DWS) is used for the production environment, the new GaussDB(DWS) cluster needs to be bound to ELB and then to EIP. Select **Do not use** here.	Buy now
Bandwidth	When **EIP** is set to **Buy now**, you need to specify the bandwidth of the EIP. The value ranges from 1 Mbit/s to 100 Mbit/s.	50 Mbit/s

Parameter	Description	Example Value
ELB	Specifies whether ELB is bound. With ELB health checks, CN requests of a cluster can be quickly forwarded to normal CNs. If a CN is faulty, the workload can be immediately shifted to a healthy node, minimizing cluster access faults. Currently, ELBs can be bound in the same VPC or across VPCs. ● **Do not use**: The load balancer is not used. If GaussDB(DWS) is used in the production environment, first bind it to ELB, and then bind it to an EIP on the ELB page. ● **Specify**: Specify an ELB to be bound to the cluster. If no available ELBs are displayed in the drop-down list, click **Create ELB** to go to the ELB page and create one as needed. **WARNING** Configure load balancing to ensure load balancing and high availability of the cluster and prevent service interruptions. You are not advised to directly connect services to a single CN.	Specify

Step 7 Click **Next: Configure Advanced Settings**.

Step 8 Configure cluster parameters.

Table 3-9 Cluster parameters

Parameter	Description	Example Value
Cluster Name	Name of the data warehouse cluster. Enter 4 to 64 characters. Only letters (case-insensitive), digits, hyphens (-), and underscores (_) are allowed. The name must start with a letter. **NOTE** If the cluster name cannot be changed on the console, contact technical support.	DWS-demo
Cluster Version	Version of the database instance installed in the cluster. The example version number is for reference only.	9.0.0
Default Database	The default database name of the cluster is **gaussdb**. **NOTE** This name cannot be changed.	gaussdb

Parameter	Description	Example Value
Administrator Account	Database administrator name. The name must: ● Consist of lowercase letters, digits, or underscores. ● Start with a lowercase letter or an underscore. ● Contain 6 to 64 characters. ● Cannot be a keyword of the GaussDB(DWS) database. For details about the keywords of the GaussDB(DWS) database, see **Keyword** in the *Data Warehouse Service (DWS) Developer Guide*.	dbadmin
Administrator Password	Password of the database administrator account. The password must: ● Contain 12 to 32 characters. ● Cannot be the username or the username spelled backwards. ● Contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-) ● Pass the weak password check. **NOTE** Change the password regularly and keep it secure.	-
Confirm Password	Enter the database administrator password again.	-
Database Port	Set the port used when the client or application connects to the database in the cluster. The port number ranges from 8000 to 30000.	8000
IPv6	Specify whether to enable the IPv6 dual stack for the cluster. If this function is enabled, a client or application can connect to the database using an IPv6 address. **NOTE** To enable IPv6, the following conditions must be met: ● The subnet configured in **Step 7** is an IPv6 dual-stack subnet. ● The cluster supports IPv6 addresses and a maximum of three NICs. ● The cluster version must be 8.2.1.210 or later.	-
Time Zone	You can set the time zone for the tenant cluster, including the system OS time zone and cluster data warehouse time zone.	-

Step 9 Select the enterprise project of the cluster. You can configure this parameter only when the Enterprise Project Management service is enabled. The default value is **default**.

An enterprise project facilitates project-level management and grouping of cloud resources and users.

You can select the default enterprise project **default** or other existing enterprise projects. To create an enterprise project, log in to the Enterprise Management console. For details, see **Enterprise Management User Guide**.

Step 10 Configure advanced parameters. Select **Default** to keep the default values of the advanced parameters. You can also select **Custom** to modify the values.

- **Backup Device**

  Set the backup device used by the current cluster. For details about the parameter configuration principles, see **Table 3-10**.

  Table 3-10 Automated snapshot parameters

Parameter	Description
Backup Device	Select **OBS** or **NFS** from the drop-down list.
NFS Backup File System Address (NFS)	NFS shared IP address. To mount the SFS shared path, enter its IP address. If successful, a mount directory will be created in the **/var/chroot/nfsbackup** directory of the cluster instance.

- **CNs**

  CNs receive access requests from the clients and return the execution results. In addition, a CN splits and distributes tasks to the DNs for parallel execution.

  The value ranges from 3 to the number of cluster nodes. The maximum value is **20** and the default value is **3**. In a large-scale cluster, you are advised to deploy multiple CNs.

- **Tag**

  A tag is a key-value pair used to identify a cluster. For details about the keys and values, see **Table 3-11**. By default, no tag is added to the cluster.

  If your organization has configured GaussDB(DWS) tag policies, you need to add tags to clusters based on the tag policies. If a tag does not comply with the tag policies, cluster creation may fail. Contact your organization administrator to learn more about tag policies.

  For details about tags, see **Overview**.

Table 3-11 Tag parameters

Parameter	Description	Example Value
Tag key	You can: – Select a predefined tag key or an existing resource tag key from the drop-down list of the text box. **NOTE** To add a predefined tag, you need to create one on TMS and select it from the drop-down list of **Tag key**. You can click **View predefined tags** to enter the **Predefined Tags** page of TMS. Then, click **Create Tag** to create a predefined tag. For more information, see section **Creating Predefined Tags** in *Tag Management Service User Guide*. – Enter a tag key in the text box. A tag key can contain a maximum of 128 characters. It cannot be an empty string or start or end with a space. The value cannot contain the following characters: *<> \\,|/ **NOTE** A key must be unique in a given cluster.	key01
Value	You can: – Select a predefined tag value or resource tag value from the drop-down list of the text box. – Enter a tag value in the text box. A tag value can contain a maximum of 255 characters, which can be an empty string. It cannot start or end with a space. The value cannot contain the following characters: *<> \\,|/	value01

Step 11 Select a billing mode. If you select the yearly/monthly mode, you also need to configure the service duration.

Table 3-12 Duration

Parameter	Function
Required Duration (Yearly/Monthly)	Configure the required duration. You get a greater discount if you purchase a longer period. **Price** is displayed at the bottom of the page for your reference. You can click **Pricing details** to view the detailed price.

Parameter	Function
Auto-renewal (Yearly/Monthly)	● By default, this option is not selected. ● Renewal rules: – Monthly subscriptions are renewed for a month each time. – Yearly subscriptions are renewed for a year each time. Example: Customer A purchases a cluster in yearly/monthly mode and select enables auto-renewal. If the cluster is subscribed to for eight months, it will be automatically renewed each month. If the cluster is subscribed to for two years, it will be automatically renewed each year. For details about the renewal fee deduction, see **Fee Deduction Rules**.

Step 12 Click **Next: Confirm**.

NOTE

If the number of requested nodes, vCPU (cores), or memory (GB) exceed the user's remaining quota, a warning dialog box is displayed, indicating that the quota is insufficient and displaying the detailed remaining quota and the current quota application. You can click **Increase quota** in the warning dialog box to submit a service ticket and apply for higher node quota. Once approved, we will update your resource quota accordingly and send you a notification. For details about quota operations, see **Quotas**.

Step 13 Click **Buy Now**. If the billing mode is yearly/monthly billing, click **Buy Now**. The payment page is displayed.

After the submission is successful, the creation starts. Click **Back to Cluster List**. The cluster management page is displayed. The initial status of the cluster is **Creating**. Cluster creation takes some time. Wait for a while. Clusters in the **Available** state are ready for use.

----End

3.2 Before You Start: Performance Management Requirements

Effective performance management of the GaussDB(DWS) database system is vital for the entire system. To prevent frequent resource overload (such as CPU, I/O, memory, and disk space) in the cluster, it is important to control and limit the services and overall resources in the cluster. Regular proactive O&M and advance scale-out planning are also necessary.

Before introducing a new service, it is crucial to evaluate and conduct pressure tests on existing resources to avoid excessive resource consumption and negative impact on the overall cluster performance. As the data volume of existing services increases, the cluster's disk space and I/O usage also grow. Therefore, periodic clearance of aged and unnecessary data is required.

This section provides an overview of the cluster's performance baseline and outlines the performance management requirements in typical service scenarios.

Its purpose is to assist users and O&M personnel in evaluating the cluster's capacity in advance and preventing resource overload.

GaussDB(DWS) Cluster Performance Baseline

In this section, you will find information about the recommended values and risk values of GaussDB(DWS) resources.

When the resource watermark exceeds the recommended value, it is crucial for O&M personnel to promptly address the issue to prevent performance degradation in scenarios such as node faults and active/standby switchover.

Exceeding the risk value for the cluster resource watermark indicates potential overload. In such cases, it is advisable to refrain from introducing new services.

Instead, it is necessary to swiftly reduce the overall cluster load through service optimization or scheduling tasks during off-peak hours. If needed, the cluster can be divided or its capacity expanded to ensure no impact on overall performance.

Table 3-13 Cluster Performance and Capacity Risks and Suggestions

Metric	Recommended Value	Impact of Exceeding the Recommended Value	Recommended Measure	Risk Value	Impact of Exceeding the Risk Value	Recommended Measure
CPU usage	Less than 60%	When the active/standby nodes are unbalanced or a node is faulty, the CPU usage of some nodes may be overloaded, causing performance degradation.	Configure a resource pool for resource isolation. For details, see **GaussDB(DWS) Resource Load Management**. Use **Real-Time Queries** and **Performance Monitoring** to capture statements with high CPU usage for service optimization. For details, see **Monitoring and Diagnosing Top SQL Statements in a GaussDB(DWS) Cluster** and .	80 %	Severe CPU contention occurs. As a result, the execution time of operators such as Stream deteriorates, and the overall cluster performance is severely affected.	Reduce the CPU load during peak hours by means of service staggering, service splitting, service optimization, and cluster scale-out. You can also set the CPU limit and quota of the resource pool. For details, see advanced system tuning operations in **Tuning Systems with High CPU Usage**.

Metric	Recommended Value	Impact of Exceeding the Recommended Value	Recommended Measure	Risk Value	Impact of Exceeding the Risk Value	Recommended Measure
CPU skew	Less than 15%	Computing skew occurs. As a result, the optimal performance of some statements in the distributed system cannot be fully utilized.	Configure rules introducing in **Exception Rules** and circuit breakers to fallbreak skew statements in advance. Optimize such services on a daily basis.	30 %	During peak hours, a single node's CPU may become overloaded, causing overall cluster performance to deteriorate due to Liebig's Law of the Minimum. This prevents other nodes from being fully utilized.	Configure rules introducing in **Exception Rules** and circuit breakers to preemptively handle skewed statements and optimize services regularly.

Metric	Recommended Value	Impact of Exceeding the Recommended Value	Recommended Measure	Risk Value	Impact of Exceeding the Risk Value	Recommended Measure
I/O usage	Less than 60%	When the active/standby status is unbalanced or a node fails, some nodes may experience I/O overload, leading to performance degradation.	Find out the services with high I/O usage by checking the monitoring data. For details, see **Performance Monitoring**. You can reduce the disk I/O usage by indexing, partition pruning, and row-column storage rectification.	90%	Severe I/O contention can occur, affecting operators such as table scanning and overall cluster performance.	Optimize high-I/O statements and stagger peak hours to maintain I/O performance. Plan for cluster scale-out in advance to reduce the I/O burden on individual nodes.
I/O read/write latency	Less than 400 milliseconds	Performance fluctuations during data read and write operations can lead to unstable query times and occasional performance degradation.	Find out the services with high I/O usage by checking the monitoring data. For details, see **Performance Monitoring**. You can reduce the disk I/O usage by indexing, partition pruning, and row-column storage rectification to reduce the read/write latency.	1000 ms	Significant deterioration in data read/write performance can cause real-time data storage services to back up, impacting overall performance.	Optimize high-I/O, high-disk, and high-concurrency statements to stagger service peaks and distribute the load more evenly.

Metric	Recommended Value	Impact of Exceeding the Recommended Value	Recommended Measure	Risk Value	Impact of Exceeding the Risk Value	Recommended Measure
Dynamic memory usage	Less than 80%	When the service traffic increases sharply or complex flexible queries are executed, an error may be reported due to insufficient memory.	Configure exception rules and memory circuit breaker. Optimize memory-intensive services by referring to **Real-Time Queries** and **Monitoring and Diagnosing Top SQL Statements in a GaussDB(DWS) Cluster**. For how to reduce the memory usage, see **Reducing Memory Usage**.	90 %	CCN queuing occurs, an error indicating insufficient memory is reported, and process OOM risks exist.	Configure exception rules and memory circuit breaker. Optimize memory-intensive services by referring to **Real-Time Queries** and **Monitoring and Diagnosing Top SQL Statements in a GaussDB(DWS) Cluster**.
Disk space usage	Less than 70%	The risk of read-only status increases when SQL statements are written to disks and the disk usage exceeds 90%.	Set thresholds for triggering disk flushing, clear data and dirty pages during off-peak hours, and plan for scale-out in advance. For details, see **Solution to High Disk Usage and Cluster Read-Only**.	80 %	The read-only risk increases after SQL statements are written to disks.	Set thresholds for triggering disk flushing, clear data and dirty pages during off-peak hours, and plan for scale-out in advance.

Metric	Recommended Value	Impact of Exceeding the Recommended Value	Recommended Measure	Risk Value	Impact of Exceeding the Risk Value	Recommended Measure
Disk space skew	Less than 15%	Severe skew occurs during operator computing or data spill to the disk. The workloads will be unevenly distributed on DNs, resulting in high disk usage on a single DN and affecting performance.	Check and handle table skew by referring to **Table Diagnosis**.	20%	Disk skew causes CPU, I/O, and memory skew, which affects the overall cluster performance and may cause the disk of a single DN to be full.	Handle table skew by referring to **Table Diagnosis**.

GaussDB(DWS) Performance Management Scenarios and Suggestions

This section introduces common performance management scenarios and offers suggestions. During service rollout and routine O&M, you need to thoroughly assess the performance capacity to avoid overloading the cluster.

Table 3-14 Performance management scenarios

Scenario	Performance Risk	Evaluation Method	Suggestion
New cluster rollout	The performance and capacity of the new cluster are uncertain before the service rollout, and there is a possibility that they may not meet the requirements.	Before launching the service, conduct a pressure test on the cluster. Both the new and old clusters should be operational for at least one service period. It is necessary to thoroughly test key services and links for performance metrics such as QPS, latency, maximum concurrency, and maximum response time. This will ensure a comprehensive evaluation of the performance and capacity of the new cluster.	Implement dynamic resource management and allocate service resource pools accordingly by referring to **GaussDB(DWS) Resource Load Management**. Configure exception rules in advance and configure circuit breaker parameters.
New service rollout	Resource preemption may arise, impacting existing services in the cluster. If new services are executed concurrently and consume resources improperly, it can result in resource overload and a decline in overall performance.	Conduct a thorough test on the new service in a test environment. Based on the test results, estimate the CPU usage, execution time, and number of concurrent services. Analyze the execution plan for the new services to ensure optimal performance.	Roll out a cluster only when the cluster's performance capacity is sufficient. Isolate new services with resource pools. Configure circuit breakers appropriately according to the test results and produce a rollback solution to swiftly revert services in the event of a fault.

Scenario	Performance Risk	Evaluation Method	Suggestion
Flexible query performance management	There are different types of SQL statements that offer flexibility in querying, but their execution efficiency and resource consumption can vary significantly. In extreme cases, a slow SQL statement can negatively impact the performance of the entire cluster.	To address this, you can gather statistics on CPU usage, memory usage, execution time, and the number of concurrent queries. For details, see **Real-Time Queries**.	For users who frequently use flexible queries, allocate them to separate resource pools that are independent of other services. This allows for better CPU and memory resource management. To promptly handle slow SQL statements, configure exception rules and circuit breakers. Remember to follow the Liebig's Law of the Minimum when granting permissions to these users. The administrator account should not be used as the primary account for flexible queries.
Inventory business increase	As services grow and more data is generated, the cluster's resource usage increases. If the cluster resources are not managed promptly, there may be a risk of overload.	Collect statistics on various metrics like dirty data, skew rate, **ANALYZE** time, number of partitions, and resource consumption of inventory services on a regular basis.	Inspect the cluster weekly, clearing dirty data from tables with a high dirty page rate, and performing **ANALYZE** on tables that have not had their statistics collected in a timely manner.

3.3 Before You Start: High Availability and Reliability Requirements

The DWS service logic consists of the service layer, GaussDB(DWS) cluster layer, OS layer, and VM layer, as shown in **Table 3-15**.

The performance of your GaussDB(DWS) service system is impacted by these logical layers. In the event of service changes or hardware faults, you may need to make temporary adjustments or perform emergency recovery.

To ensure efficient use of GaussDB(DWS) clusters and achieve a quick recovery time objective (RTO), adhere to the rules provided in **Table 3-16** when setting up the system.

Table 3-15 GaussDB(DWS) service logic layers

No.	Logic Layer	Description	Service Characteristic
1	Service layer	Applications on the service side	Service applications: gsql, jdbc, odbc, python, datastudio, and navicat.
2	GaussDB(DWS) layer	GaussDB(DWS) clusters	Service application logic is distributed to each CN in the GaussDB(DWS) cluster via SQL statements. Each CN then queries, optimizes, and forwards the data to DNs for processing. After processing, the DNs aggregate the data and return it to the CNs, which then send it back to the application.
3	OS layer	OS installed for GaussDB(DWS) clusters	This layer provides the running foundation, OS, file system, and network services for GaussDB(DWS).
4	VM layer	VM where GaussDB(DWS) clusters are located	This layer provides EVS disks, networks, and resources like CPU and memory.

Table 3-16 Cluster high availability and reliability risks and suggestions

Cluster Configuration	**Recommended Measure**	**Risk If Not Configured**
Configuring load balancing	Use ELB load balancing to access services. For details, see **Binding and Unbinding ELBs for a GaussDB(DWS) Cluster**.	Services are unavailable when the CN is faulty.
Backing up configuration data	Use the **backup function** to create redundant copies of both upstream and downstream data.	Data cannot be restored after being deleted by mistake, which affects your service usage and causes data security faults.
Configuring cluster DR	Configure cluster or service DR. For details, see **GaussDB(DWS) Cluster DR Management**.	In disaster scenarios at the cluster level, it is not possible to guarantee service continuity.

4 Connecting to a GaussDB(DWS) Cluster

4.1 Overview

If you have created a GaussDB(DWS) cluster, you can use the SQL client tool or a third-party driver such as JDBC or ODBC to connect to the cluster and access the database in the cluster.

Constraints and Limitations

⚠ WARNING

- Avoid having all business operations run under a single database user. Instead, plan different database users according to the business modules.
- For better access control of different business modules, it is better to use multiple users and permissions instead of depending on the system administrator user to run business operations.
- You are not advised to connect services to a single CN. Instead, configure **load balancing** to ensure that connections to each CN are balanced.
- After connecting to the database and completing required operations, close the database connection in a timely manner to prevent idle connections from continuously occupying resources and consuming connections and public resources.
- In the scenario where the database connection pool is used, after the database GUC parameters are set using the **SET** statement in the service, the parameters must be restored using the **RESET** statement before the connection pool is returned.
- For more information about development and design specifications, see **Development and Design Proposal**.

Connecting to a Cluster

The procedure for connecting to a cluster is as follows:

1. **Obtaining the Connection Address of a GaussDB(DWS) Cluster**
2. If SSL encryption is used, perform the operations in **Establishing Secure TCP/IP Connections in SSL Mode**.
3. Connect to the cluster and access the database in the cluster. You can choose any of the following methods to connect to a cluster:

NOTICE

- You are advised to use the officially recommended method for connecting to the database.
- Compatibility with other clients cannot be guaranteed, so it may be necessary to verify it.
- If an error occurs due to incompatibility with another client and the client cannot be replaced, try replacing the libpq driver on the client. To replace the **libpg.so** file on the client, download and extract the gsql client package by referring to **Downloading the Client**, locate the **gsql** directory, and obtain the file. Then, replace the existing **libpg.so** file in the designated directory on the client.

   - **Using the SQL Editor to Connect to a Cluster**
   - Use the SQL client tool to connect to the cluster.
     - **Using the Linux gsql Client to Connect to a Cluster**
     - **Using the Windows gsql Client to Connect to a Cluster**
     - **Using Data Studio to Connect to a GaussDB(DWS) Cluster**
   - Use a JDBC, psycopg2, or PyGreSQL driver to connect to the cluster.
     - **Using JDBC to Connect to a Cluster**
     - **Using ODBC to Connect to a Cluster**
     - **Using the Third-Party Function Library psycopg2 of Python to Connect to a Cluster**
     - **Using the Python Library PyGreSQL to Connect to a Cluster**
     - **Configuring JDBC to Connect to a Cluster (IAM Authentication Mode)**

4.2 Obtaining the Connection Address of a GaussDB(DWS) Cluster

Scenario

You can access GaussDB(DWS) clusters by different methods and the connection address of each connection method varies. This section describes how to view and obtain the private network address on the Huawei Cloud platform, public network address on the Internet, and JDBC connection strings.

To obtain the cluster connection address, use either of the following methods:

- Obtaining the cluster connection address on the Client Connections Page
- Obtaining the Cluster Access Addresses on the Cluster Information Page

Obtaining the cluster connection address on the Client Connections Page

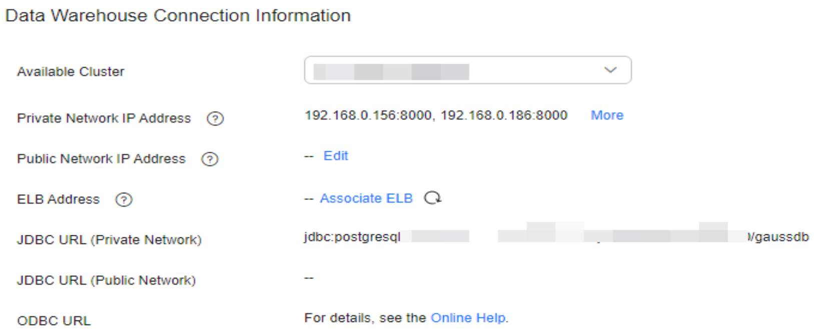
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Management** > **Client Connections**.

Step 3 In the **Data Warehouse Connection Information** area, select an available cluster.

You can only select clusters in the **Available** state.

Figure 4-1 Data warehouse connection information



Data Warehouse Connection Information

Available Cluster	[redacted]
Private Network IP Address	192.168.0.156:8000, 192.168.0.186:8000 More
Public Network IP Address	-- Edit
ELB Address	-- Associate ELB
JDBC URL (Private Network)	jdbc:postgresql [redacted] /gaussdb
JDBC URL (Public Network)	--
ODBC URL	For details, see the Online Help.

Step 4 View and obtain the cluster connection information.

- **Private Network IP Address**
- **Public Network IP Address**
- **ELB Address**
- **JDBC URL (Private Network)**
- **JDBC URL (Public Network)**
- **ODBC URL**

NOTE

- If no EIP is automatically assigned during cluster creation, **Public Network Address** is empty. If you want to use a public network address (consisting of an EIP and the database port) to access the cluster from the Internet, click **Bind EIP** to bind one.
- If an EIP is bound during cluster creation but you do not want to use the public network address to access the cluster, click **Unbind EIP** to unbind the EIP. After the EIP is unbound, **Public Network Address** is empty.
- If a cluster was not bound to ELB when it was created, the **ELB Address** parameter will be left blank. You can bind the cluster to ELB to avoid single CN failures.
- If a cluster has been bound to ELB, use the ELB address to connect to the cluster for high availability purposes.
- If the IPv6 dual stack is enabled for a GaussDB(DWS) cluster, private IPv4 and IPv6 addresses can be used. You can connect to the cluster via IPv4 or IPv6.

----End

Obtaining the Cluster Access Addresses on the Cluster Information Page

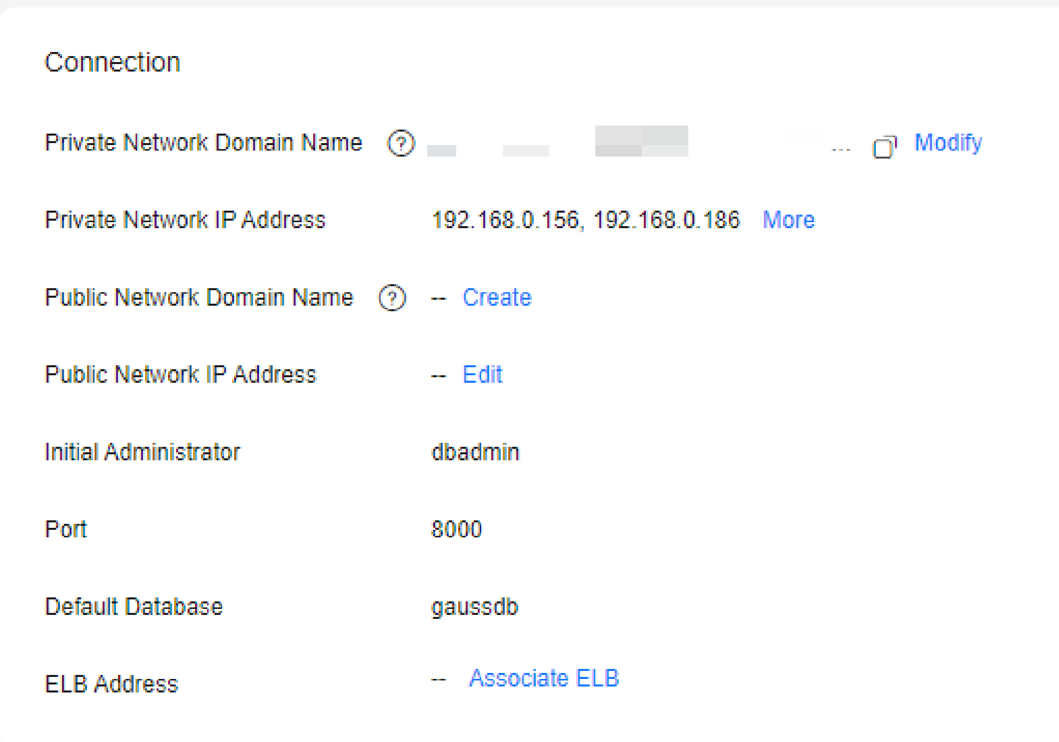
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 4 In the **Connection** area, view and obtain the cluster's access address information, including the private network address and public network address.

Figure 4-2 Access addresses



Connection

Private Network Domain Name [redacted] ... Modify

Private Network IP Address 192.168.0.156, 192.168.0.186 More

Public Network Domain Name -- Create

Public Network IP Address -- Edit

Initial Administrator dbadmin

Port 8000

Default Database gaussdb

ELB Address -- Associate ELB

Table 4-1 Connection

Parameter	Description
Private Network Domain Name	Domain name for accessing the cluster database through the internal network. The domain name corresponds to all CN IP addresses. The private network domain address is automatically generated when a cluster is created. The default naming rule is *cluster name*.dws.myhuaweicloud.com. **NOTE** - If the cluster name does not comply with the domain name standards, the prefix of the default access domain name will be adjusted accordingly. - Load balancing is not supported. You can click **Modify** to change the private network domain name. The access domain name contains 4 to 63 characters, which consists of letters, digits, and hyphens (-), and must start with a letter. For details, see **Managing GaussDB(DWS) Cluster Access Domain Names**.
Private Network IP Address	IP address for accessing the database in the cluster over the private network. **NOTE** - A private IP address is automatically generated when you create a cluster. The IP address is fixed. - The number of private IP addresses equals the number of CNs. You can log in to any node to connect to the cluster. - If you access a fixed IP address over the internal network, all the resource pools will run on a single CN. - If IPv6 is enabled for a cluster, both IPv4 and IPv6 private addresses will be displayed. You can use either of them as needed.
Public Network Domain Name	Name of the domain for accessing the database in the cluster over the public network. For details, see **Managing GaussDB(DWS) Cluster Access Domain Names**. **NOTE** Load balancing is not supported.
Public Network IP Address	IP address for accessing the database in the cluster over the public network. **NOTE** - If no EIP is assigned during cluster creation and **Public Network IP Address** is empty, click **Edit** to bind an EIP to the cluster. - If an EIP is bound during cluster creation, click **Edit** to unbind the EIP.
Initial Administrator	Database administrator specified during cluster creation. When you connect to the cluster for the first time, you need to use the initial database administrator and password to connect to the default database.

Parameter	Description
Port	Port number for accessing the cluster database through the public network or private network. The port number is specified when the cluster is created.
Default Database	Database name specified when the cluster is created. When you connect to the cluster for the first time, connect to the default database.
ELB Address	To achieve high availability and avoid single-CN failures, a new cluster needs to be bound to ELB. You are advised to use the ELB address to connect to the cluster.

----End

4.3 Using a Visualization Tool to Connect to a GaussDB(DWS) Cluster

4.3.1 Using the SQL Editor to Connect to a GaussDB(DWS) Cluster

4.3.1.1 Overview

GaussDB(DWS) provides you with a one-stop data development tool, that is, the online SQL editor, for data development, access, and processing.

The online SQL editor lets you connect to cluster databases on the GaussDB(DWS) console. It shows the database metadata details, runs and edits SQL statements, and displays the results in various charts. It also saves scripts with OBS, which can be set up globally and store SQL statements as text files.

NOTE

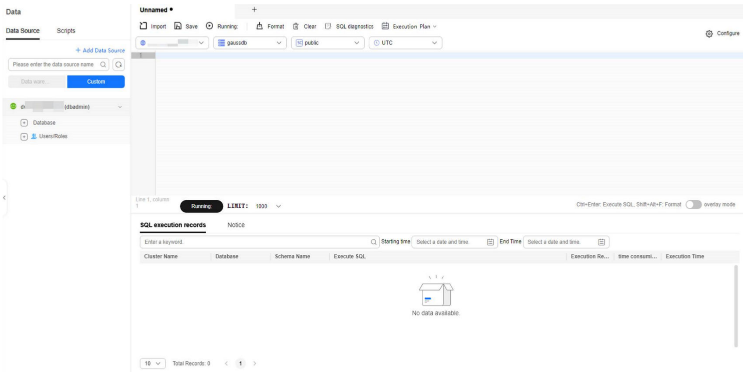
- This tool is supported only by clusters of version 8.1.3 or later.
- The editor depends on GaussDB(DWS) and OBS . You need to enable the GaussDB(DWS) cluster query and OBS query operations, and interconnect the editor with the Cloud Trace Service (CTS) service to record traces of operation APIs.

Editor Functions

- In the upper part of the editor, you can switch the data source, database, and schema.
- SQL statements can be written in the middle, where highlighting, basic syntax tips, and information about databases, schemas, tables, and fields are provided. For details about the SQL syntax, visit **SQL Syntax Reference**.
- You can format SQL statements and query execution plans with this tool. But be careful, the PERFORMANCE execution plan executes SQL statements. So, avoid using it for SQL statements that perform operations.

- You can click **Save** after writing many SQL statements. Then a dialog box will ask you to save them to the right OBS bucket.
- The query results show at the bottom in multiple pages. You can make pie, line, or bar charts from different fields. You can also export the results to an Excel file. The **SQL execution records** area displays non-query SQL statement records from the past six months.
- You can switch to the script panel and show the directory folder. The script file is saved in the created directory. For details, see **Creating a Directory**. The editor allows for two directory levels, each with a capacity of 10 folders. Each folder can hold up to 100 script files, which are stored in the corresponding OBS bucket file directory. To make things easier, the OBS bucket file address can be set globally. For details, see **Global Settings**..

Figure 4-3 SQL editor page



4.3.1.2 Using the SQL Editor to Connect to a Cluster

Data sources are used for cluster login. Currently, the GaussDB(DWS) cluster supports two login modes: custom (username + password) and IAM account. Custom login is the default login mode. With IAM account login, you create an IAM user in the database and use a token to log in.

NOTE

- IAM account login is only supported by clusters of 8.1.3.331 (included) to 8.2.0 (excluded), 8.2.1.100, and later versions. For storage-compute decoupled clusters, only clusters of 9.1.0 and later versions support this login method.
- If you log in to a cluster as a custom user or a IAM user, you can create a user or grant permissions to an existing user on the cluster user management page. For details, see **Creating a GaussDB(DWS) Database and User**.

Constraints and Limitations

- Custom login constraints:
  - Select a cluster for the new data source, enter the username and password, and test the connection. After the connection is tested, you can open the cluster data connection.
  - You are advised to select **Remember password** during login. If the database is not specified, the **GaussDB** database is used by default.

  - Depending on tenants and users, connection permissions are separated. Connections are visible to different sub-users. Only the creator of a connection can view it.
- IAM account login constraints:
  - Only IAM users who have been assigned the **DWS Database Access** role can log in to GaussDB(DWS) clusters. In this case, you need to contact a user with the **DWS Administrator** permissions to grant you the role on the current page.
  - The IAM user does not have any permissions after logging in to the GaussDB(DWS) cluster database. You need to assign permissions to the IAM user on the user management page.
- Connection timeout limit:
  - The connection timeout period is set in the background. If no operation is performed within 30 minutes, you need to log in again.
  - The connection uniquely caches the user login ID and database name to guarantee that each user connects to each database with one connection and performs each operation on one connection.
  - It is not recommended to run SQL commands on the same database from multiple windows. This can cause delays because the database establishes the same connection, and each command can only be executed after the previous one has finished.

Procedure

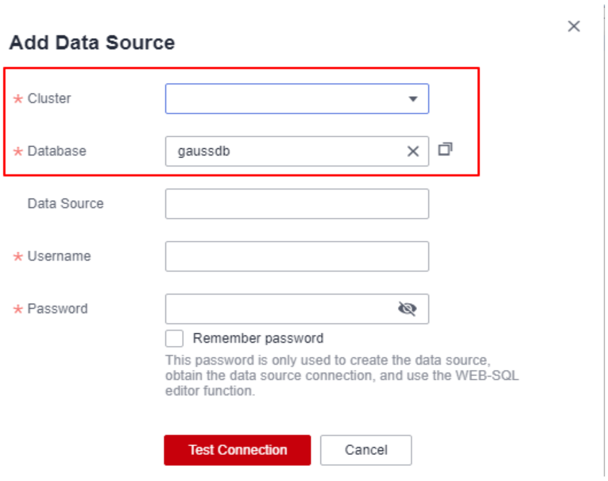
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 There are two panels on the left: **Data Warehouse** and **Custom**. Data Warehouse can only be logged in by IAM users with the **DWS Database Access** role. If the conditions are met, you can connect to a cluster database to perform operations.

Step 4 Switch to the **Custom** panel and click **Add Data Source**. (Alternatively, access the **Dedicate Cluster** page, locate the row that contains the target cluster, and click **Login** in the **Operation** column.)

- **Cluster**: Select the cluster to be connected.
- **SSL certification**: Select this option if SSL authentication is enabled for the cluster.
- **Database**: Enter the database name. For a newly created cluster, enter the default database **gaussdb** or switch to another database as required.
- **Data Source**: Set the data source name.
- **Username**: Set the username.
- **Password**: Set the password of the user. This password is used only to create data sources and data source connections for the WEB-SQL editor. If **Remember password** is selected, the default password is used for login when the data source is opened. If **Remember password** is not selected, the password needs to be entered again when the data source is opened after the page is refreshed or the login expires.



Add Data Source

* Cluster

* Database: gaussdb

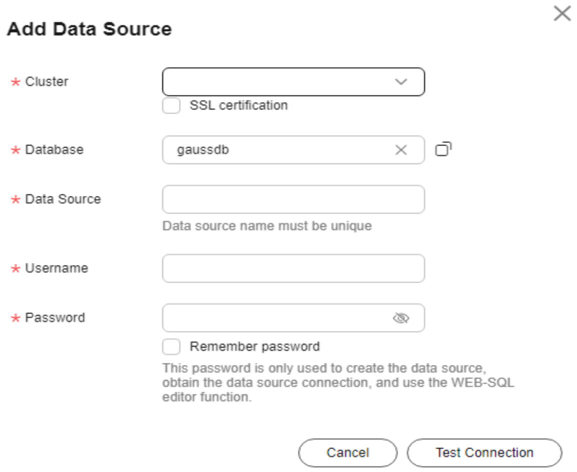
Data Source

* Username

* Password

□ Remember password

This password is only used to create the data source, obtain the data source connection, and use the WEB-SQL editor function.



Add Data Source

* Cluster

□ SSL certification

* Database: gaussdb

* Data Source

Data source name must be unique

* Username

* Password

□ Remember password

This password is only used to create the data source, obtain the data source connection, and use the WEB-SQL editor function.

Step 5 Confirm the information and click **Test Connection**.

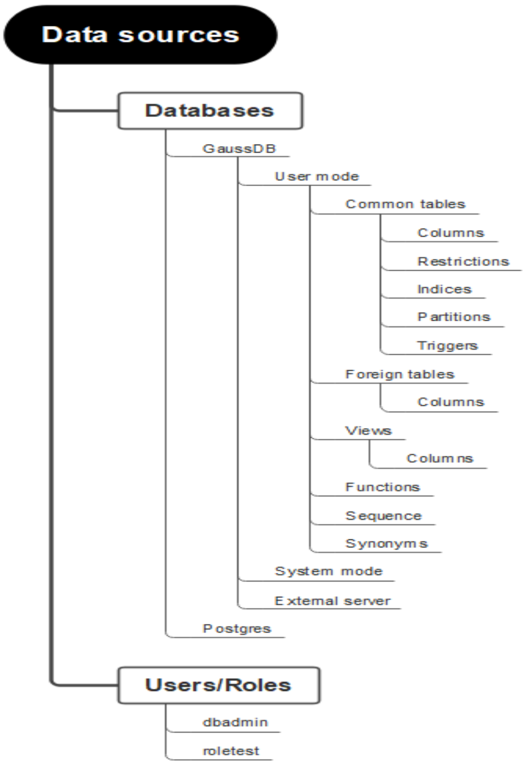
----End

4.3.1.3 Data Development Operations

Metadata management includes the hierarchical display of metadata. The hierarchy begins with the data source at the root, branching into databases and user roles. Databases include system schemas, user schemas, and foreign servers. System schemas and user schemas are distinguished by OIDs and system schemas cannot be changed or deleted. User schemas include common/partitioned tables, foreign tables, views, functions, sequences, and synonyms. A table contains columns, constraints, indexes, partitions, and triggers. The LIST and INFO APIs are provided to query the metadata lists and details.

The following figure shows the metadata list. Currently, databases, schemas, common tables, fields, indexes, constraints, and partitions can be added.

Figure 4-4 Metadata information hierarchy



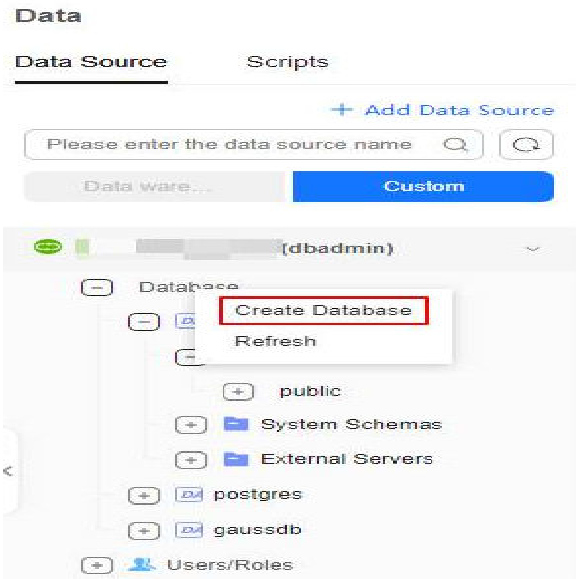
Adding a Database

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source**. After a **data source is connected**, right-click the database name and click **Create Database**.

Figure 4-5 Creating a database



Step 4 On the page for adding a database, set the parameters as required.

- **Database Name**: Set the database name.
- **Owner**: Select the new database owner from the drop-down list box.
- **Compatibility mode**: Choose a database compatibility mode rom the drop-down list. The available options include **Oracle**, **MySQL**, and **Teradata**. The default setting is **Oracle**.
- **Encoding**: Select the encoding mode of the new database from the drop-down list box. SQL_ASCII is recommended.
- **Connection Limit**: The value cannot be less than **-1**. The value **-1** indicates no limit.
- **Description**: Description of the new database.
- **SQL Preview**: You can click **Preview** to view the SQL syntax for creating the database.

Step 5 Click **OK**.

----End

Adding a Schema

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source** to add a database. For details, see **Adding a Database**. The database contains the user schema, system schema, and external server.

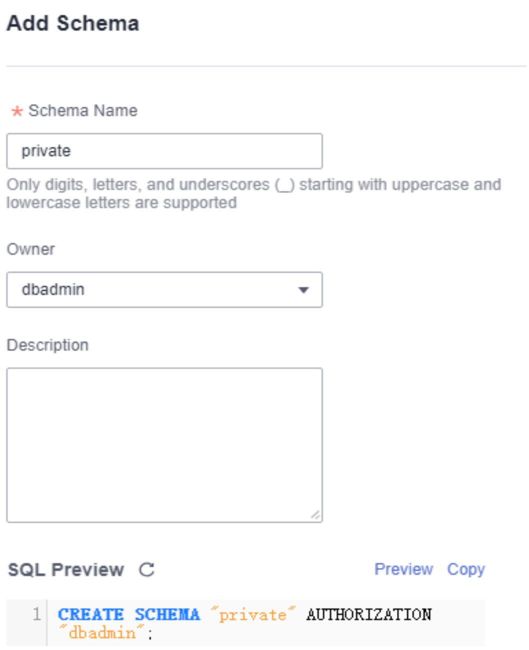
> 🕮 NOTE
>
> The system schema can only be viewed.

Step 4 Right-click the user mode name and click **Add Schema**.

Step 5 Set the parameters as required on the displayed page.

- **Schema Name**: Set the schema name.
- **Owner**: Select the owner of the new schema from the drop-down list box.
- **Description**: Description of the new schema.
- **SQL Preview**: Click **Preview** to display the SQL syntax for creating the schema.

Figure 4-6 Adding a Schema



Add Schema

* Schema Name

private

Only digits, letters, and underscores (_) starting with uppercase and lowercase letters are supported

Owner

dbadmin

Description

SQL Preview

```
CREATE SCHEMA "private" AUTHORIZATION "dbadmin";
```

Step 6 Click **OK**.

----End

Adding a Common Table

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Switch to the **Data Source** panel and add a schema (for details, see **Adding a Schema**). A schema contains structures such as common tables, foreign tables, views, functions, sequences, and synonyms.

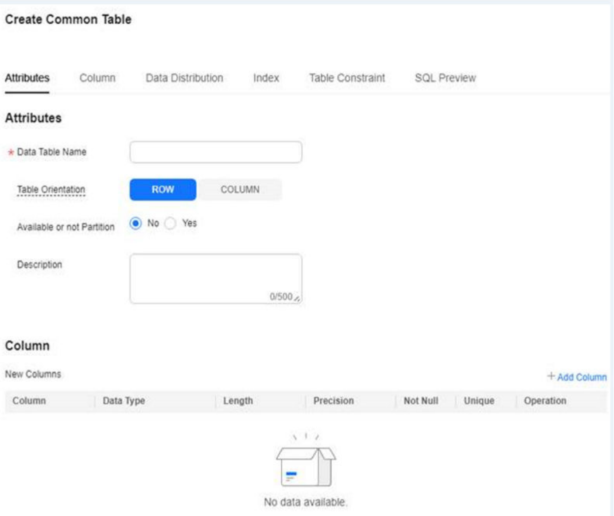
Step 4 Right-click the name of the common table and click **Create Common Table** to add a table. The dialog box for adding a common table contains options such as **Attribute**, **Column**, **Data Distribution**, **Partition**, **Index**, and **Constraint**. The **Attribute** and **Column** fields are mandatory. You can click **SQL Preview** to query the SQL statement for creating a table.

Table 4-2 Parameters for adding a data table

Tab	Description
Attribute	● **Data Table Name**: Set the data table name. ● **Table Orientation**: You can select **ROW** or **COLUMN**. ● **Available or not Partition**: Select whether to create a partitioned table. ● **Description**: Description of the new data table.

Tab	Description
Column	Click **Add Column** and set the following parameters: ● **Column Name**: Set the column name. ● **Data Type**: Select the data type of the new column from the drop-down list box. ● **Length**: Total number of digits. If this parameter is dimmed, the length is fixed. ● **Precision**: Number of decimal places. If this parameter is dimmed, the precision cannot be set. ● **Non-null**: Select whether the new column is not null. ● **Unique**: Select whether the new column is unique.
Data Distribution	The options are as follows: ● **ROUNDROBIN**: Each row of data in the table is sent to each DN in sequence. ● **REPLICATION**: Each row of data in the table exists on all DNs. Each DN has complete table data. ● **HASH**: Specified columns are hashed and data is distributed to specified DNs through mapping.
Partition	On the **Partition** panel, you can select a partition type (range partition or list partition) and optional columns (corresponding to table fields). Click **Add Partition** and set the following parameters: ● **Partition Name**: Set the partition name. ● **Partition Value**: Select a value from the range based on the optional columns.
Indexes	Click **Add Index** and set the following parameters: ● **Index Name**: Set the index name. You can select **Unique Index**. ● **Access Mode**: Select an index access mode from the drop-down list box. B-tree indexes are recommended. ● **Index Type**: The options are **Column** and **Expression**. ● **Condition Index**: The **WHERE** condition constraint can be added.
Table Constraints	Click **Add Constraint** and set the following parameters: ● **Constraint Type**: The value can be **check**, **unique**, or **primary**. ● **Expression** (**check**): Enter field constraints. ● **Constraint Name**: Set the constraint name. ● **Optional Column** (unique\primary): Select an optional column from the drop-down list box.
SQL Preview	Click **Preview** to display the SQL syntax for creating a common table.

Figure 4-7 Creating a common table



Step 5 Click **OK**.

----End

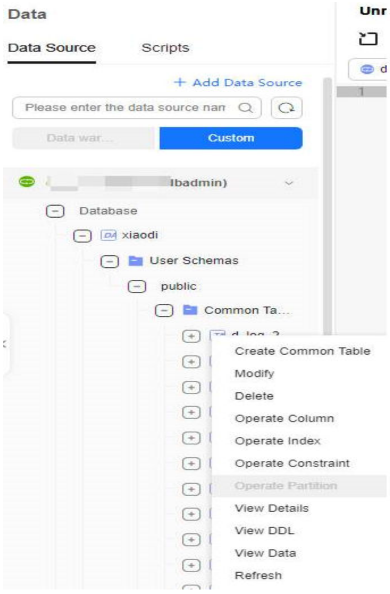
Editing a Common Table

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source**. You can edit the created common table. For details about how to create a common table, see **Adding a Common Table**.

Figure 4-8 Editing a common table



Step 4 Right-click the name of the common table name to modify it. The following table describes the modification operations.

Table 4-3 Table modification operations

Operation	Description
Modifying a common table	You can click **Modify** to modify the name and schema of a table, and specify whether it is a partitioned table.
Deleting a common table	Click **Delete** to delete a common table.
Operating columns	Click the corresponding operation button to add, edit, and delete columns. You can edit the column name, data type, length, and specify whether it is a non-NULL column. Batch adding of columns is also available.
Operating indexes	Click **Operate Index** to add indexes, edit indexes (index names), and delete indexes in batches.
Operating constraints	Click **Operate Constraint** to add constraints, edit constraints (constraint names and optional columns), and delete constraints in batches.
Operating partitions	Click **Operate Constraint** (unavailable for non-partitioned tables) to add, edit, and delete partitions. You can edit the partition name. Batch adding of partitions is also available.

NOTE

You can also right-click a constraint, index, or partition name and choose **Operate Constraint/Index/Partition** from the shortcut menu to modify the corresponding attributes.

Step 5 Confirm the information and click **OK**.

----End

Viewing Data in a Common Table

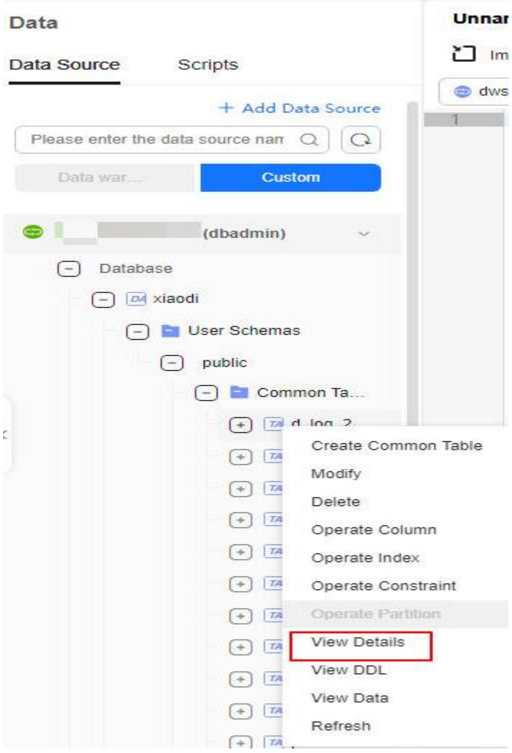
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source** and right-click the data table name.

Step 4 Click **View Details** to add, filter, edit, and delete data in a common table.

Figure 4-9 Viewing data in a common table



NOTE

Right-click a partition name and choose **View Details** to add, filter, edit, or delete partition data.

----End

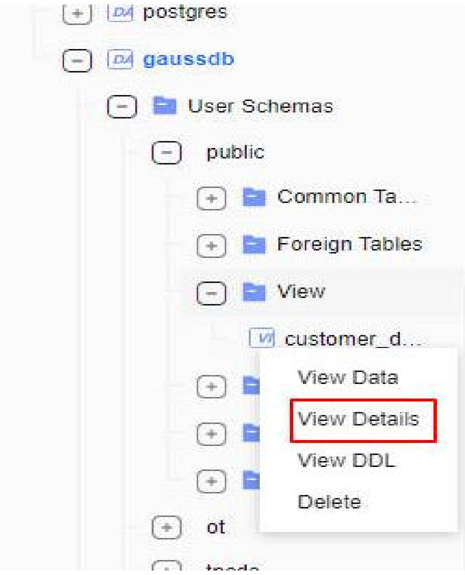
Checking Views

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source**, right-click a view name, and choose **View Details** from the shortcut menu to check the views of the database.

Figure 4-10 Checking views



----End

Importing Data

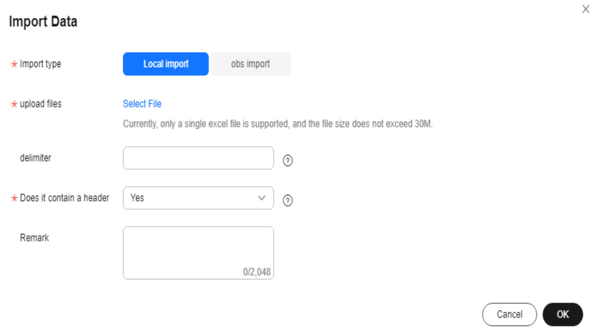
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source** and right-click the common table name. Click **Import Data** to import data from the local Excel file or OBS bucket file to the common table.

- **Local import**: When uploading an Excel file, ensure it is under 30 MB. For CSV files, select separators to divide data in each row and indicate if there is a table header. If there is no header, input data in each row according to the chosen table fields.

  Figure 4-11 Importing Local Excel Data to a Common Table

  Import Data

  * Import type: Local import | obs import

  * upload files: Select File

  Currently, only a single excel file is supported, and the file size does not exceed 30M.

  delimiter:

  * Does it contain a header: Yes

  Remark: 0/2,048

  Cancel OK

- **obs import**: Choose a file from the OBS bucket or directory. Supported file types include CSV and TEXT. Set the parameters of the foreign table to be created for importing data to the OBS bucket. Use the OBS foreign table to write the OBS bucket file to the selected common table.

  NOTE

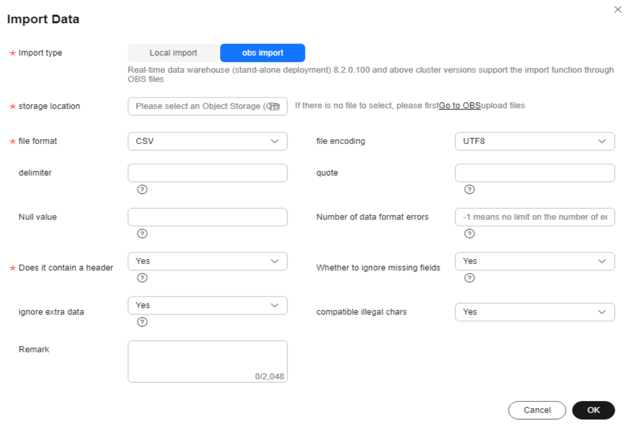
  Starting from version 8.2.0.100, storage-compute coupled data warehouses (single-node deployment) can import OBS files.

Table 4-4 OBS import parameters

Parameter	Description	Example Value
storage location	Choose a file from the OBS bucket.	-
file format	Select a file format from the drop-down list. Supported formats include CSV and TEXT.	CSV
file encoding	Select a file encoding mode from the drop-down list. UTF8 is recommended.	UTF8
Delimiter	Commas (,) are the default separators for CSV files, while tab characters are the default separators for TXT files.	,
quote (CSV format)	Quotation marks are used for CSV files. The value should be a single-byte character and cannot be the same as the delimiter or null parameter.	#
newline character (TEXT format)	When importing data in TEXT format, you can specify the newline character style. The maximum length of the newline character is 10 bytes, and multi-character newline characters are supported. The supported newline characters include common ones like **\r**, **\n**, and **\r\n**, as well as other characters or character strings like **$** and **#**.	\r
Whether not to escape (TEXT format)	You can specify whether to escape the backslash (\) and its following characters in the TEXT format.	Yes
Null value	You can specify how null values are represented in a data file.	$
Number of data format errors	Maximum number of data format errors allowed during data import. The value **-1** indicates that the number of errors is not limited.	-1
Does it contain a header (CSV format)	You can specify if the exported CSV file should contain a header row that describes each column in the table. This parameter only applies to CSV files.	Yes

Parameter	Description	Example Value
Whether to ignore missing fields	Enabling this function sets the last column of a row in a data source file to **NULL** if it is missing, without reporting an error message.	Yes
ignore extra data	You can specify whether to ignore excessive columns when the number of columns in a source data file exceeds that defined in the foreign table.	Yes
compatible illegal chars	Enabling this function allows for invalid characters during data import.	Yes

Figure 4-12 Importing OBS bucket file data to a common table



Import Data

* Import type: Local import | obs import

Real-time data warehouse (stand-alone deployment) 8.2.0.100 and above cluster versions support the import function through OBS files

* storage location: Please select an Object Storage (OB — If there is no file to select, please firstGo to OBSupload files

* file format: CSV — file encoding: UTF8

delimiter: — quote:

Null value: — Number of data format errors: -1 means no limit on the number of er

* Does it contain a header: Yes — Whether to ignore missing fields: Yes

ignore extra data: Yes — compatible illegal chars: Yes

Remark: 0/2,048

Step 4 Click **OK**.

Step 5 In the upper right corner of the page, choose **Common Functions** > **Import data list** and check whether the import is successful.

----End

Exporting Data

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

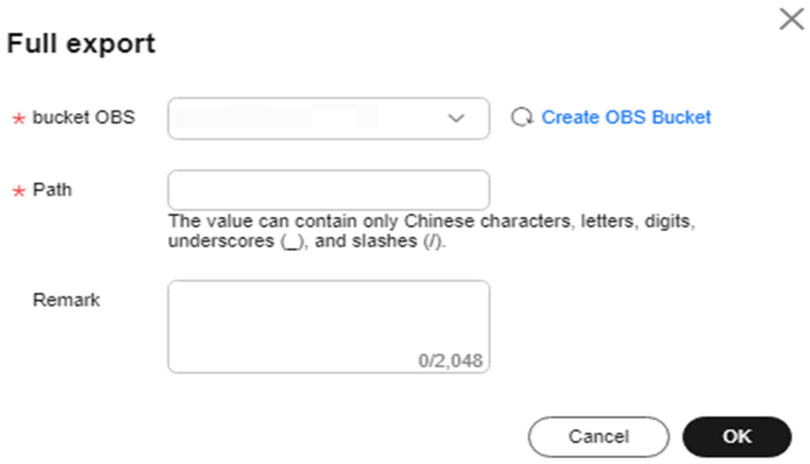
Step 3 Click **Data Sources**, log in to the data source whose data needs to be exported, and select the corresponding database and schema.

Step 4 Enter the query SQL statement in the editor box and click **Running**.

Step 5 Click **Export** under the query result.

- Local export: Export all SQL query results to an XLSX or CSV file. You can open the file on your local PC. A maximum of 5,000 records can be exported.
- Full export: Export all query SQL results to a specified path in an OBS bucket. By default, the results are exported to a CSV file.

Figure 4-13 Full data export



Full export

* bucket OBS

* Path

The value can contain only Chinese characters, letters, digits, underscores (_), and slashes (/).

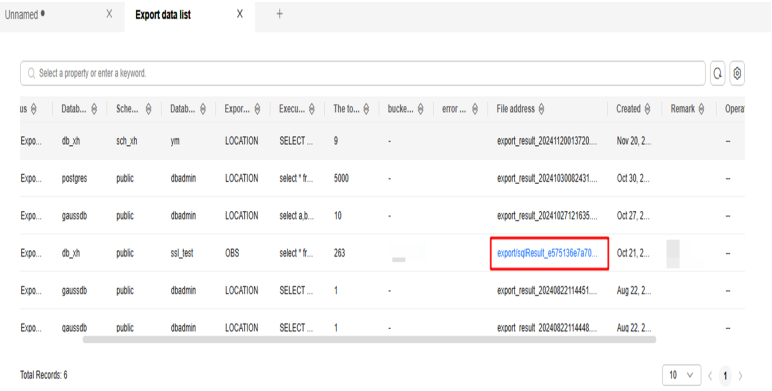
Remark

0/2,048

Step 6 Choose **Common Functions** > **Export data list** in the upper right corner of the page and view the exported task in the data export list.

Step 7 Click the path in the **File address** column to go to the OBS console to download the exported CSV file.

Figure 4-14 Exporting the task list



us	Datab...	Sche...	Datab...	Expor...	Execu...	The to...	bucke...	error ...	File address	Created	Remark	Opera
Expo...	db_xh	sch_xh	ym	LOCATION	SELECT ...	9	-		export_result_20241120013720....	Nov 20, 2...		--
Expo...	postgres	public	dbadmin	LOCATION	select * fr...	5000	-		export_result_20241030082431....	Oct 30, 2...		--
Expo...	gaussdb	public	dbadmin	LOCATION	select a,b...	10	-		export_result_20241027121635....	Oct 27, 2...		--
Expo...	db_xh	public	ssl_test	OBS	select * fr...	263			export/sqlResult_e575136e7a70...	Oct 21, 2...		--
Expo...	gaussdb	public	dbadmin	LOCATION	SELECT ...	1	-		export_result_20240822114451....	Aug 22, 2...		--
Expo...	gaussdb	public	dbadmin	LOCATION	SELECT ...	1	-		export_result_20240822114448....	Aug 22, 2...		--

Total Records: 6

----End

Submitting a SQL Task

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

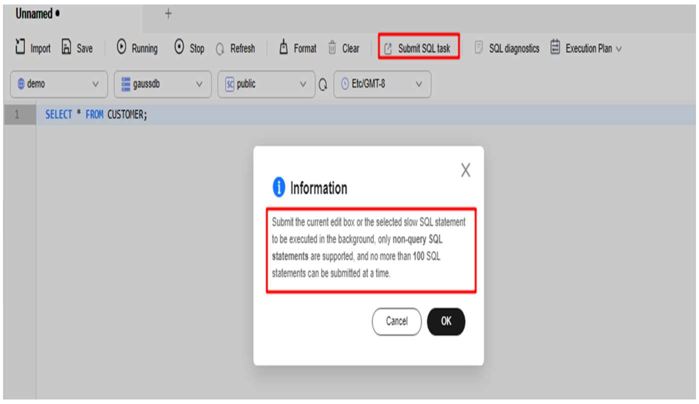
Step 3 Click **Data Sources**, log in to the data source for which you want to submit the SQL task, and select the corresponding database and schema.

Step 4 Enter a non-query SQL statement in the text box and click **Submit SQL task** to submit the selected SQL statement to the background task for execution.

NOTE

A maximum of 100 non-query SQL statements can be submitted at a time. A maximum of five SQL tasks can be executed by a user.

Figure 4-15 Submitting a SQL task



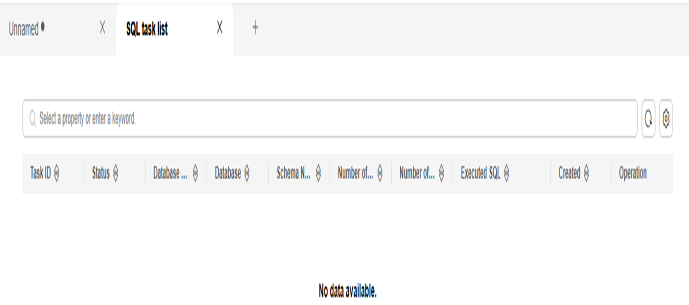
```
SELECT * FROM CUSTOMER;
```

Submit the current edit box or the selected slow SQL statement to be executed in the background, only **non-query SQL statements** are supported, and no more than **100** SQL statements can be submitted at a time.

Step 5 Choose **Common Functions** > **SQL task list** in the upper right corner of the page to view the executed SQL tasks.

Step 6 Click **Details** in the **Operation** column to view the execution status of each SQL statement.

Figure 4-16 Viewing SQL task details



Task ID	Status	Database ...	Database	Schema N...	Number of...	Number of...	Executed SQL	Created	Operation

No data available.

----End

Plan Diagnosis

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

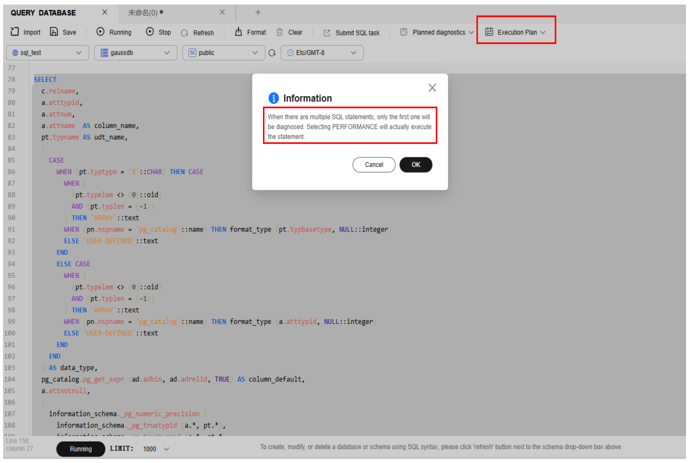
Step 3 Click **Data Sources**, log in to the data source for which you need to enter SQL statements, and select the corresponding database and schema.

Step 4 Enter a query SQL statement in the text box and click **Planned diagnostics**. You can select **EXPLAIN** or **PERFORMANCE**.

NOTE

Only one query SQL statement can be diagnosed at a time. If you enter multiple query SQL statements, the first SQL statement is diagnosed by default. If you select **PERFORMANCE**, the entered SQL statement is executed and the result is returned.

Figure 4-17 Selecting PERFORMANCE



```
SELECT
  c.relname,
  a.atttypid,
  a.attnum,
  a.attname  AS column_name,
  pt.typname AS udt_name,
  (
    CASE
      WHEN (pt.typtype = 'd'::CHAR) THEN CASE
        WHEN (
          (pt.typelem <> (0)::oid)
          AND (pt.typlen = (-1))
        ) THEN 'ARRAY'::text
        WHEN (pn.nspname = 'pg_catalog'::name) THEN format_type (pt.typbasetype, NULL::integer)
        ELSE 'USER-DEFINED'::text
      END
      ELSE CASE
        WHEN (
          (pt.typelem <> (0)::oid)
          AND (pt.typlen = (-1))
        ) THEN 'ARRAY'::text
        WHEN (pn.nspname = 'pg_catalog'::name) THEN format_type (a.atttypid, NULL::integer)
        ELSE 'USER-DEFINED'::text
      END
    END
  ) AS data_type,
  pg_catalog.pg_get_expr (ad.adbin, ad.adrelid, TRUE) AS column_default,
  a.attnotnull,
  (
    information_schema._pg_numeric_precision (
      information_schema._pg_truetypid (a.*, pt.*),
```

Information

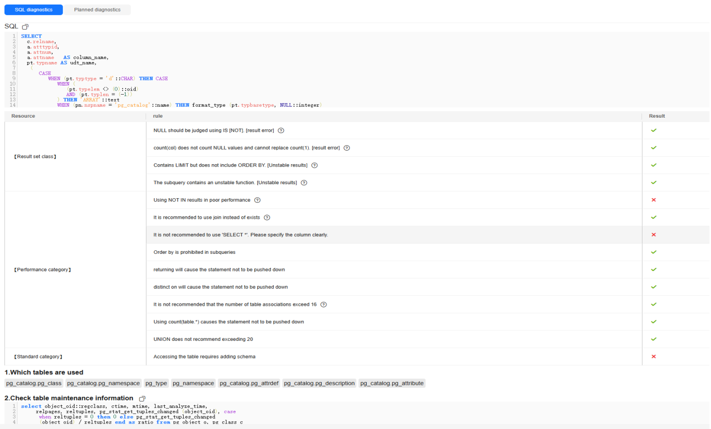
When there are multiple SQL statements, only the first one will be diagnosed. Selecting PERFORMANCE will actually execute the statement.

To create, modify, or delete a database or schema using SQL syntax, please click 'refresh' button next to the schema drop-down box above.

Step 5 Click **OK**. The **Planned diagnostics** page is displayed. You can view the SQL statement diagnosis result and plan diagnosis visualization result.

- Click **SQL diagnostics** to view the SQL statement formatting and diagnosis items.

Figure 4-18 Viewing SQL statement diagnosis results



```
SELECT
  c.relname,
  a.atttypid,
  a.attnum,
  a.attname   AS column_name,
  pt.typname AS udt_name,
  (
    CASE
      WHEN (pt.typtype = 'd'::CHAR) THEN CASE
        WHEN (
          (pt.typelem <> (0)::oid)
          AND (pt.typlen = (-1))
        ) THEN 'ARRAY'::text
        WHEN (pn.nspname = 'pg_catalog'::name) THEN format_type (pt.typbasetype, NULL::integer)
```

Resource	rule	Result
【Result set class】	NULL should be judged using IS [NOT]. [result error]	✓
	count(col) does not count NULL values and cannot replace count(1). [result error]	✓
	Contains LIMIT but does not include ORDER BY. [Unstable results]	✓
	The subquery contains an unstable function. [Unstable results]	✓
【Performance category】	Using NOT IN results in poor performance	✗
	It is recommended to use join instead of exists	✓
	It is not recommended to use 'SELECT *'. Please specify the column clearly.	✗
	Order by is prohibited in subqueries	✓
	returning will cause the statement not to be pushed down	✓
	distinct on will cause the statement not to be pushed down	✓
	It is not recommended that the number of table associations exceed 16	✓
	Using count(table.*) causes the statement not to be pushed down	✓
	UNION does not recommend exceeding 20	✓
【Standard category】	Accessing the table requires adding schema	✗

1.Which tables are used

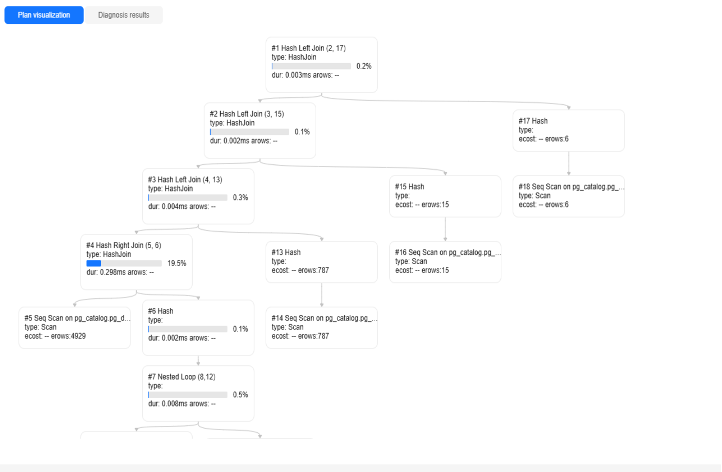
pg_catalog.pg_class pg_catalog.pg_namespace pg_type pg_namespace pg_catalog.pg_attrdef pg_catalog.pg_description pg_catalog.pg_attribute

2.Check table maintenance information

```
select object_oid::regclass, ctime, mtime, last_analyze_time,
    relpages, reltuples, pg_stat_get_tuples_changed (object_oid), case
     when reltuples = 0 then 0 else pg_stat_get_tuples_changed
      (object_oid) / reltuples end as ratio from pg_object o, pg_class c
```

- Click **Planned diagnostics** to display the plan tree node and plan diagnosis result of the SQL statement.

Figure 4-19 Viewing SQL plan diagnosis results



----End

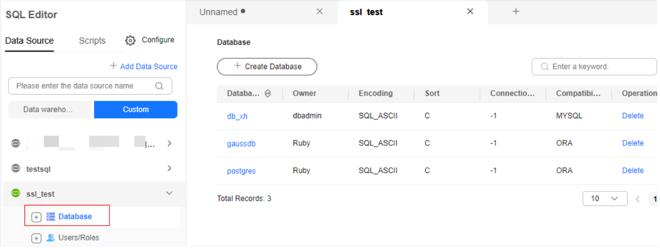
Viewing Statistics About Databases, Schemas, and Tables

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source** and double-click a database name. On the database list page that is displayed, you can search and check the detailed information about a database.

Figure 4-20 Viewing database details

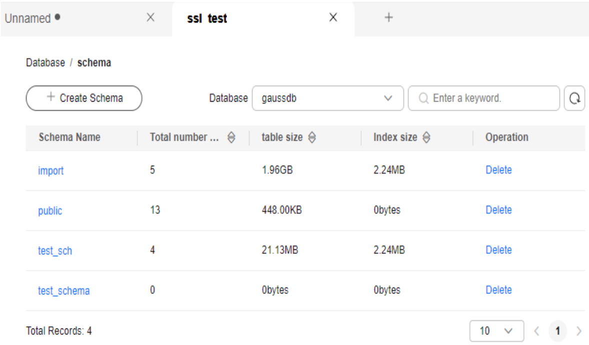


Databa...	Owner	Encoding	Sort	Connectio...	Compatibi...	Operation
db_xh	dbadmin	SQL_ASCII	C	-1	MYSQL	Delete
gaussdb	Ruby	SQL_ASCII	C	-1	ORA	Delete
postgres	Ruby	SQL_ASCII	C	-1	ORA	Delete

Total Records: 3

Step 4 Click a database name in the database list to go to the schema list and view the total number of tables, total table size, and index size.

Figure 4-21 Viewing schema information

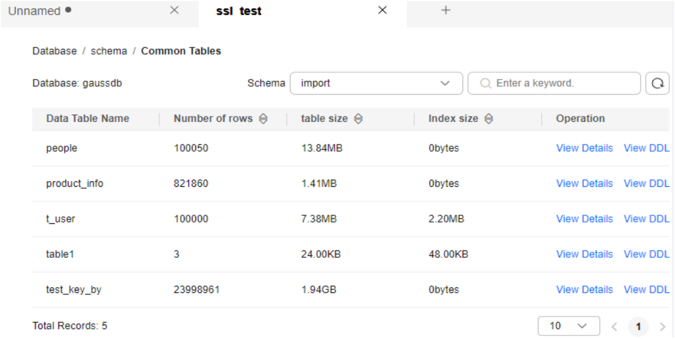


Schema Name	Total number ...	table size	Index size	Operation
import	5	1.96GB	2.24MB	Delete
public	13	448.00KB	0bytes	Delete
test_sch	4	21.13MB	2.24MB	Delete
test_schema	0	0bytes	0bytes	Delete

Total Records: 4

Step 5 Click a schema name in the schema list to go to the common table list. You can view the number of rows in the table, table size, and index size.

Figure 4-22 Viewing common table information



Unnamed ● × | ssl_test × | +

Database / schema / Common Tables

Database: gaussdb   Schema: import   Enter a keyword.

Data Table Name	Number of rows	table size	Index size	Operation
people	100050	13.84MB	0bytes	View Details View DDL
product_info	821860	1.41MB	0bytes	View Details View DDL
t_user	100000	7.38MB	2.20MB	View Details View DDL
table1	3	24.00KB	48.00KB	View Details View DDL
test_key_by	23998961	1.94GB	0bytes	View Details View DDL

Total Records: 5   10   1

----End

Sharing a Custom Data Source with Other IAM Users of the Same Tenant

Step 1 Log in to the GaussDB(DWS) console.

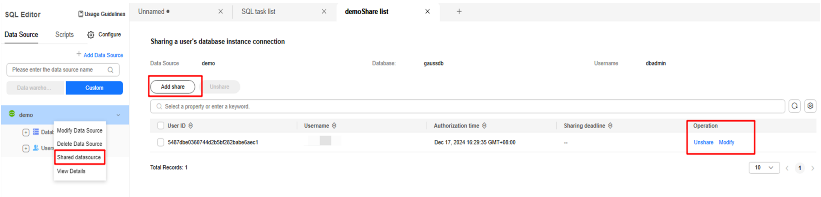
Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Data Source** and click **Custom** to create a custom data source. Remember the password you set during the creation.

Step 4 Right-click the name of the custom data source to be shared and choose **Shared datasource** to view the list of IAM users with whom this data source has been shared.

- Click **Add share** to share the data source with other IAM users of the same tenant and set the sharing expiration time.
- To cancel the sharing, click **Unshare** in the **Operation** column of the row that contains the target IAM user.
- Click **Modify** in the **Operation** column of the row that contains the target IAM user to modify the sharing expiration time and change the username.

Figure 4-23 Sharing a data source



Sharing a user's database instance connection

Data Source demo   Database: gaussdb   Username dbadmin

User ID	Username	Authorization time	Sharing deadline	Operation
5487dbe0360744d2b5bf282babe6aec1	[redacted]	Dec 17, 2024 16:29:35 GMT+08:00	--	Unshare Modify

Total Records: 1   10   1

Step 5 IAM users can easily access the data sources shared with them by viewing the custom data source list and clicking on the respective data source name to log in to the system.

----End

Creating a Directory

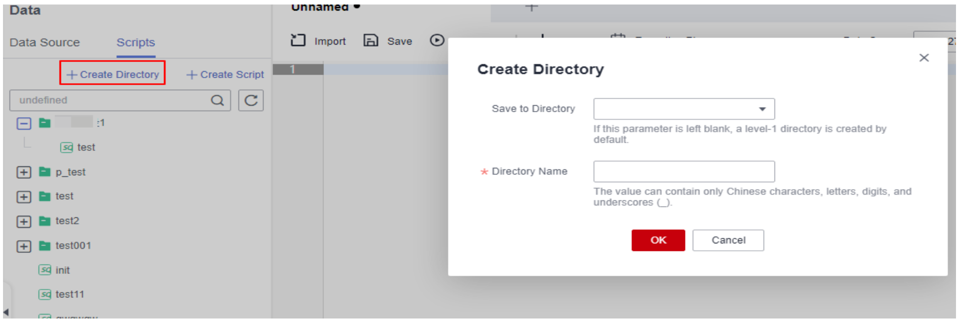
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor** to switch to the script panel.

Step 3 Click **Create Directory**.

- **Save to Directory**: Select a parent directory from the drop-down list box. If this parameter is left blank, a level-1 directory is created by default.
- **Directory Name**: Set the directory name. The value can contain only letters, digits, and underscores (_).

Figure 4-24 Creating a directory



Create Directory

Save to Directory

If this parameter is left blank, a level-1 directory is created by default.

* Directory Name

The value can contain only Chinese characters, letters, digits, and underscores (_).

Step 4 Confirm the information and click **OK**.

----End

Adding a Script

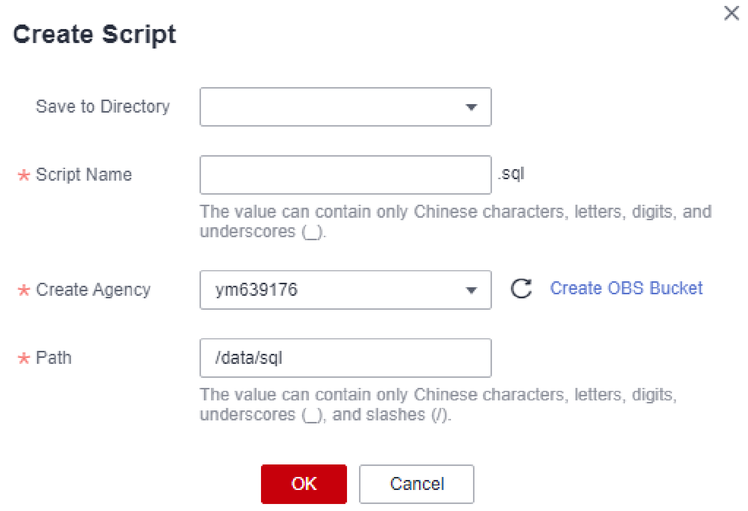
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor** to switch to the script panel.

Step 3 Click **Create Script**.

- **Save to Directory**: Select the new directory from the drop-down list box. This option is optional.
- **Script Name**: Set the script name. Only letters, digits, and underscores (_) are supported.
- **OBS Bucket**: Name of the OBS bucket for storing script files. If no OBS bucket is available, click **View OBS Bucket** to access the OBS console and create one. For details, "Managing Buckets" > "Creating a Bucket" in *Object Storage Service Console Operation Guide*.
- **Path**: User-defined directory for storing script files on OBS. Multi-level directories can be separated by slashes (/). The value is a string containing 1 to 50 characters, which cannot start with a forward slash (/). If you do not set this parameter, the system automatically adds a path by default.

Figure 4-25 Adding a script



Create Script

Save to Directory

* Script Name .sql

The value can contain only Chinese characters, letters, digits, and underscores (_).

* Create Agency: ym639176

* Path: /data/sql

The value can contain only Chinese characters, letters, digits, underscores (_), and slashes (/).

----End

Reference Syntax

- Syntax reference for adding a database: **CREATE DATABASE**
- Syntax reference for adding a schema: **CREATE SCHEMA**
- Syntax reference for adding a common table: **CREATE TABLE**

4.3.1.4 Data Development Settings

Procedure

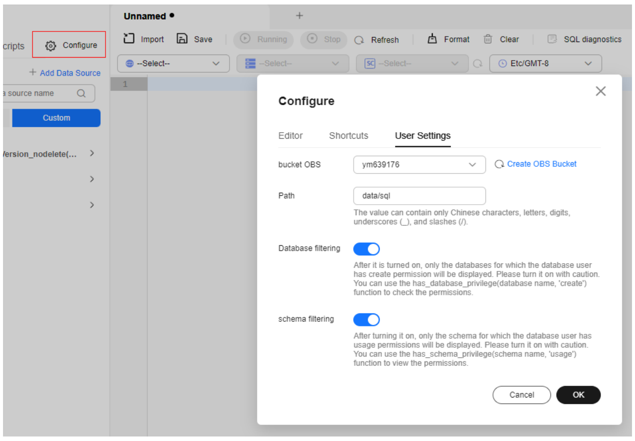
The **Editor** provides basic settings, including the operation bar, shortcut keys, and storage settings. If no OBS bucket is available, you can create one. For details, see **Managing Buckets > Creating a Bucket** in the *Object Storage Service Console Operation Guide*..

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree, choose **Data** > **SQL Editor**.

Step 3 Click **Configure** in the upper right corner and set parameters as required.

Figure 4-26 Data development settings



Configure

bucket OBS: ym639176

Path: data/sql

The value can contain only Chinese characters, letters, digits, underscores (_), and slashes (/).

Database filtering: on

After it is turned on, only the databases for which the database user has create permission will be displayed. Please turn it on with caution. You can use the has_database_privilege(database name, 'create') function to check the permissions.

schema filtering: on

After turning it on, only the schema for which the database user has usage permissions will be displayed. Please turn it on with caution. You can use the has_schema_privilege(schema name, 'usage') function to view the permissions.

Table 4-5 Setting parameters

Settings	Description
Editor	Provides basic settings for compiling SQL statements in the editor. For example, if **Autocomplete** is selected by default, some keywords can be automatically filled. If you select **Autocomplete Table/Column Name**, table or column names can be automatically filled when a table is complied.
Shortcuts	You can use shortcut keys to quickly compile SQL statements in the editor.
User Settings	These settings are required for OBS. You can choose to set a global OBS bucket, and any file directories you create will be saved to this bucket's folder by default. You can also activate database and schema permission filtering, which limits the **create** permission of the database and the **usage** permission of the schema to the current database user.

Step 4 Confirm the information and click **OK**.

----End

4.3.2 Using Data Studio to Connect to a GaussDB(DWS) Cluster

Data Studio is a SQL client tool running on the Windows operating system. It provides various GUIs for you to manage databases and database objects, as well as edit, run, and debug SQL scripts, and view execution plans. Download the Data Studio software package from the GaussDB(DWS) management console. The package can be used without installation after being decompressed.

Data Studio versions include **Windows x86** (32-bit Windows system) and **Windows x64** (64-bit Windows system).

Preparations Before Connecting to a Cluster

- You have obtained the administrator username and password for logging in to the database in the data warehouse cluster.
- You have obtained the public network address, including the IP address and port number in the data warehouse cluster. For details, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**.
- You have configured the security group of the GaussDB(DWS) cluster and added an inbound rule that allows users' IP addresses to access ports using the TCP.

  For details, see **Adding a Security Group Rule** in the *Virtual Private Cloud User Guide*.

Connecting to the Cluster Database Using Data Studio

Step 1 GaussDB(DWS) provides a Windows-based Data Studio client and the tool depends on the JDK. You need to install the JDK on the client host first.

NOTICE

Only JDK 1.8 is supported.

In the Windows operating system, you can download the required JDK version from the **official website of SDK**, and install it by following the installation guidance.

Step 2 Log in to the GaussDB(DWS) console.

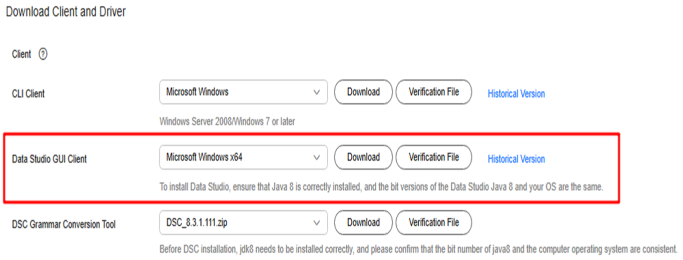
Step 3 Choose **Management** > **Client Connections**.

Step 4 On the **Download Client and Driver** page, download **Data Studio GUI Client**.

- Select **Windows x86** or **Windows x64** based on the OS type and click **Download** to download a Data Studio version that matches the current cluster.

  If clusters of different versions are available, you will download the Data Studio matching the earliest cluster version after clicking **Download**. If there is no cluster, you will download the Data Studio tool of the earliest version after clicking **Download**. GaussDB(DWS) clusters are compatible with earlier versions of Data Studio.
- Click **Historical Version** to download the corresponding Data Studio version. You are advised to download Data Studio based on the cluster version.

Figure 4-27 Downloading clients



Download Client and Driver

Client

CLI Client: Microsoft Windows

Windows Server 2008/Windows 7 or later

Data Studio GUI Client: Microsoft Windows x64

To install Data Studio, ensure that Java 8 is correctly installed, and the bit versions of the Data Studio Java 8 and your OS are the same.

DSC Grammar Conversion Tool: DSC_8.3.1.111.zip

Before DSC installation, jdk8 needs to be installed correctly, and please confirm that the bit number of java8 and the computer operating system are consistent.

If you have clusters of different versions, the system displays a dialog box, prompting you to select the cluster version and download the corresponding client. In the cluster list on the **Clusters** > **Dedicated Clusters** page, click the name of the specified cluster to go to the **Cluster Information** page and view the cluster version.

Table 4-6 Data Studio download links

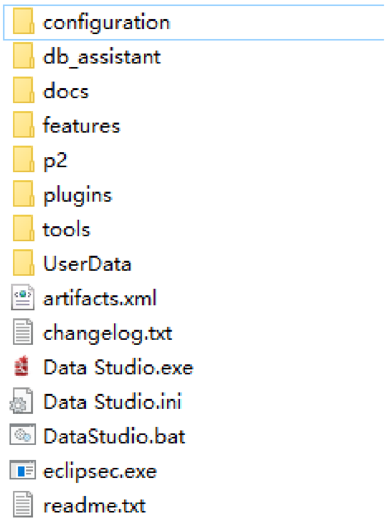
Applicable OS	Download Link	Verification File
Windows x64	**Data_Studio_8.2.x_64.zip**	**Data_Studio_8.2.x_64.zip.sha256**
	Data_Studio_8.1.x_64.zip	**Data_Studio_8.1.x_64.zip.sha256**

Applicable OS	Download Link	Verification File
	Data_Studio_8.0.x_64.zip	Data_Studio_8.0.x_64.zip.sha256
Windows x86	Data_Studio_8.2.x_32.zip	Data_Studio_8.2.x_32.zip.sha256
	Data_Studio_8.1.x_32.zip	Data_Studio_8.1.x_32.zip.sha256
	Data_Studio_8.0.x_32.zip	Data_Studio_8.0.x_32.zip.sha256

Step 5 Decompress the downloaded client software package (32-bit or 64-bit) to the installation directory.

Step 6 Open the installation directory and double-click **Data Studio.exe** to start the Data Studio client. See **Figure 4-28**.

Figure 4-28 Starting the client



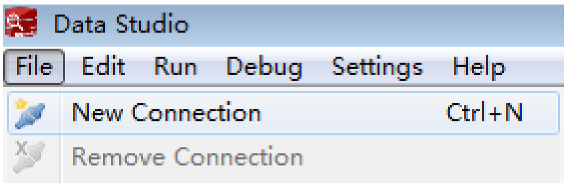
```
configuration
db_assistant
docs
features
p2
plugins
tools
UserData
artifacts.xml
changelog.txt
Data Studio.exe
Data Studio.ini
DataStudio.bat
eclipsec.exe
readme.txt
```

NOTE

If your computer blocks the running of the application, you can unlock the **Data Studio.exe** file to start the application.

Step 7 Choose **File** > **New Connection** from the main menu. See **Figure 4-29**.

Figure 4-29 New connection



Step 8 In the displayed **New Database Connection** window, enter the connection parameters.

Table 4-7 Connection parameters

Field	Description	Example Value
Database Type	Select **HUAWEI CLOUD DWS**.	HUAWEI CLOUD DWS
Connection Name	Name of the connection	DWS-demo
Host	IP address (IPv4) or domain name of the cluster to be connected	-
Port Number	Database port	8000
Database Name	Database name	gaussdb
Username	Username for connecting to the database	-
Password	Password for logging in to the database to be connected	-
Save Password	Select an option from the drop-down list: ● **Current Session Only**: The password is saved only in the current session. ● **Do Not Save**: The password is not saved.	-
Enable SSL	If **Enable SSL** is selected, the client can use SSL to encrypt connections. The SSL connection mode is more secure than common modes, so you are advised to enable SSL connection.	-

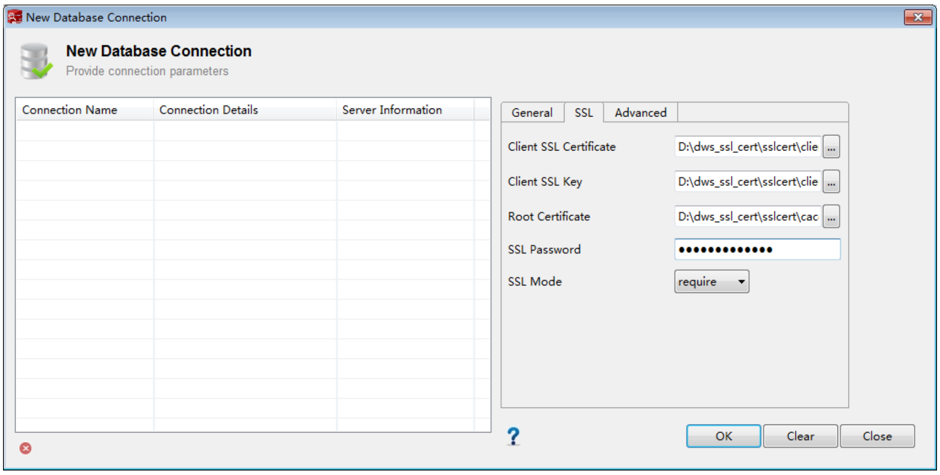
If **Enable SSL** is selected, **download the SSL certificate** and decompress it. Click the **SSL** tab and configure the following parameters:

Table 4-8 Configuring SSL parameters

Field	Description
Client SSL Certificate	Select the **sslcert\client.crt** file in the decompressed SSL certificate directory.
Client SSL Key	Only the PK8 format is supported. Select the **sslcert\client.key.pk8** file in the directory where the SSL certificate is decompressed.

Field	Description
Root Certificate	When **SSL Mode** is set to **verify-ca**, the root certificate must be configured. Select the **sslcert\cacert.pem** file in the decompressed SSL certificate directory.
SSL Cipher	Set the password for the client SSL key in PK8 format.
SSL Mode	GaussDB(DWS) supports the following SSL modes: ● require ● verify-ca GaussDB(DWS) does not support the **verify-full** mode.

Figure 4-30 Configuring SSL parameters



New Database Connection

Provide connection parameters

Client SSL Certificate: D:\dws_ssl_cert\sslcert\clie

Client SSL Key: D:\dws_ssl_cert\sslcert\clie

Root Certificate: D:\dws_ssl_cert\sslcert\cac

SSL Password: •••••••••••••

SSL Mode: require

Step 9 Click **OK** to establish the database connection.

If SSL is enabled, click **Continue** in the displayed **Connection Security Alert** dialog box.

After the login is successful, the **RECENT LOGIN ACTIVITY** dialog box is displayed, indicating that Data Studio is connected to the database. You can run the SQL statement in the **SQL Terminal** window on the Data Studio page.

For details about how to use other functions of Data Studio, press **F1** to view the Data Studio user manual.

NOTE

- Data cannot be rolled back after being added, deleted, modified, or queried in Data Studio.
- Data Studio can save connection information, excluding passwords.
- DDL/DDL and data cannot be exported in batches for the following objects:
  - **Export DDL:**

    Connection, database, foreign table, sequence, column, index, constraint, partition, function/procedure group, regular tables group, views group, schemas group, and system catalog group.
  - **Export DDL and Data:**

    Connection, database, namespace, foreign table, sequence, column, index, constraint, partition, function/procedure, view, regular tables group, schemas group, and system catalog group.

----End

4.4 Using the CLI to Connect to a GaussDB(DWS) Cluster

4.4.1 Downloading the Client

GaussDB(DWS) provides client tool packages that match the cluster versions. You can download the desired client tool package on the GaussDB(DWS) management console. For more information, see **Downloading Client Tools**.

The client tool package contains the following:

- **Linux database connection tool gsql and the script for testing sample data**

  Linux gsql is a Linux command line client running in Linux. It is used to connect to the database in a data warehouse cluster.

  The script for testing sample data is used to execute the introductory example.
- **Windows gsql**

  Windows gsql is a command line client running on the Windows OS. It is used to connect to the database in a data warehouse cluster.

  NOTE

  Only 8.1.3.101 and later cluster versions can be downloaded from the console.
- **GDS tool package**

  Gauss Data Service (GDS) is a data service tool. You can use the GDS tool to import a data file in a common file system to the GaussDB(DWS) database. The GDS tool package must be installed on the server where the data source file is located. The server where the data source file is located is called a data server or GDS server.

Downloading the Client

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Management** > **Client Connections**.

Step 3 Select the GaussDB(DWS) client of the corresponding version from the drop-down list of **CLI Client**.

Choose a corresponding client version according to the cluster version and operating system to which the client is to be installed.

Table 4-9 gsql download links

OS Type	Applicable OS	Download Link	Verification File
Windows	Windows x86_64: ● Windows 7 or later ● Windows Server 2008 or later	**dws_8.1.x_gsql_for_windows.zip**	**dws_8.1.x_gsql_for_windows.zip.sha256**
		dws_8.2.x_gsql_for_windows.zip	**dws_8.2.x_gsql_for_windows.zip.sha256**
Redhat x86_64	RHEL 6.4~7.6	**dws_client_8.2.x_redhat_x64.zip**	**dws_client_8.2.x_redhat_x64.zip.sha256**
		dws_client_8.1.x_redhat_x64.zip	**dws_client_8.1.x_redhat_x64.zip.sha256**
		dws_client_8.0.x_redhat_x64.zip	**dws_client_8.0.x_redhat_x64.zip.sha256**
SUSE x86_64	SLES 11.1~11.4, SLES 12.0~12.3	**dws_client_8.2.x_suse_x64.zip**	**dws_client_8.2.x_suse_x64.zip.sha256**
		dws_client_8.1.x_suse_x64.zip	**dws_client_8.1.x_suse_x64.zip.sha256**
		dws_client_8.0.x_suse_x64.zip	**dws_client_8.0.x_suse_x64.zip.sha256**
Euler Kunpeng_64	EulerOS 2.0 SP8	**dws_client_8.1.x_euler_kunpeng_x64.zip**	**dws_client_8.1.x_euler_kunpeng_x64.zip.sha256**

OS Type	Applicable OS	Download Link	Verification File
Redhat Kunpeng_64	CentOS-7.6-aarch64 and NeoKylin-7.6-aarch64 (adapted to Kunpeng 920 CPU)	**dws_client_8.1.x_redhat_kunpeng_x64.zip**	**dws_client_8.1.x_redhat_kunpeng_x64.zip.sha256**

NOTE

The CPU architecture of the client must be the same as that of the cluster. If the cluster uses x86 servers, select an x86 client.

Step 4 Click **Download** to download the gsql tool matching the 8.1.x cluster version. Click **Historical Version** to download the gsql tool corresponding to the cluster version.

- You are advised to download the gsql tool that matches the cluster version. That is, use gsql 8.1.x for clusters of 8.1.0 or later, and use gsql 8.2.x for clusters of 8.2.0 or later.
- The following table describes the files and folders in the Linux gsql tool package.

Table 4-10 Files and folders in the Linux gsql tool package

File or Folder	Description
bin	This folder holds the Linux executable files for gsql, which include tools gsql, GDS, gs_dump, gs_dumpall, and gs_restore. For details, see **Server Tool**.
gds	This folder contains the files of the GDS data service tool. The GDS tool is used for parallel data loading and can import the data files stored in a common file system to a GaussDB(DWS) database.
lib	This folder contains the **lib** library required for executing the gsql client.
sample	This folder contains the following directories and files: – **setup.sh**: script file for configuring the AK/SK before using gsql to import sample data – **tpcds_load_data_from_obs.sql**: script file for importing the TPC-DS sample data using the gsql client – **query_sql** directory: script file for querying the TPC-DS sample data
gsql_env.sh	Script file for configuring environment variables before running the gsql client.

- The following table describes the files and folders in the Windows gsql tool package.

Table 4-11 Files and folders in the Windows gsql tool package

File or Folder	Description
x64	This folder contains the 64-bit Windows gsql execution binary file and the dynamic library.
x86	This folder contains the 32-bit Windows gsql execution binary file and the dynamic library.

NOTE

In the cluster list on the **Clusters** > **Dedicated Clusters** page, click the name of the specified cluster to go to the **Cluster Information** page and view the cluster version.

----End

4.4.2 Using the Linux gsql Client to Connect to a Cluster

This section describes how to connect to a database through an SQL client after you create a data warehouse cluster and before you use the cluster's database. GaussDB(DWS) provides the Linux gsql client that matches the cluster version for you to access the cluster through the cluster's public or private network address.

The gsql command line client provided by GaussDB(DWS) runs on Linux. Before using it to remotely connect to a GaussDB(DWS) cluster, you need to prepare a Linux server for installing and running the gsql client. If you use a public network address to access the cluster, you can install the Linux gsql client on your own Linux server. Ensure that the Linux server has a public network address. If no EIPs are configured for your GaussDB(DWS) cluster, you are advised to create a Linux ECS for convenience purposes. For more information, see **(Optional) Preparing an ECS as the gsql Client Server**.

(Optional) Preparing an ECS as the gsql Client Server

For details about how to purchase an ECS, see **Purchasing an ECS** in the *Elastic Cloud Server Getting Started*.

The created ECS must meet the following requirements:

- ECS and GaussDB(DWS) clusters must belong to the same region and AZ.
- If you use the gsql client provided by GaussDB(DWS) to connect to the GaussDB(DWS) cluster, the ECS image must meet the following requirements:

  The image's OS must be one of the following Linux OSs supported by the gsql client:

  – The **Redhat x86_64** client can be used on the following OSs:

    ■ RHEL 6.4~7.6

    ■ CentOS 6.4~7.4

      - EulerOS 2.3
  - The **SUSE x86_64** client can be used on the following OSs:
      - SLES 11.1~11.4
      - SLES 12.0~12.3
  - The **Euler Kunpeng_64** client can be used on the following OS:
      - EulerOS 2.8
  - The **Stream Euler x86_64** client can be used on the following OS: EulerOS 2.2
  - The **Stream Euler Kunpeng_64** client can be used on the following OS:
      - EulerOS 2.8

- If the client accesses the cluster using the private network address, ensure that the created ECS is in the same VPC as the GaussDB(DWS) cluster.

  For details about VPC operations, see **VPC and Subnet** in the *Virtual Private Cloud User Guide*.

- If the client accesses the cluster using the public network address, ensure that both the created ECS and GaussDB(DWS) cluster have an EIP.

  When purchasing an ECS, set **EIP** to **Buy now** or **Specify**.

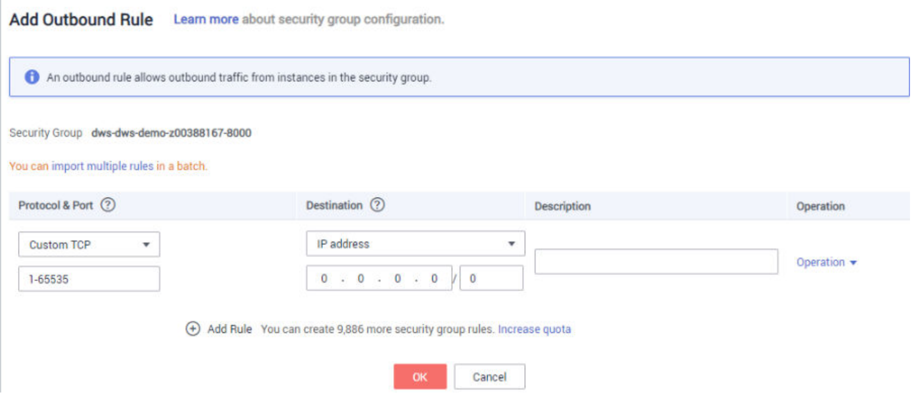
- The security group rules of the ECS must enable communication between the ECS and the port that the GaussDB(DWS) cluster uses to provide services.

  For details about security group operations, see **Security Group** in the *Virtual Private Cloud User Guide*.

  Ensure that the security group of the ECS contains rules meeting the following requirements. If the rules do not exist, add them to the security group:

  - **Transfer Direction**: **Outbound**
  - Protocol: The protocol must contain TCP. For example, **TCP** or **All**.
  - **Port**: The value must contain the database port that provides services in the GaussDB(DWS) cluster. For example, set this parameter to **1-65535** or a specific GaussDB(DWS) database port.
  - Destination: The IP address set here must contain the IP address of the GaussDB(DWS) cluster to be connected. **0.0.0.0/0** indicates any IP address.

Figure 4-31 Outbound rule



Add Outbound Rule   Learn more about security group configuration.

An outbound rule allows outbound traffic from instances in the security group.

Security Group   dws-dws-demo-z00388167-8000

You can import multiple rules in a batch.

Protocol & Port	Destination	Description	Operation
Custom TCP 1-65535	IP address 0 . 0 . 0 . 0 / 0		Operation

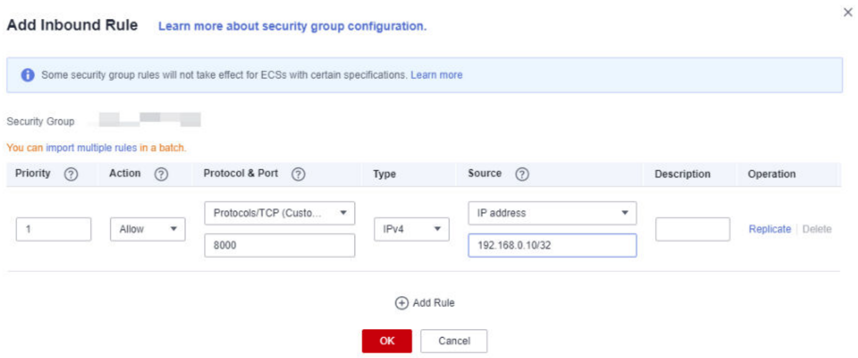
Add Rule   You can create 9,886 more security group rules. Increase quota

- The security group rules of the data warehouse cluster must ensure that GaussDB(DWS) can receive network access requests from clients.

  Ensure that the cluster's security group contains rules meeting the following requirements. If the rules do not exist, add them to the security group:

  - **Transfer Direction**: **Inbound**
  - **Protocol**: The protocol must contain TCP. For example, **TCP** or **All**.
  - **Port**: Set this parameter to the servicing database port of the GaussDB(DWS) cluster. Example: 8000.
  - Source IP Address: The IP address set here must contain the IP address of the GaussDB(DWS) client host. Example: 192.168.0.10/32.

Figure 4-32 Inbound rule



Add Inbound Rule Learn more about security group configuration.

Some security group rules will not take effect for ECSs with certain specifications. Learn more

Security Group [redacted]

You can import multiple rules in a batch.

Priority	Action	Protocol & Port	Type	Source	Description	Operation
1	Allow	Protocols/TCP (Custo... 8000	IPv4	IP address 192.168.0.10/32		Replicate | Delete

Downloading the Linux gsql Client and Connecting to a Cluster

Step 1 Download the Linux gsql client by referring to **Downloading the Client**, and use an SSH file transfer tool (such as WinSCP) to upload the client to a target Linux server.

You are advised to download the gsql tool that matches the cluster version. That is, use gsql 8.1.x for clusters of 8.1.0 or later, and use gsql 8.2.x for clusters of 8.2.0 or later. To download gsql 8.2.x, replace **dws_client_8.1.x_redhat_x64.zip** with **dws_client_8.2.x_redhat_x64.zip**. The **dws_client_8.1.x_redhat_x64.zip** is used as an example.

The user who uploads the client must have the full control permission on the target directory on the host to which the client is uploaded.

Alternatively, you can remotely manage the Linux server where the gsql is to be installed in SSH mode and run the following command in the Linux command window to download the Linux gsql client:

```
wget https://obs.ap-southeast-1.myhuaweicloud.com/dws/download/dws_client_8.1.x_redhat_x64.zip --no-check-certificate
```

Step 2 Use the SSH tool to remotely manage the host where the client is installed.

For details about how to log in to an ECS, see **Login Using an SSH Password** in the *Elastic Cloud Server User Guide*.

Step 3 (Optional) To connect to the cluster in SSL mode, configure SSL authentication parameters on the host where the client is installed. For details, see **Establishing Secure TCP/IP Connections in SSL Mode**.

NOTE

The SSL connection mode is more secure than the non-SSL mode. You are advised to connect the client to the cluster in SSL mode.

Step 4 Run the following commands to decompress the client:

```
cd <Path for saving the client>
unzip dws_client_8.1.x_redhat_x64.zip
```

In the preceding commands:

- *<Path_for_storing_the_client>*: Replace it with the actual path.
- *dws_client_8.1.x_redhat_x64.zip*: This is the client tool package name of **RedHat x86**. Replace it with the actual name.

Step 5 Run the following command to configure the GaussDB(DWS) client:

```
source gsql_env.sh
```

If the following information is displayed, the gsql client is successfully configured:

```
All things done.
```

Step 6 Connect to the database in the GaussDB(DWS) cluster using the gsql client. Replace the values of each parameter with actual values.

```
gsql -d <Database_name> -h <Cluster_address> -U <Database_user> -p <Database_port> -W
<Cluster_password> -r
```

The parameters are described as follows:

- *Database_name*: Enter the name of the database to be connected. If you use the client to connect to the cluster for the first time, enter the default database **gaussdb**.
- *Cluster_address*: For details about how to obtain this address, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**. If a public network address is used for connection, set this parameter to **Public Network Address** or **Public Network Domain Name**. If a private network address is used for connection, set this parameter to **Private Network Address** or **Private Network Domain Name**. If ELB is used for connection, set this parameter to **ELB Address**.
- *Database_user*: Enter the username of the cluster's database. If you use the client to connect to the cluster for the first time, set this parameter to the default administrator configured during cluster creation, for example, **dbadmin**.
- *Database_port*: Enter the database port set during cluster creation.

For example, run the following command to connect to the default database **gaussdb** in the GaussDB(DWS) cluster:

```
gsql -d gaussdb -h 10.168.0.74 -U dbadmin -p 8000 -W password -r
```

If the following information is displayed, the connection succeeded:

```
gaussdb=>
```

----End

gsql Command Reference

For more information about the gsql commands, see the ***Data Warehouse Service (DWS) Tool Guide***.

(Optional) Importing TPC-DS Sample Data Using gsql

GaussDB(DWS) users can import data from external sources to data warehouse clusters. This section describes how to import sample data from OBS to a data warehouse cluster and perform querying and analysis operations on the sample data. The sample data is generated based on the standard TPC-DS benchmark test.

TPC-DS is the benchmark for testing the performance of decision support. With TPC-DS test data and cases, you can simulate complex scenarios, such as big data set statistics, report generation, online query, and data mining, to better understand functions and performance of database applications.

NOTE

Currently, TPC-DS sample data can be imported only in the CN North-Beijing1 region.

Step 1 Use the SSH remote connection tool to log in to the server where the gsql client is installed and go to the gsql directory. The **/opt** directory is used as an example for storing the gsql client.

```
cd /opt
```

Step 2 Switch to the specified directory and set the AK and SK for importing sample data and the OBS access address.

```
cd sample
/bin/bash setup.sh -ak <Access_Key_Id> -sk <Secret_Access_Key> -obs_location obs.ap-
southeast-1.myhuaweicloud.com
```

If the following information is displayed, the settings are successful:

```
setup successfully!
```

NOTE

<Access_Key_Id> and *<Secret_Access_Key>*: indicate the AK and SK, respectively. For how to obtain the AK and SK, see **Creating Access Keys (AK and SK)**. Replace the parameters in the statements with the obtained values.

Step 3 Go back to previous directory and run the gsql environment variables.

```
cd ..
source gsql_env.sh
cd bin
```

Step 4 Import the sample data to the data warehouse.

Command format:

```
gsql -d <Database name> -h <Public network address of the cluster> -U <Administrator> -p <Data
warehouse port number> -f <Path for storing the sample data script> -r
```

Sample command:

```
gsql -d gaussdb -h 10.168.0.74 -U dbadmin -p 8000 -f /opt/sample/tpcds_load_data_from_obs.sql -r
```

NOTE

In the preceding command, sample data script **tpcds_load_data_from_obs.sql** is stored in the sample directory (for example, **/opt/sample/**) of the GaussDB(DWS) client.

After you enter the administrator password and successfully connect to the database in the cluster, the system will automatically create a foreign table to associate the sample data outside the cluster. Then, the system creates a target table for saving the sample data and imports the data to the target table using the foreign table.

The time required for importing a large dataset depends on the current GaussDB(DWS) cluster specifications. Generally, the import takes about 10 to 20 minutes. If information similar to the following is displayed, the import is successful.

```
Time:1845600.524 ms
```

Step 5 In the Linux command window, run the following commands to switch to a specific directory and query the sample data:

```
cd /opt/sample/query_sql/
/bin/bash tpcds100x.sh
```

Step 6 Enter the cluster's public network IP address, access port, database name, user who accesses the database, and password of the user as prompted.

- The default database name is **gaussdb**.
- Use the administrator username and password configured during cluster creation as the username and password for accessing the database.

After the query is complete, a directory for storing the query result, such as **query_output_20170914_072341**, will be generated in the current query directory, for example, **sample/query_sql/**.

----End

4.4.3 Using the Windows gsql Client to Connect to a Cluster

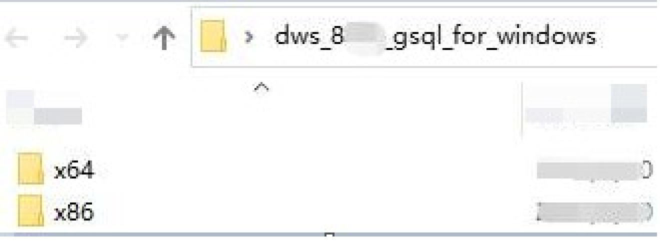
This section describes how to connect to a database through an SQL client after you create a data warehouse cluster and before you use the cluster's database. GaussDB(DWS) provides the Windows gsql client that matches the cluster version for you to access the cluster through the cluster's public or private network address.

Procedure

Step 1 Install and run the gsql client on the local Windows server (in Windows CLI). Windows Server 2008/Windows 7 and later are supported.

Step 2 Download the Windows gsql client by referring to **Downloading the Client** and decompress the package to a local folder.

Figure 4-33 Windows gsql client folder



Step 3 On the local server, click **Start**, search for **cmd**, and run the program as the administrator. Alternatively, press **Win**+**R** to open the Windows CLI.

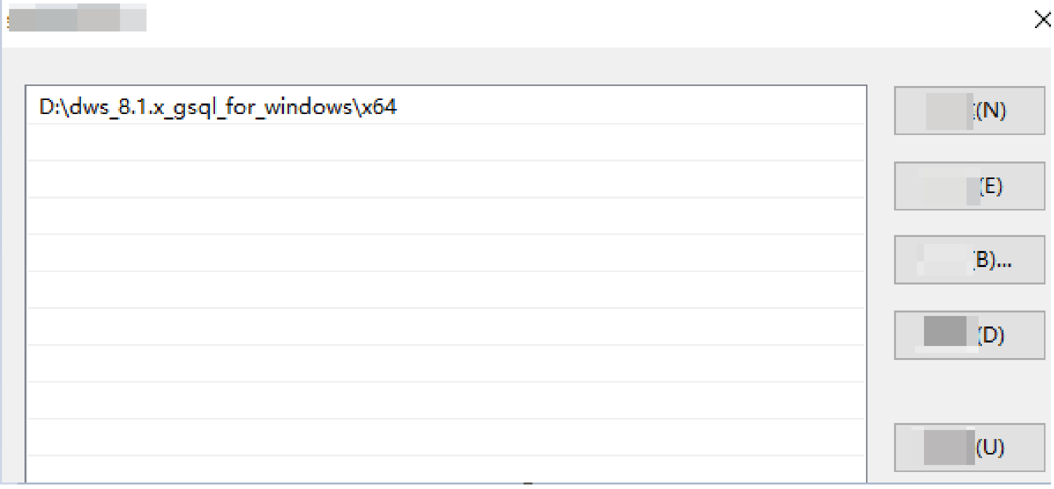
Step 4 Set environment variables. For a 32-bit OS, select the **x86** folder. For a 64-bit OS, select the **x64** folder.

Method 1: Configure environment variables in the Windows CLI. Open the command prompt and run the **set path=***<window_gsql>***;%path%** command, where *<window_gsql>* indicates the folder path where the Windows gsql client was decompressed to in the previous step. For example:

```
set path=C:\Users\xx\Desktop\dws_8.1.x_gsql_for_windows\x64;%path%
```

Method 2: In the **Control Panel** window, search for **System** and click **View advanced system settings**. Click the **Advanced** tab, and click **Environment Variables**. Select the **Path** parameter and click **Edit**. Add the gsql path in the parameter value. For example:

Figure 4-34 Configuring Windows environment variables


D:\dws_8.1.x_gsql_for_windows\x64

Step 5 (Optional) To connect to the cluster in SSL mode, configure SSL authentication parameters on the server where the client is installed. For details, see **Establishing Secure TCP/IP Connections in SSL Mode**.

NOTE

The SSL connection mode is more secure than the non-SSL mode. You are advised to connect the client to the cluster in SSL mode.

Step 6 In the Windows CLI, run the following command to connect to the database in the GaussDB(DWS) cluster using the gsql client:

```
gsql -d <Database_name> -h <Cluster_address> -U <Database_user> -p <Database_port> -W
<Cluster_password> -r
```

The parameters are as follows:

- **Database name**: Enter the name of the database to be connected. If you use the client to connect to the cluster for the first time, enter the default database **gaussdb**.
- **Cluster address**: For details about how to obtain this address, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**. If a public network address is used for connection, set this parameter to the public network domain name. If a private network address is used for connection, set this parameter to the private network domain name. If ELB is used for connection, set this parameter to **ELB Address**.
- **Database user**: Enter the username of the cluster's database. If you use the client to connect to the cluster for the first time, set this parameter to the default administrator configured during cluster creation, for example, **dbadmin**.
- **Database port**: Enter the database port set during cluster creation.

For example, run the following command to connect to the default database **gaussdb** in the GaussDB(DWS) cluster:

```
gsql -d gaussdb -h 10.168.0.74 -U dbadmin -p 8000 -W password -r
```

If the following information is displayed, the connection succeeded:

```
gaussdb=>
```

----End

Precautions

1. The default character encoding of the Windows command prompt is GBK, and the default value of **client_encoding** of Windows gsql is **GBK**. Some characters encoded using UTF-8 cannot be displayed in Windows gsql.

   Suggestion: Ensure the file specified using **-f** uses UTF-8 encoding, and set the default encoding format to **UTF-8** (**set client_encoding='utf-8';**).

2. Paths in Windows gsql must be separated by slashes (/), or an error will be reported. In a meta-command, the backslash (\) indicates the start of a meta-command. If the backslash is enclosed in single quotation marks ('\'), it is used for escape.

   ```
   gaussdb=> \i D:\test.sql
   D:: Permission denied
   postgres=> \i D:/test.sql
   id
   ----
     1
   (1 row)
   ```

3. To use the **\!** meta command to run a system command in Windows gsql, be sure to use the path separator required by the system command. Generally, the path separator is a backslash (\).

   ```
   gaussdb=> \! type D:/test.sql
   Incorrect syntax.
   gaussdb=> \! type D:\test.sql
   select 1 as id;
   ```

4. Windows gsql does not support the **\parallel** meta-command.

   ```
   gaussdb=> \parallel
   ERROR: "\parallel" is not supported in Windows.
   ```

5. In Linux shell, single quotation marks ('') and double quotation marks ("") can be used to enclose strings. In Windows, only double quotation marks can be used.

```
gsql -h 192.168.233.189 -p 8109 -d postgres -U odbcuser -W password -c "select 1 as id"
 id
----
  1
(1 row)
```

   If single quotation marks are used, an error will be reported and the input will be ignored.

```
gsql -h 192.168.233.189 -p 8109 -d postgres -U odbcuser -W password -c 'select 1 as id'
gsql: warning: extra command-line argument "1" ignored
gsql: warning: extra command-line argument "as" ignored
gsql: warning: extra command-line argument "id'" ignored
ERROR:  unterminated quoted string at or near "'select"
LINE 1: 'select
```

6. If Windows gsql is idle for a long time after a connection is established, the connection session times out, and an SSL error is reported. In this case, you need to log in again. The following error is reported:

```
SSL SYSCALL error: Software caused connection abort (0x00002745/10053), remote datanode
<NULL>, error: Result too large
```

7. In Windows, press **Ctrl**+**C** to exit gsql. If **Ctrl**+**C** are pressed during input, the input will be ignored and you will be forced to exit gsql.

   Enter **as** and press **Ctrl**+**C**. After **\q** is displayed, exit gsql.

```
gaussdb=> select 1
gaussdb=> as \q
```

8. Windows gsql cannot connect to a database using the LATIN1 character encoding. The error information is as follows:

```
gsql: FATAL: conversion between GBK and LATIN1 is not supported
```

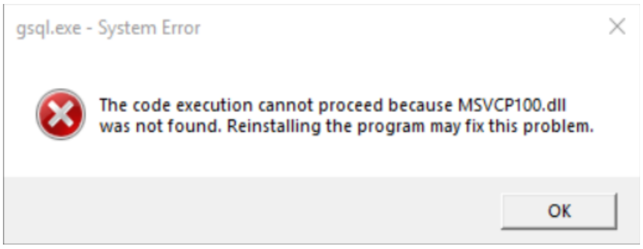
9. The location of the **gsqlrc.conf** file:

   The default **gsqlrc** path is **%APPDATA%/postgresql/gsqlrc.conf**. You can also set the path using the **PSQLRC** variable.

```
set PSQLRC=C:\Users\xx\Desktop\dws_8.1.x_gsql_for_windows\x64\gsqlrc.conf
```

10. **MSVCP100.dll** may be missing in the Windows Server system. When you use **gsql**, the following error message is displayed.

   Figure 4-35 Error message



gsql.exe - System Error

The code execution cannot proceed because MSVCP100.dll was not found. Reinstalling the program may fix this problem.

   Solution: Add the **MSVCP100.dll** file. You can download the C++ redistributable program package and install the **vcredist_x86.exe/ vcredist_x64.exe** package to supplement the required dynamic link library file.

gsql Command Reference

For more information about the gsql commands, see the ***Data Warehouse Service (DWS) Tool Guide***.

4.4.4 Establishing Secure TCP/IP Connections in SSL Mode

GaussDB(DWS) supports the standard SSL. As a highly secure protocol, SSL authenticates bidirectional identification between the server and client using digital signatures and digital certificates to ensure secure data transmission. To support SSL connection, GaussDB(DWS) has obtained the formal certificates and keys for the server and client from the CA certification center. It is assumed that the key and certificate for the server are **server.key** and **server.crt** respectively; the key and certificate for the client are **client.key** and **client.crt** respectively, and the name of the CA root certificate is **cacert.pem**.

The SSL connection mode is more secure. By default, the SSL feature in a cluster allows SSL and non-SSL connections from the client. For security purposes, you are advised to connect to the cluster via SSL from the client. Ensure the certificate, private key, and root certificate of the GaussDB(DWS) server have been configured by default. To forcibly use an SSL connection, configure the **require_ssl** parameter in the **Require SSL Connection** area of the cluster's **Security Settings** page on the GaussDB(DWS) management console. Require SSL Connection on the Security Settings page of the cluster. For more information, see **Configuring SSL Connection** and **Combinations of SSL Connection Parameters on the Client and Server**.

The client or JDBC/ODBC driver needs to use SSL connection. Configure related SSL connection parameters in the client or application code. The GaussDB(DWS) management console provides the SSL certificate required by the client. The SSL certificate contains the default certificate, private key, root certificate, and private key password encryption file required by the client. Download the SSL certificate to the host where the client is installed, and specify the path of the certificate on the client. For more information, see **Configuring Digital Certificate Parameters Related to SSL Authentication on the gsql Client** and **SSL Authentication Modes and Client Parameters**.

NOTE

Using the default certificate may pose security risks. To improve system security, you are advised to periodically change the certificate to prevent password cracking. If you need to replace the certificate, contact the database customer service.

Configuring SSL Connection

Prerequisites

- Changes made to security configuration parameters require a cluster restart to take effect. Otherwise, the cluster will be temporarily unavailable.
- To modify the cluster's security configuration, ensure that the following conditions are met:
  - The cluster status is **Available** or **Unbalanced**.
  - The **Task Information** cannot be set to **Creating snapshot**, **Scaling out**, **Configuring**, or **Restarting**.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of a cluster. On the page that is displayed, click **Security Settings**.

By default, **Configuration Status** is **Synchronized**, which indicates that the latest database result is displayed.

Step 4 In the **SSL Connection** area, enable **Require SSL Connection** (recommended).

indicates the function is enabled. The **require_ssl** is set to **1**, indicating that the server forcibly requires the SSL connection.

indicates the function is disabled (default value). The **require_ssl** parameter is set to **0**, indicating that the server does not require SSL connections. For details about how to configure the **require_ssl** parameter, see **require_ssl (Server)**.

NOTE

- If the gsql client or ODBC driver provided by GaussDB(DWS) is used, GaussDB(DWS) supports the TLSv1.2 SSL protocol.
- If the JDBC driver provided by GaussDB(DWS) is used, GaussDB(DWS) supports SSL protocols, such as SSLv3, TLSv1, TLSv1.1, and TLSv1.2. The SSL protocol used between the client and the database depends on the Java Development Kit (JDK) version used by the client. Generally, JDK supports multiple SSL protocols.

Step 5 Click **Apply**.

The system automatically saves the SSL connection settings. On the **Security Settings** page, **Configuration Status** is **Applying**. After **Configuration Status** changes to **Synchronized**, the settings have been saved and taken effect.

----End

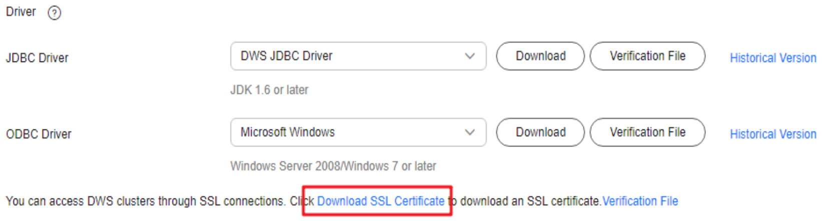
Configuring Digital Certificate Parameters Related to SSL Authentication on the gsql Client

After a data warehouse cluster is deployed, the SSL authentication mode is enabled by default. The server certificate, private key, and root certificate have been configured by default. You need to configure the client parameters.

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Client Connections**.

Step 2 In the **Driver** area, click **download an SSL certificate**.

Figure 4-36 Downloading an SSL certificate



Driver

JDBC Driver: DWS JDBC Driver

JDK 1.6 or later

ODBC Driver: Microsoft Windows

Windows Server 2008/Windows 7 or later

You can access DWS clusters through SSL connections. Click Download SSL Certificate to download an SSL certificate.Verification File

Step 3 Use a file transfer tool (such as WinSCP) to upload the SSL certificate to the host where the client is installed.

For example, save the downloaded certificate **dws_ssl_cert.zip** to the **/home/dbadmin/dws_ssl/** directory.

Step 4 Use an SSH remote connection tool (such as PuTTY) to log in to the host where the gsql client is installed and run the following commands to go to the directory where the SSL certificate is stored and decompress the SSL certificate:

```
cd /home/dbadmin/dws_ssl/
unzip dws_ssl_cert.zip
```

Step 5 Run the export command and configure digital certificate parameters related to SSL authentication on the host where the gsql client is installed.

There are two SSL authentication modes: bidirectional authentication and unidirectional authentication. The client environment variables to be configured vary according to the authentication mode. For details, see **SSL Authentication Modes and Client Parameters**.

The following parameters must be configured for bidirectional authentication:

```
export PGSSLCERT="/home/dbadmin/dws_ssl/sslcert/client.crt"
export PGSSLKEY="/home/dbadmin/dws_ssl/sslcert/client.key"
export PGSSLMODE="verify-ca"
export PGSSLROOTCERT="/home/dbadmin/dws_ssl/sslcert/cacert.pem"
```

The following parameters must be configured for unidirectional authentication:

```
export PGSSLMODE="verify-ca"
export PGSSLROOTCERT="/home/dbadmin/dws_ssl/sslcert/cacert.pem"
```

NOTICE

- You are advised to use bidirectional authentication for security purposes.
- The environment variables configured for a client must contain the absolute file paths.

Step 6 Change the client private key permissions.

The permissions on the client's root certificate, private key, certificate, and encrypted private key file must be **600**. If the permissions do not meet the requirement, the client cannot connect to the cluster in SSL mode.

```
chmod 600 client.key
chmod 600 client.crt
chmod 600 client.key.cipher
chmod 600 client.key.rand
chmod 600 cacert.pem
```

----End

SSL Authentication Modes and Client Parameters

There are two SSL authentication modes: bidirectional authentication and unidirectional authentication. Table **Table 4-12** shows the differences between these two modes. You are advised to use bidirectional authentication for security purposes.

Table 4-12 Authentication modes

Authentication Mode	Description	Environment Variables Configured on a Client	Maintenance
Bidirectional authentication (recommended)	The client verifies the server's certificate and the server verifies the client's certificate. The connection can be set up only after the verifications are successful.	Set the following environment variables: ● PGSSLCERT ● PGSSLKEY ● PGSSLROOTCERT ● PGSSLMODE	This authentication mode is applicable to scenarios that require high data security. When using this mode, you are advised to set the **PGSSLMODE** client variable to **verify-ca** for network data security purposes.
Unidirectional authentication	The client verifies the server's certificate, whereas the server does not verify the client's certificate. The server loads the certificate information and sends it to the client. The client verifies the server's certificate according to the root certificate.	Set the following environment variables: ● PGSSLROOTCERT ● PGSSLMODE	To prevent TCP-based security attacks, you are advised to use the SSL certificate authentication. In addition to configuring the client root certificate, you are advised to set the **PGSSLMODE** variable to **verify-ca** on the client.

Configure environment variables related to SSL authentication on the client. For details, see **Table 4-13**.

NOTE

The path of environment variables is set to */home/dbadmin/***dws_ssl/** as an example. Replace it with the actual path.

Table 4-13 Client parameters

Environment Variable	Description	Value Description
PGSSLCERT	Specifies the certificate files for a client, including the public key. Certificates prove the legal identity of the client and the public key is sent to the remote end for data encryption.	The absolute path of the files must be specified, for example: `export PGSSLCERT='/home/dbadmin/dws_ssl/sslcert/client.crt'` (No default value)
PGSSLKEY	Specifies the client private key file used to decrypt the digital signatures and the data encrypted using the public key.	The absolute path of the files must be specified, for example: `export PGSSLKEY='/home/dbadmin/dws_ssl/sslcert/client.key'` (No default value)
PGSSLMODE	Specifies whether to negotiate with the server about SSL connection and specifies the priority of the SSL connection.	Values and meanings: ● **disable**: only tries to establish a non-SSL connection. ● **allow**: tries to establish a non-SSL connection first, and then an SSL connection if the first attempt fails. ● **prefer**: tries to establish an SSL connection first, and then a non-SSL connection if the first attempt fails. ● **require**: only tries to establish an SSL connection. If there is a CA file, perform the verification according to the scenario in which the parameter is set to **verify-ca**. ● **verify-ca**: tries to establish an SSL connection and check whether the server certificate is issued by a trusted CA. ● **verify-full**: GaussDB(DWS) does not support this mode. Default value: **prefer** **NOTE** When an external client accesses a cluster, the error message "ssl SYSCALL error" is displayed on some nodes. In this case, run **export PGSSLMODE="allow"** or **export PGSSLMODE="prefer"**.

Environment Variable	Description	Value Description
PGSSLROOTCERT	Specifies the root certificate file for issuing client certificates. The root certificate is used to verify the server certificate.	The absolute path of the files must be specified, for example: `export PGSSLROOTCERT='/home/dbadmin/dws_ssl/sslcert/certca.pem'` Default value: null
PGSSLCRL	Specifies the certificate revocation list file, which is used to check whether a server certificate is in the list. If the certificate is in the list, it is invalid.	The absolute path of the files must be specified, for example: `export PGSSLCRL='/home/dbadmin/dws_ssl/sslcert/sslcrl-file.crl'` Default value: null

Combinations of SSL Connection Parameters on the Client and Server

Whether the client uses the SSL encryption connection mode and whether to verify the server certificate depend on client parameter **sslmode** and server (cluster) parameters **ssl** and **require_ssl**. The parameters are as follows:

- **ssl (Server)**

  The **ssl** parameter indicates whether to enable the SSL function. **on** indicates that the function is enabled, and **off** indicates that the function is disabled.

  - The default value is **on** and you cannot set this parameter on the GaussDB(DWS) console.

- **require_ssl (Server)**

  The **require_ssl** parameter specifies whether the server forcibly requires SSL connection. This parameter is valid only when **ssl** is set to **on**. **on** indicates that the server forcibly requires SSL connection. **off** indicates that the server does not require SSL connection.

  - The default value is **off**. You can set the **require_ssl** parameter in the **Require SSL Connection** area of the cluster's **Security Settings** page on the GaussDB(DWS) console.

- **sslmode (Client)**

  You can set this parameter in the SQL client tool.

  - In the gsql command line client, this parameter is the **PGSSLMODE** parameter.
  - On the Data Studio client, this parameter is the **SSL Mode** parameter.

The combinations of client parameter **sslmode** and server parameters **ssl** and **require_ssl** are as follows.

Table 4-14 Combinations of SSL connection parameters on the client and server

ssl (Server)	sslmode (Client)	require_ssl (Server)	Result
on	disable	on	The server requires SSL, but the client disables SSL for the connection. As a result, the connection cannot be set up.
	disable	off	The connection is not encrypted.
	allow	on	The connection is encrypted.
	allow	off	The connection is not encrypted.
	prefer	on	The connection is encrypted.
	prefer	off	The connection is encrypted.
	require	on	The connection is encrypted.
	require	off	The connection is encrypted.
	verify-ca	on	The connection is encrypted and the server certificate is verified.
	verify-ca	off	The connection is encrypted and the server certificate is verified.
off	disable	on	The connection is not encrypted.
	disable	off	The connection is not encrypted.
	allow	on	The connection is not encrypted.
	allow	off	The connection is not encrypted.
	prefer	on	The connection is not encrypted.
	prefer	off	The connection is not encrypted.
	require	on	The client requires SSL, but SSL is disabled on the server. Therefore, the connection cannot be set up.
	require	off	The client requires SSL, but SSL is disabled on the server. Therefore, the connection cannot be set up.
	verify-ca	on	The client requires SSL, but SSL is disabled on the server. Therefore, the connection cannot be set up.
	verify-ca	off	The client requires SSL, but SSL is disabled on the server. Therefore, the connection cannot be set up.

4.5 Using a Third-Party Database Adapter for GaussDB(DWS) Cluster Connection

4.5.1 Using the JDBC and ODBC Drivers to Connect to a Cluster

4.5.1.1 Development Specifications

If the connection pool mechanism is used during application development, comply with the following specifications: If you do not do so, the status of connections in the connection pool will remain, which affects subsequent operations using the connection pool.

- If the GUC parameter is set in a connection, you must execute **SET SESSION AUTHORIZATION DEFAULT;RESET ALL;** to clear the connection status before returning the connection to the connection pool.
- If a temporary table is used, it must be deleted before the connection is returned to the connection pool.

4.5.1.2 JDBC Version Description

Version 8.3.1.200

New features

- JDBC load balancing can automatically detect the live CN list.
  - The **cnListRefreshSwitch** parameter is added to enable automatic detection of the live CN list. The default value is **off**.
  - The **cnListRefreshDelay** parameter is added to specify the delay for enabling automatic detection of the live CN list. The default value is 1,800,000 milliseconds. This parameter only works when **cnListRefreshSwitch** is set to **on**.
  - The **cnListRefreshPeriod** parameter is added to specify the period for automatically detecting the live CN list. The default value is 1,800,000 milliseconds. This parameter only works when **cnListRefreshSwitch** is set to **on**.
- JDBC supports certificate revocation.

  The **sslCrl** parameter is added to specify the path of the revoked certificate. By default, this parameter is left blank.
- JDBC supports database reconnection.
  - The **autoReconnect** parameter is added to specify whether to enable automatic database reconnection. The default value is **false**.
  - The **reConnectCount** parameter is added to specify the number of automatic reconnection times. The default value is **10**. This parameter takes effect when **autoReconnect** is set to **true**. If the number of

connection attempts exceeds this parameter's value, reconnection will fail.

Version 8.3.0.202

- New features:

  The **tcpKeepAlive** configuration is added. When **tcpKeepAlive** is set to **true**, the following parameters take effect:

  Default value:

  a. **TCP_KEEPIDLE=30**: The detection starts after the connection is idle for 30 seconds.

  b. **TCP_KEEPCOUNT=9**: A total of nine detections are performed.

  c. **TCP_KEEPINTERVAL=30**: The detection interval is 30s.

  NOTE

  The JDK varies according to the operating system. Some platforms, such as Windows, Red Hat, and SUSE, may not support this parameter.

- Fixed vulnerability:

  CVE-2024-1597

Version 8.3.0.201

- Fixed bug:

  Multiple functions cannot be automatically split when they are executed at a time.

Version 8.3.0

- Fixed bug:

  loadBalanceHosts=false does not take effect.

Version 8.2.1.300

- Fixed bug:

  The NVARCHAR array type is incompatible.

- Fixed vulnerability:

  CVE-2022-41946

Version 8.2.1.1

The defaultQueryMetaData parameter is added to specify whether to query SQL metadata by default. The default value is **false**.

JDBC supports the raw type, which requires the querying of metadata. If you want to use JDBC to perform operations on the raw type, set **defaultQueryMetaData** to true.

If this parameter is enabled, **prepareStatement** is incompatible with the syntax **create table as**. You can replace it by **Statement**.

Version 8.2.1

- Fixed bug:
  a. An error is reported when **reWriteBatchedInserts** is used to insert data in batches.
  b. "Invalid input syntax for type oid: 03032VLM" is reported when data is imported from Spark to GaussDB(DWS).

Version 8.2.0

- New feature: Compatibility with the Oracle Raw data type. The usage is as follows:
  – Insertion or Modification
    ```
    byte[] bytes = oracleResultSet.getBytes(2)
    prepareStatement.setBytes(bytes)
    // Or
    prepareStatement.setObject(bytes)
    ```
  – Query
    ```
    resultSet.getBytes()
    resultSet.getObject()
    ```
- Fixed bug:

  The field length obtained by the getColumnDisplaySize() method is incorrect.
- Fixed vulnerabilities:

  CVE-2022-26520

  CVE-2022-31197

Version 8.1.3.100

- New features

  The nvarchar2 object can be obtained through resultSet.getNString.
- Fixed vulnerabilities:

  The dependency package fastjson is upgraded to 1.2.83.

Version 8.1.3

Upgrade to the open source version 42.2.23.

- New features
  – The nvarchar2 type is supported.
  – The nvarchar2 object can be obtained through resultSet.getObject.
- Fixed vulnerabilities

  CVE-2022-21724

  NOTE

  For JDBC 8.1.3 and later versions, JDK 1.8 is required.

Version 8.1.1.300

- New features
  – The nvarchar2 type is supported.

  - The nvarchar2 object can be obtained through resultSet.getObject.
- Fixed vulnerabilities

Version 8.1.1.100

- New features

  By default, the driver reports the OS user. To disable this function, you can set **connectionExtraInfo=false**.

  ```
  jdbc:postgresql://host:port/database?connectionExtraInfo=false
  ```

- Fixed vulnerabilities

  Jackson was upgraded.

4.5.1.3 Downloading the JDBC or ODBC Driver

The JDBC or ODBC driver is used to connect to data warehouse clusters. You can download the JDBC or ODBC driver provided by GaussDB(DWS) from the management console or use the open-source JDBC or ODBC driver.

Open-Source JDBC or ODBC Driver

GaussDB(DWS) also supports open-source JDBC and ODBC drivers: PostgreSQL JDBC 9.3-1103 or later; PostgreSQL ODBC 09.01.0200 or later

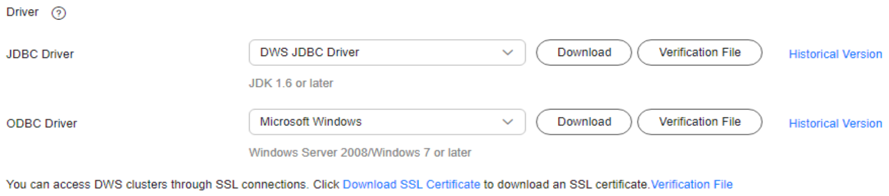
Downloading the JDBC or ODBC Driver

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Client Connections**.

Step 3 In the **Driver** area, choose a driver that you want to download.

Figure 4-37 Downloading the driver



You can access DWS clusters through SSL connections. Click Download SSL Certificate to download an SSL certificate.Verification File

- **JDBC Driver**

  Method 1:

  Select **DWS JDBC Driver** and click **Download** to download the JDBC driver matching the current cluster version. The driver package name is **dws_8.1.x_jdbc_driver.zip**. After the package is decompressed, there will be two JAR packages **gsjdbc4.jar** and **gsjdbc200.jar**.

  - **gsjdbc4.jar**: The **gsjdbc4.jar** driver package is compatible with PostgreSQL. Its class names and class structures are the same as those of the PostgreSQL driver. Applications that run in PostgreSQL can be directly migrated to the current system.

- **gsjdbc200.jar**: If a JVM process needs to access PostgreSQL and GaussDB(DWS) at the same time, this driver package must be used. In this package, the main class name is **com.huawei.gauss200.jdbc.Driver** (that is, **org.postgresql** is replaced with **com.huawei.gauss200.jdbc**). The URL prefix of the database connection is **jdbc:gaussdb**. Other parameters are the same as those of **gsjdbc4.jar**.

If clusters of different versions are available, you will download the JDBC driver matching the earliest cluster version after clicking **Download**. If there is no cluster, you will download the JDBC driver of the earliest version after clicking **Download**. GaussDB(DWS) clusters are compatible with earlier versions of JDBC drivers.

Click **Historical Version** to download the corresponding JDBC driver version. You are advised to download the JDBC driver based on the cluster version.

The JDBC driver can be used on all platforms and depends on JDK 1.6 or later.

If you have clusters of different versions, the system displays a dialog box, prompting you to select the cluster version and download the driver corresponding to the cluster version. In the cluster list on the **Clusters** > **Dedicated Clusters** page, click the name of the specified cluster to go to the **Cluster Information** page and view the cluster version.

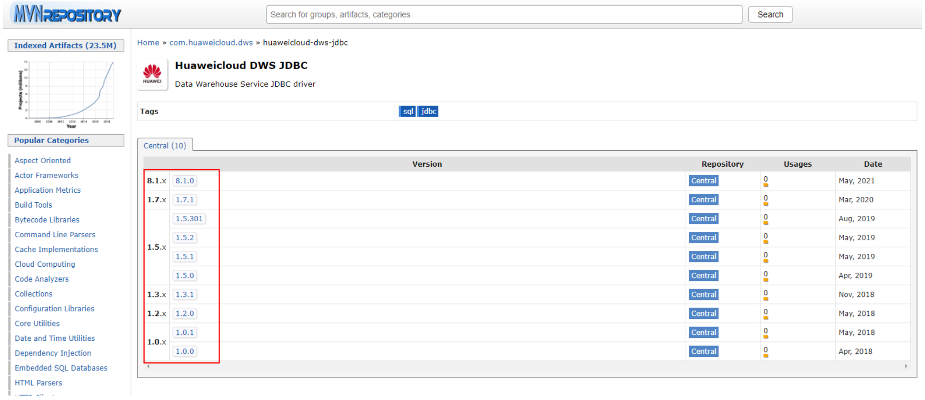
Table 4-15 JDBC driver download address

Driver	Download Link	Download Verification File
DWS JDBC Driver	**dws_9.1.x_jdbc_driver.zip**	**dws_9.1.x_jdbc_driver.zip.sha256**
	dws_8.3.x_jdbc_driver.zip	**dws_8.3.x_jdbc_driver.zip.sha256**
	dws_8.2.x_jdbc_driver.zip	**dws_8.2.x_jdbc_driver.zip.sha256**
	dws_8.1.x_jdbc_driver.zip	**dws_8.1.x_jdbc_driver.zip.sha256**
DWS ARM JDBC Driver	**dws_euler_kunpeng_jdbc.zip**	**dws_euler_kunpeng_jdbc.zip.sha256**

Method 2:

Download the SDK software package by configuring the Maven repository. Click **Add Maven Dependency**. The following page is displayed.

Figure 4-38 Maven page



Home » com.huaweicloud.dws » huaweicloud-dws-jdbc

Huaweicloud DWS JDBC

Data Warehouse Service JDBC driver

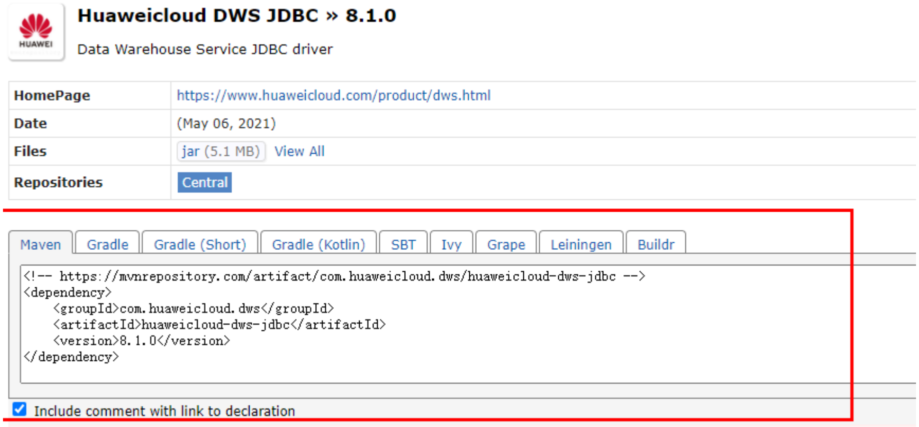
Tags: sql jdbc

Central (10)

	Version	Repository	Usages	Date
8.1.x	8.1.0	Central	0	May, 2021
1.7.x	1.7.1	Central	0	Mar, 2020
1.5.x	1.5.301	Central	0	Aug, 2019
	1.5.2	Central	0	May, 2019
	1.5.1	Central	0	May, 2019
	1.5.0	Central	0	Apr, 2019
1.3.x	1.3.1	Central	0	Nov, 2018
1.2.x	1.2.0	Central	0	May, 2018
1.0.x	1.0.1	Central	0	May, 2018
	1.0.0	Central	0	Apr, 2018

In the list shown in **Figure 4-38**, the first column indicates the cluster version, and the second column indicates the version number of the GaussDB(DWS) JDBC driver package. Select the driver package based on the cluster version and go to the following page:

Figure 4-39 Maven dependency



Huaweicloud DWS JDBC » 8.1.0

Data Warehouse Service JDBC driver

HomePage	https://www.huaweicloud.com/product/dws.html
Date	(May 06, 2021)
Files	jar (5.1 MB) View All
Repositories	Central

```
<!-- https://mvnrepository.com/artifact/com.huaweicloud.dws/huaweicloud-dws-jdbc -->
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.0</version>
</dependency>
```

☑ Include comment with link to declaration

Copy the Maven repository information and add it to the **pom.xml** file. For example, add the following code configuration to the **pom.xml** file:

- gsjdbc4.jar

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.0</version>
</dependency>
```

- gsjdbc200.jar

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.1.1-200</version>
</dependency>
```

Method 3:

Configure dependencies by configuring **gradle**.

Obtain the Maven repository information by referring to method 2 and add that information in the **gradle** configuration file.

  - gsjdbc4.jar

    ```
    implementation("com.huaweicloud.dws:huaweicloud-dws-jdbc:8.1.3")
    ```

  - gsjdbc200.jar

    ```
    implementation("com.huaweicloud.dws:huaweicloud-dws-jdbc:8.1.3-200")
    ```

- **ODBC Driver**

  Select a corresponding version and click **Download** to download the ODBC driver matching the current cluster version. If clusters of different versions are available, you will download the ODBC driver matching the earliest cluster version after clicking **Download**. If there is no cluster, you will download the ODBC driver of the earliest version after clicking **Download**. GaussDB(DWS) clusters are compatible with earlier versions of ODBC drivers.

  Click **Historical Version** to download the corresponding ODBC driver version. You are advised to download the ODBC driver based on the cluster version.

  NOTE

  The ODBC driver is incompatible with Windows Server 2016.

Table 4-16 ODBC driver download address

Applicable OS	Download Link	Download Verification File
Windows	dws_8.2.x_odbc_driver_for_windows.zip	dws_8.2.x_odbc_driver_for_windows.zip.sha256
	dws_8.1.x_odbc_driver_for_windows.zip	dws_8.1.x_odbc_driver_for_windows.zip.sha256
EulerOS Arm	dws_8.2.x_odbc_driver_for_arm_euler.zip	dws_8.2.x_odbc_driver_for_arm_euler.zip.sha256
	dws_8.1.x_odbc_driver_for_arm_euler.zip	dws_8.1.x_odbc_driver_for_arm_euler.zip.sha256
Red Hat Arm	dws_8.2.x_odbc_driver_for_arm_redhat.zip	dws_8.2.x_odbc_driver_for_arm_redhat.zip.sha256
	dws_8.1.x_odbc_driver_for_arm_redhat.zip	dws_8.1.x_odbc_driver_for_arm_redhat.zip.sha256
Redhat x86_64	dws_8.2.x_odbc_driver_for_x86_redhat.zip	dws_8.2.x_odbc_driver_for_x86_redhat.zip.sha256
	dws_8.1.x_odbc_driver_for_x86_redhat.zip	dws_8.1.x_odbc_driver_for_x86_redhat.zip.sha256
SUSE x86_64	dws_8.2.x_odbc_driver_for_x86_suse.zip	dws_8.1.x_odbc_driver_for_x86_suse.zip.sha256
	dws_8.1.x_odbc_driver_for_x86_suse.zip	dws_8.1.x_odbc_driver_for_x86_suse.zip.sha256

----End

4.5.1.4 Using JDBC to Connect to a Cluster

In GaussDB(DWS), you can use a JDBC driver to connect to a database on Linux or Windows. The driver can connect to the database through an ECS on the Huawei Cloud platform or over the Internet.

When using the JDBC driver to connect to the data warehouse cluster, determine whether to enable SSL authentication. SSL authentication is used to encrypt communication data between the client and the server. It safeguards sensitive data transmitted over the Internet. You can download a self-signed certificate file on the GaussDB(DWS) management console. To make the certificate take effect, you must configure the client program using the OpenSSL tool and the Java keytool.

NOTE

The SSL mode delivers higher security than the common mode. You are advised to enable SSL connection when using JDBC to connect to a GaussDB(DWS) cluster.

For details about how to use the JDBC API, see the official documentation.

Prerequisites

- You have installed JDK 1.6 or later and configured environment variables.
- You have downloaded the JDBC driver. For details, see **Downloading the JDBC or ODBC Driver**.

  GaussDB(DWS) also supports open-source JDBC driver: PostgreSQL JDBC 9.3-1103 or later.
- You have downloaded the SSL certificate file. For details, see **Downloading an SSL Certificate**.

Using a JDBC Driver to Connect to a Database

The procedure for connecting to the database using a JDBC driver in a Linux environment is similar to that in a Windows environment. The following describes the connection procedure in a Windows environment.

Step 1 Determine whether you want to use the SSL mode to connect to the GaussDB(DWS) cluster.

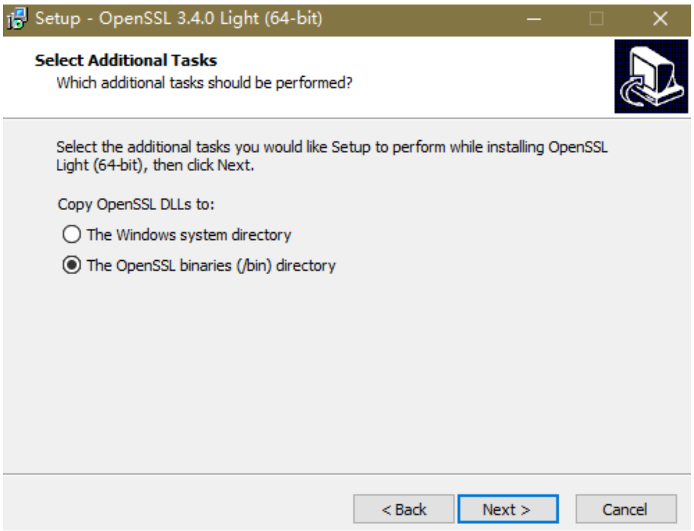
- If yes, enable SSL connection by referring to **Configuring SSL Connection**. SSL connection is enabled by default. Then go to **Step 2**.
- If no, disable SSL connection by referring to **Configuring SSL Connection** and go to **Step 4**.

Step 2 (Optional) On Linux, use WinSCP to upload the downloaded SSL certificate file to the Linux environment.

Step 3 Configure the certificate to enable SSL connection.

1. Download the OpenSSL tool for Windows at **https://slproweb.com/products/Win32OpenSSL.html**. The latest stable version is 3.4. All earlier versions (including 1.1.1, 1.1.0, 1.0.2, 1.0.0, and 0.9.8) are not supported and should not be used. Download **Win64 OpenSSL v3.4.0 Light**.
2. Double-click the installation package **Win64OpenSSL_Light-3.4.0.exe** and install it to the default path on drive C. Copy the DLLs to the OpenSSL

directory, as shown in the following figure. Retain the default settings in the remaining steps until the installation is complete.



Select Additional Tasks

Which additional tasks should be performed?

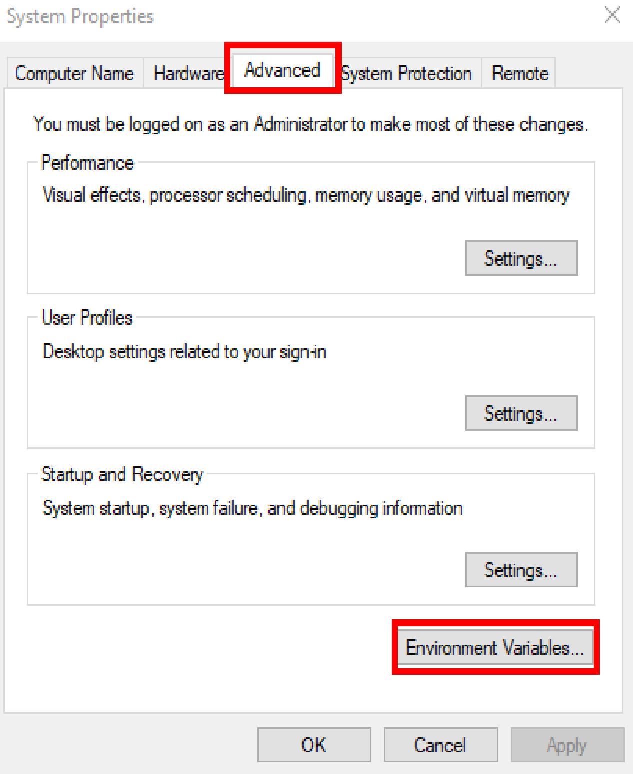
Select the additional tasks you would like Setup to perform while installing OpenSSL Light (64-bit), then click Next.

Copy OpenSSL DLLs to:

○ The Windows system directory

● The OpenSSL binaries (/bin) directory

3. Install an environment variable. Click **Start** in the lower left corner of the local PC, right-click **This PC**, choose **More** > **Properties** > **View advanced system settings**. Switch to the **Advanced** tab and click **Environment Variables**.



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

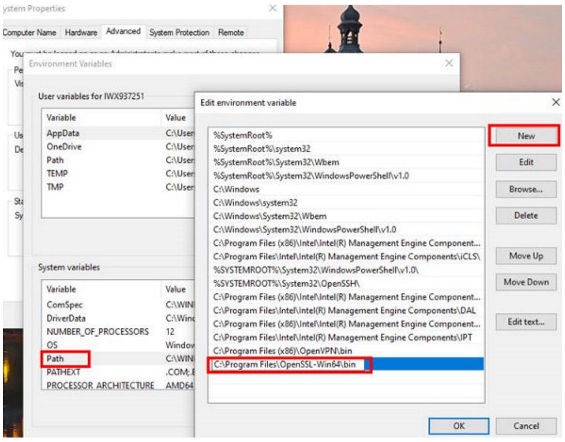
User Profiles

Desktop settings related to your sign-in

Startup and Recovery

System startup, system failure, and debugging information

4. In the **System variables** area, double-click **Path** and click **New** in the window displayed. Add the OpenSSL **bin** path to the last line, for example, **C:\Program Files\OpenSSL-Win64\bin**, and click **OK**. Click **OK** again and the variable is configured successfully.



User variables for IWX937251

Variable	Value
AppData	C:\User
OneDrive	C:\User
Path	C:\User
TEMP	C:\User
TMP	C:\User

System variables

Variable	Value
ComSpec	C:\WIN
DriverData	C:\Winc
NUMBER_OF_PROCESSORS	12
OS	Window
Path	C:\WIN
PATHEXT	.COM;.E
PROCESSOR_ARCHITECTURE	AMD64

```
%SystemRoot%
%SystemRoot%\system32
%SystemRoot%\System32\Wbem
%SystemRoot%\System32\WindowsPowerShell\v1.0
C:\Windows
C:\Windows\system32
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0
C:\Program Files (x86)\Intel\Intel(R) Management Engine Component...
C:\Program Files\Intel\Intel(R) Management Engine Components\iCLS\
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files (x86)\Intel\Intel(R) Management Engine Component...
C:\Program Files\Intel\Intel(R) Management Engine Components\DAL
C:\Program Files (x86)\Intel\Intel(R) Management Engine Component...
C:\Program Files\Intel\Intel(R) Management Engine Components\IPT
C:\Program Files (x86)\OpenVPN\bin
C:\Program Files\OpenSSL-Win64\bin
```

5. Decompress the package to obtain the certificate file. Decompression path **C:\** is used as an example.

   You are advised to store the certificate file in a path of the English version and can specify the actual path when configuring the certificate. If the path is incorrect, a message stating that the file does not exist will be prompted.

6. Open **Command Prompt** and switch to the **C:\dws_ssl_cert\sslcert** path. Run the following commands to import the root license to the truststore:

   ```
   openssl x509 -in cacert.pem -out cacert.crt.der -outform der
   keytool -keystore mytruststore -alias cacert -import -file cacert.crt.der
   ```

   – *cacert.pem* indicates the root certificate obtained after decompression.

   – *cacert.crt.der* indicates the generated intermediate file. You can store the file to another path and change the file name to your desired one.

   – *mytruststore* indicates the generated truststore name and *cacert* indicates the alias name. Both parameters can be modified.

   Enter the truststore password as prompted and answer **y**.

7. Convert the format of the client private key.

   ```
   openssl pkcs12 -export -out client.pkcs12 -in client.crt -inkey client.key
   ```

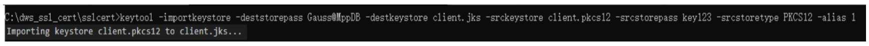
   Enter the client private key password **Gauss@MppDB**. Then enter and confirm the self-defined private key password.

8. Import the private key to the keystore.

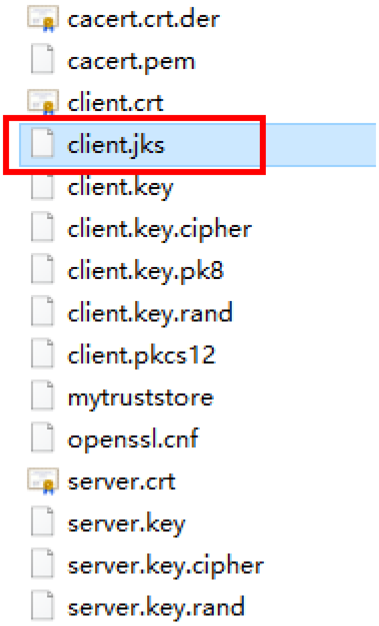
   ```
   keytool -importkeystore -deststorepass Gauss@MppDB -destkeystore client.jks -srckeystore
   client.pkcs12 -srcstorepass Password -srcstoretype PKCS12 -alias 1
   ```

NOTE

- In the preceding command, *Password* is an example. Replace it with the actual password.
- If information similar to the following is displayed and no error is reported, the import is successful. The target key file **client.jks** will be generated in **C:\dws_ssl_cert\sslcert**.



```
C:\dws_ssl_cert\sslcert>keytool -importkeystore -deststorepass Gauss@MppDB -destkeystore client.jks -srckeystore client.pkcs12 -srcstorepass key123 -srcstoretype PKCS12 -alias 1
Importing keystore client.pkcs12 to client.jks...
```



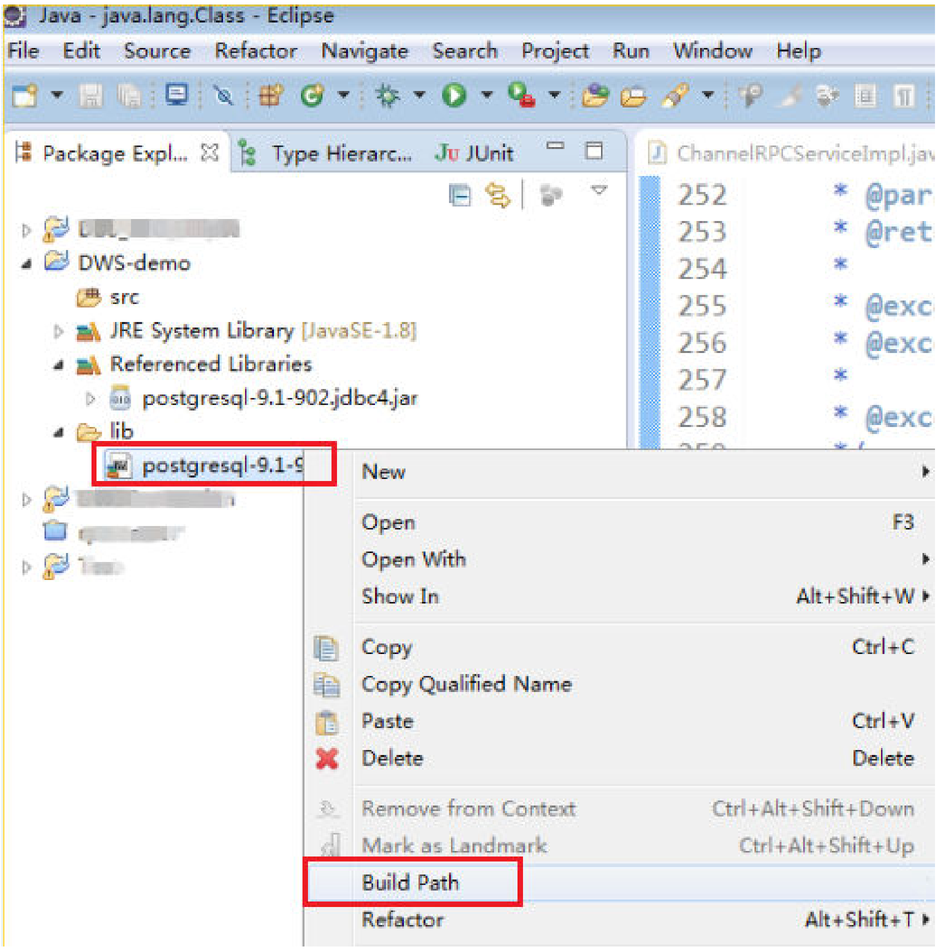
cacert.crt.der
cacert.pem
client.crt
client.jks
client.key
client.key.cipher
client.key.pk8
client.key.rand
client.pkcs12
mytruststore
openssl.cnf
server.crt
server.key
server.key.cipher
server.key.rand

Step 4 Download the driver package **dws_8.1.x_jdbc_driver.zip** and decompress it. There will be two JDBC drive JAR packages, **gsjdbc4.jar** and **gsjdbc200.jar**. Use either of them as required.

Step 5 Add the JAR file to the application project so that applications can reference the JAR file.

Take the Eclipse project as an example. Store the JAR file to the project directory, for example, the **lib** directory in the project directory. In the Eclipse project, right-click the JAR file in the **lib** directory and choose **Build Path** to reference the JAR file.

Figure 4-40 Referencing a JAR file


Java - java.lang.Class - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Expl...
Type Hierarc...
JUnit
ChannelRPCServiceImpl.jav
DWS-demo
src
JRE System Library [JavaSE-1.8]
Referenced Libraries
postgresql-9.1-902.jdbc4.jar
lib
postgresql-9.1-9
New
Open F3
Open With
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Mark as Landmark Ctrl+Alt+Shift+Up
Build Path
Refactor Alt+Shift+T

Alternatively, you can use another method. In the Maven project, you can directly add the GaussDB(DWS) JDBC driver as a dependency item to the POM file. The following shows an example:

- gsjdbc4.jar

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.0</version>
</dependency>
```

- gsjdbc200.jar

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.1.1-200</version>
</dependency>
```

NOTE

For details about the image repository address configured in **setting.xml**, see **https://mvnrepository.com/**.

Step 6 Load the driver.

The following methods are available:

- Using a code: **Class.forName("org.postgresql.Driver");**

- Using a parameter during the JVM startup: **java -Djdbc.drivers=org.postgresql.Driver jdbctest**

  NOTE

  The JDBC driver package downloaded on GaussDB(DWS) contains both **gsjdbc4.jar** and **gsjdbc200.jar**.

  - **gsjdbc4.jar**: The **gsjdbc4.jar** driver package is compatible with PostgreSQL. Its class names and class structures are the same as those of the PostgreSQL driver. Applications that run in PostgreSQL can be directly migrated to the current system.
  - **gsjdbc200.jar**: If a JVM process needs to access PostgreSQL and GaussDB(DWS) at the same time, this driver package must be used. In this package, the main class name is **com.huawei.gauss200.jdbc.Driver** (replace **org.postgresql** with **com.huawei.gauss200.jdbc**). The URL prefix of the database connection is **jdbc:gaussdb**. Other parameters are the same as those of **gsjdbc4.jar**.
  - The GaussDB(DWS) driver package downloaded from the Maven repository is the same as the **gsjdbc4** driver package.

Step 7 Call the **DriverManager.getConnection()** method of JDBC to connect to GaussDB(DWS) databases.

The JDBC API does not provide the connection retry capability. You need to implement the retry processing in the service code.

DriverManager.getConnection() methods:

- DriverManager.getConnection(String url);
- DriverManager.getConnection(String url, Properties info);
- DriverManager.getConnection(String url, String user, String password);

Table 4-17 Database connection parameters

Parameter	Description
url	Specifies the database connection descriptor, which can be viewed on the management console. For details, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**. The URL format is as follows: ● jdbc:postgresql:database ● jdbc:postgresql://host/database ● jdbc:postgresql://host:port/database ● jdbc:postgresql://host:port[,host:port][...]/database **NOTE** ● If **gsjdbc200.jar** is used, change **jdbc:postgresql** to **jdbc:gaussdb**. – **database** indicates the name of the database to be connected. – **host** indicates the name or IP address of the database server. If an ELB is bound to the cluster, set **host** to the IP address of the ELB. – **port** indicates the port number of the database server. By default, the database running on port 8000 of the local host is connected. – Multiple IP addresses and ports can be configured. JDBC balances load by random access and failover, and will automatically ignore unreachable IP addresses. Separate multiple pairs of IP addresses and ports by commas (,). Example: **jdbc:postgresql://10.10.0.13:8000,10.10.0.14:8000/database** ● If JDBC is used to connect to a cluster, only JDBC connection parameters can be configured in a cluster address. Variables cannot be added.

Parameter	Description
info	Specifies database connection properties. Common properties include the following: ● **user**: a string type. It indicates the database user who creates the connection task. ● **password**: a string type. It indicates the password of the database user. ● **ssl**: a boolean type. It indicates whether to use the SSL connection. ● **loggerLevel**: string type. It indicates the volume of log data sent to the LogStream or LogWriter specified in the DriverManager. Currently, **OFF**, **DEBUG**, and **TRACE** are supported. **DEBUG** indicates that only logs of **DEBUG** or a higher level are printed, generating little log information. **TRACE** indicates that logs of the **DEBUG** and **TRACE** levels are displayed, generating detailed log information. The default value is **OFF**, indicating that no logs will be displayed. ● **prepareThreshold**: integer type. It indicates the number of **PreparedStatement** executions required before requests are converted to prepared statements in servers. The default value is **5**. ● **batchMode**: boolean type. It indicates whether to connect the database in batch mode. ● **fetchsize**: integer type. It indicates the default fetch size for statements in the created connection. ● **ApplicationName**: string type. It indicates an application name. The default value is **PostgreSQL JDBC Driver**. ● **allowReadOnly**: boolean type. It indicates whether to enable the read-only mode for connection. The default value is **false**. If the value is not changed to **true**, the execution of **connection.setReadOnly** does not take effect. ● **blobMode**: string type. It is used to set the **setBinaryStream** method to assign values to different data types. The value **on** indicates that values are assigned to the BLOB data type and **off** indicates that values are assigned to the BYTEA data type. The default value is **on**. ● **currentSchema**: string type. It specifies the schema used for connecting to the database. ● **defaultQueryMetaData**: Boolean. It specifies whether to query SQL metadata by default. The default value is **false**. After this function is enabled, raw data operations are supported. However, it is incompatible with the **create table as** and **select into** operations in **PrepareStatement**. ● **connectionExtraInfo**: boolean type. This parameter indicates whether the JDBC driver reports the driver deployment path and process owner to the database.

Parameter	Description
	NOTE The value can be **true** or **false**. The default value is **true**. If **connectionExtraInfo** is set to **true**, the JDBC driver reports the driver deployment path and process owner to the database and displays the information in the **connection_info** parameter. In this case, you can query the information from **PG_STAT_ACTIVITY** or **PGXC_STAT_ACTIVITY**. ● **TCP_KEEPIDLE=30**: The detection starts after the connection is idle for 30s. This parameter is valid only when **tcpKeepAlive** is set to **true**. ● **TCP_KEEPCOUNT=9**: A total of nine detections are performed. This parameter is valid only when **tcpKeepAlive** is set to **true**. ● **TCP_KEEPINTERVAL=30**: The detection interval is 30s. This parameter is valid only when **tcpKeepAlive** is set to **true**. ● **cnListRefreshSwitch (string)**: determines if JDBC automatically detects the CN list. Set to **on** for automatic detection, and **off** to disable it. The default value is **off**. ● **cnListRefreshDelay (integer)**: specifies the start time for scanning the CN liveness list. The default value is **1800000** (in milliseconds). This parameter is valid only when **cnListRefreshSwitch** is set to **on**. ● **cnListRefreshPeriod (integer)**: specifies the interval for scanning the CN list. The default value is **1800000** (in milliseconds). This parameter is valid only when **cnListRefreshSwitch** is set to **on**. ● **autoReconnect (boolean)**: enables or disables automatic reconnection for database connections. The value **true** enables it. The default value is **false**. ● **reConnectCount (integer)**: specifies the number of automatic reconnection attempts. The default value is **10**. This parameter is valid only when **autoReconnect** is set to **true**. If reconnections exceed this number, the reconnection fails. ● **sslCrl**: a string type that sets the path for the revoked certificate used by JDBC. The default value is **null**.
user	Specifies the database user.
password	Specifies the password of the database user.

The following describes the sample code used to encrypt the connection using the SSL certificate:

```
// The following code obtains the database SSL connection operation and encapsulates the operation as an API.
public static Connection GetConnection(String username, String passwd) {
    // Define the driver class.
    String driver = "org.postgresql.Driver";
        //Set keyStore.
    System.setProperty("javax.net.ssl.trustStore", "mytruststore");
    System.setProperty("javax.net.ssl.keyStore", "client.jks");
    System.setProperty("javax.net.ssl.trustStorePassword", "password");
    System.setProperty("javax.net.ssl.keyStorePassword", "password");

    Properties props = new Properties();
    props.setProperty("user", username);
    props.setProperty("password", passwd);
    props.setProperty("ssl", "true");

    String url = "jdbc:postgresql://" + "10.10.0.13" + ':' + "8000" + '/' + "gaussdb";
    Connection conn = null;

    try {
        // Load the driver.
        Class.forName(driver);
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
    try {
        // Create a connection.
        conn = DriverManager.getConnection(url, props);
        System.out.println("Connection succeed!");
    } catch (SQLException throwables) {
        throwables.printStackTrace();
        return null;
    }
    return conn;
}
```

Step 8 Run SQL statements.

1. Run the following command to create a statement object:
   ```
   Statement stmt = con.createStatement();
   ```
2. Run the following command to execute the statement object:
   ```
   int rc = stmt.executeUpdate("CREATE TABLE tab1(id INTEGER, name VARCHAR(32));");
   ```
3. Run the following command to release the statement object:
   ```
   stmt.close();
   ```

Step 9 Call **close()** to close the connection.

----End

Sample Code

This code sample illustrates how to develop applications based on the JDBC API provided by GaussDB(DWS).

NOTE

Before completing the following example, you need to create a stored procedure. For details, see **Tutorial: Development Using JDBC or ODBC**.

```
create or replace procedure testproc
(
    psv_in1 in integer,
    psv_in2 in integer,
    psv_inout in out integer
)
as
begin
    psv_inout := psv_in1 + psv_in2 + psv_inout;
end;
/
```

```
//DBtest.java
//gsjdbc4.jar is used as an example. If gsjdbc200.jar is used, replace the driver class name org.postgresql
with com.huawei.gauss200.jdbc and replace the URL prefix jdbc:postgresql with jdbc:gaussdb.
//Demonstrate the main steps for JDBC development, including creating databases, creating tables, and
```

```
inserting data.

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.CallableStatement;
import java.sql.Types;

public class DBTest {
//Create a database connection. Replace the following IP address and database with the actual database
connection address and database name.
  public static Connection GetConnection(String username, String passwd) {
    String driver = "org.postgresql.Driver";
    String sourceURL = "jdbc:postgresql://10.10.0.13:8000/database";
    Connection conn = null;
    try {
      // Load the database driver.
      Class.forName(driver).newInstance();
    } catch (Exception e) {
      e.printStackTrace();
      return null;
    }

    try {
      //Create a database connection.
      conn = DriverManager.getConnection(sourceURL, username, passwd);
      System.out.println("Connection succeed!");
    } catch (Exception e) {
      e.printStackTrace();
      return null;
    }

    return conn;
  };

  //Run the common SQL statements to create table customer_t1.
  public static void CreateTable(Connection conn) {
    Statement stmt = null;
    try {
      stmt = conn.createStatement();

      //Run the common SQL statements.
      int rc = stmt
          .executeUpdate("CREATE TABLE customer_t1(c_customer_sk INTEGER, c_customer_name
VARCHAR(32));");

      stmt.close();
    } catch (SQLException e) {
      if (stmt != null) {
        try {
          stmt.close();
        } catch (SQLException e1) {
          e1.printStackTrace();
        }
      }
      e.printStackTrace();
    }
  }

  //Run the prepared statements and insert data in batches.
  public static void BatchInsertData(Connection conn) {
    PreparedStatement pst = null;

    try {
      //Generate the prepared statements.
      pst = conn.prepareStatement("INSERT INTO customer_t1 VALUES (?,?)");
      for (int i = 0; i < 3; i++) {
        //Add parameters.
        pst.setInt(1, i);
        pst.setString(2, "data " + i);
        pst.addBatch();
      }
      //Execute batch processing.
      pst.executeBatch();
      pst.close();
    } catch (SQLException e) {
      if (pst != null) {
        try {
          pst.close();
        } catch (SQLException e1) {
        e1.printStackTrace();
        }
      }
      e.printStackTrace();
    }
  }

  //Run the precompiled statement to update the data.
  public static void ExecPreparedSQL(Connection conn) {
    PreparedStatement pstmt = null;
    try {
      pstmt = conn
          .prepareStatement("UPDATE customer_t1 SET c_customer_name = ? WHERE c_customer_sk = 1");
      pstmt.setString(1, "new Data");
      int rowcount = pstmt.executeUpdate();
      pstmt.close();
    } catch (SQLException e) {
      if (pstmt != null) {
        try {
          pstmt.close();
        } catch (SQLException e1) {
          e1.printStackTrace();
        }
      }
      e.printStackTrace();
    }
  }


//Execute the storage procedure.
  public static void ExecCallableSQL(Connection conn) {
    CallableStatement cstmt = null;
    try {

      cstmt=conn.prepareCall("{? = CALL TESTPROC(?,?,?)}");
      cstmt.setInt(2, 50);
      cstmt.setInt(1, 20);
      cstmt.setInt(3, 90);
      cstmt.registerOutParameter(4, Types.INTEGER);  //Register a parameter of the out type. Its value is an
integer.
      cstmt.execute();
      int out = cstmt.getInt(4);  //Obtain the out parameter.
      System.out.println("The CallableStatment TESTPROC returns:"+out);
      cstmt.close();
    } catch (SQLException e) {
      if (cstmt != null) {
        try {
          cstmt.close();
        } catch (SQLException e1) {
          e1.printStackTrace();
        }
      }
      e.printStackTrace();
    }
  }
```

```
 /**
  * Main program, which gradually invokes each static method.
  * @param args
 */
 public static void main(String[] args) {
   //Create a database connection. Replace User and Password with the actual database user name and
password.
   Connection conn = GetConnection("User", "Password");

   //Create a table.
   CreateTable(conn);

   //Insert data in batches.
   BatchInsertData(conn);

   //Run the precompiled statement to update the data.
   ExecPreparedSQL(conn);

   //Execute the storage procedure.
   ExecCallableSQL(conn);

   //Close the database connection.
   try {
     conn.close();
   } catch (SQLException e) {
     e.printStackTrace();
   }

 }

}
```

4.5.1.5 Configuring JDBC to Connect to a Cluster (Load Balancing Mode)

Context

If you use JDBC to connect to only one CN in the cluster, this CN may be overloaded and other CN resources wasted. It also incurs single-node failure risks.

To avoid these problems, you can use JDBC to connect to multiple CNs. The following three methods are available:

- Connection using ELB: An ELB distributes access traffic to multiple ECSs for traffic control based on forwarding policies. It improves the fault tolerance capability of application programs.
- To connect to a cluster using JDBC load balancing, include at least one internal IP address of the CN in the URL. The system will scan all CN IP addresses automatically. JDBC load balancing functions like ELB, randomly connecting to a CN.
- Connection in multi-host mode: Use JDBC to configure multiple nodes, which is similar to ELB.

Method 1: Using ELB to Connect to a Cluster

Step 1 Obtain the Elastic Load Balance address. On the console, go to the details page of a cluster and obtain the ELB IP address. For details, see **Associating and Disassociating ELB**.

Step 2 Configure the driver. For details, see **Downloading the JDBC or ODBC Driver**.

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.1.1</version>
</dependency>
```

Step 3 Obtain the database connection.

```
private static final String USER_NAME = "dbadmin";
private static final String PASSWORD = "password";
// jdbc:postgresql://ELB_IP:PORT/dbName"
private static final String URL = "jdbc:postgresql://100.95.153.169:8000/gaussdb";
private static Properties properties = new Properties();
static {
    properties.setProperty("user", USER_NAME);
    properties.setProperty("password", PASSWORD);
}
/**
 * Obtain the database connection.
 */
public static Connection getConnection() {
    Connection connection = null;
    try {
        connection = DriverManager.getConnection(URL, properties);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return connection;
}
```

----End

Method 2: Using JDBC Load Balancing to Connect to a Cluster (Recommended)

Step 1 Obtain the private IP address. Open the specified cluster topology page on the console and obtain the internal IP address of the CN. For details, see **Viewing the GaussDB(DWS) Cluster Topology**.

Step 2 Configure the driver. For details, see **Downloading the JDBC or ODBC Driver**.

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.3.1.200</version>
</dependency>
```

Step 3 Obtain the database connection. For how to set URL parameters, see **Using JDBC to Connect to a Cluster**.

```
private static final String USER_NAME = "dbadmin";
private static final String PASSWORD = "password";
// jdbc:postgresql://host1:port1,host2:port2/dbName"
private static final String URL = "jdbc:postgresql://
100.95.146.194:8000,100.95.148.220:8000,100.93.0.221:8000/gaussdb?
loadBalanceHosts=true&cnListRefreshSwitch=on&cnListRefreshDelay=100000&cnListRefreshPeriod=5000
;
private static Properties properties = new Properties();
static {
    properties.setProperty("user", USER_NAME);
    properties.setProperty("password", PASSWORD);
}
/**
 * Obtain the database connection.
 */
public static Connection getConnection() {
    Connection connection = null;
    try {
        connection = DriverManager.getConnection(URL, properties);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return connection;
}
```

----End

Method 3: Connecting to the Cluster in Multi-host Mode

Step 1 Obtain the EIP. Go to the details page of a cluster on the console and obtain the EIP.

Step 2 Configure the driver. For details, see **Downloading the JDBC or ODBC Driver**.

```
<dependency>
    <groupId>com.huaweicloud.dws</groupId>
    <artifactId>huaweicloud-dws-jdbc</artifactId>
    <version>8.1.1.1</version>
</dependency>
```

Step 3 Obtain the database connection.

```
private static final String USER_NAME = "dbadmin";
private static final String PASSWORD = "password";
// jdbc:postgresql://host1:port1,host2:port2/dbName"
private static final String URL = "jdbc:postgresql://
100.95.146.194:8000,100.95.148.220:8000,100.93.0.221:8000/gaussdb?loadBalanceHosts=true";
private static Properties properties = new Properties();
static {
    properties.setProperty("user", USER_NAME);
    properties.setProperty("password", PASSWORD);
}
/**
 * Obtain the database connection.
 */
public static Connection getConnection() {
    Connection connection = null;
    try {
        connection = DriverManager.getConnection(URL, properties);
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return connection;
}
```

----End

4.5.1.6 Configuring JDBC to Connect to a Cluster (IAM Authentication Mode)

Overview

GaussDB(DWS) allows you to access databases using IAM authentication. When you use the JDBC application program to connect to a cluster, set the IAM username, credential, and other information as you configure the JDBC URL. After doing this, when you try to access a database, the system will automatically generate a temporary credential and a connection will be set up.

NOTE

- Currently, only clusters 1.3.1 and later versions and their corresponding JDBC drivers can access the databases in IAM authentication mode. Download the JDBC driver. For details, see **Downloading the JDBC or ODBC Driver**.

IAM supports two types of user credential: password and Access Key ID/Secret Access Key (AK/SK). JDBC connection requires the latter.

The IAM account you use to access a database must be granted with the **DWS Database Access** permission. Only users with both the **DWS Administrator** and **DWS Database Access** permissions can connect to GaussDB(DWS) databases using the temporary database user credentials generated based on IAM users.

The **DWS Database Access** permission can only be granted to user groups. Ensure that your IAM account is in a user group with this permission.

On IAM, only users in the **admin** group have the permissions to manage users. This requires that your IAM account be in the **admin** user group. Otherwise, contact the IAM account administrator to grant your IAM account this permission.

The process of accessing a database is as follows:

1. **Granting an IAM Account the GaussDB(DWS) Database Access Permission**
2. **Creating an IAM User Credential**
3. **Configuring the JDBC Connection to Connect to a Cluster Using IAM Authentication**

Granting an IAM Account the GaussDB(DWS) Database Access Permission

Step 1 Log in to the Huawei Cloud management console. In the service list, choose **Management & Governance** > **Identity and Access Management** to enter the IAM management console.

Step 2 Modify the user group to which your IAM user belongs. Set a policy for, grant the **DWS Database Access** permission to, and add your IAM user to it.

Only users in the **admin** user group of IAM can perform this step. In IAM, only users in the **admin** user group can manage users, including creating user groups and users and setting user group rights.

For details, see **"Viewing or Modifying User Group Information"** in the ***Identity and Access Management User Guide***.

You can also create an IAM user group, and set a policy for, grant the **DWS Administrator** and **DWS Database Access** permissions to, and add your IAM user to it. For details, see **Creating a User Group and Assigning Permissions** in the *Identity and Access Management User Guide*.

----End

Creating an IAM User Credential

You can log in to the management console to create an AK/SK pair or use an existing one.

Step 1 Log in to the management console.

Step 2 Move the cursor to the username in the upper right corner and choose **My Credentials**.

Step 3 Choose **Access Keys** to view the existing access keys. You can also click **Create Access Key** to create a new one.

The AK/SK pair is so important that you can download the private key file containing the AK/SK information only when you create the pair. On the management console, you can only view the AKs. If you have not downloaded the file, obtain it from your administrator or create an AK/SK pair again.

NOTE

Each user can create a maximum of two AK/SK pairs, which are valid permanently. To ensure account security, change your AK/SK pairs periodically and keep them safe.

----End

Configuring the JDBC Connection to Connect to a Cluster Using IAM Authentication

Configuring JDBC Connection Parameters

Table 4-18 Database connection parameters

Parameter	Description
url	gsjdbc4.jar/gsjdbc200.jar database connection descriptor. The JDBC API does not provide the connection retry capability. You need to implement the retry processing in the service code. The URL example is as follows: jdbc:dws:iam://dws-IAM-demo:ap-southeast-1/gaussdb?**AccessKeyID**=XXXXXXXXXXXXXXXXXXXX&**SecretAccessKey**=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX&**DbUser**=user_test&**AutoCreate**=true **JDBC URL parameters**: ● **jdbc:dws:iam** is a prefix in the URL format. ● **dws-IAM-demo** indicates the name of the cluster containing the database. ● **ap-southeast-1** indicates the region where the cluster resides. JDBC accesses the GaussDB(DWS) cluster in the corresponding region and delivers the IAM certificate to the cluster for IAM user authentication. The GaussDB(DWS) service address has been recorded in the JDBC configuration file. For details about GaussDB(DWS) regions, visit **Regions and Endpoints**. ● **gaussdb** indicates the name of the database to which you want to connect. ● **AccessKeyID** and **SecretAccessKey** are the access key ID and secret access key corresponding to the IAM user specified by **DbUser**. ● Set **DbUser** to the IAM username. Note that the current version does not support hyphens (-) in the IAM username. – If the user specified by **DbUser** exists in the database, the temporary user credential has the same permissions as the existing user. – If the user specified by **DbUser** does not exist in the database and the value of **AutoCreate** is **true**, a new user named by the value of **DbUser** is automatically created. The created user is a common database user by default. ● Parameter **AutoCreate** is optional. The default value is **false**. This parameter indicates whether to automatically create a database user named by the value of **DbUser** in the database. – The value **true** indicates that a user is automatically created. If the user already exists, the user will not be created again. – The value **false** indicates that a user is not created. If the username specified by **DbUser** does not exist in the database, an error is returned. ● **addressType** indicates the type of the address used for the connection. The default value is **auto**. – **auto**: The selection priority is EIP, ELB, and private IP address. If **auto** is selected, the system chooses an IP address for the connection.

Parameter	Description
	– **eip**: The system chooses an EIP for the connection. – **elb**: The system chooses an ELB for the connection. The selection priority is **elb_public** and **elb_private**. – **elb_public**: Use the ELB public IP address for the connection. – **elb_private**: Use ELB private IP address for the connection.
info	Database connection properties. Common properties include the following: ● **ssl**: a boolean type. It indicates whether the SSL connection is used. ● **loglevel**: an integer type. It sets the log amount recorded in DriverManager for LogStream or LogWriter. Currently, **org.postgresql.Driver.DEBUG** and **org.postgresql.Driver.INFO** logs are supported. If the value is **1**, only **org.postgresql.Driver.INFO** (little information) is recorded. If the value is greater than or equal to **2**, **org.postgresql.Driver.DEBUG** and **org.postgresql.Driver.INFO** logs are printed, and detailed log information is generated. Its default value is **0**, which indicates that no logs are printed. ● **charSet**: a string type. It indicates character sets used when data is sent from the database or the database receives data. ● **prepareThreshold**: an integer type. It is used to determine the execution times of PreparedStatement before the information is converted into prepared statements on the server. The default value is **5**.

Example

```
//The following uses gsjdbc4.jar as an example.
// The following code encapsulates the database connection obtaining operations into an API. You can
connect to the database by specifying the region where the cluster is located, cluster name, access key ID,
secret access key, and the corresponding IAM username.
public static Connection GetConnection(String clustername, String regionname, String AK, String SK,
    String username) {
    // Driver class.
    String driver = "org.postgresql.Driver";
    // Database connection descriptor.
    String sourceURL = "jdbc:dws:iam://" + clustername + ":" + regionname + "/postgresgaussdb?" +
"AccessKeyID="
        + AK + "&SecretAccessKey=" + SK + "&DbUser=" + username + "&autoCreate=true";

    Connection conn = null;

    try {
        // Load the driver.
        Class.forName(driver);
    } catch (ClassNotFoundException e) {
        return null;
    }
    try {
        // Create a connection.
        conn = DriverManager.getConnection(sourceURL);
        System.out.println("Connection succeed!");
    } catch (SQLException e) {
        return null;
    }
    return conn;
}
```

4.5.1.7 Third-party Connection Pool of the JDBC Configuration Database

Context

GaussDB(DWS) does not have its own JDBC connection pool, and the inherited PostgreSQL connection pool is offline. Use third-party connection pools like Druid, HikariCP, or DBCP 2.

NOTE

- The connection pool inherited by JDBC from PostgreSQL has been brought offline and is not recommended.
- Determine the version of the JDBC and driver to be downloaded and how to set the connection pool parameters based on the site requirements.

Configuring the DBCP 2 Connection Pool

Step 1 Download the JDBC driver package. For details, see **Downloading the JDBC or ODBC Driver**.

- Download the **commons-dbcp2** driver package from **https://commons.apache.org/dbcp/download_dbcp.cgi**.
- Download the **commons-logging** driver package from **https://commons.apache.org/proper/commons-logging/download_logging.cgi**.
- Download the **commons-pool2** driver package from **https://commons.apache.org/proper/commons-pool/download_pool**.

Step 2 Add the JDBC driver package and the **commons-dbcp2**, **commons-logging**, and **commons-pool2** driver packages to the project and configure parameters related to the database connection pool.

NOTE

- Enabling **removeAbandoned** allows the connection pool to reclaim and reuse a discarded connection. This occurs when the conditions (getNumIdle() < 2) and (getNumActive() > getMaxTotal() - 3) are met.
  - For example, if **maxTotal** is set to **20**, there are 18 active connections and one connection is restricted. In this case, **removeAbandoned** is triggered.
  - An active connection is deleted only when it is not used for a period of time specified by **removeAbandonedTimeout**. The default value is 300 seconds.
  - Traversing a result set does not count as usage. Creating a statement, prepared statement, callable statement, or executing a query resets the **lastUsed** property of its parent connection.
- In high-load systems, setting **maxIdle** to a small value may cause new connections to close immediately. This is because active threads close connections faster than those that open connections. As a result, the number of idle connections is greater than the value of **maxIdle**. In a high-load system, the most appropriate value of **maxIdle** is various, but the default value is a good start point.

Table 4-19 Parameters of the DBCP 2 connection pool

Parameter	Default Value	Description
driverClassName	Enter the value of **org.postgresql.Driver**.	Name of the database driver.
url	-	URL for connecting to the database.
username	-	Username.
password	-	Password.
connectionProperties	-	The connection parameters are sent to the JDBC driver when a new connection is set up. The string must be in the format of **[Parameter name=Parameter value;]**. **NOTE** The username and password attributes need to be specified. Therefore, the two parameters do not need to be included here.
defaultAutoCommit	-	Automatic submission. By default, the connection created through the current connection pool is in the automatic submission state. If this parameter is not set, the **setAutoCommit** method is not invoked.
defaultReadOnly	-	Read-only setting. By default, the connection created through the current connection pool is read-only. If the connection is not set, the **setReadOnly** method is not invoked.
defaultTransactionIsolation	-	Transaction isolation level. The default transaction isolation policy is used for connections created through this pool. The value can be one of the following: ● **NONE** ● **READ_COMMITTED** ● **READ_UNCOMMITTED** ● **REPEATABLE_READ** ● **SERIALIZABLE**
defaultCatalog	-	The default catalog is used for connections created through this pool.

Parameter	Default Value	Description
cacheState	true	Cache status of the connection pool. If this parameter is set to **true**, the current read-only status and auto-commit settings are cached during the first read or write operation after the resource pool connects. This eliminates the need for additional database queries on subsequent **getter** calls. If the underlying connection is accessed directly, changes to the read-only state or auto-commit settings will not update the cache. Set this parameter to **false** to disable caching in such cases.
defaultQueryTimeout	null	Query timeout interval. ● Enter an integer, which is used to specify the query timeout interval when a statement is created. ● If the value is **null**, the default driver settings are used.
enableAutoCommitOnReturn	true	When a connection is returned to the pool, the connection is automatically submitted. Setting it to **true** will return the connection to the pool with **autoCommit** set to **true** by default.
rollbackOnReturn	true	Roll back all operations when the connection is returned to the pool. Setting it to **true** will automatically execute **"rollback()"** when the connection is returned to the pool, provided that auto submission is enabled.
initialSize	0	Number of initial connections. Number of connections created during initialization when the current connection pool is started. The initial version is 1.2.
maxTotal	8	Maximum number of active connections in the pool. A negative value means there is no limit.
maxIdle	8	Maximum number of idle connections in the pool. Excess idle connections are released when returned to the pool. A negative value means there is no limit.

Parameter	Default Value	Description
minIdle	0	Minimum number of idle connections. Minimum number of idle connections to retain in the pool. If the number of idle connections falls below this value, new idle connections are created. A value of **0** means no idle connections are created. **NOTE** The value takes effect only when **timeBetweenEvictionRunsMillis** is set to a positive number.
maxWaitMillis	-	Maximum waiting time for obtaining a connection from the connection pool. ● If this parameter is set to **–1** and no connection is available, the connection pool waits indefinitely until a connection is obtained. ● If the parameter is set to *N*, the connection pool waits for *N* milliseconds. If the waiting time is insufficient, an exception is thrown.
validationQuery	SELECT 1	Query confirmation SQL statement, which validates the connection before it is returned to the caller by the connection pool. ● If specified, the query must be a **SELECT** statement that returns at least one row of data. ● If no value is specified, the connection is verified by invoking the **"isValid()"** method.
validationQueryTimeout	-	Query timeout interval for valid SQL statements, in seconds. If the parameter is set to a positive number, the value is transferred to the **"setQueryTimeOut()"** method of the JDBC driver. The setting takes effect for the SQL statement for confirming the validity of the query.
testOnCreate	false	Whether to verify the validity of a connection immediately after creation. If verification fails, the creation attempt fails.
testOnBorrow	true	Whether to verify the validity of a connection when it is leased from the pool. If verification fails, the connection is released and another is leased.

Parameter	Default Value	Description
testOnReturn	false	Whether to verify the validity of a connection before returning it to the pool.
testWhileIdle	false	Whether to verify the validity of idle connections using an evictor, if available. Invalid connections are released.
timeBetweenEvictionRunsMillis	-1	Hibernate time (in milliseconds) for the idle object eviction thread. A non-positive value disables the thread.
numTestsPerEvictionRun	3	Number of objects checked during the running of each idle object eviction thread.
minEvictableIdleTimeMillis	1000 * 60 * 30	Minimum number of milliseconds in which objects that meet the eviction conditions are idle in the pool. Minimum duration for releasing an idle connection, in milliseconds.
softMinEvictableIdleTimeMillis	-1	Minimum number of milliseconds in which objects that meet the eviction conditions are idle in the pool. Idle connections are released after at least *N* milliseconds, provided that at least the number of connections specified by minIdle is retained in the pool. If **miniEvictableIdleTimeMillis** is set to a positive number, the idle connection evictor checks **miniEvictableIdleTimeMillis** first, and then **softMinEvictableIdleTimeMillis** and the **minIdle** condition.
maxConnLifetimeMillis	-1	Maximum lifetime of a connection (in milliseconds). Connections exceeding this time fail on the next activation, passivation, or verification. A value of **0** or negative means unlimited lifetime.
logExpiredConnections	true	Whether to write logs when an expired connection is closed by the pool. If a connection's lifespan exceeds **maxConnLifetimeMillis**, it will be reclaimed by the connection pool and a log will be generated by default. If this parameter is set to **false**, no log will be written.
connectionInitSqls	-	This parameter executes a set of SQL statements to initialize a physical connection when it is first created. These statements run only once per connection.

Parameter	Default Value	Description
lifo	true	Last in first out. - Last in first out. If this parameter is set to **true**, the connection pool returns the last used connection first (if there are available idle connections in the pool). - If this parameter is set to **false**, the pool operates as a FIFO queue and obtains connections from the idle connection instance pool in the sequence in which they are returned.
poolPreparedStatements	false	This determines whether the preprocessing statement pool in the connection pool will be applied.

maxOpenPreparedStatements	-	Maximum number of statements that can be allocated in the statement pool at the same time. A negative value means no limit.
		This setting also applies to the pre-processed statement pool. When a statement pool is created for each connection, the pre-processed statements generated by the following method are included.
		`public PreparedStatement prepareStatement(String sql)`
		`public PreparedStatement prepareStatement(String sql, int resultSetType, int resultSetConcurrency)`
		NOTE
		Ensure that connections leave resources for other statements by setting **maxOpenPreparedStatements** to a value less than the maximum number of cursors.

accessToUnderlyingConnectionAllowed	false	This controls whether the PoolGuard can access underlying connections.
removeAbandonedOnMaintenance removeAbandonedOnBorrow	false	Whether to delete abandoned connections that have been abandoned for a period longer than the time specified by **removeAbandonedTimout**. If the value is **true**, connections unused for longer than **removeAbandonedTimeout** are considered abandoned and removed. Creating or executing statements resets the **lastUsed** property of the parent connection. Setting this parameter to **true** helps recover connections in applications with few write operations.

Parameter	Default Value	Description
removeAbandonedTimeout	300	Timeout interval for removing a discarded connection, in seconds.
logAbandoned	false	Whether to enable stack tracing for discarded statements or connected code in an application. When enabled, stack traces for discarded statements and connection-related logs will be overwritten each time a connection is opened or a statement is created.
abandonedUsageTracking	false	When this parameter is set to **true**, the connection pool records stack traces each time a method is called on a pooled connection, retaining the latest stack trace to aid in debugging abandoned connections. **NOTE** Setting this parameter to **true** will increase the overhead. Exercise caution when performing this operation.
fastFailValidation	false	This parameter refers to the quick failure of validation statements if a fatal exception occurs, without executing **isValid()** or the validation query. Fatal exceptions include specific **SQL_STATE** codes. ● 57P01 (ADMIN SHUTDOWN) ● 57P02 (CRASH SHUTDOWN) ● 57P03 (CANNOT CONNECT NOW) ● 01002 (SQL92 disconnect error) ● JZ0C0 (Sybase disconnect error) ● JZ0C1 (Sybase disconnect error) ● Any SQL_STATE code that starts with "08" Exception codes need to be overwritten. For details, see **disconnectionSqlCodes**.
disconnectionSqlCodes	-	Exception code, which is an SQL_STATE code separated by commas (,). This parameter is valid only when **fastFailValidation** is set to **true**.
jmxName	-	This parameter registers a DataSource as a JMX MBean with a specified name that adheres to the JMX object name syntax.
registerConnectionMBean	true	Whether to register and connect to the JMX MBean.

----End

Configuring the Hikari CP Connection Pool

Step 1 Download the JDBC driver package. For details, see **Downloading the JDBC or ODBC Driver**.

- Download the HikariCP driver package from **https://mvnrepository.com/artifact/com.zaxxer/HikariCP/4.0.3**.
- Download the SLF4J driver package from **https://www.slf4j.org/download.html**.

Step 2 Add the JDBC, HikariCP, and SLF4J driver packages to the project and configure parameters related to the database connection pool.

Table 4-20 Hikari CP connection pool parameters

Parameter	Default Value	Description
driverClassName	Enter the value of **org.postgresql.Driver**.	Name of the database driver.
jdbcUrl	-	URL for connecting to the database.
username	-	Username.
password	-	Password.
autoCommit	true	Whether to automatically submit transactions when the connection returns to the connection pool.
connectionTimeout	30000	Maximum timeout interval for obtaining connections from the connection pool.
idleTimeout	60000	Maximum lifetime of an idle connection. This setting takes effect only when the value of **minimumIdle** is less than that of **maximumPoolSize**. ● If the number of idle connections is greater than the value of **minimumIdle** and the idle time of a connection is greater than the value of **idleTimeout**, the connection is deleted from the connection pool. ● **0** indicates no timeout.
keepaliveTime	0	Interval for checking whether idle connections are available, in milliseconds. **0** indicates that the function is disabled.
maxLifetime	1800000	Maximum connection lifetime, in milliseconds. **0** indicates no limit.

Parameter	Default Value	Description
connectionTestQuery	-	Query statement for connection detection.
minimumIdle	10	Minimum number of idle connections. To improve performance, you are advised not to set this parameter. The size of the connection pool is fixed.
maximumPoolSize	10	Maximum number of connections.
metricRegistry	-	This parameter can only be accessed through programmatic configuration or the IoC container. This parameter specifies the Codahale/ Dropwizard MetricRegistry instance used by the pool to record various metrics.
healthCheckRegistry	-	This parameter can only be accessed through programmatic configuration or the IoC container. This parameter specifies the Codahale/ Dropwizard HealthCheckRegistry instance used by the pool to record health information.
poolName	-	Name of a connection pool.
initializationFailTimeout	1	Whether the connection pool fails to initialize quickly. ● If the value is greater than 0, the system attempts to obtain a connection within the specified duration (**connectionTimeout** + **initializationFailTimeout**). If unsuccessful, the pool is not enabled, and an exception is thrown. ● If the value is 0, the system attempts to obtain and verify the connection. If verification fails, the pool is not enabled. ● If the value is less than 0, the pool starts without attempting connection initialization.
isolateInternalQueries	false	Whether to isolate HikariCP queries in a transaction. This setting takes effect when **autoCommit** is set to **false**.

Parameter	Default Value	Description
allowPoolSuspension	false	Whether to allow the connection pool to be suspended and resumed through JMX. When the connection pool is suspended, the connection does not time out until the connection pool is restored.
readOnly	false	Whether the connection is read-only.
registerMbeans	false	Whether to enable JMX.
catalog	-	Default database **catalog**.
connectionInitSql	-	SQL statement executed after the connection pool is initialized.
transactionIsolation	-	Default transaction isolation level.
validationTimeout	5000	Timeout interval for connection detection. The value must be greater than the value of **connectionTimeout**. The minimum value is **250**.
leakDetectionThreshold	0	Maximum duration a connection can be lent out. The minimum value is 2000 milliseconds, used for logging connection leakage.
schema	-	Default database **schema**.
threadFactory	-	The **java.util.concurrent.ThreadFactory** instance used by the connection pool for thread creation. This parameter can only be accessed through programmatic configuration or the IoC container.
scheduledExecutor	-	The **java.util.concurrent.ScheduledExecutor-Service** instance used by the connection pool to execute scheduled tasks. This parameter can only be accessed through programmatic configuration or the IoC container.

----End

Configuring the Druid Connection Pool

Step 1 Download the JDBC driver package. For details, see **Downloading the JDBC or ODBC Driver**.

Download the Druid driver package from **https://druid.apache.org/downloads/**.

Step 2 Add the JDBC and Druid driver packages to the project and configure parameters related to the database connection pool.

Table 4-21 Druid connection pool parameters

Parameter	Default Value	Description
url	-	URL for connecting to the database.
username	-	Username.
password	-	Password.
driverClassName	Enter the value of **org.postgresql.Driver**.	Name of the database driver.
initialSize	0	Number of physical connections established during initialization. Initialization occurs when the **init** method is invoked explicitly or when the **getConnection** method is invoked for the first time.
maxActive	8	Maximum number of connections in the thread pool.
minIdle	0	Minimum number of idle threads in the thread pool. Druid periodically scans the number of connections. If the number exceeds the specified parameter, redundant connections are closed. If fewer connections are available, new ones are created. This parameter helps manage connections during high request volumes, though it can be time-consuming.
connectTimeout	-	Timeout interval for connecting to the database, in milliseconds.
socketTimeout	-	Timeout interval for the socket to connect to the database, in milliseconds.
maxWait	-1	Waiting time for a new request when the connections in the connection pool are used up, in milliseconds. **–1** indicates infinite waiting until timeout occurs.
poolPreparedStatements	false	Whether to cache preparedStatement, that is, PSCache. The PSCache greatly improves the performance of the database that supports cursors.

Parameter	Default Value	Description
maxOpenPreparedStatements	-	If PSCache is enabled, the value of this parameter must be greater than **0**. If the value is greater than **0**, **poolPreparedStatements** will be automatically set to **true**.
validationQuery	SELECT 1	SQL statement used to check whether a connection is valid. If **validationQuery** is null, the **testOnBorrow**, **testOnReturn**, and **testWhileIdle** parameters do not take effect because the three parameters are used to verify the validity of the database connection by running the SQL statement specified by **validationQuery**.
testOnBorrow	-	When applying for a connection, the **validationQuery** command checks its validity. This configuration may reduce performance, so use it cautiously.
testOnReturn	-	When a connection is returned, the **validationQuery** command checks its validity. This configuration may also impact performance, so use it cautiously.
testWhileIdle	true	Whether a connection should be checked when it is requested. It is best to set this parameter to **true** to ensure security without compromising performance. If the idle time is greater than the value of **timeBetweenEvictionRunMills**, running the **validationQuery** command to verify the connection's validity will not have any effect.

Parameter	Default Value	Description
timeBetweenEvictionRunsMillis	60s	The **validationQuery** command checks connection validity. If the number of idle connections exceeds **minIdle**, redundant connections are closed. If fewer idle connections are available, new ones are added. Connections not used within the time specified by **timeBetweenEvictionRunsMillis** are disabled. This parameter also: 1. Sets the interval for the Destroy thread to check connections. 2. Functions as a reference for checking **testWhileIdle**. For details, see the description of the **testWhileIdle** attribute.
minEvictableIdleTimeMillis	30min	Maximum lifetime of an idle connection before eviction. If the time since the last activity exceeds **minEvictableIdleTimeMillis**, the connection is closed by the Destroy thread. **NOTE** This parameter conflicts with the **timeBetweenEvictionRunsMillis** parameter. You can leave this parameter empty.
connectionInitSqls	-	The SQL statement is executed when the physical connection is initialized.
exceptionSorter	-	When the database throws some unrecoverable exceptions, the connection is discarded.
filters	-	This parameter configures an extension plug-in using an alias. The attribute type is string. Common plug-ins include the filters used for monitoring and statistics: ● **stat**: monitoring statistics ● **log4j**: log record ● **wall**: SQL injection prevention
proxyFilters	-	The type is **List<com.alibaba.druid,filter.Filter>**. You can configure both **filter** and **proxyFilters**.

Parameter	Default Value	Description
removeAbandoned	false	Whether to reclaim leaked connections. When **getNumActive()** approaches **getMaxActive()**, the system reclaims invalid connections not used within the **removeAbandonedTimeout** period (300 seconds by default). Connections exceeding this timeout are forcibly closed.
removeAbandonedTimeout	300s	Time limit for Druid to forcibly reclaim connections, in seconds. Druid will forcibly reclaim a connection from the pool after a specified time has elapsed since the connection was established, starting from the moment the program retrieves the connection from the pool.
logAbandoned	false	Whether to print a log when reclaiming leaked connections. This parameter specifies whether to record the stack information of the current thread to logs when the **removeAbandoned** occurs.
removeAbandonedTimeoutMillis	5min	Timeout interval for reclaiming connections. If **removeAbandoned** is set to **true**, Druid periodically checks whether the thread pool overflows. If the thread pool is not in the running state and the specified time is exceeded, the thread pool is reclaimed.
maxEvictableIdleTimeMillis	7hours	Maximum idle time. The default value is 7 hours.
maxPoolPrepareStatementPerConnectionSize	20	Maximum number of SQL statements that can be cached for each connection.
keepAlive	false	Number of minIdle connections to maintain when the pool is initialized. If the number of connections falls below **minIdle** and idle time exceeds **minEvictableIdleTimeMillis**, the **keepAlive** operation is performed to maintain the **minIdle** value.

Parameter	Default Value	Description
notFullTimeoutRetryCount	0	Number of retry times when the sum of the number of lent connections in the connection pool and the number of available connections is less than the maximum allowed connections. The default value is **0**.
logSlowSql	false	Whether to print slow SQL statements. The value should be of the Boolean type.

----End

4.5.1.8 Using ODBC to Connect to a Cluster

GaussDB(DWS) allows you to use an ODBC driver to connect to the database through an ECS on the Huawei Cloud platform or over the Internet.

For details about how to use the ODBC API, see the official document.

Prerequisites

- You have downloaded ODBC driver packages **dws_x.x.x_odbc_driver_for_xxx.zip** (for Linux) and **dws_odbc_driver_for_windows.zip** (for Windows). For details, see **Downloading the JDBC or ODBC Driver**.

  GaussDB(DWS) also supports open-source ODBC driver: PostgreSQL ODBC 09.01.0200 or later.
- You have downloaded the open-source unixODBC code file 2.3.0 from **https://sourceforge.net/projects/unixodbc/files/unixODBC/2.3.0/unixODBC-2.3.0.tar.gz/download**.
- You have downloaded the SSL certificate file. For details, see **Downloading an SSL Certificate**.

Using an ODBC Driver to Connect to a Database (Linux)

Step 1 Upload the ODBC package and code file to the Linux environment and decompress them to the specified directory.

Step 2 Log in to the Linux environment as user **root**.

Step 3 Prepare **unixODBC**.

1. Decompress the **unixODBC** code file.
```
tar -xvf unixODBC-2.3.0.tar.gz
```
2. Compile the code file and install the driver.
```
cd unixODBC-2.3.0
./configure --enable-gui=no
make
make install
```

NOTE

- After the unixODBC is compiled and installed, the ***.so.2** library file will be in the installation directory. To create the ***.so.1** library file, change **LIB_VERSION** in the configure file to **1:0:0**.

```
LIB_VERSION="1:0:0"
```

- This driver dynamically loads the **libodbcinst.so.*** library files. If one of the library files is successfully loaded, the library file is loaded. The loading priority is **libodbcinst.so** > **libodbcinst.so.1** > **libodbcinst.so.1.0.0** > **libodbcinst.so.2** > **libodbcinst.so.2.0.0**.

  For example, a directory can be dynamically linked to **libodbcinst.so.1**, **libodbcinst.so.1.0.0**, and **libodbcinst.so.2**. The driver file loads **libodbcinst.so** first. If **libodbcinst.so** cannot be found in the current environment, the driver file searches for **libodbcinst.so.1**, which has a lower priority. After **libodbcinst.so.1** is loaded, the loading is complete.

Step 4 Replace the driver file. (This document uses the **dws_8.1.x_odbc_driver_for_x86_redhat.zip** package of Red Hat as an example.)

1. Decompress the **dws_8.1.x_odbc_driver_for_x86_redhat.zip** package.

```
unzip dws_8.1.x_odbc_driver_for_x86_redhat.zip
```

2. Copy all files in the **lib** directory to **/usr/local/lib**. If there are files with the same name, overwrite them.
3. Copy **psqlodbcw.la** and **psqlodbcw.so** in the **odbc/lib** directory to **/usr/local/lib**.

Step 5 Run the following command to modify the configuration of the driver file:

```
vi /usr/local/etc/odbcinst.ini
```

Copy the following content to the file:

```
[DWS]
Driver64=/usr/local/lib/psqlodbcw.so
```

The parameters are as follows:

- **[DWS]**: indicates the driver name. You can customize the name.
- **Driver64** or **Driver**: indicates the path where the dynamic library of the driver resides. For a 64-bit operating system, search for **Driver64** first. If **Driver64** is not configured, search for **Driver**.

Step 6 Run the following command to modify the data source file:

```
vi /usr/local/etc/odbc.ini
```

Copy the following content to the configuration file, save the modification, and exit.

```
[DWSODBC]
Driver=DWS
Servername=10.10.0.13
Database=gaussdb
Username=dbadmin
Password=password
Port=8000
Sslmode=allow
```

Parameter	Description	Example Value
[DSN]	Data source name.	[DWSODBC]

Parameter	Description	Example Value
Driver	Driver name, corresponding to **DriverName** in **odbcinst.ini**.	Driver=DWS
Servername	IP address of the server. When the cluster is bound to an ELB, set this parameter to the IP address of the ELB.	Servername=10.10.0.13
Database	Name of the database to be connected to.	Database=gaussdb
Username	Database username.	Username=dbadmin
Password	Database user password.	Password=*password*
Port	Port number of the server.	Port=8000

Parameter	Description	Example Value
Sslmode	SSL certification mode. This parameter is enabled for the cluster by default. Values and meanings: ● **disable**: only tries to establish a non-SSL connection. ● **allow**: tries establishing a non-SSL connection first, and then an SSL connection if the attempt fails. ● **prefer**: tries establishing an SSL connection first, and then a non-SSL connection if the attempt fails. ● **require**: only tries establishing an SSL connection. If there is a CA file, perform the verification according to the scenario in which the parameter is set to **verify-ca**. ● **verify-ca**: tries establishing an SSL connection and checks whether the server certificate is issued by a trusted CA. ● **verify-full**: not supported by GaussDB(DWS) **NOTE** The SSL mode delivers higher security than the common mode. By default, the SSL function is enabled in a cluster to allow SSL or non-SSL connections from the client. You are advised to use the SSL mode when using ODBC to connect to a GaussDB(DWS) cluster.	Sslmode=allow

NOTE

- You can view the values of **Servername** and **Port** on the GaussDB(DWS) management console. Log in to the GaussDB(DWS) management console and click **Client Connections**. In the **Data Warehouse Connection String** area, select the target cluster and obtain **Private Network Address** or **Public Network Address**. For details, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**.

Step 7 Configure environment variables.

```
vi ~/.bashrc
```

Add the following information to the configuration file:

```
export LD_LIBRARY_PATH=/usr/local/lib/:$LD_LIBRARY_PATH
export ODBCSYSINI=/usr/local/etc
export ODBCINI=/usr/local/etc/odbc.ini
```

Step 8 Import environment variables.

```
source ~/.bashrc
```

Step 9 Run the following commands to connect to the database:

```
/usr/local/bin/isql -v DWSODBC
```

If the following information is displayed, the connection is successful:

```
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
SQL>
```

----End

Using an ODBC Driver to Connect to a Database (Windows)

Step 1 Decompress ODBC driver package **dws_odbc_driver_for_windows.zip** (for Windows) and install **psqlodbc.msi**.

Step 2 Decompress the SSL certificate package to obtain the certificate file.

You have the option to deploy the certificate either automatically or manually, depending on your requirements.

- Automatic deployment:

  Double-click the **sslcert_env.bat** file to trigger automatic deployment of the certificate to a default location.

  NOTE

  - The **sslcert_env.bat** file ensures the purity of the certificate environment. When the **%APPDATA%\postgresql** directory exists, a message will be prompted asking you whether you want to remove related directories. If you want to remove related directories, back up files in the directory.

- Manual deployment:
  - Create a new folder named **postgresql** in the **%APPDATA%** directory.
  - Copy files **client.crt**, **client.key**, **client.key.cipher**, and **client.key.rand** to the **%APPDATA%\postgresql** directory and change **client** in the file name to **postgres**. For example, change the name of **client.key** to **postgres.key**.
  - Copy **cacert.pem** to **%APPDATA%\postgresql** and change the name of **cacert.pem** to **root.crt**.

Step 3 Open Driver Manager.

GaussDB(DWS) provides 32-bit and 64-bit ODBC drivers. Choose the version suitable for your system when configuring the data source. (Assume the Windows system drive is drive C. If another disk drive is used, modify the path accordingly.)

- If you want to develop 32-bit programs in the 64-bit OS and have installed the 32-bit driver, open the 32-bit Driver Manager at **C:\Windows\SysWOW64\odbcad32.exe**.

  Do not choose **Control Panel** > **System and Security** > **Administrative Tools** > **Data Sources (ODBC)** directly.

  NOTE

  WOW64 is the acronym for Windows 32-bit on Windows 64-bit. **C:\Windows\SysWOW64\** stores the 32-bit environment on a 64-bit system.

- If you want to develop 64-bit programs in the 64-bit OS and have installed the 64-bit driver, open the 64-bit Driver Manager at **C:\Windows\System32\odbcad32.exe**.

  Do not choose **Control Panel** > **System and Security** > **Administrative Tools** > **Data Sources (ODBC)** directly.

  NOTE

  C:\Windows\System32\ stores the environment consistent with the current OS. For technical details, see Windows technical documents.

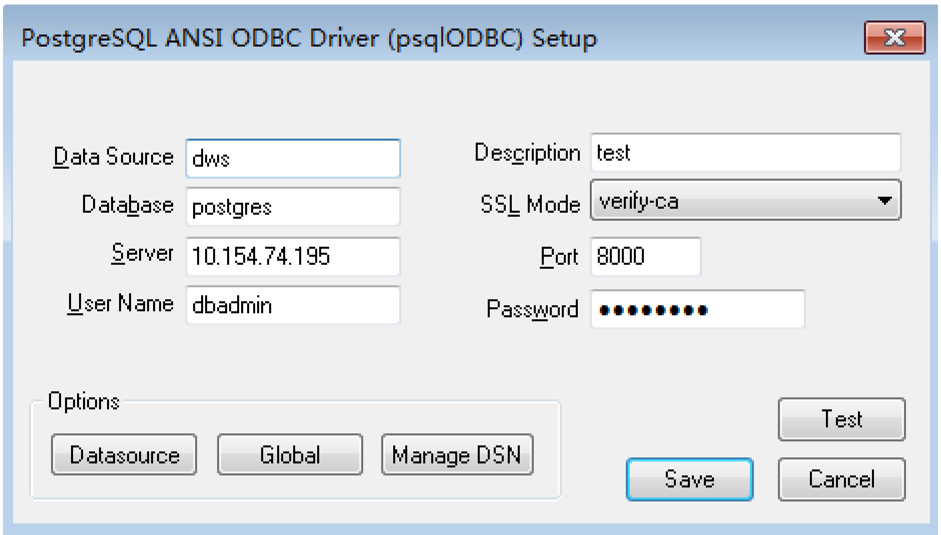
- In a 32-bit OS, open **C:\Windows\System32\odbcad32.exe**.

  Alternatively, click **Computer**, and choose **Control Panel**. Click **Administrative Tools** and click **Data Sources (ODBC)**.

Step 4 Configure a data source to be connected to.

1. On the **User DSN** tab, click **Add** and choose **PostgreSQL Unicode** for setup.

Figure 4-41 Configuring a data source to be connected to



PostgreSQL ANSI ODBC Driver (psqlODBC) Setup

Data Source: dws
Description: test
Database: postgres
SSL Mode: verify-ca
Server: 10.154.74.195
Port: 8000
User Name: dbadmin
Password: ••••••••

You can view the values of **Server** and **Port** on the GaussDB(DWS) management console. Log in to the GaussDB(DWS) console and click **Client Connections**. In the **Data Warehouse Connection String** area, select the target cluster and obtain **Private Network Address** or **Public Network**

Address. For details, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**.

2. Click **Test** to verify that the connection is correct. If **Connection successful** is displayed, the connection is correct.

Step 5 Compile an ODBC sample program to connect to the data source.

The ODBC API does not provide the database connection retry capability. You need to implement the connection retry processing in the service code.

The sample code is as follows:

```
// This example shows how to obtain GaussDB(DWS) data through the ODBC driver.
// DBtest.c (compile with: libodbc.so)
#include <stdlib.h>
#include <stdio.h>
#include <sqlext.h>
#ifdef WIN32
#include <windows.h>
#endif
SQLHENV       V_OD_Env;        // Handle ODBC environment
SQLHSTMT      V_OD_hstmt;      // Handle statement
SQLHDBC       V_OD_hdbc;       // Handle connection
char          typename[100];
SQLINTEGER    value = 100;
SQLINTEGER    V_OD_erg,V_OD_buffer,V_OD_err,V_OD_id;
int main(int argc,char *argv[])
{
    // 1. Apply for an environment handle.
    V_OD_erg = SQLAllocHandle(SQL_HANDLE_ENV,SQL_NULL_HANDLE,&V_OD_Env);
    if ((V_OD_erg != SQL_SUCCESS) && (V_OD_erg != SQL_SUCCESS_WITH_INFO))
    {
        printf("Error AllocHandle\n");
        exit(0);
    }
    // 2. Set environment attributes (version information).
    SQLSetEnvAttr(V_OD_Env, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0);
    // 3. Apply for a connection handle.
    V_OD_erg = SQLAllocHandle(SQL_HANDLE_DBC, V_OD_Env, &V_OD_hdbc);
    if ((V_OD_erg != SQL_SUCCESS) && (V_OD_erg != SQL_SUCCESS_WITH_INFO))
    {
        SQLFreeHandle(SQL_HANDLE_ENV, V_OD_Env);
        exit(0);
    }
    // 4. Set connection attributes.
    SQLSetConnectAttr(V_OD_hdbc, SQL_ATTR_AUTOCOMMIT, SQL_AUTOCOMMIT_ON, 0);
    // 5. Connect to a data source. You do not need to enter the username and password if you have
configured them in the odbc.ini file. If you have not configured them, specify the name and password of
the user who wants to connect to the database in the SQLConnect function.
    V_OD_erg = SQLConnect(V_OD_hdbc, (SQLCHAR*) "gaussdb", SQL_NTS,
                          (SQLCHAR*) "", SQL_NTS,  (SQLCHAR*) "", SQL_NTS);
    if ((V_OD_erg != SQL_SUCCESS) && (V_OD_erg != SQL_SUCCESS_WITH_INFO))
    {
       printf("Error SQLConnect %d\n",V_OD_erg);
       SQLFreeHandle(SQL_HANDLE_ENV, V_OD_Env);
       exit(0);
    }
    printf("Connected !\n");
    // 6. Set statement attributes.
    SQLSetStmtAttr(V_OD_hstmt,SQL_ATTR_QUERY_TIMEOUT,(SQLPOINTER *)3,0);
    // 7. Apply for a statement handle.
    SQLAllocHandle(SQL_HANDLE_STMT, V_OD_hdbc, &V_OD_hstmt);
    // 8. Executes an SQL statement directly.
    SQLExecDirect(V_OD_hstmt,"drop table IF EXISTS testtable",SQL_NTS);
    SQLExecDirect(V_OD_hstmt,"create table testtable(id int)",SQL_NTS);
    SQLExecDirect(V_OD_hstmt,"insert into testtable values(25)",SQL_NTS);
    // 9. Prepare for execution.
    SQLPrepare(V_OD_hstmt,"insert into testtable values(?)",SQL_NTS);
    // 10. Bind parameters.
    SQLBindParameter(V_OD_hstmt,1,SQL_PARAM_INPUT,SQL_C_SLONG,SQL_INTEGER,0,0,
                 &value,0,NULL);
    // 11. Execute the ready statement.
    SQLExecute(V_OD_hstmt);
    SQLExecDirect(V_OD_hstmt,"select id from testtable",SQL_NTS);
    // 12. Obtain the attributes of a certain column in the result set.
    SQLColAttribute(V_OD_hstmt,1,SQL_DESC_TYPE,typename,100,NULL,NULL);
    printf("SQLColAtrribute %s\n",typename);
    // 13. Bind the result set.
    SQLBindCol(V_OD_hstmt,1,SQL_C_SLONG, (SQLPOINTER)&V_OD_buffer,150,
            (SQLLEN *)&V_OD_err);
    // 14. Collect data using SQLFetch.
    V_OD_erg=SQLFetch(V_OD_hstmt);
    // 15. Obtain and return data using SQLGetData.
    while(V_OD_erg != SQL_NO_DATA)
    {
        SQLGetData(V_OD_hstmt,1,SQL_C_SLONG,(SQLPOINTER)&V_OD_id,0,NULL);
        printf("SQLGetData ----ID = %d\n",V_OD_id);
        V_OD_erg=SQLFetch(V_OD_hstmt);
    };
    printf("Done !\n");
    // 16. Disconnect from the data source and release handles.
    SQLFreeHandle(SQL_HANDLE_STMT,V_OD_hstmt);
    SQLDisconnect(V_OD_hdbc);
    SQLFreeHandle(SQL_HANDLE_DBC,V_OD_hdbc);
    SQLFreeHandle(SQL_HANDLE_ENV, V_OD_Env);
    return(0);
}
```

----End

4.5.2 Using the Third-Party Function Library psycopg2 of Python to Connect to a Cluster

After creating a GaussDB(DWS) cluster and using the third-party database adapter psycopg2 to connect to the cluster, you can use Python to access GaussDB(DWS) and perform various operations on data tables.

Preparations Before Connecting to a Cluster

- An EIP has been bound to the GaussDB(DWS) cluster.
- You have obtained the administrator username and password for logging in to the database in the GaussDB(DWS) cluster.

  MD5 algorithms may by vulnerable to collision attacks and cannot be used for password verification. Currently, GaussDB(DWS) uses the default security design. By default, MD5 password verification is disabled, and this may cause failures of connections from open source clients. You are advised to check whether the value of **password_encryption_type** is **1**. If the value is not **1**, change it. For how to change the value, see **Modifying GUC Parameters of the GaussDB(DWS) Cluster**. Then change the password of the database user to be used.

NOTE

- For security purposes, GaussDB(DWS) no longer uses MD5 to store password digests by default. As a result, the open-source drives and clients may fail to connect to the database. To use the MD5 algorithm used in an open-source protocol, you must modify your password policy and create a new user, or change the password of an existing user.
- The database stores the hash digest of passwords instead of password text. During password verification, the system compares the hash digest with the password digest sent from the client (salt operations are involved). If you change your cryptographic algorithm policy, the database cannot generate a new hash digest for your existing password. For connectivity purposes, you must manually change your password or create a new user. The new password will be encrypted using the hash algorithm and stored for authentication in the next connection.

- You have obtained the public network address, including the IP address and port number in the GaussDB(DWS) cluster. For details, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**.
- You have installed the third-party database adapter Psycopg2. Download address: **https://pypi.org/project/psycopg2/**. For details about installation and deployment, see **https://www.psycopg.org/install/**.

NOTE

- In CentOS and Red Hat OS, run the following **yum** command:

```
yum install python-psycopg2
```

- Psycopg2 depends on the libpq dynamic library of PostgreSQL (32-bit or 64-bit version, whichever matches the psycopg2 bit version). In Linux, you can run the **yum** command and do not need to install the library. Before using Psycopg2 in Windows, install libpq in either of the following ways:
  - Install PostgreSQL and configure the libpq, ssl, and crypto dynamic libraries in the environment variable **PATH**.
  - Install psqlodbc and use the libpq, ssl, and crypto dynamic libraries carried by the PostgreSQL ODBC driver.

Version

There are many versions of GaussDB(DWS) clusters, Python, and Psycopg2. The following table lists only the supported mainstream versions.

Table 4-22

Psycopg2 Version	Python Version	GaussDB(DWS) Cluster Version
2.7.x	3.8.x	8.1.3 or later
	3.9.x	8.1.3 or later
2.8.x	3.8.x	8.1.3 or later
	3.9.x	8.1.3 or later
2.9.x	3.8.x	8.1.3 or later
	3.9.x	8.1.3 or later

Constraints

Psycopg2 is a PostgreSQL-based client interface, and its functions are not fully supported by GaussDB(DWS). For details, see **Table 4-23**.

NOTE

The following APIs are supported based on Python 3.8.5 and Psycopg 2.9.1.

Table 4-23 Psycopg2 APIs supported by GaussDB(DWS)

Class Name	Usage	Function/Member Variable	Support	Remarks
connections	basic	cursor(*name=None*, *cursor_factory=None*, *scrollable=None*, *withhold=False*)	Y	-
		commit()	Y	-
		rollback()	Y	-
		close()	Y	-
	Two-phase commit support methods	xid(*format_id*, *gtrid*, *bqual*)	Y	-
		tpc_begin(*xid*)	Y	-
		tpc_prepare()	N	The kernel does not support explicit **PREPARE TRANSACTION**.
		tpc_commit([*xid*])	Y	-
		tpc_rollback([*xid*])	Y	-
		tpc_recover()	Y	-
		closed	Y	-
		cancel()	Y	-
		reset()	N	**DISCARD ALL** is not supported.
		dsn	Y	-

Class Name	Usage	Function/Member Variable	Support	Remarks
	Transaction control methods and attributes.	set_session(*isolation_level=None*, *readonly=None*, *deferrable=None*, *autocommit=None*)	Y	The database does not support the setting of **default_transaction_read_only** in a session.
		autocommit	Y	-
		isolation_level	Y	-
		readonly	N	The database does not support the setting of **default_transaction_read_only** in a session.
		deferrable	Y	-
		set_isolation_level(*level*)	Y	-
		encoding	Y	-
		set_client_encoding(enc)	Y	-
		notices	N	The database does not support **listen**/**notify**.
		notifies	Y	-
		cursor_factory	Y	-
		info	Y	-
		status	Y	-
		lobject	N	The database does not support operations related to large objects.

Class Name	Usage	Function/Member Variable	Support	Remarks
	Methods related to asynchronous support	poll()	Y	-
		fileno()	Y	-
		isexecuting()	Y	-
	Interoperation with other C API modules	pgconn_ptr	Y	-
		get_native_connection()	Y	-
	informative methods of the native connection	get_transaction_status()	Y	-
		protocol_version	Y	-
		server_version	Y	-
		get_backend_pid()	Y	The obtained PID is not the background PID, but the ID of the logical connection.
		get_parameter_status(parameter)	Y	-
		get_dsn_parameters()	Y	-
cursor	basic	description	Y	-
		close()	Y	-
		closed	Y	-
		connection	Y	-
		name	Y	-
		scrollable	N	The database does not support **SCROLL CURSOR**.
		withhold	N	The **withhold cursor** needs to be closed before the commit operation.

Class Name	Usage	Function/Member Variable	Support	Remarks
	Commands execution methods	execute(*query*, *vars=None*)	Y	-
		executemany(*query*, *vars_list*)	Y	-
		callproc(*procname*[, *parameters*])	Y	-
		mogrify(*operation*[, *parameters*])	Y	-
		setinputsizes(*sizes*)	Y	-
		fetchone()	Y	-
		fetchmany([*size=cursor.arraysize*])	Y	-
		fetchall()	Y	-
		scroll(*value*[, *mode='relative'*])	N	The database does not support **SCROLL CURSOR**.
		arraysize	Y	-
		itersize	Y	-
		rowcount	Y	-
		rownumber	Y	-
		lastrowid	Y	-
		query	Y	-
		statusmessage	Y	-
		cast(*oid*, *s*)	Y	-
		tzinfo_factory	Y	-
		nextset()	Y	-
		setoutputsize(*size*[, *column*])	Y	-
	COPY-related methods	copy_from(*file*, *table*, *sep='||t'*, *null='||||N'*, *size=8192*, *columns=None*)	Y	-
		copy_to(*file*, *table*, *sep='||t'*, *null='||||N'*, *columns=None*)	Y	-
		copy_expert(*sql*, *file*, *size=8192*)	Y	-

Class Name	Usage	Function/Member Variable	Support	Remarks
	Interoperation with other C API modules	pgresult_ptr	Y	-

Using the Third-Party Function Library psycopg2 to Connect to a Cluster (Linux)

Step 1 Log in to the Linux environment as user **root**.

Step 2 Run the following command to create the **python_dws.py** file:

```
vi python_dws.py
```

Copy and paste the following content to the **python_dws.py** file:

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from __future__ import print_function

import psycopg2


def create_table(connection):
    print("Begin to create table")
    try:
        cursor = connection.cursor()
        cursor.execute("drop table if exists test;"
                       "create table test(id int, name text);")
        connection.commit()
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("Table created successfully")
        cursor.close()


def insert_data(connection):
    print("Begin to insert data")
    try:
        cursor = connection.cursor()
        cursor.execute("insert into test values(1,'number1');")
        cursor.execute("insert into test values(2,'number2');")
        cursor.execute("insert into test values(3,'number3');")
        connection.commit()
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("Insert data successfully")
        cursor.close()


def update_data(connection):
    print("Begin to update data")
    try:
        cursor = connection.cursor()
        cursor.execute("update test set name = 'numberupdated' where id=1;")
        connection.commit()
        print("Total number of rows updated :", cursor.rowcount)
        cursor.execute("select * from test order by 1;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("After Update, Operation done successfully")


def delete_data(connection):
    print("Begin to delete data")
    try:
        cursor = connection.cursor()
        cursor.execute("delete from test where id=3;")
        connection.commit()
        print("Total number of rows deleted :", cursor.rowcount)
        cursor.execute("select * from test order by 1;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("After Delete,Operation done successfully")


def select_data(connection):
    print("Begin to select data")
    try:
        cursor = connection.cursor()
        cursor.execute("select * from test order by 1;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except psycopg2.ProgrammingError as e:
        print(e)
        print("select failed")
    else:
        print("Operation done successfully")
        cursor.close()


if __name__ == '__main__':
    try:
        conn = psycopg2.connect(host='10.154.70.231',
                                port='8000',
                                database='gaussdb',  # Database to be connected
                                user='dbadmin',
                                password='password')  # Database user password
    except psycopg2.DatabaseError as ex:
        print(ex)
        print("Connect database failed")
    else:
        print("Opened database successfully")
        create_table(conn)
        insert_data(conn)
        select_data(conn)
        update_data(conn)
        delete_data(conn)
        conn.close()
```

Step 3 Change the public network address, cluster port number, database name, database username, and database password in the **python_dws.py** file based on the actual cluster information.

The psycopg2 API does not provide the connection retry capability. You need to implement the retry processing in the service code.

```
    conn = psycopg2.connect(host='10.154.70.231',
                            port='8000',
                            database='gaussdb',  # Database to be connected
                            user='dbadmin',
                            password='password')  # Database user password
```

Step 4 Connect to the cluster using the third-party database adapter Psycopg.

```
python python_dws.py
```

----End

Using the Third-Party Function Library psycopg2 to Connect to a Cluster (Windows)

Step 1 In the Windows operating system, click the **Start** button, enter **cmd** in the search box, and click **cmd.exe** in the result list to open the command-line interface (CLI).

Step 2 In the CLI, run the following command to create the **python_dws.py** file:

```
type nul> python_dws.py
```

Copy and paste the following content to the **python_dws.py** file:

```
#!/usr/bin/python
# -*- coding:UTF-8 -*-

from __future__ import print_function

import psycopg2


def create_table(connection):
    print("Begin to create table")
    try:
        cursor = connection.cursor()
        cursor.execute("drop table if exists test;"
                       "create table test(id int, name text);")
        connection.commit()
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("Table created successfully")
        cursor.close()


def insert_data(connection):
    print("Begin to insert data")
    try:
        cursor = connection.cursor()
        cursor.execute("insert into test values(1,'number1');")
        cursor.execute("insert into test values(2,'number2');")
        cursor.execute("insert into test values(3,'number3');")
        connection.commit()
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("Insert data successfully")
        cursor.close()


def update_data(connection):
    print("Begin to update data")
    try:
        cursor = connection.cursor()
        cursor.execute("update test set name = 'numberupdated' where id=1;")
        connection.commit()
        print("Total number of rows updated :", cursor.rowcount)
        cursor.execute("select * from test order by 1;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("After Update, Operation done successfully")


def delete_data(connection):
    print("Begin to delete data")
    try:
        cursor = connection.cursor()
        cursor.execute("delete from test where id=3;")
        connection.commit()
        print("Total number of rows deleted :", cursor.rowcount)
        cursor.execute("select * from test order by 1;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except psycopg2.ProgrammingError as e:
        print(e)
    else:
        print("After Delete,Operation done successfully")


def select_data(connection):
    print("Begin to select data")
    try:
        cursor = connection.cursor()
        cursor.execute("select * from test order by 1;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except psycopg2.ProgrammingError as e:
        print(e)
        print("select failed")
    else:
        print("Operation done successfully")
        cursor.close()


if __name__ == '__main__':
    try:
        conn = psycopg2.connect(host='10.154.70.231',
                                port='8000',
                                database='postgresgaussdb',  # Database to be connected
                                user='dbadmin',
                                password='password')  # Database user password
    except psycopg2.DatabaseError as ex:
        print(ex)
        print("Connect database failed")
    else:
        print("Opened database successfully")
        create_table(conn)
        insert_data(conn)
        select_data(conn)
        update_data(conn)
        delete_data(conn)
        conn.close()
```

Step 3 Change the public network address, cluster port number, database name, database username, and database password in the **python_dws.py** file based on the actual cluster information.

```
    conn = psycopg2.connect(host='10.154.70.231',
                            port='8000',
                            database='gaussdb',  # Database to be connected
                            user='dbadmin',
                            password='password')  # Database user password
```

Step 4 On the CLI, run the following command to use Psycopg to connect to the cluster:

```
python python_dws.py
```

----End

Why CN Retry Is Not Supported When Psycopg2 Is Connected to a Cluster?

With the CN retry feature, GaussDB(DWS) retries a statement that failed to be executed and identifies the failure type. For details, see **Automatic Retry upon SQL Statement Execution Errors**. However, in a session connected using Psycopg2, a failed SQL statement will report an error and stop to be executed. In a primary/standby switchover, if a failed SQL statement is not retried, the following error will be reported. If the switchover is complete during an automatic retry, the correct result will be returned.

```
psycopg2.errors.ConnectionFailure: pooler: failed to create 1 connections, Error Message: remote node
dn_6003_6004, detail: could not connect to server: Operation now in progress
```

Error causes:

1. Psycopg2 sends the **BEGIN** statement to start a transaction before sending an SQL statement.
2. CN retry does not support statements in transaction blocks.

Solution:

- In synchronous connection mode, end the transaction started by the driver.

```
cursor = conn.cursor()
# End the transaction started by the driver.
cursor.execute("end; select * from test order by 1;")
rows = cursor.fetchall()
```

- Start a transaction in an asynchronous connection. For details, visit the PyScopg official website at: **https://www.psycopg.org/docs/advanced.html?highlight=async**

```
#!/usr/bin/env python3
# _*_ encoding=utf-8 _*_

import psycopg2
import select

# Wait function provided by Psycopg2 in asynchronous connection mode
#For details, see https://www.psycopg.org/docs/advanced.html?highlight=async.
def wait(conn):
    while True:
        state = conn.poll()
        if state == psycopg2.extensions.POLL_OK:
            break
        elif state == psycopg2.extensions.POLL_WRITE:
            select.select([], [conn.fileno()], [])
        elif state == psycopg2.extensions.POLL_READ:
            select.select([conn.fileno()], [], [])
        else:
            raise psycopg2.OperationalError("poll() returned %s" % state)
```

```
def psycopg2_cnretry_sync():
    # Create a connection.
    conn = psycopg2.connect(host='10.154.70.231',
                            port='8000',
                            database='gaussdb',  # Database to be connected
                            user='dbadmin',
                            password='password',  # Database user password
                            async=1) # Use the asynchronous connection mode.
    wait(conn)

    # Execute a query.
    cursor = conn.cursor()
    cursor.execute("select * from test order by 1;")
    wait(conn)
    rows = cursor.fetchall()
    for row in rows:
        print(row[0], row[1])

    # Close the connection.
    conn.close()

if __name__ == '__main__':
    psycopg2_cnretry_async()
```

4.5.3 Using the Python Library PyGreSQL to Connect to a Cluster

After creating a data warehouse cluster and using the third-party function library PyGreSQL to connect to the cluster, you can use Python to access GaussDB(DWS) and perform various operations on data tables.

Preparations Before Connecting to a Cluster

- An EIP has been bound to the data warehouse cluster.
- You have obtained the administrator username and password for logging in to the database in the data warehouse cluster.

  MD5 algorithms may by vulnerable to collision attacks and cannot be used for password verification. Currently, GaussDB(DWS) uses the default security design. By default, MD5 password verification is disabled, and this may cause failures of connections from open source clients. You are advised to check whether the value of **password_encryption_type** is **1**. If the value is not **1**, change it. For how to change the value, see **Modifying GUC Parameters of the GaussDB(DWS) Cluster**. Then change the password of the database user to be used.

  NOTE

  - For security purposes, GaussDB(DWS) no longer uses MD5 to store password digests by default. As a result, the open-source drives and clients may fail to connect to the database. To use the MD5 algorithm used in an open-source protocol, you must modify your password policy and create a new user, or change the password of an existing user.
  - The database stores the hash digest of passwords instead of password text. During password verification, the system compares the hash digest with the password digest sent from the client (salt operations are involved). If you change your cryptographic algorithm policy, the database cannot generate a new hash digest for your existing password. For connectivity purposes, you must manually change your password or create a new user. The new password will be encrypted using the hash algorithm and stored for authentication in the next connection.

- You have obtained the public network address, including the IP address and port number in the data warehouse cluster. For details, see **Obtaining the Connection Address of a GaussDB(DWS) Cluster**.
- You have installed the third-party function library PyGreSQL.

  Download address: **http://www.pygresql.org/download/index.html**
- For details about the installation and deployment operations, see **http://www.pygresql.org/contents/install.html**

NOTE

- In CentOS and Red Hat OS, run the following **yum** command:

```
yum install PyGreSQL
```

- PyGreSQL depends on the libpq dynamic library of PostgreSQL (32-bit or 64-bit version, whichever matches the PyGreSQL bit version). In Linux, you can run the **yum** command and do not need to install the library. Before using PyGreSQL in Windows, you need to install libpq in either of the following ways:
  - Install PostgreSQL and configure the libpq, ssl, and crypto dynamic libraries in the environment variable **PATH**.
  - Install **psqlodbc** and use the **libpq**, **ssl**, and **crypto** dynamic libraries carried by the PostgreSQL ODBC driver.

Constraints

PyGreSQL is a PostgreSQL-based client interface, and its functions are not fully supported by GaussDB(DWS). For details, see **Table 4-24**.

NOTE

The following APIs are supported based on Python 3.8.5 and PyGreSQL 5.2.4.

Table 4-24 PyGreSQL APIs supported by DWS

PyGreSQL		Yes	Remarks
Module functions and constants	connect – Open a PostgreSQL connection	Y	-
	get_pqlib_version – get the version of libpq	Y	-
	get/set_defhost – default server host [DV]	Y	-
	get/set_defport – default server port [DV]	Y	-
	get/set_defopt – default connection options [DV]	Y	-
	get/set_defbase – default database name [DV]	Y	-
	get/set_defuser – default database user [DV]	Y	-

PyGreSQL		Yes	Remarks
	get/set_defpasswd – default database password [DV]	Y	-
	escape_string – escape a string for use within SQL	Y	-
	escape_bytea – escape binary data for use within SQL	Y	-
	unescape_bytea – unescape data that has been retrieved as text	Y	-
	get/set_namedresult – conversion to named tuples	Y	-
	get/set_decimal – decimal type to be used for numeric values	Y	-
	get/set_decimal_point – decimal mark used for monetary values	Y	-
	get/set_bool – whether boolean values are returned as bool objects	Y	-
	get/set_array – whether arrays are returned as list objects	Y	-
	get/set_bytea_escaped – whether bytea data is returned escaped	Y	-
	get/set_jsondecode – decoding JSON format	Y	-
	get/set_cast_hook – fallback typecast function	Y	-
	get/set_datestyle – assume a fixed date style	Y	-
	get/set_typecast – custom typecasting	Y	-
	cast_array/record – fast parsers for arrays and records	Y	-
	Type helpers	Y	-

PyGreSQL		Yes	Remarks
	Module constants	Y	-
Connection – The connection object	query – execute a SQL command string	Y	-
	send_query - executes a SQL command string asynchronously	Y	-
	query_prepared – execute a prepared statement	Y	-
	prepare – create a prepared statement	Y	-
	describe_prepared – describe a prepared statement	Y	-
	reset – reset the connection	Y	-
	poll - completes an asynchronous connection	Y	-
	cancel – abandon processing of current SQL command	Y	-
	close – close the database connection	Y	-
	transaction – get the current transaction state	Y	-
	parameter – get a current server parameter setting	Y	-
	date_format – get the currently used date format	Y	-
	fileno – get the socket used to connect to the database	Y	-
	set_non_blocking - set the non-blocking status of the connection	Y	-
	is_non_blocking - report the blocking status of the connection	Y	-
	getnotify – get the last notify from the server	N	The database does not support **listen**/**notify**.

PyGreSQL		Yes	Remarks
	inserttable – insert a list into a table	Y	Use double quotation marks ("") to quote **\n** in the **copy** command.
	get/set_notice_receiver – custom notice receiver	Y	-
	putline – write a line to the server socket [DA]	Y	-
	getline – get a line from server socket [DA]	Y	-
	endcopy – synchronize client and server [DA]	Y	-
	locreate – create a large object in the database [LO]	N	Operations related to large objects
	getlo – build a large object from given oid [LO]	N	Operations related to large objects
	loimport – import a file to a large object [LO]	N	Operations related to large objects
	Object attributes	Y	-
The DB wrapper class	Initialization	Y	-
	pkey – return the primary key of a table	Y	-
	get_databases – get list of databases in the system	Y	-
	get_relations – get list of relations in connected database	Y	-
	get_tables – get list of tables in connected database	Y	-
	get_attnames – get the attribute names of a table	Y	-
	has_table_privilege – check table privilege	Y	-

PyGreSQL		Yes	Remarks
	get/set_parameter – get or set run-time parameters	Y	-
	begin/commit/rollback/ savepoint/release – transaction handling	Y	-
	get – get a row from a database table or view	Y	-
	insert – insert a row into a database table	Y	-
	update – update a row in a database table	Y	-
	upsert – insert a row with conflict resolution	Y	-
	query – execute a SQL command string	Y	-
	query_formatted – execute a formatted SQL command string	Y	-
	query_prepared – execute a prepared statement	Y	-
	prepare – create a prepared statement	Y	-
	describe_prepared – describe a prepared statement	Y	-
	delete_prepared – delete a prepared statement	Y	-
	clear – clear row values in memory	Y	-
	delete – delete a row from a database table	Y	A tuple must have unique key or primary key.
	truncate – quickly empty database tables	Y	-
	get_as_list/dict – read a table as a list or dictionary	Y	-

PyGreSQL		Yes	Remarks
	escape_literal/identifier/string/bytea – escape for SQL	Y	-
	unescape_bytea – unescape data retrieved from the database	Y	-
	encode/decode_json – encode and decode JSON data	Y	-
	use_regtypes – determine use of regular type names	Y	-
	notification_handler – create a notification handler	N	The database does not support **listen**/**notify**.
	Attributes of the DB wrapper class	Y	-
Query methods	getresult – get query values as list of tuples	Y	-
	dictresult/dictiter – get query values as dictionaries	Y	-
	namedresult/namediter – get query values as named tuples	Y	-
	scalarresult/scalariter – get query values as scalars	Y	-
	one/onedict/onenamed/onescalar – get one result of a query	Y	-
	single/singledict/singlenamed/singlescalar – get single result of a query	Y	-
	listfields – list fields names of previous query result	Y	-
	fieldname, fieldnum – field name/number conversion	Y	-
	fieldinfo – detailed info about query result fields	Y	-
	ntuples – return number of tuples in query object	Y	-

PyGreSQL		Yes	Remarks
	memsize – return number of bytes allocated by query result	Y	-
LargeObject – Large Objects	open – open a large object	N	Operations related to large objects
	close – close a large object	N	Operations related to large objects
	read, write, tell, seek, unlink – file-like large object handling	N	Operations related to large objects
	size – get the large object size	N	Operations related to large objects
	export – save a large object to a file	N	Operations related to large objects
	Object attributes	N	Operations related to large objects
The Notification Handler	Instantiating the notification handler	N	The database does not support **listen**/**notify**.
	Invoking the notification handler	N	The database does not support **listen**/**notify**.
	Sending notifications	N	The database does not support **listen**/**notify**.
	Auxiliary methods	N	The database does not support **listen**/**notify**.
pgdb			
Module functions and constants	connect – Open a PostgreSQL connection	Y	-

PyGreSQL		Yes	Remarks
	get/set/reset_typecast – Control the global typecast functions	Y	-
	Module constants	Y	-
	Errors raised by this module	Y	-
Connection – The connection object	close – close the connection	Y	-
	commit – commit the connection	Y	-
	rollback – roll back the connection	Y	-
	cursor – return a new cursor object	Y	-
	Attributes that are not part of the standard	Y	-
Cursor – The cursor object	description – details regarding the result columns	Y	-
	rowcount – number of rows of the result	Y	-
	close – close the cursor	Y	-
	execute – execute a database operation	Y	-
	executemany – execute many similar database operations	Y	-
	callproc – Call a stored procedure	Y	-
	fetchone – fetch next row of the query result	Y	-
	fetchmany – fetch next set of rows of the query result	Y	-
	fetchall – fetch all rows of the query result	Y	-
	arraysize - the number of rows to fetch at a time	Y	-

PyGreSQL		Yes	Remarks
	Methods and attributes that are not part of the standard	Y	-
Type – Type objects and constructors	Type constructors	Y	-
	Type objects	Y	-

Using the Third-Party Function Library PyGreSQL to Connect to a Cluster (Linux)

Step 1 Log in to the Linux environment as user **root**.

Step 2 Run the following command to create the **python_dws.py** file:

```
vi python_dws.py
```

Copy and paste the following content to the **python_dws.py** file:

```
#!/usr/bin/env python3
# _*_ encoding:utf-8 _*_

from __future__ import print_function

import pg


def create_table(connection):
    print("Begin to create table")
    try:
        connection.query("drop table if exists test;"
                         "create table test(id int, name text);")
    except pg.InternalError as e:
        print(e)
    else:
        print("Table created successfully")


def insert_data(connection):
    print("Begin to insert data")
    try:
        connection.query("insert into test values(1,'number1');")
        connection.query("insert into test values(2,'number2');")
        connection.query("insert into test values(3,'number3');")
    except pg.InternalError as e:
        print(e)
    else:
        print("Insert data successfully")


def update_data(connection):
    print("Begin to update data")
    try:
        result = connection.query("update test set name = 'numberupdated' where id=1;")
        print("Total number of rows updated :", result)
        result = connection.query("select * from test order by 1;")
        rows = result.getresult()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Update, Operation done successfully")


def delete_data(connection):
    print("Begin to delete data")
    try:
        result = connection.query("delete from test where id=3;")
        print("Total number of rows deleted :", result)
        result = connection.query("select * from test order by 1;")
        rows = result.getresult()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Delete,Operation done successfully")


def select_data(connection):
    print("Begin to select data")
    try:
        result = connection.query("select * from test order by 1;")
        rows = result.getresult()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1])
    except pg.InternalError as e:
        print(e)
        print("select failed")
    else:
        print("Operation done successfully")


if __name__ == '__main__':
    try:
        conn = pg.DB(host='10.154.70.231',
                     port=8000,
                     dbname='gaussdb', # Database to be connected
                     user='dbadmin',
                     passwd='password')  # Database user password
    except pg.InternalError as ex:
        print(ex)
        print("Connect database failed")
    else:
        print("Opened database successfully")
        create_table(conn)
        insert_data(conn)
        select_data(conn)
        update_data(conn)
        delete_data(conn)
        conn.close()
```

Alternatively, use the dbapi interface.

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from __future__ import print_function

import pg
import pgdb


def create_table(connection):
    print("Begin to create table")
    try:
        cursor = connection.cursor()
        cursor.execute("drop table if exists test;"
                       "create table test(id int, name text);")
        connection.commit()
    except pg.InternalError as e:
        print(e)
    else:
        print("Table created successfully")
        cursor.close()


def insert_data(connection):
    print("Begin to insert data")
    try:
        cursor = connection.cursor()
        cursor.execute("insert into test values(1,'number1');")
        cursor.execute("insert into test values(2,'number2');")
        cursor.execute("insert into test values(3,'number3');")
        connection.commit()
    except pg.InternalError as e:
        print(e)
    else:
        print("Insert data successfully")
        cursor.close()


def update_data(connection):
    print("Begin to update data")
    try:
        cursor = connection.cursor()
        cursor.execute("update test set name = 'numberupdated' where id=1;")
        connection.commit()
        print("Total number of rows updated :", cursor.rowcount)
        cursor.execute("select * from test;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Update, Operation done successfully")


def delete_data(connection):
    print("Begin to delete data")
    try:
        cursor = connection.cursor()
        cursor.execute("delete from test where id=3;")
        connection.commit()
        print("Total number of rows deleted :", cursor.rowcount)
        cursor.execute("select * from test;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Delete,Operation done successfully")


def select_data(connection):
    print("Begin to select data")
    try:
        cursor = connection.cursor()
        cursor.execute("select * from test;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
        print("select failed")
    else:
        print("Operation done successfully")
        cursor.close()


if __name__ == '__main__':
    try:
        conn = pgdb.connect(host='10.154.70.231',
                            port='8000',
                            database='gaussdb', # Database to be connected
                            user='dbadmin',
                            password='password') # Database user password
    except pg.InternalError as ex:
        print(ex)
        print("Connect database failed")
    else:
        print("Opened database successfully")
        create_table(conn)
        insert_data(conn)
        select_data(conn)
        update_data(conn)
        delete_data(conn)
        conn.close()
```

Step 3 Change the public network address, cluster port number, database name, database username, and database password in the **python_dws.py** file based on the actual cluster information.

NOTE

The PyGreSQL API does not provide the connection retry capability. You need to implement the retry processing in the service code.

```
        conn = pgdb.connect(host='10.154.70.231',
                            port='8000',
                            database='gaussdb', # Database to be connected
                            user='dbadmin',
                            password='password') # Database user password
```

Step 4 Run the following command to connect to the cluster using the third-party function library PyGreSQL:

```
python python_dws.py
```

----End

Using the Third-Party Function Library PyGreSQL to Connect to a Cluster (Windows)

Step 1 In the Windows operating system, click the **Start** button, enter **cmd** in the search box, and click **cmd.exe** in the result list to open the command-line interface (CLI).

Step 2 In the CLI, run the following command to create the **python_dws.py** file:

```
type nul> python_dws.py
```

Copy and paste the following content to the **python_dws.py** file:

```
#!/usr/bin/env python3
# _*_ encoding:utf-8 _*_

from __future__ import print_function

import pg

```

```
def create_table(connection):
    print("Begin to create table")
    try:
        connection.query("drop table if exists test;"
                         "create table test(id int, name text);")
    except pg.InternalError as e:
        print(e)
    else:
        print("Table created successfully")


def insert_data(connection):
    print("Begin to insert data")
    try:
        connection.query("insert into test values(1,'number1');")
        connection.query("insert into test values(2,'number2');")
        connection.query("insert into test values(3,'number3');")
    except pg.InternalError as e:
        print(e)
    else:
        print("Insert data successfully")


def update_data(connection):
    print("Begin to update data")
    try:
        result = connection.query("update test set name = 'numberupdated' where id=1;")
        print("Total number of rows updated :", result)
        result = connection.query("select * from test order by 1;")
        rows = result.getresult()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Update, Operation done successfully")


def delete_data(connection):
    print("Begin to delete data")
    try:
        result = connection.query("delete from test where id=3;")
        print("Total number of rows deleted :", result)
        result = connection.query("select * from test order by 1;")
        rows = result.getresult()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Delete,Operation done successfully")


def select_data(connection):
    print("Begin to select data")
    try:
        result = connection.query("select * from test order by 1;")
        rows = result.getresult()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1])
    except pg.InternalError as e:
        print(e)
        print("select failed")
    else:
        print("Operation done successfully")
```

```


if __name__ == '__main__':
    try:
        conn = pg.DB(host='10.154.70.231',
                     port=8000,
                     dbname='gaussdb', # Database to be connected
                     user='dbadmin',
                     passwd='password')  # Database user password
    except pg.InternalError as ex:
        print(ex)
        print("Connect database failed")
    else:
        print("Opened database successfully")
        create_table(conn)
        insert_data(conn)
        select_data(conn)
        update_data(conn)
        delete_data(conn)
        conn.close()
```

Alternatively, use the dbapi interface.

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

from __future__ import print_function

import pg
import pgdb


def create_table(connection):
    print("Begin to create table")
    try:
        cursor = connection.cursor()
        cursor.execute("drop table if exists test;"
                       "create table test(id int, name text);")
        connection.commit()
    except pg.InternalError as e:
        print(e)
    else:
        print("Table created successfully")
        cursor.close()


def insert_data(connection):
    print("Begin to insert data")
    try:
        cursor = connection.cursor()
        cursor.execute("insert into test values(1,'number1');")
        cursor.execute("insert into test values(2,'number2');")
        cursor.execute("insert into test values(3,'number3');")
        connection.commit()
    except pg.InternalError as e:
        print(e)
    else:
        print("Insert data successfully")
        cursor.close()


def update_data(connection):
    print("Begin to update data")
    try:
        cursor = connection.cursor()
        cursor.execute("update test set name = 'numberupdated' where id=1;")
        connection.commit()
        print("Total number of rows updated :", cursor.rowcount)
        cursor.execute("select * from test;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Update, Operation done successfully")


def delete_data(connection):
    print("Begin to delete data")
    try:
        cursor = connection.cursor()
        cursor.execute("delete from test where id=3;")
        connection.commit()
        print("Total number of rows deleted :", cursor.rowcount)
        cursor.execute("select * from test;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
    else:
        print("After Delete,Operation done successfully")


def select_data(connection):
    print("Begin to select data")
    try:
        cursor = connection.cursor()
        cursor.execute("select * from test;")
        rows = cursor.fetchall()
        for row in rows:
            print("id = ", row[0])
            print("name = ", row[1], "\n")
    except pg.InternalError as e:
        print(e)
        print("select failed")
    else:
        print("Operation done successfully")
        cursor.close()


if __name__ == '__main__':
    try:
        conn = pgdb.connect(host='10.154.70.231',
                            port='8000',
                            database='gaussdb', # Database to be connected
                            user='dbadmin',
                            password='password') # Database user password
    except pg.InternalError as ex:
        print(ex)
        print("Connect database failed")
    else:
        print("Opened database successfully")
        create_table(conn)
        insert_data(conn)
        select_data(conn)
        update_data(conn)
        delete_data(conn)
        conn.close()
```

Step 3 Change the public network address, cluster port number, database name, database username, and database password in the **python_dws.py** file based on the actual cluster information.

The PyGreSQL API does not provide the connection retry capability. You need to implement the retry processing in the service code.

```
    conn = pgdb.connect(host='10.154.70.231',
                        port='8000',
                        database='gaussdb', # Database to be connected
                        user='dbadmin',
                        password='password') # Database user password
```

Step 4 Run the following command to connect to the cluster using the third-party function library PyGreSQL:

```
python python_dws.py
```

----End

5 Creating a GaussDB(DWS) Database and User

The default database **gaussdb** of GaussDB(DWS) is not used as the customer's service database. You can use multiple databases to ensure service isolation. When you first connect to **gaussdb** as the system administrator (**dbadmin**), it is important to plan the service databases, users, and roles based on the service requirements. This involves creating a service and transferring any existing upstream service data to GaussDB(DWS).

A role is a set of permissions. For details about the relationship between users and roles, see **Permissions Management** in the *Developer Guide*. You can create common roles, such as a role for database creation, before creating a user. Then, you can assign the created role to the user.

Users, roles, and permissions can be exported. For details, see **Exporting a User**, **Exporting User Permissions**, **Exporting Roles**, and **Exporting Role Permissions**.

Constraints and Limitations

- Avoid having all business operations run under a single database user. Instead, plan different database users according to the business modules.
- For better access control of different business modules, it is better to use multiple users and permissions instead of depending on the system administrator user to run business operations.
- For more information about development and design specifications, see **Development and Design Proposal**.

Creating a Database

You can use the DDL syntax or SQL editor to create a table.

- DDL syntax: For details about the syntax, see "CREATE DATABASE".
- SQL editor: For details, see **Using the SQL Editor to Connect to a GaussDB(DWS) Cluster**.

Creating a Role

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 Click **Roles** and click **Create Role**. The role creation page is displayed.

Step 5 Configure role information. The parameters are described as follows:

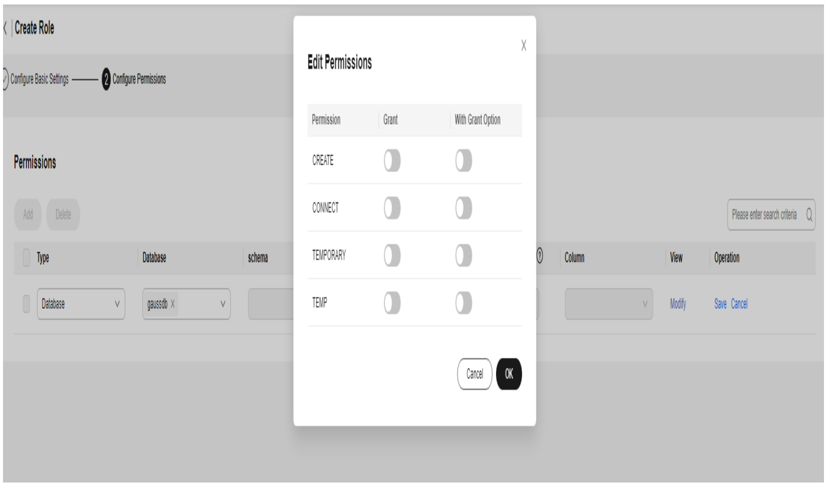
Table 5-1 Parameters for configuring role information

Parameter	Description	Example Value
Role Name	The value must start with a letter and can contain a maximum of 63 characters, including letters, digits, and underscores (_).	DWS-demo
Expires	Expiration time of the role permissions.	-
System Administrator	Whether the role has the system administrator rights.	-
Create Database	Whether the role has the permission to create databases.	-
Create Role	Whether the role has the permission to create users and roles.	-
Inherit Permissions	Whether the role inherits the permissions from its role group. By default, this function is enabled and it is best to keep it that way.	-

Step 6 Confirm the settings and click **Next**.

Step 7 Configure the permissions of the role.

Click **Add** to add a permission configuration. Select the database object type and the corresponding objects. Then, select permissions. For details about permission definitions, see "DCL Syntax" > "GRANT" in **GaussDB(DWS) SQL Overview**.



Step 8 After the authorization is complete, click **Create**.

----End

Creating a Database User

You can use the DDL syntax or create a table on the GaussDB(DWS) console. For details about the DDL syntax, see "CREATE USER".

NOTE

- If the current console does not support this feature, contact technical support.
- After a cluster is created, the users or roles created with it cannot be modified.
- Before using this function, ensure that the cluster is available.

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 On the **Users** page, click **Create User**.

Step 5 Set the parameters on the **Configure Basic Settings** page.

Table 5-2 Parameters on the Configure Basic Settings page

Parameter	Description	Example Value
Username	The value must start with a letter and can contain a maximum of 63 characters, including letters, digits, and underscores (_).	DWS-demo

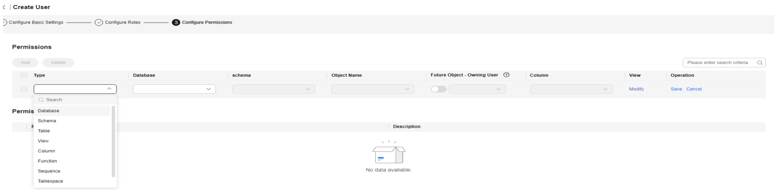
Parameter	Description	Example Value
Password	Enter a value that is 12 to 32 characters long and can contain letters, digits, underscores (_), and special characters. **NOTE** Contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-)	-
Maximum Connections	Maximum number of connections between the user and the database. The value **–1** indicates that the number of connections is not limited.	–1
Expires	Expiration time of the user's permissions.	-
System Administrator	Whether the user is a system administrator.	-
Create Database	Whether the user has the permission to create databases.	-
Create Role	Whether the user has the permission to create users and roles.	-
Inherit Permissions	Whether the user inherits permissions from its user group. By default, this function is enabled and it is best to keep it that way.	-

Step 6 Confirm the settings and click **Next**.

Step 7 On the **Configure Roles** page, select the role to be assigned to the user and click **Next**.

Step 8 Configure permissions not included in the roles of the user.

Click **Add** to add a permission configuration. Select the database object type and corresponding database object, and select the permission to complete assignment. For details about permission definitions, see "DCL Syntax" > "GRANT" in **GaussDB(DWS) SQL Overview**.



Step 9 After the authorization is complete, click **Create**.

----End

Modifying a User

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

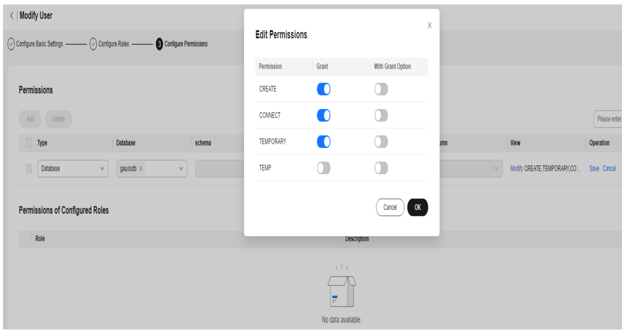
Step 3 In the navigation pane, choose **User Management**.

Step 4 In the user list, select a user and click **Modify**. The page for modifying user details is displayed.

Step 5 Modify the user information. For details, see Table 5-2. After confirming that the information is correct, click **Next**.

Step 6 Select the role you want to grant to the user and click **Next**.

Step 7 After selecting a permission type, you can click **Edit** in the **Operation** column and click **Modify** in the **Permission** column to add or remove a permission.



Modify User

Configure Basic Settings — Configure Roles — 3 Configure Permissions

Permissions

Add Delete

Type	Database	schema	...umn	View	Operation
Database	gaussdb ×			Modify CREATE,TEMPORARY,CO...	Save Cancel

Permissions of Configured Roles

Role	Description

No data available.

Edit Permissions

Permission	Grant	With Grant Option
CREATE	●	○
CONNECT	●	○
TEMPORARY	●	○
TEMP	○	○

Step 8 Confirm the permissions. Click **Save**.

----End

Deleting a User

Prerequisites

To prevent any problems with deleting a user, check for dependencies between database objects (such as tables) beforehand. If there are any dependencies, delete them first before proceeding with the user deletion.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 Select a user from the user list and click **Delete**. A confirmation dialog box is displayed.

Step 5 Click **OK**.

----End

Exporting a User

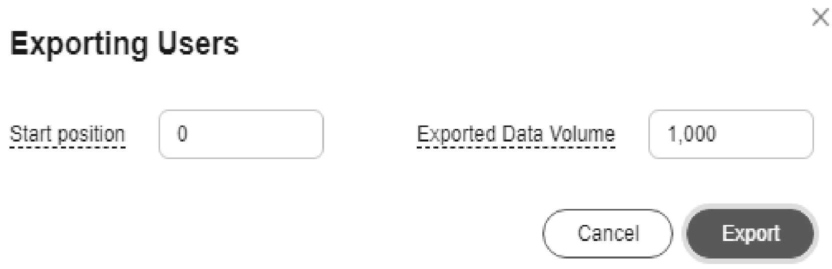
Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 Click **Export** in the upper part of the user list and select the number of records to be exported to export the user list.

Figure 5-1 Exporting a user



Exporting Users

Start position: 0

Exported Data Volume: 1,000

Step 5 Confirm the configurations and click **Export**.

----End

Exporting User Permissions

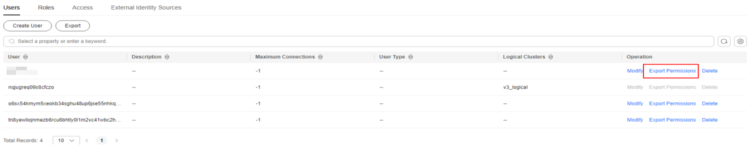
Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 Select a user from the user list and click **Export Permissions** to export the user permission list.

Figure 5-2 Exporting permissions



Users
Roles
Access
External Identity Sources
Create User
Export
Select a property or enter a keyword.

User	Description	Maximum Connections	User Type	Logical Clusters	Operation
[redacted]	--	-1	--	--	Modify Export Permissions Delete
nqjugreq09s8cfczo	--	-1	--	v3_logical	Modify Export Permissions Delete
o6sx54kmym5xeokb34sghu48up6jse55nhkq...	--	-1	--	--	Modify Export Permissions Delete
tn8yewliojnmezb6rcu6bhtly0l1m2vc41wbc2h...	--	-1	--	--	Modify Export Permissions Delete

----End

Modifying a Role

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 In the role list, select a user and click **Modify**. The page for modifying role details is displayed.

Step 5 Modify the role information. For the parameter description, see **Table 5-1**.

Step 6 Confirm the settings and click **Next**.

Step 7 Configure permissions. Select a permission type as required, click **Edit** in the **Operation** column, and click **Modify** in the **Permission** column to add or remove permissions.

Step 8 Confirm the permissions. Click **Save**.

----End

Deleting a Role

Prerequisites

To prevent any problems with deleting a role, check for dependencies such as database objects beforehand. If there are any dependencies, delete them first before proceeding with the role deletion.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

Step 4 Select a role from the role list and click **Delete**. A confirmation dialog box is displayed.

Step 5 Click **OK** to delete the role.

----End

Exporting Roles

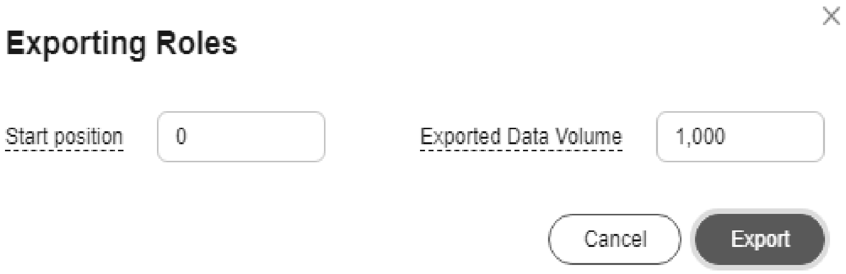
Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management** and click **Roles** to switch to the role list page.

Step 4 Click **Export** in the upper part of the role list and select the number of roles to be exported.

Figure 5-3 Exporting roles



Start position: 0

Exported Data Volume: 1,000

Step 5 Confirm the configurations and click **Export**.

----End

Exporting Role Permissions

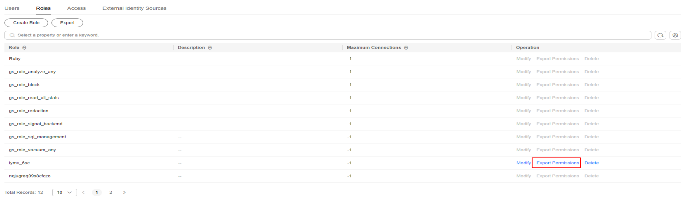
Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Dedicated Clusters** > **Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management** and click **Roles** to switch to the role list page.

Step 4 Select a user from the role list, click **Export Permissions**, and select the number of records to be exported.

Figure 5-4 Exporting role permissions


Users
Roles
Access
External Identity Sources
Create Role
Export
Select a property or enter a keyword.

Role	Description	Maximum Connections	Operation
Ruby	--	-1	Modify Export Permissions Delete
gs_role_analyze_any	--	-1	Modify Export Permissions Delete
gs_role_block	--	-1	Modify Export Permissions Delete
gs_role_read_all_stats	--	-1	Modify Export Permissions Delete
gs_role_redaction	--	-1	Modify Export Permissions Delete
gs_role_signal_backend	--	-1	Modify Export Permissions Delete
gs_role_sql_management	--	-1	Modify Export Permissions Delete
gs_role_vacuum_any	--	-1	Modify Export Permissions Delete
iymx_6sc	--	-1	Modify Export Permissions Delete
nqjugreq09s8cfczo	--	-1	Modify Export Permissions Delete

Total Records: 12 10 < 1 2 >

----End

6 Migrating Service Data to a GaussDB(DWS) Cluster

6.1 Migrating Data to a GaussDB(DWS) Cluster Using GDS-Kafka

6.1.1 Overview

GaussDB(DWS) helps you migrate data from multiple sources and integrate diverse data sources, quick and easy. Currently, data can be migrated from Kafka, MySQL, Oracle, and IoT to GaussDB(DWS).

NOTE

- This feature is supported only in 8.2.0 or later.
- Data cannot be migrated from one GaussDB(DWS) database to another due to the limitations of the data source being used.

Supported Data Sources

Source Data Source	Destination Data Source	Description
Kafka	GaussDB(DWS)	-
MySQL	GaussDB(DWS)	-
Oracle	GaussDB(DWS)	-
IOT	GaussDB(DWS)	-

6.1.2 Managing Instances

Overview

Data migration provides independent clusters for secure and reliable data migration. Clusters are isolated from each other and cannot access each other. With instance management, you can easily create and manage clusters by purchasing GDS-Kafka instances. GDS-Kafka consumes and caches data from Kafka. If the data cache time or size reaches a preconfigured threshold, GDS-Kafka will copy the data to a GaussDB(DWS) temporary table, and then insert or update data in the temporary table.

- The format of data generated by the Kafka message producer is specified by the **kafka.source.event.type** parameter. For details, see **Message Formats Supported by GDS-Kafka**.
- In GDS-Kafka, you can directly insert data for tables without primary keys, or update data by merging. Direct insert can achieve better performance, because it does not involve update operations. Determine your update mode based on the target table type in GaussDB(DWS). The data import mode is determined by the **app.insert.directly** parameter and whether a primary key exists. For details, see **GDS-Kafka Data Import Modes**.

NOTE

- GDS-kafka only allows lowercase target table and column names.
- GDS-Kafka deletes historical data based on **pos** in the extended field. If imported data involves the delete operation, the extended field must be used.

Purchasing a GDS-Kafka Instance

To use the data migration feature, you need to purchase a GDS-kafka instance (cluster). Cluster instances provide secure and reliable data migration services. Clusters are isolated from each other.

Constraints

- Currently, only standalone clusters are supported.
- Only the pay-per-use billing mode is supported.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 In the upper right corner of the page, click **Buy GDS-Kafka Instance**. Configure cluster parameters.

Table 6-1 Parameter description

Parameter	Description	Example Value
CPU Architecture	The following CPU architectures can be selected: ● **x86** ● **Kunpeng** **NOTE** The x86 and Kunpeng architectures differ only in their underlying structure, which is not sensible to the application layer. Both architectures use the same SQL syntax. If you need to create a cluster and find that x86 servers are not enough, you can opt for the Kunpeng architecture.	x86
Flavor	Select a node flavor.	-
Capacity	Storage capacity of a node.	-
Current Flavor	Current flavor of the cluster.	-
Name	Set the name of the data warehouse cluster. Enter 4 to 64 characters. Only case-insensitive letters, digits, hyphens (-), and underscores (_) are allowed. The value must start with a letter. Letters are not case-sensitive.	-
Version	Version of the database instance installed in the cluster.	-
VPC	Specify a VPC to isolate the cluster's network. If you create a data warehouse cluster for the first time and have not configured the VPC, click **View VPC**. On the VPC management console that is displayed, create a VPC as needed.	-
Subnet	Specify a VPC subnet. A subnet provides dedicated network resources that are isolated from other networks, improving network security.	-
Security Group	Specify a VPC security group. A security group restricts access rules to enhance security when GaussDB(DWS) and other services access each other.	-

Parameter	Description	Example Value
EIP	Specify whether users can use a client to connect to a cluster's database over the Internet. The following methods are supported: ● **Do not use**: Do not specify any EIPs here. If GaussDB(DWS) is used in the production environment, first bind it to ELB, and then bind it to an EIP on the ELB page. ● **Buy now**: Specify bandwidth for EIPs, and the system will automatically assign EIPs with dedicated bandwidth to clusters. You can use the EIPs to access the clusters over the Internet. The bandwidth name of an automatically assigned EIP starts with the cluster name. ● **Specify**: Specify an EIP to be bound to the cluster. If no available EIPs are displayed in the drop-down list, click **Create EIP** to go to the **Elastic IP** page and create an EIP as needed. The bandwidth can be customized.	-
Enterprise Project	Select the enterprise project of the cluster. You can configure this parameter only when the Enterprise Project Management service is enabled. The default value is **default**.	default

Step 4 If the configuration is correct, click **Buy Now**.

----End

Viewing Instance Details

On the instance details page, you can view the basic information and network information about the cluster.

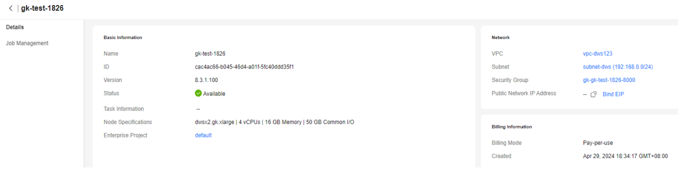
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 Click the name of an instance to go to the instance details page.

Figure 6-1 Viewing Instance Details



Basic Information

Name gk-test-1826

ID cac4ac66-b045-46d4-a01f-5fc40ddd35f1

Version 8.3.1.100

Status Available

Task Information --

Node Specifications dwsx2.gk.xlarge | 4 vCPUs | 16 GB Memory | 50 GB Common I/O

Enterprise Project default

Network

VPC vpc-dws123

Subnet subnet-dws (192.168.0.0/24)

Security Group gk-gk-test-1826-8000

Public Network IP Address -- Bind EIP

Billing Information

Billing Mode Pay-per-use

Created Apr 29, 2024 18:34:17 GMT+08:00

----End

Message Formats Supported by GDS-Kafka

Table 6-2 Message formats supported by GDS-Kafka

Kafka Source Event Type	Format	Description
cdc.drs.avro	Internal format of Huawei Cloud DRS. DRS generates data in the avro format used by Kafka. GDS-Kafka can directly interconnect with DRS to parse and import the data.	None

Kafka Source Event Type	Format	Description

drs.cdc	To use the avro format for **drs.cdc**, specify the Maven dependency of GDS-Kafka-common and GDS-Kafka-source in the upstream programs of Kafka, and then create and fill in the **Record** object. A **Record** object represents a table record. It will be serialized into a **byte[]** array, produced and sent to Kafka, and used by the downstream GDS-Kafka.
	In the following example, the target table is the **person** table in the **public** schema. The **person** table consists of the **id**, **name**, and **age** fields. The **op_type** is **U**, which indicates an update operation. This example changes the **name** field from **a** to **b** in the record with the ID **0**, and changes the value of the **age** field from **18** to **20**.
	`Record record = new Record();`
	`// Set the schema and table name of the target table.`
	`record.setTableName("public.person");`
	`// Set the field list.`
	`List<Field> fields = new ArrayList<>();`
	`fields.add(new Field("id", 0));`
	`fields.add(new Field("name", 1));`
	`fields.add(new Field("age", 2));`
	`record.setFields(fields);`
	`// Set the field value list before the table record is updated.`
	`List<Object> before = new ArrayList<>();`
	`before.add(new Integer(0, "0"));`
	`before.add(new Character("utf-8",`
	`ByteBuffer.wrap("a".getBytes(StandardCharsets.UTF_8))));`
	`before.add(new Integer(0, "18"));`
	`record.setBeforeImages(before);`
	`// Set the field value list after the table record is updated.`
	`List<Object> after = new ArrayList<>();`
	`after.add(new Integer(0, "0"));`
	`after.add(new Character("utf-8",`
	`ByteBuffer.wrap("b".getBytes(StandardCharsets.UTF_8))));`
	`after.add(new Integer(0, "20"));`
	`record.setAfterImages(after);`
	`// Set the operation type.`
	`record.setOperation("U");`
	`// Set the operation time.`
	`record.setUpdateTimestamp(325943905);`
	`// Serialize the record object into a byte[] array.`
	`byte[] msg =`
	`Record.getEncoder().encode(record).array();`
		Standard avro format:
		● The **tableName** field is used to describe the target table and schema names that the current record belongs to. [Mandatory]
		● The **operation** field is used to describe the operation type of the current record. **I** indicates insert, **U** indicates update, and **D** indicates deletion. [Mandatory]
		● **updateTimestamp** indicates the time when an operation is performed on the source end. [Optional]
		● The **beforeImages** list describes the information before the current record is updated or deleted. The fields in the **before body** correspond to those in the target table. [Mandatory for U/D]
		● The **afterImages** list describes the updated or

Kafka Source Event Type	Format	Description
		newly inserted information of the current record. [Mandatory for U/D] ● The **fields** list describes the field list of the current table record. The index **values** of the fields must be in the same sequence as those in **beforeImage** and **afterImage**. [Mandatory]

Kafka Source Event Type	Format	Description

cdc.json	In the following example, the target table is the **person** table in the **public** schema. The **person** table consists of the **id**, **name**, and **age** fields. The **op_type** is **U**, which indicates an update operation. This example changes the **name** field from **a** to **b** in the record with the ID **1**, and changes the value of the **age** field from **18** to **20**.
	`{`
	`"table": "public.person",`
	`"op_type": "U",`
	`"op_ts": "1668426344",`
	`"current_ts": "1668426344",`
	`"before": {`
	`"id":"1",`
	`"name":"a",`
	`"age": 18`
	`},`
	`"after": {`
	`"id":"1",`
	`"name":"b",`
	`"age": 20`
	`}`
	`}`
		Standard JSON format:
		● The **table** field describes the target table and schema names that the current record belongs to. [Mandatory]
		● The **op_type** field is used to describe the operation type of the current record. **I** indicates insert, **U** indicates update, and **D** indicates deletion. [Mandatory]
		● **op_ts** indicates the time when an operation is performed on the source end. [Optional]
		● **current_ts** indicates the time when a message is imported to Kafka. [Optional]
		● The **before** object describes the information before the current record is updated or deleted. The fields in the **before body** correspond to those in the target table. [Mandatory for U/D]

Kafka Source Event Type	Format	Description
		● The **after** object list describes the update or newly inserted information of the current record. [Mandatory for U/D]

industrial.iot.json	`{`
	`"header": {`
	`"thing_id":"a0001",`
	`"instance_id":"1",`
	`"thing_model_name":"computer",`
	`"timestamp":"1668426344"`
	`},`
	`"body": {`
	`"status":"Normal",`
	`"temperature":"10",`
	`"working_time":"10000"`
	`},`
	`}`
		IoT data format:
		● **thing_model_name** in **header** indicates the table name. [Mandatory]
		● The values of **thing_id**, **instance_id**, and **timestamp** in **header** and the content in the body comprise the fields of the current record.
		● IoT data is time series data and does not involve update or deletion. Only insert operations are involved.

Kafka Source Event Type	Format	Description

industrial.iot.recursion.json	{
	"header": {
	"thing_id":"a0001",
	"instance_id":"1",
	"thing_model_name":"computer",
	"timestamp":"1668426344"
	},
	"body": {
	"status":"Normal",
	"temperature":"10",
	"property":{
	  "key1":"1",
	  "key2":2
	},
	"working_time":"10000"
	},
	}
		IoT data format:
		● **thing_model_name** in **header** indicates the table name. [Mandatory]
		● The values of **thing_id**, **instance_id**, and **timestamp** in **header** and the content in the body comprise the fields of the current record.
		● IoT data is time series data and does not involve update or deletion. Only insert operations are involved.
		● In this data format, the key and value of **body** are added to the **property** and **value** fields in the new format to generate multiple pieces of new data. In this way, rows are converted to columns.

Kafka Source Event Type	Format	Description

industrial.iot.event.json.independent.table	{
	"event_id":"1",
	"event_name":"test",
	"start_time":"1970-1-1T00:00:00.000Z",
	"end_time":"1970-1-1T00:00:00.000Z",
	"fields":{
	    "field1":"value1",
	    "field2":2
	    }
	}
		IoT event stream data format:
		● **event_name** indicates a table name. [Mandatory]
		● **event_id**, **start_time**, **end_time**, and **fields** comprise the field content of a record. [Mandatory]
		● IoT event stream data is time series data and does not involve update or deletion. Only insert operations are involved.

Kafka Source Event Type	Format	Description

industrial.iot.json.multi.events	{
	"event_id":"1",
	"event_name":"test",
	"start_time":"1970-1-1T00:00:00.000Z",
	"end_time":"1970-1-1T00:00:00.000Z",
	"fields":{
	    "field1":"value1",
	    "field2":2,
	    "field3":{
	        "key1":"1",
	        "key2":2
	        }
	    }
	}
		IoT event stream data format:
		● **event_name** indicates a table name. [Mandatory]
		● **event_id**, **start_time**, **end_time**, and **fields** comprise the field content of a record. [Mandatory]
		● IoT event stream data is time series data and does not involve update or deletion. Only insert operations are involved.
		● In this data format, the key and value of **fields** are added to the **field_name** and **field_value** fields in the new format to generate multiple pieces of new data. In this way, rows are converted to columns.

GDS-Kafka Import Modes

To import GDS-Kafka data to the database, copy the data to a temporary table, and then merge or insert the data. The following table describes their usage and scenarios.

Table 6-3 GDS-Kafka import modes

Operation	Direct Insertion	Primary Key Table	Import Mode
insert	**true** (only for tables without primary keys)	No	Use **INSERT SELECT** to write data from the temporary table to the target table.
	false	Yes	Merge data from the temporary table to the target table based on the primary key.
		No	Use **INSERT SELECT** to write data from the temporary table to the target table.
delete	**true** (only for tables without primary keys)	No	Use **INSERT SELECT** to write data from the temporary table to the target table.
	false **NOTE** You can mark deletion by configuring the **app.del.flag** parameter. The flag of a deleted record will be set to **1**.	Yes	● If the **delflag** field is set, merge will be performed based on the primary key. If a matched primary key is found, and the value of **pos** in the target table is smaller than that in the temporary table, the **delflag** field will be set to 1. Otherwise, a new record will be inserted. ● If the **delflag** field is not set, a matched primary key is found, and the value of **pos** in the target table is smaller than that in the temporary table, the record will be deleted from the target table.

Operation	Direct Insertion	Primary Key Table	Import Mode
		No	● If the **delflag** field is set, all the fields in the temporary table will be used to match and merge with the target table. If a matched record is found, and the value of **pos** in the target table is smaller than that in the temporary table, the **delflag** field will be set to **1**. Otherwise, a new record will be inserted. ● If the **delflag** field is not set, all the fields in the temporary table will be used to match the target table. If a matched record is found, and the value of **pos** in the target table is smaller than that in the temporary table, the matched record will be deleted from the target table.
update	**true** (only for tables without primary keys)	No	Use **INSERT SELECT** to write data from the temporary table to the target table.
	false **NOTE** The update operation is split. The message in **before** or **beforeImage** is processed as a delete operation, and the message in **after** or **afterImage** is processed as an insert operation. Then, the message is saved to the database based on the insert and delete operations.	Yes	Equivalent to the insert+delete operation on a table with a primary key.
		No	Equivalent to the insert+delete operation on a table without a primary key.

6.1.3 Managing Connections

Description

Before creating a data migration task, you need to create a connection, so that the cluster can read and write the data source. A migration job requires a source connection and a destination connection. Data sources that support exporting are used as source connections and data sources that support importing are used as destination connections.

The connection parameters you can configure vary according to the data source. This section describes how to create these connections.

Prerequisites

- A GDS-kafka cluster has been created.
- The GDS-kafka cluster can communicate with the destination data source.
  - If the destination data source is an on-premises database, you need the Internet or Direct Connect. If the Internet is used for communication, ensure that an EIP has been bound to the GDS-kafka cluster, the security group of GDS-kafka allows outbound traffic from the host where the off-cloud data source is located, the host where the data source is located can access the Internet, and the connection port has been enabled in the firewall rules.
  - If the destination data source is a cloud service, the following requirements must be met for network interconnection:
    - If the GDS-kafka cluster and the cloud service are in different regions, the Internet or a Direct Connect is required for enabling communication between the CDM cluster and the cloud service. If the Internet is used for communication, ensure that an EIP has been bound to the GDS-kafka cluster, the host where the data source is located can access the Internet, and the port has been enabled in the firewall rules.
    - If the GDS-kafka cluster and the cloud service are in the same region, VPC, subnet, and security group, they can communicate with each other by default. If they are in the same VPC but in different subnets or security groups, you must configure routing rules and security group rules. For more information, see **Configuring Routes** and **Security Group Configuration**.
    - The cloud service instance and the cluster belong to the same enterprise project. If they do not, you can modify the enterprise project of the workspace.
- You have obtained the URL, account, and password for accessing the destination data source. The account is granted with the read and write permissions on the data source.

Creating a Connection

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Connections**.

Step 3 Click **Create Connection**.

Step 4 Configure connection parameters. For more information, see **Connection parameters**.

Table 6-4 Connection parameters

Protocol	Parameter	Mandatory	Description
Kafka	Connection Name	Yes	Connection name, which can be customized. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
	Type	Yes	Currently, MRS Kafka, IoT Kafka, DMS Kafka, and Default Kafka are supported. Default Kafka is an open-source Kafka.
	Service Address	Yes	Kafka connection address. Format: *Domain name* + *Port number or IP address* + *Port number*
	Topics	Yes	List of Kafka topics, which are separated by commas (,).
	Ciphertext Access	No	This function must be enabled during Kafka authentication. The SASL_SSL and SASL_PLAINTEXT protocols are supported.
	User	No	Username for connecting to Kafka
	Password	No	Password for connecting to Kafka.
	SSL Authentication	No	Whether the SSL protocol is supported.
	Certificate	No	SSL certificate in binary JKS format.
	Certificate Password	No	Certificate encryption password.

Protocol	Parameter	Mandatory	Description

	Host Configuration	No	MRS-Kafka configuration parameter. When you connect to MRS-Kafka in security mode, you need to configure the host file of the VM where Gds-Kafka resides. Therefore, you need to upload the host file to be modified. The file format can only be TXT. The file content is as follows:
			`192.168.4.111  node-master1JuQr.mrs-yd8z.com`
			`192.168.4.204  node-master3mgqy.mrs-yd8z.com`
			`192.168.4.221  node-master2Ktgg.mrs-yd8z.com`
			The information on the left is the IP address of the Kafka broker. If MRS-Kafka and GDS-Kafka are not in the same VPC, replace the IP address with a public IP address. The information on the right is the host name of the broker. You can log in to FusionInsightManage and access the Kafka cluster to obtain the host name corresponding to the broker instance.

	Security mode	No	MRS-Kafka configuration parameter. When the security mode is enabled, Kerberos authentication is required.
	Krb5 File	No	MRS-Kafka configuration parameter. When the security mode is enabled, you need to upload the krb5 file. This file is the authentication credential of the machine-machine account applied for on FusionInsight Manager of MRS. **NOTE** If MRS-Kafka and GDS-Kafka are not in the same VPC, replace the internal IP address of the broker in the file with the public IP address.

Protocol	Parameter	Mandatory	Description
	Keytab File	No	MRS-Kafka configuration parameter. When the security mode is enabled, you need to upload the Keytab file. This file is the authentication credential of the machine-machine account applied for on FusionInsight Manager of MRS.
	Account	No	MRS-Kafka configuration parameter. It is a machine-machine account applied for on FusionInsight Manager of MRS.
	SSL	No	MRS-Kafka configuration parameter. When SSL is enabled, you need to upload the SSL certificate and key.
	Authentication Mechanism	No	DMS-Kafka configuration parameter. It indicates the security authentication protocol.
MySQL	Connection Name	Yes	Connection name, which can be customized. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
	Service Address	Yes	MySQL connection address. Format: *Domain name* + *Port number or IP address* + *Port number*
	User	Yes	Username for logging in to the database.
	Password	Yes	Password used to log in to the database.
	Database	Yes	MySQL database name.
Oracle	Connection Name	Yes	Connection name, which can be customized. Only letters, numbers, underscores (_), and hyphens (-) are allowed.

Protocol	Parameter	Mandatory	Description
	Service Address	Yes	Oracle connection address. Format: *Domain name* + *Port number or IP address* + *Port number*
	User	Yes	Username for logging in to the database.
	Password	Yes	Password used to log in to the database.
	Database	Yes	Oracle database name.
	Schema	Yes	Schema name. You can configure one or more schema names and use commas (,) to separate them.
IoT	Service Address	Yes	Address of the iot-edge-node page. Format: domain name or IP address
	User	Yes	Account for logging in to the IoT platform.
	Password	Yes	Password for logging in to the IoT platform.
DWS	Connection Name	Yes	Connection name, which can be customized. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
	Service Address	Yes	GaussDB(DWS) connection address. Format: *Domain name* + *Port number or IP address* + *Port number*, for example, **192.168.0.10:8000**.
	User	Yes	Username for logging in to the database.
	Password	Yes	Password used to log in to the database.
	Database	Yes	GaussDB(DWS) database name.

Protocol	Parameter	Mandatory	Description
	Schema	Yes	Name of a schema in the GaussDB(DWS) database.

Step 5 Confirm the information and click **OK**.

----End

Modifying a Connection

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Connections**.

Step 3 In the **Operation** column of a connection, click **Modify**.

Step 4 In the dialog box for modifying connection configurations, modify the connection configuration based on the rules.

Step 5 Confirm the information and click **OK**.

----End

Deleting a Connection

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Connections**.

Step 3 In the **Operation** column of a connection, click **Delete**.

Step 4 In the displayed dialog box, click **OK**.

----End

6.1.4 Managing Table Mappings

Mapping Overview

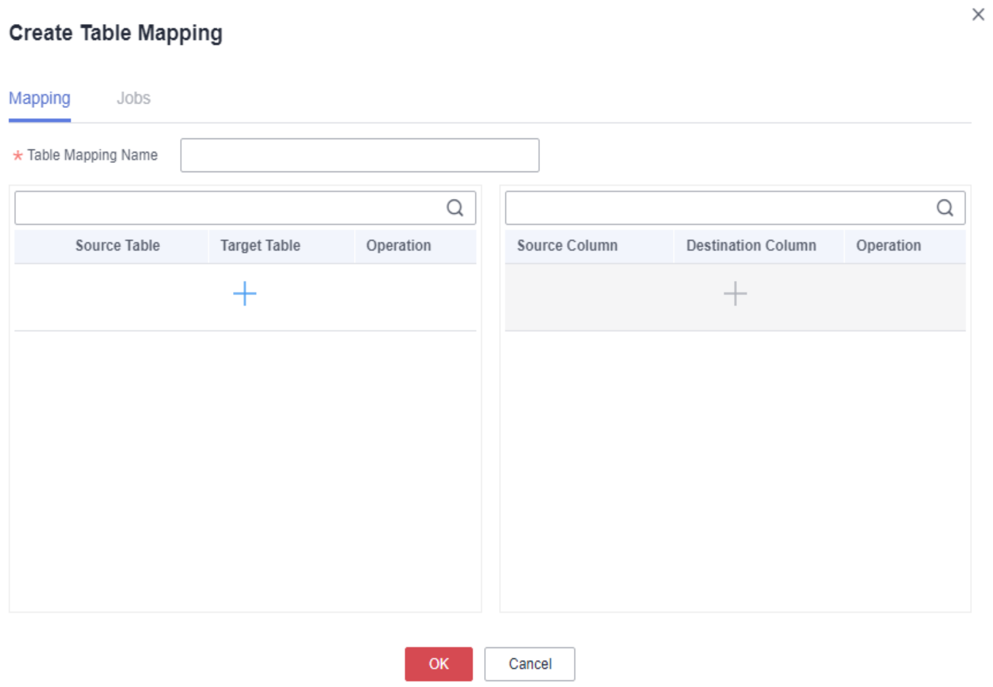
Before creating a job, you need to create a mapping to map the table structures of the source and destination databases, facilitating data migration between databases.

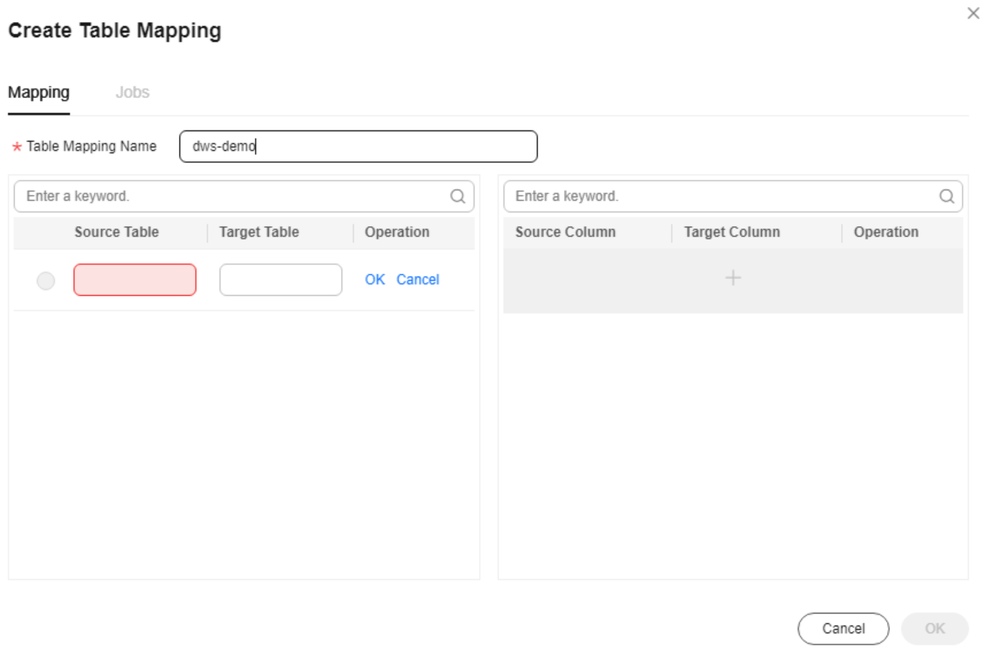
Creating a Table Mapping

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Table Mappings**.

Step 3 Click **Create Table Mapping**.





* Table Mapping Name: dws-demo

Step 4 Configure parameters.

1. Click + in the list on the left. Configure **Table Mapping Name**, **Source Table**, and **Target Table**.
2. Click + in the list on the right and configure the parameters.

NOTE

If no column mappings are specified in the list on the right, all the columns with the same name will be mapped with by default.

Step 5 Confirm the information and click **OK**.

----End

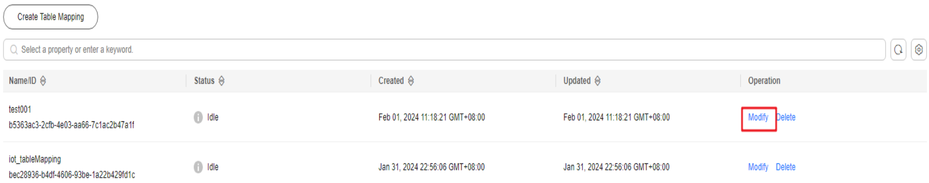
Modifying a Table Mapping

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Table Mappings**.

Step 3 In the **Operation** column of a table mapping, click **Modify**.

Figure 6-2 Modifying table mapping configurations



Name/ID	Status	Created	Updated	Operation
test001 b5363ac3-2cfb-4e03-aa66-7c1ac2b47a1f	Idle	Feb 01, 2024 11:18:21 GMT+08:00	Feb 01, 2024 11:18:21 GMT+08:00	Modify Delete
iot_tableMapping bec28936-b4df-4606-93be-1a22b429fd1c	Idle	Jan 31, 2024 22:56:06 GMT+08:00	Jan 31, 2024 22:56:06 GMT+08:00	Modify Delete

Step 4 In the dialog box for modifying table mapping configurations, modify the table mapping configuration based on the rules.

Step 5 Confirm the information and click **OK**.

----End

Checking a Table Mapping

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Table Mappings**.

Step 3 In the **Operation** column of a table mapping, click **Modify**.

Step 4 In the **Modify Table Mapping** dialog box, click **Jobs** to view the bound jobs.

----End

Deleting a Table Mapping

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Table Mappings**.

Step 3 In the **Operation** column of a table mapping, click **Delete**.

Step 4 In the displayed dialog box, click **OK**.

----End

6.1.5 Managing Jobs

After creating a cluster instance, you can customize a job, enable a job, and migrate data.

You can create jobs to migrate data or automatically create tables.

- Data migration: Data is migrated from Kafka to GaussDB(DWS).
- Automatic table creation: Tables and fields in the source database are synchronized to GaussDB(DWS), but data is not migrated.

Creating a Job

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane, click **Manage Job**.

Step 5 Click **Data Migration** or **Create Table**. (By default, the **Kafka Connection** parameter cannot be configured if you click **Create Table**.)

Step 6 Enter the job name, configure **Kafka Connection**, **DWS Cluster Connection**, and **Customized Table/Field Mapping**, and click **Test Connection**.

Step 7 Check to ensure the connection passes the test, and click **Next**.

Step 8 Click **Next** and confirm the settings.

Step 9 Click **OK**.

Step 10 Return to the job list. In the **Operation** column of the job, click **Start**. For details, see **Starting a Job**.

----End

Viewing Job Details

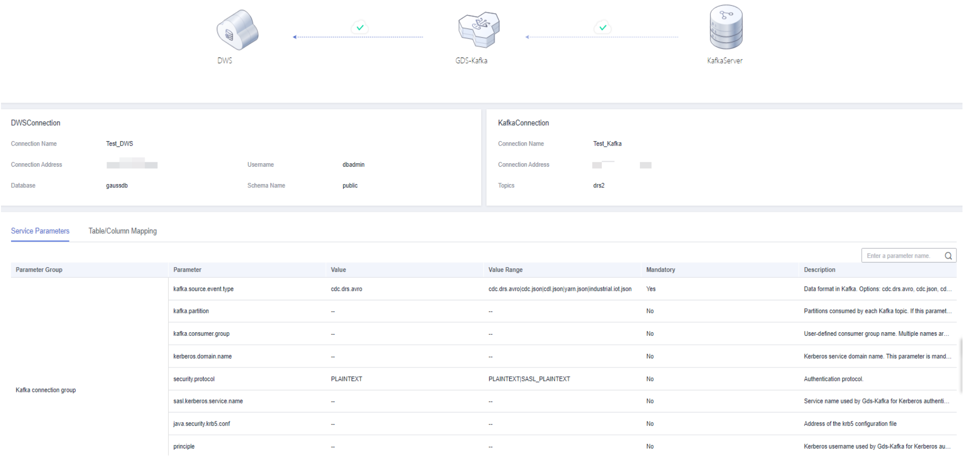
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane, click **Manage Job**.

Step 5 Click a job name to go to the details page. Check the job information, including the connections, service parameters, and table/column mappings.



DWSConnection

Connection Name Test_DWS

Connection Address [redacted] Username dbadmin

Database gaussdb Schema Name public

KafkaConnection

Connection Name Test_Kafka

Connection Address [redacted]

Topics drs2

Parameter Group	Parameter	Value	Value Range	Mandatory	Description
Kafka connection group	kafka.source.event.type	cdc.drs.avro	cdc.drs.avro|cdc.json|cdl.json|yarn.json|industrial.iot.json	Yes	Data format in Kafka. Options: cdc.drs.avro, cdc.json, cd...
	kafka.partition	--	--	No	Partitions consumed by each Kafka topic. If this paramet...
	kafka.consumer.group	--	--	No	User-defined consumer group name. Multiple names ar...
	kerberos.domain.name	--	--	No	Kerberos service domain name. This parameter is mand...
	security.protocol	PLAINTEXT	PLAINTEXT|SASL_PLAINTEXT	No	Authentication protocol.
	sasl.kerberos.service.name	--	--	No	Service name used by Gds-Kafka for Kerberos authenti...
	java.security.krb5.conf	--	--	No	Address of the krb5 configuration file
	principle	--	--	No	Kerberos username used by Gds-Kafka for Kerberos au...

----End

Starting a Job

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane, click **Manage Job**.

Step 5 In the **Operation** column of a job, click **Start**.

Step 6 In the displayed dialog box, click **OK** to start the job.

----End

Stopping a Job

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane, click **Manage Job**.

Step 5 In the **Operation** column of a job, click **Stop**.

Step 6 In the displayed dialog box, click **OK** to stop the job.

----End

Deleting a Job

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Data** > **Data Integration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane, click **Manage Job**.

Step 5 In the **Operation** column of a job, click **Delete**.

Step 6 Click **OK**.

----End

6.2 Data Source Management

6.2.1 MRS Data Sources

6.2.1.1 MRS Data Source Usage Overview

MRS Cluster Overview

MRS is a big data cluster running based on the open-source Hadoop ecosystem. It provides the industry's latest cutting-edge storage and analysis capabilities of massive volumes of data, satisfying your data storage and processing requirements. For details about MRS, see the ***MapReduce Service User Guide***.

You can use Hive/Spark (analysis cluster of MRS) to store massive volumes of service data. Hive/Spark data files are stored in HDFS. On GaussDB(DWS), you can connect a data warehouse cluster to MRS clusters, read data from HDFS files, and write the data to GaussDB(DWS) when the clusters are on the same network.

NOTE

Currently, storage-compute coupled data warehouses (standalone mode) cannot import data from MRS.

Operation Process

Perform the following operations to import data from MRS to a data warehouse cluster:

1. Prerequisites
   a. Create an MRS cluster. For details, see **Buying a Custom Cluster**.
   b. Create an HDFS foreign table for querying data from the MRS cluster over APIs of a foreign server.

      For details, see **Importing Data from MRS to a Data Warehouse Cluster** in *Data Warehouse Service (DWS) Data Migration and Synchronization*.

      NOTE

      - Multiple MRS data sources can exist on the same network, but one GaussDB(DWS) cluster can connect to only one MRS cluster at a time.
2. In the data warehouse cluster, create an MRS data source connection according to **Creating an MRS Data Source Connection**.
3. Import data from an MRS data source to the cluster. For details, see **Importing Data from MRS to a Cluster**.

4. (Optional) When the HDFS configuration of the MRS cluster changes, update the MRS data source configuration on GaussDB(DWS). For details, see **Updating the MRS Data Source Configuration**.

6.2.1.2 Creating an MRS Data Source Connection

Scenario

Before GaussDB(DWS) reads data from MRS HDFS, you need to create an MRS data source connection that functions as a channel of transporting data warehouse cluster data and MRS cluster data.

Impact on the System

- You can create only one MRS data source connection in the data warehouse cluster at a time.
- When an MRS data source connection is being created, the system automatically adds inbound and outbound rules to security groups of the data warehouse cluster and MRS cluster. Nodes in the same subnet can be accessed.
- For the MRS cluster with Kerberos authentication enabled, the system automatically adds a **Machine-Machine** user that belongs to user group **supergroup** to the MRS cluster.

Prerequisites

- You have created a data warehouse cluster and recorded the VPC and subnet where the cluster resides.
- An MRS cluster of the analysis type has been created.

Procedure

Step 1 Log in to the Huawei Cloud console.

Step 2 Go to the MRS console and create an MRS cluster.

Configure parameters as required. For details, see "Cluster Operation Guide > Custom Creation of a Cluster" in the *MapReduce Service User Guide*.

- The VPC of the MRS cluster must be the same as that of the data warehouse cluster.
- Select an MRS cluster version. The following versions are supported:
  - For clusters of version 8.1.1.300 and later, MRS clusters support versions 1.6.*, 1.7.*, 1.8.*, 1.9.*, 2.0.*, 3.0.*, 3.1.*, 3.2.*, 3.3.*, and later (*indicates a number).
  - For clusters earlier than version 8.1.1.300, MRS clusters support versions 1.6.*, 1.7.*, 1.8.*, 1.9.*, and 2.0.* (*indicates a number).
- Select the Hadoop component.

If you already have a qualified MRS cluster, skip this step.

Step 3 Go to the GaussDB(DWS) console.

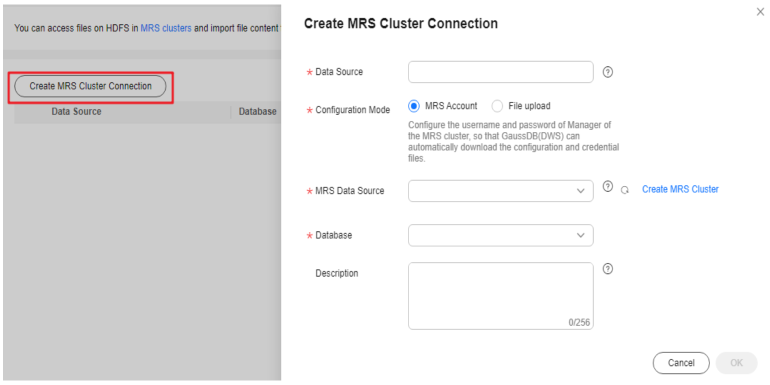
Step 4 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 5 In the cluster list, click the name of a cluster. The **Cluster Information** page is displayed.

Step 6 In the navigation tree on the left, choose **Data Sources** > **MRS Data Sources**.

Step 7 Click **Create MRS Cluster Connection** and configure parameters.

Figure 6-3 Creating an MRS Data Source Connection



You can access files on HDFS in MRS clusters and import file content

Create MRS Cluster Connection

Configure the username and password of Manager of the MRS cluster, so that GaussDB(DWS) can automatically download the configuration and credential files.

Table 6-5 MRS common connection parameters

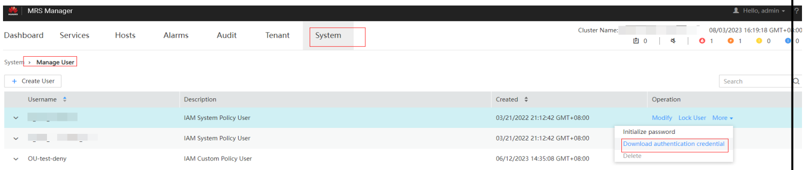
Parameter	Description
Data Source	GaussDB(DWS) database server name. It can contain 3 to 63 characters, including lowercase letters, numbers, and underscores (_), and must start with a lowercase letter.
Configuration Mode	The way in which the system obtains files. The options are as follows: ● **MRS Account**: Configure the username and password of the Manager of the MRS cluster. The system will log in to the Manager and automatically download configuration and verification files. For more information, see **Table 6-6**. ● **File upload**: Download the configuration file from the Manager of the MRS cluster and manually upload it. You can use this method for Kerberos authentication. For more information, see **Table 6-7**. **NOTE** – If you select **File upload**, ensure that MRS can communicate with the GaussDB(DWS) cluster.
Database	Database where the data source is located.
Description	Description of the connection.

Table 6-6 Parameters of the MRS Account mode

Parameter	Description
MRS Data Source	Select an MRS cluster that can be connected to GaussDB(DWS) from the drop-down list box. By default, the custom, hybrid, and analytical MRS clusters that are in the same VPC and subnet as the current GaussDB(DWS) cluster and available to the current user are displayed. After you select an MRS cluster, the system automatically displays whether Kerberos authentication is enabled for the selected cluster. Click **View MRS Cluster** to view its detailed information. If the **MRS Data Source** drop-down list is empty, click **Create MRS Cluster** to create an MRS cluster.
MRS Account	Account used when a GaussDB(DWS) cluster connects to an MRS cluster.
Password	Password of the connection user. If you change the password, you need to create a connection again. **NOTICE** Ensure the account has been used for logging in to MRS Manager. If you use a new account, you will be asked to change your password when you first log in. In this case, the MRS data source will fail to be configured.
Use a Machine-Machine Account	Creates a machine-machine account named dws in MRS and uses it for interaction with MRS. This account is in the **supergroup** group and has all permissions. If the switch is toggled off, the configured man-machine account will be used. Ensure this account has the permission to access data, or a message will be displayed during data source access, indicating the required file does not exist.

Table 6-7 Parameters of the File upload mode

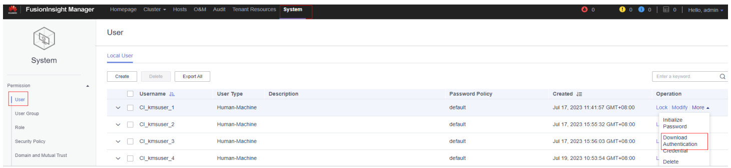
<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Authentication Credential</td><td>Keytab file of a user A credential file downloaded from Manager of the MRS cluster. File name format: <b>Username_Timestamp_keytab.tar</b>
<ul>
<li><b>For MRS 2.x or earlier</b>, choose <b>System</b> &gt; <b>Manage User</b>. In the <b>Operation</b> column of a user, choose <b>More</b> &gt; <b>Download authentication credential</b>.



MRS Manager | Hello, admin
Dashboard | Services | Hosts | Alarms | Audit | Tenant | System
Cluster Name: [redacted] 08/03/2023 16:19:18 GMT+08:00
System › Manage User
+ Create User | Search

<table>
<tr><th>Username</th><th>Description</th><th>Created</th><th>Operation</th></tr>
<tr><td>[redacted]</td><td>IAM System Policy User</td><td>03/21/2022 21:12:42 GMT+08:00</td><td>Modify Lock User More</td></tr>
<tr><td>[redacted]</td><td>IAM System Policy User</td><td>03/21/2022 21:12:42 GMT+08:00</td><td></td></tr>
<tr><td>OU-test-deny</td><td>IAM Custom Policy User</td><td>06/12/2023 14:35:08 GMT+08:00</td><td></td></tr>
</table>

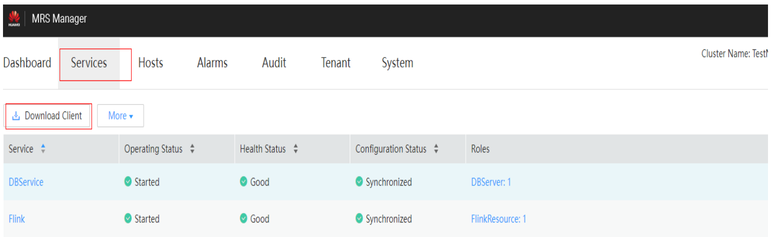
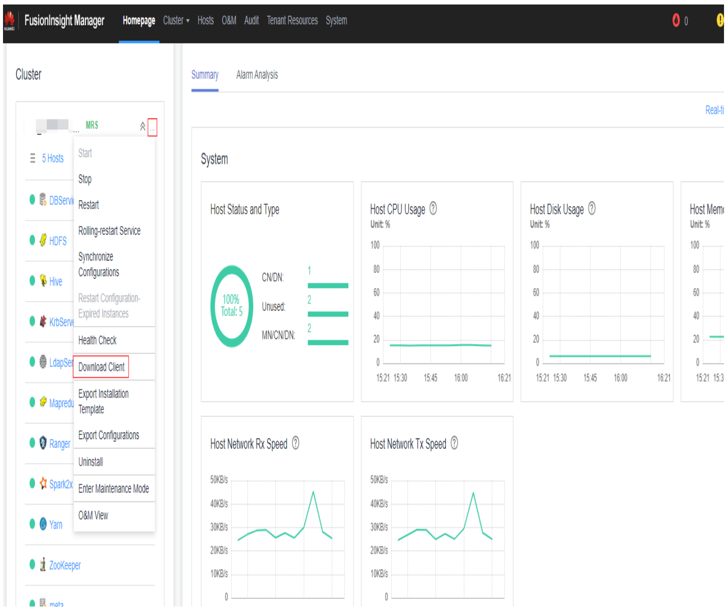
More: Initialize password | Download authentication credential | Delete
</li>
<li><b>For MRS 3.x or later</b>, choose <b>System</b> &gt; <b>Permission</b> &gt; <b>User</b>. In the <b>Operation</b> column of a user, choose <b>More</b> &gt; <b>Download Authentication Credential</b>.



FusionInsight Manager | Homepage | Cluster | Hosts | O&amp;M | Audit | Tenant Resources | System | Hello, admin
System
Permission | User | User Group | Role | Security Policy | Domain and Mutual Trust
User
Local User
Create | Delete | Export All | Enter a keyword.

<table>
<tr><th>Username</th><th>User Type</th><th>Description</th><th>Password Policy</th><th>Created</th><th>Operation</th></tr>
<tr><td>CI_kmsuser_1</td><td>Human-Machine</td><td></td><td>default</td><td>Jul 17, 2023 11:41:57 GMT+08:00</td><td>Lock Modify More</td></tr>
<tr><td>CI_kmsuser_2</td><td>Human-Machine</td><td></td><td>default</td><td>Jul 17, 2023 15:55:32 GMT+08:00</td><td></td></tr>
<tr><td>CI_kmsuser_3</td><td>Human-Machine</td><td></td><td>default</td><td>Jul 17, 2023 15:56:03 GMT+08:00</td><td></td></tr>
<tr><td>CI_kmsuser_4</td><td>Human-Machine</td><td></td><td>default</td><td>Jul 19, 2023 10:53:54 GMT+08:00</td><td></td></tr>
</table>

More: Initialize Password | Download Authentication Credential | Delete
</li>
</ul>
</td></tr>
</table>

Parameter	Description
Client Profile	Client configuration files of HDFS, Hive, and hosts. When downloading the client, set **Select Client Type** to **Configuration Files Only**. ● **For MRS 2.x or earlier**, choose **Services** and click **Download Client**.  MRS Manager Dashboard Services Hosts Alarms Audit Tenant System Cluster Name: Test Download Client More Service | Operating Status | Health Status | Configuration Status | Roles DBService | Started | Good | Synchronized | DBServer: 1 Flink | Started | Good | Synchronized | FlinkResource: 1 ● **For MRS 3.x or later**, choose **Homepage**. Click the **More** icon and choose **Download Client**.  FusionInsight Manager Homepage Cluster Hosts O&M Audit Tenant Resources System Cluster Summary Alarm Analysis MRS 5 Hosts DBService HDFS Hive KrbServer LdapServer Mapreduce Ranger Spark2x Yarn ZooKeeper Start Stop Restart Rolling-restart Service Synchronize Configurations Restart Configuration-Expired Instances Health Check Download Client Export Installation Template Export Configurations Uninstall Enter Maintenance Mode O&M View System Host Status and Type 100% Total: 5 CN/DN: 1 Unused: 2 MN/CN/DN: 2 Host CPU Usage Unit: % Host Disk Usage Unit: % Host Memory Unit: % 100 80 60 40 20 0 15:21 15:30 15:45 16:00 16:21 Host Network Rx Speed Host Network Tx Speed 50KB/s 40KB/s 30KB/s 20KB/s 10KB/s 0

Step 8 Click **OK** to save the connection.

Configuration Status turns to **Creating**. You can view the connection that is successfully created in the MRS data source list and the connection status is **Available**.

NOTE

- In the **Operation** column, you can click **Update Configurations** to update **MRS Cluster Status** and **Configuration Status**. During configuration update, you cannot create a connection. The system checks whether the security group rule is correct. If the rule is incorrect, the system rectifies the fault. For details, see **Updating the MRS Data Source Configuration**.
- In the **Operation** column, you can click **Delete** to delete the unnecessary connection. When deleting a connection, you need to manually delete the security group rule.
- If the security group rules are not deleted, nodes in the data warehouse cluster can still communicate with nodes in the MRS cluster. If you have strict requirements on network security, manually delete the rules.

----End

6.2.1.3 Updating the MRS Data Source Configuration

Scenario

For MRS, if the following parameter configurations of the HDFS cluster change, data may fail to be imported to the data warehouse cluster from the HDFS cluster. Before importing data using the HDFS cluster, you must update the MRS data source configuration.

Prerequisites

You have created an MRS data source connection for the data warehouse cluster.

Impact on the System

When you are updating an MRS data source connection, the data warehouse cluster will automatically restart and cannot provide services.

Procedure

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, click **MRS Data Sources**.

Step 3 In the MRS data source list, select the MRS data source that you want to update. In the **Operation** column, click **Update Configurations**.

MRS Cluster Status and **Configuration Status** of the current connection will be updated. During configuration update, you cannot create a connection. The system checks whether the security group rule is correct. If the rule is incorrect, the system rectifies the fault. The following table describes the parameters.

Table 6-8 Parameter description

Parameter	Description
dfs.client.read.shortcircuit	Specifies whether to enable the local read function.

Parameter	Description
dfs.client.read.shortcircuit.skip.checksum	Specifies whether to skip data verification during the local read.
dfs.client.block.write.replace-datanode-on-failure.enable	Specifies whether to replace the location storing copies with the new node when data blocks fail to be written to HDFS.
dfs.encrypt.data.transfer	Specifies whether to enable data encryption. The value **true** indicates that the channels are encrypted. The channels are not encrypted by default. **NOTE** ● This parameter is available only for clusters with Kerberos authentication enabled. ● This parameter is valid only when **hadoop.rpc.protection** is set to **privacy**.
dfs.encrypt.data.transfer.algorithm	Specifies the encryption and decryption algorithm for key transmission. This parameter is valid only when **dfs.encrypt.data.transfer** is set to **true**. The default value is **3des**, indicating that the 3DES algorithm is used for encryption.
dfs.encrypt.data.transfer.cipher.suites	Specifies the encryption and decryption algorithm for the transmission of actually stored data. If this parameter is not specified, the cryptographic algorithm specified by **dfs.encrypt.data.transfer.algorithm** is used for data encryption. The default value is **AES/CTR/NoPadding**.
dfs.replication	Specifies the default number of data copies.
dfs.blocksiz	Specifies the default size of a data block.
hadoop.security.authentication	Specifies the security authentication mode.

Parameter	Description
hadoop.rpc.protection	Specifies the RPC communication protection mode. Default value: ● Security mode (Kerberos authentication enabled): **privacy** ● Common mode (Kerberos authentication disabled): **authentication** **NOTE** ● **authentication**: indicates that only authentication is required. ● **integrity**: indicates that authentication and consistency check need to be performed. ● **privacy**: indicates that authentication, consistency check, and encryption need to be performed.
dfs.domain.socket.path	Specifies the locally used **Domain socket** path.

----End

6.2.2 Managing OBS Data Sources

GaussDB(DWS) allows you to access data on OBS by using an agency. You can create a GaussDB(DWS) agency, grant the OBS OperateAccess or OBS Administrator permission to the agency, and bind the agency to an OBS data source you created. In this way, you can access data on OBS by using OBS foreign tables.

NOTE

- This feature is supported only in 8.2.0 or later.
- For the OBS data source of a cluster, only one of the creation, modification, and deletion operations can be performed at a time.

Creating an OBS Agency

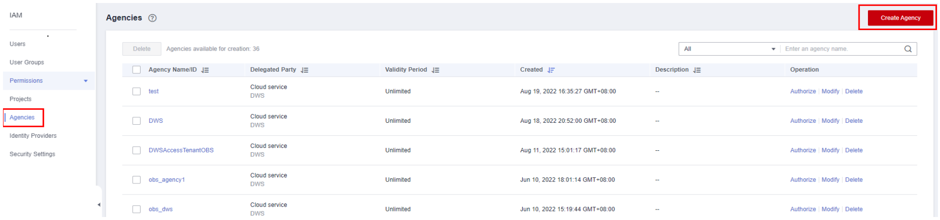
Scenario

Before creating an OBS data source, create an agency that grants GaussDB(DWS) the OBS OperateAccess or OBS Administrator permission.

Procedure

Step 1 Click your account in the upper right corner of the page and choose **Identity and Access Management**.

Step 2 In the navigation pane on the left, choose **Agency**. In the upper right corner, click **Create Agency**.

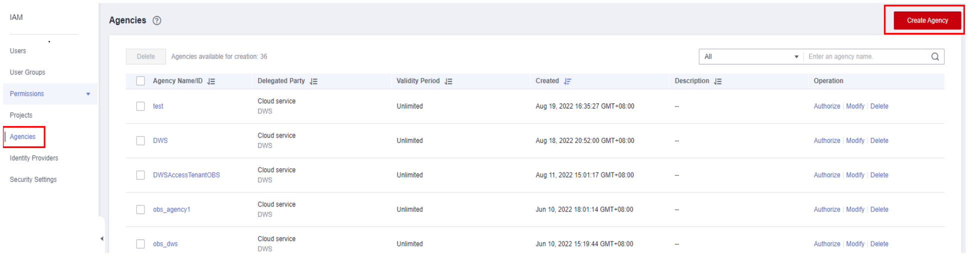


Agencies available for creation: 36

Agency Name/ID	Delegated Party	Validity Period	Created	Description	Operation
test	Cloud service DWS	Unlimited	Aug 19, 2022 16:35:27 GMT+08:00	--	Authorize | Modify | Delete
DWS	Cloud service DWS	Unlimited	Aug 18, 2022 20:52:00 GMT+08:00	--	Authorize | Modify | Delete
DWSAccessTenantOBS	Cloud service DWS	Unlimited	Aug 11, 2022 15:01:17 GMT+08:00	--	Authorize | Modify | Delete
obs_agency1	Cloud service DWS	Unlimited	Jun 10, 2022 18:01:14 GMT+08:00	--	Authorize | Modify | Delete
obs_dws	Cloud service DWS	Unlimited	Jun 10, 2022 15:19:44 GMT+08:00	--	Authorize | Modify | Delete

Step 3 Select **Cloud Service** and set **Cloud Service** to **DWS**.

Step 4 Click **Next** to grant the OBS OperateAccess or OBS Administrator permission to the agency.



Agencies available for creation: 36

Agency Name/ID	Delegated Party	Validity Period	Created	Description	Operation
test	Cloud service DWS	Unlimited	Aug 19, 2022 16:35:27 GMT+08:00	--	Authorize | Modify | Delete
DWS	Cloud service DWS	Unlimited	Aug 18, 2022 20:52:00 GMT+08:00	--	Authorize | Modify | Delete
DWSAccessTenantOBS	Cloud service DWS	Unlimited	Aug 11, 2022 15:01:17 GMT+08:00	--	Authorize | Modify | Delete
obs_agency1	Cloud service DWS	Unlimited	Jun 10, 2022 18:01:14 GMT+08:00	--	Authorize | Modify | Delete
obs_dws	Cloud service DWS	Unlimited	Jun 10, 2022 15:19:44 GMT+08:00	--	Authorize | Modify | Delete

Step 5 Click **Next**. Select **All resources** or specific resources, confirm the information, and click **Submit**.

----End

Creating an OBS Data Source

Prerequisites

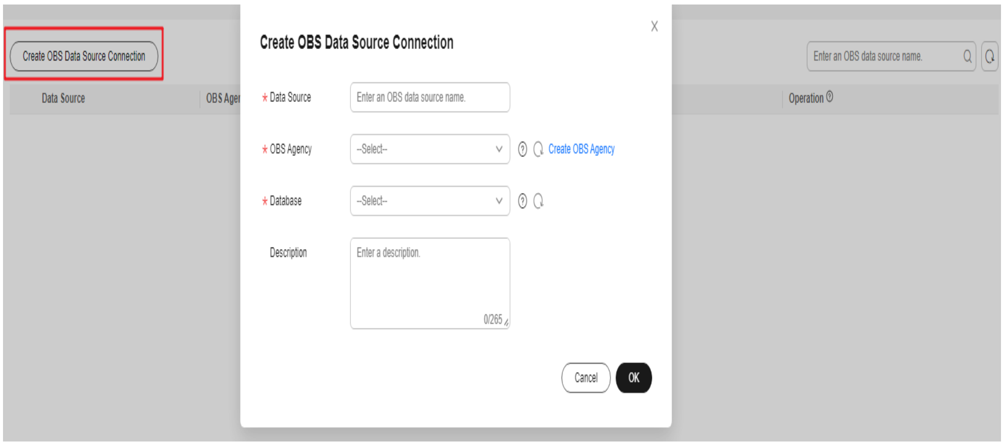
An agency has been created to grant GaussDB(DWS) the OBS OperateAccess permission.

Procedure

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **OBS Data Source**.

Step 3 Click **Create OBS Cluster Connection** and configure parameters.



Create OBS Data Source Connection
Data Source
OBS Ager
Enter an OBS data source name.
Operation
X
Create OBS Agency
Cancel
OK

Create OBS Data Source Connection

* Data Source: Enter an OBS data source name.

* OBS Agency: --Select--

* Database: --Select--

Description: Enter a description. 0/265

Table 6-9 OBS data source connection parameters

Parameter	Description
Data Source	Name of the OBS data source connection to be created. You can assign a personalized value to this parameter. The data source name is used as the server name specified in the statement for creating an OBS foreign table.
OBS Agency	Agency with the OBS OperateAccess permission to be granted to GaussDB(DWS)
Database	Database where the OBS data source connection is to be created
Description	Description about the OBS data source connection

Step 4 Confirm the settings and click **OK**. The creation takes about 10 seconds.

----End

Updating the OBS Data Source Configuration

Scenario

After an OBS data source connection is created, GaussDB(DWS) periodically updates the temporary agency information used by the data source. If the automatic update fails for 24 hours, the data source connection will be unavailable. To solve this problem, manually update the information on the console.

Procedure

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **OBS Data Source**.

Step 3 In the **Operation** column of an OBS data source, click **Update Configuration**.

Step 4 Confirm the settings and click **OK**. The update takes about 10 seconds.

----End

Changing the OBS Data Source Agency

Scenario

You can change the agency bound to the OBS data source.

Procedure

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **OBS Data Source**.

Step 3 In the **Operation** column of a data source, click **Manage Agency**. In the dialog box that is displayed, select a new agency.

Step 4 Confirm the settings and click **OK**. The change takes about 10 seconds.

----End

Deleting an OBS Data Source

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **OBS Data Source**.

Step 3 In the **Operation** column of an OBS data source, click **Delete**.

Step 4 Confirm the settings and click **OK**. The deletion takes about 10 seconds.

----End

Using an OBS Data Source

GaussDB(DWS) uses foreign tables to access data on OBS. The **SERVER** parameters specified for accesses with and without an agency are different.

If you access OBS without an agency, the **SERVER** provided on the console contains parameters **access_key** and **secret_access_key**, which are the AK and SK of the OBS access protocol, respectively.

If you access OBS with an agency, the **SERVER** provided on the console contains the **access_key**, **secret_access_key**, and **security_token** parameters, which are the temporary AK, temporary SK, and the **SecurityToken** value of the temporary security credential in IAM, respectively.

After the OBS agency and OBS data source are created, you can obtain the **SERVER** information on the console. Assume that the OBS data source name is **obs_server**. The way users create and use foreign tables with an agency is the same as the way they do without an agency. For how to use the OBS data source, see **Importing Data from OBS**.

The following example shows how common user **jim** reads data from OBS through a foreign table.

1. Repeat the preceding steps to create an OBS data source named **obs_server**.
2. Connect to the database as system administrator dbadmin, create a common user, and grant the common user the permission to use OBS servers and OBS foreign tables. Replace **{Password}** with the actual password and **obs_server** with the actual OBS data source name.
   ```
   CREATE USER jim PASSWORD '{Password}';
   ALTER USER jim USEFT;
   GRANT USAGE ON FOREIGN SERVER obs_server TO jim;
   ```
3. Connect to the database as common user **jim** and create an OBS foreign table **customer_address** that does not contain partition columns.

   In the following command, replace **obs_server** with the name of the created OBS data source. Replace **/user/obs/region_orc11_64stripe1/** with the actual OBS directory for storing data files. **user** indicates the OBS bucket name.

```
CREATE FOREIGN TABLE customer_address
(
    ca_address_sk             integer               not null,
    ca_address_id             char(16)              not null,
    ca_street_number          char(10)                      ,
    ca_street_name            varchar(60)                   ,
    ca_street_type            char(15)                    ,
    ca_suite_number           char(10)                      ,
    ca_city                   varchar(60)                 ,
    ca_county                 varchar(30)                   ,
    ca_state                  char(2)                     ,
    ca_zip                    char(10)                    ,
    ca_country                varchar(20)                   ,
    ca_gmt_offset             decimal(36,33)                  ,
    ca_location_type          char(20)
)
SERVER obs_server OPTIONS (
    FOLDERNAME '/user/obs/region_orc11_64stripe1/',
    FORMAT 'ORC',
    ENCODING 'utf8',
    TOTALROWS  '20'
)
DISTRIBUTE BY roundrobin;
```

4. Query data stored in OBS by using a foreign table.

```
SELECT COUNT(*) FROM customer_address;
count
-------
20
(1row)
```

6.2.3 Managing LakeFormation Data Sources

On the GaussDB(DWS console), you can create a LakeFormation data source on the console to access metadata on LakeFormation.

NOTE

- This feature is for limited commercial use only. It is available for storage-compute decoupled clusters of 9.0.1 and later cluster versions or storage-compute coupledclusters in version 8.2.1.300 and later.
- LakeFormation is connected through VPC Endpoint. When you create a LakeFormation data source, if there is no VPC endpoint for your GaussDB(DWS) cluster, it will be automatically created. VPC endpoints incur extra fees.

Prerequisites

- A LakeFormation instance is available. For details, see section "Creating a LakeFormation Instance" in *LakeFormation Usage Guide*.
- Create an agency with LakeFormation permissions (including the minimum permissions). For details about how to configure the permissions, see section "Data Permission Authorization" in *LakeFormation Usage Guide*. If the permissions are not correctly configured, an error will be reported.
- For an IAM user to use GaussDB(DWS) to call APIs on the LakeFormation management plane, the user must have LakeFormation permissions (at least **lakeformation:instance:access** and **lakeformation:instance:describe**).

Creating a LakeFormation Data Source

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **LakeFormation Data Sources**.

Step 3 Click **Create LakeFormation Data Source Connection** and configure parameters.

Figure 6-4 Creating a LakeFormation data source connection

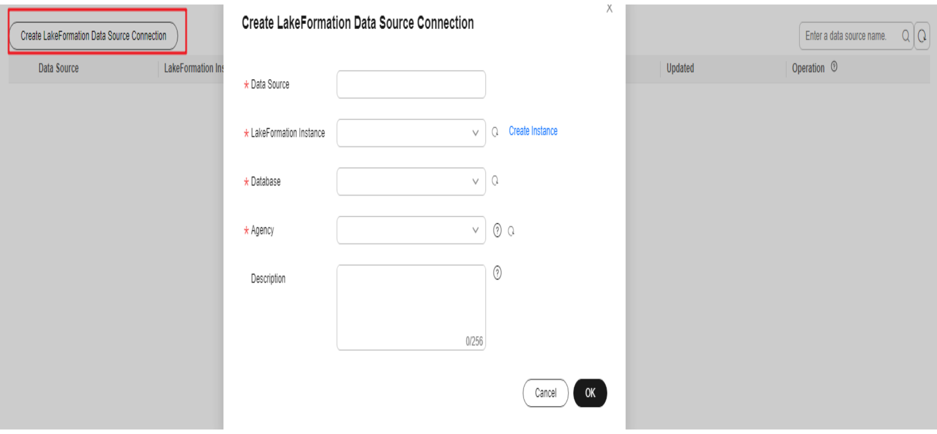


Table 6-10 LakeFormation data source connection parameters

Parameter	Description
Data Source	Name of the LakeFormation data source connection to be created
LakeFormation Instance	LakeFormation cluster instance to be bound
Database	Database where the LakeFormation data source connection is to be created
Agency	An agency authorized by LakeFormation. GaussDB(DWS) interacts with LakeFormation through this agency token o obtain metadata.
Description	Description about the LakeFormation data source connection

Step 4 Confirm the settings and click **OK**. The creation takes about 1 minute.

----End

Updating a Configuration

Scenario

- After a data source connection is created, its VPC endpoint is deleted by mistake and the data source cannot be used.
- The agency needs to be changed.
- A token fails to be updated. After this issue is fixed, the token needs to be updated immediately.

Procedure

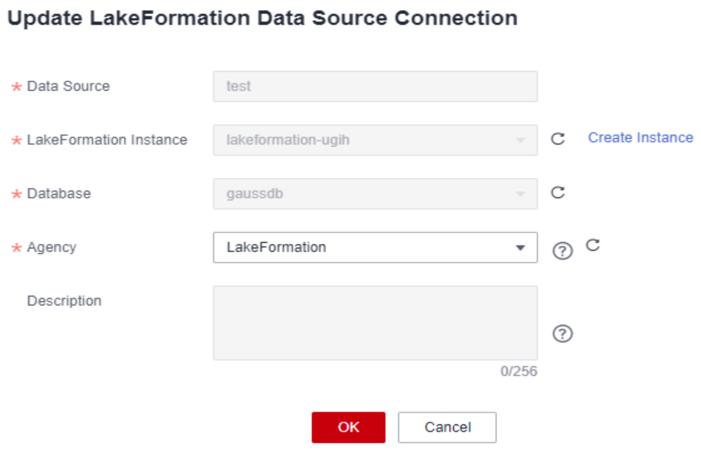
Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **LakeFormation Data Sources**.

Step 3 In the **Operation** column of a LakeFormation data source, click **Update Configuration**.

Step 4 During the update, you can only change the agency. After confirming that the agency is correct, click **OK** to submit the update. The update takes about 1 minute.

Figure 6-5 Updating a LakeFormation data source connection


Update LakeFormation Data Source Connection

* Data Source: test
* LakeFormation Instance: lakeformation-ugih
* Database: gaussdb
* Agency: LakeFormation
Description:

----End

Deleting a LakeFormation Data Source

Step 1 On the GaussDB(DWS) console, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of a cluster. On the page that is displayed, choose **Data Sources** > **LakeFormation Data Sources**.

Step 3 In the **Operation** column of a LakeFormation data source, click **Delete**.

Step 4 Confirm the settings and click **OK**. The deletion takes about 10 seconds.

----End

Using a LakeFormation Data Source

For details about how to use the LakeFormation data source, see "Data Migration > Data Import > Using LakeFormation to Import Data" in *Data Warehouse Service Developer Guide*

7 GaussDB(DWS) Cluster Data Security and Encryption

7.1 Enabling Separation of Duties for GaussDB(DWS) Database Users

Scenario

By default, the administrator specified when you create a GaussDB(DWS) cluster is the database system administrator. The administrator can create other users and view the audit logs of the database. That is, separation of permissions is disabled.

To ensure cluster data security, GaussDB(DWS) supports separation of duties for clusters. Different types of users have different permissions.

For details about the default permissions mode and the separation of permissions mode, see **Separation of Permissions** in the *Data Warehouse Service (DWS) Developer Guide*.

Impact on the System

- After you modified the security parameters and the modifications take effect, the cluster may be restarted, which makes the cluster unavailable temporarily.
- When a storage-compute decoupled cluster is created, a logical cluster is created by default. After the separation of duties is enabled, only the system administrator has the permission to create, modify, delete, and allocate logical clusters. Accessing a logical cluster requires permissions.

Prerequisites

To modify the cluster's security configuration, ensure that the following conditions are met:

- The cluster status is **Available** or **Unbalanced**.
- The target cluster should not be undergoing any node additions, specification changes, configurations, upgrades, redistribution operations, or restarts.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of a cluster. On the page that is displayed, click **Security Settings**.

By default, **Configuration Status** is **Synchronized**, which indicates that the latest database result is displayed.

Step 4 On the **Security Settings** page, configure separation of permissions.

When separation of permissions is enabled, configure the username and password for **Security Administrator** and **Audit Administrator**. Then the system automatically creates these two users. You can use these two users to connect to the database and perform database-related operations. **By default, this function is disabled.**

Table 7-1 Security parameters

Parameter	Description	Example Value
Security Administrator	The name must: ● Consist of lowercase letters, digits, or underscores. ● Start with a lowercase letter or an underscore. ● Contain 6 to 64 characters. ● Cannot be a keyword of the GaussDB(DWS) database. For details about the keywords of the GaussDB(DWS) database, see **Keyword** in the *Data Warehouse Service (DWS) Developer Guide*.	security_admin
Password	The password must: ● Contain 12 to 32 characters. ● Cannot be the username or the username spelled backwards. ● Contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\\=-) ● Perform the weak password check on the passwords that you have created.	-
Confirm Password	Enter the password of the security administrator again.	-

Parameter	Description	Example Value
Audit Administrator	The name must: ● Consist of lowercase letters, digits, or underscores. ● Start with a lowercase letter or an underscore. ● Contain 6 to 64 characters. ● Cannot be a keyword of the GaussDB(DWS) database. For details about the keywords of the GaussDB(DWS) database, see **Keyword** in the *Data Warehouse Service (DWS) Developer Guide*.	audit_admin
Password	The password must: ● Contain 12 to 32 characters. ● Cannot be the username or the username spelled backwards. ● Contain at least 3 of the following character types: uppercase letters, lowercase letters, digits, and special characters ~!@#%^&*()-_=+|[{}];:,<.>/? ● Pass the weak password check.	-
Confirm Password	Enter the password of the audit administrator again.	-

Step 5 Click **Apply**.

Step 6 In the displayed **Save Configuration** dialog box, select or deselect **Restart the cluster** and click **Yes**.

- If you select **Restart the cluster**, the system saves the settings on the **Security Settings** page and restarts the cluster immediately. After the cluster is restarted, the security settings take effect immediately.
- If you do not select **Restart the cluster**, the system only saves the settings on the **Security Settings** page. Later, you need to manually restart the cluster for the security settings to take effect.

After the security settings are complete, **Configuration Status** can be one of the following on the **Security Settings** page:

- **Applying**: The system is saving the settings.
- **Synchronized**: The settings have been saved and taken effect.
- **Take effect after restart**: The settings have been saved but have not taken effect. Restart the cluster for the settings to take effect.

----End

7.2 Using KMS to Encrypt GaussDB(DWS) Clusters

7.2.1 Overview

Encrypting GaussDB(DWS) Databases

In GaussDB(DWS), you can enable database encryption for a cluster to protect static data. After you enable encryption, data of the cluster and its snapshots is encrypted. Encryption is an optional and immutable setting that can be configured during cluster creation. To encrypt an unencrypted cluster, you must export all data from the unencrypted cluster and import it into a new cluster that has database encryption enabled. GaussDB(DWS) encrypts data as it is written to the database, and automatically decrypts it when queried, returning the results to the user.

If encryption is required, enable it during cluster creation. Although encryption is an optional setting of GaussDB(DWS), you are advised to enable this setting for clusters to protect data.

NOTICE

- Only storage-compute decoupled clusters of version 9.1.0 and later support database encryption.
- The database encryption function cannot be disabled once it is enabled. For details, see **Encrypting the Database**. After a normal cluster is created, you can convert it to an encrypted cluster.
- After **Encrypt DataStore** is enabled, the key cannot be disabled, deleted, or frozen when being used. Otherwise, the cluster becomes abnormal and the database becomes unavailable.
- Snapshots created after the database encryption function is enabled cannot be restored using open APIs.

Viewing Database Encryption Information

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 Click the name of a cluster. The **Cluster Information** page is displayed.

Step 4 In the **Data Encryption Information** area on the cluster information page, view the database encryption information, as shown in **Table 7-2**.

Table 7-2 Data encryption information

Parameter	Description
Key Name	Indicates the database encryption key of the cluster when **Encrypt DataStore** is enabled.
Last Key Rotation Time	Indicates the time when the last encryption key is rotated when **Encrypt DataStore** is enabled.
Cryptographic Algorithm	Indicates the encryption algorithm of the cluster when **Encrypt DataStore** is enabled. Encryption algorithms include: ● AES256 (general encryption algorithm, SM algorithms not supported) ● SM4 (compatible with international algorithms)

NOTE

If database encryption is disabled by default during cluster creation, the encryption module is not displayed on the cluster details page.

----End

Encrypting GaussDB(DWS) Databases Using KMS

When you choose KMS to manage GaussDB(DWS) keys, a three-layer key management structure is adopted, including the cluster master key (CMK), cluster encryption key (CEK), and database encryption key (DEK).

- The CMK is used to encrypt the CEK and is stored in KMS.
- The CEK is used to encrypt the DEK. The CEK plaintext is stored in the data warehouse cluster's memory, and the ciphertext is stored in GaussDB(DWS).
- The DEK is used to encrypt database data. The DEK plaintext is stored in the data warehouse cluster's memory, and the ciphertext is stored in GaussDB(DWS).

The procedure of using the keys is as follows:

1. You choose a CMK.
2. GaussDB(DWS) randomly generates the CEK and DEK plaintext.
3. KMS uses the CMK you choose to encrypt the CEK plaintext and imports the encrypted CEK ciphertext to GaussDB(DWS).
4. GaussDB(DWS) uses the CEK plaintext to encrypt the DEK plaintext and saves the encrypted DEK ciphertext.
5. GaussDB(DWS) transfers the DEK plaintext to the cluster and loads it to the cluster's memory.

When the cluster is restarted, it automatically requests the DEK plaintext from GaussDB(DWS) through an API. GaussDB(DWS) loads the CEK and DEK ciphertext to the cluster's memory, invokes KMS to decrypt the CEK using the CMK, loads the

CEK to the memory, decrypts the DEK using the CEK plaintext, loads the DEK to the memory, and returns it to the cluster.

Rotating Encryption Keys

Encryption key rotation is used to update the ciphertext stored on GaussDB(DWS). On GaussDB(DWS), you can rotate the encrypted CEK of an encrypted cluster.

The procedure of rotating the keys is as follows:

1. The GaussDB(DWS) cluster starts key rotation.
2. GaussDB(DWS) decrypts the CEK ciphertext stored on GaussDB(DWS) based on the CMK to obtain the CEK plaintext.
3. Use the obtained CEK plaintext to decrypt the DEK ciphertext in GaussDB(DWS) to obtain the DEK plaintext.
4. GaussDB(DWS) randomly generates new CEK plaintext.
5. GaussDB(DWS) uses the new CEK plaintext to encrypt the DEK and saves the encrypted DEK ciphertext.
6. Use the CMK to encrypt the new CEK plaintext and import the encrypted CEK ciphertext to GaussDB(DWS).

You can plan the key rotation interval based on service requirements and data types. To improve data security, you are advised to periodically rotate the keys to prevent the keys from being cracked. Once you find that your keys may have been disclosed, rotate the keys in time.

NOTE

- When GaussDB(DWS) rotates the cluster's CEK, snapshots of the cluster do not need CEK rotation, because the CEK is not stored in snapshots. The CEK plaintext is stored in the GaussDB(DWS) cluster memory, and the ciphertext is stored in GaussDB(DWS).
- The DEK is not updated during key rotation, so data encryption and decryption are not affected.

7.2.2 Rotating Encryption Keys

If you have enabled the **Encrypt DataStore** function in **Advanced Settings** during cluster creation, you can rotate the encryption keys for the cluster after the cluster is created successfully. When a normal cluster is converted to an encrypted cluster, you can rotate the encryption key for the cluster. Each key rotation will update the CEK once. During the key rotation, the cluster is still in **Available** status.

Rotating Encryption Keys for GaussDB(DWS) Clusters

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 In the **Data Encryption Information** area, click **Key Rotation**.

Step 5 In the dialog box that is displayed, click **Yes**.

----End

7.2.3 Converting an Ordinary Cluster to an Encrypted Cluster

GaussDB(DWS) allows you to convert an unencrypted cluster to an encrypted cluster when the cluster status is **Available** on the console. To ensure data security, converting a cluster to an encrypted cluster is an **irreversible high-risk operation** and will restart the cluster. As a result, services may be unavailable for a short period of time. Exercise caution when performing this operation.

NOTE

If the current console does not support this feature, contact technical support.

Creating a KMS Agency

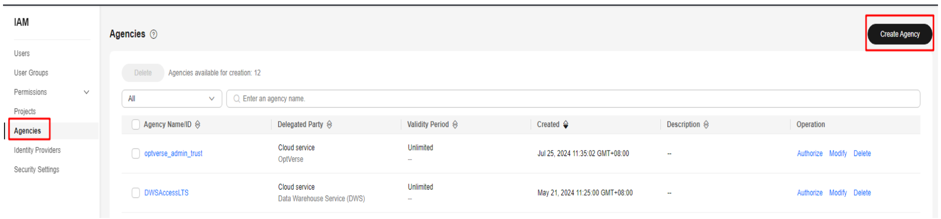
Scenario

Before converting a cluster to an encrypted cluster, you need to create an agency that grants the KMS Administrator permissions to GaussDB(DWS).

Procedure

Step 1 Click your account in the upper right corner of the page and choose **Identity and Access Management**.

Step 2 In the navigation pane on the left, choose **Agency**. In the upper right corner, click **Create Agency**.

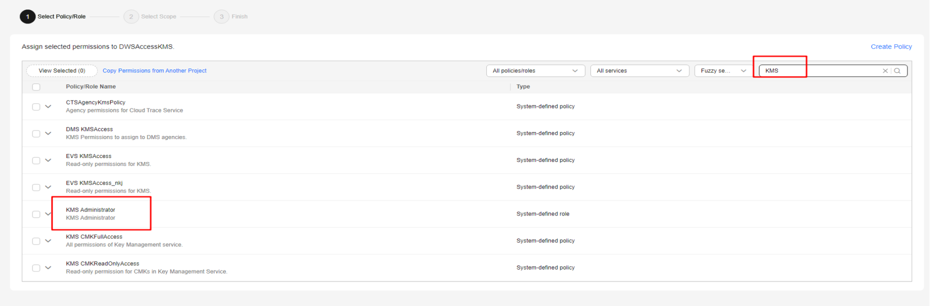


Agencies available for creation: 12

Agency Name/ID	Delegated Party	Validity Period	Created	Description	Operation
optverse_admin_trust	Cloud service OptVerse	Unlimited --	Jul 25, 2024 11:35:02 GMT+08:00	--	Authorize Modify Delete
DWSAccessLTS	Cloud service Data Warehouse Service (DWS)	Unlimited --	May 21, 2024 11:25:00 GMT+08:00	--	Authorize Modify Delete

Step 3 Select **Cloud Service** and set **Cloud Service** to **DWS**.

Step 4 Click **Finish**. In the displayed dialog box, click **OK** to grant the **KMS Administrator** permission to the agency.



Assign selected permissions to DWSAccessKMS.

Policy/Role Name	Type
CTSAgencyKmsPolicy Agency permissions for Cloud Trace Service	System-defined policy
DMS KMSAccess KMS Permissions to assign to DMS agencies.	System-defined policy
EVS KMSAccess Read-only permissions for KMS.	System-defined policy
EVS KMSAccess_nkj Read-only permissions for KMS.	System-defined policy
KMS Administrator KMS Administrator	System-defined role
KMS CMKFullAccess All permissions of Key Management service.	System-defined policy
KMS CMKReadOnlyAccess Read-only permission for CMKs in Key Management Service.	System-defined policy

Step 5 Click **Next**. Select **All resources** or specific resources, confirm the information, and click **Submit**.

----End

Procedure

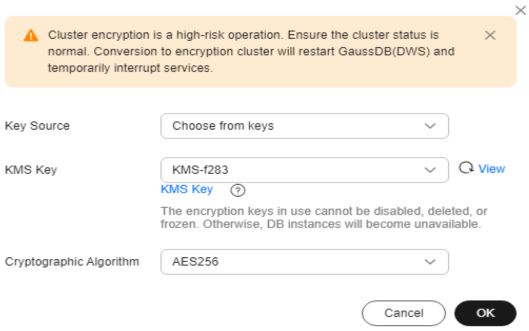
Step 1 Log in to the GaussDB(DWS) console. In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, locate the row that contains the target cluster and choose **More** > **Convert to Encrypted Cluster** in the **Operation** column.

Step 3 In the dialog box that is displayed, select the key source, key name, and encryption algorithm to convert the cluster to an encrypted cluster.

- Method 1: Select a key name. You can **create a resource share** to share KMS resources with other members. After **accepting the sharing invitation**, members can select the shared KMS resource from the key source.

Figure 7-1 Select from existing keys



Cluster encryption is a high-risk operation. Ensure the cluster status is normal. Conversion to encryption cluster will restart GaussDB(DWS) and temporarily interrupt services.

Key Source: Choose from keys

KMS Key: KMS-f283

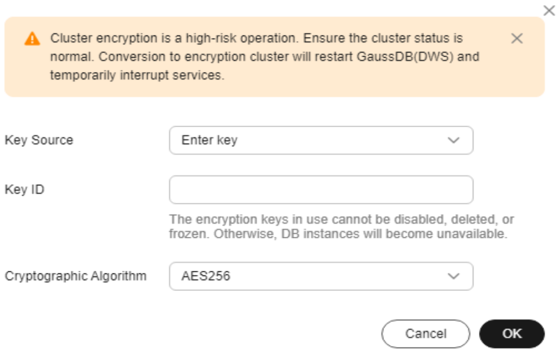
The encryption keys in use cannot be disabled, deleted, or frozen. Otherwise, DB instances will become unavailable.

Cryptographic Algorithm: AES256

- Method 2: Enter the key ID. Enter the key ID used for authorizing the current tenant. For details, see **Viewing a CMK**.

  When you grant permissions on the **Creating a Grant** page, the authorized object must be an account instead of a user. The authorized operations must at least contain **Querying key details**, **Encrypting data**, and **Decrypting data**.

Figure 7-2 Key ID



Cluster encryption is a high-risk operation. Ensure the cluster status is normal. Conversion to encryption cluster will restart GaussDB(DWS) and temporarily interrupt services.

Key Source: Enter key

Key ID:

The encryption keys in use cannot be disabled, deleted, or frozen. Otherwise, DB instances will become unavailable.

Cryptographic Algorithm: AES256

Table 7-3 Description

Parameter	Description
Key Source	You can select a key name from the key list or directly enter a key name.
Cryptographic Algorithm	Encryption algorithms include: ● AES256 (general encryption algorithm, SM algorithms not supported) ● SM4 (compatible with international algorithms)

NOTE

- The database encryption function cannot be disabled once it is enabled.
- After **Encrypt DataStore** is enabled, the key cannot be disabled, deleted, or frozen when being used. Otherwise, the cluster becomes abnormal and the database becomes unavailable.
- Snapshots created after the database encryption function is enabled cannot be restored using open APIs.
- By default, only Huawei Cloud accounts or users with **Security Administrator** permissions can query and create agencies. IAM users under an account do not have the permission to query or create agencies by default. Contact a user with that permission and complete the authorization on the current page.

Step 4 After the conversion, you can click the cluster name to go to the **Cluster Details** page to view the cluster details. For details, see **Viewing Database Encryption Information**.

----End

7.3 Enabling Critical Operation Protection for the GaussDB(DWS) Console

Scenario

GaussDB(DWS) protects mission-critical operations. If you want to perform a mission-critical operation on the management console, you must enter a credential for identity verification. You can perform the operation only after your identity is verified. For account security, it is a good practice to enable operation protection. The setting will take effect for both the account and users under the account.

Currently, the following operations are supported: binding an EIP, scaling out a cluster, changing specifications, deleting a cluster, restarting a cluster, starting a cluster, stopping a cluster, adding or deleting a CN, upgrading a cluster, modifying parameters, deleting idle nodes, and enabling or disabling auto scaling.

Enabling Critical Operation Protection

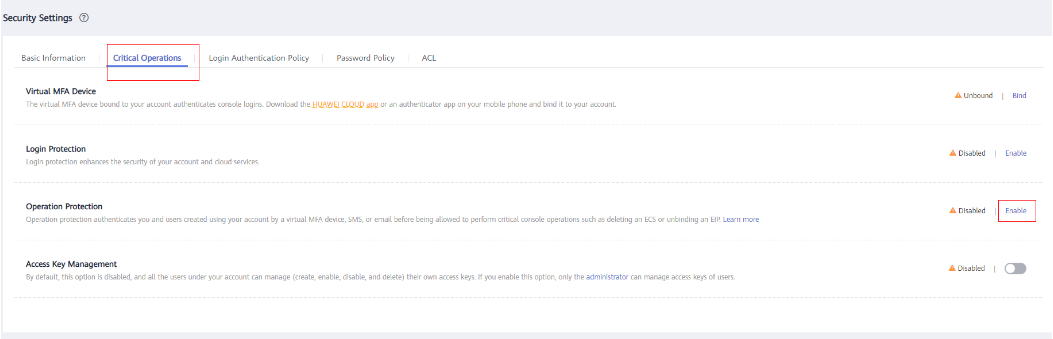
Operation protection is disabled by default. To enable it, perform the following steps:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Move the cursor to the username in the upper right corner of the page and click **Security Settings** from the drop-down list.

Step 3 On the **Security Settings** page, click the **Critical Operations** tab. Click **Enable** in the **Operation Protection** area.

Figure 7-3 Critical Operations



Virtual MFA Device

The virtual MFA device bound to your account authenticates console logins. Download the HUAWEI CLOUD app or an authenticator app on your mobile phone and bind it to your account.

Login Protection

Login protection enhances the security of your account and cloud services.

Operation Protection

Operation protection authenticates you and users created using your account by a virtual MFA device, SMS, or email before being allowed to perform critical console operations such as deleting an ECS or unbinding an EIP. Learn more

Access Key Management

By default, this option is disabled, and all the users under your account can manage (create, enable, disable, and delete) their own access keys. If you enable this option, only the administrator can manage access keys of users.

Step 4 On the **Operation Protection** page, select **Enable** to enable operation protection.

NOTE

- When IAM users created using your account perform a critical operation, they will be prompted to choose a verification method from email, SMS, and virtual MFA device.
  - If a user is only associated with a mobile number, only SMS verification will be available.
  - If a user is only associated with an email address, only email verification will be available.
  - If the user has not bound an email address, a mobile number, or a virtual MFA device, the user needs to bind one to continue with the critical operation.
- Change your phone number or email address for verification in **My Account** on the management console. For details, see **IAM Basic Information**.

Step 5 After operation protection is enabled, when you perform a mission-critical operation, the system will protect the operation.

For example, when you delete a cluster, a verification dialog box for mission-critical operation protection is displayed. You need to select a mode to perform verification. This helps avoid risks and losses caused by misoperations.

----End

Disabling Operation Protection

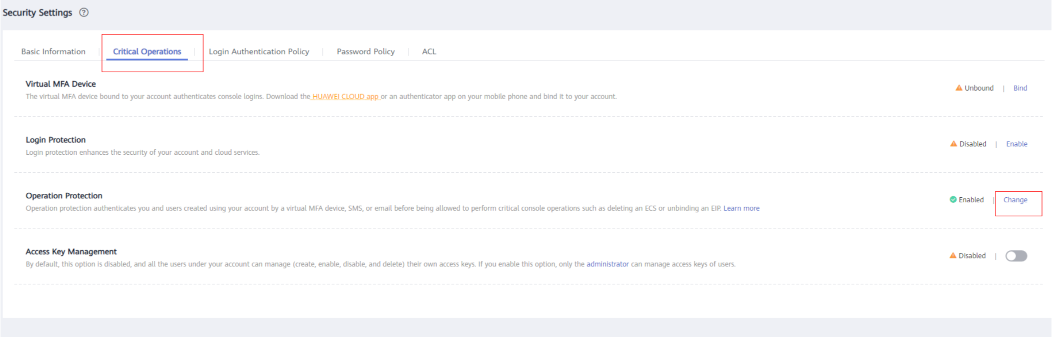
To disable operation protection, perform the following steps:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Move the cursor to the username in the upper right corner of the page and click **Security Settings** from the drop-down list.

Step 3 On the **Security Settings** page, click the **Critical Operations** tab. Click **Change** in the **Operation Protection** area.

Figure 7-4 Modifying operation protection settings



Security Settings

Basic Information | Critical Operations | Login Authentication Policy | Password Policy | ACL

Virtual MFA Device
The virtual MFA device bound to your account authenticates console logins. Download the HUAWEI CLOUD app or an authenticator app on your mobile phone and bind it to your account.
Unbound | Bind

Login Protection
Login protection enhances the security of your account and cloud services.
Disabled | Enable

Operation Protection
Operation protection authenticates you and users created using your account by a virtual MFA device, SMS, or email before being allowed to perform critical console operations such as deleting an ECS or unbinding an EIP. Learn more
Enabled | Change

Access Key Management
By default, this option is disabled, and all the users under your account can manage (create, enable, disable, and delete) their own access keys. If you enable this option, only the administrator can manage access keys of users.
Disabled |

Step 4 On the **Operation Protection** page, select **Disable** and click **OK**.

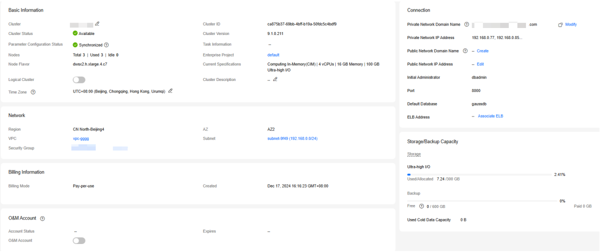
----End

8 GaussDB(DWS) Cluster Management

8.1 Viewing GaussDB(DWS) Cluster Details

Log in to the GaussDB(DWS) console. In the navigation tree on the left, click **Clusters** > **Dedicated Clusters**. In the cluster list, locate the required cluster and click its name. The **Cluster Information** page is displayed.

Figure 8-1 Cluster Details



Basic Information

Cluster	[redacted]	Cluster ID	ca875b37-69bb-4bff-b19a-50fdc5c4bdf9
Cluster Status	Available	Cluster Version	9.1.0.211
Parameter Configuration Status	Synchronized	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	dwsx2.h.xlarge.4.c7	Current Specifications	Computing In-Memory(CIM) | 4 vCPUs | 16 GB Memory | 100 GB Ultra-high I/O
Logical Cluster		Cluster Description	--
Time Zone	UTC+08:00 (Beijing, Chongqing, Hong Kong, Urumqi)		

Network

Region	CN North-Beijing4	AZ	AZ2
VPC	vpc-gggg	Subnet	subnet-9f49 (192.168.0.0/24)
Security Group	[redacted]		

Billing Information

Billing Mode	Pay-per-use	Created	Dec 17, 2024 16:16:23 GMT+08:00

O&M Account

Account Status	--	Expires	--
O&M Account			

Connection

Private Network Domain Name	[redacted].com Modify
Private Network IP Address	192.168.0.77, 192.168.0.85...
Public Network Domain Name	-- Create
Public Network IP Address	-- Edit
Initial Administrator	dbadmin
Port	8000
Default Database	gaussdb
ELB Address	-- Associate ELB

Storage/Backup Capacity

Storage

Ultra-high I/O 2.41%

Used/Allocated 7.24 /300 GB

Backup 0%

Free 0 / 600 GB Paid 0 GB

Used Cold Data Capacity 0 B

On the **Cluster Information** page, you can view the following information:

- **Basic Information**: **Table 8-1** lists the related parameters.
- **Connection**: **Table 8-2** describes the parameters.
- **Network**: **Table 8-3** lists the related parameters.
- **Billing Information**: **Table 8-4** describes the parameters.
- **Storage/Backup Capacity**: **Table 8-5** describes the parameters.
- **O&M Account**: **Table 8-6** describes the related parameters.
- **Data Encryption Information**: **Table 8-7** lists the related parameters.

NOTE

You can view this module if you enable the data encryption function when creating a cluster.

Table 8-1 Basic information

Parameter	Description
Cluster Name	Cluster name specified when a cluster is created.
Cluster Status	Cluster running status. For details, see **Cluster Status**.
Parameter Configuration Status	Parameter configuration status of a cluster.
Task Information	Cluster task status. For details, see **Cluster Task Information**.
Current Specifications	Current node specifications.
Nodes	Number of nodes in the cluster.
Logical Clusters	You can enable it as required. The **Logical Clusters** menu item will be displayed after you enable it.
Cluster ID	ID of the cluster.
Cluster Version	Cluster version information.
Node Flavor	Node flavor of the cluster.
Enterprise Project	Enterprise project to which a cluster belongs. You can click the enterprise project name to view and edit it on the console of the Enterprise Project service.
Time Zone	The cluster time zone affects the node OS, log files, and data warehouse. You can change the time zone for the node OS and log files, but not for the data warehouse databases. To change the time zone of the data warehouse databases, use the GUC parameter **timezone**. For details, see **Modifying GUC Parameters of the GaussDB(DWS) Cluster**.

Table 8-2 Connection

Parameter	Description
Private Network Domain Name	Domain name for accessing the cluster database through the internal network. The domain name corresponds to all CN IP addresses. The private network domain address is automatically generated when a cluster is created. The default naming rule is *cluster name*.dws.myhuaweicloud.com. **NOTE** ● If the cluster name does not comply with the domain name standards, the prefix of the default access domain name will be adjusted accordingly. ● Load balancing is not supported. You can click **Modify** to change the private network domain name. The access domain name contains 4 to 63 characters, which consists of letters, digits, and hyphens (-), and must start with a letter. For details, see **Managing GaussDB(DWS) Cluster Access Domain Names**.
Private Network IP Address	IP address for accessing the database in the cluster over the private network. **NOTE** ● A private IP address is automatically generated when you create a cluster. The IP address is fixed. ● The number of private IP addresses equals the number of CNs. You can log in to any node to connect to the cluster. ● If you access a fixed IP address over the internal network, all the resource pools will run on a single CN. ● If IPv6 is enabled for a cluster, both IPv4 and IPv6 private addresses will be displayed. You can use either of them as needed.
Public Network Domain Name	Name of the domain for accessing the database in the cluster over the public network. For details, see **Managing GaussDB(DWS) Cluster Access Domain Names**. **NOTE** Load balancing is not supported.
Public Network IP Address	IP address for accessing the database in the cluster over the public network. **NOTE** ● If no EIP is assigned during cluster creation and **Public Network IP Address** is empty, click **Edit** to bind an EIP to the cluster. ● If an EIP is bound during cluster creation, click **Edit** to unbind the EIP.
Initial Administrator	Database administrator specified during cluster creation. When you connect to the cluster for the first time, you need to use the initial database administrator and password to connect to the default database.

Parameter	Description
Port	Port number for accessing the cluster database through the public network or private network. The port number is specified when the cluster is created.
Default Database	Database name specified when the cluster is created. When you connect to the cluster for the first time, connect to the default database.
ELB Address	To achieve high availability and avoid single-CN failures, a new cluster needs to be bound to ELB. You are advised to use the ELB address to connect to the cluster.

Table 8-3 Network

Parameter	Description
Region	Current working zone of the cluster.
AZ	AZ selected during cluster creation
VPC	VPC selected during cluster creation. A VPC is a secure, isolated, and logical network environment. After a data warehouse cluster is created, its VPC cannot be changed. However, you can edit and modify the current VPC. You can click the VPC name to go to the VPC details page to configure it. For details about VPC operations, see **Modifying a VPC** in the *Virtual Private Cloud User Guide*.
Subnet	Subnet selected during cluster creation. A subnet provides dedicated network resources that are isolated from other networks, improving network security. After a data warehouse cluster is created, its subnet cannot be changed. However, you can edit and modify the current subnet. You can click the subnet name to go to the subnet details page to configure it. For details about subnet operations, see **Modifying a Subnet** in the *Virtual Private Cloud User Guide*.

Parameter	Description
Security Group	Security group selected during cluster creation. After a GaussDB(DWS) cluster is created, you can change the security group. You can also add, delete, or modify security group rules in the current security group. Changing the security group of a cluster may cause brief service disruption. Exercise caution when performing this operation. For better network performance, do not select more than five security groups. ● To change the security group, click **Modify** on the right of the security group name, select the security group name to be changed, and click **OK**. ● Modifying an existing security group rule: Click the security group name to go to the security group details page. For details about security group operations, see **Security Group** in the *Virtual Private Cloud User Guide*.

Table 8-4 Billing information

Parameter	Description
Billing Mode	Billing mode. ● Pay-per-use ● Yearly/Monthly
Created	Time when a pay-per-use or yearly/monthly cluster is created.
Order (for yearly/monthly billing)	Order number of a yearly/monthly cluster.
Expiration Date (for yearly/monthly billing)	Expiration time of a yearly/monthly cluster.

Table 8-5 Storage/Backup capacity

Parameter	Description
Storage	The storage class **Ultra-high I/O** and the storage space usage are displayed. **NOTE** ● The used storage capacity does not include data on OBS foreign tables. It includes only GaussDB(DWS) data, including files, logs, snapshots, and indexes. ● The available storage space is half of the actual disk capacity.
Backup	The space in use, free space, and charged space of the cluster are displayed.
Cold Data Used Capacity (storage-compute decoupling)	OBS hot data capacity used by storage-compute decoupled clusters.
Used Cold Partition Data Capacity	OBS capacity used by cold data. **NOTE** OBS capacity usage. It is synchronized every hour.
Used Capacity of OBS Foreign Tables	OBS capacity used by the foreign tables of the default OBS server of the cluster: **default_obs_foreign_table_server**. **NOTE** OBS capacity usage. It is synchronized every hour.

Table 8-6 O&M account

Parameter	Description
O&M Account	Specifies whether to enable the cluster O&M account. Check the created O&M account. Its name format is **om_user_***First_eight_numbers_in_cluster_ID*. The **gs_role_analyze_any**, **gs_role_vacuum_any**, **gs_role_read_all_stats**, and **gs_role_signal_backend** roles will be assigned to the account. For details, see **Preset Roles**.
Account Status	Displays the status of the current cluster O&M account, which can be **Normal** or **Expired**.
Expires	Indicates the expiration time of the O&M account of the current cluster.

Parameter	Description
Extend by 8h	- For a normal account, its validity period is extended to 8 hours later than its expiration time. - For an expired account, its validity period is extended to 8 hours later than the current time.

Table 8-7 Data encryption information

Parameter	Description
Key Name	Indicates the database encryption key of the cluster when **Encrypt DataStore** is enabled.
Last Key Rotation Time	Indicates the time when the last encryption key is rotated when **Encrypt DataStore** is enabled.

Changing a Cluster Name

You can change the name of a created GaussDB(DWS) cluster.

After the cluster name is changed, the names of all nodes in the current cluster are changed accordingly.

NOTE

- If the cluster name cannot be changed on the console, contact technical support to upgrade the console.
- If the cluster name fails to be changed, the cluster functions are not affected. You can contact technical support rectify the fault.

Constraints

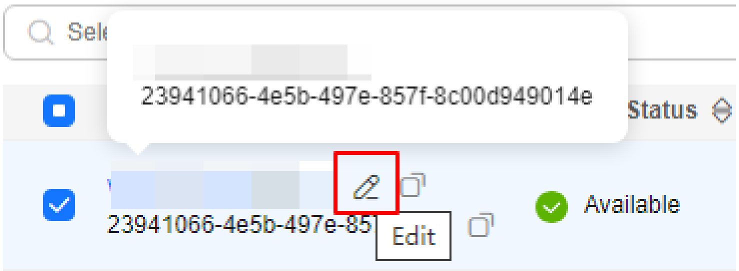
If the cluster is in the **Unavailable** status or is performing other tasks, the cluster name cannot be changed. You can change the cluster name only after the cluster status changes to **Available** or the running tasks are complete.

Method 1:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the cluster list, click the modification icon next to a cluster name to modify the cluster.

Figure 8-2 Changing the name of a cluster in the cluster list



Step 3 In the displayed dialog box, enter a new cluster name.

Step 4 Confirm the information and click **OK**.

----End

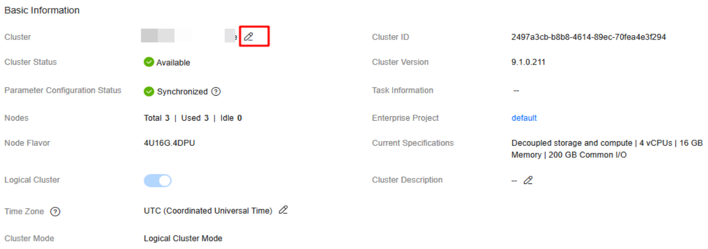
Method 2:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the cluster list, click the name of a cluster.

Step 3 On the displayed **Cluster Details** page, click the modification icon next to the cluster name in the **Basic Information** area.

Figure 8-3 Changing the cluster name on the cluster details page



Basic Information

Cluster	[redacted]	Cluster ID	2497a3cb-b8b8-4614-89ec-70fea4e3f294
Cluster Status	✓ Available	Cluster Version	9.1.0.211
Parameter Configuration Status	✓ Synchronized ?	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	4U16G.4DPU	Current Specifications	Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O
Logical Cluster		Cluster Description	--
Time Zone ?	UTC (Coordinated Universal Time)		
Cluster Mode	Logical Cluster Mode		

Step 4 After confirming that the information is correct, click **OK** to deliver the cluster modification task. After the task is complete, the cluster name is changed.

----End

8.2 Checking the GaussDB(DWS) Cluster Status

On the **Clusters** > **Dedicated Clusters** page of the GaussDB(DWS) console, you can view the general information about a cluster in the cluster list, such as the cluster status, task information, recent events, and node flavor.

Querying General Information of a Cluster

Log in to the GaussDB(DWS) console. In the navigation pane, click **Clusters** > **Dedicated Clusters**. The cluster list displays all clusters. If there are a large number of clusters, you can turn pages to view the clusters in any status.

In the upper part of the cluster list, click the search box and search for the required cluster based on the filter criteria (cluster name, cluster status, task information, node specifications, billing mode, recent events, and enterprise project). Click ↻ to refresh the cluster list and billing mode. You can also click **Search by Tag** to search for clusters based on cluster tags. For details, see **Searching for Clusters Based on Tags**.

Clusters are listed in chronological order by default, with the most recent clusters displayed at the top. **Table 8-8** describes the cluster list parameters.

Table 8-8 Cluster list parameters

Parameter	Description
Cluster Name	Cluster name specified when a cluster is created. **NOTE** If the cluster name cannot be changed on the console, contact technical support.
Cluster Status	Cluster running status. For details, see **Cluster Status**.
Task Information	Cluster task status. For details, see **Cluster Task Information**.
Node Flavor	Node flavors of clusters. For details, see **GaussDB(DWS) Pricing**.
Billing Mode	Cluster billing mode. ● In pay-per-use mode, the cluster creation time is displayed. ● In yearly/monthly mode, the cluster expiration time is displayed. For details, see **Yearly/Monthly**.
Recent Events	Number of recent events in a cluster. You can click the number to view event details.
Enterprise Project	Enterprise project to which a cluster belongs.

Parameter	Description
Operation	- **Log In**: For details, see **Using the SQL Editor to Connect to a Cluster**. - **Monitoring Panel**: For details, see **Viewing GaussDB(DWS) Cluster Monitoring Information on the Monitoring Panel (DMS)**. - **More** – **View Metric**: For details, see **Viewing GaussDB(DWS) Cluster Monitoring Information on Cloud Eye**. – **Restart**: Click **Restart** to restart a cluster. For details, see **Starting, Stopping, and Deleting a GaussDB(DWS) Cluster**. – **Scale Out**: For details, see **Scaling Out a Cluster**. – **Change all specifications**: For details, see **Changing All Specifications**. – **Scale In**: For details, see **Scaling In a Cluster**. – **Redistribute**: For details, see **Redistributing Data**. – **Expand Disk Capacity**: For details, see **Disk Capacity Expansion of an EVS Cluster**. – **Reset Password**: For details, see **Resetting the Password the GaussDB(DWS) Database Administrator**. – **Create Snapshot**: For details, see **Manual Snapshots**. – **Delete**: Click **Delete** to delete a cluster. For details, see **Deleting a Cluster**. – **Change node flavor**: For details, see **Using the Elastic Specification Change**. – **Manage CN**: For details, see **Adding or Deleting a CN in a GaussDB(DWS) Cluster**.

Cluster Status

Table 8-9 Cluster status description

Status	Description
Available	Indicates that the cluster runs properly.

Status	Description
Read-only	A cluster goes into this state when the disk usage of the cluster or a single node in the cluster is greater than 90%. The cluster can still work in this state but supports only query operations. Write operations are not supported. When the cluster status becomes read-only, remove the status by referring to **Removing the Read-only Status**. If the status cannot be removed, contact technical support engineers. After the read-only status is canceled for the cluster, you are advised to perform the following operations: ● Use the SQL client tool to connect to the database as the administrator and run the following command to periodically clear and reclaim the storage space: `VACUUM FULL;` After you delete data stored in GaussDB(DWS) data warehouses, dirty data may be generated possibly because the disk space is not released. This results in disk space waste. It is recommended that the storage space be cleared periodically. ● You are advised to check the disk capacity and analyze whether the existing cluster specifications meet service requirements. If not, expand the cluster capacity. For details, see **Scaling Out a Cluster**.
Unbalanced	If the role of a GTM or DN in the cluster is different from the initial role, the cluster is in the **Unbalanced** state. In the **Unbalanced** state, the number of primary instances on some nodes increases. As a result, the load pressure is high. In this case, the cluster is normal, but the overall performance is not as good as that in a balanced state. You are advised to switch a cluster to the **Available** state during off-peak hours. For details, see **Performing a Primary/Standby Switchback**.
Redistributing	A cluster goes into this state when it detects that the service data on the original nodes is significantly larger than that on the new node after a new node is added to the cluster. In this case, the system automatically redistributes data on all nodes. The cluster can still work in this state.
Redistribution failed	A cluster goes into this state when data redistribution fails, but no data loss occurs. The cluster can still work in this state. You are advised to contact technical support.
Degraded	A cluster goes into this state when some nodes in the cluster are faulty, but the whole cluster runs properly. You are advised to contact technical support.
Unavailable	A cluster goes into this state when it cannot provide database services. You are advised to contact technical support.
Creating	A cluster goes into this state when it is being created.
Creation failed	A cluster goes into this state when it fails to be created.

Status	Description
Creating, restoring	A cluster goes into this state when it is being restored from a snapshot.
Deleting	A cluster goes into this state when it is being deleted.
Frozen for legal reasons	Indicates that the cluster is frozen for legal reasons. In this case, the cluster cannot be deleted or unsubscribed, and its name cannot be changed.
Frozen	Indicates that the cluster is frozen (excluding legal reasons). In this case, the cluster name cannot be changed. If your account balance is insufficient and fee deduction fails, the retention period starts. During the retention period, the service resources will be frozen and cannot be used, but resources and data are reserved. To unfreeze the clusters, you need to top up your account to ensure that the account balance is not **0**. For details, see **How Do I Renew My Service?**
To be restarted	This status indicates that GUC parameters have been modified in the cluster and the modification can take effect only after the cluster is restarted. Before the cluster is restarted, some O&M operations cannot be performed. After you manually restart the cluster, the GUC parameter takes effect and the cluster status changes to **Available**.
Stopped	Indicates that the cluster is stopped.

Cluster Task Information

Table 8-10 Task information description

Status	Description
Creating snapshot	Indicates that a snapshot is being created in the cluster.
Snapshot creation failed	Indicates that a snapshot fails to be created.
Observing	Indicates that the cluster is to be submitted after the automatic upgrade.
Configuring	Indicates that the system is storing modifications of cluster parameters.
Restarting	Indicates that a cluster is being restarted.
Restart failed	Indicates that a cluster fails to be restarted.

Status	Description
Converting to encryption cluster	Indicates that the cluster is being converted to an encrypted cluster.
Encryption cluster conversion failed	Indicates that the cluster fails to be encrypted.
Scaling out	Indicates that a cluster is being scaled out.
Scale-out failed	Indicates that a cluster fails to be scaled out.
Expanding disk capacity	Indicates that disk capacity is being expanded.
Disk expansion failed	Indicates that disk capacity fails to be expanded.
Associating ELB	Indicates that ELB is being associated.
Failed to associate ELB	Indicates that ELB fails to be associated.
Disassociating ELB	Indicates that ELB is being disassociated.
Failed to disassociate ELB	Indicates that ELB fails to be disassociated.
Checking scale-in	The service is checking whether a cluster can be scaled in.
Scale-in check failed	A cluster does not meet the scale-in requirements. For example: ● The value of **default_storage_nodegroup** is not **installation**. ● In the cluster database, **data_redis** is a reserved redistribution schema, but the schema contains user tables. ● The cluster disk space does not meet the scale-in requirements. For details, see **Scaling In a Cluster**.
Scaling in	A cluster is being scaled in.
Scale-in failed	The cluster scale-in fails. You need to manually scale in the cluster again as soon as possible, or your services will be affected.
Switching back	The primary/standby relationship of a cluster is being restored.

Status	Description
Switchback failed	The primary/standby relationship of a cluster fails to be restored. Possible causes are as follows. For details, see **Management Plane Error Code Reference**. ● Redo operations are being performed on DNs. Wait until the operations are completed and try again. ● Failed to query DN redo information. Check tenant logs to identify the failure cause. ● Primary/standby catchup is in progress. Wait until it is completed and try again. ● Failed to query primary/standby catchup information. Check tenant logs to identify the failure cause. ● Primary/standby catchup failed. Contact technical support or try again later. Check tenant logs to identify the failure cause. ● The cluster is abnormal.
Changing node flavor	The cluster is being scaled.
Node flavor change failed	All specifications change failed
Waiting for payment	The order for changing a pay-per-use cluster to a yearly/monthly cluster has not been paid. After the order is paid or canceled, the status will change.
Changing all specifications	All the specifications of the cluster being changed.
All specifications change failed	Specifications change failed because of insufficient quotas or permissions, or abnormal cluster status.
Maintaining	A maintenance change operation, such as cluster upgrade or plug-in upgrade, is being performed on the cluster.
Maintain_failure	A cluster fails to be restarted.
Stopping	Indicates that the cluster is being stopped.
Starting	Indicates that the cluster is being started.
Inspecting	Indicates that the cluster is being inspected before the change.
Inspection failed	Indicates that the cluster inspection fails.

Yearly/Monthly

Table 8-11 Yearly/Monthly billing mode description

Status	Description
Expire in XX	Remaining duration of a yearly/monthly cluster. You can renew the subscription, change the payment mode to pay-per-use, and unsubscribe from the subscription.
Expired. XX until frozen	A yearly/monthly cluster enters the grace period if it is not renewed upon expiration. In the grace period, a yearly/monthly cluster is still available and can be renewed, but it cannot be changed to pay-per-use or unsubscribed from.
Frozen. XX until deletion	The grace period of a yearly/monthly cluster ends and the cluster enters the retention period. The cluster can be renewed, but it cannot be changed to pay-per-use or unsubscribed from.
Change to pay-per-use after XX	After the validity period of a yearly/monthly cluster expires, the cluster becomes a pay-per-use cluster. The cluster can be renewed or unsubscribed from, but it cannot be changed to pay-per-use.
Frozen (due to violation)	Frozen by public security institutions. The cluster can be renewed, but it cannot be changed to pay-per-use or unsubscribed from. The cluster still incurs charges during the freezing period.
Frozen (due to violation) and will be deleted after XX	Frozen due to violations of regulations. The cluster can be renewed, but it cannot be changed to pay-per-use or unsubscribed from. The cluster still incurs charges during the freezing period.

8.3 Viewing the GaussDB(DWS) Cluster Topology

Overview

A topology shows all the nodes in a cluster. You can check the node statuses, processes, and IP addresses.

NOTE

- You can check the topology structure and node processes.
- Only cluster versions 8.0.0 and later can display the topology structure. Only cluster versions 8.2.0 and later can display node processes.

Viewing the Cluster Topology

Step 1 Log in to the GaussDB(DWS) console.

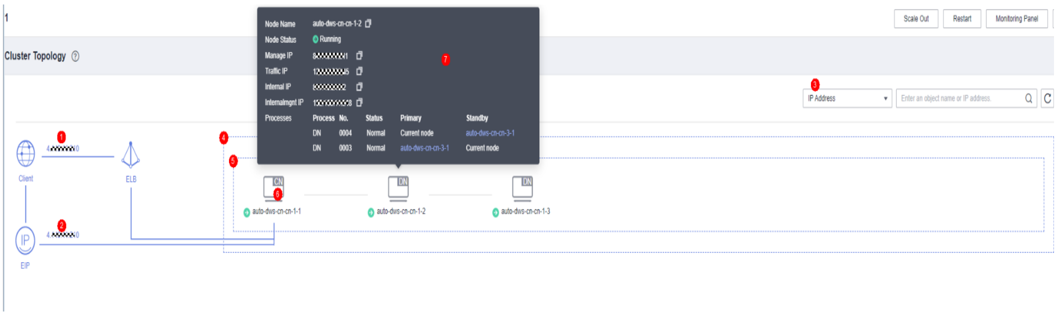
Step 2 In the cluster list, click the name of a cluster.

Step 3 On the **Cluster Details** page, click the **Cluster Topology** tab.

Step 4 In the pper right corner of the page, you can select **IP Address** or **Node Name**. After entering the IP address or node name in the search box, you can view the location of the IP address or node name in the cluster topology.

----End

Topology Overview



Node Name auto-dws-cn-cn-1-2
Node Status Running
Manage IP [redacted]
Traffic IP [redacted]
Internal IP [redacted]
Internalmgnt IP [redacted]

Processes

Process	No.	Status	Primary	Standby
DN	0004	Normal	Current node	auto-dws-cn-cn-3-1
DN	0003	Normal	auto-dws-cn-cn-3-1	Current node

This figure shows a topology. The elements marked in the figure are as follows:

1. Public IP address of the ELB bound to the cluster. If no public IP addresses are bound to the ELB, the service address is displayed.
2. EIP bound to the cluster.
3. Search category. You can perform exact search by IP address or node name.
4. Rings in the cluster.
5. This box represents a ring, with each line and icon indicating a node within the ring. If there are at least three cluster rings created, you can view the distributed deployment of CNs.
6. A node. The type of the node is displayed in the upper right corner of the icon. Currently, the type can only be CN or DN. If there is a CN process on the node, **CN** is displayed. If there are no CN processes on the node, **DN** is displayed.
7. Node details, including the node name, status, IP addresses, and task process. Node details are displayed when you hover your cursor over a node icon.

Terms in the Topology View

Table 8-12 Cluster structure description

Name	Description	Usage
ELB	Elastic Load Balance (ELB) automatically distributes incoming traffic across multiple backend servers based on listening rules you configure.	If the private IP address or EIP of a CN is used to connect to a GaussDB(DWS) cluster, the failure of this CN will lead to a cluster connection failure. If a private or public domain name is used for connection, the DNS service randomly selects a private IP address or EIP for each client. This cannot balance loads or avoid single-CN failures. ELB is used to solve these problems. For details, see **Binding and Unbinding ELBs for a GaussDB(DWS) Cluster**.
EIP	The Elastic IP (EIP) service provides static public IP addresses and scalable bandwidths that enable your cloud resources to communicate with the Internet.	EIPs can be bound to or unbound from ECSs, BMSs, virtual IP addresses, load balancers, and NAT gateways.
Ring	A security ring is used for isolating faulty servers. A fault in a ring does not affect servers outside the ring.	Data on a DN has multiple copies in a ring, and will not be lost even if the DN server is faulty. For example, if Server1 in a ring is faulty, the standby DN1 on Server2, the standby DN2 on Server3, and the standby DN3 on Server3 are still running. The loads of servers in a ring are still balanced. A cluster can run properly as long as the number of faulty servers does not exceed the number of rings. **NOTE** The ring is the minimum unit for a scale-out. When you scale out a cluster, the added nodes must be a multiple of the ring quantity.

Table 8-13 Node IP addresses

Name	Description	Usage
Manage IP	IP address used by a data warehouse node to communicate with the management plane	It is used by the management plane to deliver commands, and used by the node to report node status and monitoring information.
Traffic IP	IP address of a data warehouse node for external access.	This IP address can be bound to an EIP or ELB, or directly connect to a VPC.
Internal IP	IP address used for communication inside a data warehouse cluster.	-
Internalmgnt IP	IP address used by nodes to send internal management commands in a data warehouse cluster.	-

Table 8-14 Node processes

Name	Description	Usage
CMS	A Cluster Manager (CM) manages and monitors the running status of functional units and physical resources in the distributed system, ensuring system stability. CM Server (CMS) is a module of CM.	A CM consists of CM Agent, OM Monitor, and CM Server. ● CM Agent monitors the running status of primary and standby GTMs, CNs, and primary and standby DNs on the host, and reports the status to CM Server. In addition, it executes the arbitration instruction delivered by CM Server. A CM Agent process runs on each server. ● OM Monitor monitors scheduled tasks of CM Agent and restarts CM Agent when CM Agent stops. If CM Agent cannot be restarted, the server will be unavailable. In this case, you need to manually rectify this fault. **NOTE** A CM Agent restart fails probably because of lack of system resources, which rarely happens. ● CM Server checks whether the current system is normal according to the instance status reported by CM Agent. In the case of exceptions, CM Server delivers recovery commands to CM Agent. GaussDB(DWS) deploys CM Server in primary/standby mode to ensure system HA. CM Agent

Name	Description	Usage
		connects to the primary CM Server. If the primary CM Server is faulty, the standby CM Server is promoted to primary to prevent single-CM faults.
GTM	A Global Transaction Manager (GTM) generates and maintains the globally unique information, such as the transaction ID, transaction snapshot, and timestamp.	A cluster includes only one pair of GTMs: one primary and one standby GTM.
CN	A Coordinator (CN) receives access requests from applications, and returns execution results to the client; splits tasks and allocates task fragments to different DNs for parallel processing.	CNs in a cluster have equivalent roles and return the same result for the same DML statement. Load balancers can be added between CNs and applications to ensure that CNs are transparent to applications. If a CN is faulty, the load balancer connects its applications to another CN. CNs need to connect to each other in the distributed transaction architecture. To reduce heavy load caused by excessive threads on GTMs, no more than 10 CNs should be configured in a cluster.

Name	Description	Usage
CCN	Central Coordinator (CCN)	GaussDB(DWS) handles the global resource load in a cluster using the Central Coordinator (CCN) for adaptive dynamic load management. When the cluster is started for the first time, the CM selects the CN with the smallest ID as the CCN. If the CCN is faulty, CM replaces it with a new one.
DN	A Data Node (DN) stores data in row-store, column-store, or hybrid mode, executes data query tasks, and returns execution results to CNs.	There are multiple DNs in the cluster. Each DN stores part of data. If DNs are not deployed in primary/standby mode and a DN is faulty, data on the DN will be inaccessible.

8.4 Managing GaussDB(DWS) Cluster Connections

8.4.1 Managing GaussDB(DWS) Cluster Access Domain Names

Overview

A domain name is a string of characters separated by dots to identify the location of a computer or a computer group on the Internet, for example, www.example.com. You can enter a domain name in the address box of the web browser to access a website or web application.

On GaussDB(DWS), you can access clusters using the private network domain name or the public network domain name.

Private network domain name: Name of the domain for accessing the database in the cluster through the private network. The private network domain name is automatically generated when you create a cluster. The default naming rule is *cluster name*.dws.myhuaweicloud.com. If the cluster name does not comply with the domain name standards, the prefix of the default access domain name will be adjusted accordingly.

Public network domain name: Name of the domain for accessing the database in the cluster through the public network. If a cluster is not bound to an EIP, it cannot be accessed using the public network domain name. If you bind an EIP during cluster creation, the public network domain name is automatically generated. The default naming rule is *cluster name*.dws.huaweiclouds.com.

NOTE

Neither public nor private domain names support load balancing. To use load balancing, see **Configuring JDBC to Connect to a Cluster (Load Balancing Mode)**.

After a cluster is created, you can set private and public domain names for accessing the cluster as required. The operations are as follows:

- **Modifying a Private Network Domain Name**
- **Creating a Public Network Domain Name**
- **Modifying a Public Network Domain Name**
- **Releasing a Public Network Domain Name**

Modifying a Private Network Domain Name

The private network domain name is automatically generated during cluster creation. After the cluster is created, you can modify the default domain name based on site requirements.

To modify the private network domain name, perform the following steps:

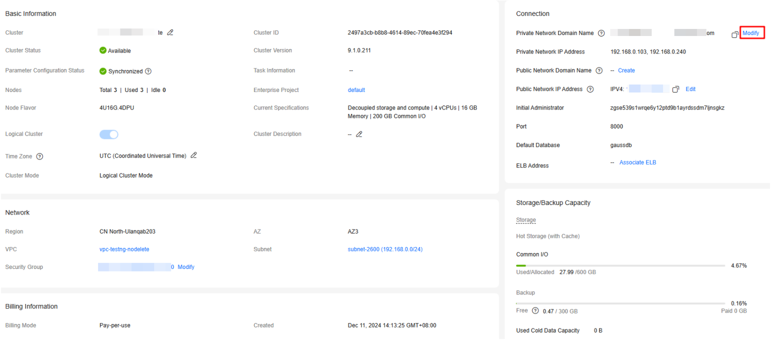
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 In the **Connection** area, click **Modify** next to the automatically generated **Private Network Domain Name**.

Figure 8-4 Modifying a private network domain name



Basic Information

Cluster	[redacted] te	Cluster ID	2497a3cb-b8b8-4614-89ec-70fea4e3f294
Cluster Status	Available	Cluster Version	9.1.0.211
Parameter Configuration Status	Synchronized	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	4U16G.4DPU	Current Specifications	Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O
Logical Cluster		Cluster Description	--
Time Zone	UTC (Coordinated Universal Time)		
Cluster Mode	Logical Cluster Mode		

Network

Region	CN North-Ulanqab203	AZ	AZ3
VPC	vpc-testng-nodelete	Subnet	subnet-2600 (192.168.0.0/24)
Security Group	[redacted]0 Modify		

Billing Information

Billing Mode	Pay-per-use	Created	Dec 11, 2024 14:13:25 GMT+08:00

Connection

Private Network Domain Name	[redacted] om Modify
Private Network IP Address	192.168.0.103, 192.168.0.240
Public Network Domain Name	-- Create
Public Network IP Address	IPV4: [redacted] Edit
Initial Administrator	zgse539s1wrqe6y12ptd9b1ayrdssdm7ljnsgkz
Port	8000
Default Database	gaussdb
ELB Address	-- Associate ELB

Storage/Backup Capacity

Storage

Hot Storage (with Cache)

Common I/O 4.67%
Used/Allocated 27.99 /600 GB

Backup 0.16%
Free 0.47 / 300 GB Paid 0 GB

Used Cold Data Capacity 0 B

Step 5 In the **Modify Private Network Domain Name** dialog box, enter the target domain name and click **OK**.

The private network domain name contains 4 to 63 characters, which consists of letters, digits, and hyphens (-) and must start with a letter.

After the domain name is modified, click copy button next to the private network domain name to copy it.

----End

Creating a Public Network Domain Name

A cluster is not bound to an EIP by default during cluster creation. That is, cluster access using the public network is disabled. After a cluster is created, if you want to access it over the public network, bind an EIP to the cluster and create a public network domain name.

NOTE

By default, only Huawei Cloud accounts or users with **Security Administrator** permissions can query and create agencies. By default, the IAM users in those accounts cannot query or create agencies. When the users use the EIP, the system makes the binding function unavailable. Contact a user with the **DWS Administrator** permissions to authorize the agency on the current page.

To create a public network domain name, perform the following steps:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 In the **Connection** area, **Public Network Domain Name** and **Public Network IP Address** are empty. Click **Edit** to bind the cluster with an EIP.

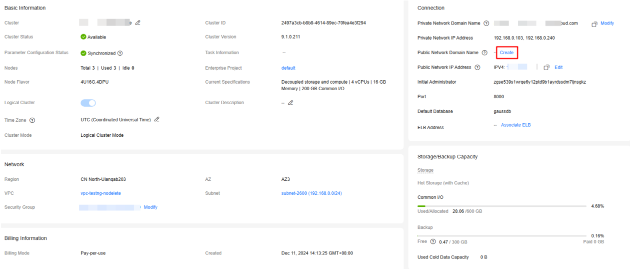
Step 5 In the **Edit Elastic IP** dialog box, select an EIP from the drop-down list to bind it to a specified CN.

If no available EIPs are displayed, click **View EIP** to go to the **Elastic IP** page and create an EIP that satisfies your needs. After the new EIP is created, click the refresh icon next to the drop-down list. The newly created EIP will be displayed in the **EIP** drop-down list.

After the EIP is bound successfully, the specific public network IP address is displayed in the **Connection** area.

Step 6 In the **Connection** area, click **Create** next to **Public Network Domain Name** to create a public network domain name for the cluster.

Figure 8-5 Creating a public network domain name



Basic Information

Cluster	[redacted]	Cluster ID	2497a3cb-b8b8-4614-89ec-70fea4e3f294
Cluster Status	Available	Cluster Version	9.1.0.211
Parameter Configuration Status	Synchronized	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	4U16G.4DPU	Current Specifications	Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O
Logical Cluster		Cluster Description	--
Time Zone	UTC (Coordinated Universal Time)		
Cluster Mode	Logical Cluster Mode		

Network

Region	CN North-Ulanqab203	AZ	AZ3
VPC	vpc-testng-nodelete	Subnet	subnet-2600 (192.168.0.0/24)
Security Group	[redacted] Modify		

Billing Information

Billing Mode	Pay-per-use	Created	Dec 11, 2024 14:13:25 GMT+08:00

Connection

Private Network Domain Name	[redacted] Modify
Private Network IP Address	192.168.0.103, 192.168.0.240
Public Network Domain Name	-- Create
Public Network IP Address	IPV4: [redacted] Edit
Initial Administrator	zgse539s1wrqe6y12ptd9b1ayrdssdm7ljnsgkz
Port	8000
Default Database	gaussdb
ELB Address	-- Associate ELB

Storage/Backup Capacity

Storage

Hot Storage (with Cache)

Common I/O

4.68%

Used/Allocated 28.06 /600 GB

Backup

0.16%

Free 0.47 / 300 GB Paid 0 GB

Used Cold Data Capacity 0 B

Step 7 In the **Apply for Public Network Domain Name** dialog box, enter the target domain name and click **OK**.

The public network domain name contains 4 to 63 characters, which consists of letters, digits, and hyphens (-) and must start with a letter.

The specific public network domain name is displayed in the **Connection** area after being created. Click copy button ⧉ to copy the public network domain name.

----End

Modifying a Public Network Domain Name

If you bind an EIP during cluster creation, the public network domain name is automatically generated. After a cluster is created, you can modify the public network domain name as required.

To modify the public network domain name, perform the following steps:

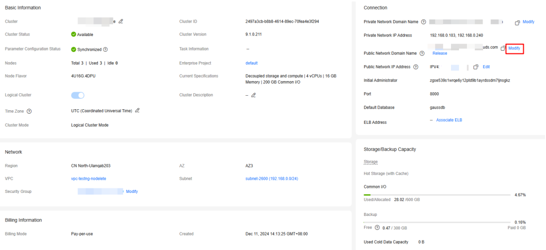
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 Click **Modify** next to the **Public Network Domain Name** in the **Connection** area.

Figure 8-6 Modifying a public network domain name



Basic Information

Cluster [redacted]e

Cluster Status Available

Parameter Configuration Status Synchronized

Nodes Total 3 | Used 3 | Idle 0

Node Flavor 4U16G.4DPU

Logical Cluster

Time Zone UTC (Coordinated Universal Time)

Cluster Mode Logical Cluster Mode

Cluster ID 2497a3cb-b8b8-4614-89ec-70fea4e3f294

Cluster Version 9.1.0.211

Task Information --

Enterprise Project default

Current Specifications Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O

Cluster Description --

Network

Region CN North-Ulanqab203

VPC vpc-testng-nodelete

Security Group [redacted] Modify

AZ AZ3

Subnet subnet-2600 (192.168.0.0/24)

Billing Information

Billing Mode Pay-per-use

Created Dec 11, 2024 14:13:25 GMT+08:00

Connection

Private Network Domain Name [redacted] Modify

Private Network IP Address 192.168.0.103, 192.168.0.240

Public Network Domain Name [redacted]uds.com Modify Release

Public Network IP Address IPV4: [redacted] Edit

Initial Administrator zgse539s1wrqe6y12ptd9b1ayrdssdm7ljnsgkz

Port 8000

Default Database gaussdb

ELB Address -- Associate ELB

Storage/Backup Capacity

Storage

Hot Storage (with Cache)

Common I/O 4.67%

Used/Allocated 28.02 /600 GB

Backup 0.16%

Free 0.47 / 300 GB Paid 0 GB

Used Cold Data Capacity 0 B

Step 5 In the **Modify Public Network Domain Name** dialog box, enter the target domain name and click **OK**.

----End

Releasing a Public Network Domain Name

After a cluster is created, you can release unnecessary public network domain names.

To do so, perform the following steps:

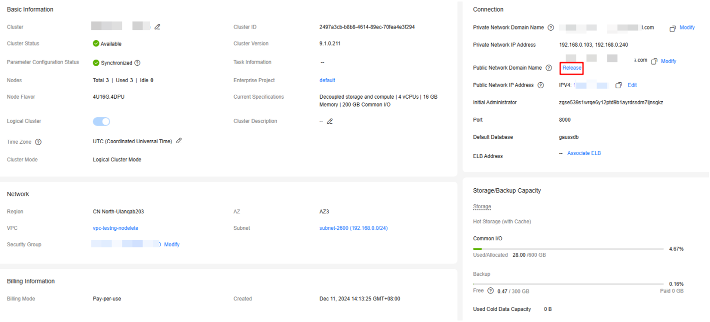
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 Click **Release** next to the **Public Network Domain Name** in the **Connection** area.

Figure 8-7 Releasing a public network domain name



Basic Information

Cluster	[redacted]	Cluster ID	2497a3cb-b8b8-4614-89ec-70fea4e3f294
Cluster Status	Available	Cluster Version	9.1.0.211
Parameter Configuration Status	Synchronized	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	4U16G.4DPU	Current Specifications	Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O
Logical Cluster		Cluster Description	--
Time Zone	UTC (Coordinated Universal Time)		
Cluster Mode	Logical Cluster Mode		

Network

Region	CN North-Ulanqab203	AZ	AZ3
VPC	vpc-testng-nodelete	Subnet	subnet-2600 (192.168.0.0/24)
Security Group	[redacted] Modify		

Billing Information

Billing Mode	Pay-per-use	Created	Dec 11, 2024 14:13:25 GMT+08:00

Connection

Private Network Domain Name	[redacted] l.com Modify
Private Network IP Address	192.168.0.103, 192.168.0.240
Public Network Domain Name	[redacted] s.com Modify Release
Public Network IP Address	IPV4: [redacted] Edit
Initial Administrator	zgse539s1wrqe6y12ptd9b1ayrdssdm7ljnsgkz
Port	8000
Default Database	gaussdb
ELB Address	-- Associate ELB

Storage/Backup Capacity

Storage

Hot Storage (with Cache)

Common I/O 4.67%

Used/Allocated 28.00 /600 GB

Backup 0.16%

Free 0.47 / 300 GB Paid 0 GB

Used Cold Data Capacity 0 B

Step 5 In the **Release Domain Name** dialog box, click **Yes**.

----End

8.4.2 Binding and Unbinding ELBs for a GaussDB(DWS) Cluster

Overview

If the private IP address or EIP of a CN is used to connect to a cluster, the failure of this CN will lead to cluster connection failure. If a private or public domain name is used for connection, the DNS service randomly selects a private IP address or EIP for each client. This cannot balance loads or avoid single-CN failures. ELB is used to solve these problems.

An ELB distributes access traffic to multiple ECSs for traffic control based on forwarding policies. It improves the fault tolerance capability of application programs. For details, see the ***Elastic Load Balance User Guide***.

With ELB health checks, CN requests of a cluster can be quickly forwarded to normal CNs. If a CN is faulty, the workload can be immediately shifted to a healthy node, minimizing cluster access faults. Currently, ELBs can be bound in the same VPC or across VPCs.

NOTE

- This feature is supported only in cluster version 8.1.1.200 or later.
- To ensure load balancing and high availability and prevent service interruption, ensure created clusters are to an ELB.
- When you bind a cluster to ELB across VPCs, you can bind it to a dedicated load balancer.
- ELB does not support cross-database access.

Constraints and Limitations

- To bind an ELB to a GaussDB(DWS) cluster, the ELB must be in the same region, VPC, and enterprise project as the cluster.

- Only dedicated load balancers can be bound to GaussDB(DWS).

NOTICE

Load balancing is not supported in regions where the dedicated load balancer is not available. You can check whether dedicated load balancers are supported on the ELB console.

- The ELB to be associated must use TCP and has a private IP address.
- When creating an ELB instance, determine its specifications based on your service access traffic. You are advised to select the maximum specifications. On the GaussDB(DWS) console, you can bind to an ELB instance but cannot change its specifications.
- You only need to create a load balancer if you want to use ELB. GaussDB(DWS) automatically creates the required ELB listeners and backend server groups.
- When creating a load balancer, ensure that the listeners do not use the same port as the database. Otherwise, ELB cannot be associated.
- When you associate ELB, the **ROUND_ROBIN** policy is set by default. In addition, the health check interval is set to 10 seconds, the timeout duration is set to 50 seconds, and the number of maximum retries is set to 3. Exercise caution when you modify these ELB parameters.
- When you bind a cluster to ELB across VPCs, you can only bind it to a dedicated load balancer.
- Before you bind a cluster to ELB across VPCs, ensure that the subnet segment of the cluster VPC is different from that of the ELB VPC.
- When you disassociate ELB from a cluster, related cluster information is cleared on GaussDB(DWS) but the load balancer is not deleted. Delete the load balancer in time to prevent unnecessary costs.
- If you need to access the ELB cluster using a public IP address or domain name, bind an EIP or domain name on the ELB management console.
- If the cluster is an IPv4 cluster, only IPv4 ELBs can be bound. If the cluster is an IPv6 dual-stack cluster, only IPv6 dual-stack ELBs can be bound.

Associating ELB

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

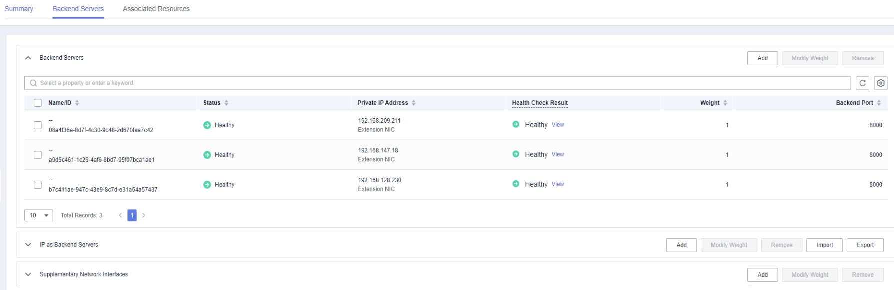
Step 4 On the **Basic Information** page that is displayed, click **Associate ELB** and select the ELB name. If no load balancer exists, create one on the ELB management console. Then refresh the GaussDB(DWS) page and associate ELB with the cluster.

NOTE

By default, the ELB in the VPC of the cluster is selected for GaussDB(DWS). If you select **Bind to ELB in another VPC**, the list of ELBs in other VPCs will be displayed for you to choose from. Before binding your cluster to an ELB across VPCs, ensure the cluster VPC has been connected to the ELB VPC. For details, see **Prerequisites for Binding an ELB to a Cluster Across VPCs**.

Step 5 After the request is delivered, go back to the **Clusters** page. Task information **Associating ELB** of the cluster is displayed. The process takes some time.

Step 6 Log in to the ELB management console, choose **Elastic Load Balance** > **Load Balancers**, click the name of the bound load balancer, switch to the **Backend Server Groups** tab, and check whether the cluster CNs are associated with the load balancer.



☐ Name/ID	Status	Private IP Address	Health Check Result	Weight	Backend Port
☐ -- 08a4f36e-8d7f-4c30-9c48-2d670fea7c42	Healthy	192.168.209.211 Extension NIC	Healthy View	1	8000
☐ -- a9d5c461-1c26-4af6-8bd7-95f07bca1ae1	Healthy	192.168.147.18 Extension NIC	Healthy View	1	8000
☐ -- b7c411ae-947c-43e9-8c7d-e31a54a57437	Healthy	192.168.128.230 Extension NIC	Healthy View	1	8000

10 Total Records: 3 1

NOTE

If the health check result indicates that the ELB backend nodes are deleted, ignore the problem.

Step 7 In the **Basic Information** area of the **Cluster Information** page, check the **ELB Address**, which is used for connecting to the cluster.

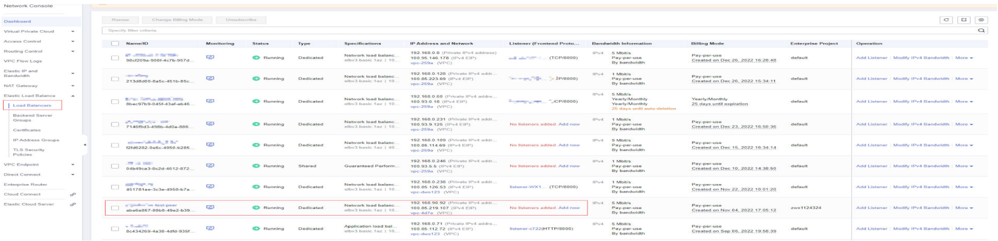
----End

Prerequisites for Binding an ELB to a Cluster Across VPCs

Enabling ELB for a cross-VPC backend server

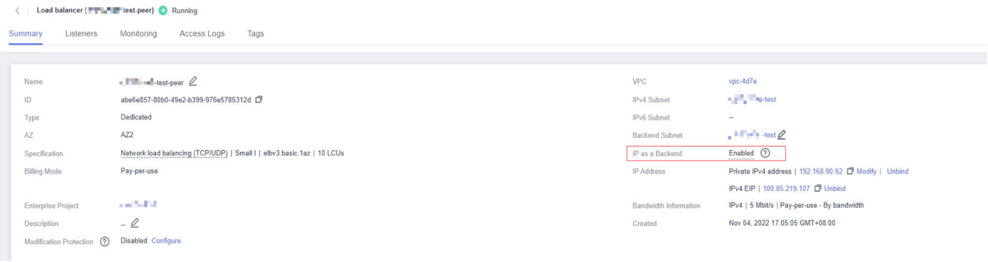
Step 1 Log in to the ELB console.

Step 2 In the ELB list, click the name of a dedicated ELB to go to its details page.



☐ Name/ID	Monitoring	Status	Type	Specifications	IP Address and Network	Listener (Frontend Proto...	Bandwidth Information	Billing Mode	Enterprise Project	Operation
☐ 98cf209a-908f-4c7b-957d...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 10...	192.168.0.8 (Private IPv4 address) 100.95.146.178 (IPv4 EIP) vpc-259a (VPC)	(TCP/8000)	IPv4 5 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Dec 26, 2022 16:28:48	default	Add Listener | Modify IPv4 Bandwidth | More
☐ 213d8d00-8a5c-451b-85c...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 10...	192.168.0.128 (Private IPv4 addr... 100.85.223.69 (IPv4 EIP) vpc-259a (VPC)	P/8000)	IPv4 1 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Dec 26, 2022 15:34:11	default	Add Listener | Modify IPv4 Bandwidth | More
☐ 8bec97b9-045f-43af-ab46...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 10...	192.168.0.60 (Private IPv4 addre... 100.93.0.18 (IPv4 EIP) vpc-259a (VPC)	CP/8000)	IPv4 5 Mbit/s Yearly/Monthly Yearly/Monthly 25 days until auto-deletion	Yearly/Monthly 25 days until expiration	default	Add Listener | Modify IPv4 Bandwidth | More
☐ 7146f9d3-498b-4d0a-886...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 10...	192.168.0.231 (Private IPv4 addr... 100.93.9.125 (IPv4 EIP) vpc-259a (VPC)	No listeners added. Add now	IPv4 1 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Dec 23, 2022 16:50:36	default	Add Listener | Modify IPv4 Bandwidth | More
☐ f2fd6282-0e6c-4050-b285...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 40...	192.168.0.109 (Private IPv4 addr... 100.85.114.69 (IPv4 EIP) vpc-259a (VPC)	No listeners added. Add now	IPv4 5 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Dec 15, 2022 16:34:14	default	Add Listener | Modify IPv4 Bandwidth | More
☐ 04b49ca3-8c2d-4612-872...		Running	Shared	Guaranteed Perform...	192.168.0.246 (Private IPv4 addr... 100.93.5.5 (IPv4 EIP) vpc-259a (VPC)	No listeners added. Add now	IPv4 1 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Dec 10, 2022 14:38:50	default	Add Listener | Modify IPv4 Bandwidth | More
☐ 451781ae-3c3e-4958-b7a...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 10...	192.168.0.238 (Private IPv4 addr... 100.85.126.53 (IPv4 EIP) vpc-dws123 (VPC)	listener-WX1... (TCP/8000)	IPv4 1 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Nov 22, 2022 10:01:20	default	Add Listener | Modify IPv4 Bandwidth | More
☐ -test-peer abe6e857-80b0-49e2-b39...		Running	Dedicated	Network load balanc... elbv3.basic.1az | 10...	192.168.90.92 (Private IPv4 addr... 100.85.219.107 (IPv4 EIP) vpc-4d7e (VPC)	No listeners added. Add now	IPv4 5 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Nov 04, 2022 17:05:12	zwx1124324	Add Listener | Modify IPv4 Bandwidth | More
☐ 8c434269-4a38-4dfd-935f...		Running	Dedicated	Application load bal... elbv3.basic.1az | 10...	192.168.0.71 (Private IPv4 addre... 100.85.112.72 (IPv4 EIP) vpc-dws123 (VPC)	listener-c722(HTTP/8000)	IPv4 5 Mbit/s Pay-per-use By bandwidth	Pay-per-use Created on Sep 06, 2022 19:58:39	default	Add Listener | Modify IPv4 Bandwidth | More

Step 3 On the **Summary** page, enable **IP as a Backend**, confirm the information, and click **OK**.

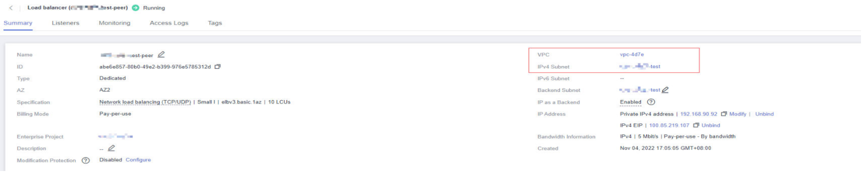


Load balancer ([redacted]test-peer) Running

Summary | Listeners | Monitoring | Access Logs | Tags

Name	[redacted]-test-peer	VPC	vpc-4d7e
ID	abe6e857-80b0-49e2-b399-976e5785312d	IPv4 Subnet	[redacted]-test
Type	Dedicated	IPv6 Subnet	--
AZ	AZ2	Backend Subnet	[redacted]-test
Specification	Network load balancing (TCP/UDP) | Small I | elbv3.basic.1az | 10 LCUs	IP as a Backend	Enabled
Billing Mode	Pay-per-use	IP Address	Private IPv4 address | 192.168.90.92 Modify | Unbind IPv4 EIP | 100.85.219.107 Unbind
Enterprise Project	[redacted]	Bandwidth Information	IPv4 | 5 Mbit/s | Pay-per-use - By bandwidth
Description	--	Created	Nov 04, 2022 17:05:05 GMT+08:00
Modification Protection	Disabled Configure		

Step 4 Check the VPC and subnet segment.



Load balancer ([redacted]test-peer) Running

Summary | Listeners | Monitoring | Access Logs | Tags

Name	[redacted]-test-peer	VPC	vpc-4d7e
ID	abe6e857-80b0-49e2-b399-976e5785312d	IPv4 Subnet	[redacted]-test
Type	Dedicated	IPv6 Subnet	--
AZ	AZ2	Backend Subnet	[redacted]-test
Specification	Network load balancing (TCP/UDP) | Small I | elbv3.basic.1az | 10 LCUs	IP as a Backend	Enabled
Billing Mode	Pay-per-use	IP Address	Private IPv4 address | 192.168.90.92 Modify | Unbind IPv4 EIP | 100.85.219.107 Unbind
Enterprise Project	[redacted]	Bandwidth Information	IPv4 | 5 Mbit/s | Pay-per-use - By bandwidth
Description	--	Created	Nov 04, 2022 17:05:05 GMT+08:00
Modification Protection	Disabled Configure		

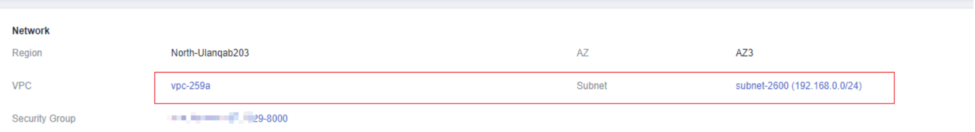
----End

Connecting the cluster VPC and the ELB VPC (by using VPC peering as an example)

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the cluster list, click the name of a cluster to go to the cluster details page. Check the VPC and subnet segment of the cluster.



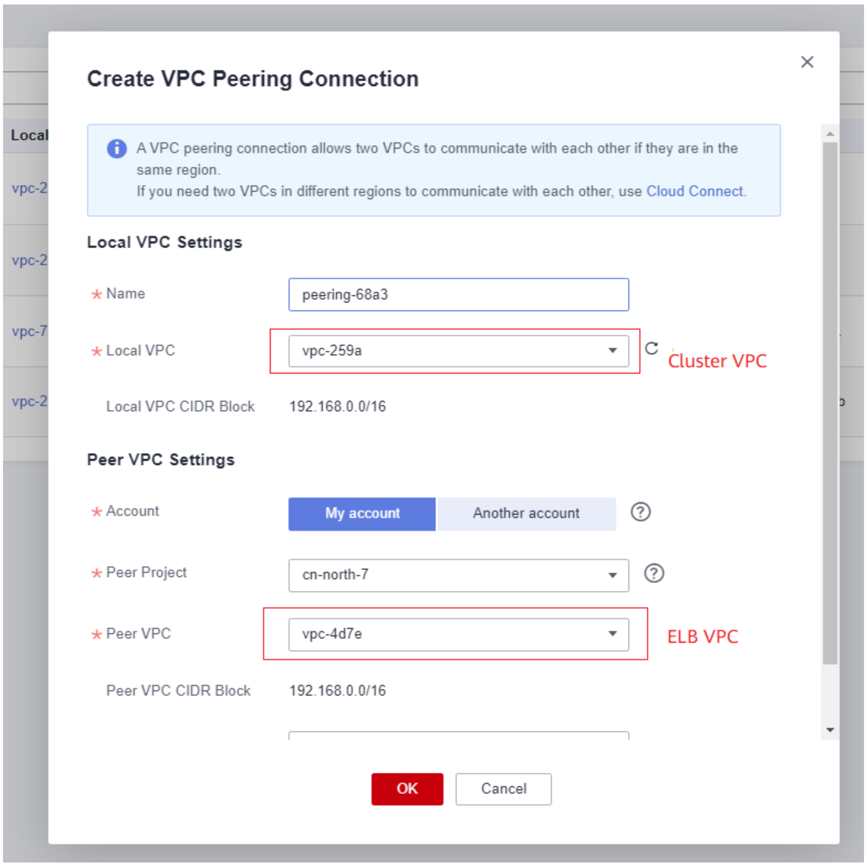
Network

Region	North-Ulanqab203	AZ	AZ3
VPC	vpc-259a	Subnet	subnet-2600 (192.168.0.0/24)
Security Group	[redacted]29-8000		

Step 4 Log in to the VPC management console. Choose **My VPCs** in the navigation pane and locate the VPC for which you want to create a VPC peering connection.

Step 5 Choose **VPC Peering Connections**. In the upper right corner of the page, click **Create VPC Peering Connection**.

Step 6 On the displayed page, set **Local VPC** to the cluster VPC, and set **Peer VPC** to the VPC of the ELB. Confirm the settings and click **OK**.



Create VPC Peering Connection

A VPC peering connection allows two VPCs to communicate with each other if they are in the same region.
If you need two VPCs in different regions to communicate with each other, use Cloud Connect.

Local VPC Settings

* Name	peering-68a3
* Local VPC	vpc-259a
Local VPC CIDR Block	192.168.0.0/16

Cluster VPC

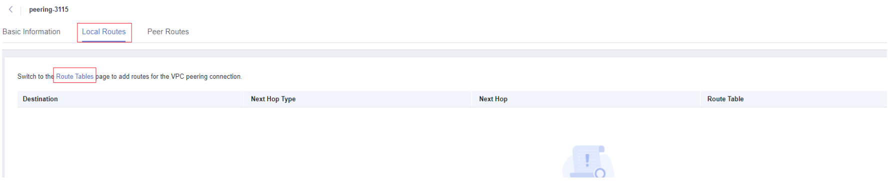
Peer VPC Settings

* Account	My account / Another account
* Peer Project	cn-north-7
* Peer VPC	vpc-4d7e
Peer VPC CIDR Block	192.168.0.0/16

ELB VPC

Step 7 Click **Add Route** to add the route information.

Step 8 Click the name of the created VPC peering connection. On the displayed page, click the **Local Routes** tab, click **Route Tables**, and add the route table of the cluster VPC.

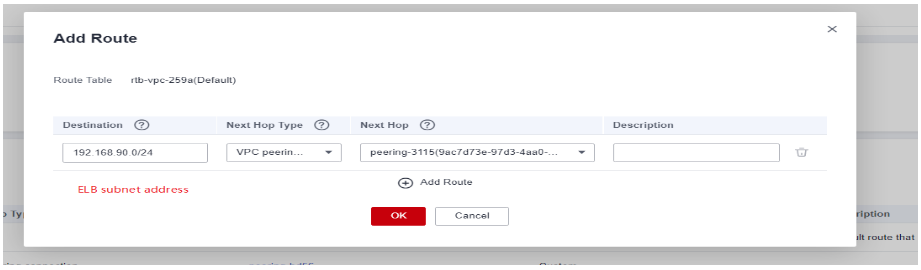


peering-3115

Switch to the Route Tables page to add routes for the VPC peering connection.

Destination	Next Hop Type	Next Hop	Route Table

Step 9 In the local route table, set **Destination** to the subnet CIDR block of the ELB VPC, set **Next Hop Type** to **VPC peering connection**, and set **Next Hop** to the created VPC peering connection. Click **OK**.



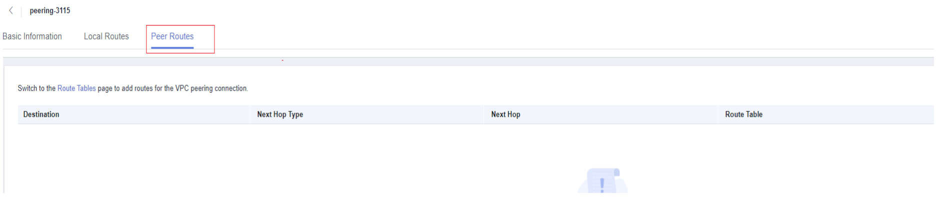
Add Route

Route Table rtb-vpc-259a(Default)

Destination	Next Hop Type	Next Hop	Description
192.168.90.0/24	VPC peerin...	peering-3115(9ac7d73e-97d3-4aa0-...	

ELB subnet address

Step 10 Go to the basic information page of the created VPC peering connection, click the **Peer Routes** tab, click **Route Tables**, and add the route table of the ELB VPC.

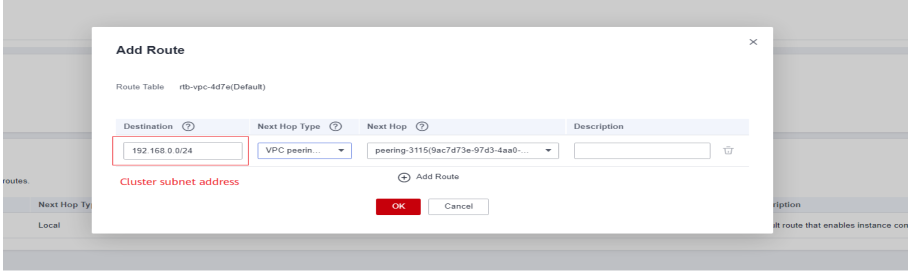


peering-3115

Switch to the Route Tables page to add routes for the VPC peering connection.

Destination	Next Hop Type	Next Hop	Route Table

Step 11 In the peer route table, set **Destination** to the subnet CIDR block of the cluster VPC, set **Next Hop Type** to **VPC peering connection**, and set **Next Hop** to the created VPC peering connection. Click **OK**.



Add Route

Route Table rtb-vpc-4d7e(Default)

Destination	Next Hop Type	Next Hop	Description
192.168.0.0/24	VPC peerin...	peering-3115(9ac7d73e-97d3-4aa0-...	

Cluster subnet address

Step 12 After the cluster is created, the network between the VPC where the cluster resides and the VPC where the load balancer resides is connected. For details, see section **Binding an ELB**.

----End

Disassociating ELB

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 4 On the **Basic Information** page that is displayed, click **Disassociate ELB**.

Step 5 After the request is delivered, go back to the **Clusters** page. Task information **Dissociating ELB** of the cluster is displayed. The process takes some time.

Step 6 Log in to the ELB management console, click the name of the dissociated ELB, switch to the **Backend Server Groups** tab, and check whether the cluster CNs are deleted.

----End

8.4.3 Adding or Deleting a CN in a GaussDB(DWS) Cluster

Overview

After a cluster is created, the number of required CNs varies with service requirements. The CN management function enables you to adjust the number of CNs in the cluster. The operations are as follows:

- **Adding CNs**
- **Deleting CNs**

NOTE

- This feature is supported only in cluster version 8.1.1 or later.
- Only cluster versions 8.1.3.300 and later (excluding 8.2.0) support online CN addition, deletion, and concurrent addition of multiple CNs.

Constraints and Limitations

- During resource provisioning, the default number of CNs is 3. You can adjust the number of CNs based on the number of provisioned nodes. The number of CNs ranges from 2 to 20.
- Do not perform other O&M operations when adding or deleting a CN.
- Adding CNs consumes lots of CPU and I/O resources, which will greatly impact job performance. You are advised to perform this operation during off-peak hours or after services are stopped.
- If a fault occurs when you add a CN node and the rollback fails, try adding the CN again. The deletion of a CN node cannot be rolled back.
- For a CN that fails to be added, you can only retry the addition. For a CN that fails to be deleted, you can only retry the deletion. Other O&M operations are not allowed for such CNs.
- If DDL operations, such as schema and function creation, are performed during CN deletion, an error may be reported because the deleted CN cannot be found. In this case, try again.
- If one of your CNs is abnormal, you can only delete this abnormal CN. If two or more CNs are abnormal, you can delete CNs only after the CNs are recovered from faults.

Adding CNs

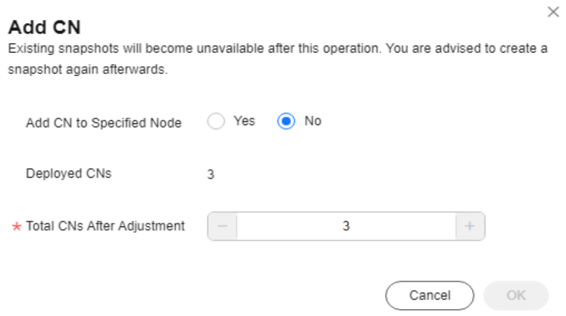
Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to which you want to add CNs.

Step 3 In the **Operation** column of the specified cluster, choose **More** > **Manage CN** > **Add CN Node**.

Step 4 On the displayed page, determine whether to add a CN to a specified node.

- If you select **No**, you can set the **CN quantity after adjustment**.



Add CN

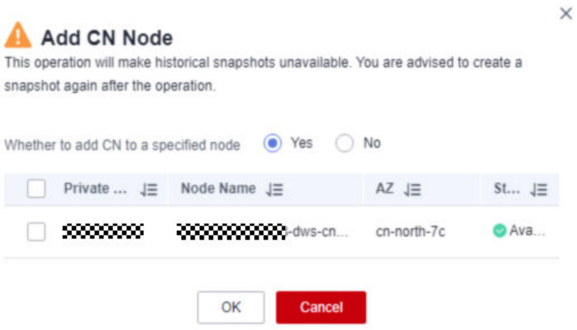
Existing snapshots will become unavailable after this operation. You are advised to create a snapshot again afterwards.

Add CN to Specified Node ○ Yes ● No

Deployed CNs 3

* Total CNs After Adjustment 3

- If you select **Yes**, specify the node.

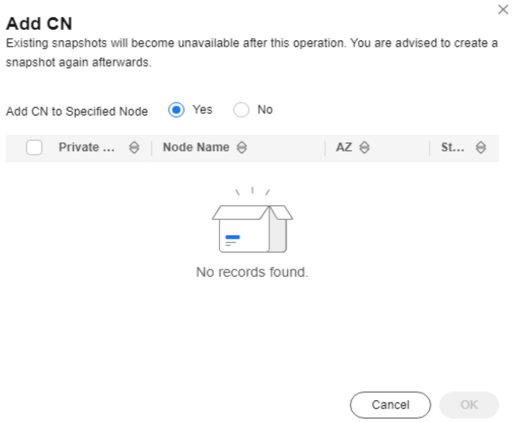


Add CN Node

This operation will make historical snapshots unavailable. You are advised to create a snapshot again after the operation.

Whether to add CN to a specified node ● Yes ○ No

□	Private ...	Node Name	AZ	St...
□	[redacted]	[redacted]-dws-cn...	cn-north-7c	Ava...



Add CN

Existing snapshots will become unavailable after this operation. You are advised to create a snapshot again afterwards.

Add CN to Specified Node ● Yes ○ No

□	Private ...	Node Name	AZ	St...

No records found.

NOTICE

- Before adding a CN, ensure that the cluster is in the **Available** or **Unbalanced** state.
- The number of CNs cannot exceed the total number of nodes after adjustment.
- You cannot add more CNs than the number of CNs that have already been deployed.

Step 5 Click **OK**.

----End

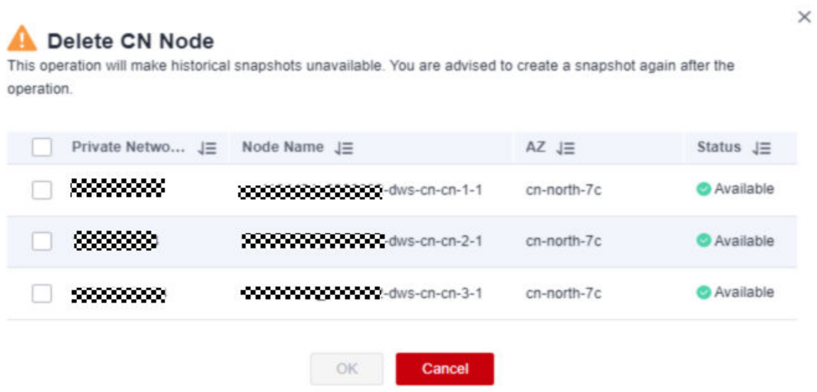
Deleting CNs

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster from which you want to delete CNs.

Step 3 In the **Operation** column of the specified cluster, choose **More** > **Manage CN** > **Delete CN Node**.

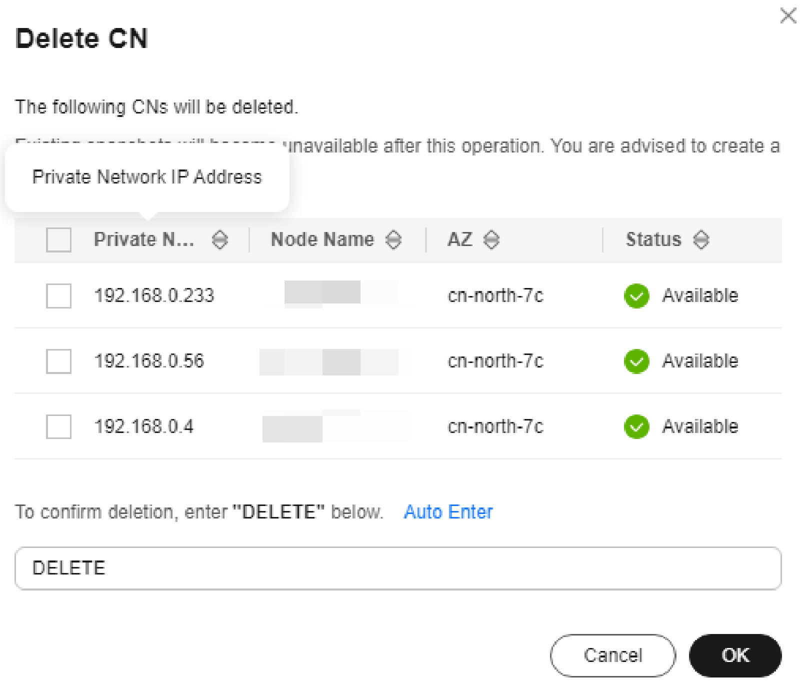
Step 4 On the displayed page, select the CN node to be deleted. After confirming that the information is correct, enter **DELETE** or click **One-Click Input** and click **OK** to delete the CN node.



Delete CN Node

This operation will make historical snapshots unavailable. You are advised to create a snapshot again after the operation.

□	Private Netwo...	Node Name	AZ	Status
□	[redacted]	[redacted]-dws-cn-cn-1-1	cn-north-7c	Available
□	[redacted]	[redacted]-dws-cn-cn-2-1	cn-north-7c	Available
□	[redacted]	[redacted]-dws-cn-cn-3-1	cn-north-7c	Available



Delete CN

The following CNs will be deleted.

Existing snapshots will become unavailable after this operation. You are advised to create a

□	Private N...	Node Name	AZ	Status
□	192.168.0.233	[redacted]	cn-north-7c	Available
□	192.168.0.56	[redacted]	cn-north-7c	Available
□	192.168.0.4	[redacted]	cn-north-7c	Available

To confirm deletion, enter "DELETE" below. Auto Enter

DELETE

NOTICE

- At least two CN must be retained.
- When deleting a CN from a multi-AZ cluster, reserve a normal CN node in each AZ. Faulty CN nodes (if any) can be deleted.
- When you delete a CN, the cluster must be in the **Available**, **Degraded**, or **Unbalanced** state.
- If an elastic IP address has been bound to a CN, the CN cannot be deleted.
- If abnormal nodes exist, only the abnormal CNs can be deleted.
  - If one CN is faulty, only this CN can be deleted.
  - If two or more CNs are faulty, no CN can be deleted.

----End

8.4.4 Managing GaussDB(DWS) Database Connections

Scenario

By default, a database supports a certain number of connections. Administrators can manage database connections to learn about the connection performance of the current database or increase the connection limit so that more users or applications can connect to the database at the same time.

Maximum Number of Connections

The number of connections supported by a cluster depends on its node flavor.

Table 8-15 Number of supported connections

Parameter	Description	Number of CN Connections	Number of DN Connections
max_connections	Specifies the maximum number of concurrent connections to the database.	800	Max (Number of vCPU cores/Number of DNs on a single node x 120 + 24, 5000)
max_pool_size	Specifies the maximum number of connections between the connection pool of a CN and another CN or DN.		
max_prepared_transactions	Specifies the maximum number of transactions that can stay in the **prepared** state simultaneously.		

NOTE

For details about CNs and DNs, see **Logical Cluster Architecture**.

Viewing the Maximum Number of Connections

Method 1: Post cluster creation, select the cluster name on the GaussDB(DWS) management console to access the **Parameter Modification** page and view the value of **max_connections**.

Method 2: Use an SQL client tool to establish a database connection within the cluster and execute an SQL query to obtain the value of **max_connections**.

```
SHOW max_connections;
```

Information similar to the following is displayed, showing that the maximum number of database connections is **200** by default.

```
max_connections
-----------------
200
(1 row)
```

Viewing the Number of Used Connections

Step 1 Use the SQL client tool to connect to the database in a cluster.

Step 2 View the number of connections in scenarios described in **Table 8-16**.

NOTICE

Except for database and user names that are enclosed with double quotation marks (") during creation, uppercase letters are not allowed in the database and user names in the commands in the following table.

Table 8-16 Viewing the number of connections

Description	Command

View the maximum number of sessions connected to a specific user.	Run the following command to view the maximum number of sessions connected to user **dbadmin**.
	`SELECT ROLNAME,ROLCONNLIMIT FROM PG_ROLES WHERE ROLNAME='dbadmin';`
	Information similar to the following is displayed. **-1** indicates that the number of sessions connected to user **dbadmin** is not limited.
	` rolname  | rolconnlimit`
	`----------+--------------`
	` dwsadmin |           -1`
	`(1 row)`

View the number of session connections that have been used by a user.	Run the following command to view the number of session connections that have been used by **dbadmin**.
	`SELECT COUNT(*) FROM V$SESSION WHERE USERNAME='dbadmin';`
	Information similar to the following is displayed. **1** indicates the number of session connections used by user **dbadmin**.
	` count`
	`-------`
	`     1`
	`(1 row)`

Description	Command

View the maximum number of sessions connected to a specific database.	View the upper limit of connections used by **gaussdb**.
	`SELECT DATNAME,DATCONNLIMIT FROM PG_DATABASE WHERE DATNAME='gaussdb';`
	Information similar to the following is displayed. **-1** indicates that the number of sessions connected to database **gaussdb** is not limited.
	` datname  | datconnlimit`
	`----------+--------------`
	` gaussdb |           -1`
	`(1 row)`

View the number of session connections that have been used by a database.	View the number of session connections that have been used by **gaussdb**.
	`SELECT COUNT(*) FROM PG_STAT_ACTIVITY WHERE DATNAME='gaussdb';`
	Information similar to the following is displayed. **1** indicates the number of session connections used by database **gaussdb**.
	` count`
	`-------`
	`     1`
	`(1 row)`

View the number of session connections that have been used by all users.	Run the following command to view the number of session connections that have been used by all users:
	`SELECT COUNT(*) FROM PG_STAT_ACTIVITY;`
	` count`
	`-------`
	`    10`
	`(1 row)`

----End

8.5 GaussDB(DWS) Resource Load Management

8.5.1 Overview

The system resources (CPU, memory, I/O, and storage resources) of a database are limited. When multiple types of services (such as data loading, batch analysis, and real-time query) are running at the same time, they may compete for resources and hinder operations. As a result, the throughput decreases and the overall query performance deteriorates. To avoid this problem, resources must be properly allocated.

GaussDB(DWS) provides the resource management function. You can put resources into different resource pools, which are isolated from each other. Then, you can associate database users with these resource pools. When a user starts a SQL query, the query will be transferred to the resource pool associated with the user. You can specify the number of queries that can be concurrently executed in a resource pool, the upper limit of memory used for a single query, and the memory and CPU resources that can be used by a resource pool. In this way, you can limit and isolate the resources occupied by different workloads, properly utilizing resources to process hybrid database loads and achieve high query performance. After a cluster is converted into a logical cluster, you can create, modify, or delete a resource pool in the logical cluster.

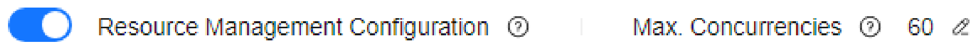
NOTICE

- This feature is supported only by clusters of version 8.0 or later.
- Resources cannot be managed during offline scale-out. If a resource management plan is enabled, stop it before performing offline scale-out.
- A storage-compute coupled data warehouse (standalone) does not support resource management.

Enabling or Disabling Resource Management

You can enable or disable resource management, and configure the maximum global concurrency. **Max. Concurrent Queries** refers to the maximum concurrent queries on a single CN. If you disable **Resource Management**, all resource management functions will be unavailable.

Figure 8-8 Enabling Resource Management Configuration



Resource Management Functions

The resource management functions of GaussDB(DWS) can be classified into the following types based on managed resources:

- Computing resource management. It is implemented using resource pools. Computing resources are isolated and controlled to prevent cluster-level issues caused by abnormal SQL queries. Computing resource management includes concurrency management, memory management, CPU management, and exception rules. For details, see **Resource Pool**.
- Storage space management: Storage is managed at user and schema level to prevent disk exhaustion, which makes the database read only. For details, see **Workspace Management**.
- Resource management plan: Resources are managed automatically based on a preconfigured plan, which can flexibly cope with complex scenarios. For details, see **Importing or Exporting a Resource Management Plan**.

The resource management functions of GaussDB(DWS) can be classified into the following types based on when they are implemented:

- Management before a query

  The service checks whether there are sufficient resources for a query. If there are, the query can be executed. If there are not, the query waits in a queue, and can be executed only after resources are released by other queries. Concurrency and memory are managed in this phase.

- Management during a query

  During query execution, resources used by the query are managed and controlled to prevent cluster exceptions caused by time-consuming SQL statements. Memory, CPU, storage space, and exception rules are managed in this phase.

Simple and Complex Queries

GaussDB(DWS) supports fine-grained resource management. Before resource management is implemented, queries are classified into complex queries (with long execution time and high resource consumption) and simple queries (with short execution time and low resource consumption). Simple and complex queries also differ in their estimated memory usage.

- The estimated memory usage of a simple query is less than 32 MB.
- The estimated memory usage of a complex query is 32 MB or higher.

In a hybrid load database, complex queries often occupy a large number of resources for a long time. A simple query queued after a complex query is time consuming, because it has to wait for the complex query to complete and resources to be freed up. To improve execution efficiency and system throughput, GaussDB(DWS) provides the short query acceleration function, managing simple queries separately.

- If short query acceleration is enabled, simple queries and complex queries are managed separately. Simple queries do not need to compete with complex queries for resources.
- If short query acceleration is disabled, simple and complex queries are under the same resource management rules.

To prevent a large number of simple queries from consuming too many resources during acceleration, concurrency management is performed on the queries. Resource management is not performed, because it may affect query performance and system throughput.

NOTE

Queries are categorized based on estimated memory usage, but the estimation does not equal the actual usage, nor does it reflect the query duration or CPU usage. In resource pools that are insensitive to performance and only run specific services, you can disable short query acceleration to manage resources and handle exceptions for simple queries.

8.5.2 Resource Pool

8.5.2.1 Feature Description

GaussDB(DWS) resource pools provide concurrency management, memory management, CPU management, and exception rules.

Concurrency Management

Concurrency represents the maximum number of concurrent queries in a resource pool. Concurrency management can limit the number of concurrent queries to reduce resource contention and improve resource utilization.

In the **Short Query Configuration** area, you can enable or disable the short query acceleration function. To change the number of simple statements (**-1** by default. **0** or **-1** indicates that the concurrent short queries are not controlled), you can enable short query acceleration.

The concurrency management rules are as follows:

- If short query acceleration is enabled, complex queries are under resource pool concurrency control, and simple queries are under short query concurrency control.
- If short query acceleration is disabled, complex and simple queries are both under resource pool concurrency control. Short query concurrency control is invalid.

Memory Management

Each resource pool occupies a certain percentage of memory.

Memory management aims to prevent out of memory (OOM) in a database, isolate the memory of different resource pools, and to control memory usage. Memory is managed from the following aspects:

- Global memory management

  To prevent OOM, set the global memory upper limit (**max_process_memory**) to a proper value. Global memory management before a query controls memory usage to prevent OOM management. Global memory management during a query prevents errors during query execution.

  - Management before a query

    The service checks the estimated memory usage of a query in the slow queue, and compares it with the actual usage. The estimation will be adjusted if it is smaller than the actual usage. Before a query is executed, the service checks whether the available memory is sufficient for the query. If yes, the query can be executed directly. If no, the query needs to be queued and executed after other queries release resources.

  - Management during a query

    During a query, the service checks whether the requested memory exceeds a certain limit. If yes, an error will be reported, and memory occupied by the query will be released.

- Resource pool memory management

  Resource pool memory management puts a limit on dedicated quotas. A workload queue can only use the memory allocated to it, and cannot use idle memory in other resource pools.

  The resource pool memory is allocated in percentage. The value range is 0 to 100. The value **0** indicates that the resource pool does not perform memory management. The value **100** indicates that the resource pool performs memory management and can use all the global memory.

  The sum of memory percentages allocated to all resource pools cannot exceed 100. Resource pool memory management is performed only before a query in the slow queue starts. It works in a way similar to the global memory management before a query. Before a query in the slow queue in a resource pool is executed, its memory usage is estimated. If the estimation is greater than the resource pool memory, the query needs to be queued and can be executed only after earlier queries in the pool are complete and resources released.

CPU Management

CPU share and CPU limit can be managed.

- CPU share: If the system is heavily loaded, CPU resources are allocated to resource pools based on the specific CPU shares. If the system not busy, this configuration does not take effect.
- CPU limit: It specifies the maximum number of CPU cores used by a resource pool. The resource usage of jobs in the resource pool cannot exceed this limit no matter whether the system is busy or not.

In the Resource Configuration area, you can modify the CPU time limit and CPU usage limit.

Choose either of the preceding management methods as needed. In CPU share management, CPUs can be shared and fully utilized, but resource pools are not isolated and may affect the query performance of each other. In CPU limit management, the CPUs of different resource pools are isolated, but this may result in the waste of idle resources.

NOTE

The CPU limit is supported only by clusters of version 8.1.3 or later.

Exception Rules

To avoid query blocking or performance deterioration, you can configure exception rules to let the service automatically identify and handle abnormal queries, preventing slow SQL statements from occupying too many resources for a long time.

In the **Associated Exception Rules** area, you can view the exception rules bound to the current resource pool, bind new exception rules, and unbind existing exception rules. For more information, see **Table 8-17**.

NOTE

- The cluster version 8.2.1 and later supports downgradation of exception rules. All exception rules support downgradation behaviors. After downgradation, only network resource preemption is downgraded to a low priority. Downgraded network queries are scheduled only when there is no normal queries.
- Only clusters of version 8.2.0 or later support association and unbinding of exception rules.

Table 8-17 Exception rule parameters

Parameter	Description	Value Range (0 Means No Limit)	Operation
Blocking Time	Job blocking time. It refers to the total time spent in global and local concurrent queuing. The unit is second. For example, if the blocking time is set to 300s, a job executed by a user in the resource pool will be terminated after being blocked for 300 seconds.	An integer in the range 1 to 2,147,483,647 . The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**
Execution Time	Time that has been spent in executing the job, in seconds. For example, if **Time required for execution** is set to 100s, a job executed by a user in the resource pool will be terminated after being executed for more than 100 seconds.	An integer in the range 1 to 2,147,483,647 . The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**
Total CPU time on all DNs.	Total CPU time spent in executing a job on all DNs, in seconds.	An integer in the range 1 to 2,147,483,647 . The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**
Interval for Checking CPU Skew Rate	Interval for checking the CPU skew, in seconds. This parameter must be set together with **Total CPU Time on All DNs**.	An integer in the range 1 to 2,147,483,647 . The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**
Total CPU Time Skew Rate on All DNs	CPU time skew rate of a job executed on DNs. The value depends on the setting of **Interval for Checking CPU Skew Rate**.	An integer in the range 1 to 100. The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**
Data Spilled to Disk Per DN	Allowed maximum job data spilled to disks on a DN. The unit is MB. **NOTE** This rule is supported only by clusters of version 8.2.0 or later.	An integer in the range 1 to 2,147,483,647 . The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**

Parameter	Description	Value Range (0 Means No Limit)	Operation
Average CPU Usage Per DN	Average CPU usage of a job on each DN. If **Interval for Checking CPU Skew Rate** is configured, the interval takes effect for this parameter. If the interval is not configured, the check interval is 30 seconds by default. **NOTE** This rule is supported only by clusters of version 8.2.0 or later.	An integer in the range 1 to 100. The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**
Maximum Bandwidth on a Single DN	Maximum network bandwidth (MB) for a job on a single DN. **NOTE** This rule is supported only by clusters of version 8.2.1 or later.	An integer in the range 1 to 2,147,483,647. The value **0** indicates no limit.	**Terminated**, **Downgraded**, or **Not limited**

8.5.2.2 Creating a Resource Pool

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 Click **Add Resource Pool**.

NOTE

Up to 63 resource pools can be created.

Step 5 Configure the resource pool. For more information, see **Table 8-18**.

Table 8-18 Resource pool parameters

Parameter	Description	Default Value
Name	Resource pool name	-

Parameter	Description	Default Value
CPU Resource (%)	- CPU share: Percentage of CPU time that can be used by users associated with the current resource pool to execute jobs. The value is an integer ranging from 1 to 99. - CPU limit: Maximum percentage of CPU cores used by a database user in a resource pool. The value is an integer ranging from 0 to 100. **0** indicates no limit. **NOTE** - The sum of the parameter values of all the resource pools cannot exceed 99%. If there is only one resource pool, the CPU share parameter does not take effect. - The CPU share parameter takes effect only when CPU contention occurs. For example, resource pools A and B are bound to CPU 1. If A and B are both running, the parameter takes effect. If there is only A running, the parameter does not take effect. - The sum of the CPU limits of all the resource pools cannot exceed 100%. The default value is 0. - The CPU limit is supported only by clusters of version 8.1.3 or later.	-
Memory Resource (%)	Percentage of the memory that can be used by a resource pool. You can manage memory and query concurrency separately or jointly. Under joint management, jobs can be delivered only when both the memory and concurrency conditions are met.	0 (not limited)
Storage Resource (MB)	Size of the available space for permanent tables. This parameter indicates the total tablespace of all DNs in a resource pool. Available space of a single DN = Configured value/Number of DNs.	-1 (not limited)
Complex Statement Concurrency	Maximum number of concurrent queries in a resource pool. You can manage memory and query concurrency separately or jointly. Under joint management, jobs can be delivered only when both the memory and concurrency conditions are met.	10
Network Bandwidth Weight	Weight for network scheduling. The value is an integer ranging from 1 to 2147483647. The default value is -1. **CAUTION** Only clusters of 8.2.1 and later versions support the network bandwidth weight feature. Storage-compute decoupled clusters do not support this.	-1 (not limited)

Step 6 Confirm the information and click **OK**.

----End

8.5.2.3 Modifying a Resource Pool

You can modify the parameters of a resource pool on the resource management page.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 In the **Resource Pools** drop-down list, click the name of a resource pool, including **Short Query Configuration**, **Resource Configuration**, **Associated Exception Rules**, and **Associated User**.

Step 5 Modify the short query configuration. Set the parameters as required and click **Save** on the right.

Parameter	Description	Value
Short Query Acceleration	Whether to enable short query acceleration. This function is enabled by default.	Enable
Simple Statement Concurrency	A short query is a job whose estimated memory used for execution is less than 32 MB. The default value **-1** indicates that the job is not controlled.	10

Step 6 Modify the resource configuration.

1. Click **Edit** on the right and modify the parameters according to **Table 8-19**.

Table 8-19 Resource pool parameters

Parameter	Description	Default Value
Name	Resource pool name	-

Parameter	Description	Default Value
CPU Resource (%)	– CPU share: Percentage of CPU time that can be used by users associated with the current resource pool to execute jobs. The value is an integer ranging from 1 to 99. – CPU limit: Maximum percentage of CPU cores used by a database user in a resource pool. The value is an integer ranging from 0 to 100. **0** indicates no limit. **NOTE** – The sum of the parameter values of all the resource pools cannot exceed 99%. If there is only one resource pool, the CPU share parameter does not take effect. – The CPU share parameter takes effect only when CPU contention occurs. For example, resource pools A and B are bound to CPU 1. If A and B are both running, the parameter takes effect. If there is only A running, the parameter does not take effect. – The sum of the CPU limits of all the resource pools cannot exceed 100%. The default value is 0. – The CPU limit is supported only by clusters of version 8.1.3 or later.	-
Memory Resource (%)	Percentage of the memory that can be used by a resource pool. You can manage memory and query concurrency separately or jointly. Under joint management, jobs can be delivered only when both the memory and concurrency conditions are met.	0 (not limited)
Storage Resource (MB)	Size of the available space for permanent tables. This parameter indicates the total tablespace of all DNs in a resource pool. Available space of a single DN = Configured value/Number of DNs.	-1 (not limited)
Complex Statement Concurrency	Maximum number of concurrent queries in a resource pool. You can manage memory and query concurrency separately or jointly. Under joint management, jobs can be delivered only when both the memory and concurrency conditions are met.	10
Network Bandwidth Weight	Weight for network scheduling. The value is an integer ranging from 1 to 2147483647. The default value is -1. **CAUTION** Only clusters of 8.2.1 and later versions support the network bandwidth weight feature. Storage-compute decoupled clusters do not support this.	-1 (not limited)

NOTE

The CPU limit is supported only by clusters of version 8.1.3 or later.

2. Click **OK**.

Step 7 Associate exception rules.

1. Click **Associated Exception Rules** on the left.
2. Select the exception rules to be associated from the current exception rule list. You can select multiple exception rules at a time.
3. Click **OK**.
4. To unbind an exception rule, click **Disassociate Rule**.

NOTE

- Only clusters of version 8.2.0 or later support association and unbinding of exception rules.
- The default exception rules take effect for users not associated with any resource pools, and for users whose resource pools do not have any exception rules configured. If a user-defined rule is associated with a resource pool, this rule prevails in the pool.
  - The default exception rules are supported only by clusters of version 8.2.0 or later. After a cluster of an earlier version is upgraded to version 8.2.0 or later, the default exception rules do not take effect. You can create exception rules as needed.
  - The cluster version 8.2.1 supports downgradation of exception rules. All exception rules support downgradation behaviors. After downgradation, only network resource preemption is downgraded to a low priority. Downgraded network queries are scheduled only when there is no normal queries.
  - A resource pool can be associated with up to 16 exception rules.
- A resource pool can be associated with multiple groups of exception rules, which work in an OR way. One group of exception rules works if all its conditions are met. For example, a resource pool is associated with two groups of rules. One group specifies **elapsedtime=2400**, and the other group specifies **elapsedtime=1200** and **memsize=2000**. If the execution time of a job reaches 1200 seconds and the memory usage reaches 2000 MB, or if the execution time reaches 2400 seconds, the job will be terminated.

Step 8 Associate users.

1. Click **User Association** on the left.
2. In the current user list, select the users to be associated. You can select multiple users at a time.
3. Click **OK**.
4. To disassociate a user, click **Disassociate User**.

NOTE

- The resources used by a user to run jobs can be controlled only after the user is added to a resource pool.
- A database user can be added to only one resource pool. Users removed from a resource pool can be added to another pool.
- Database administrators cannot be associated.
- If no resource pools are associated with a user, the user will be associated with **default_pool** by default, and its resource usage will be restricted by **default_pool**. The **default_pool** will be automatically created after resource management is enabled.

----End

8.5.2.4 Deleting a Resource Pool

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 In the **Resource Pools** area on the left, click the name of a resource pool.

Step 5 Click **Delete Resource Pool**.

NOTE

Deleting a resource pool that is associated with a database user is not allowed. To delete the resource pool, you need to first disassociate it from the database user.

----End

8.5.3 Resource Management Plan

8.5.3.1 Managing Resource Management Plans

Overview

The resource management plan is an advanced resource management feature provided by GaussDB(DWS). You can create a resource management plan, add multiple stages to the plan, and configure different queue resource ratios for the stages. After a plan is started, it automatically changes the resource configurations in different stages as scheduled. If you need to run services in different stages with different proportions of resources, you can create a resource management plan to automatically change resource configurations in different stages.

NOTE

Resource management plans are supported in version 8.1.0.100 or later.

Creating a Resource Management Plan

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Click to the **Resource Management Plans** tab and click **Add**.

Step 5 Enter a plan name and click **OK**.

NOTICE

- Before creating a resource management plan, you must design and create a resource pool. For details, see **Creating a Resource Pool**.
- You can create up to 10 resource management plans.

----End

Starting a Resource Management Plan

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Switch to the **Resource Management Plans** tab page and click **Start**/**Stop** .

Step 5 After confirming that the information is correct, click **OK** in the dialog box that is displayed to start or stop the plan.

NOTICE

- Only one plan can be started for each cluster.
- A plan must have at least two stages before it can be started.

----End

Viewing the Execution Logs of a Resource Management Plan

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Click **Resource Management Plans**. In the plan execution logs area, click **View** to view the plan execution logs.

----End

Deleting a Resource Management Plan

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Click **Resource Management Plans** and click **Delete** to delete the current resource management plan.

NOTICE

You cannot delete a running resource management plan.

----End

8.5.3.2 Managing Resource Management Plan Stages

Prerequisites

The following conditions must be met when you add or modify a resource management plan:

- The total CPU share of all resource pools does not exceed 99%.
- The total CPU limit of all resource pools does not exceed 100%.

NOTE

- The CPU usage limit can be configured only in 8.1.3 and later versions.
- The default start time is the UTC time. The next execution time is your local time.

Adding a Resource Management Plan Stage

Step 1 Log in to the GaussDB(DWS) console.

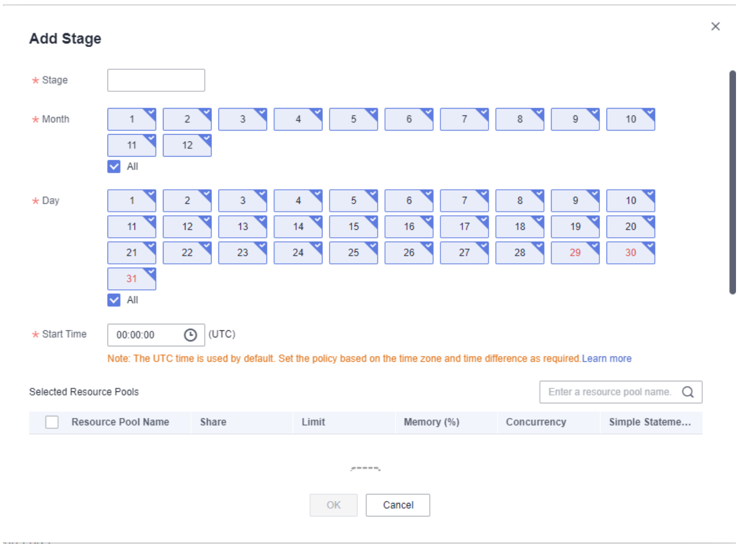
Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Go to the plan details page and click **Add** in the **Plan stage** area. On the **Add Stage** page, enter the stage name and configure the resource information. Confirm the configuration and click **OK**.

NOTICE

- Stages cannot be added to a running resource management plan.
- You can add a maximum of 48 stages for each plan.
- The switchover time of all phases in a plan cannot be the same.
- Configure the time, date, and month. Do not set an invalid date, for example, February 30.



Add Stage

* Stage

* Month 1 2 3 4 5 6 7 8 9 10 11 12

☑ All

* Day 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

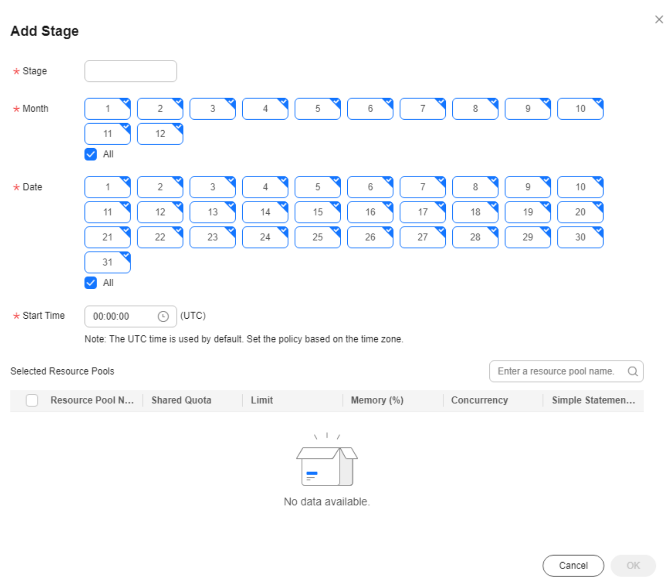
☑ All

* Start Time 00:00:00 (UTC)

Note: The UTC time is used by default. Set the policy based on the time zone and time difference as required.Learn more

Selected Resource Pools

□	Resource Pool Name	Share	Limit	Memory (%)	Concurrency	Simple Stateme...



Add Stage

* Stage

* Month 1 2 3 4 5 6 7 8 9 10 11 12

☑ All

* Date 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

☑ All

* Start Time 00:00:00 (UTC)

Note: The UTC time is used by default. Set the policy based on the time zone.

Selected Resource Pools

□	Resource Pool N...	Shared Quota	Limit	Memory (%)	Concurrency	Simple Statemen...

No data available.

----End

Modifying a Resource Management Plan Stage

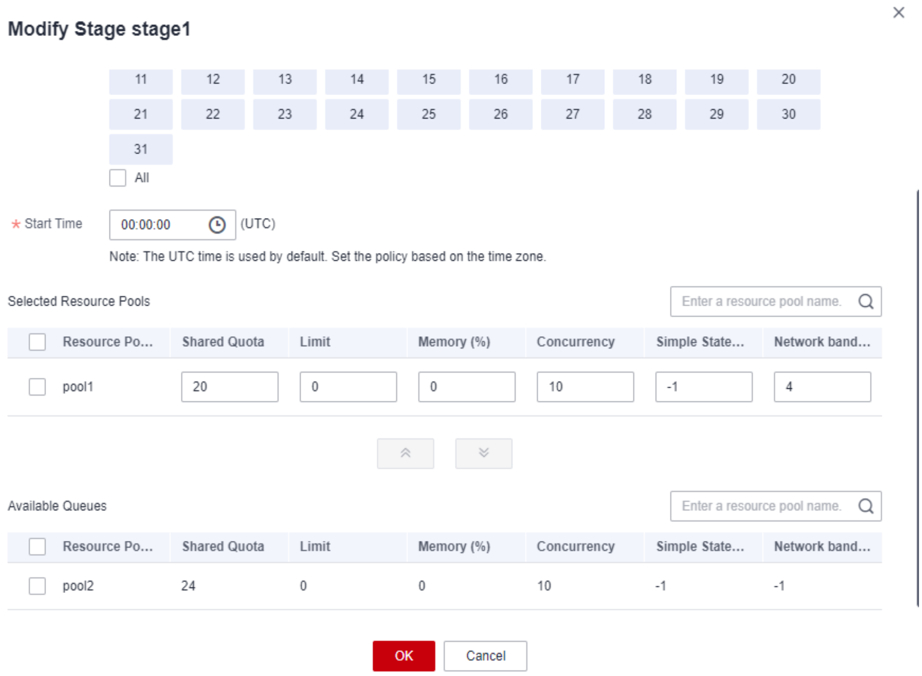
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Switch to the **Resource Management Plans** tab page and click **Modify** in the **Operation** column of the plan stage.

Step 5 Modify parameters, such as the stage changing time and resource configurations.



Modify Stage stage1

11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31									

□ All

* Start Time 00:00:00 (UTC)

Note: The UTC time is used by default. Set the policy based on the time zone.

Selected Resource Pools

□	Resource Po...	Shared Quota	Limit	Memory (%)	Concurrency	Simple State...	Network band...
□	pool1	20	0	0	10	-1	4

Available Queues

□	Resource Po...	Shared Quota	Limit	Memory (%)	Concurrency	Simple State...	Network band...
□	pool2	24	0	0	10	-1	-1

NOTE

Only clusters of the version 8.2.1 and later support the network bandwidth weight.

----End

Manually Changing the Resource Management Plan Stage

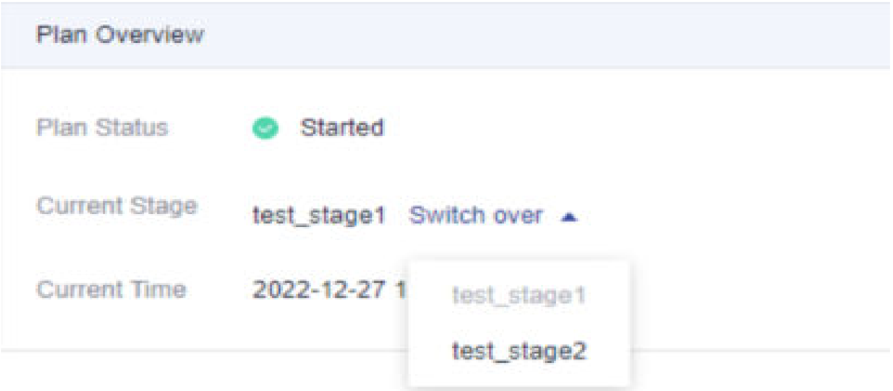
If a running plan needs to be switched to a stage in advance, you can manually do it.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Click **Resource Management Plans** and click the switch button in the plan overview area, and select the target stage.



Plan Overview

Plan Status Started

Current Stage test_stage1

Current Time 2022-12-27 1

----End

Importing/Exporting Resource Management Plan Stages

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Switch to the **Resource Management Plans** tab page. In the plan stages area, click **Import**/**Export** to import or export a resource management plan stage.

NOTICE

- Configurations cannot be imported to a running resource management plan.
- Ensure there is a resource pool before import.

----End

Deleting a Resource Management Plan Stage

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Switch to the **Resource Management Plans** tab page and click **Delete** in the **Operation** column of the plan stage.

----End

NOTICE

Stages in a running resource management plan cannot be deleted.

8.5.4 Workspace Management

Overview

Your cluster may run out of space if disk usage is not controlled, resulting in cluster exceptions and service interruption. Once disks are full, it takes long and huge efforts to recover workloads. Setting a database to read-only can reduce disk usage, but it also interrupts services. To solve this problem, GaussDB(DWS) provides multi-dimensional storage management. You can limit the permanent space that can be occupied by a schema; and can limit the usage of permanent space, temporary space, and operator space for a user.

- Schema level: Schema space management allows you to query database and schema space information in a cluster and modify the total schema space.

- User level: User space management allows you to limit users' space usage, preventing task execution from being blocked due to insufficient storage space. When you create a user in GaussDB(DWS), you can specify the space available to the user. The following types of storage space can be managed:
  - Permanent space (**PREM SPACE**)

    Space occupied by permanent tables (non-temporary tables) created by users
  - Temporary space (**TEMP SPACE**)

    Space occupied by temporary tables created by users
  - Operator spill space (**SPILL SPACE**)

    During query execution, if the actual memory usage is greater than estimated, the query may be spilled to disks. The storage space occupied in this case is called operator spill space. You can control a user's operator spill space usage during query execution.

NOTE

- This feature is supported only in cluster version 8.1.1 or later.
- Currently, the GaussDB(DWS) management plane only supports schema space management.

Procedure

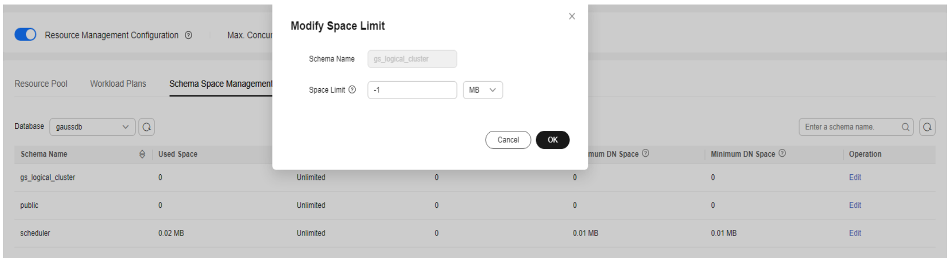
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 On the **Schema Space Manage** page, select a database.

Step 5 In the row where the scheme to be edited resides, click **Edit** and modify the space limit.



Modify Space Limit

Schema Name: gs_logical_cluster

Space Limit: -1 MB

Database: gaussdb

Schema Name	Used Space			mum DN Space	Minimum DN Space	Operation
gs_logical_cluster	0	Unlimited	0	0	0	Edit
public	0	Unlimited	0	0	0	Edit
scheduler	0.02 MB	Unlimited	0	0.01 MB	0.01 MB	Edit

Step 6 Click **OK**.

NOTE

- The space quota limits only common users but not database administrators. Therefore, when the used space is equal to the space limit, the actual used space may exceed the specified value.
- Quota for a single DN = Total quota/Number of DNs. Therefore, the configured value may be slightly different from the displayed value.

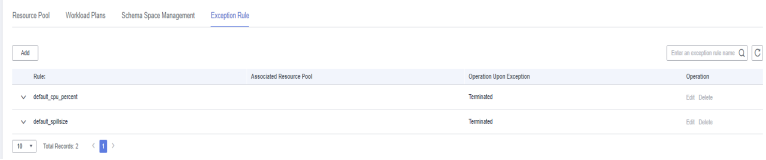
----End

8.5.5 Exception Rules

Feature Description

Some complex statements may consume a large number of resources for computing, leading to performance deterioration of the entire GaussDB(DWS) database. To maintain system stability, GaussDB(DWS) allows you to customize exception rules, and terminate/downgrade the tasks that hit the rules. You can use SQL syntax to configure exception rules based on your resource and workload conditions, and associate the rules with resource pools. The system has a default exception rule to maintain stability when resources are insufficient, in case no user-defined rule is set up.

Figure 8-9 Exception rules



Rule:	Associated Resource Pool	Operation Upon Exception	Operation
default_cpu_percent		Terminated	Edit Delete
default_spillsize		Terminated	Edit Delete

NOTICE

- This feature is supported only by clusters of version 8.2.0 or later. The exception rule "Maximum Bandwidth on a Single DN" is supported only by clusters of version 8.2.1 or later.
- The cluster version 8.2.1 supports downgradation of exception rules. All exception rules support downgradation behaviors. After downgradation, only network resource preemption is downgraded to a low priority. Downgraded network queries are scheduled only when there is no normal queries.
- The default exception rules are supported only by clusters of version 8.2.0 or later. After a cluster of an earlier version is upgraded to version 8.2.0 or later, the default exception rules do not take effect. You can create exception rules as needed.
- The default exception rule **default_memsize** is added to the cluster version 9.1.0.100, but it takes effect only in the newly installed cluster version 9.1.0.100 or later. When the cluster is upgraded to 9.1.0.100 or later, the default exception rules do not take effect. You can create rules as required.
- A resource pool can be associated with multiple groups of exception rules, which work in an OR way. One group of exception rules works if all its conditions are met. For example, a resource pool is associated with two groups of rules. One group specifies **elapsedtime=2400**, and the other group specifies **elapsedtime=1200** and **memsize=2000**. If the execution time of a job reaches 1200 seconds and the memory usage reaches 2,000 MB, or if the execution time reaches 2,400 seconds, the job will be terminated.
- Rules in the same exception rule group take effect only if all the conditions are met. For example, if you set **elapsedtime=1000** and **memsize=500**, it indicates that a job is terminated only if its execution time reaches 1,000 seconds and its memory usage reaches 500 MB. If only one of the thresholds is reached, the job will not be terminated.
- The default exception rules take effect for users not associated with any resource pools, and for users whose resource pools do not have any exception rules configured. If a user-defined rule is associated with a resource pool, this rule prevails in the pool.

User-defined Exception Rules and Default Exception Rules

The following table describes the user-defined exception rules and default exception rules supported by the current GaussDB(DWS) version.

Table 8-20 User-defined exception rules

Exception Threshold Type	Description	Value Range (-1 disables a parameter. 0 is not supported.)	Operation upon Exception
Blocking Time	Job blocking duration, in seconds. The time includes the total time spent in global and local concurrent queuing. The queuing time of each substatement (if any) in a statement is also counted.	-1 or 1 to INT64_MAX-1	Terminate / Downgrade
Execution Time	Execution duration of a job, in seconds. The time indicates the duration from the start point of execution to the current time point. The execution time of each substatement (if any) in a statement is also counted.	-1 or 1 to INT64_MAX-1	Terminate / Downgrade
Total CPU time on all DNs.	Total CPU time spent in executing a job on all DNs, in seconds.	-1 or 1 to INT64_MAX-1	Terminate / Downgrade
Total CPU Time Skew Rate on All DNs	CPU time skew of a job executed on DNs. The value depends on the setting of **elapsedtime**. The system starts to check the CPU time skew of a job every 5 seconds after the job execution time reaches **elapsedtime**.	**-1**, or 1 to 100	Terminate / Downgrade
Average CPU Usage Per DN	Average CPU usage of a job executed across all DNs.	**-1**, or 1 to 100	Terminate / Downgrade
Data Spilled to Disk Per DN	Allowed maximum job data spilled to disks on a DN. The unit is MB.	-1 or 1 to INT64_MAX-1	Terminate / Downgrade
Maximum Bandwidth on a Single DN	Maximum network bandwidth (MB) for a job on a single DN.	-1 or 1 to INT64_MAX-1	Terminate / Downgrade

Table 8-21 Default exception rules

Rule Name	Description	Operation upon Exception
default_cpu_percent	This rule is triggered if multiple jobs are running in a cluster, and the CPU usage of a resource pool reaches 90%. (If no resource pools are configured, the total CPU usage of the cluster is checked). This rule terminates the job whose execution time reached 15 minutes and average CPU usage exceeded 50%.	Terminate
default_spillsize	This rule is triggered if the size of data spilled to disk on a single DN reaches 1/10 of the instance space during job execution in the cluster.	Terminate
default_memsize	This event is triggered when the memory used by a job on a single DN reaches 80% or more of the minimum available memory of all DNs in the default cluster. **This rule is supported only by clusters of version 9.1.0.100 or later.**	Terminate

Creating an Exception Rule

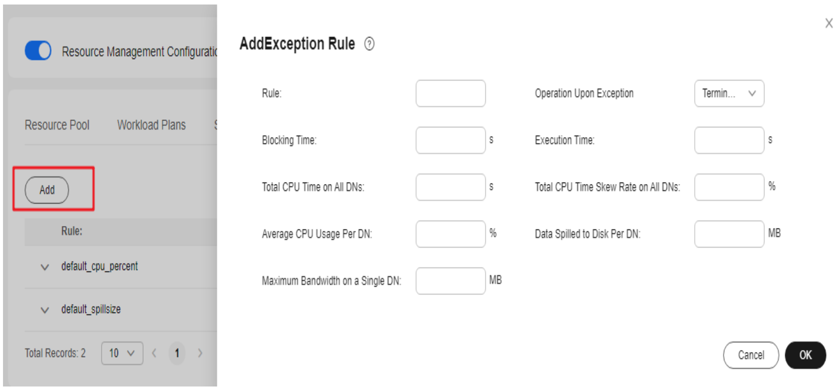
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click the name of the target cluster. The **Basic Information** page is displayed.

Step 3 In the navigation pane on the left, choose **Resource Management** and switch to the **Exception Rules** tab page.

Step 4 Click **Add** to add an exception rule.

Figure 8-10 Adding an exception rule



Step 5 Click **OK**.

NOTE

- After an exception rule is created, it does not take effect immediately. You need to associate it to a resource pool. For details, see **1. Associate exception rules.**.
- The cluster version 8.2.1 supports downgradation of exception rules. All exception rules support downgradation behaviors. After downgradation, only network resource preemption is downgraded to a low priority. Downgraded network queries are scheduled only when there are no normal queries.

----End

Editing an Exception Rule

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click the name of the target cluster. The **Basic Information** page is displayed.

Step 3 In the navigation pane on the left, choose **Resource Management** and switch to the **Exception Rules** tab page.

Step 4 Locate the row that contains the target exception rule and click **Edit** in the **Operation** column to edit the exception rule.

NOTE

- When editing an exception rule, if you want to delete an exception rule threshold, clear the value or set it to **-1**.
- If the exception threshold is changed during job execution, the new threshold will take effect for the statement being executed.

----End

Deleting an Exception Rule

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click the name of the target cluster. The **Basic Information** page is displayed.

Step 3 In the navigation pane on the left, choose **Resource Management** and switch to the **Exception Rules** tab page.

Step 4 Locate the row that contains the target exception rule and click **Delete** in the **Operation** column to delete the rule.

NOTE

If an exception rule has been associated to a resource pool, the exception rule cannot be deleted. You need to disassociate the exception rule from the resource pool before deleting it.

Step 5 Click **OK**.

----End

8.6 Managing GaussDB(DWS) Logical Clusters

8.6.1 Logical Cluster Overview

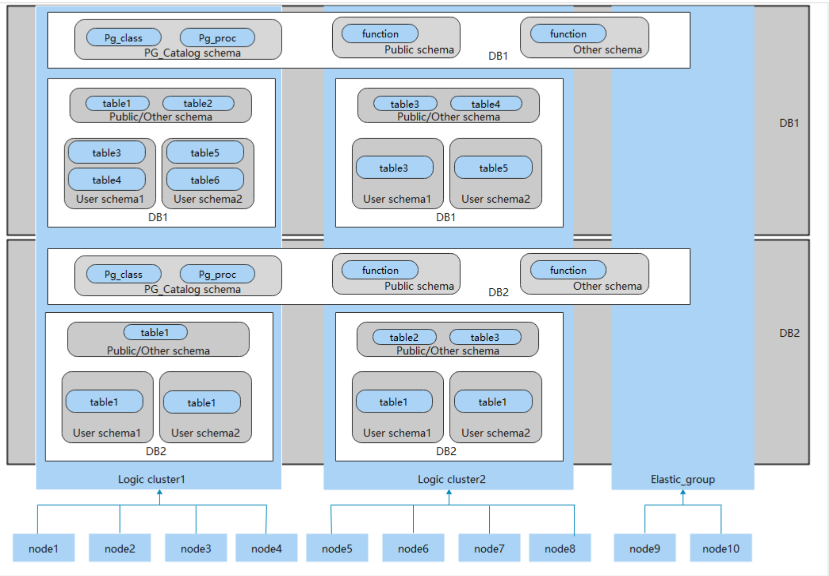
Concepts

A physical cluster can be divided into Node Groups, which are logical clusters. All physical nodes in a physical cluster are divided into multiple logical clusters. A logical cluster is essentially a node group that contains one or more physical nodes. Each physical node belongs to only one logical cluster, and user data tables can only be distributed within the same logical cluster. The data of each logical cluster is isolated from the others. The physical resources allocated to a logical cluster are mainly used for operations on its own data tables, but also for interactive queries with other logical clusters. An enterprise can deploy services on different logical clusters to implement unified service management, and meanwhile isolate the data and resources of services.

Logical clusters are created by dividing nodes of a physical cluster. Tables in a database can be allocated to different physical nodes by logical cluster. A logical cluster can contain tables from multiple databases. **Figure 8-11** shows the relationships between logical clusters, databases, and tables.

An elastic cluster is a cluster that always exists in logical cluster mode and consists of nodes that are not part of any logical cluster. It is a special node group that can have multiple or zero DNs. An elastic cluster cannot be manually created. When the first logical cluster is created in a physical cluster, an elastic cluster is also automatically created and all physical nodes not belonging to the logical cluster are automatically added to the elastic cluster. DNs in the elastic cluster will be used for logical clusters created later. To create a logical cluster, ensure that your logical cluster has DNs. (DNs are not required only when you create the first logical cluster in physical cluster mode.) You can add new physical nodes to the elastic cluster through scale-out.

Figure 8-11 Relationships between logical clusters, databases, and tables



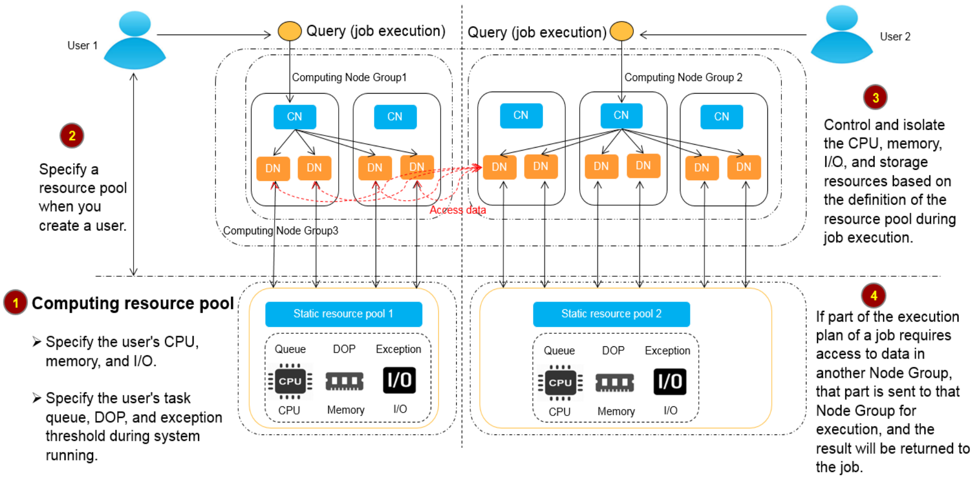
NOTE

- Logical clusters are supported in 8.1.0.100 or later.
- You are advised to allocate tables in a database to the same logical cluster.
- A logical cluster is not an independent sub-cluster. It can isolate data, resource, and permissions, but cannot be independently operated or maintained.
- The **Change all specifications** option does not support logical clusters.
- If the original physical cluster contains data, it is not possible to switch the logical cluster of a cluster. Ensure the original physical cluster is empty during the switchover.

Logical Cluster Architecture

Figure 8-12 shows the architecture of a physical cluster divided into multiple logical clusters. Nodes in the physical cluster are divided into Node Groups. The jobs of users 1 and 2 are executed in different Node Groups. The two users can define resource pools within their own logical cluster to control resources (CPU, memory, and I/O) used for different jobs. If some jobs of user 1 need to access the data of user 2, they can access data across Node Groups after being authorized. For a logical cluster, you can configure resources accessible across logical clusters to ensure its resources are sufficient.

Figure 8-12 Logical Cluster Architecture


User 1
Query (job execution)
Query (job execution)
User 2
Computing Node Group1
Computing Node Group 2
CN
CN
CN
CN
CN
DN
DN
DN
DN
DN
DN
DN
DN
DN
DN
Access data
Computing Node Group3
2
Specify a resource pool when you create a user.
3
Control and isolate the CPU, memory, I/O, and storage resources based on the definition of the resource pool during job execution.
1
Computing resource pool
Specify the user's CPU, memory, and I/O.
Specify the user's task queue, DOP, and exception threshold during system running.
Static resource pool 1
Queue
DOP
Exception
CPU
Memory
I/O
Static resource pool 2
Queue
DOP
Exception
CPU
Memory
I/O
4
If part of the execution plan of a job requires access to data in another Node Group, that part is sent to that Node Group for execution, and the result will be returned to the job.

A physical cluster is divided into multiple logical ones. You can define a resource pool for each of them based on service requirements. User tables are not distributed across logical clusters. If services do not access data across logical clusters, they will not compete for resources. Resources can be allocated to jobs in the same logical cluster by using resource pools. If necessary, you can let services access data across logical clusters, and control the resources used for such access to reduce resource competition between jobs within and outside a logical cluster.

After creating a physical cluster, you need to decide whether to divide it into logical clusters. You cannot divide it into logical clusters if you have already created user tables before, because these user tables are distributed on all physical nodes. For more information about the limitations, see **Constraints and Limitations**. If you want to manage an existing cluster (for example, a database cluster built in a version earlier than 8.1.0.100) as a logical cluster, you can upgrade the cluster to 8.1.0.100 or later and then convert all the nodes in the cluster into a single logical cluster. Then, add nodes to the physical cluster and create another logical cluster on the new nodes.

Operations on logical clusters include:

- **Adding/Deleting a Logical Cluster**:
  - Creating a logical cluster: After converting a physical cluster into a logical cluster, you can group some physical nodes into a logical cluster by specifying the name and the nodes of the logical cluster.
  - Deleting a logical cluster: You can delete a logical cluster with a specified name. After the logical cluster is deleted, the released physical nodes are removed from the physical cluster.
- **Managing Logical Clusters**:
  - Modifying a logical cluster: You can add or remove nodes from a logical cluster as needed.
  - Resource management (logical cluster mode): You can manage resources in a specified logical cluster (supported only by 8.1.3.*x* and later versions).
  - Scaling out a logical cluster: This operation increases the number of physical nodes in the logical cluster and redistributes tables in the logical cluster to the new physical nodes.
  - Restarting a logical cluster: This operation restarts all DNs in the logical cluster. Considering the impact on the entire physical cluster, the DNs in a logical cluster cannot be stopped or started individually.
  - Scaling in a logical cluster: Select and scale in a host ring from an elastic cluster.
- **Elastically Adding or Deleting a Logical Cluster**: Computing logical clusters can be created and deleted during the scheduled period of time to dynamically scale computing resources.

Constraints and Limitations

- The smallest unit of the creation, scale-out, and scale-in of a logical cluster is a ring. A ring consists of at least three hosts, where the primary, standby, and secondary DNs are deployed.
- During the logical cluster switchover, if the original physical cluster has data, the cluster will be locked. You can run simple DML statements, such as adding, deleting, modifying, and querying data. However, running complex DDL statements, such as operating database objects, will block services and report errors. Exercise caution when performing this operation.
- A logical cluster cannot be independently backed up or restored.
- A logical cluster cannot be independently upgraded.
- A physical cluster cannot be rolled back to a physical cluster after it is converted to a logical cluster.
- In logical cluster mode, only logical clusters can be created, and Node Groups cannot be created. In addition, Node Groups cannot be created in a logical cluster.
- O&M operations (creation, deletion, editing, scale-out, scale-in, and restart) of logical clusters cannot be performed concurrently.
- Public database objects (excluding system catalogs, foreign tables, and views) are distributed on all nodes in a physical cluster. After a node of the logical

cluster is restarted, the DDL operations performed by other logical clusters on the objects will be interrupted.

- In logical cluster mode, each DN only contains the tables in the logical cluster that the DN belongs to. User-defined functions need to be created on all DNs. Therefore, **%type** cannot be used to reference table field types in the function body.
- In logical cluster mode, the **WITH RECURSIVE** statement cannot be pushed down.
- In logical cluster mode, partitions can be swapped only in the same logical cluster. Partitioned tables and common tables in different logical clusters cannot be swapped.
- In logical cluster mode, if the function parameters or return values contain table types, these table types must belong to the same logical cluster.
- In logical cluster mode, when you create a foreign table using **CREATE TABLE... LIKE**, the source table and the foreign table to be created must be in the same logical cluster.
- In logical cluster mode, tables cannot be created schemas (by using **CREATE SCHEMA... CREATE TABLE** statements). Create a schema, and then create tables in the schema.
- A logical cluster does not support the architecture of one primary node and multiple standby nodes. A logical cluster takes effect only in the architecture of one primary node, one standby node, and one secondary node.
- A logical cluster user cannot access the global temporary tables created by another logical cluster user.

Required permissions on tools

The following describes user permissions for database objects in logical clusters:

- The **CREATE ON NODE GROUP** permission can be granted to any user or role for performing operations such as creating tables in a logical cluster.
  - If the schema specified for a created table is a private schema of a user (that is, the schema has the same name as the user and the owner of the schema is the user), the owner of the created table defaults to the user. You do not need to associate the table with a logical cluster.
  - When a user associated with a logical cluster creates a table, if the **to group** clause is not specified, the table will be created in that logical cluster. The logical cluster associated with the user can be changed.
  - If a user is not associated with any logical cluster, when the user creates a table, the table will be created in the logical cluster specified by **default_storage_nodegroup**. If **default_storage_nodegroup** is set to **installation**, the table will be created in the first logical cluster. In logical cluster mode, the logical cluster with the smallest OID is set as the first logical cluster. If **default_storage_nodegroup** is not set, its value is **installation** by default.
  - You can create read-only logical clusters in a storage-compute decoupled cluster. If a user is associated with a read-only logical cluster, session-level temporary tables (local temporary tables and volatile temporary tables, excluding global temporary tables) can be created only in the read-only logical cluster. If the user creates other common and foreign tables, the

tables will be created in the logical cluster specified by **default_storage_nodegroup**. If **default_storage_nodegroup** is set to **installation**, the table will be created in the first logical cluster.

  - The system administrator can run the **ALTER ROLE** command to set the default storage nodegroup for each user. For details about the syntax, see **ALTER ROLE**

- Table creation rules
  - If **to group** is not specified for a user table but **default_storage_nodegroup** is set, tables will be created in the specified logical cluster.
  - If **default_storage_nodegroup** is set to **installation**, tables will be created in the first logical cluster, that is, the logical cluster with the smallest OID.
- The owner of a table can be changed to any user. However, you need to check the schema and node group permissions when performing operations on the table.
- A system administrator can be associated with a logical cluster and can create tables in multiple logical clusters.
  - If the system administrator is associated with a logical cluster and **to group** is not specified when you create a table, the table will be created in the associated logical cluster by default. If **to group** is specified, the table is created in the specified logical cluster.
  - If the system administrator is not associated with a logical cluster and **to group** is not specified, tables are created in the logical cluster of **default_storage_nodegroup**. For details, see the **table creation rules**.
- System administrator permissions can be granted to a user associated with a logical cluster, but the **table creation rules** also apply.
- The logical cluster permission for accessing non-table objects (such as schemas/sequences/functions/triggers) will not be checked.
- A resource pool must be associated with a logical cluster.
  - A logical cluster can be associated with multiple resource pools but a resource pool can be associated with only one logical cluster.
  - Jobs executed by logical cluster users associated with a resource pool can only use resources in the resource pool.
  - You do not need to create a workload group to define the number of concurrent jobs in a logical cluster. Therefore, workload groups are not required for logical clusters.
- When a logical cluster is deleted, only the table, foreign table, and resource pool objects are deleted.
  - Objects dependent on the tables (including the partly dependent sequences/functions/triggers) in the logical cluster will also be deleted.
  - Logical cluster associations with its users and parent-child tenants will be removed during the process. As a result, the users will be associated with the default **installation** node group and with the default global resource pool.
- A logical cluster user can create a database if granted the permission.

Replication Table Node Group

A replication table node group is a special node group in logical cluster mode. It can contain one or more logical clusters, but can only create replication tables. One typical scenario is to create public dimension tables. If multiple logical clusters require some common dimension tables, create a replication table node group and add the common dimension tables to it. The logical clusters contained in the replication table node group can access these dimension tables on the local DNs, with no need to access the tables on other DNs. If a logical cluster is scaled in, the replication table node group will be scaled accordingly. If the logical cluster is deleted, the replication table node group will be scaled in. However, if the replication table node group contains only one logical cluster and the logical cluster is deleted, the replication table node group will also be deleted. In this case, create tables in a logical cluster instead.

Create a replication table node group using the **CREATE NODE GROUP** SQL statement and delete one using **DROP NODE GROUP**. Before deleting a replication table node group, delete all table objects in the node group.

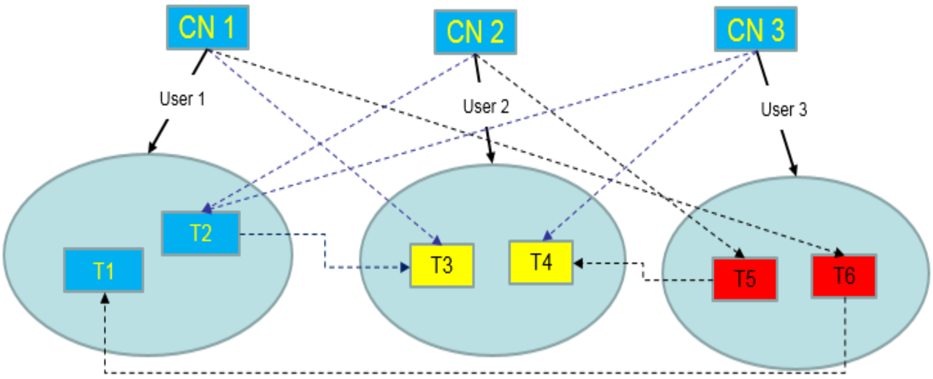
NOTE

Creation of replication table node groups is supported in 8.1.2 or later.

Application Scenarios

Scenario 1: Isolating data with different resource requirements

Figure 8-13 Logical cluster division based on resource requirements



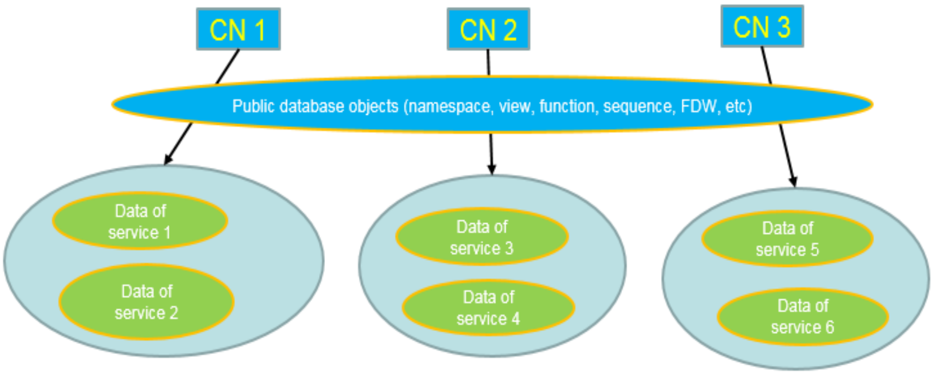
As shown in the preceding figure, data with different resource requirements is stored in different logical clusters, and different logical clusters also support mutual access. This ensures that functions are not affected while resources are isolated.

- Tables T1 and T2 are used to calculate a large amount of data and generate report data (for example, bank batch processing). This process involves large batch import and big data query, which consume a lot of memory and I/O resources of nodes and take a long time. However, such a query does not require high real-time performance. Therefore, the data of these two tables can be separated into a different logical cluster.

- Tables T3 and T4 contain some computing data and real-time data, which are mainly used for service point query and real-time query. These queries need high real-time performance. To prevent the interference of other high-load operations, the data of these two tables can be separated into a different logical cluster.
- Tables T5 and T6 are mainly used for OLTP operations with high concurrency. Data in these tables is frequently updated and sensitive to I/O. To prevent the impact of big data query on I/O, the data of these two tables can be separated into a different logical cluster.

Scenario 2: Isolating data for different services and enhancing the multi-tenancy of a data cluster

Figure 8-14 Logical cluster-based multi-service data and multi-tenant management



A large database cluster often stores data for various services. Each service has its own data tables. To allocate resources for different services, you can create multiple tenants. Specifically, assign different service users to different tenants to minimize resource contention among services. As the service scale grows continuously, the number of services in the cluster system also increases. Creating multiple tenants becomes less effective in controlling resource competition. Since each table is distributed across all DNs of a database cluster, every data table operation may involve all DNs, which increases network load and system resource consumption. Simply scaling up the cluster is not enough to solve this problem. Therefore, multiple logical clusters can be created to handle the increasing number of services, as shown in the figure above.

You can create a separate logical cluster and assign new services to it. This way, new services have little impact on existing services. Also, if the service scale in existing logical clusters grows, you can scale out the existing logical clusters.

NOTE

A logical cluster is not suitable for managing multiple independent database systems. An independent database system requires independent O&M and needs to be managed, monitored, backed up, and upgraded separately. Moreover, faults must be isolated between clusters. Logical clusters cannot achieve independent O&M and complete fault isolation.

8.6.2 Adding/Deleting a Logical Cluster

Adding a Logical Cluster

Precautions

- If you access the **Logical Clusters** page for the first time, the metadata of the logical cluster created at the backend is synchronized to the frontend. After the synchronization is complete, you can view information about the logical clusters at the frontend. The logical cluster name is case sensitive. For example, metadata of **lc1** and **LC1** cannot be synchronized.
- The original resource pool configuration is cleared when the cluster is converted from physical to logical. The resource pool information configured after the cluster is converted to a logical cluster will be bound to the logical cluster.

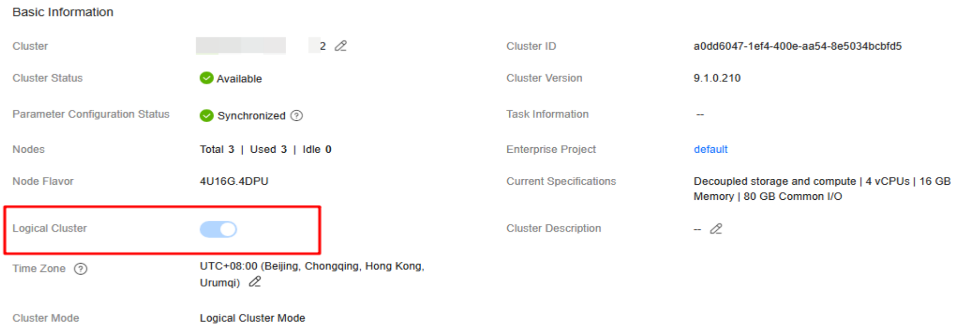
Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 Enable the logical cluster function. The **Logical Clusters** menu item will be displayed in the navigation pane on the left.

Figure 8-15 Logical clusters

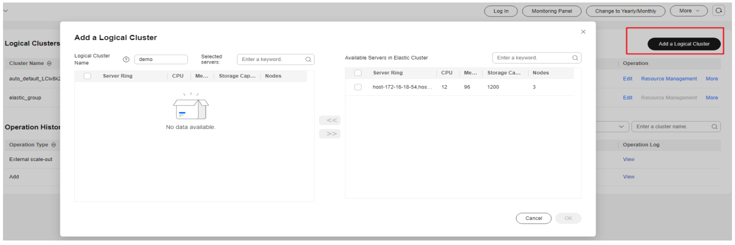


Basic Information

Cluster	[redacted] 2	Cluster ID	a0dd6047-1ef4-400e-aa54-8e5034bcbfd5
Cluster Status	Available	Cluster Version	9.1.0.210
Parameter Configuration Status	Synchronized	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	4U16G.4DPU	Current Specifications	Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 80 GB Common I/O
Logical Cluster		Cluster Description	--
Time Zone	UTC+08:00 (Beijing, Chongqing, Hong Kong, Urumqi)		
Cluster Mode	Logical Cluster Mode		

Step 4 Go to the **Logical Clusters** tab and click **Add Logical Cluster**.

Step 5 Move the ring you want to add from the right to the left panel, enter the logical cluster name, and click **OK**.



Add a Logical Cluster

Logical Cluster Name: demo

Selected servers: Enter a keyword.

Server Ring	CPU	Me...	Storage Cap...	Nodes

No data available.

Available Servers in Elastic Cluster: Enter a keyword.

Server Ring	CPU	Me...	Storage Ca...	Nodes
host-172-16-18-54,hos...	12	96	1200	3

Cluster Name	Operation
auto_default_LCiv8k2	Edit Resource Management More
elastic_group	Edit Resource Management More

Enter a cluster name.

Operation Type	Operation Log
External scale-out	View
Add	View

----End

Deleting Logical Clusters

Precautions

- The first manually added logical cluster cannot be deleted.
- Nodes of the deleted logical cluster are added to the elastic cluster.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane on the left, switch to the **Logical Clusters** page, locate the row that contains the logical cluster to be deleted, and click **Delete** in the **Operation** column.

Step 4 Confirm that all information is correct and click **OK**.

----End

8.6.3 Managing Logical Clusters

Editing a Logical Cluster

Precautions

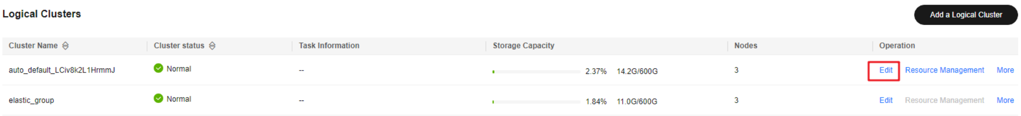
- Nodes are added to or removed from a logical cluster by ring.
- At least one ring must be reserved in a logical cluster.
- The ring removed from the logical cluster will be added to the elastic cluster.
- Logical clusters of version 8.1.3 and later support online scale-out. Clusters of version 8.2.1.100 and later support job termination.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **Logical Clusters** and click **Edit** in the **Operation** column of the target cluster.



Logical Clusters

Cluster Name	Cluster status	Task Information	Storage Capacity	Nodes	Operation
auto_default_LCiv8k2L1HrmmJ	Normal	--	2.37% 14.2G/600G	3	Edit Resource Management More
elastic_group	Normal	--	1.84% 11.0G/600G	3	Edit Resource Management More

Step 4 Add a node to the logical cluster by moving the selected ring from the right to the left, or remove a node from the logical cluster by moving the selected ring from the left to the right, and click **OK**.

Step 5 When adding a node, select online or offline scale-out as needed.

Step 6 If you select online scale-out, you can configure job termination. When job termination is enabled and there is congestion during online scale-out, the system will wait for the specified duration before terminating the congested jobs. The duration value must be an integer ranging from **30** to **1200**.

----End

Managing Resources (in a Logical Cluster)

Precautions

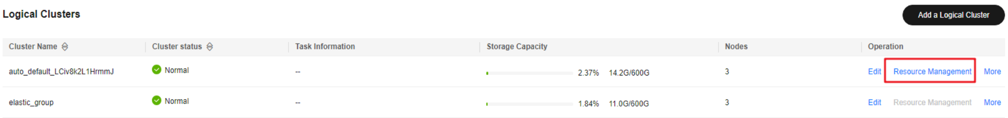
The original resource pool configuration is cleared when the cluster is converted from physical to logical. You have to add the resource pool again if you want to configure it after the conversion.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **Logical Cluster Management**. In the **Operation** column of a logical cluster, click **Resource Management Configurations**. On the displayed page, you can manage resources in a logical cluster. For details, see **GaussDB(DWS) Resource Load Management**.



Logical Clusters

Cluster Name	Cluster status	Task Information	Storage Capacity	Nodes	Operation
auto_default_LCiv8k2L1HrmmJ	Normal	--	2.37% 14.2G/600G	3	Edit Resource Management More
elastic_group	Normal	--	1.84% 11.0G/600G	3	Edit Resource Management More

----End

Restarting Logical Clusters

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane on the left, switch to the **Logical Clusters** page, locate the row that contains the logical cluster to be restarted, and click **More** > **Restart** in the **Operation** column.

Step 4 Confirm that all information is correct and click **OK**.

----End

Scaling Out a Logical Cluster

Prerequisites

- Logical clusters of version 8.1.3 and later support online scale-out.
- Before a scale-out, you need to enable the logical cluster mode and add a logical cluster.
- After scaling out or scaling in a logical cluster, you need to reconfigure the backup policy for full backup. For details, see **Configuring an Automated Snapshot Policy**.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 On the displayed **Clusters** page, choose **More** > **Scale Node** > **Scale Out**.

Step 3 On the scale-out page, select a logical or elastic cluster. Enabling online scale-out also allows you to enable congested job termination. If congestion occurs during online scale-out and job termination is enabled, the system will wait for the specified duration before terminating congested jobs. The duration value must be an integer between **30** and **1200**.

NOTE

Clusters of version 8.2.1.100 and later support job termination.

----End

Scaling In a Logical Cluster

Constraints

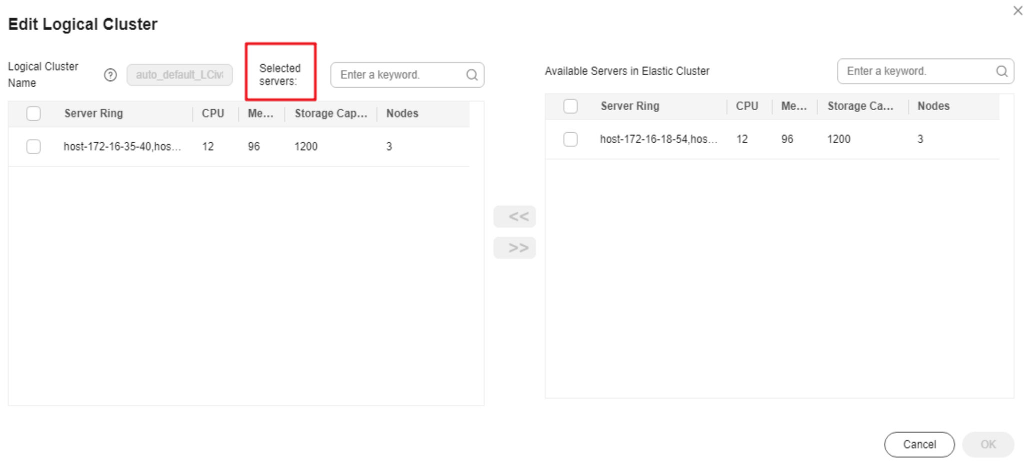
- A host ring with CN nodes cannot be scaled in.
- A host ring with GTM and CM nodes cannot be scaled in.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane on the left, switch to the **Logical Clusters** page. Locate the row that contains the target elastic pool and click **Edit** in the **Operation** column to delete nodes from the elastic pool (move the selected ring from the left to the right).



Edit Logical Cluster

Logical Cluster Name: auto_default_LCM

Selected servers:

Server Ring	CPU	Me...	Storage Cap...	Nodes
host-172-16-35-40,hos...	12	96	1200	3

Available Servers in Elastic Cluster

Server Ring	CPU	Me...	Storage Ca...	Nodes
host-172-16-18-54,hos...	12	96	1200	3

Step 4 Click **OK**.

----End

8.6.4 Elastically Adding or Deleting a Logical Cluster

Context

Logical clusters can be manually added or deleted, and also have the ability to automatically expand or shrink as needed. Computing logical clusters can be created and deleted during the scheduled period of time to dynamically scale computing resources.

- A custom scheduled scaling plan creates a logical cluster that provides computing power. Once such logical cluster is associated, user's queries are handled by it, while table creation statements are still managed by the original logical cluster. For instructions on how to configure a custom scheduled scaling plan, see **Configuring a Custom Scheduled Scaling Plan**.
  - A user can be bound to only one computing logical cluster.
  - If a user associated with the read-only logical cluster has workloads in progress when the cluster is deleted, an error may be reported.
- An auto scaling plan creates a logical cluster that facilitates parallel expansion. Once such logical cluster is associated with the primary logical cluster, specific queries from the primary logical cluster are routed to the logical cluster, but table creation statements are still executed in the original logical cluster. For how to configure an auto scaling plan, see **Configuring an Auto Scaling Plan**.

  To reduce statement queuing time, use parallel expansion. When the primary cluster is overwhelmed by high-concurrency tasks and lacks resources like memory, GaussDB(DWS) will automatically add extra cluster capacity to process the increased read and write statements. The performance and data consistency are consistent for users, regardless of whether the statement is executed on the primary cluster or the parallel extended cluster. You can set up a resource pool to control which statements are directed to a parallel extended cluster. When parallel expansion is enabled, qualifying statements are directed to the concurrent expansion cluster instead of being queued. Here are the limitations of the parallel expansion feature:

- Only V3 tables and foreign tables are supported. If the table is a replication table, only **SELECT** is supported.
- Only **SELECT**, **INSERT**, **UPDATE**, and **DELETE** statements are supported.
- **COPY** cannot be used to import data.
- The **UPSERT** statement is not supported.
- Transaction blocks are not supported.
- Stored procedures are not supported.
- Recursive statements with the **RETURNING** clause and **WITH RECURSIVE** are not supported.
- Lightweight update is not supported.
- **INSERT** statements generated by a single **VALUES** or **generate_series** are not supported.

NOTE

- You can manually add or delete plans for storage-compute decoupled clusters. For earlier versions, contact technical support to upgrade them.
- Only storage-compute decoupled and ECS clusters of 9.1.0 and later versions support auto elastic addition and deletion.
- In a yearly/monthly storage-compute decoupled cluster, nodes are automatically added when logical clusters are added as scheduled. Nodes are billed on a pay-per-use basis.

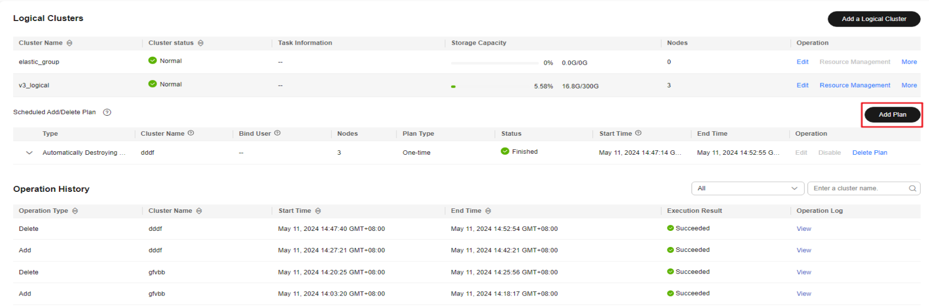
Configuring a Custom Scheduled Scaling Plan

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 After storage-compute decoupled clusters are created, they are in logical cluster mode.

- Storage-compute decoupled cluster: In the navigation pane on the left, click **Logical Clusters**. On the displayed page, click **Add Plan** to configure a proper scheduling plan.


Logical Clusters

Cluster Name	Cluster status	Task Information	Storage Capacity	Nodes	Operation
elastic_group	Normal	--	0% 0.0G/0G	0	Edit Resource Management More
v3_logical	Normal	--	5.58% 16.8G/300G	3	Edit Resource Management More

Scheduled Add/Delete Plan

Type	Cluster Name	Bind User	Nodes	Plan Type	Status	Start Time	End Time	Operation
Automatically Destroying ...	dddf	--	3	One-time	Finished	May 11, 2024 14:47:14 G...	May 11, 2024 14:52:55 G...	Edit Disable Delete Plan

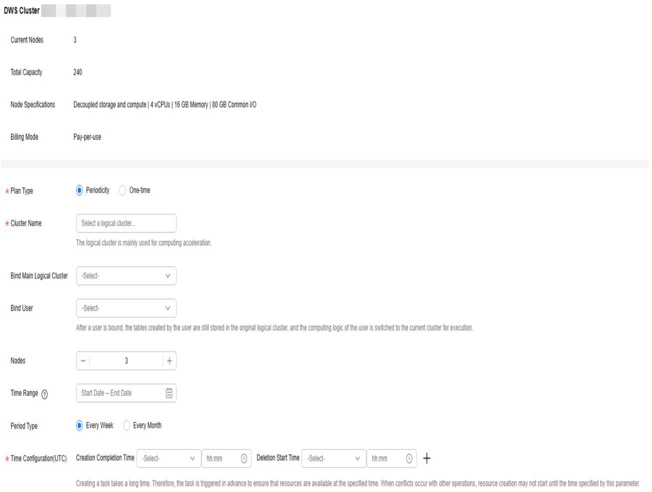
Operation History

Operation Type	Cluster Name	Start Time	End Time	Execution Result	Operation Log
Delete	dddf	May 11, 2024 14:47:40 GMT+08:00	May 11, 2024 14:52:54 GMT+08:00	Succeeded	View
Add	dddf	May 11, 2024 14:27:21 GMT+08:00	May 11, 2024 14:42:21 GMT+08:00	Succeeded	View
Delete	gfvbb	May 11, 2024 14:20:25 GMT+08:00	May 11, 2024 14:25:56 GMT+08:00	Succeeded	View
Add	gfvbb	May 11, 2024 14:03:20 GMT+08:00	May 11, 2024 14:18:17 GMT+08:00	Succeeded	View

Step 4 Select a plan type. It can be:

- **Periodicity**: The plan is executed once in every specified period (week or month). A logical cluster is created or deleted as scheduled as long as it does not conflict with other O&M operations. You can set the weekly time intervals for creating or deleting logical clusters.

Figure 8-16 Setting Plan Type to Periodicity



DWS Cluster [redacted]

Current Nodes 3

Total Capacity 240

Node Specifications Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 80 GB Common I/O

Billing Mode Pay-per-use

* Plan Type ● Periodicity ○ One-time

* Cluster Name Select a logical cluster...

The logical cluster is mainly used for computing acceleration.

Bind Main Logical Cluster -Select-

Bind User -Select-

After a user is bound, the tables created by the user are still stored in the original logical cluster, and the computing logic of the user is switched to the current cluster for execution.

Nodes − 3 +

Time Range Start Date – End Date

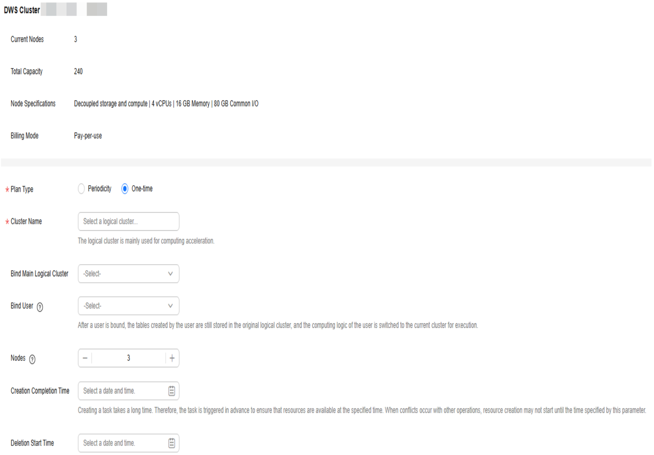
Period Type ● Every Week ○ Every Month

* Time Configuration(UTC) Creation Completion Time -Select- hh:mm Deletion Start Time -Select- hh:mm +

Creating a task takes a long time. Therefore, the task is triggered in advance to ensure that resources are available at the specified time. When conflicts occur with other operations, resource creation may not start until the time specified by this parameter.

- **One-time**: The plan is executed only once in the specified period.

Figure 8-17 Selecting One-time



DWS Cluster [redacted]

Current Nodes 3

Total Capacity 240

Node Specifications Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 80 GB Common I/O

Billing Mode Pay-per-use

* Plan Type ○ Periodicity ● One-time

* Cluster Name Select a logical cluster...

The logical cluster is mainly used for computing acceleration.

Bind Main Logical Cluster -Select-

Bind User -Select-

After a user is bound, the tables created by the user are still stored in the original logical cluster, and the computing logic of the user is switched to the current cluster for execution.

Nodes − 3 +

Creation Completion Time Select a date and time.

Creating a task takes a long time. Therefore, the task is triggered in advance to ensure that resources are available at the specified time. When conflicts occur with other operations, resource creation may not start until the time specified by this parameter.

Deletion Start Time Select a date and time.

Step 5 Click **OK**. In the scheduled plan list, you can view the plan details and next execution time.

NOTE

To avoid affecting services and ensure resources are available at the scheduled time, the plan may be skipped if it conflict with O&M operations, and may be executed about 20 minutes earlier than planned if the cluster creation is time-consuming.

----End

Configuring an Auto Scaling Plan

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

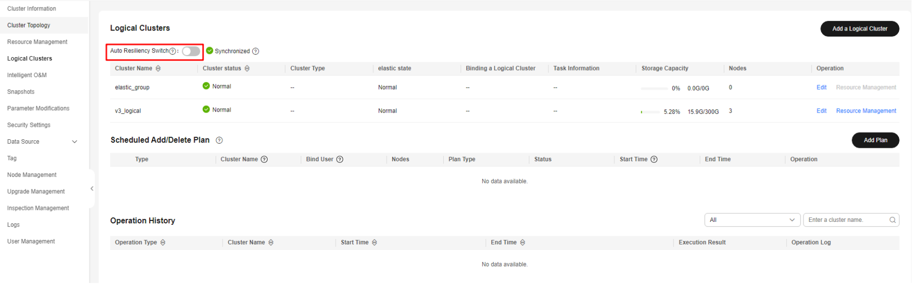
Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 Click **Logical Clusters** on the left to access the page for cluster management. Because once you have created a storage-compute decoupled cluster, it will be in logical cluster mode.

Step 4 Click **Auto Resiliency Switch** to enable the auto scaling function for logical clusters.

Enabling this function allows the system to efficiently manage node resources by detecting service demand and creating or deleting read-only elastic logical clusters as needed. This means that unused nodes are free of charge and the system can adapt to peak and off-peak hours seamlessly.

Figure 8-18 Enabling auto scaling for logical clusters



Logical Clusters

Auto Resiliency Switch: ○ ✓ Synchronized

Cluster Name	Cluster status	Cluster Type	elastic state	Binding a Logical Cluster	Task Information	Storage Capacity	Nodes	Operation
elastic_group	✓ Normal	--	Normal	--	--	0% 0.0G/0G	0	Edit Resource Management
v3_logical	✓ Normal	--	Normal	--	--	5.28% 15.9G/300G	3	Edit Resource Management

Scheduled Add/Delete Plan

Type	Cluster Name	Bind User	Nodes	Plan Type	Status	Start Time	End Time	Operation

No data available.

Operation History

All | Enter a cluster name.

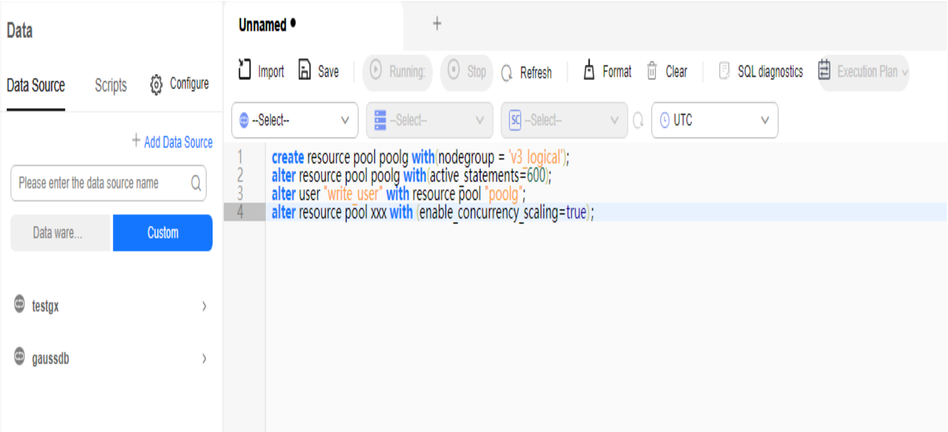
Operation Type	Cluster Name	Start Time	End Time	Execution Result	Operation Log

No data available.

Step 5 After auto scaling is enabled, run SQL statements after completing the operation described in **Using the SQL Editor to Connect to a Cluster** to create and set a resource pool, and bind a user to the resource pool to enable auto scaling for the specified resource pool.

```
create resource pool poolg with(nodegroup = 'v3_logical');
alter user "write_user" with resource pool "poolg";
alter resource pool xxx  with (enable_concurrency_scaling=true);
```

Figure 8-19 Setting a resource pool



```
create resource pool poolg with(nodegroup = 'v3_logical');
alter resource pool poolg with(active_statements=600);
alter user "write_user" with resource pool "poolg";
alter resource pool xxx with (enable_concurrency_scaling=true);
```

----End

8.6.5 Tutorial: Converting a Physical Cluster That Contains Data into a Logical Cluster

Scenario

A large database cluster usually contains a large amount of data put in different tables. With the **resource management** feature, you can create resource pools to isolate the resources of different services. Different service users can be allocated to different resource pools to reduce resource (CPU, memory, I/O, and storage) competition between services.

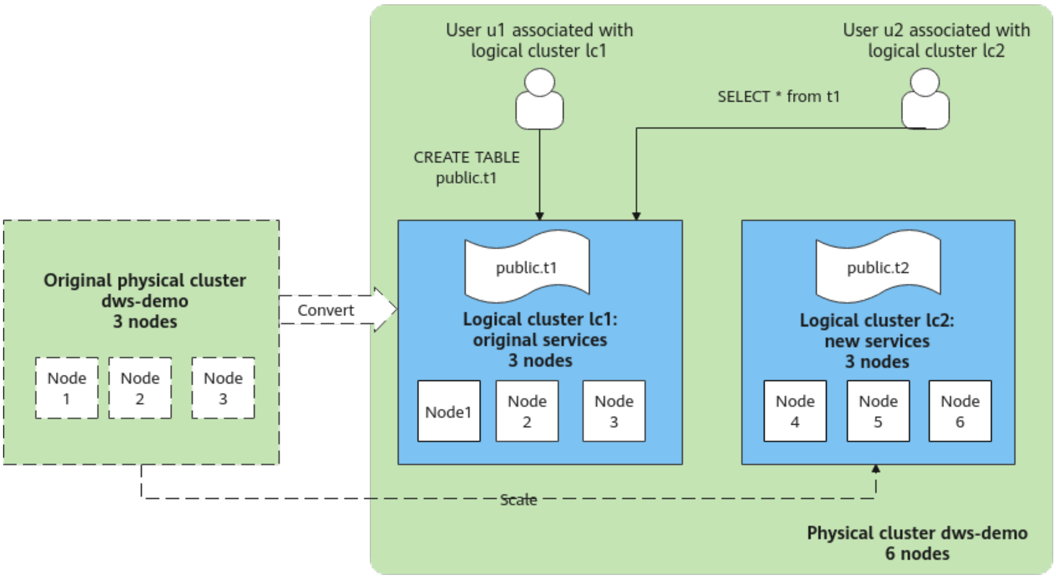
As the service scale grows, the number of services in the cluster system also increases. Creating multiple resource pools becomes less effective in controlling resource competition. GaussDB(DWS) uses the distributed architecture and its data is distributed on multiple nodes. Each table is distributed across all DNs in the cluster, an operation on a data table may involve all DNs, which increases network loads and system resource consumption. To solve this problem, scale-out is not effective. You are advised to divide a GaussDB(DWS) cluster into multiple logical clusters.

You can create a separate logical cluster and assign new services to it. This way, new services have little impact on existing services. Also, if the service scale in existing logical clusters grows, you can scale out the existing logical clusters.

Figure 8-20 shows an example. The original service data tables of a company are stored in the original physical cluster **dws-demo** (in green). After services are switched over to the logical cluster **lc1** (in blue), a new logical cluster **lc2** is added to the physical cluster through scale-out. The original service data tables are switched to logical cluster **lc1**, and new service data tables are written to logical cluster **lc2**. In this way, the data of old and new services is isolated. User **u2** associated with logical cluster **lc2** can access the tables of logical cluster **lc1** across logical clusters after authorization.

- **Cluster scale**: Scale out the original physical cluster from three nodes to six nodes and split it into two logical clusters.
- **Service isolation**: New and old service data is isolated in different logical clusters.

Figure 8-20 Accessing data across logical clusters


User u1 associated with logical cluster lc1
User u2 associated with logical cluster lc2
SELECT * from t1
CREATE TABLE public.t1
Original physical cluster dws-demo 3 nodes
Node 1
Node 2
Node 3
Convert
public.t1
Logical cluster lc1: original services 3 nodes
Node1
Node 2
Node 3
public.t2
Logical cluster lc2: new services 3 nodes
Node 4
Node 5
Node 6
Scale
Physical cluster dws-demo 6 nodes

Creating a Cluster and Preparing Table Data

Step 1 Create a cluster. For details, see **Creating a GaussDB(DWS) Storage-Compute Coupled Cluster**.

Step 2 After connecting to the database, create table **name** as the system administrator **dbadmin** and insert two data records into the table.

```
CREATE TABLE name (id int, name varchar(20));
INSERT INTO name VALUES (1,'joy'),(2,'lily');
```

----End

Converting to Logical Cluster lc1

NOTICE

During the conversion, you can run simple DML statements, such as adding, deleting, modifying, and querying data. Complex DDL statements, such as operations on database objects, will block services. You are advised to perform the conversion during off-peak hours.

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters > Dedicated Cluster**. Click the name of a cluster to go to the **Cluster Information** page.

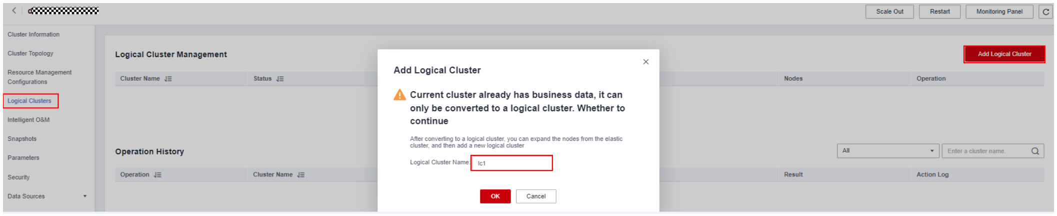
Step 2 Toggle on the **Logical Cluster** switch.

Step 3 In the navigation pane on the left, choose **Logical Clusters**.

Step 4 Click **Add Logical Cluster** in the upper right corner, enter the logical cluster name **lc1**, and click **OK**.

During the switchover, the current cluster is unavailable. Wait for about 2 minutes (the conversion time varies depending on the service data volume). If **lc1** is displayed on the logical cluster page, the conversion is successful.

Figure 8-21 Adding a logical cluster



Add Logical Cluster

Current cluster already has business data, it can only be converted to a logical cluster. Whether to continue

After converting to a logical cluster, you can expand the nodes from the elastic cluster, and then add a new logical cluster

Logical Cluster Name: lc1

----End

Adding Nodes to the elastic_group Cluster

Step 1 Return to the **Cluster Management** page. In the **Operation** column of the cluster, choose **More** > **Scale Node** > **Scale Out**.

Step 2 Set **New Nodes** to **3**. Enable **Online Scale-out**. Set **elastic_group** as the target logical cluster. Confirm the settings, select the confirmation check box, and click **Next: Confirm**.

Step 3 Click **Next: Confirm**, and then click **OK**.

Wait for about 10 minutes until the scale-out is successful.

----End

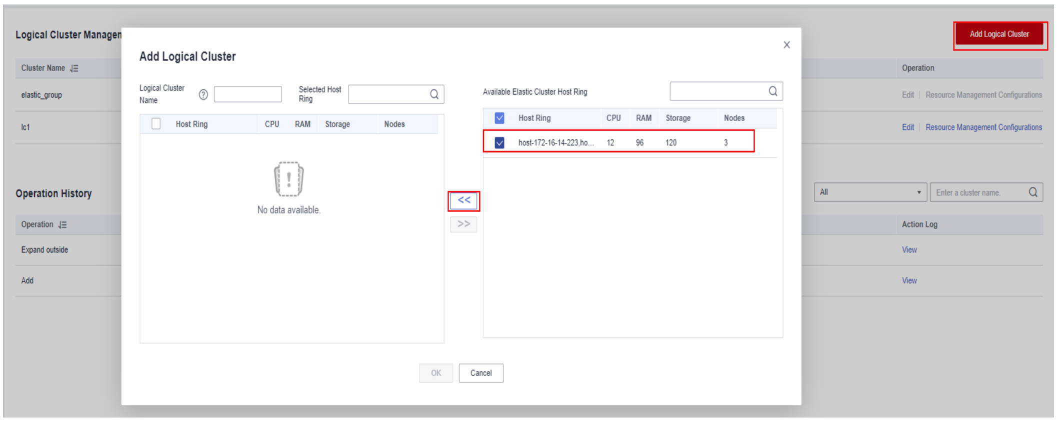
Adding Logical Cluster lc2

Step 1 On the **Cluster Management** page, click the name of a cluster to go to the cluster details page. In the navigation pane, choose **Logical Clusters**.

Step 2 Click **Add Logical Cluster** in the upper right corner, select three nodes from the right pane to add to the left pane, enter the logical cluster name **lc2**, and click **OK**.

After about 2 minutes, the logical cluster is successfully added.

Figure 8-22 Adding a logical cluster



Add Logical Cluster

Logical Cluster Name:

Selected Host Ring:

☐	Host Ring	CPU	RAM	Storage	Nodes

No data available.

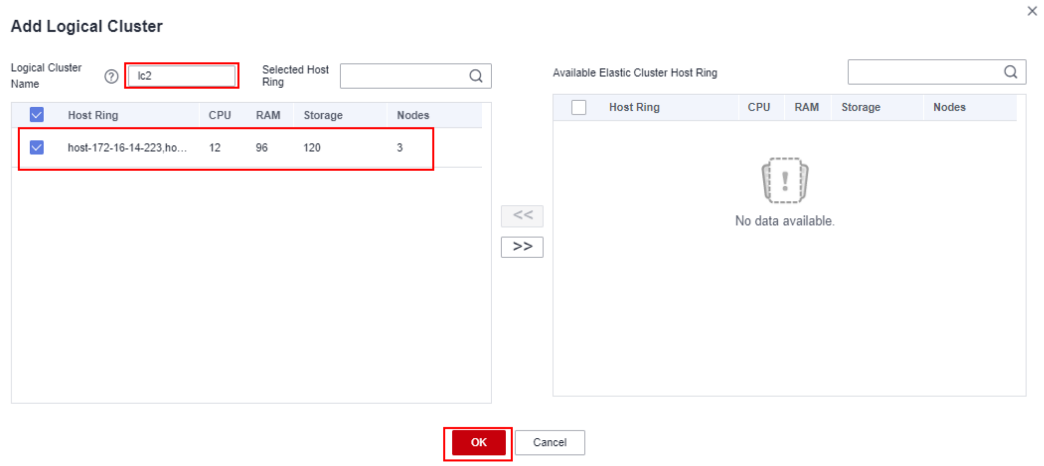
Available Elastic Cluster Host Ring

☑	Host Ring	CPU	RAM	Storage	Nodes
☑	host-172-16-14-223,ho...	12	96	120	3

Cluster Name	Operation
elastic_group	Edit | Resource Management Configurations
lc1	Edit | Resource Management Configurations

Operation	Action Log
Expand outside	View
Add	View

Figure 8-23 Selecting a host ring



Logical Cluster Name: lc2

☑	Host Ring	CPU	RAM	Storage	Nodes
☑	host-172-16-14-223,ho...	12	96	120	3

----End

Creating Logical Clusters, Associating Them with Users, and Querying Data Across Logical Clusters

Step 1 Connect to the database as the system administrator and run the following SQL statement to query the original service table **name**.

Verify that service data can be queried after the conversion.

```
SELECT * FROM name;
```

Step 2 Create logical clusters **lc1** and **lc2** for **u1** and **u2**, respectively.

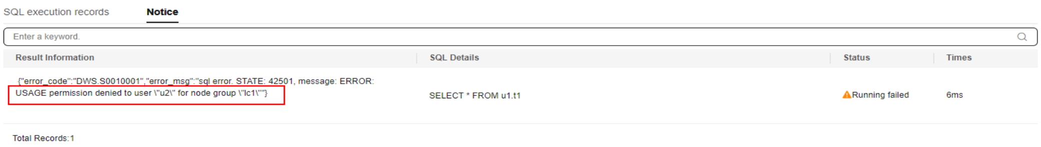
```
CREATE USER u1 NODE GROUP "lc1" PASSWORD '{password}';
CREATE USER u2 NODE GROUP "lc2" PASSWORD '{password}';
```

Step 3 Log in to the database as user **u1**, create table **u1.t1**, insert two data records into the table, and grant user **u2** the permission to access the table.

```
CREATE TABLE u1.t1 (id int, name varchar(20));
INSERT INTO u1.t1 VALUES (1,'joy'),(2,'lily');
GRANT USAGE ON SCHEMA u1 TO u2;
GRANT SELECT ON TABLE u1.t1 TO u2;
```

Step 4 Log in to the database as user **u2** and query data in the original service table **t1**. A message is displayed, indicating that you do not have the permission to access logical cluster **lc1**. The result shows that even if user **u1** has authorized user **u2** to access the table, the table cannot be accessed because it is in different logical clusters. This proves that data is isolated between logical clusters.

```
SELECT * FROM u1.t1;
```



Result Information	SQL Details	Status	Times
{"error_code":"DWS.S0010001","error_msg":"sql error. STATE: 42501, message: ERROR: USAGE permission denied to user \"u2\" for node group \"lc1\""}	SELECT * FROM u1.t1	⚠Running failed	6ms

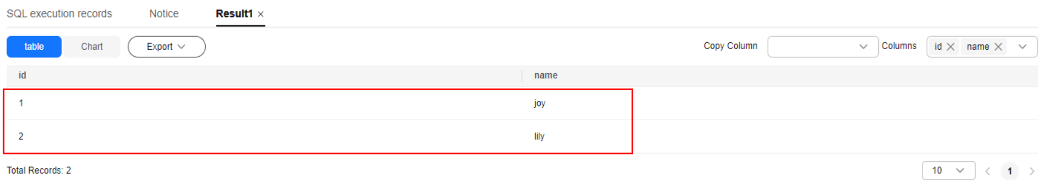
Total Records:1

Step 5 Switch back to system administrator **dbadmin** and grant the access permission of logical cluster **lc1** to user **u2**.

```
GRANT USAGE ON NODE GROUP lc1 TO u2;
```

Step 6 Switch to user **u2** and query the **t1** table. This proves that the user bound to logical cluster **lc2** can query the original service table **t1** across logical clusters. In this way, data is shared between logical clusters.

```
SELECT * FROM u1.t1;
```



id	name
1	joy
2	lily

Total Records: 2

----End

8.6.6 Tutorial: Dividing a New Physical Cluster into Logical Clusters

Scenario

This section describes how to divide a new six-node physical cluster (having no service data) into two logical clusters. If your physical cluster already has service data, perform operations by referring to **Tutorial: Converting a Physical Cluster That Contains Data into a Logical Cluster**.

Prerequisites

Create a six-node cluster. For details, see **Creating a GaussDB(DWS) Storage-Compute Coupled Cluster**.

Dividing a Cluster into Logical Clusters

Step 1 On the **Cluster Management** page, click the name of a cluster to go to the cluster details page. In the navigation pane, choose **Logical Clusters**.

Step 2 Click **Add Logical Cluster** in the upper right corner, select a host ring (three nodes) on the right, add it to the list on the left, enter the logical cluster name **lc1**, and click **OK**.

After about 2 minutes, the logical cluster is added.

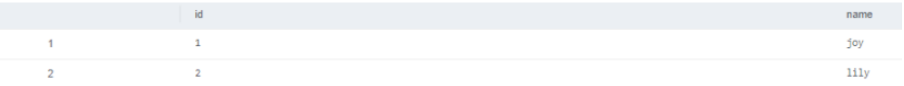
Step 3 Repeat the preceding steps to create the second logical cluster **lc2**.

----End

Creating Logical Clusters, Associating Them with Users, and Querying Data Across Logical Clusters

Step 1 Connect to the database as system administrator **dbadmin** and run the following SQL statement to check whether the logical cluster is created:

```
SELECT group_name FROM PGXC_GROUP;
```



	id	name
1	1	joy
2	2	lily

Step 2 Create users **u1** and **u2** and associate them with logical clusters **lc1** and **lc2**, respectively.

```
CREATE USER u1  NODE GROUP "lc1" password '{password}';
CREATE USER u2  NODE GROUP "lc2" password '{password}';
```

Step 3 Switch to user **u1**, create table **t1**, and insert data into the table.

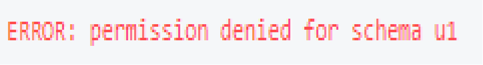
```
SET ROLE u1 PASSWORD '{password}';
CREATE TABLE u1.t1 (id int);
INSERT INTO u1.t1 VALUES (1),(2);
```

Step 4 Switch to user **u2**, create table **t2**, and insert data into the table.

```
SET ROLE u2 PASSWORD '{password}';
CREATE TABLE u2.t2 (id int);
INSERT INTO u2.t2 VALUES (1),(2);
```

Step 5 Query the **u1.t1** table as user **u2**. The command output indicates that the user does not have the permission.

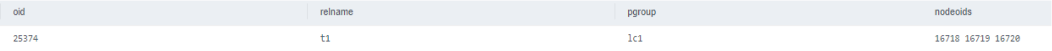
```
SELECT * FROM u1.t1;
```



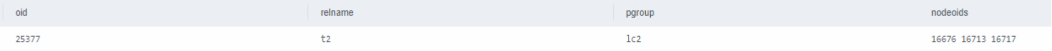
```
ERROR: permission denied for schema u1
```

Step 6 Switch back to the system administrator **dbadmin** and query the **u1.t1** and **u2.t2** tables, which are created in clusters **lc1** and **lc2**, respectively, corresponding to two services. In this way, data is isolated based on logical clusters.

```
SET ROLE dbadmin PASSWORD '{password}';
SELECT p.oid,relname,pgroup,nodeoids FROM pg_class p LEFT JOIN pgxc_class pg ON p.oid = pg.pcrelid
WHERE p.relname = 't1';
SELECT p.oid,relname,pgroup,nodeoids FROM pg_class p LEFT JOIN pgxc_class pg ON p.oid = pg.pcrelid
WHERE p.relname = 't2';
```



oid	relname	pgroup	nodeoids
25374	t1	lc1	16718 16719 16720



oid	relname	pgroup	nodeoids
25377	t2	lc2	16676 16713 16717

Step 7 Grant user **u2** the permissions to access logical cluster **lc1**, schema **u1**, and table **u1.t1**.

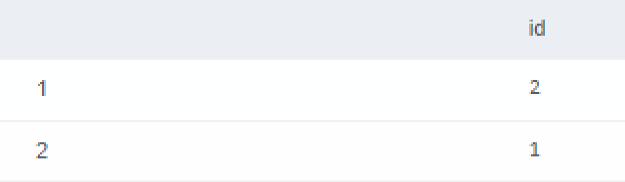
```
GRANT usage ON NODE GROUP lc1 TO u2;
GRANT usage ON SCHEMA u1 TO u2;
GRANT select ON TABLE u1.t1 TO u2;
```

NOTE

Logical clusters implement permission isolation (by node groups) based on physical clusters. To let a user access data across logical clusters, you need to grant the logical cluster (node-group layer) permissions, schema permissions, and table permissions to the user in sequence. If no logical cluster permissions are granted, the error message "permission denied for node group xx" will be displayed.

Step 8 Switch to user **u2** and query the **u1.t1** table. The query is successful. The logical cluster implements data isolation and allows cross-logical cluster access after user authorization.

```
SET ROLE u2 PASSWORD '{password}';
SELECT * FROM u1.t1;
```



	id
1	2
2	1

----End

8.6.7 Tutorial: Setting a Read-Only Logical Cluster and Binding It to a User

Scenario

If your workloads vary greatly in different periods of time, a three-node cluster may be unable to handle all the throughput during peak hours; but a six-node cluster may be too large, wasting resources and increasing costs. In this case, you can follow this tutorial and the instructions in **Elastically Adding or Deleting a Logical Cluster** to use only three nodes during off-peak hours at night, six nodes during daytime, and nine nodes during peak hours.

This tutorial describes how to configure a new logical cluster (without service data) as read-only and switch some users to the cluster. In this way, tables created by those users are still in the original Node Group, but the computing logic is switched to the read-only logical cluster.

Prerequisites

A six-node cluster has been created and divided into two logical clusters: **v3_logical** and **lc1**. The **lc1** cluster has no service data. For details, see **Creating a GaussDB(DWS) Storage-Compute Decoupled Cluster**.

Configuring a Read-Only Logical Cluster and Switching Users to the Cluster

Step 1 Connect to the database as system administrator **dbadmin** and run the following SQL statement to check whether the logical cluster is created:

```
SELECT group_name FROM PGXC_GROUP;
```

Step 2 Set logical cluster **lc1** to be read-only.

```
SET xc_maintenance_mode=on;
ALTER NODE GROUP lc1 SET READ ONLY;
SET xc_maintenance_mode=off;
```

Step 3 Create a user.

```
create user testuser password 'testuser12#$%';
```

Step 4 Bind the user to the logical cluster **lc1**. Replace variables in the following statements (such as **testuser** and **lc1**) as needed.

Find the NodeGroup of the user. If a record can be found, set the record to the **default_storage_nodegroup** of the user so that the tables created by the user will still be in the original Node Group. If no records are found, directly run the two ALTER statements in the end.

```
SELECT nodegroup FROM pg_user WHERE usename='testuser';
ALTER USER testuser SET default_storage_nodegroup='nodegroup'; // Replace nodegroup with the node
group name obtained in the preceding SQL statement.
```

Bind the user to the new read-only logical cluster. In this way, the computing logic of the user is switched to the read-only logical cluster for execution.

```
ALTER USER testuser NODE GROUP lc1;
ALTER USER testuser SET enable_cudesc_streaming=ON;
```

----End

8.7 Modifying GUC Parameters of the GaussDB(DWS) Cluster

After a cluster is created, you can modify the cluster's database parameters as required. On the GaussDB(DWS) console, you can configure common database parameters. For details, see **Modifying Parameters**. You can also view the parameter modification history. For details, see **Viewing Parameter Change History**. Click **Export** to export all parameter settings of the cluster. You can run SQL commands to view or set other database parameters. For details, see **Configuring GUC Parameters**.

Prerequisites

You can modify parameters only when no task is running in the cluster.

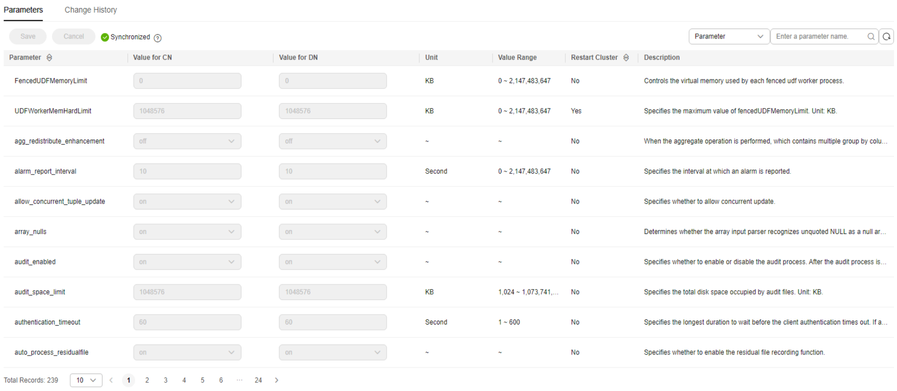
Modifying Parameters

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 Click the **Parameters** tab and modify the parameter values. Then click **Save**.

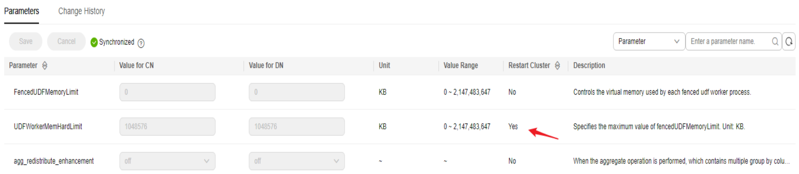


Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
FencedUDFMemoryLimit	0	0	KB	0 ~ 2,147,483,647	No	Controls the virtual memory used by each fenced udf worker process.
UDFWorkerMemHardLimit	1048576	1048576	KB	0 ~ 2,147,483,647	Yes	Specifies the maximum value of fencedUDFMemoryLimit. Unit: KB.
agg_redistribute_enhancement	off	off	~	~	No	When the aggregate operation is performed, which contains multiple group by colu...
alarm_report_interval	10	10	Second	0 ~ 2,147,483,647	No	Specifies the interval at which an alarm is reported.
allow_concurrent_tuple_update	on	on	~	~	No	Specifies whether to allow concurrent update.
array_nulls	on	on	~	~	No	Determines whether the array input parser recognizes unquoted NULL as a null ar...
audit_enabled	on	on	~	~	No	Specifies whether to enable or disable the audit process. After the audit process is...
audit_space_limit	1048576	1048576	KB	1,024 ~ 1,073,741,...	No	Specifies the total disk space occupied by audit files. Unit: KB.
authentication_timeout	60	60	Second	1 ~ 600	No	Specifies the longest duration to wait before the client authentication times out. If a...
auto_process_residualfile	on	on	~	~	No	Specifies whether to enable the residual file recording function.

Total Records: 239 10 < 1 2 3 4 5 6 ··· 24 >

Step 5 In the **Modification Preview** dialog box, confirm the modifications and click **Save**.

Step 6 You can determine whether you need to restart the cluster after parameter modification based on the **Restart Cluster** column.



Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
FencedUDFMemoryLimit	0	0	KB	0 ~ 2,147,483,647	No	Controls the virtual memory used by each fenced udf worker process.
UDFWorkerMemHardLimit	1048576	1048576	KB	0 ~ 2,147,483,647	Yes	Specifies the maximum value of fencedUDFMemoryLimit. Unit: KB.
agg_redistribute_enhancement	off	off	-	-	No	When the aggregate operation is performed, which contains multiple group by colu...

NOTE

- If cluster restart is not required for a parameter, the parameter modification takes effect immediately.
- If cluster restart is required for parameter modifications to take effect, the new parameter values will be displayed on the page after the modification, but will not take effect until the cluster is restarted. Before a restart, the cluster status is **To be restarted**, and some O&M operations are disabled.

----End

Viewing Parameter Change History

Perform the following steps to view the parameter modification history and check whether the modifications have taken effect:

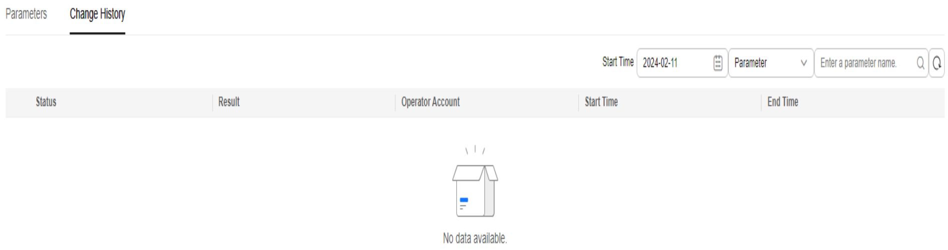
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 Click the **Modify Records** tab.



Status	Result	Operator Account	Start Time	End Time

No data available.

NOTE

- If a parameter can take effect immediately after modification, its status will change to **Synchronized** after you modify it.
- If a parameter can take effect only after a cluster restart, its status will change to **To be restarted** after you modify it. You can click the expansion icon on the left to view the parameters that have not taken effect. After the cluster is restarted, the status of the record will change to **Synchronized**.

Step 5 By default, only the change history within a specified period is displayed. To check the entire change history of a parameter, search for it in the search box in the upper right corner.

----End

Exporting the Parameter List

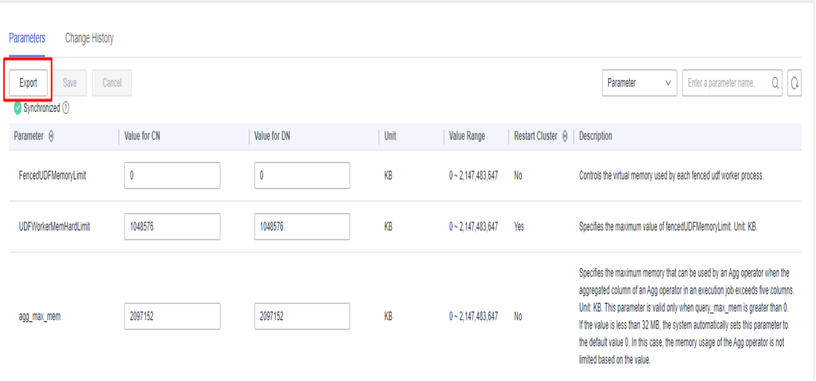
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 Click **Parameters** and click **Export**. You can export cluster configuration parameters.

Figure 8-24 Exporting parameter settings



Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
FencedUDFMemoryLimit	0	0	KB	0 ~ 2,147,483,647	No	Controls the virtual memory used by each fenced udf worker process.
UDFWorkerMemHardLimit	1048576	1048576	KB	0 ~ 2,147,483,647	Yes	Specifies the maximum value of fencedUDFMemoryLimit. Unit: KB.
agg_max_mem	2097152	2097152	KB	0 ~ 2,147,483,647	No	Specifies the maximum memory that can be used by an Agg operator when the aggregated column of an Agg operator in an execution job exceeds five columns. Unit: KB. This parameter is valid only when query_max_mem is greater than 0. If the value is less than 32 MB, the system automatically sets this parameter to the default value 0. In this case, the memory usage of the Agg operator is not limited based on the value.

----End

Parameter Description

There are a large number of database parameters. You can search for and view the parameters on the **Parameter Modification** page. For details, see **Modifying Parameters**. The default values of the parameters are for reference only. For more information, see **Setting GUC Parameters**.

8.8 Managing GaussDB(DWS) Tags

8.8.1 Overview

A tag is a key-value pair customized by users and used to identify cloud resources. It helps users to classify and search for cloud resources.

Tags are composed of key-value pairs.

- A key in a tag can have multiple values.
- A cloud resource must have a unique key.

On GaussDB(DWS), after creating a cluster, you can add identifiers to items such as the project name, service type, and background information using tags. If you use tags in other cloud services, you are advised to create the same tag key-value pairs for cloud resources used by the same business to keep consistency.

GaussDB(DWS) supports the following tags:

- Resource tags

  Non-global tags created on GaussDB(DWS)

- Predefined tags

  Predefined tags created on Tag Management Service (TMS). Predefined tags are global tags.

  For details about predefined tags, see the *Tag Management Service User Guide*.

On GaussDB(DWS), tags can be added to the following resources:

- Cluster

  Tags can be added to a cluster when the cluster is being created or after it is successfully created. You can search for the cluster in the cluster list using tags.

  Each cluster can have a maximum of 20 tags.

  After you add tags to a cluster and then create a snapshot for the cluster, the tags cannot be restored if you use the snapshot to restore the cluster. Instead, you need to add tags again.

  When a cluster is deleted, non-predefined tags associated with the cluster are also deleted. Predefined tags need to be deleted on TMS.

8.8.2 Managing Tags

This section describes how to search for clusters based on tags and how to add, modify, and delete tags for clusters.

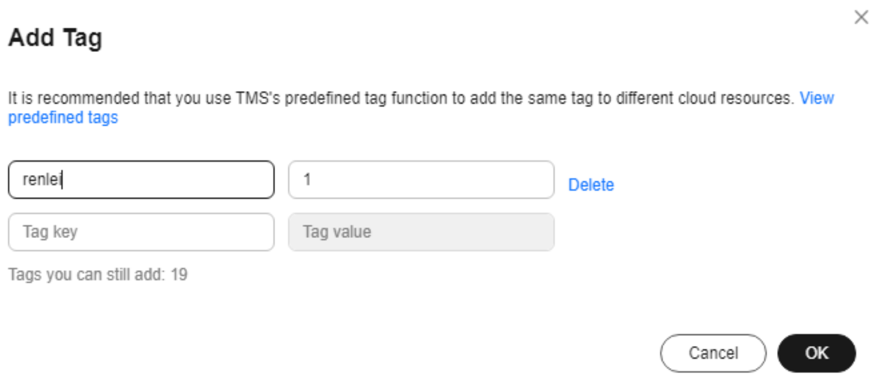
Adding a Tag to a Cluster

Step 1 On the **Clusters** > **Dedicated Clusters** page, click the name of the cluster to which a tag is to be added, and choose **Tag**.

Step 2 Click **Add Tag**.

Step 3 Configure tag information in the **Add Tag** dialog box. The value of a key cannot be left blank.

Figure 8-25 Adding a tag to a cluster



Add Tag

It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. View predefined tags

renlei | 1

Tags you can still add: 19

Table 8-22 Tag parameters

Parameter	Description	Example Value
Tag key	You can: - Select a predefined tag key or an existing resource tag key from the drop-down list of the text box. **NOTE** To add a predefined tag, you need to create one on TMS and select it from the drop-down list of **Tag key**. You can click **View predefined tags** to enter the **Predefined Tags** page of TMS. Then, click **Create Tag** to create a predefined tag. For more information, see **Creating Predefined Tags** in the *Tag Management Service User Guide*. - Enter a tag key in the text box. A tag key can contain a maximum of 128 characters. It cannot be an empty string, start with **_sys_**, or start or end with a space. Only letters, digits, spaces, and the following characters are allowed: _ . : = + - @ **NOTE** A key must be unique in a given cluster.	key01
Tag value	You can: - Select a predefined tag value or resource tag value from the drop-down list of the text box. - Enter a tag value in the text box. A tag value can contain a maximum of 255 characters, which can be an empty string. It cannot start or end with a space. Only letters, digits, spaces, and the following characters are allowed: _ . : = + - @	value01

Step 4 Click **OK**.

----End

Searching for Clusters Based on Tags

You can quickly locate a tagged cluster using tags.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

Step 3 Click the search box above the cluster list and select the **Resource Tag** filter.

Step 4 Click the tag key to be searched for and select the corresponding tag value. Click the search box again to add more tag filters.

Search by tag supports only the keys and values that exist in the drop-down list. If no tag key or value is available, create a tag for the cluster. For details, see **Adding a Tag to a Cluster**.

Step 5 Click **Search**. The target cluster will be displayed in the cluster list.

----End

Modifying a Tag

Step 1 On the **Clusters** > **Dedicated Clusters** page, click the name of the cluster to which a tag is to be added, and choose **Tag**.

Step 2 Locate the row that contains the tag to be modified, and click **Edit** in the **Operation** column. The **Edit Tag** dialog box is displayed.

Step 3 Enter the new key value in the **Value** text box.

Step 4 Click **OK**.

----End

Deleting a Tag

Step 1 On the **Clusters** > **Dedicated Clusters** page, click the name of the cluster from which a tag is to be deleted, and click **Tag**.

Step 2 Locate the row that contains the tag to be deleted, click **Delete** in the **Operation** column. The **Delete Tag** dialog box is displayed.

Step 3 After confirming that the information is correct, enter **DELETE** or click **Auto Enter** and click **OK** to delete the tag.

----End

8.9 Resetting the Password the GaussDB(DWS) Database Administrator

GaussDB(DWS) allows you to reset the password of the database administrator. If a database administrator forgets their password or the account is locked because the number of consecutive incorrect password attempts reaches the upper limit, the database administrator can reset the password on the **Clusters** > **Dedicated Clusters** page. After the password is reset, the account can be automatically unlocked. You can set the maximum number of incorrect password attempts (10 by default) by configuring the **failed_login_attempts** parameter on the **Parameter** page of the cluster. For details, see **Modifying GUC Parameters of the GaussDB(DWS) Cluster**.

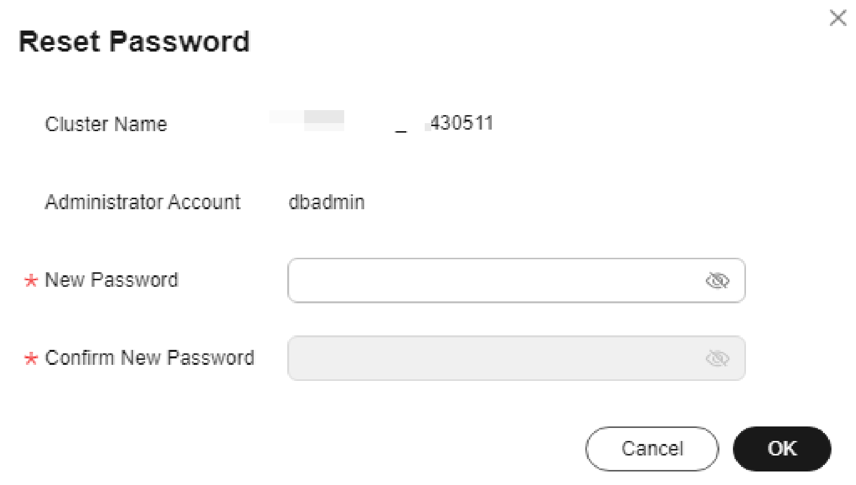
Resetting a Password

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Reset Password**.

Figure 8-26 Password resetting



Reset Password

Cluster Name [redacted] _ 430511

Administrator Account dbadmin

* New Password

* Confirm New Password

Step 4 On the displayed **Reset Password** page, set a new password, confirm the password, and then click **OK**.

The password must:

- Contain 12 to 32 characters.
- Cannot be the username or the username spelled backwards.
- Contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-)
- Pass the weak password check.
- Cannot be the same as the old password and cannot be the reverse of the old password.
- Cannot use a historical password.

NOTE

If the default database administrator account of the cluster is deleted or renamed, password resetting fails.

----End

8.10 Starting, Stopping, and Deleting a GaussDB(DWS) Cluster

Restarting a cluster

If a cluster is in the **Unbalanced** state or cannot work properly, you may need to restart it for restoration. After modifying a cluster's configurations, such as security settings and parameters, manually restart the cluster to make the configurations take effect.

NOTE

If your cluster is in arrears, this function may be unavailable. Please top up your account in time.

Impact on the System

- A cluster cannot provide services during the restart. Therefore, before the restart, ensure that no task is running and all data is saved.

  If the cluster is processing service data, such as importing data, querying data, creating snapshots, or restoring snapshots, cluster restarting will cause file damage or restart failure. You are advised to stop all cluster tasks before restarting the cluster.

  View the **Session Count** and **Active SQL Count** metrics to check whether the cluster has active events. For details, see **Viewing GaussDB(DWS) Cluster Monitoring Information on Cloud Eye**.
- The time required for restarting a cluster depends on the cluster scale and services. Generally, it takes about 3 minutes to restart a cluster. The duration does not exceed 20 minutes.
- If the restart fails, the cluster may be unavailable. Try again later or contact technical support.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

Step 3 In the **Operation** column of the cluster to be restarted, choose **More** > **Restart**.

Step 4 In the dialog box that is displayed, click **Yes**.

Task Information changes to **Restarting**. When **Cluster Status** changes to **Available** again, the cluster is successfully restarted.

----End

Stopping a Cluster

If a cluster is no longer used, you can stop the cluster to bring services offline.

NOTE

- If the current console does not support this feature, contact technical support. Billing resumes after the cluster is started.
- After the cluster is stopped, ECS basic resources (vCPUs and memory) are no longer reserved. When you start the service again, it may fail to be started due to insufficient resources. In this case, wait for a while and try again later.
- For details about cluster billing after the cluster is stopped, see **Stopping Billing**.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 On the **Dedicated Clusters** page, locate the row that contains the target dedicated cluster, click **More** > **Stop** in the **Operation** column.

Step 4 In the dialog box that is displayed, click **Yes**.

The **Task Information** of the cluster changes to **Stopping**. If the **Cluster Status** changes to **Stopped**, the cluster is stopped successfully.

----End

Starting a Cluster

You can start a stopped cluster to restore cluster services.

NOTE

If the current console does not support this feature, contact technical support. Billing resumes after the cluster is started.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 On the **Dedicated Clusters** page, locate the row that contains the target dedicated cluster, click **More** > **Start** in the **Operation** column.

Step 4 In the dialog box that is displayed, click **Yes**.

The **Task Information** of the cluster changes to **Starting**. If the **Cluster Status** changes to **Available**, the cluster is started successfully.

----End

Deleting a Cluster

If you do not need to use a cluster, perform the operations in this section to delete it.

NOTE

- If your cluster is in arrears, this function may be unavailable. Please top up your account in time.
- You cannot delete a cluster that is either read-only or being scaled. Wait until the scaling process is finished or the read-only state is canceled before attempting to delete it.
- If a cluster has DR tasks, the cluster cannot be deleted. You need to delete the DR tasks and then delete the cluster.

Impact on the System

Deleted clusters cannot be recovered. Additionally, you cannot access user data and automated snapshots in a deleted cluster because the data and snapshots are automatically deleted. If you delete a cluster, its manual snapshots will not be deleted.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click 📍 in the upper left corner of the management console to select a region.

Step 3 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be deleted.

Step 4 In the row of a cluster, choose **More** > **Delete**.

Step 5 In the displayed dialog box, confirm the deletion. You can determine whether to perform the following operations:

- Create a snapshot for the cluster.

  If the cluster status is normal, click **Create Snapshot**. On the snapshot list page, click **Create Snapshot** to create a snapshot for the cluster to be deleted. For details, see **Manual Snapshots**. In the row of a cluster, choose **More** > **Delete**.

- Delete associated resources.
  - Release the EIP bound to a cluster.

    If an EIP is bound to the cluster, you are advised to select **EIP** to release the EIP of the cluster to be deleted. If you do not release the EIP, you can bind it to another cluster or cloud resource and it will be billed based on the EIP pricing rule of VPC.

  - Delete automated snapshots.
  - Delete manual snapshots.

    If you have created a manual snapshot, you can select **Manual Snapshot** to delete it.

Step 6 After confirming that the information is correct, enter **DELETE** or click **Auto Enter** and click **OK** to delete the cluster. The cluster status in the cluster list will change to **Deleting** and the cluster deletion progress will be displayed.

If the cluster to be deleted uses an automatically created security group that is not used by other clusters, the security group is automatically deleted when the cluster is deleted.

----End

8.11 Managing Enterprise Projects

An enterprise project is a cloud resource management mode. Enterprise Management provides users with comprehensive management in cloud-based finance. The Enterprise Management console differs from typical management consoles as it focuses on resource management rather than independent control and configuration of cloud products. It assists enterprises in managing finances within the hierarchy of companies, departments, and projects.

Users who have enabled the Enterprise Project Management service can use it to manage cloud service resources.

Binding an Enterprise Project

You can select an enterprise project during cluster creation to associate it with the cluster. For details, see **Creating a GaussDB(DWS) Storage-Compute Coupled Cluster**. The **Enterprise Project** drop-down list displays the projects you created. In addition, the system has a built-in enterprise project (**default**). If you do not select an enterprise project for the cluster, the default project is used.

Note that the Enterprise Project Management service is still in the OBT. Only users with the OBT permission can set enterprise projects. Common users cannot view the enterprise project information.

During cluster creation, if the cluster is successfully bound to an enterprise project, the cluster will be successfully created. If the binding fails, the system sends an alarm and the cluster fails to be created.

Snapshots of a cluster retain the association between the cluster and its enterprise project. When the cluster is restored, the association is also restored.

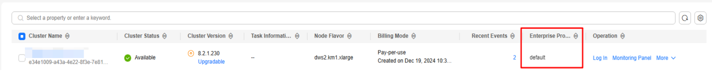
When you delete a cluster, the association between the cluster and its enterprise project is automatically deleted.

Viewing Enterprise Projects

After a cluster is created, you can view the associated enterprise project in the cluster list and **Cluster Information** page. You can query only the cluster resources of the project on which you have the access permission.

- In the cluster list on the **Clusters** page, view the enterprise project to which the cluster belongs.

Figure 8-27 Viewing the enterprise project

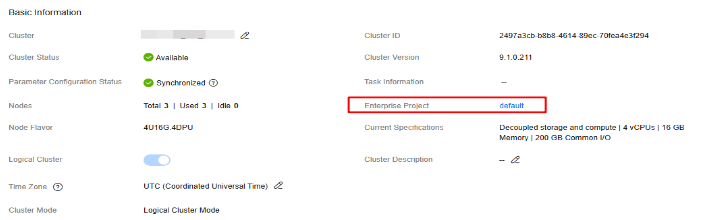


Select a property or enter a keyword.

Cluster Name	Cluster Status	Cluster Version	Task Informati...	Node Flavor	Billing Mode	Recent Events	Enterprise Pro...	Operation
[redacted] e34e1009-a43a-4e22-8f3e-7e81...	Available	8.2.1.230 Upgradable	--	dws2.km1.xlarge	Pay-per-use Created on Dec 19, 2024 10:3...	2	default	Log In Monitoring Panel More

- In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed, on which you can view the enterprise project associated with the cluster. Click the enterprise project name to view and edit it on the Enterprise Management console.

Figure 8-28 Viewing the enterprise project



Basic Information

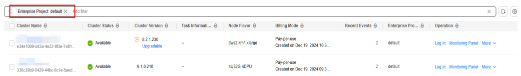
Cluster	[redacted]	Cluster ID	2497a3cb-b8b8-4614-89ec-70fea4e3f294
Cluster Status	Available	Cluster Version	9.1.0.211
Parameter Configuration Status	Synchronized	Task Information	--
Nodes	Total 3 | Used 3 | Idle 0	Enterprise Project	default
Node Flavor	4U16G.4DPU	Current Specifications	Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O
Logical Cluster		Cluster Description	--
Time Zone	UTC (Coordinated Universal Time)		
Cluster Mode	Logical Cluster Mode		

- When querying the resource list of a specified project on the Enterprise Management console, you can also query the GaussDB(DWS) resources.

Searching for Clusters by Enterprise Project

Log in to the GaussDB(DWS) console and click **Clusters** > **Dedicated Clusters**. Click the search box above the cluster list and select **Enterprise Project**. Enter the project name and click the search button to view all clusters associated with the project.

Figure 8-29 Search by enterprise projects



Cluster Name	Cluster Status	Cluster Version	Task Informati...	Node Flavor	Billing Mode	Recent Events	Enterprise Pro...	Operation
[redacted] e34e1009-a43a-4e22-8f3e-7e81...	Available	8.2.1.230 Upgradable	--	dws2.km1.xlarge	Pay-per-use Created on Dec 19, 2024 10:3...	2	default	Log In Monitoring Panel More
[redacted] 336c38b9-5429-44bc-8c1e-5aed...	Available	9.1.0.210	--	4U32G.4DPU	Pay-per-use Created on Dec 19, 2024 09:3...	2	default	Log In Monitoring Panel More

Migrating a Cluster to or Out of an Enterprise Project

A GaussDB(DWS) cluster can be associated with only one enterprise project. After a cluster is created, you can migrate it from its current enterprise project to another one on the Enterprise Management console, or migrate the cluster from another enterprise project to a specified enterprise project. After the migration, the cluster is associated with the new enterprise project. The association between the cluster and the original enterprise project is automatically released. For details, see "Resource Management" > "Managing Enterprise Project Resources" in the *Enterprise Management User Guide*.

Enterprise Project-Level Authorization

If permissions preset in the system cannot meet requirements, you can customize policies and grant the policies to user groups for refined access control. As an independent managed object, the enterprise project can be bound to a user group, and the customized policy can be granted to the user group. This implements refined authorization at the enterprise project level.

Step 1 Log in to the IAM console and create a custom policy.

Refer to the following to create the policy:

- Use the IAM administrator account, that is, the user in the admin user group, because only the IAM administrator has the permissions to create users and user groups and modify user group permissions.
- GaussDB(DWS) is a project-level service, so its **Scope** must be set to **Project-level services**. If this policy is required to take effect for multiple projects, authorization is required to each project.
- Some GaussDB(DWS) policy templates are preconfigured on IAM. When creating a custom policy, you can select one of the following templates and modify the policy authorization statement based on the template:
  - **DWS FullAccess**: all execution permissions for GaussDB(DWS)
  - **DWS ReadOnlyAccess**: read-only permission for GaussDB(DWS)
  - **DWS Administrator**: all execution permissions for GaussDB(DWS)
  - **DWS Database Access**: Users granted this permission can generate temporary database user credentials based on IAM users to connect to databases in the data warehouse clusters.
- You can add permissions corresponding to GaussDB(DWS) operations or RESTful APIs listed in **List of Supported Actions** to the action list in the policy authorization statement, so that the policy can obtain the permissions.

  For example, if **dws:cluster:create** is added to the action list of a policy statement, the policy has the permission to create or restore clusters.
- If you want to use other services, grant related operation permissions on these services. For details, see the help documents of related services.

For example, when creating a data warehouse cluster, you need to configure the VPC to which the cluster belongs. To obtain the VPC list, add permission **vpc:*:get*** to the policy statement.

Policy example:

- Example in which multiple operation permissions are supported

  The following policy has the permissions to create/restore/restart/delete a cluster, set security parameters, and reset passwords.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dws:cluster:create",
                "dws:cluster:restart",
                "dws:cluster:delete",
                "dws:cluster:setParameter",
                "dws:cluster:resetPassword",
                "ecs:*:get*",
                "ecs:*:list*",
                "vpc:*:get*",
                "vpc:*:list*"
            ]
        }
    ]
}
```

- Example of wildcard (*) usage

  The following policy has all operation permissions on GaussDB(DWS) snapshots.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dws:snapshot:*",
                "ecs:*:get*",
                "ecs:*:list*",
                "vpc:*:get*",
                "vpc:*:list*"
            ]
        }
    ]
}
```

Step 2 Click **Enterprise** in the upper right corner of the management console to enter the Enterprise Management console.

Step 3 Choose **Personnel Management > User Group Management** in the left navigation tree. Then, create a user group and add users to it, add the user group to a project, and grant the newly created custom policy to the group so that users in the group can obtain the permissions defined by the policy.

For details, see **Authorizing a User Group to Manage an Enterprise Project** in the *Enterprise Management User Guide*.

----End

9 GaussDB(DWS) Cluster O&M

9.1 Viewing GaussDB(DWS) Cluster Monitoring Information on the Monitoring Panel (DMS)

9.1.1 Database Monitoring Overview

Overview

DMS is provided by GaussDB(DWS) to ensure the fast and stable running of databases. It collects, monitors, and analyzes the disk, network, and OS metric data used by the service database, as well as key performance metric data of cluster running. It also diagnoses database hosts, instances, and service SQL statements based on the collected metrics to expose key faults and performance problems in a database in a timely manner, and guides customers to optimize and resolve the problems.

NOTE

- Database monitoring is supported by 8.1.1.200 and later versions.
- A storage-compute coupled data warehouse (standalone) does not support database monitoring.
- The database monitoring function and Cloud Eye monitor different data sources. In database monitoring, the size of a database is the total disk space used by the database, including the space occupied due to bloating.

Entering the Database Monitoring Page

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

----End

9.1.2 Monitoring Metrics

You can check the status and available resources of a cluster and learn about its real-time resource consumption through the GaussDB(DWS) monitoring items.

Table 9-1 describes GaussDB(DWS) monitoring metrics.

Table 9-1 GaussDB(DWS) monitoring metrics

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Abnormal Job Monitoring Statistics	Query ID	String	30s	Collect data regarding abnormal jobs whose status is **aborted** in the **PGXC_WLM_SESSION_HISTORY** view.	N/A
	Statement executed for exception handling	String			
	Block time before the statement is executed	≥ 0			
	Elapsed time when the statement is executed	≥ 0			
	Total time used by the CPU on the DN when the statement is executed for exception handling	≥ 0			
	CPU usage skew on the DN when the statement is executed for exception handling	≥ 0			
	cgroups used for exception handling during statement execution	String			
	Status of a statement after exception handling	String			
	Exception handling action executed by the statement	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Reason why the statement is processed abnormally	String			
Node Status Statistics	Host name	String	60s	Gather information on the current state and specifics of each ECS host instance (VM) in a cluster.	N/A
	Host status	String			
Instances by Status	Host name	String	60s	Gather information on the current state and specifics of each CN/DN instance in the cluster.	N/A
	Instance type	String			
	Instance role	String			
	Instance status	String			
	Cause of the instance status.	String			
Cluster Status	Cluster status	String	30s	Monitor the cluster status.	N/A
	Whether an primary/ standby switchover has occurred	String			
	Whether redistribution has occurred	String			
	Whether the current cluster is read-only	String			
CPU usage	The default value is **ALL**.	String	30s	Gather CPU usage data from ECS instances (VMs) to monitor node CPU usage. High CPU usage can lead to SQL queuing and slow queries.	85 %
	User-mode CPU time %	≥ 0.0			
	CPU time the process with a negative **nice** value (%)	≥ 0.0			
	Kernel mode time (%)	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	I/O wait time (%)	≥ 0.0			
	Hard interruption time (%)	≥ 0.0			
	Software interrupt time (%)	≥ 0.0			
	Time spent by the virtual CPU in involuntary waiting when the virtual machine manager serves another virtual processor (%)	≥ 0.0			
	Time spent running the virtual processor (%)	≥ 0.0			
	Idle time except for disk waiting operations (%)	≥ 0.0			
	Hyper-threading capability	Yes/No			
	Hyper-threading Enabled (Yes/No)	Yes/No			
	Number of processes in the runnable status	≥ 0			
	Number of processes waiting for I/O completion	≥ 0			
Active Sessions	Database name	String	30s	Collect information on the current cluster's active sessions.	N/A
	Instance name	String			
	Number of all user sessions	≥ 0			
	Distinct username	≥ 0			
	Distinct application name	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Number of queries in the **active** or **fastpathfunctioncall** state.	≥ 0			
Disk Capacity Statistics	Instance name	String	86400s	Collect statistics on the disk space used by each database in the current cluster.	N/A
	Database name	String			
	Database size	≥ 0			
Transaction Status	Database name	String	60s	Gather data on the operational status of databases in the current cluster, such as the number of updated, deleted, and inserted rows, transactions, and deadlocks.	N/A
	Instance name	String			
	Number of rows returned through a global database scan	≥ 0			
	Number of rows returned by querying indexes in the database	≥ 0			
	Number of rows inserted by queries in this database	≥ 0			
	Number of rows updated by queries in this database	≥ 0			
	Number of rows deleted by queries in this database	≥ 0			
	Number of transactions in this database that have been committed	≥ 0			
	Number of transactions in this database that have been rolled back	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Number of deadlocks detected in this database	≥ 0			
	Number of disk blocks read in this database	≥ 0			
	Number of disk blocks found in the buffer cache in the current database, that is, the number of blocks hit in the cache. (This only includes hits in the GaussDB(DWS) buffer cache, not in the file system cache.)	≥ 0			
	Time spent reading data file blocks by backends in this database, in milliseconds	≥ 0.0			
	Time spent reading data file blocks by backends in this database, in milliseconds	≥ 0.0			
	Number of queries canceled due to database recovery conflicts (conflicts that occur only on the standby server).	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Number of temporary files created by queries in this database. This parameter calculates all temporary files (such as sorting or hashing) and ignores the **log_temp_files** setting.	≥ 0			
	Total amount of data written to temporary files by queries in this database. This parameter calculates all temporary files and ignores the **log_temp_files** setting.	≥ 0			
	Database capacity, in bytes.	≥ 0			
	Number of rows returned by global database scanning in a unit time	≥ 0			
	Number of rows returned by querying indexes in the database in a unit time	≥ 0			
	Number of rows inserted through database query in a unit time	≥ 0			
	Number of rows updated through database query in unit time	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Number of rows deleted by querying the database in a unit time	≥ 0			
	Number of transactions that have been submitted in the database per unit time	≥ 0			
	Number of transactions that have been rolled back in the database per unit time	≥ 0			
	Number of deadlocks retrieved in the database per unit time	≥ 0			
File Handle	Name of the disk file system	String	30s	Gather data on the disk inode information of the cluster to monitor inode usage. High inode usage can pose risks.	90 %
	Total inode capacity (unit: KB)	≥ 0			
	Used capacity (unit: KB)	≥ 0			
Node Disk Usage	Name of the disk file system	≥ 0	30s	Track the disk usage of every ECS instance and switch the cluster to read-only mode when it hits 90%.	90 %
	Total space (unit: KB)	≥ 0			
	Used capacity (unit: KB)	≥ 0			
	Available capacity (unit: KB)	≥ 0			
	Disk usage	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
gsar NIC Usage Statistics	Node name	String	30s	Monitor the running status of the gsar NIC.	N/A
	NIC name	String			
	NIC IP address	String			
	Data received by the NIC (unit: KB)	≥ 0			
	Number of packets received by the NIC	≥ 0			
	Average length of received packets (unit: byte)	≥ 0			
	Number of received data packets that are dropped by the NIC	≥ 0			
	Port transmit discard rate	≥ 0.0			
	Data sent by the network adapter (unit: KB)	≥ 0			
	Number of packets sent by the NIC	≥ 0			
	Average length of received packets (unit: byte)	≥ 0			
gsar TCP Statistics	Number of retransmitted TCP packets due to timeout	≥ 0	30s	Monitor the TCP retransmission rate.	Number of retransmitted TCP packets > 0
	Number of sent TCP packets	≥ 0			
	Number of retransmitted TCP packets	≥ 0			
	TCP retransmission rate	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Node Disk I/O Statistics	Disk name (devicename)	String	30s	Keep track of the I/O status of every disk on a node, which is indicated by the data transfer rates and the number of read and write operations. Excessive data transfer rates could affect cluster services.	350 MB/s
	Number of transmissions per second (transferpersecond). The size of each transmission is unknown.	≥ 0.0			
	Amount of data read from the device per second (unit: KB)	≥ 0.0			
	Amount of data written to the device per second (unit: KB)	≥ 0.0			
	Total amount of read data (unit: KB)	≥ 0.0			
	Total amount of written data (unit: KB)	≥ 0.0			
	Number of times that read requests to the device are combined per second	≥ 0.0			
	Number of times that write requests to the device are combined per second	≥ 0.0			
	Number of completed reads per second	≥ 0.0			
	Number of completed writes per second	≥ 0.0			
	Amount of data read per second (unit: KB)	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Amount of data written per second (unit: KB)	≥ 0.0			
	Average data volume of each I/O operation (unit: sector)	≥ 0.0			
	Average request queue length	≥ 0.0			
	Average waiting time for each I/O request (unit: ms)	≥ 0.0			
	Average processing time for each I/O request (unit: ms)	≥ 0.0			
	Percentage of the time when the I/O queue is not empty (I/O operation time divided by the total time)	≥ 0.0			
Instance Memory Monitoring Statistics	Instance name	String	60s	Monitor instance and dynamic memory, gather memory usage statistics for each CN and DN. If the instance memory usage goes beyond the threshold, there may not be enough instance memory in the cluster.	85 %
	Memory size occupied by the instance	≥ 0.0			
	Memory size used by a process	≥ 0.0			
	Maximum dynamic memory	≥ 0.0			
	Used dynamic memory	≥ 0.0			
	Dynamic peak memory	≥ 0.0			
	Maximum dynamic shared memory context	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Dynamic peak value of the shared memory context	≥ 0.0			
	Maximum shared memory	≥ 0.0			
	Used shared memory	≥ 0.0			
	Maximum memory allowed by column store	≥ 0.0			
	Memory used in column store	≥ 0.0			
	Maximum memory that can be used by the communication library	≥ 0.0			
	Used memory size of the communication library	≥ 0.0			
	Peak memory usage of the communication library	≥ 0.0			
	Maximum memory that can be used by top SQLs to record historical job monitoring information	≥ 0.0			
	Peak memory usage of the top SQL that records historical job monitoring information	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Memory used by top SQLs to record historical job monitoring information	≥ 0.0			
	Other used memory	≥ 0.0			
	Memory size occupied by pooler connections	≥ 0.0			
	Memory size occupied by pooler idle connections	≥ 0.0			
	Memory size used by column-store compression and decompression	≥ 0.0			
	Memory reserved for the UDFWorker process	≥ 0.0			
	Memory size used by the MMAP	≥ 0.0			
Instance Resource Statistics	Instance name	String	60s	Gather resource usage statistics for each instance in a cluster.	85 %
	Read the value (CPU usage %) in **postmaster.pID/ cm_server.pID/ gtm.pID/etcd.pID**.	≥ 0.0			
	Read the value (memory usage %) in **postmaster.pID/ cm_server.pID/ gtm.pID/etcd.pID**.	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Instance Disk Size Statistics	Instance name	String	86400s	Collect statistics on instance disk usage and monitor it.	85 %
	Storage location	String			
	Disk space used by all databases on the current instance	≥ 0			
Node Memory Statistics	Size of all available RAMs, that is, the remaining physical memory minus the reserved bits and kernel usage (unit: KB)	≥ 0	30s	Gather memory usage statistics for the ECS instance where the cluster is located. This metric tracks statistics on the VM OS-level memory, which differs from the instance memory.	70 %
	Unused memory in the system. The value is **lowfree**+**highfree** (unit: KB).	≥ 0			
	Size of the cache used for the block device (unit: KB)	≥ 0			
	Size of the file buffer (unit: KB)	≥ 0			
	Total swap space (unit: KB)	≥ 0			
	Size of the RAM memory temporarily stored in the swap file (unit: KB)	≥ 0			
	Memory size of the virus-infected page (unit: KB)	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Network Status Statistics	NIC name	String	30s	Collect NIC status for each node in the cluster to monitor lost packets on the cluster NIC and network throughput.	N/A
	NIC status (up/down)	up/down			
	NIC speed (1,000 Mbit/s or 100 Mbit/s)	≥ 0			
	Total data received by the NIC (unit: byte)	≥ 0			
	Number of packets received by the NIC	≥ 0			
	Total number of NIC receiving errors	≥ 0			
	Number of received data packets that are dropped by the NIC	≥ 0			
	Number of FIFO buffer errors during reception	≥ 0			
	Number of received packet frame errors	≥ 0			
	Number of received compressed data packets	≥ 0			
	Number of received multicast frames	≥ 0			
	Total data sent by the NIC (unit: byte)	≥ 0			
	Total number of packets sent by the NIC	≥ 0			
	Total number of NIC sending errors	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Total number of data packets discarded by the NIC during transmission	≥ 0			
	Number of FIFO buffer errors during sending	≥ 0			
	Number of collisions detected on the sending interface	≥ 0			
	Number of carrier losses detected by the device driver during transmission	≥ 0			
	Number of sent compressed data packets	≥ 0			
	Specifies whether NIC multi-queue is supported.	Yes/No			
	NIC multi-queue is enabled.	Yes/No			
	Specifies the CPU affinity of a multi-queue NIC.	String			
	Indicates whether the NIC works in duplex mode.	String			
	Network speed	≥ 0.0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Node SQL Statistics	Node name	String	60s	Use the **PGXC_SQL_COUNT** view to query the number of running SQL statements on each node and monitor the cluster's QPS.	N/A
	Username	String			
	Number of **SELECT** statements	≥ 0			
	Number of **UPDATE** statements	≥ 0			
	Number of **INSERT** statements	≥ 0			
	Number of **DELETE** statements	≥ 0			
	Number of **MERGEINTO** statements	≥ 0			
	Number of **DDL** statements	≥ 0			
	Number of **DML** statements	≥ 0			
	Number of **DCL** statements	≥ 0			
	Total response time of **SELECT** statements	≥ 0			
	Average response time of **SELECT** statements	≥ 0			
	Maximum response time of **SELECT** statements	≥ 0			
	Minimum response time of **SELECT** statements	≥ 0			
	Total response time of **UPDATE** statements	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Average response time of **UPDATE** statements	≥ 0			
	Maximum response time of **UPDATE** statements	≥ 0			
	Minimum response time of **UPDATE** statements	≥ 0			
	Total response time of **DELETE** statements	≥ 0			
	Average response time of **DELETE** statements	≥ 0			
	Maximum response time of **DELETE** statements	≥ 0			
	Minimum response time of **DELETE** statements	≥ 0			
	Total response time of **INSERT** statements	≥ 0			
	Average response time of **INSERT** statements	≥ 0			
	Maximum response time of **INSERT** statements	≥ 0			
	Minimum response time of **INSERT** statements	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Difference between the numbers of **SELECT** statements at two time points	≥ 0			
	Difference between the numbers of **UPDATE** statements at two time points	≥ 0			
	Difference between the numbers of **INSERT** statements at two time points	≥ 0			
	Difference between the numbers of **DELETE** statements at two time points	≥ 0			
	Difference between the numbers of **MERGE INTO** statements at two time points	≥ 0			
	Difference between the numbers of **DDL** statements at two time points	≥ 0			
	Difference between the numbers of **DML** statements at two time points	≥ 0			
	Difference between the numbers of **DCL** statements at two time points	≥ 0			
	Difference between the total **SELECT** response time at two time points	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Difference between the average **SELECT** response time at two time points	≥ 0			
	Difference between the maximum **SELECT** response time at two time points	≥ 0			
	Difference between the minimum **SELECT** response time at two time points	≥ 0			
	Difference between the total **UPDATE** response time at two time points	≥ 0			
	Difference between the average **UPDATE** response time at two time points	≥ 0			
	Difference between the maximum **UPDATE** response time at two time points	≥ 0			
	Difference between the minimum **UPDATE** response time at two time points	≥ 0			
	Difference between the total **DELETE** response time at two time points	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Difference between the average **DELETE** response time at two time points	≥ 0			
	Difference between the maximum **DELETE** response time difference at two time points	≥ 0			
	Difference between the minimum **DELETE** response time difference at two time points	≥ 0			
	Difference between the total **INSERT** response time at two time points	≥ 0			
	Difference between the average **INSERT** response time at two time points	≥ 0			
	Difference between the maximum **INSERT** response time at two time points	≥ 0			
	Difference between the minimum **INSERT** response time at two time points	≥ 0			
System Status Statistics	TCP protocol stack retransmission rate (%)	≥ 0.0	30s	Collect the TCP protocol, stack protocol, and stack retransmission rate of the ECS instance server.	>0

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Top SQL Statistics	Database name	String	60s	Collect SQL statements running on each CN in the current cluster to identify deadlock SQL statements, slow SQL statements, or SQL statements with high resource usage. SQL statements can be scanned and killed on the management plane.	N/A
	Instance name	String			
	Thread ID (session ID and session ID)	String			
	Internal query_ID used for statement execution	String			
	Job type, which can be set using the guc parameter **query_band**. The default value is a null string.	String			
	The value is obtained from the **query_band** field. The position is 0.	String			
	The value is obtained from the **query_band** field. The position is 1.	String			
	Username used for connecting to the backend	String			
	Name of the application that is connected to the backend	String			
	IP address of the client connected to the backend. A null value suggests a Unix socket connection or an internal server process, such as autovacuum.	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Host name of the connected client, as reported by a reverse DNS lookup of **client_addr**. This column will only be non-null for IP connections, and only when **log_hostname** is enabled.	String			
	TCP port number used by the client to communicate with the backend. If the Unix socket is used, the value is -1.	String			
	Whether the backend is currently waiting on a lock. If yes, the value is **true**.	Yes/No			
	Time when the statement starts to be executed	≥ 0			
	Block time before the statement is executed. The unit is ms.	≥ 0			
	Duration that a statement has been executed. The unit is ms.	≥ 0			
	Estimated execution time of a statement. The unit is ms.	≥ 0			
	Estimated remaining time of statement execution. The unit is ms.	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Resource status in workload management	String			
	Resource pool used by the user	String			
	Priority of a job in the resource pool. The options are as follows: - 1: **Low** - 2: **Medium** - 4: **High** - 8: **Rush**	≥ 0			
	Cgroup used by the statement	String			
	Minimum memory peak of a statement across all DNs. The unit is MB.	≥ 0			
	Maximum memory peak of a statement across all DNs. The unit is MB.	≥ 0			
	Average memory usage during statement execution. The unit is MB.	≥ 0			
	Memory usage skew of a statement among DNs.	≥ 0			
	Estimated memory used by the statement. The unit is MB.	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Statement spill information on all DNs	String			
	Minimum spilled data among all DNs when a spill occurs. The default value is 0 (unit: MB).	≥ 0			
	Maximum spilled data among all DNs when a spill occurs. The default value is 0 (unit: MB).	≥ 0			
	Average spilled data among all DNs when a spill occurs. The default value is 0 (unit: MB).	≥ 0			
	DN spill skew when a spill occurs	≥ 0			
	Minimum execution time of a statement across all DNs. The unit is ms.	≥ 0			
	Maximum execution time of a statement across all DNs. The unit is ms.	≥ 0			
	Average execution time of a statement across all DNs. The unit is ms.	≥ 0			
	Execution time skew of a statement among DNs.	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Minimum CPU time of a statement across all DNs. The unit is ms.	≥ 0			
	Maximum CPU time of a statement across all DNs. The unit is ms.	≥ 0			
	Total CPU time of a statement across all DNs. The unit is ms.	≥ 0			
	CPU time skew of a statement among DNs.	≥ 0			
	Warning. The following warnings and warnings related to SQL self-diagnosis tuning are displayed:	String			
	Average IOPS peak of a statement across all DNs. It is counted by ones in a column-store table and by ten thousands in a row-store table.	≥ 0			
	I/O skew of a statement among DNs.	≥ 0			
	Maximum IOPS of a statement across all DNs. It is counted by ones in a column-store table and by ten thousands in a row-store table.	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Minimum IOPS of a statement across all DNs. It is counted by ones in a column-store table and by ten thousands in a row-store table.	≥ 0			
	Query statement	String			
	Query plan	String			
	Real-time running status of the current query statement. The value can be **active**, **idle**, **idleintransaction**, **idleintransaction** (**aborted**), **fastpathfunctioncall**, or **disabled**.	String			
	Running status of the query statement in the resource pool. The value can be **pending**, **running**, **finished**, **aborted**, **active** or **unknown**.	String			
	Statement attribute (**ordinary**, **simple**, **complicated**, or **internal**).	String			
	Fast and slow lanes (**fast** or **slow**)	String			
	Whether a query is a system query	Yes/No			
	Whether the query is a system query (adapting to monitor)	Yes/No			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Time when this process was started, that is, when the client connected to the server	≥ 0			
	Execution time so far	≥ 0			
	Time when the current transaction was started (**NULL** if no transactions are active) If the current query is the first of its transaction, the value of this column is the same as that of the **query_start** column.	≥ 0			
	Time when the **status** was changed in the previous time	≥ 0			
	Time when the statement starts to be executed	≥ 0			
	Actual execution duration of the statement, in seconds.	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
Historical Queries	Database name	String	180s	Collect historical data from the top SQL view to analyze SQL statements from the past, identify deadlock or slow SQL statements, and optimize cluster performance by rectifying any issues found.	N/A
	Instance name	String			
	Username	String			
	Name of the application that is connected to the backend	String			
	IP address of the client connected to the backend. A null value suggests a Unix socket connection or an internal server process, such as autovacuum.	String			
	Host name of the connected client, as reported by a reverse DNS lookup of **client_addr**. This column will only be non-null for IP connections, and only when **log_hostname** is enabled.	String			
	TCP port number used by the client to communicate with the backend. If the Unix socket is used, the value is -1.	String			
	Job type, which can be set using the guc parameter **query_band**. The default value is a null string.	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	The value is obtained from the **query_band** field. The position is 0.	String			
	The value is obtained from the **query_band** field. The position is 1.	String			
	Duration that a statement is blocked before being executed, including the statement parsing and optimization duration. The unit is ms.	≥ 0			
	Execution start time of a statement (unit: ms)	≥ 0			
	Execution end time of a statement (unit: ms)	≥ 0			
	Duration that a statement has been executed (unit: ms)	≥ 0			
	Estimated statement execution time (unit: ms)	≥ 0			
	Statement execution end status: ● **Finished** (normal) ● **Aborted** (Abnormal)	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Exception information displayed if the final statement execution status is **aborted**.	String			
	Resource pool used by the user	String			
	Priority of a job in the resource pool. The options are as follows: ● 8: **Rush** ● 4: **High** ● 2: **Medium** ● 1: **Low**	≥ 0			
	Cgroup used by the statement	String			
	Minimum memory peak of a statement across all DNs. The unit is MB.	≥ 0			
	Maximum memory peak of a statement across all DNs. The unit is MB.	≥ 0			
	Average memory usage during statement execution. The unit is MB.	≥ 0			
	Memory usage skew of a statement among DNs	≥ 0			
	Statement spill information on all DNs	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Minimum spilled data among all DNs when a spill occurs. The unit is MB. The default value is **0**.	≥ 0			
	Maximum spilled data among all DNs when a spill occurs. The unit is MB. The default value is **0**.	≥ 0			
	Average spilled data among all DNs when a spill occurs. The unit is MB. The default value is **0**.	≥ 0			
	DN spill skew when a spill occurs	≥ 0			
	Minimum execution time of a statement across all DNs. The unit is ms.	≥ 0			
	Maximum execution time of a statement across all DNs. The unit is ms.	≥ 0			
	Average execution time of a statement across all DNs. The unit is ms.	≥ 0			
	Execution time skew of a statement among DNs.	≥ 0			
	Minimum CPU time of a statement across all DNs. The unit is ms.	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Maximum CPU time of a statement across all DNs. The unit is ms.	≥ 0			
	Total CPU time of a statement across all DNs. The unit is ms.	≥ 0			
	CPU time skew of a statement among DNs.	≥ 0			
	Minimum IOPS of a statement across all DNs. It is counted by ones in a column-store table and by ten thousands in a row-store table.	≥ 0			
	Maximum IOPS of a statement across all DNs. It is counted by ones in a column-store table and by ten thousands in a row-store table.	≥ 0			
	Average IOPS peak of a statement across all DNs. It is counted by ones in a column-store table and by ten thousands in a row-store table.	≥ 0			
	I/O skew of a statement among DNs	≥ 0			
	Warning. The following warnings and warnings related to SQL self-diagnosis tuning are displayed:	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Query ID	String			
	Statement executed	String			
	Execution plan of a statement	String			
	Logical cluster of the user running the statement	String			
Schema Usage Statistics	Database name	String	3600s	Monitor schema usage in a cluster by collecting usage data for each schema.	85 %
	Schema name	String			
	Used capacity (unit: byte)	≥ 0			
	Total capacity (unit: byte)	≥ 0			
Session Statistics	Database name	String	180s	Gather session information for each CN in a cluster, including statistics on idle sessions and lock holdings. Use the management console to clear any idle sessions.	N/A
	Instance name	String			
	Thread ID (It can be used as a session ID or a connection ID.)	String			
	Database username	String			
	User application name	String			
	Client address	String			
	Host name of the client	String			
	TCP port number used by the client to communicate with the background. If the Unix socket is used, the value is -1.	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Time when this process was started, that is, when the client connected to the server	≥ 0			
	Execution time so far.	≥ 0			
	Time when the current transaction was started (**NULL** if no transactions are active). If the current query is the first of its transaction, the value of this column is the same as that of the **query_start** column.	≥ 0			
	Time when the **status** was changed in the previous time	≥ 0			
	Whether the backend is currently waiting on a lock. If yes, the value is **true**.	Yes/No			
	Current overall state of this backend.	String			
	Resource pool used by the user	String			
	Actual execution duration of the statement, in seconds.	≥ 0			
	ID of a query	String			
	Time when the statement starts to be executed	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	cgroups currently used by the statement.	String			
	Lock type	String			
	Lock mode	String			
	Indicates whether to hold a lock when lock waiting exists. The value is true.	Yes/No			
	Resource that is waiting for the lock	String			
	Statement type	String			
	Query SQL.	String			
	Indicates whether the query is performed by the system.	Yes/No			
	Query plan	String			
SQL Probe Statistics	Query ID of a probe task	String	30s	Monitor cluster performance by collecting statistics on the duration of SQL probe execution, to detect sudden deterioration.	N/A
	Cluster ID	String			
	Cluster project ID	String			
	Task type of the probe SQL	String			
	Time when a probe SQL task is created	≥ 0			
	SQL execution time	≥ 0			
	probe_ID of the associated probe SQL	String			
	Probe	String			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Task status of the SQL probe. The options are as follows: - **Running** - **Success** - **Fail**	String			
Table Dirty Page Rate Statistics	Database name	String	7200s	Gather information on dirty pages for cluster tables, as a high dirty page rate can cause a decline in table query and insertion performance.	50 %
	Schema name	String			
	Table name (full name)	String			
	Table owner	String			
	Table size (unit: byte)	≥ 0			
	Dirty page rate	≥ 0.0			
Table Skew Monitoring Statistics	Database name	String	7200s	Identify tables in the cluster with a skew rate exceeding 5%, as this can negatively affect query performance.	10 %
	Schema name	String			
	Table name (full name)	String			
	Table owner	String			
	Table size (unit: byte)	≥ 0			
	Skew rate	≥ 0.0			
Resource Pool Statistics	Load resource pool	String	120s	Monitor system resource usage and queuing status of the cluster resource pool by collecting relevant information.	N/A
	CPU quota of the resource pool	≥ 0			
	Memory quota of the resource pool	≥ 0			
	Disk quota of the resource pool	≥ 0			

Monitored Object	Metric	Value Range	Monitoring Interval (Raw Data)	Metric Usage	Recommended Threshold
	Maximum number of concurrent simple jobs allowed by the resource pool	≥ 0			
	Maximum number of concurrent queries allowed by the resource pool	≥ 0			
	CPU usage of the resource pool	≥ 0.0			
	Memory usage of the resource pool	≥ 0.0			
	Disk usage of the resource pool	≥ 0.0			
	Number of concurrent simple jobs in the resource pool	≥ 0			
	Current number of concurrent requests in the resource pool	≥ 0			
Resource Pool User Statistics	Load resource pool	String	30s	Gather data on users in the cluster resource pool and track their resource usage.	N/A
	CPU quota of a user.	≥ 0			
	Memory quota of a user	≥ 0			
	Specifies the disk quota of a user	≥ 0			
	CPU usage of a user	≥ 0.0			
	Memory usage of a user	≥ 0.0			
	Disk usage of a user	≥ 0.0			
	User ID	String			
	Username	String			

9.1.3 Cluster Overview

Cluster Overview

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, click **Cluster Overview**.

On the page that is displayed, you can view the cluster status, real-time resource consumption, cluster resource consumption, and key database metrics.

----End

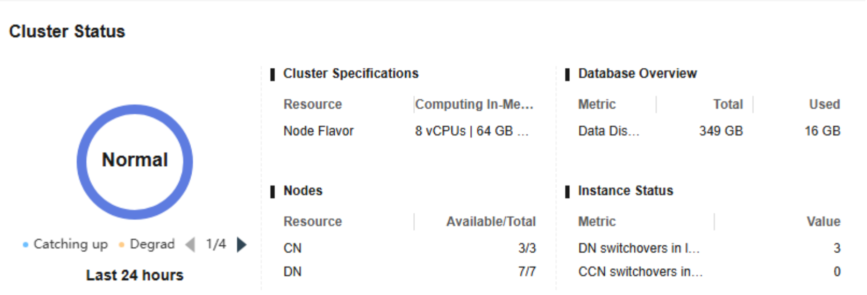
NOTE

Metrics can be collected and displayed on the cluster overview page only if their collection items are enabled. If a collection item is disabled, its metric will not be displayed, and a prompt will be displayed indicating this problem. In this case, you are advised to enable the collection item.

Cluster Status

In the **Cluster Status** area, you can view the statistics about the current cluster status and instance status, including cluster statistics in the last 24 hours, cluster specifications, available/total CNs and DNs, used/total disk capacity, the number of CCN switchovers in the last 24 hours, and the number of primary/standby DN switchovers in the last 24 hours.

Figure 9-1 Cluster Status



Cluster Status

Normal

• Catching up • Degrad ◀ 1/4 ▶

Last 24 hours

Cluster Specifications

Resource	Computing In-Me...
Node Flavor	8 vCPUs | 64 GB ...

Nodes

Resource	Available/Total
CN	3/3
DN	7/7

Database Overview

Metric	Total	Used
Data Dis...	349 GB	16 GB

Instance Status

Metric	Value
DN switchovers in l...	3
CCN switchovers in...	0

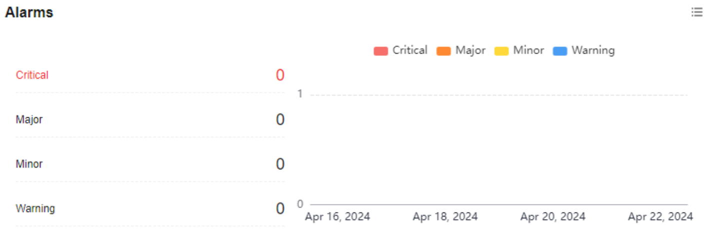
NOTE

The OBS usage details are displayed for storage-compute decoupled clusters.

Alarms

In the **Alarms** area, you can view all the uncleared alarms of the current cluster and the alarms generated in the last seven days. You can click **More** in the upper right corner to view details about the existing cluster alarms. For details, see **Viewing GaussDB(DWS) Cluster Alarms**.

Figure 9-2 Alarms



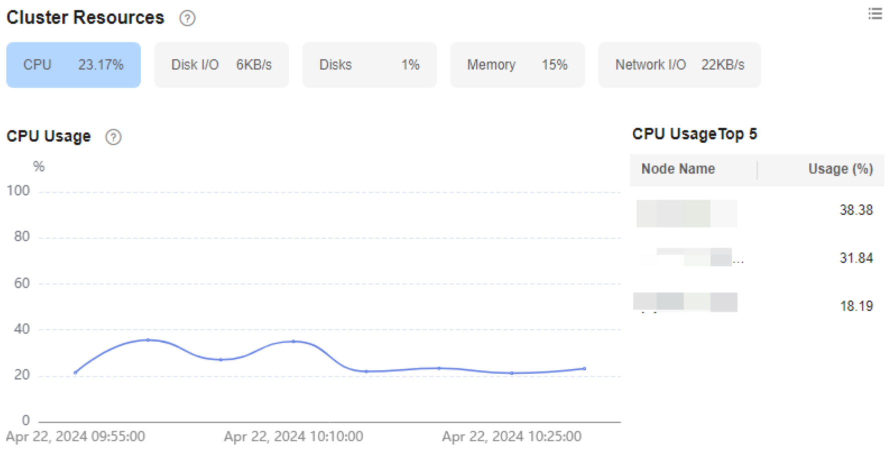
Alarms

Critical	0
Major	0
Minor	0
Warning	0

Cluster Resources

In the **Cluster Resources** area, you can view the resource usage of the current cluster, including the average CPU usage, disk I/O, disk usage, memory usage, and network I/O. You can click the metric of a resource to view its trend in the last 24 hours and the top five services that are occupying this resource. You can click **More** in the upper right corner of the area to go to the Node Monitoring page. Nodes are sorted by the metric value. For details, see **Node Monitoring**.

Figure 9-3 Cluster Resources



Cluster Resources

CPU 23.17% | Disk I/O 6KB/s | Disks 1% | Memory 15% | Network I/O 22KB/s

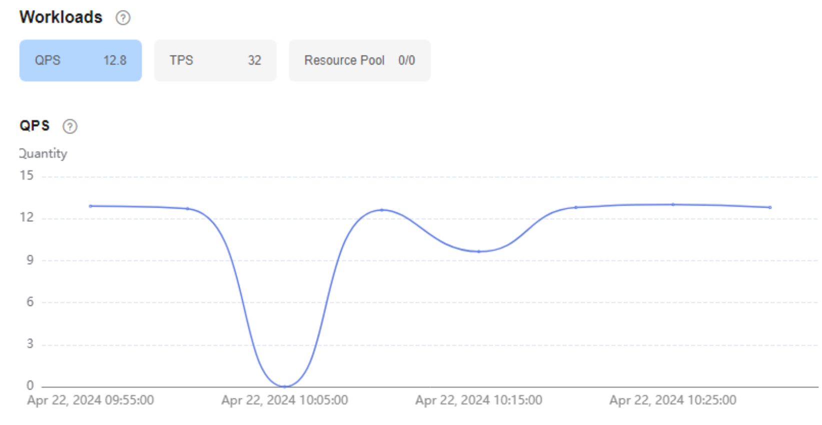
CPU Usage Top 5

Node Name	Usage (%)
[redacted]	38.38
[redacted]	31.84
[redacted]	18.19

Workloads

In the **Workloads** area, you can view the workload metrics of the current database, including TPS, QPS, stacked SQL queries, and resource pools. You can also click a workload metric to view its trend in the last 24 hours. The **SQL Stack Queries** metric depends on the real-time query monitoring function. If this function is disabled, no data will be displayed for the metrics.

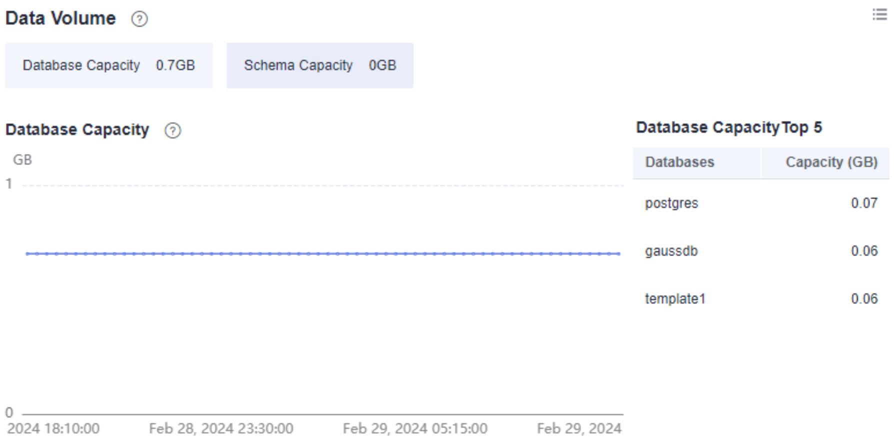
Figure 9-4 Workloads



Data Volume

In the **Data Volume** area, you can view the used capacity of the current database and schema. You can click a capacity metric to view the database or schema capacity trend in the last 24 hours and the top five databases or schemas ranked by capacity usage in the current cluster. You can click **More** in the upper right corner of the area to go to the **Database Monitoring** page. Databases are sorted by used capacity. For details, see **Database Monitoring**.

Figure 9-5 Data Volume



Databases	Capacity (GB)
postgres	0.07
gaussdb	0.06
template1	0.06

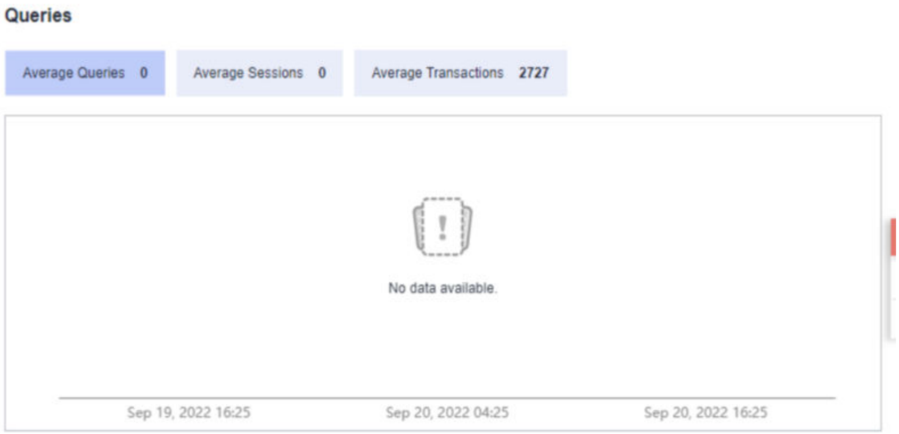
NOTE

The database capacity data is collected once a day. Therefore, the data volume fluctuates greatly. To view real-time capacity monitoring information, choose **Node Monitoring** > **Disks**.

Queries

In the **Queries** area, you can check the average number of queries, sessions, and transactions. You can click a metric to view its trend in the last 24 hours. The **Average Queries** and **Average Sessions** metrics depend on the real-time query monitoring function. If this function is disabled, no data will be displayed for the metrics.

Figure 9-6 Queries



9.1.4 Monitoring

9.1.4.1 Node Monitoring

Node Monitoring

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, choose **Monitoring** > **Node Monitoring**.

On the page that is displayed, view the real-time consumption of nodes, memory, disks, disk I/O, and network I/O.

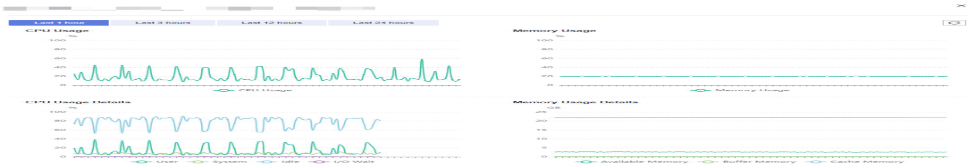
----End

Overview

On the **Overview** tab page, you can view the key resources of a specified node based on the node name, including:

- Node Name
- CPU Usage (%)
- Memory Usage (%)
- Average Disk Usage (%)
- IP Address
- Disk I/O (KB/s)
- TCP Protocol Stack Retransmission Rate (%)
- Network I/O (KB/s)
- Status

- Monitoring: You can click in the **Monitoring** column to view the overall performance metric topology of the node in the last 1 hour, last 3 hours, last 12 hours, last 24 hours, last 7 days, or last 15 days.

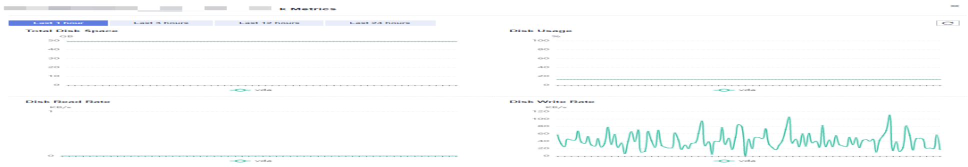


Disks

On the **Disks** tab page, view the real-time disk resource consumption of a node by node name and disk name, including:

- Node Name
- Disk Name
- Disk Type
  - System disk
  - Data disk
  - Log disk
- Disk Capacity (GB)
- Disk Usage (%)
- Disk Read Rate (KB/s)
- Disk Write Rate (KB/s)
- I/O Wait Time (await, ms)
- I/O Service Time (svctm, ms)
- IOPS

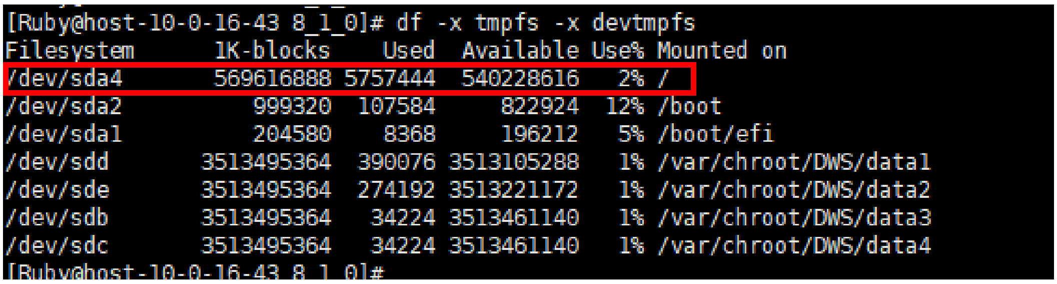
- Monitoring: You can click in the **Monitoring** column to view the disk performance metric topology of the node in the last 1 hour, last 3 hours, last 12 hours, or last 24 hours.



NOTE

The sum of the used disk space and available disk space is not equal to the total disk space. This is because a small amount of space is reserved in each default partition for system administrators to use. Even if common users have run out of space, system administrators can log in to the system and use their space required for solving problems.

Run the Linux **df** command to collect the disk capacity information, as shown in the following figure.



```
[Ruby@host-10-0-16-43 8_1_0]# df -x tmpfs -x devtmpfs
Filesystem      1K-blocks    Used  Available Use% Mounted on
/dev/sda4       569616888 5757444  540228616   2% /
/dev/sda2          999320  107584     822924  12% /boot
/dev/sda1          204580    8368     196212   5% /boot/efi
/dev/sdd       3513495364  390076 3513105288   1% /var/chroot/DWS/data1
/dev/sde       3513495364  274192 3513221172   1% /var/chroot/DWS/data2
/dev/sdb       3513495364   34224 3513461140   1% /var/chroot/DWS/data3
/dev/sdc       3513495364   34224 3513461140   1% /var/chroot/DWS/data4
[Ruby@host-10-0-16-43 8_1_0]#
```

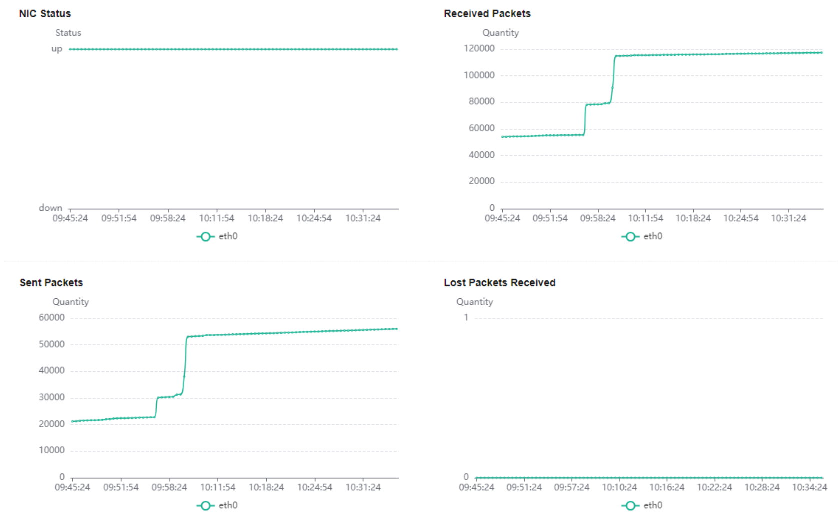
/dev/sda4: Used(5757444) + Available(540228616) != Total(569616888)

- **Filesystem**: path name of the device file corresponding to the file system. Generally, it is a hard disk partition.
- **IK-blocks**: number of data blocks (1024 bytes) in a partition.
- **Used**: number of data blocks used by the disk.
- **Available**: number of available data blocks on the disk.
- **Use%**: percentage of the space used by common users. Even if the space is used up, the partition still reserves the space for system administrators.
- **Mounted on**: mount point of the file system.

Network

On the **Network** tab page, view the real-time network resource consumption of a node by node name and NIC name, including:

- Node Name
- NIC Name
- NIC Status
- NIC Speed (Mbps)
- Received Packets
- Sent Packets
- Lost Packets Received
- Receive Rate (KB/s)
- Transmit Rate (KB/s)
- Monitoring: You can click 🖥 in the **Monitoring** column to view the network performance metric topology of the node in the last 1 hour, last 3 hours, last 12 hours, or last 24 hours.



9.1.4.2 Performance Monitoring

Performance Monitoring

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Monitoring** > **Performance Monitoring**. The **Performance Monitoring** page displays the resource consumption trends of clusters, databases, nodes, and instances.

You can select a time range and check the performance trend in this range.

- By default, the monitoring information of the last hour is displayed.
- You can view the monitoring information of the last seven days.

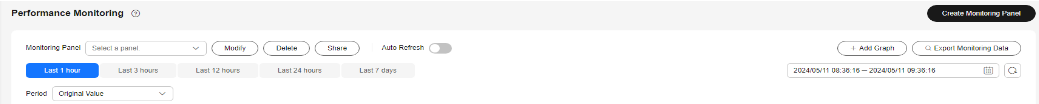
----End

Monitoring Panel

You can configure monitoring views by customizing monitoring panels. Monitoring panels are bound to users. After logging in to the system, you can view the user-defined monitoring panels.

- Creating a monitoring panel: You can click **Create Panel** to customize a monitoring panel.
- Modifying a monitoring panel: You can click **Modify** to change the name of a monitoring panel.
- Deleting a monitoring panel: You can click **Delete** to delete a monitoring panel. The default monitoring panel cannot be deleted.
- Sharing a monitoring panel: You can click **Share** to share a monitoring panel. The recipients can view the panel but cannot modify it.

Figure 9-7 Monitoring Panel



Performance Monitoring

Monitoring Panel: Select a panel.

Auto Refresh

Last 1 hour

2024/05/11 08:36:16 – 2024/05/11 09:36:16

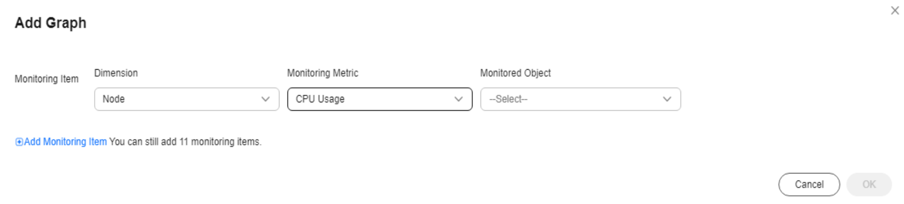
Period: Original Value

Adding a Monitoring View

Currently, DMS provides monitoring views for clusters, databases, nodes, and instances. You can click **Add View** to add a monitoring view as required. The monitoring metrics are as follows:

- Cluster-level monitoring metrics: CPU usage, memory usage, disk usage, disk I/O, network I/O, status, number of abnormal CNs, read-only or not, number of sessions, number of active sessions, number of deadlocks, number of abnormal DNs, DN CPU usage, average number of transactions per second, average number of queries per second, capacity, mode capacity, number of stacked SQL statements, number of queries, resource pools, and number of transactions.
- Database monitoring metrics: number of active sessions, number of sessions, number of inserted rows, number of updated rows, number of deleted rows, and capacity.
- Instance-level monitoring metrics: instance memory usage and dynamic memory usage.
- Node-level monitoring metrics: CPU usage, memory usage, average disk usage, TCP protocol stack retransmission rate, disk I/O, network I/O, total disk capacity, disk usage, disk read rate, disk write rate, disk I/O waiting time, disk I/O service time, disk I/O usage, NIC status, number of received packets, number of sent packets, number of discarded received packets, receiving rate, sending rate, CPU usage, and memory usage.

Figure 9-8 Adding a Monitoring View



Add Graph

Monitoring Item

Dimension: Node

Monitoring Metric: CPU Usage

Monitored Object: --Select--

Add Monitoring Item You can still add 11 monitoring items.

NOTE

- A maximum of 20 views can be added to each panel. Adding too many views will increase the number of page requests and the rendering time.
- A maximum of 20 monitored objects can be selected in the node dimension. This feature is supported only in 8.1.3.310 and later cluster versions.

Exporting Monitoring Data

Performance Monitoring supports data export. You can click **Export Data** to further process data. By default, data in all monitoring views on the current page is exported. The export time range is subject to the selected time range.

NOTE

Performance Monitoring allows data aggregation of different periods. You can aggregate raw data based on the corresponding sampling period to display indicator trends of a longer period.

9.1.4.3 Database Monitoring

Database Monitoring

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, choose **Monitoring** > **Database Monitoring**.

The **Database Monitoring** page displays the real-time and historical resource consumption a database.

----End

Database Resource Consumption

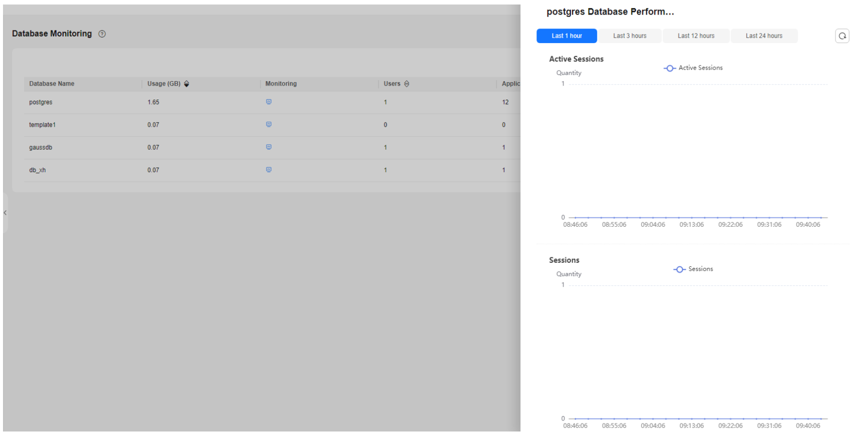
You can select a database to view its resource usage. For details about the metrics, see **Monitoring Metrics**. including:

On the Database Monitoring page, you can check the database name, usage (GB), monitoring status, number of users, number of applications, number of sessions, number of queries, number of inserted rows, number of updated rows, number of deleted rows, number of deadlocks, number of temporary files, and temporary file capacity.

Database Trend Monitoring

In the **Monitoring** column of a database, click 🖥 to view the performance indicators of the database, including:

- Capacity
- Sessions
- Queries



Database Monitoring

Database Name	Usage (GB)	Monitoring	Users	Applic
postgres	1.65	▤	1	12
template1	0.07	▤	0	0
gaussdb	0.07	▤	1	1
db_xh	0.07	▤	1	1

9.1.4.4 Real-Time Queries

Going to the Real-time Query Page

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane, choose **Monitoring** > **Queries**.

You can check the real-time information about all queries and sessions running in the cluster.

----End

NOTICE

- Real-time query is supported only in clusters of version 8.1.2 and later.
- To enable real-time query monitoring, choose **Settings** > **Monitoring**, click **Monitoring Collection**, and enable **Real-Time Query Monitoring**. For details, see **Monitoring Collection**. Enabling real-time query may cause a large amount of data. Exercise caution when performing this operation.

Prerequisites

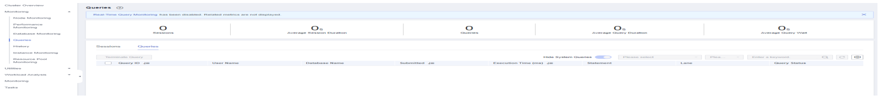
You need to set GUC parameters before viewing data on the monitoring page. If GUC parameters are not set, real-time or historical query may be unavailable. However, if this parameter is set, the cluster performance may deteriorate. Therefore, you need to balance the settings of related parameters. The following table lists the recommended GUC parameter settings. For how to modify parameters, see **Modifying GUC Parameters of the GaussDB(DWS) Cluster**. For details about the parameters, see **Setting GUC Parameters**.

Table 9-2 Recommended GUC parameter settings

GUC Parameter	CN Configuration	DN Configuration
max_active_statements	10	10
enable_resource_track	on	on
resource_track_level	query	query
resource_track_cost	0	0
resource_track_duration	10	10
enable_resource_record	on	on
session_statistics_memory	1000 MB	1000 MB

Querying Information

You can view the queries statistics, the number of sessions, average session duration (time of all session connections divided by the number of sessions), number of queries, average query duration, and average query waiting time.



Real-Time Query Monitoring has been disabled. Related metrics are not displayed.

0	0s	0	0s	0s
Sessions	Average Session Duration	Queries	Average Query Duration	Average Query Wait

Query ID	User Name	Database Name	Submitted	Execution Time (ms)	Statement	Lane	Query Status

Checking Live Sessions

On the **Sessions** page, you can browse the real-time information about all running queries. You can click the setting button in the upper right corner of the list to select the metrics to be displayed in the list. The metrics are as follows:

Session ID, username, session duration, application name, QueryBand, client IP address, access CN, session status, start time, lock mode, lock holding status, locked object, query SQL, lock wait, current query duration, and current query start time.

Here are the different session statuses:

- **idle**: The backend is waiting for new client commands.
- **active**: The backend is executing queries.
- **idle in transaction**: The backend is in a transaction, but there is no statement being executed in the transaction.
- **idle in transaction (aborted)**: The backend is in a transaction, but there are statements failed in the transaction.
- **fastpath function call**: The backend is executing a **fast-path** function.

NOTE

- You can click a session ID to view the queries in the current session. For details, see **Viewing Real-time Query Monitoring Details**.
- To terminate a session, select the session, click **Terminate a Session**, and confirm your operation.
- If you want to terminate all idle sessions, click **Clear Idle Sessions**.
- The fine-grained permission control function is added. Only users with the operate permission are able to terminate sessions. For users with the read-only permission, the **Terminate a Session** button is grayed out.

Checking Real-time Queries

In the real-time query area, you can view details on all queries that are currently active in the cluster during a specific time period. To customize the metrics displayed in the list, click the settings button in the top right corner. The metrics are as follows:

Query ID, username, application name, database name, resource pool, submission time, blocking time (ms), execution time (ms), minimum CPU time (ms), maximum CPU time (ms), CPU time (ms), CPU time skew (%), DN spilling information, minimum spilled data among all DNs (MB), maximum spilled data among all DNs (MB), average spill to disk (MB), DN spill skew, query statement, access CN, client IP address, fast and slow lanes, query status, session ID, queuing status, job type, task name, task instance, TCP port, waiting or not, estimated total execution time (ms), estimated remaining time (ms), cgroup, minimum memory peak of DN (MB), maximum memory peak of DN (MB), average memory usage (MB), memory usage skew ratio across DNs, estimated memory usage (MB), minimum execution time of a statement across all DNs (ms), maximum execution time of a statement across all DNs (ms), average execution time of a statement on all DNs (ms), and execution time skew of a statement among DNs, alarms, average IOPS peak of a statement across all DNs (times/s for column-store tables and 10,000 times/s for row-store tables), I/O skew of a statement among DNs, statement status, and statement attributes.

Here are the different query statuses:

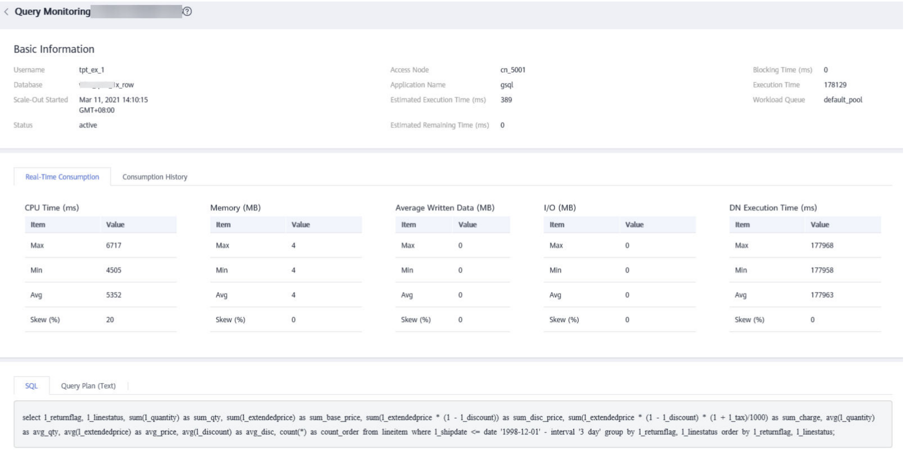
- **idle**: The backend is waiting for new client commands.
- **active**: The backend is executing queries.
- **idle in transaction**: The backend is in a transaction, but there is no statement being executed in the transaction.
- **idle in transaction (aborted)**: The backend is in a transaction, but there are statements failed in the transaction.
- **fastpath function call**: The backend is executing a **fast-path** function.

NOTE

- You can click a query ID to view the monitoring details. However, details cannot be displayed for queries whose ID is **0**. Query **0** indicates that an exception occurs during the query.
- To terminate a query, select the query, click **Terminate Query**, and confirm your operation.
- The fine-grained permission control function is added. Only users with the operate permission are able to terminate queries. For users with the read-only permission, the **Terminate Query** button is grayed out.
- The fast and slow lanes are selected based on the cost in the execution plan. If the optimizer estimates that the memory usage of a statement is greater than 32 MB, the statement enters the slow lane. Otherwise, the statement enters the fast lane.

Viewing Real-time Query Monitoring Details

You can click a query ID to view the query details, including the basic information of query statements, real-time and historical resource consumption, SQL description, and query plan.



Query Monitoring [redacted]

Basic Information

Username	tpt_ex_1	Access Node	cn_5001	Blocking Time (ms)	0
Database	[redacted]x_row	Application Name	gsql	Execution Time	178129
Scale-Out Started	Mar 11, 2021 14:10:15 GMT+08:00	Estimated Execution Time (ms)	389	Workload Queue	default_pool
Status	active	Estimated Remaining Time (ms)	0		

CPU Time (ms)

Item	Value
Max	6717
Min	4505
Avg	5352
Skew (%)	20

Memory (MB)

Item	Value
Max	4
Min	4
Avg	4
Skew (%)	0

Average Written Data (MB)

Item	Value
Max	0
Min	0
Avg	0
Skew (%)	0

I/O (MB)

Item	Value
Max	0
Min	0
Avg	0
Skew (%)	0

DN Execution Time (ms)

Item	Value
Max	177968
Min	177958
Avg	177963
Skew (%)	0

```
select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedprice) as sum_base_price, sum(l_extendedprice * (1 - l_discount)) as sum_disc_price, sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)/1000) as sum_charge, avg(l_quantity)
as avg_qty, avg(l_extendedprice) as avg_price, avg(l_discount) as avg_disc, count(*) as count_order from lineitem where l_shipdate <= date '1998-12-01' - interval '3 day' group by l_returnflag, l_linestatus order by l_returnflag, l_linestatus;
```

9.1.4.5 Historical Queries

Going to the Historical Query Page

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Monitoring** > **History**.

All the historical queries in the current cluster will be displayed.

----End

NOTE

- Historical queries can be viewed only in clusters of version 8.1.2 and later.
- To enable historical query monitoring, choose **Settings** > **Monitoring**, click **Monitoring Collection**, and enable **Historical Query Monitoring**. For details, see **Monitoring Collection**. Enabling history query may cause a large amount of data. Exercise caution when performing this operation.

Checking Historical Queries

In the **History** area, you can browse all historical query information based on the specified time period. You can click the setting button in the upper right corner of the list to select the metrics to be displayed in the list. The metrics are as follows:

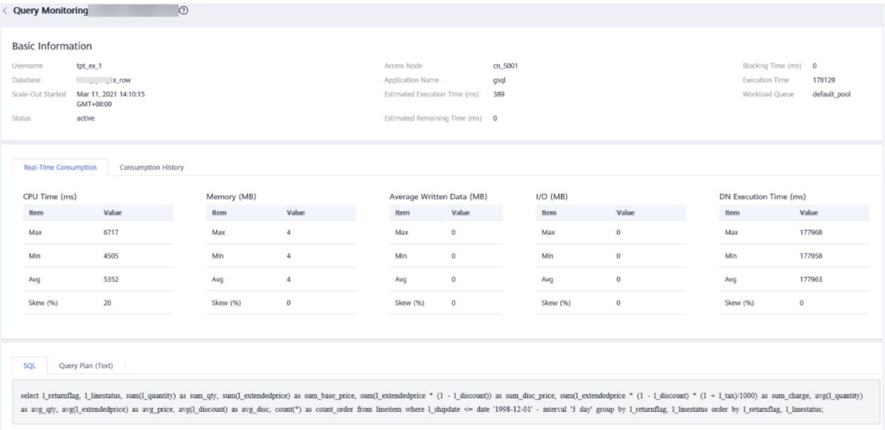
Query ID, username, application name, database name, resource pools, submission time, blocking time (ms), execution time (ms), CPU time (ms), CPU time skew (%), average spill to disk (MB), query statement, access CN, client IP address, query status, completion time, estimated total execution time (ms), and cancellation reason.

NOTE

If you do not want to see historical system queries, you can toggle on **Hide System Queries**.

Viewing Historical Query Monitoring Details

You can click a historical query ID to view the query details, such as basic information about query statements, real-time resource consumption during execution, complete SQL statements, and query plans.



Query Monitoring [redacted]

Basic Information

Username	tpt_ex_1	Access Node	cn_5001	Blocking Time (ms)	0
Database	[redacted]x_row	Application Name	gsql	Execution Time	178129
Scale-Out Started	Mar 11, 2021 14:10:15 GMT+08:00	Estimated Execution Time (ms)	389	Workload Queue	default_pool
Status	active	Estimated Remaining Time (ms)	0		

Real-Time Consumption | Consumption History

CPU Time (ms)

Item	Value
Max	6717
Min	4505
Avg	5352
Skew (%)	20

Memory (MB)

Item	Value
Max	4
Min	4
Avg	4
Skew (%)	0

Average Written Data (MB)

Item	Value
Max	0
Min	0
Avg	0
Skew (%)	0

I/O (MB)

Item	Value
Max	0
Min	0
Avg	0
Skew (%)	0

DN Execution Time (ms)

Item	Value
Max	177968
Min	177958
Avg	177963
Skew (%)	0

SQL | Query Plan (Text)

```
select l_returnflag, l_linestatus, sum(l_quantity) as sum_qty, sum(l_extendedprice) as sum_base_price, sum(l_extendedprice * (1 - l_discount)) as sum_disc_price, sum(l_extendedprice * (1 - l_discount) * (1 + l_tax)/1000) as sum_charge, avg(l_quantity) as avg_qty, avg(l_extendedprice) as avg_price, avg(l_discount) as avg_disc, count(*) as count_order from lineitem where l_shipdate <= date '1998-12-01' - interval '3 day' group by l_returnflag, l_linestatus order by l_returnflag, l_linestatus;
```

9.1.4.6 Instance Monitoring

Instance Monitoring

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Monitoring** > **Instance Monitoring**.

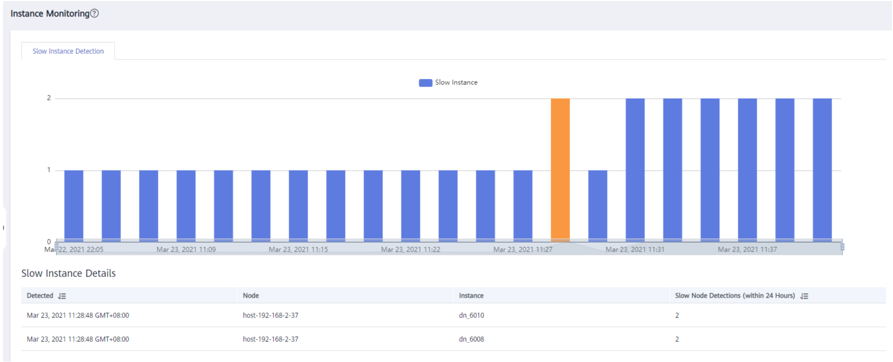
On the **Instance Monitoring** page, you can view the real-time and historical information about detected slow instances.

----End

Slow Instance Detection

DMS can automatically configure and start the slow instance detection script on cluster CNs, periodically collect the cache table of the script, and report the detected slow instance data. You can view the number of slow instances detected within 24 hours and the distribution status in the time dimension on the GUI to quickly locate the slow nodes in the cluster and analyze the root causes.

The **Instance Monitoring** page consists of two parts. The upper part displays the time distribution chart of detected slow instances, that is, the number of slow instances detected in different detection periods. The lower part displays slow instance details. When you select any bar in the time distribution chart, details about the detection time, node name, instance name, and number of detections (within 24 hours) of slow instances are displayed.



Slow Instance Details

Detected	Node	Instance	Slow Node Detections (within 24 Hours)
Mar 23, 2021 11:28:48 GMT+08:00	host-192-168-2-37	dn_6010	2
Mar 23, 2021 11:28:48 GMT+08:00	host-192-168-2-37	dn_6008	2

NOTE

If the period of an instance exceeds 240 seconds, the instance is reported as a slow instance.

9.1.4.7 Resource Pool Monitoring

Accessing the Resource Monitoring Page

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Monitoring** > **Resource Pool Monitoring**.

You can check the real-time statistics and resource consumption history about resource pools.

----End

Resource Pool

You can check user-defined resource pools, real-time and historical resource consumption, and the resource quotas of resource pools. You can click the setting button in the upper right corner of the list to select the metrics to be displayed in the list. The metrics are as follows:

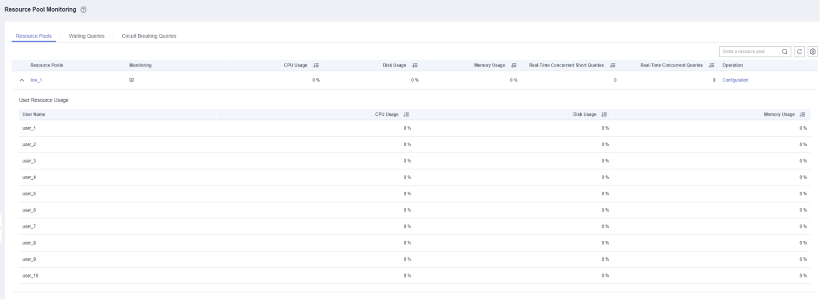
- **Resource Pool**: Resource pool name.
- **Monitoring**: You can click the monitoring icon to display the historical consumption trends of resources such as the CPU, memory, and disk.
- **CPU Usage**: real-time CPU usage of a resource pool.
- CPU share: percentage of CPU time that can be used by users associated with the current resource pool to execute jobs.
- **Storage**: storage space of a resource pool.
- **Disk Usage**: real-time disk usage of a resource pool.
- **Memory Resource**: memory quota of a resource pool.
- **Memory Usage**: percentage of used memory.
- **Real-Time Concurrent Short Queries**: number of concurrent simple queries in a resource pool. Concurrent simple queries are not controlled by the resource pool.
- **Simple Statement Concurrency**: quota of simple concurrent queries in a resource pool.
- **Real-Time Concurrent Queries**: number of concurrent complex queries in a resource pool. Concurrent complex queries are controlled by the resource pool.
- **Complex Statement Concurrency**: quota of complex concurrent queries in a resource pool.
- **Operation**: resource pool configurations.

User Resource Usage

Click the arrow next to a resource pool name to expand resource usage details. The metrics are as follows:

- **User Name**: name of a user in the current resource pool
- **CPU Usage**: real-time CPU usage of a user.
- **CPU Share**: percentage of CPU time that can be used by users associated with the current resource pool to execute jobs.
- **Storage Resource**: storage space used by a user.
- **Disk Usage**: disks used by a user.
- **Memory Resource**: memory used by a user.
- **Memory Usage**: percentage of memory used by a user.

Figure 9-9 User Resource Usage



Resource Pool Monitoring

Resource Pools	Monitoring	CPU Usage	Disk Usage	Memory Usage	Real-Time Concurrent Short Queries	Real-Time Concurrent Queries	Operation
line_1		0 %	0 %	0 %	0	0	Configuration

User Resource Usage

User Name	CPU Usage	Disk Usage	Memory Usage
user_1	0 %	0 %	0 %
user_2	0 %	0 %	0 %
user_3	0 %	0 %	0 %
user_4	0 %	0 %	0 %
user_5	0 %	0 %	0 %
user_6	0 %	0 %	0 %
user_7	0 %	0 %	0 %
user_8	0 %	0 %	0 %
user_9	0 %	0 %	0 %
user_10	0 %	0 %	0 %

Queries Waiting in a Resource Pool

You can view the queries waiting in a resource pool in real time to check workload status.

- **User**: user name of a query statement
- **Application**: application name of a query statement
- **Database**: name of the database to which a query statement is connected
- **Queuing Status**: queuing status of a query statement in a resource pool
- **Wait Time (ms)**: waiting time before a query statement is executed.
- **Resource Pool**: resource pool that the query belongs to
- **Query Statement**: details of a query statement submitted by a user

Checking Circuit Breaking Queries

You can view the status of a triggered circuit breaking query in a resource pool.

- **Query ID**: ID of a circuit breaking query
- **Query Statement**: circuit breaking query statement
- **Blocking Time (ms)**: blocking time of a circuit breaking statement.
- **Execution Time (ms)**: execution time of a circuit breaking statement.
- **CPU Time (ms)**: CPU time consumed by a circuit breaking statement.
- **CPU Skew (%)**: CPU skew of a circuit breaking statement on each DN
- **Exception Handling**: exception handling method of a circuit breaking statement
- **Status**: real-time status of a circuit breaking statement

9.1.5 Utilities

9.1.5.1 SQL Diagnosis

Prerequisites

To enable SQL diagnosis, enable monitoring on real-time and historical queries on the **Queries** and **History** tabs, respectively. For details, see **Monitoring Collection**.

Viewing SQL Diagnosis

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

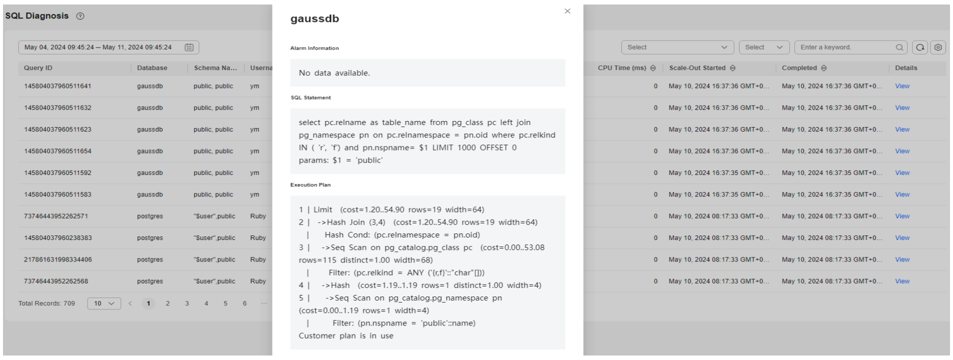
Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Utilities** > **SQL Diagnosis**. The metrics include:

- Query ID
- Database
- Schema Name
- User Name
- Client
- Client IP Address
- Running Time (ms)
- CPU Time (ms)
- Scale-Out Started
- Completed
- Details

Step 5 On the **SQL Diagnosis** page, you can view the SQL diagnosis information. In the **Details** column of a specified query ID, click **View** to view the detailed SQL diagnosis result, including:

- Alarm Information
- SQL Statement
- Execution Plan



SQL Diagnosis

May 04, 2024 09:45:24 — May 11, 2024 09:45:24

Query ID	Database	Schema Na...	Userna	CPU Time (ms)	Scale-Out Started	Completed	Details
145804037960511641	gaussdb	public, public	ym	0	May 10, 2024 16:37:36 GMT+0...	May 10, 2024 16:37:36 GMT+0...	View
145804037960511632	gaussdb	public, public	ym	0	May 10, 2024 16:37:36 GMT+0...	May 10, 2024 16:37:36 GMT+0...	View
145804037960511623	gaussdb	public, public	ym	0	May 10, 2024 16:37:36 GMT+0...	May 10, 2024 16:37:36 GMT+0...	View
145804037960511654	gaussdb	public, public	ym	0	May 10, 2024 16:37:36 GMT+0...	May 10, 2024 16:37:36 GMT+0...	View
145804037960511592	gaussdb	public, public	ym	0	May 10, 2024 16:37:35 GMT+0...	May 10, 2024 16:37:35 GMT+0...	View
145804037960511583	gaussdb	public, public	ym	0	May 10, 2024 16:37:35 GMT+0...	May 10, 2024 16:37:35 GMT+0...	View
73746443952262571	postgres	"$user",public	Ruby	0	May 10, 2024 08:17:33 GMT+0...	May 10, 2024 08:17:33 GMT+0...	View
145804037960238383	postgres	"$user",public	Ruby	0	May 10, 2024 08:17:33 GMT+0...	May 10, 2024 08:17:33 GMT+0...	View
217861631998334406	postgres	"$user",public	Ruby	0	May 10, 2024 08:17:33 GMT+0...	May 10, 2024 08:17:33 GMT+0...	View
73746443952262568	postgres	"$user",public	Ruby	0	May 10, 2024 08:17:33 GMT+0...	May 10, 2024 08:17:33 GMT+0...	View

Total Records: 709

gaussdb

Alarm Information

No data available.

SQL Statement

```
select pc.relname as table_name from pg_class pc left join
pg_namespace pn on pc.relnamespace = pn.oid where pc.relkind
IN ( 'r', 'f') and pn.nspname= $1 LIMIT 1000 OFFSET 0
params: $1 = 'public'
```

Execution Plan

```
1 | Limit  (cost=1.20..54.90 rows=19 width=64)
2 |  ->Hash Join (3,4)  (cost=1.20..54.90 rows=19 width=64)
  |     Hash Cond: (pc.relnamespace = pn.oid)
3 |   ->Seq Scan on pg_catalog.pg_class pc  (cost=0.00..53.08
rows=115 distinct=1.00 width=68)
  |       Filter: (pc.relkind = ANY ('{r,f}'::"char"[]))
4 |   ->Hash  (cost=1.19..1.19 rows=1 distinct=1.00 width=4)
5 |     ->Seq Scan on pg_catalog.pg_namespace pn
(cost=0.00..1.19 rows=1 width=4)
  |        Filter: (pn.nspname = 'public'::name)
Customer plan is in use
```

----End

Setting GUC Parameters

GUC parameters related to SQL diagnosis are as follows. For details, see "GUC Parameters" in the *Data Warehouse Service (DWS) Developer Guide*.

- **enable_resource_track**
  - Value range: boolean

  - Default value: **on**
  - Expected DMS value: **on** (for reference only)
  - Function: Specifies whether to enable the real-time resource monitoring function.

NOTICE

If this parameter is enabled without other GUC-related parameters correctly configured, real-time resource consumption cannot be recorded.

- **resource_track_cost**
  - Value range: an integer ranging from –1 to INT_MAX
  - Default value: **0**
  - Expected DMS value: **0** (for reference only)
  - Function: Specifies the minimum execution cost of statement resource monitoring for the current session. This parameter is valid only when **enable_resource_track** is **on**.

NOTICE

If this parameter is set to a small value, more statements will be recorded, causing record expansion and affecting cluster performance.

- **resource_track_level**
  - Value range: enumerated type
  - Default value: **query**
  - Expected DMS value: **query** (for reference only)
  - Function: Specifies the resource monitoring level for the current session. This parameter is valid only when **enable_resource_track** is **on**.

NOTICE

If the resource monitoring is set to operator-level, performance will be greatly affected.

- **resource_track_duration**
  - Value range: an integer ranging from 0 to INT_MAX, in seconds
  - Default value: **60**.
  - Expected DMS value: **0** (for reference only)
  - Function: Specifies the minimum statement execution time that determines whether information about jobs of a statement recorded in the real-time view will be dumped to a historical view after the statement is executed. That is, only statements whose execution time exceeds the specified time are recorded in the historical view. This parameter is valid only when **enable_resource_track** is **on**.

NOTICE

If this parameter is set to a small value, the batch processing mechanism for dumping kernel statements becomes invalid, affecting the kernel performance.

- **topsql_retention_time**
  - Value range: an integer ranging from 0 to 3650, in days
  - Default value: **30**
  - Expected DMS value: **14** (for reference only)
  - Function: Specifies the aging time of **pgxc_wlm_session_info** data in the view.

NOTICE

If this parameter is set to **0**, data will not be aged, which will cause storage expansion.

- **enable_resource_record**
  - Value range: boolean
  - Default value: **on**
  - Expected DMS value: **on** (for reference only)
  - Function: Specifies whether to enable the archiving function for resource monitoring records. When this function is enabled, records in the history views (**GS_WLM_SESSION_HISTORY** and **GS_WLM_OPERATOR_HISTORY**) are archived to the info views (**GS_WLM_SESSION_INFO** and **GS_WLM_OPERATOR_INFO**) every 3 minutes. After the archiving, records in the history views are deleted.

NOTICE

When this parameter is enabled, you are advised to set **topsql_retention_time** properly to configure the aging time. Otherwise, data in the **GS_WLM_SESSION_INFO** or **GS_WLM_OPERATOR_INFO** table will expand.

9.1.5.2 SQL Probes

You can upload and verify SQL probes, execute probe tasks in one click, and periodically execute probe tasks. Alarms can be reported for timeout SQL probes. The procedure is as follows:

- **Adding a SQL Probe**
- **Enabling or Disabling a SQL Probe**
- **Modifying an SQL Probe**
- **Deleting a SQL Probe**

- **Executing a SQL Probe in One Click**

NOTE

- The SQL probe is supported only in 8.1.1.300 and later versions. To use it in earlier versions, contact technical support.
- Only **SELECT** statements can be used as SQL probes.
- Up to 20 SQL probes can be configured.
- To create an SQL probe, you must have the GaussDB(DWS) FullAccess permission.
- To enable the SQL probe function, choose **Monitoring Settings** > **Monitoring Collection** and enable the **SQL Probe** metric. For details, see **Monitoring Collection**. The default collection frequency is 30s.

Adding a SQL Probe

Step 1 Log in to the GaussDB(DWS) console.

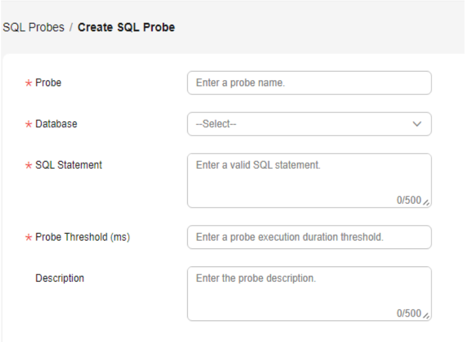
Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane, choose **Utilities** > **SQL Probes**. Click **Add SQL Probe**.

Step 5 Configure SQL probe parameters.

- **Probe Name**: Name of a probe.
- **Database**: Database where the probe SQL statement is to be executed.
- **SQL Statement**: Probe SQL statement to be executed. (Only **SELECT** statements are allowed).
- **Probe Threshold (ms)**: Timeout threshold of probe SQL execution.
- **Description**: Probe SQL statement description.



SQL Probes / **Create SQL Probe**

* Probe	Enter a probe name.
* Database	--Select--
* SQL Statement	Enter a valid SQL statement. 0/500
* Probe Threshold (ms)	Enter a probe execution duration threshold.
Description	Enter the probe description. 0/500

Step 6 Confirm the SQL probe information and click **Confirm**.

----End

Enabling or Disabling a SQL Probe

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, choose **Utilities** > **SQL Probes**.

Step 5 In the probe list, click **Enable** (or **Disable**) in the **Operation** column of a probe.

Step 6 Confirm the information and click **OK**.

----End

Modifying an SQL Probe

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, choose **Utilities** > **SQL Probes**.

Step 5 In the probe list, click **Modify** in the **Operation** column of a probe.

Step 6 On the **Modify Probe** page, modify the SQL probe parameters as required and click **OK**.

----End

Deleting a SQL Probe

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, choose **Utilities** > **SQL Probes**.

Step 5 In the probe list, click **Delete** in the **Operation** column of a probe.

Step 6 Confirm the information and click **OK**.

----End

Executing a SQL Probe in One Click

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane on the left, choose **Utilities** > **SQL Probes**.

Step 5 In the probe list, select a probe and click **Run**. The system will execute the selected probe and update information about the probe.

Step 6 Confirm the information and click **OK**.

----End

9.1.5.3 Table Diagnosis

GaussDB(DWS) provides statistics and diagnostic tools for you to learn table status, including:

- **Skew Rate**: monitors and analyzes data table statistics in the cluster, and displays information about the 50 largest tables whose skew rate is higher than 5%.
- **Dirty Page Rate**: monitors and analyzes data table statistics in the cluster, and displays information about the 50 largest tables whose skew rate is higher than 50%.
- **DDL Audit**: DDL review is a type of SQL review. To prevent improper DDL design from affecting services, this tool checks whether DDL metadata is standard, detecting potential table definition problems in advance. The check result can also be used as a reference for locating performance issues.

NOTE

- Only 8.1.1.x and later versions support the table skew rate and dirty page rate features. For earlier versions, contact technical support.
- Only 8.1.1.300 and later versions support the DDL review feature. For earlier versions, contact technical support.
- The collection period of the table skew rate and dirty page rate can be configured on the **Monitoring Collection** page of the cluster. Frequent collection may affect cluster performance. Set a proper period based on your cluster workloads.

Skew Rate

Context

Improper distribution columns can cause severe skew during operator computing or data spill to disk. The workloads will be unevenly distributed on DNs, resulting in high disk usage on a single DN and affecting performance. You can query your table size and skew rate, and change the distribution columns of tables with severe skew. In cluster versions 8.1.0 and later, you can use the syntax **ALTER TABLE**. In other cluster versions, perform the operations described in **How Do I Change Distribution Columns?**.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation tree on the left, choose **Utilities** > **Table Diagnosis** and click the **Skew Rate** tab. The tables that meet the statistics collection conditions in the cluster are displayed.

----End

Dirty Page Rate

Context

DML operations on tables may generate dirty data, which unnecessarily occupies cluster storage. You can query the dirty page rate of tables, and optimize large tables and tables with high dirty page rate. For details, see **Solution to High Disk Usage and Cluster Read-Only**.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation tree on the left, choose **Utilities** > **Table Diagnosis** and click the **Dirty Page Rate** tab. The tables that meet the statistics collection conditions in the cluster are displayed.

----End

DDL Audit

Viewing and Exporting DDL Audit Results

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation tree on the left, choose **Utilities** > **Table Diagnosis**, and click the **DDL Audit** tab. The audit results are displayed.

NOTE

The selected audit items are displayed on the **DDL Audit** tab by default. You can configure the audit items on the **Monitoring Collection** tab. For more information, see **Table 9-3**.

Table 9-3 Audit items

Item	Description
Number of Distribution Keys (disKeyCount)	If there is no data skew, use no more than four distribution keys. Generally, if you use many distribution keys, data can be evenly distributed in a cluster, thus avoid data skew. However, if too many distribution keys are used, the storage performance and joint query performance may deteriorate. You are advised to configure no more than four distribution keys. ● Storage performance issue: When data is added, the hash function calculates the result of each distribution column, aggregates the results, and then determine where to distribute data. A large number of distribution keys require time-consuming, complex calculation. ● Union query performance issue: During multi-table join query, if all the columns of the distribution key are involved in the join condition, data does not need to be redistributed in the execution plan. If a large number of distribution keys are used, some of them may not be the columns involved in the join condition, and data redistribution may occur, which consumes many resources and takes long.
Number of Index Columns/PCKs (indexKeyOrPckCount)	It is recommended that the number of partial cluster keys (PCKs)/columns of an index be less than or equal to 4. ● A large number of index columns require many resources to maintain index data, and are likely to contain duplicate indexes. ● While column-store data is imported, PCK columns are compared and calculated to determine CU division. A large number of PCKs will consume many resources and much time, affecting performance. To efficiently filter CUs in a query, the prefixes of the columns involved in the query conditions must be PCK columns. (For example, if the PCK columns are **a**, **b**, and **c**, the query criteria must be **a>? and b>? and c>?**.) Otherwise, all the CUs must be traversed, and data clustering does not contribute to query acceleration.

Item	Description
Invalid PCKs (invalidPck)	Do not create invalid PCK columns. In 8.1.1 and later versions, the cluster can filter and compare data of the char, int8, int2, int4, text, bpchar, varchar, date, time, timestamp and timestamptz types. If a column of an unsupported data type is used as a PCK, the column is an invalid PCK column. It does not take effect during CU filtering and will consume resources for its maintenance.
numeric Data Usage (validityOfNumeric)	When numeric data types are used, use integers if possible. If the precision requirement is not high, use the float fixed-length data type. The float fixed-length data type has better computing performance than the numeric variable-length data type. That is, if the numeric type is used, you are advised to specify the scale and precision within 38 bits. When the numeric type is used for calculation, the underlying layer attempts to convert the calculation to the calculation between int and bigint to improve the calculation efficiency. That is, the large integer optimization of the data type is used. In 8.1.1 and later versions, if no precision is specified, a maximum of 131,072 digits can be placed before the decimal point and a maximum of 16,383 digits can be placed after the decimal point. That is, the maximum scale and precision are used. In this case, large integer optimization cannot be performed during calculation, and the calculation efficiency decreases accordingly.
Index Column Width (widthOfIndexKey)	Generally, wide index columns are character string columns, which do not involve compare operations and will lead to large indexes that consume unnecessary space. Specify a value smaller than 64 bytes.
Replication Table Size (sizeOfCopyTable)	Tables that occupy more storage space than the threshold (100 MB) on a single DN will be identified. For such tables, you are advised to use common associated columns as distribution keys (generally with one primary key). The cluster supports replication tables. A replication table maintains a full copy of data on each node and is mainly used to store data of enumerated types. If a table contains too much data, it will occupy a large amount of space. In addition, in a union query, the node traverses all table data, which may take a longer time than the union query after the table type is changed to distribution table. (Although data may be redistributed in the distribution table, the amount of data traversed by each node decreases.)

Item	Description
Skew Detection for Single-Distribution-Key Tables (recognitionOfDataSkew)	Data skew of single-distribution-key tables is detected by statistics. This audit applies only to tables with one distribution key.
Distribution Key Usage (validityOfDiskey)	In a cluster, you are not advised to use a column of the Boolean or date type as a distribution column, because it may cause data skew.
Number of Cached Sequence Values (cacheSizeOfSequence)	Specify a number greater than 100. If a table column uses sequences, its **next_value** is obtained from the cached value in the local node. If cached sequence values are used up, a request will be sent asking GTM to obtain the value again. If a large amount of data is added but only a few values are cached, GTM will receive many requests, and may get overloaded and even break down. To avoid this problem, you are advised to set the cache value to a value greater than 100 when creating a sequence.
Optimizable Indexes (optimizableIndexKey)	Scenarios where indexes can be optimized: ● The index column of an index is the first *N* columns of another index. ● The index columns of two indexes are the same, but the orders are different.

Step 5 If the review result of an item is **Failed**, click **View** to go to the details page.

Step 6 Click **Export** in the upper left corner to export the audit result.

----End

Manually Auditing DDL Items

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation tree on the left, choose **Utilities** > **Table Diagnosis**, and click the **DDL Audit** tab. On the page that is displayed, select the items to be audited and click **One-Click Audit**.

----End

9.1.6 Workload Analysis

9.1.6.1 Workload Analysis Overview

The workload analysis tool of GaussDB(DWS) collects and analyzes database performance data. You can create workload snapshots to record cluster workload data in a specified period. A workload diagnosis report can be generated based on two workload information snapshots within a certain time segment. Workload Diagnosis Report (WDR) provides performance data in a specified period and presents the data on HTML web pages. It helps you detect exceptions, diagnose problems, and optimize performance. It is a powerful tool for database performance tuning.

NOTE

- The WDR function is available only in 8.1.1.300 and later cluster versions.
- Workload diagnosis reports can be stored only in OBS.

9.1.6.2 Workload Snapshots

You can check the basic information about the cluster workload snapshots, manually create a snapshot, and configure snapshot parameters.

Checking Workload Snapshots

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Cluster** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Snapshot**. Workload snapshots will be displayed.

----End

Creating a Workload Snapshot

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Cluster** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Snapshot**. Workload snapshots will be displayed.

Step 5 Click **Create Snapshot**. Enter a snapshot name and click **OK**.

NOTE

Before creating a workload snapshot, ensure that the performance view snapshot parameter is enabled. For details, see **Configuring Workload Snapshot Parameters**.

----End

Configuring Workload Snapshot Parameters

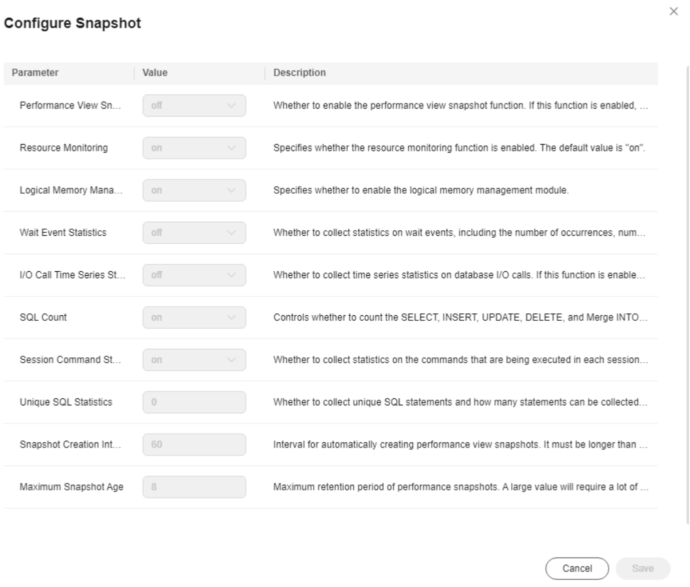
Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Snapshot**. Workload snapshots will be displayed.

Step 5 Click **Configure Snapshot** in the upper right corner. In the dialog box that is displayed, check or modify GUC parameters. For details, see **Table 9-4**.



Configure Snapshot

Parameter	Value	Description
Performance View Sn...	off	Whether to enable the performance view snapshot function. If this function is enabled, ...
Resource Monitoring	on	Specifies whether the resource monitoring function is enabled. The default value is "on".
Logical Memory Mana...	on	Specifies whether to enable the logical memory management module.
Wait Event Statistics	off	Whether to collect statistics on wait events, including the number of occurrences, num...
I/O Call Time Series St...	off	Whether to collect time series statistics on database I/O calls. If this function is enable...
SQL Count	on	Controls whether to count the SELECT, INSERT, UPDATE, DELETE, and Merge INTO...
Session Command St...	on	Whether to collect statistics on the commands that are being executed in each session...
Unique SQL Statistics	0	Whether to collect unique SQL statements and how many statements can be collected...
Snapshot Creation Int...	60	Interval for automatically creating performance view snapshots. It must be longer than ...
Maximum Snapshot Age	8	Maximum retention period of performance snapshots. A large value will require a lot of ...

----End

Table 9-4 Workload snapshot parameters

Name	Default Value	Description
Performance view snapshot (**enable_wdr_snapshot**)	off	Whether to enable the performance view snapshot function. If this function is enabled, GaussDB(DWS) will periodically create snapshots for certain system performance views and save them to disk. You can also manually create snapshots.
Resource monitoring (**enable_resource_track**)	on	Whether to enable the resource monitoring function. Resource statistics parameters are valid only if this parameter is enabled.

Name	Default Value	Description
Logical memory management module (**enable_memory_limit**)	on	Whether to enable the logical memory management module.
Wait event statistics (**enable_track_wait_event**)	off	Whether to collect statistics on wait events, including the number of occurrences, number of failures, duration, maximum waiting time, minimum waiting time, and average waiting time.
I/O call time series statistics (**track_io_timing**)	off	Whether to collect time series statistics on database I/O calls. If this function is enabled, the collector will periodically query the OS time, which may cause heavy overhead on certain platforms.
SQL count (**track_sql_count**)	The default value is **off** for versions earlier than 8.1.3 and **on** for 8.1.3 and later versions.	Whether to collect statistics on the number of the **SELECT**, **INSERT**, **UPDATE**, **DELETE**, and **MERGE INTO** statements that are being executed in each session, the response time of the **SELECT**, **INSERT**, **UPDATE**, and **DELETE** statements, and the number of DDL, DML, and DCL statements. This parameter takes effect only if **track_activities** is set to **on**.
Session command statistics (**track_activities**)	on	Whether to collect statistics on the commands that are being executed in each session.
Unique SQL statistics (**instr_unique_sql_count**)	0	Whether to collect unique SQL statements and how many statements can be collected.
Snapshot creation interval (**wdr_snapshot_interval**)	60	Interval for automatically creating performance view snapshots. It must be longer than the time needed to create a snapshot. The unit is minute.
Maximum snapshot retention period (**wdr_snapshot_retention_days**)	8	Maximum retention period of performance snapshots. A large value will require a lot of disk space. The unit is day.

9.1.6.3 Workload Reports

You can create, download, and delete work diagnosis reports, and check historical workload diagnosis reports.

NOTE

To create a workload report, obtain the required OBS bucket permissions first.

Checking Workload Reports

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**. Workload reports will be displayed.

----End

Generating a Workload Report

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**.

Step 5 Click **Generate Report**. In the displayed dialog box, configure the following parameters and click **OK**:

Table 9-5 Parameters for generating a report

Parameter	Description	Example Value
Report Name	User-defined. Ensure that the name is unique and contains a maximum of 100 characters, including digits, letters, and underscores (_).	test_show
Object Model	The object types are as follows: ● **node**: The performance data of a specified node will be provided. ● **cluster**: The performance data of the entire cluster will be provided.	node

Parameter	Description	Example Value
Node Name	User defined	dn_6005_l006
Content Type	The options are as follows: ● **summary**: A report contains only brief analysis and calculation results. ● **detail**: A report contains only detailed metric data. ● **all**: A report contains content of both the summary and detail reports.	all
Starting Snapshot	User defined **NOTE** The time of the starting snapshot start must be earlier than that of the ending snapshot.	-
Ending Snapshot	User defined	-
OBS Bucket	Bucket name, which is used to store reports.	test123
OBS Path	Storage directory, which can be customized. Multi-level directories can be separated by slashes (/) and cannot start with slashes (/). Up to 50 characters are allowed.	wdr

----End

Downloading Workload Reports in Batches

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**.

Step 5 Select reports and click **Download**.

NOTE

Up to 10 report records can be downloaded at a time.

----End

Deleting Workload Reports in Batches

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**.

Step 5 Select reports and click **Delete**.

----End

Deleting a Workload Diagnosis Report

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**.

Step 5 Click **Delete** in the **Operation** column of a report to delete the report record and file.

----End

Configuring Workload Report Parameters

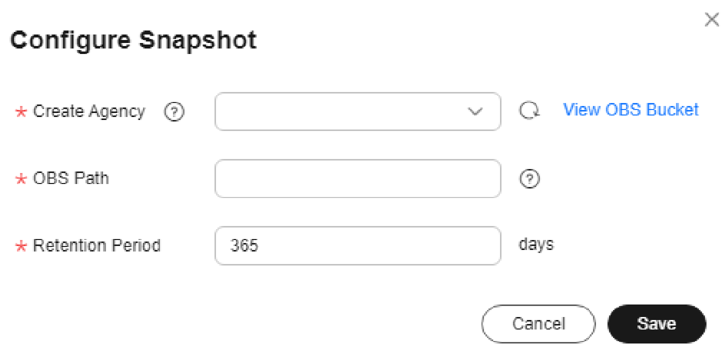
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster for which you want to perform workload analysis.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**. The database monitoring page is displayed.

Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**.

Step 5 Click **Configure Report** in the upper right corner. In the displayed dialog box, set the report retention period and OBS parameters.



Configure Snapshot

* Create Agency

View OBS Bucket

* OBS Path

* Retention Period: 365 days

----End

9.1.7 Settings

The **Monitoring** page displays the collection period of monitoring metrics.

NOTE

- The cluster monitoring function is enabled by default.
- Disable the function if the cluster is being recovered. Enable the function when the fault is rectified.
- When a node in the cluster is powered off or the management IP address of the cluster is unavailable, the cluster monitoring switch and the button for configuring cluster indicator collection are unavailable.

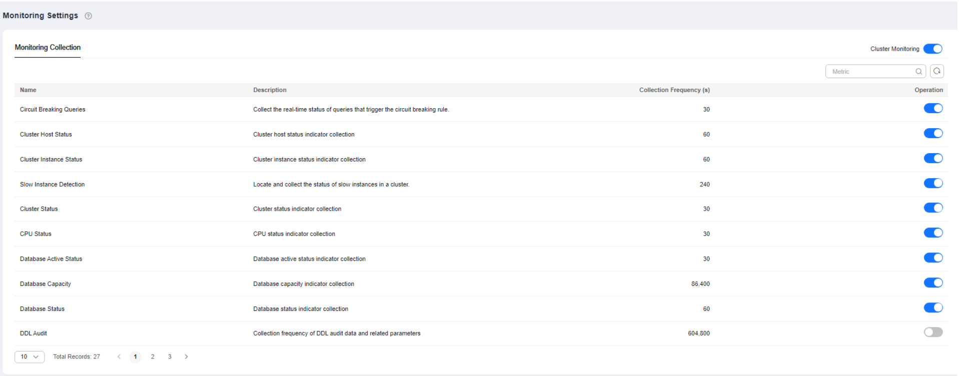
Monitoring Collection

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Monitoring Settings** > **Monitoring Collection**. You can reconfigure the collection frequency or disable the collection of the monitoring item.



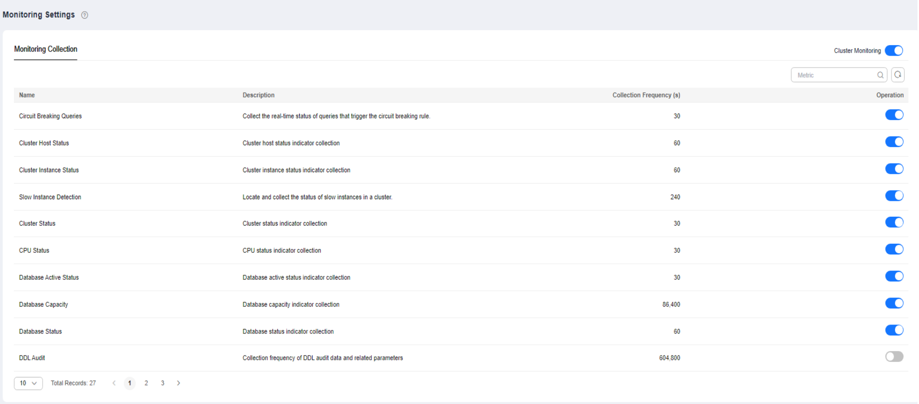
Monitoring Settings

Monitoring Collection

Cluster Monitoring ●

Name	Description	Collection Frequency (s)	Operation
Circuit Breaking Queries	Collect the real-time status of queries that trigger the circuit breaking rule.	30	●
Cluster Host Status	Cluster host status indicator collection	60	●
Cluster Instance Status	Cluster instance status indicator collection	60	●
Slow Instance Detection	Locate and collect the status of slow instances in a cluster.	240	●
Cluster Status	Cluster status indicator collection	30	●
CPU Status	CPU status indicator collection	30	●
Database Active Status	Database active status indicator collection	30	●
Database Capacity	Database capacity indicator collection	86,400	●
Database Status	Database status indicator collection	60	●
DDL Audit	Collection frequency of DDL audit data and related parameters	604,800	○

10 Total Records: 27



Monitoring Settings

Monitoring Collection

Cluster Monitoring ●

Name	Description	Collection Frequency (s)	Operation
Circuit Breaking Queries	Collect the real-time status of queries that trigger the circuit breaking rule.	30	●
Cluster Host Status	Cluster host status indicator collection	60	●
Cluster Instance Status	Cluster instance status indicator collection	60	●
Slow Instance Detection	Locate and collect the status of slow instances in a cluster.	240	●
Cluster Status	Cluster status indicator collection	30	●
CPU Status	CPU status indicator collection	30	●
Database Active Status	Database active status indicator collection	30	●
Database Capacity	Database capacity indicator collection	86,400	●
Database Status	Database status indicator collection	60	●
DDL Audit	Collection frequency of DDL audit data and related parameters	604,800	○

10 Total Records: 27 < 1 2 3 >

----End

9.1.8 Checking Task Details

On the task details page, you can view the status of tasks, such as enabling, disabling, resetting, and modifying cluster monitoring collection items; one-click DDL review; load snapshot generation; load diagnosis report generation; session termination; query termination; and the addition, modification, deletion, and one-click execution of probes.

NOTE

Only 8.1.3.110 and later cluster versions support the task details page.

Prerequisites

Tasks executed by users are related to SQL probes, load analysis, DDL one-click review, or monitoring collection items.

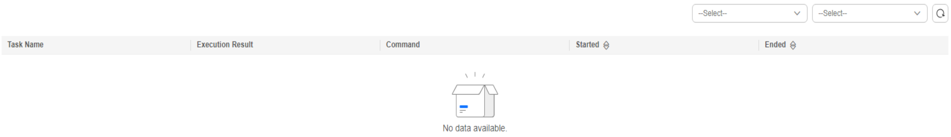
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the cluster, click **Monitoring Panel**. The database overview page is displayed.

Step 4 In the navigation pane on the left, choose **Tasks** to view the execution details of the commands delivered by the cluster. Task information includes the task name, task execution result, task command, start time, and end time.



Task Name	Execution Result	Command	Started	Ended

No data available.

----End

9.1.9 Typical Scenarios

9.1.9.1 SQL Diagnosis

Symptom

The execution of SQL statements takes a long time, resulting in great resource consumption.

Troubleshooting Process

If the execution efficiency of SQL statements is low, optimization suggestions are provided after the kernel executes the SQL statements. You can query the execution history to retrieve optimization suggestions and further optimize SQL statements to improve query efficiency.

Troubleshooting Procedure

Step 1 On the **SQL Diagnosing** page, select a time period that does not seem right.

Step 2 Search for SQL statements based on indicators such as the start time, end time, and running duration of the statement.

Step 3 Click **View** to view SQL optimization suggestions.

Step 4 Optimize the SQL statement based on suggestions.

----End

9.2 Viewing GaussDB(DWS) Cluster Monitoring Information on Cloud Eye

Function

This section describes how to check cluster metrics on Cloud Eye. By monitoring cluster running metrics, you can identify the time when the database cluster is abnormal and analyze potential activity problems based on the database logs, improving database performance. This section describes the metrics that can be monitored by Cloud Eye as well as their namespaces and dimensions. You can use the management console or **APIs** provided by Cloud Eye to query the monitoring metrics and alarms generated by GaussDB(DWS).

Namespace

SYS.DWS

Cluster Monitoring Metrics

With the GaussDB(DWS) monitoring metrics provided by Cloud Eye, you can obtain information about the cluster running status and performance. This information will provide a better understanding of the node-level information.

Table 9-6 describes GaussDB(DWS) monitoring metrics.

Table 9-6 GaussDB(DWS) monitoring metrics

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
dws001_shared_buffer_hit_ratio	Cache Hit Ratio	Ratio of requested data that already exists in the cache. This refers to the ratio between the data currently stored in the cache and the total amount of data that has been requested. A higher cache hit ratio means higher cache usage of the system, fewer times that data needs to be read from the disk or network, and faster system response speed. Unit: Percent	0% to 100%	Data warehouse cluster	4 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
dws002_in_memory_sort_ratio	In-memory Sort Ratio	Ratio of the extra memory space used by the sorting algorithm to the memory space occupied by the sorted data. In a merge sort, for example, the size of the merge buffer is often proportional to the size of the sorted data, so the in-memory ratio is usually between 10% and 50%. Unit: Percent	0% to 100%	Data warehouse cluster	4 minutes
dws003_physical_reads	File Reads	Total number of database file reads	> 0	Data warehouse cluster	4 minutes
dws004_physical_writes	File Writes	Total number of database file writes	> 0	Data warehouse cluster	4 minutes
dws005_physical_reads_per_second	File Reads per Second	Number of database file reads per second	≥ 0	Data warehouse cluster	4 minutes
dws006_physical_writes_per_second	File Writes per Second	Number of database file writes per second	≥ 0	Data warehouse cluster	4 minutes
dws007_db_size	Data Volume	Total size of data in the database, in MB	≥ 0 MB	Data warehouse cluster	4 minutes

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
dws008_active_sql_count	Active SQL Count	Number of active SQLs in the database	≥ 0	Data warehouse cluster	4 minutes
dws009_session_count	Session Count	Number of sessions that access the database	≥ 0	Data warehouse cluster	4 minutes
dws010_cpu_usage	CPU Usage	CPU usage of each node in a cluster, in percentage	0% to 100%	Data warehouse node	1 minute
dws011_mem_usage	Memory Usage	Memory usage of each node in a cluster, in percentage **NOTE** After the console is upgraded to 8.3.0.202, the memory usage includes the memory occupied by the cache. Therefore, the value of this metric increases compared with that before the upgrade.	0% to 100%	Data warehouse node	1 minute
dws012_iops	IOPS	Number of I/O requests processed by each node in the cluster per second	≥ 0	Data warehouse node	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
dws013_bytes_in	Network Input Throughput	Data input to each node in the cluster per second over the network Unit: byte/s	≥ 0 bytes/s	Data warehouse node	1 minute
dws014_bytes_out	Network Output Throughput	Data sent to the network per second from each node in the cluster Unit: byte/s	≥ 0 bytes/s	Data warehouse node	1 minute
dws015_disk_usage	Disk Usage	Disk usage of each node in a cluster, in percentage	0% to 100%	Data warehouse node	1 minute
dws016_disk_total_size	Total Disk Size	Total disk space of each node in the cluster Unit: GB	100 to 2000 GB	Data warehouse node	1 minute
dws017_disk_used_size	Used Disk Space	Used disk space of each node in the cluster Unit: GB	0 to 3600 GB	Data warehouse node	1 minute
dws018_disk_read_throughput	Disk Read Throughput	Data volume read from each disk in the cluster per second Unit: byte/s	≥ 0 bytes/s	Data warehouse node	1 minute
dws019_disk_write_throughput	Disk Write Throughput	Data volume written to each disk in the cluster per second Unit: byte/s	≥ 0 bytes/s	Data warehouse node	1 minute

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
dws020_avg_disk_sec_per_read	Average Time per Disk Read	Average time used each time when a disk reads data Unit: second	> 0s	Data warehouse node	1 minute
dws021_avg_disk_sec_per_write	Average Time per Disk Write	Average time used each time when data is written to a disk Unit: second	> 0s	Data warehouse node	1 minute
dws022_avg_disk_queue_length	Average Disk Queue Length	Average I/O queue length of a disk	≥ 0	Data warehouse node	1 minute
dws_024_dn_diskio_util	DN I/O usage	Average disk I/O usage of DNs in a cluster	0% to 100%	Data warehouse instance	1 minute

Dimensions

Key	Value
datastore_id	Data warehouse cluster ID
dws_instance_id	Data warehouse node ID

Cluster and Node Monitoring Information

Step 1 Log in to the GaussDB(DWS) console and choose **Clusters** > **Dedicated Clusters**.

Step 2 **View the cluster information**. In the cluster list, click **View Metric** in the **Operation** column where a specific cluster resides. The Cloud Eye management console is displayed. By default, the cluster monitoring information on the Cloud Eye management console is displayed.

Additionally, you can specify a specific monitoring metric and the time range to view the performance curve.

Step 3 **View the node information**. Click ‹ to return to the Cloud Eye management console. On the **Data Warehouse Nodes** tab page in the right pane, you can view metrics of each node in the cluster.

Additionally, you can specify a specific monitoring metric and the time range to view the performance curve.

Cloud Eye also supports the ability to compare the monitoring metrics of multiple nodes. For details, see **Comparing the Monitoring Metrics of Multiple Nodes**.

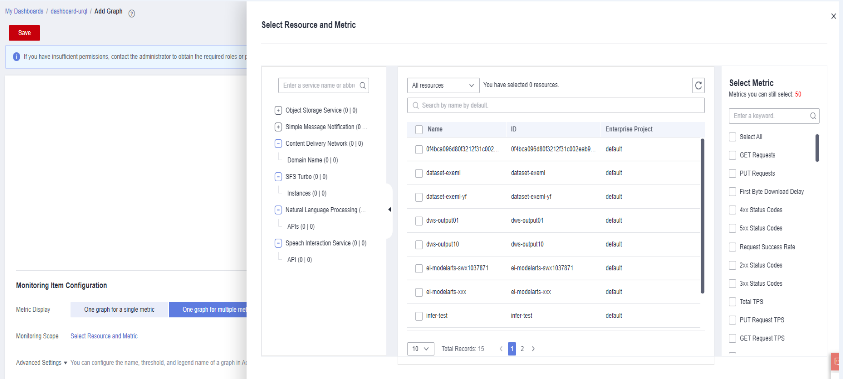
----End

Comparing the Monitoring Metrics of Multiple Nodes

Step 1 In the navigation pane of the Cloud Eye management console, choose **Dashboards** > **My Dashboards**. Click the name of the dashboard for which you want to add a graph. On the **My Dashboards** page that is displayed, click **Add Graph**.

Step 2 On the **Add Graph** page, you can select **Line Chart** or **Bar Chart** to display the graph. After confirming that the information is correct, click **OK**.

For example, select **Line Chart** and **One View for Multiple Metrics** to compare the CPU usage of three GaussDB(DWS) nodes. The following table describes the parameters.


My Dashboards / dashboard-urql / Add Graph
Save
If you have insufficient permissions, contact the administrator to obtain the required roles or p
Monitoring Item Configuration
Metric Display
One graph for a single metric
One graph for multiple met
Monitoring Scope
Select Resource and Metric
Advanced Settings
You can configure the name, threshold, and legend name of a graph in Ad
Select Resource and Metric
Enter a service name or abbr
Object Storage Service (0 | 0)
Simple Message Notification (0 ...
Content Delivery Network (0 | 0)
Domain Name (0 | 0)
SFS Turbo (0 | 0)
Instances (0 | 0)
Natural Language Processing (...
APIs (0 | 0)
Speech Interaction Service (0 | 0)
API (0 | 0)
All resources
You have selected 0 resources.
Search by name by default.
10
Total Records: 15
1
2
Select Metric
Metrics you can still select: 50
Enter a keyword.
Select All
GET Requests
PUT Requests
First Byte Download Delay
4xx Status Codes
5xx Status Codes
Request Success Rate
2xx Status Codes
3xx Status Codes
Total TPS
PUT Request TPS
GET Request TPS

Name	ID	Enterprise Project
0f4bca096d80f3212f31c002...	0f4bca096d80f3212f31c002eab9...	default
dataset-exeml	dataset-exeml	default
dataset-exeml-yf	dataset-exeml-yf	default
dws-output01	dws-output01	default
dws-output10	dws-output10	default
ei-modelarts-swx1037871	ei-modelarts-swx1037871	default
ei-modelarts-xxx	ei-modelarts-xxx	default
infer-test	infer-test	default

Table 9-7 Configuration example

Parameter	Example Value
Resource Type	DWS
Dimension	Data Warehouse Node
Monitored Object	dws-demo-dws-cn-cn-2-1 dws-demo-dws-cn-cn-1-1 dws-demo-dws-dn-1-1
Metric	CPU Usage

Step 3 Click **OK**.

On the selected **My Dashboards** page, you can view the metric trend on the newly added monitoring graph. You can click the zoom in button to zoom in and view detailed metric comparison data.

----End

Creating Alarm Rules

GaussDB(DWS) enables you to customize alarm rules for monitoring specific objects and notification policies, ensuring you stay informed about its running status in a timely manner.

A GaussDB(DWS) alarm rule includes the alarm rule name, monitored object, metric, threshold, monitoring interval, and whether to send a notification. This section describes how to set GaussDB(DWS) alarm rules.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 Locate the row containing the target cluster, click **View Metric** in the **Operation** column to enter the Cloud Eye management console and view the GaussDB(DWS) monitoring information.

The status of the target cluster must be **Available**. Otherwise, you cannot create alarm rules.

Step 4 In the left navigation pane of the Cloud Eye management console, choose **Alarm Management** > **Alarm Rules**.

Step 5 On the **Alarm Rules** page, click **Create Alarm Rule** in the upper right corner.

Step 6 On the **Create Alarm Rule** page, set parameters as prompted.

1. Configure the rule name and description.
2. Configure the alarm parameters as prompted.

Table 9-8 Configuring alarm parameters

Parameter	Description	Example Value
Resource Type	Name of the cloud service resource for which the alarm rule is configured.	Data Warehouse Service
Dimension	Metric dimension of the alarm rule. You can select **Data Warehouse Nodes** or **Data Warehouses**.	Data Warehouse Node
Monitoring Scope	Resource scope to which an alarm rule applies. Select **Specific resources** and select one or more monitoring objects. For GaussDB(DWS), select the cluster ID or node ID in the dialog box that is displayed.	Specific resources

Parameter	Description	Example Value
Trigger Rule	You can select an associated template, use an existing template or create a custom template as required.	Create manually
Template	This parameter is valid only when **Use template** is selected. Select the template to be imported. If no alarm template is available, click **Create Custom Template** to create one that meets your requirements.	-
Alarm Policy	This parameter is valid only when **Create manually** is selected. Set the policy that triggers an alarm. For example, trigger an alarm if the CPU usage equals to or is greater than 80% for 3 consecutive periods. **Table 9-6** lists the GaussDB(DWS) monitoring metrics.	-
Alarm Severity	Severity of an alarm. Valid values are **Critical**, **Major**, **Minor**, and **Informational**.	Major

3. Configure the alarm notification parameters as prompted.

Table 9-9 Configuring alarm notifications

Parameter	Description	Example Value
Alarm Notification	Whether to notify users when alarms are triggered. Notifications can be sent as emails or text messages, or HTTP/HTTPS requests sent to the servers. You can enable (recommended) or disable **Alarm Notification**.	Enable
Validity Period	Cloud Eye sends notifications only within the validity period specified in the alarm rule. For example, if **Validity Period** is set to **00:00-8:00**, Cloud Eye sends notifications only within 00:00-8:00.	-

Parameter	Description	Example Value
Notification Object	Name of the topic to which the alarm notification is sent. If you enable **Alarm Notification**, you need to select a topic. If no desired topics are available, create one first, whereupon the SMN service is invoked. For details about how to create a topic, see the *Simple Message Notification User Guide*. For details about how to create a topic, see the **Simple Message Notification User Guide**.	-
Trigger Condition	Condition for triggering the alarm. You can select **Generated alarm**, **Cleared alarm**, or both.	-

4. After the configuration is complete, click **Next**.

   After the alarm rule is created, if the metric data reaches the specified threshold, Cloud Eye will immediately inform you that an exception has occurred.

----End

Transferring Data to OBS

Raw data of metrics is kept for two days on Cloud Eye. You can enable OBS and save the raw data to OBS so that it can be saved for a longer time.

For how to configure OBS storage transfer, see "Viewing Alarm History" > "Configuring OBS Data Storage" in the ***Cloud Eye User Guide***.

9.3 Monitoring and Diagnosing Top SQL Statements in a GaussDB(DWS) Cluster

Context

GaussDB(DWS) offers a multi-dimensional optimization and diagnosis function to enhance self-O&M capability for tenants. It helps identify slow and abnormal SQL statements to ensure the smooth and efficient operation of user services. This allows you to query, diagnose, and analyze historical data, perform real-time queries and analysis, and conduct real-time session analysis.

- **Historical query analysis**: offers the ability to diagnose exceptions by monitoring the top SQL statements in the historical data. It presents SQL trend statistics and analysis curves that showcase the historical execution trend of SQL statements. It identifies top SQL exceptions, detects slow SQL statements that consume significant resources and have long execution times, and displays the count of abnormal SQL statements for each type. It also filters out these abnormal SQL statements. Additionally, it provides a one-click diagnosis for individual SQL statements, including checking the statement, diagnosing the execution plan, and visualizing the results. This assists users in analyzing the execution plans and performance impact of their SQL statements.

- **Real-time Query Analysis**: analyzes the real-time resource consumption and query plans that are being executed based on the real-time top SQL statement monitoring. This page provides information on real-time concurrency and user distribution, lock waiting jobs, slow SQL statements, SQL queues, service concurrency trend chart, real-time queries, and real-time data aggregation.
- **Real-Time Session Analysis**: analyzes the real-time session query details. This page displays the number of real-time sessions and their distribution among users, the number of idle and active sessions, the number of jobs in the CCN queue, session quantity trend chart, real-time session list, and real-time data aggregation.

NOTE

- This feature is supported only by clusters of version 8.1.3 or later.
- The real-time query monitoring function is disabled by default. To enable it, choose **Settings** > **Monitoring**, click **Monitoring Collection**, and enable **Real-Time Query Monitoring**. For details, see **Monitoring Collection**. Exercise caution when enabling this as it may generate a large amount of data.
- The historical query monitoring function is disabled by default. To enable it, choose **Settings** > **Monitoring**, click **Monitoring Collection**, and enable **Historical Query Monitoring**. For details, see **Monitoring Collection**. Exercise caution when enabling this as it may generate a large amount of data.

Accessing the Optimization Diagnosis Page

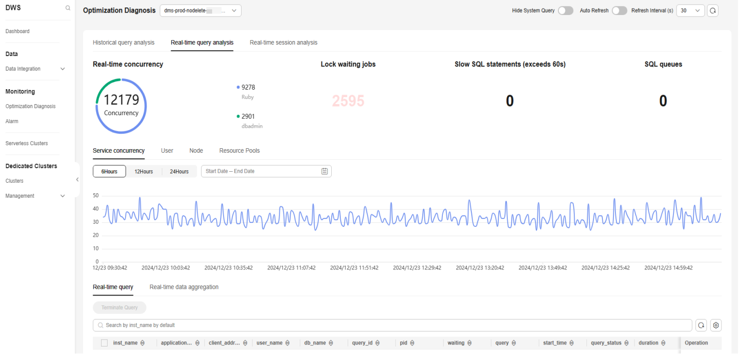
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Monitoring** > **Optimization Diagnosis** on the navigation pane.

Step 3 Select the cluster to be optimized from the drop-down list in the upper left corner of the page. Click **Historical query analysis**, **Real-time query analysis**, and **Real-time session analysis** as needed.

- Toggle on **Hide System Query** in the upper right corner to hide the system user queries.
- To automatically refresh data, toggle on **Auto Refresh** in the upper right corner. You can also set the desired refresh interval by specifying **Refresh Interval (s)** for the data to update at regular intervals.

Figure 9-10 Optimization Diagnosis page

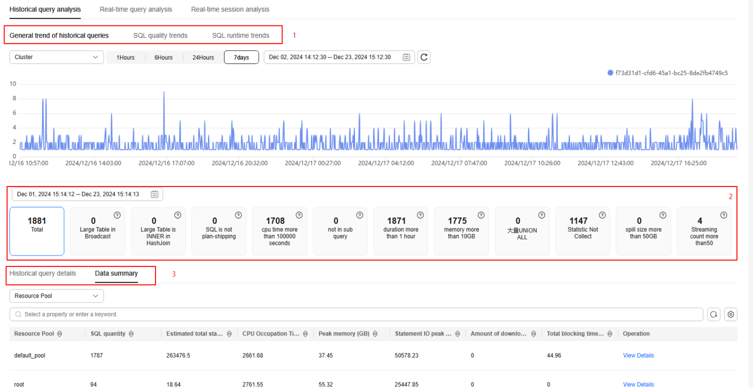


----End

Historical query analysis

This **Historical query analysis** page contains three tab pages: **General trend of historical queries**, **SQL quality trends**, and **SQL runtime trends**. On this page, you can find an overview of SQL quality, detailed information about the top historical SQL queries, perform one-click diagnosis of the execution plan and SQL statements for the top historical SQL statements, and get a summary of historical data.

Figure 9-11 Historical query analysis


Historical query analysis
Real-time query analysis
Real-time session analysis
General trend of historical queries
SQL quality trends
SQL runtime trends
1
Cluster
1Hours
6Hours
24Hours
7days
Dec 02, 2024 14:12:30 – Dec 23, 2024 15:12:30
f73d31d1-cfd6-45a1-bc25-8de2fb4749c5
10
8
6
4
2
0
12/16 10:57:00
2024/12/16 14:03:00
2024/12/16 17:07:00
2024/12/16 20:32:00
2024/12/17 00:27:00
2024/12/17 04:12:00
2024/12/17 07:47:00
2024/12/17 10:26:00
2024/12/17 12:43:00
2024/12/17 16:25:00
Dec 01, 2024 15:14:12 – Dec 23, 2024 15:14:13
2
1881 Total
0 Large Table in Broadcast
0 Large Table is INNER in HashJoin
0 SQL is not plan-shipping
1708 cpu time more than 100000 seconds
0 not in sub query
1871 duration more than 1 hour
1775 memory more than 10GB
0 大量UNION ALL
1147 Statistic Not Collect
0 spill size more than 50GB
4 Streaming count more than50
Historical query details
Data summary
3
Resource Pool
Select a property or enter a keyword.

Resource Pool	SQL quantity	Estimated total sta...	CPU Occupation Ti...	Peak memory (GB)	Statement IO peak ...	Amount of downlo...	Total blocking time...	Operation
default_pool	1787	263476.5	2661.68	37.45	50578.23	0	44.96	View Details
root	94	18.64	2761.55	55.32	25447.85	0	0	View Details

- **General trend of historical queries**: displays aggregated statistics by different dimensions (cluster, user, instance, application name, and resource pool). You can also check the number of SQL statements executed per minute within a specified period on this tab page.
- **SQL quality trends**: collects statistics on the number of SQL statements of each exception type within a specified time range.
- **SQL runtime trends**: collects statistics on the historical SQL execution duration (minimum, average, and maximum durations) of a cluster.
- SQL statement performance overview: displays the number of abnormal SQL statements of each type. You can click different cards to filter the exception types.

  Exception types include execution plan pushdown failures, CPU usage exceeding 100,000 seconds, the presence of **NOT IN** subqueries, query duration exceeding 1 hour, memory usage exceeding 10 GB, disk space usage exceeding 50 GB, and the number of streaming tasks exceeding 50.

NOTE

For how to customize exception thresholds for different cluster, contact technical support. Exception diagnosis rules can be flexibly configured based on the historical top SQL fields in a GaussDB(DWS) cluster.

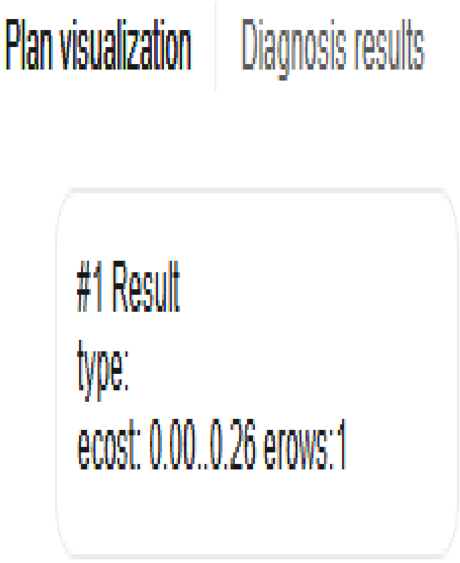
- **Historical query details**: displays the details of all historical queries. You can search by criteria or sort all fields to quickly find desired information. Click the setting button in the upper right corner of the list to display or hide columns.
  - **View details**: You can click **View details** to check the details about each historical top SQL statement. Click **View Details** in the **Operation** column of the row that contains the target top SQL statement to view its details, including the basic information, real-time resource consumption during execution, complete SQL statement, and query plan.
  - **One-click diagnosis**: Click **One-click diagnosis** in the **Operation** column of the row that contains the target top SQL statement to view its static check and execution plan diagnosis result and visualize the result.
    - **Planned diagnostics**: parses the execution plan character string based on the execution plan format and diagnoses the execution plan in the historical top SQL tables. A visualized tree chart is used to display information about each node, such as the execution duration, type, and number of scanned rows.

      The exception types that can be identified include redistribution exceptions, estimation exceptions, computing skews, partition scanning exceptions, and cross-logical cluster queries.

      NOTE

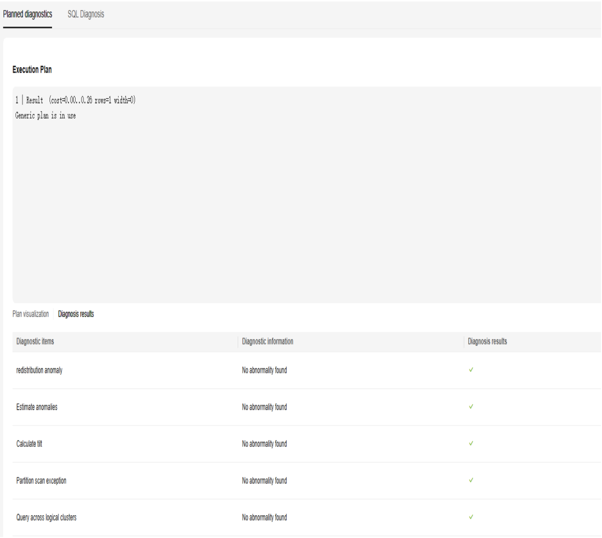
      To check the execution duration of each step in the execution plan, set **resource_track_level** to **perf**.

Figure 9-12 Plan visualization



#1 Result
type:
ecost: 0.00..0.26 erows:1

Figure 9-13 Plan diagnosis



Execution Plan

```
1 | Result  (cost=0.00..0.26 rows=1 width=0)
Generic plan is in use
```

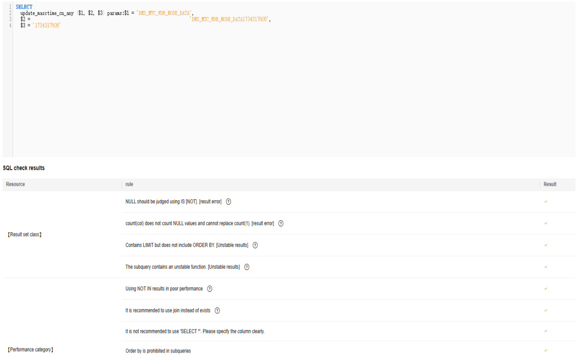
Diagnostic items	Diagnostic information	Diagnosis results
redistribution anomaly	No abnormality found	✓
Estimate anomalies	No abnormality found	✓
Calculate tilt	No abnormality found	✓
Partition scan exception	No abnormality found	✓
Query across logical clusters	No abnormality found	✓

- **SQL Diagnosis**: performs a static check on user-written SQL statements based on SQL development specifications. It analyzes non-compliant SQL statements and provides rectification suggestions.

  NOTE

  The SQL development specifications are created based on the usage of GaussDB(DWS) and are only meant for reference. The provided specifications may need to be revised according to actual usage. For details, see **GaussDB(DWS) Development Design Specifications**.

Figure 9-14 SQL diagnosis



```
SELECT
  update_maxctime_cn_any ($1, $2, $3) params:$1 = 'DMS_MTC_WDR_NODE_DATA',
  $2 =
                                                'DMS_MTC_WDR_NODE_DATA1734317608',
  $3 = '1734317608'
```

SQL check results

Resource	rule	Result
【Result set class】	NULL should be judged using IS [NOT]. [result error] ⓘ	✓
	count(col) does not count NULL values and cannot replace count(1). [result error] ⓘ	✓
	Contains LIMIT but does not include ORDER BY. [Unstable results] ⓘ	✓
	The subquery contains an unstable function. [Unstable results] ⓘ	✓
【Performance category】	Using NOT IN results in poor performance ⓘ	✓
	It is recommended to use join instead of exists ⓘ	✓
	It is not recommended to use 'SELECT *'. Please specify the column clearly.	✓
	Order by is prohibited in subqueries	✓

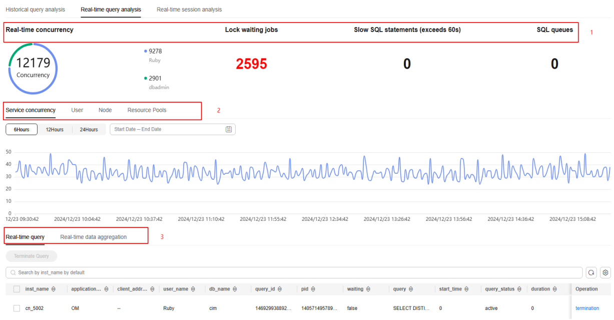
- **Data summary**: aggregates historical top SQL data by various dimensions (database, user, resource pool, application name, instance, and unique SQL ID) and displays the number of SQL statements and their resource consumption.

  Click **View Details** to view the resource consumption details of a particular object.

Real-time Query Analysis

This page provides information on real-time concurrency and user distribution, lock waiting jobs, slow SQL statements, SQL queues, service concurrency trend chart, real-time queries, and real-time data aggregation.

Figure 9-15 Real-time query analysis



□	inst_name	application...	client_addr...	user_name	db_name	query_id	pid	waiting	query	start_time	query_status	duration	Operation
□	cn_5002	OM	--	Ruby	cim	146929938892...	140571495789...	false	SELECT DISTI...	0	active	0	termination

- **Real-time Concurrency**: displays the number of SQL statements currently running in the cluster and the concurrency distribution for each user.
- **Lock waiting jobs**: indicates the number of SQL statements waiting for locks in the cluster.
- **Slow SQL statements**: displays the number of SQL statements whose duration exceeds 60s.
- **SQL queues**: displays the total number of all queuing SQL statements in the clusters, resource pools, and CCNs.
- **Service concurrency**: refers to the number of services running concurrently in a cluster during a specific time period. This information is displayed in a line chart, making it easier to collect and compare statistics. You can click **User**, **Node**, or **Resource Pools** to view this information in different dimensions.
- **Real-time query**: allows you to view all the details of running queries in the cluster. You can sort and filter displayed items to help you quickly locate the

information you need. Click the setting button in the upper right corner of the list to display or hide columns.

- Click **termination** in the **Operation** column to terminate a real-time top SQL query.
- Select multiple real-time top SQL queries and click **Terminate Query** above the list to terminate them.
- Click **Execute Plan** in the **Operation** column to view a visualized tree chart of real-time top SQL execution plans, including the execution duration, type, and number of scanned rows of each node.

NOTE

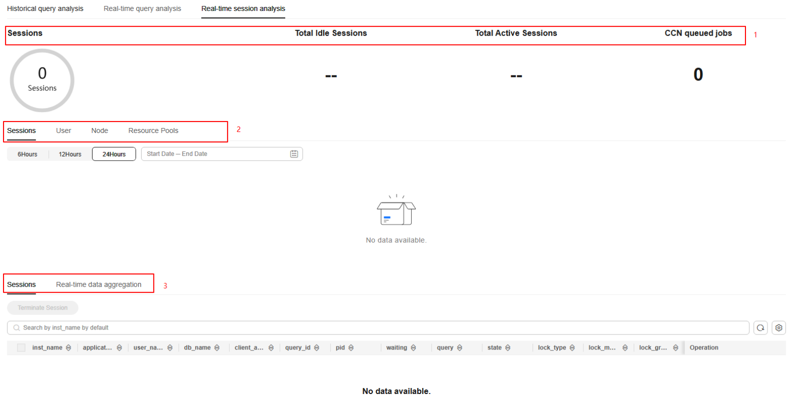
The fine-grained permission control function is added. Only users with the operate permission are able to terminate queries. For users with the read-only permission, the **Terminate Query** button is grayed out.

- **Real-time data aggregation**: summarizes the top SQL query data in real-time, categorizing it by different dimensions such as node, query ID, user, and resource pool. It provides information on the number of running SQL statements, queuing SQL statements, slow SQL statements, complex statements, lock waiting jobs, and simple statements.

Real-Time Session Analysis

This page displays the number of real-time sessions and their distribution among users, the number of idle and active sessions, the number of jobs in the CCN queue, session quantity trend chart, real-time session list, and real-time data aggregation.

Figure 9-16 Real-time session analysis



- **Sessions**: displays the total number of real-time sessions in the cluster and their distribution among users.
- **Total Idle Sessions**: refers to the total number of idle sessions in the cluster.
- **Total Active Sessions**: refers to the total number of active sessions in the cluster.
- **CCN queued jobs**: displays the total number of jobs in the CCN queue.
- **Sessions**: refers to the number of sessions in a cluster during a specific time period. This information is displayed in a line chart, making it easier to collect

and compare statistics. You can click **User**, **Node**, or **Resource Pools** to view this information in different dimensions.

- **Sessions**: allows you to view the real-time information about all running sessions. You can sort and filter displayed items to help you quickly locate the information you need. Click the setting button in the upper right corner of the list to display or hide columns.
  - Click **termination** in the **Operation** column to terminate a real-time session.
  - Select multiple sessions and click **Terminate Session** above the list to terminate them.

  NOTE

  The fine-grained permission control function is added. Only users with the operate permission are able to terminate sessions. For users with the read-only permission, the **Terminate a Session** button is grayed out.
- **Real-time data aggregation**: summarizes session data by various dimensions (node, user, and resource pool) and displays metrics such as the number of active sessions, the number of idle sessions, CPU usage time (s), average memory usage (MB), estimated memory usage (MB), and spilled data volume (MB).

9.4 Viewing GaussDB(DWS) Cluster Alarms

9.4.1 Alarm Management

Overview

Alarm management includes viewing and configuring alarm rules and subscribing to alarm information. Alarm rules display alarm statistics and details of the past week for users to view tenant alarms. In addition to providing a set of default GaussDB(DWS) alarm rules, this feature allows you to modify alarm thresholds based on your own services. GaussDB(DWS) alarm notifications are sent using the SMN service.

NOTE

- This feature is supported only in cluster version 8.1.1.200 and later.
- Currently, alarms cannot be categorized and managed by enterprise project.

Visiting the Alarms Page

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Monitoring** > **Alarm**.

Step 3 On the page that is displayed:

- **Existing Alarm Statistics**

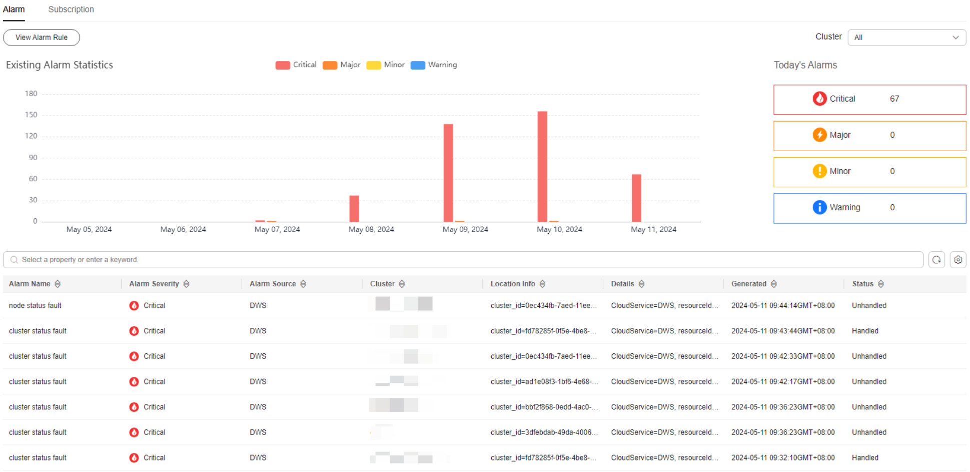
  Statistics of the existing alarms in the past seven days are displayed by alarm severity in a bar chart. In this way, you can see clearly the number and category of the alarms generated in the past week.

- **Today's Alarms**

  Statistics of the existing alarms on the current day are displayed by alarm severity in a list. In this way, you can see clearly the number and category of the unhandled alarms generated on the day.

- Alarm details

  Details about all alarms, handled and unhandled, in the past seven days are displayed in a table for you to quickly locate faults, including the alarm name, alarm severity, alarm source, cluster name, location, description, generation date, and status.



Today's Alarms	
Critical	67
Major	0
Minor	0
Warning	0

Alarm Name	Alarm Severity	Alarm Source	Cluster	Location Info	Details	Generated	Status
node status fault	Critical	DWS	[redacted]	cluster_id=0ec434fb-7aed-11ee...	CloudService=DWS, resourceId...	2024-05-11 09:44:14GMT+08:00	Unhandled
cluster status fault	Critical	DWS	[redacted]	cluster_id=fd78285f-0f5e-4be8-...	CloudService=DWS, resourceId...	2024-05-11 09:43:44GMT+08:00	Handled
cluster status fault	Critical	DWS	[redacted]	cluster_id=0ec434fb-7aed-11ee...	CloudService=DWS, resourceId...	2024-05-11 09:42:33GMT+08:00	Unhandled
cluster status fault	Critical	DWS	[redacted]	cluster_id=ad1e08f3-1bf6-4e68-...	CloudService=DWS, resourceId...	2024-05-11 09:42:17GMT+08:00	Unhandled
cluster status fault	Critical	DWS	[redacted]	cluster_id=bbf2f868-0edd-4ac0-...	CloudService=DWS, resourceId...	2024-05-11 09:36:23GMT+08:00	Unhandled
cluster status fault	Critical	DWS	[redacted]	cluster_id=3dfebdab-49da-4006...	CloudService=DWS, resourceId...	2024-05-11 09:36:23GMT+08:00	Unhandled
cluster status fault	Critical	DWS	[redacted]	cluster_id=fd78285f-0f5e-4be8-...	CloudService=DWS, resourceId...	2024-05-11 09:32:10GMT+08:00	Handled

NOTE

The alarm data displayed (a maximum of 30 days) is supported by the Event Service microservice.

----End

Alarm Types and Alarms

NOTE

The alarm policy is triggered based on the current configuration.

Table 9-10 Threshold alarms of DMS alarm sources

Type	Name	Severity	Description
Default	Node CPU Usage Exceeds the Threshold	Urgent	This alarm is generated if the threshold of CPU usage (system + user) of any node in the cluster is exceeded within the specified period and the constraint is not met. The alarm will be cleared when the CPU usage (system + user) is lower than the threshold and the constraint is not met.

Type	Name	Severity	Description
Default	Node Data Disk Usage Exceeds the Threshold	Urgent: > 85%; Important: > 80%	This alarm is generated if the threshold of data disk (**/var/chroot/DWS/data***[n]*) usage of any node in the cluster is exceeded within the specified period and the constraint is not met. The alarm will be cleared when the data disk (**/var/chroot/DWS/data***[n]*) usage is lower than the threshold and the constraint is not met.
Default	Node Data Disk I/O Usage Exceeds the Threshold	Urgent	This alarm is generated if the threshold of data disk (**/var/chroot/DWS/data***[n]*) I/O usage (**util**) of any node in the cluster is exceeded within the specified period and the constraint is not met. The alarm will be cleared when the data disk (**/var/chroot/DWS/data***[n]*) I/O usage (**util**) is lower than the threshold and the constraint is not met.
Default	Node Data Disk Latency Exceeds the Threshold	Important	This alarm is generated if the threshold of data disk (**/var/chroot/DWS/data***[n]*) I/O latency (**await**) of any node in the cluster is exceeded within the specified period and the constraint is not met. The alarm will be cleared when the data disk (**/var/chroot/DWS/data***[n]*) I/O latency (**await**) is lower than the threshold and the constraint is not met.
Default	Data Spilled to Disks of the Query Statement Exceeds the Threshold	Urgent	This alarm is generated if the threshold of data flushed to disks of the SQL statement in the cluster is exceeded within the specified period and the constraint is not met. The alarm can be cleared only after you handle the SQL statement.
Default	Number of Queuing Query Statements Exceeds the Threshold	Urgent	This alarm is generated if the threshold of the number of queuing SQL statements is exceeded within the specified period. The alarm will be cleared when the number of queuing SQL statements is less than the threshold.

Type	Name	Severity	Description
Default	Queue Congestion in the Default Cluster Resource Pool	Urgent	This alarm is generated if the queue in the default resource pool of a cluster is congested and no alarm suppression conditions are met. This alarm will be cleared if the queue is not congested.
Default	Long SQL Probe Execution Duration in a Cluster	Urgent	This alarm is generated if the DMS alarm module detects a SQL probe execution duration on a server and no alarm suppression conditions are met. If no execution duration exceeds the threshold, the alarm will be automatically cleared. **NOTE** The alarm is supported only in 8.1.1.300 and later cluster versions. For earlier versions, contact technical support.
Default	A Vacuum Full Operation That Holds a Table Lock for A Long Time Exists in the Cluster	Important	In a specified period, the DMS alarm module detects that VACUUM FULL has been running for a long time in the cluster and blocks other operations. This alarm is generated if there are other SQL statements in the lock wait state and no suppression conditions are met. This alarm will be cleared if VACUUM FULL in the cluster did not cause lock wait. **NOTE** If this alarm is generated, contact technical support engineers.
Default	Instance Memory Usage of a Cluster Node Exceeds the Threshold	Urgent	This alarm is generated if the DMS alarm module detects the instance memory usage on a node in a cluster exceeds the threshold and no alarm suppression conditions are met. If the usage decreases, the alarm will be automatically cleared. **NOTE** If this alarm is generated, contact technical support engineers.

Type	Name	Severity	Description
Default	Dynamic Memory Usage of a Cluster Node Exceeds the Threshold	Urgent	This alarm is generated if the DMS alarm module detects the dynamic memory usage on a node in a cluster exceeds the threshold and no alarm suppression conditions are met. If the usage decreases, the alarm will be automatically cleared. **NOTE** If this alarm is generated, contact technical support engineers.
Default	Disk Usage of a GaussDB(DWS) Cluster Resource Pool Exceeds the Threshold	Urgent	The DMS alarm module generates an alarm if the disk usage of the cluster resource pool exceeds the set threshold within a specific time frame and the suppression conditions are not met. The alarm is cleared when the DMS alarm module detects that the disk usage of the cluster resource pool is below the threshold. **NOTE** If this alarm is generated, contact technical support engineers.
Default	Session Usage in a GaussDB(DWS) Cluster Exceeds the Threshold	Urgent	The DMS alarm module generates an alarm if the session usage in the cluster exceeds the set threshold within a specific time frame and the suppression conditions are not met. The alarm is cleared when the DMS alarm module detects that the session usage in the cluster is below the threshold. **NOTE** If this alarm is generated, contact technical support engineers.
Default	Active Session Usage in a GaussDB(DWS) Cluster Exceeds the Threshold	Urgent	The DMS alarm module generates an alarm if the active session usage in the cluster exceeds the set threshold within a specific time frame and the suppression conditions are not met. The alarm is cleared when the DMS alarm module detects that the active session usage in the cluster is below the threshold. **NOTE** If this alarm is generated, contact technical support engineers.

Type	Name	Severity	Description
Default	Number of Database Deadlocks in a GaussDB(DWS) Cluster Exceeds the Threshold	Urgent	If the number of deadlocks in the cluster database exceeds the threshold within a specific time frame and the suppression conditions are not met, the DMS alarm module will generate an alarm. The alarm will be cleared once the DMS alarm module detects that the number of deadlocks in the cluster database is below the threshold. **NOTE** If this alarm is generated, contact technical support engineers.
Default	Database Session Usage of the GaussDB(DWS) Cluster Exceeds the Threshold	Urgent	The DMS alarm module will generate an alarm if the session usage of the cluster database goes over the threshold within a specific time frame and the suppression conditions are not met. The alarm will be resolved by the DMS alarm module once it detects that the session usage of the cluster database is below the threshold. **NOTE** If this alarm is generated, contact technical support engineers.
Custom	*Name of the user-defined threshold alarm*	*User-defined alarm severity*	*Alarm description*

9.4.2 Alarm Rules

Overview

- Concepts related to threshold alarms
  - Alarm rule: consists of the alarm rule name, rule description, clusters associated with the rule, alarm policy triggering relationship, and alarm policy. An alarm rule can apply to one or all clusters, and can consist of one or more policies. The relationship between alarm policies can be selected in **Triggered Policies**. Each alarm policy consists of the triggers and constraint of each alarm rule.
  - Alarm policy: consists of the triggers, constraint, and alarm severity for an alarm metric.
  - Alarm metric: indicates a database cluster metric, which is generally time series data, for example, node CPU usage and amount of data flushed to disks.

- Alarm rule types
  - Default rule: best practices of GaussDB(DWS) threshold alarms.
  - User-defined rule: personalized alarm rules by configuring or combining monitoring metrics. (The current version supports only user-defined alarm rules of schema usage.)
- Alarm rule operations
  - Modify: modifies an alarm rule. All alarm rules apply (all items of user-defined alarm rules but only some items of the default alarm rules).
  - Enable/Disable: enables or disables an alarm rule. All alarm rules apply. When an alarm rule is enabled, it is added to the check list of the alarm engine and can be triggered normally. Disabled rules are not in the check list.
  - Delete: deletes an alarm rule. You can delete only user-defined rules. Default alarm rules cannot be deleted.

Precautions

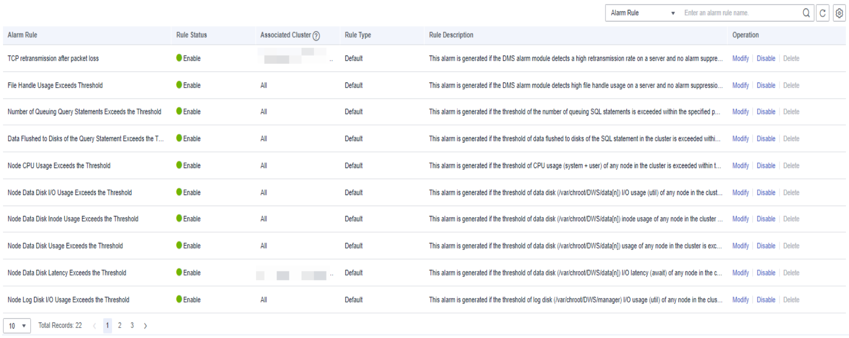
After a cluster is migrated, to monitor alarms of the new cluster, change the cluster bound to the alarm rule to the new cluster. You can also create an alarm rule for the new cluster.

Viewing Alarm Rules

Step 1 Log in to the GaussDB(DWS) console.

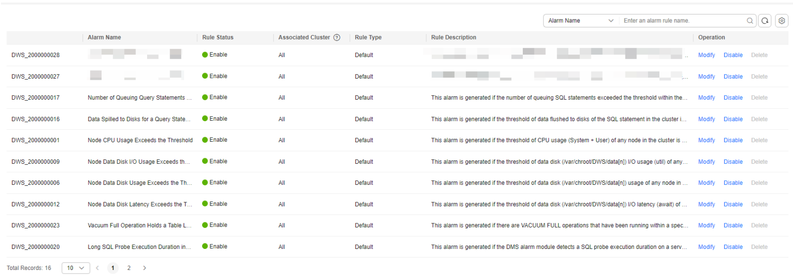
Step 2 In the navigation tree on the left, choose **Monitoring** > **Alarm**.

Step 3 Click **View Alarm Rule** in the upper left corner. On the page that is displayed, you can see the threshold alarm rules of database cluster monitoring metrics, as shown in the following figure.



Alarm Rule	Rule Status	Associated Cluster	Rule Type	Rule Description	Operation
TCP retransmission after packet loss	● Enable	[redacted]	Default	This alarm is generated if the DMS alarm module detects a high retransmission rate on a server and no alarm suppre...	Modify | Disable | Delete
File Handle Usage Exceeds Threshold	● Enable	All	Default	This alarm is generated if the DMS alarm module detects high file handle usage on a server and no alarm suppressio...	Modify | Disable | Delete
Number of Queuing Query Statements Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of the number of queuing SQL statements is exceeded within the specified p...	Modify | Disable | Delete
Data Flushed to Disks of the Query Statement Exceeds the T...	● Enable	All	Default	This alarm is generated if the threshold of data flushed to disks of the SQL statement in the cluster is exceeded withi...	Modify | Disable | Delete
Node CPU Usage Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of CPU usage (system + user) of any node in the cluster is exceeded within t...	Modify | Disable | Delete
Node Data Disk I/O Usage Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) I/O usage (util) of any node in the clust...	Modify | Disable | Delete
Node Data Disk Inode Usage Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) inode usage of any node in the cluster ...	Modify | Disable | Delete
Node Data Disk Usage Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) usage of any node in the cluster is exc...	Modify | Disable | Delete
Node Data Disk Latency Exceeds the Threshold	● Enable	[redacted]	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) I/O latency (await) of any node in the c...	Modify | Disable | Delete
Node Log Disk I/O Usage Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of log disk (/var/chroot/DWS/manager) I/O usage (util) of any node in the clus...	Modify | Disable | Delete

10 ▾ Total Records: 22 ‹ 1 2 3 ›



	Alarm Name	Rule Status	Associated Cluster	Rule Type	Rule Description	Operation
DWS_2000000028	[redacted]	● Enable	All	Default	[redacted]	Modify Disable Delete
DWS_2000000027	[redacted]	● Enable	All	Default	[redacted]	Modify Disable Delete
DWS_2000000017	Number of Queuing Query Statements ...	● Enable	All	Default	This alarm is generated if the number of queuing SQL statements exceeded the threshold within the...	Modify Disable Delete
DWS_2000000016	Data Spilled to Disks for a Query State...	● Enable	All	Default	This alarm is generated if the threshold of data flushed to disks of the SQL statement in the cluster i...	Modify Disable Delete
DWS_2000000001	Node CPU Usage Exceeds the Threshold	● Enable	All	Default	This alarm is generated if the threshold of CPU usage (System + User) of any node in the cluster is ...	Modify Disable Delete
DWS_2000000009	Node Data Disk I/O Usage Exceeds th...	● Enable	All	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) I/O usage (util) of any...	Modify Disable Delete
DWS_2000000006	Node Data Disk Usage Exceeds the Th...	● Enable	All	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) usage of any node in ...	Modify Disable Delete
DWS_2000000012	Node Data Disk Latency Exceeds the T...	● Enable	All	Default	This alarm is generated if the threshold of data disk (/var/chroot/DWS/data[n]) I/O latency (await) of ...	Modify Disable Delete
DWS_2000000023	Vacuum Full Operation Holds a Table L...	● Enable	All	Default	This alarm is generated if there are VACUUM FULL operations that have been running within a spec...	Modify Disable Delete
DWS_2000000020	Long SQL Probe Execution Duration in...	● Enable	All	Default	This alarm is generated if the DMS alarm module detects a SQL probe execution duration on a serv...	Modify Disable Delete

Total Records: 16 10 ⌄ ‹ 1 2 ›

----End

Modifying an Alarm Rule

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Monitoring** > **Alarm**.

Step 3 Click **View Alarm Rule** in the upper left corner.

Step 4 On the **Alarm Rules** page that is displayed, click **Modify** in the **Operation** column of the target alarm rule.

NOTE

- Read-only users (with the DWS ReadOnlyAccess permission) cannot modify alarm rules.
- You can modify only some items of the default rules (associated cluster, alarm policy threshold, time period, and alarm constraint). User-defined rules support modification of all items.

Table 9-11 Alarm rule parameters

Parameter	Description	Example Value
Alarm Rule	The rule name contains 6 to 64 characters and must start with a non-digit character.	-
Description	User-defined description, which contains a maximum of 490 characters.	-
Associated Cluster	You can select a cluster of the current tenant from the drop-down list box as the monitoring cluster of the alarm module.	All
Triggered Policies	Policy triggering relationships are as follows: ● **Independent**: Alarm policies are triggered independently of each other. ● **Priority**: Alarm policies are triggered by priority. Policies of a lower priority will be automatically triggered after those of a higher priority.	Independent

Parameter	Description	Example Value
Alarm Policy	The alarm policies are as follows: ● **Metric**: GaussDB(DWS) monitoring metric, which is the data source used by the alarm engine for threshold determination. ● **Alarm Object**: databases in the selected cluster and schemas in the selected databases. ● **Trigger**: calculation rule for threshold determination of a monitoring metric. Select the average value within a period of time of a metric to reduce the probability of alarm oscillation. ● **Constraint**: suppresses the repeated triggering and clearance of alarms of the same type within the specified period. ● **Alarm Severity**: includes **Urgent**, **Important**, **Minor**, and **Prompt**.	-

Step 5 Confirm the information and click **OK**.

----End

Creating an Alarm Rule

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation tree on the left, choose **Monitoring** > **Alarm**.

Step 3 Click **View Alarm Rule** in the upper left corner.

Step 4 Click **Create Alarm Rule** in the upper right corner. You can configure items, such as the alarm rule name, rule description, associated cluster, and alarm policy. For details, see **Table 9-11**.

NOTE

Currently, only alarm rules of schema usage metrics can be created on GaussDB(DWS).

----End

9.4.3 Alarm Subscriptions

You can subscribe to GaussDB(DWS) alarm notifications to receive notifications by SMS message, email, or application when an alarm of a specified severity is generated.

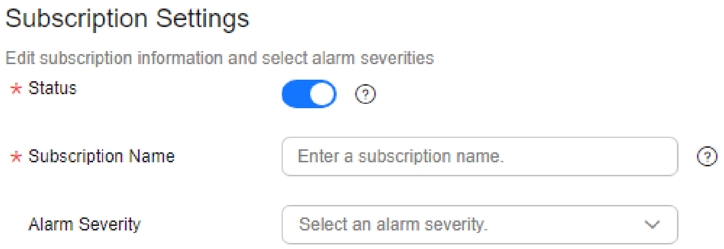
Creating a Subscription

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 In the navigation pane on the left, choose **Management** > **Alarm** and click **Subscriptions**.

Step 3 Click **Create Subscription** in the upper left corner of the page.

Step 4 In the **Subscription Settings** area, configure the basic information and alarm severity of the subscription.



Subscription Settings

Edit subscription information and select alarm severities

* Status

* Subscription Name: Enter a subscription name.

Alarm Severity: Select an alarm severity.

Table 9-12 Subscription parameters

Parameter	Description
Status	Whether to enable the alarm subscription. When you disable a subscription, you will not receive the corresponding alarm notifications, but the subscription will not be deleted.
Subscription Name	Name of the alarm subscription: ● Contains only letters, digits, hyphens (-), and underscores (_), and must start with a letter or digit. ● Contains 1 to 256 characters.
Cluster	Select the cluster to subscribe to. Note that only one cluster can be subscribed to multiple alarms.

Step 5 The **Subscribed Alarms** area displays the subscribed alarms by subscription settings. Select an SMN topic from the drop-down list.

To create a topic, click **Create Topic**. The SMN console is displayed.

NOTE

The selected topic must have granted GaussDB(DWS) the permission for publishing messages to the topic. To grant permissions, configure topic policies on the SMN management console. When configuring the topic policy, select **DWS** for services that can publish messages to this topic.

Step 6 Confirm the information and click **OK**.

----End

Modifying a Subscription

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Management** > **Alarm** and click **Subscriptions**.

Step 3 In the **Operation** column of the target subscription, click **Edit**.

Step 4 On the **Edit Subscription** page displayed, modify the parameters. For details, see **Step 4** to **5**.

Step 5 Click **OK**.

----End

Deleting a Subscription

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Management** > **Alarm** and click **Subscriptions**.

Step 3 In the **Operation** column of the target subscription, click **Delete**. A confirmation dialog box is displayed.

Step 4 Click **Yes** to delete the subscription.

----End

9.4.4 Alarm Handling

9.4.4.1 DWS_2000000001 Node CPU Usage Exceeds the Threshold

Description

GaussDB(DWS) collects the CPU usage of each node in a cluster every 30 seconds. If the average CPU usage of a node in the last 10 minutes (configurable) exceeds 90% (configurable), an alarm is reported indicating that the node CPU usage exceeds the threshold. If the average usage is lower than 85% (that is, the reporting threshold minus 5%), the alarm is cleared.

NOTE

If the average CPU usage of a node is always greater than the alarm threshold, the alarm is generated again 24 hours (configurable).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000001	Management plane alarm	Urgent: > 90%	Operation alarm	GaussDB(DWS)	Yes

Parameters

Category	Name	Description
Location information	Name	Node CPU Usage Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

If the CPU usage is high for a long time, service processes may respond slowly or become unavailable.

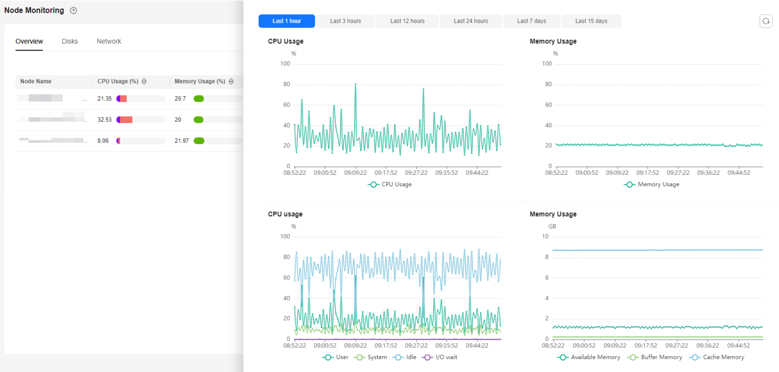
Possible Causes

- Complex services occupy a large number of CPU resources.
- The CPU configuration of the cluster is too low to meet service requirements.

Handling Procedure

Step 1 Check the CPU usage of each node.

1. Log in to the GaussDB(DWS) console.
2. Choose **Monitoring** > **Alarm**, select the cluster for which the alarm is generated in the cluster selection drop-down list in the upper right corner, view the alarm information of the cluster in the last seven days, and locate the name of the node for which the alarm is generated based on the location information.
3. On the **Clusters** > **Dedicated Clusters** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.
4. Choose **Monitoring** > **Node Monitoring** > **Overview** to view the CPU usage of each node in the current cluster. Click on the right to view the CPU performance metrics in the last 1, 3, 12, or 24 hours and see whether there is a sharp increase in the CPU usage.



Node Monitoring

Node Name	CPU Usage (%)	Memory Usage (%)
[redacted]	21.35	20.7
[redacted]	32.53	20
[redacted]	8.06	21.97

- If the CPU usage frequently increases and then returns to normal in a short period of time, it indicates that the CPU usage temporarily spikes during service execution. In this case, you can adjust the alarm threshold through **Step 2** to reduce the number of reported alarms.
- If the CPU usage remains high for a long time, it indicates that the cluster is overloaded. In this case, check cluster services by referring to **Step 3** or enhance the cluster flavor. For details, see **Changing the Node Flavor**.

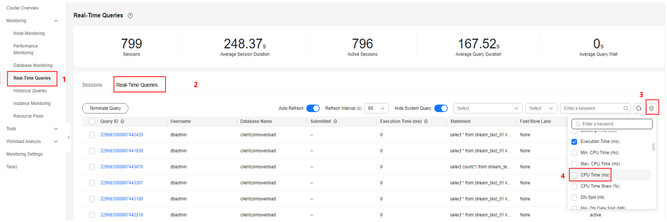
Step 2 **Check whether the CPU usage alarm configuration is proper.**

1. Choose **Monitoring** > **Alarm** and click **View Alarm Rule**.
2. Locate the row that contains the **Node CPU Usage Exceeds the Threshold**, and click **Modify** in the **Operation** column. The **Modifying an Alarm Rule page** is displayed.
3. Adjust the alarm threshold and detection period. A higher alarm threshold and a longer detection period indicate a lower alarm sensitivity. For details about the GUI configuration, see **Alarm Rules**.

Step 3 **Check whether the CPU usage of the current cluster service is too high.**

1. On the monitoring page, choose **Monitoring** > **Queries**, click ⚙, and select **CPU Time (ms)** to view the query with the longest CPU time.

Figure 9-17 Viewing CPU time information

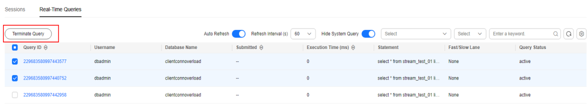


799	248.37s	796	167.52s	0s
Sessions	Average Session Duration	Active Sessions	Average Query Duration	Average Query Wait

Query ID	Username	Database Name	Submitted	Execution Time (ms)	Statement	Fast/Slow Lane	
229683580997442429	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	
229683580997441936	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	
229683580997443670	dbadmin	clientconnoverload	--	0	select count(*) from stream_te...	None	
229683580997443307	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	
229683580997443189	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	
229683580997442216	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	active

2. After confirming with the service side, select the query ID to be stopped and click **Stop Query**.

Figure 9-18 Terminating a query



	Query ID	Username	Database Name	Submitted	Execution Time (ms)	Statement	Fast/Slow Lane	Query Status
☑	229683580997443577	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	active
☑	229683580997440752	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	active
□	229683580997442958	dbadmin	clientconnoverload	--	0	select * from stream_test_01 li...	None	active

Step 4 See advanced tuning operations in **"Troubleshooting" > "Cluster Performance"**.

----End

Alarm Clearance

After the CPU usage decreases, the alarm is automatically cleared.

9.4.4.2 DWS_2000000006 Node Data Disk Usage Exceeds the Threshold

Description

GaussDB(DWS) collects the usage of all disks on each node in a cluster every 30 seconds.

- If the maximum disk usage in the last 10 minutes (configurable) exceeds 80% (configurable), a major alarm is reported. If the average disk usage is lower than 75% (that is, the alarm threshold minus 5%), this major alarm is cleared.
- If the maximum disk usage in the last 10 minutes (configurable) exceeds 85% (configurable), a critical alarm is reported. If the average disk usage is lower than 85% (that is, the alarm threshold minus 5%), this critical alarm is cleared.

NOTE

If the maximum disk usage is always greater than the alarm threshold, the system generates an alarm again 24 hours later (configurable).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000006	Management plane alarm	Urgent: > 85%; important: > 80%	Operation alarm	GaussDB(DWS)	Yes

Parameters

Category	Name	Description
Location information	Name	Node Data Disk Usage Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

If the cluster data volume or temporary data spill size increases and the usage of any single disk exceeds 90%, the cluster becomes read-only, affecting customer services.

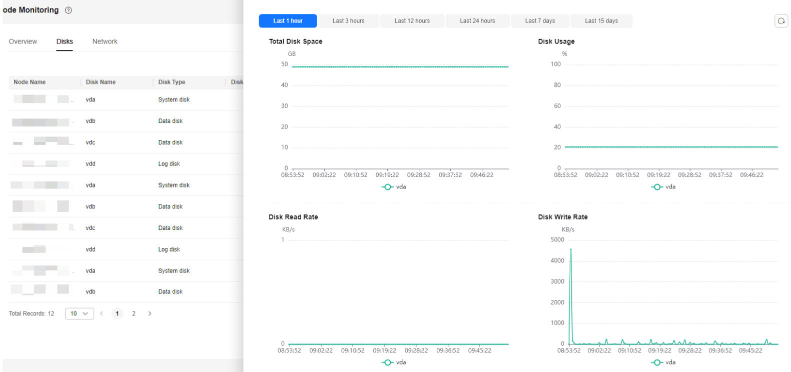
Possible Causes

- The service data volume increases rapidly, and the cluster disk capacity configuration cannot meet service requirements.
- Dirty data is not cleared in a timely manner.
- There are skew tables.

Handling Procedure

Step 1 **Check the disk usage of each node.**

1. Log in to the GaussDB(DWS) console.
2. Choose **Monitoring** > **Alarm**, select the current cluster from the cluster selection drop-down list in the upper right corner and view the alarm information of the cluster in the last seven days. Locate the name of the node for which the alarm is generated and the disk information based on the location information.
3. On the **Cluster** > **Dedicated Clusters** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.
4. Choose **Monitoring** > **Node Monitoring** > **Disks** to view the usage of each disk on the current cluster node. If you want to view the historical monitoring information about a disk on a node, click on the right to view the disk performance metrics in the last 1, 3, 12, or 24 hours.



Node Name	Disk Name	Disk Type	Disk
[redacted]	vda	System disk	
[redacted]	vdb	Data disk	
[redacted]	vdc	Data disk	
[redacted]	vdd	Log disk	
[redacted]	vda	System disk	
[redacted]	vdb	Data disk	
[redacted]	vdc	Data disk	
[redacted]	vdd	Log disk	
[redacted]	vda	System disk	
[redacted]	vdb	Data disk	

Total Records: 12 10 < 1 2 >

   - If the data disk usage frequently increases and then returns to normal in a short period of time, it indicates that the disk usage temporarily spikes due to service execution. In this case, you can adjust the alarm threshold through **Step 2** to reduce the number of reported alarms.
   - If the usage of a data disk exceeds 90%, read-only is triggered and error **cannot execute INSERT in a read-only transaction** is reported for write-related services. In this case, you can refer to **Step 3** to delete unnecessary data.
   - If the usage of more than half of the data disks in the cluster exceeds 70%, the data volume in the cluster is large. In this case, refer to **Step 4** to clear data or perform **Disk Capacity Expansion**.
   - If the difference between the highest and lowest data disk usage in the cluster exceeds 10%, refer to **Step 5** to handle data skew.

Step 2 **Check whether the alarm configuration is proper.**

1. Return to the GaussDB(DWS) console, choose **Monitoring** > **Alarm** and click **View Alarm Rule**.
2. Locate the row that contains **Node Data Disk Usage Exceeds the Threshold** and click **Modify** in the **Operation** column. On the **Modifying an Alarm Rule** page, view the configuration parameters of the current alarm.
3. Adjust the alarm threshold and detection period. A higher alarm threshold and a longer detection period indicate a lower alarm sensitivity. For details about the GUI configuration, see **Alarm Rules**.
4. If the data disk specification is high, you are advised to increase the threshold based on historical disk monitoring metrics. Otherwise, perform other steps. If the problem persists, you are advised to perform **Disk Capacity Expansion**.

Step 3 Check whether the cluster is in the read-only state.

1. When a cluster is in read-only state, stop the write tasks to prevent data loss caused by disk space exhaustion.
2. Return to the GaussDB(DWS) console and choose **Clusters** > **Dedicated Clusters**. In the row of the abnormal cluster whose cluster status is **Read-only**, click **Cancel Read-only**.
3. In the displayed dialog box, confirm the information and click **OK** to cancel the read-only state for the cluster. For details, see **Removing the Read-only Status**.
4. After the read-only mode is disabled, use the client to connect to the database and run the **DROP**/**TRUNCATE** command to delete unnecessary data.

   NOTE

   You are advised to lower the disk usage to below 70%. Check whether there are other tables that need to be rectified by referring to **Step 4** and **Step 5**.

Step 4 Check whether the usage of more than half of the data disks in the cluster exceeds 70%.

1. Run the **VACUUM FULL** command to clear data. For details, see **Solution to High Disk Usage and Cluster Read-Only**. Connect to the database, run the following SQL statement to query tables whose dirty page rate exceeds 30%, and sort the tables by size in descending order:

```
SELECT schemaname AS schema, relname AS table_name, n_live_tup AS analyze_count,
pg_size_pretty(pg_table_size(relid)) as table_size, dirty_page_rate
FROM PGXC_GET_STAT_ALL_TABLES
WHERE schemaName NOT IN ('pg_toast', 'pg_catalog', 'information_schema', 'cstore', 'pmk')
AND dirty_page_rate > 30
ORDER BY table_size DESC, dirty_page_rate DESC;
```

   The following is an example of the possible execution result of the SQL statement (the dirty page rate of a table is high):

```
 schema | table_name | analyze_count | table_size | dirty_page_rate
--------+------------+---------------+------------+-----------------
 public | test_table |          4333 | 656 KB     |           71.11
(1 row)
```

2. If any result is displayed in the command output, clear the tables with a high dirty page rate in serial mode.

```
VACUUM FULL ANALYZE schema.table_name
```

NOTICE

The **VACUUM FULL** operation occupies extra defragmentation space, which is Table size x (1 – Dirty page rate). As a result, the disk usage temporarily increases and then decreases. Ensure that the remaining space of the cluster is sufficient and will not trigger read-only when the **VACUUM FULL** operation is performed. You are advised to start from small tables. In addition, the **VACUUM FULL** operation holds an exclusive lock, during which access to the operated table is blocked. You need to properly arrange the execution time to avoid affecting services.

3. If no command output is displayed, no table with a high dirty page rate exists. You can expand the node or disk capacity of the cluster based on the following data warehouse types to prevent service interruption caused by read-only triggered by further disk usage increase.
   a. To scale out a storage-compute coupled data warehouse with cloud SSDs, see **Disk Capacity Expansion of an EVS Cluster**.
   b. To scale out a storage-compute coupled data warehouse with local SSDs or a standalone system, see **Scaling Out a Cluster**.

Step 5 Check whether the difference between the highest and lowest data disk usages in the cluster exceeds 10%.

1. If the data disk usage differs greatly, connect to the database and run the following SQL statement to check there are skew tables in the cluster:

```
SELECT schemaname, tablename, pg_size_pretty(totalsize), skewratio FROM pgxc_get_table_skewness
WHERE skewratio > 0.05 ORDER BY totalsize desc;
```

   The following is an example of the possible execution result of the SQL statement:

```
 schemaname |      tablename      | pg_size_pretty | skewratio
------------+---------------------+----------------+-----------
 scheduler  | workload_collection | 428 MB         |      .500
 public     | test_table          | 672 KB         |      .429
 public     | tbl_col             | 104 KB         |      .154
 scheduler  | scheduler_storage   | 32 KB          |      .250
(4 rows)
```

2. If the SQL statement output is displayed, select another distribution column for the table with severe skew based on the table size and skew rate. For 8.1.0 and later versions, use the **ALTER TABLE** syntax to adjust the distribution column. For other versions, see **How Do I Adjust Distribution Columns?**

----End

Alarm Clearance

After the disk usage decreases, the alarm is automatically cleared.

9.4.4.3 DWS_2000000009 Node Data Disk I/O Usage Exceeds the Threshold

Description

GaussDB(DWS) collects the data disk I/O usage of each cluster node every 30 seconds. This alarm is generated when the average usage of a data disk on a node exceeds 90% (configurable) in the last 10 minutes (configurable), and is automatically cleared when the average usage drops below 85% (alarm threshold minus 5%).

NOTE

- If the data disk I/O usage of a node is always greater than the alarm threshold, the alarm is generated again 24 hours later (configurable).
- When using an SSD storage-based cluster, disk I/O can surpass 100% as the service volume grows. But, this does not always mean there is a performance bottleneck. To confirm the alarm's validity, you should evaluate the service's actual running status.

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000009	Management plane alarm	Urgent: > 90%	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	Node Data Disk I/O Usage Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

- High disk I/O usage affects data read and write performance, thereby affecting cluster performance.
- A large number of disk writes occupy the disk capacity. If the disk capacity exceeds 90%, the cluster becomes read-only.

Possible Causes

- A large number of read or write operations are performed during peak hours.
- A large amount of data spills to disks due to the execution of complex statements.
- Data is scanned by the Scan operator.

Handling Procedure

Step 1 Choose **Dedicated Clusters** > **Clusters**, locate the row that contains the target cluster, and click **Monitoring** in the **Operation** column.

Step 2 In the navigation pane on the left, choose **Monitoring** > **Node Monitoring**. On the **Node Monitoring** page, click the **Disks** tab to view the data disk I/O usage and disk I/O rate.

If the disk I/O rate is high and the data disk usage keeps increasing, it indicates that services are writing data to disks. This may be caused by complex queries.

Step 3 Click **Queries** in the navigation pane to view the real-time queries.

If the execution time of a statement exceeds the expected time, stop the query and check the disk I/O usage again. For details, see 2.

----End

Alarm Clearance

This alarm is automatically cleared when the data disk I/O usage drops to a certain value.

9.4.4.4 DWS_2000000012 Node Data Disk Latency Exceeds the Threshold

Description

GaussDB(DWS) collects the data disk latency of each node in the cluster every 30 seconds. This alarm is generated when the average latency of a data disk on a node exceeds 400 ms (configurable) in the last 10 minutes (configurable), and is automatically cleared when the average latency drops below 400 ms.

NOTE

If the data disk latency of a node is always greater than the alarm threshold, this alarm is generated again after 24 hours (configurable).

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000012	Management plane alarm	Important: > 400 ms	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	Node Data Disk Latency Exceeds the Threshold

Category	Name	Description
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

High disk latency will slow down the data read/write speed, causing the cluster performance to deteriorate.

Possible Causes

The database is in peak hours and there are a large number of read and write requests.

Handling Procedure

Step 1 On the **Clusters** > **Dedicated Clusters** page, locate the row that contains the target cluster and click **Monitoring** in the **Operation** column.

Step 2 In the navigation pane on the left, choose **Monitoring** > **Node Monitoring**. On the **Node Monitoring** page, view the CPU usage, disk usage, and memory usage.

If the CPU usage and disk I/O rate are high, the cluster is in peak hours. You can adjust the latency threshold based on service requirements. For details, see **3**.

Step 3 Return to the console home page. In the navigation pane on the left, choose **Management** > **Alarms**. On the displayed page, click **View Alarm Rule** in the upper left corner.

Step 4 Locate the row that contains **Node Data Disk Latency Exceeds the Threshold**, and click **Modify** in the **Operation** column. On the displayed page, change the threshold.

----End

Alarm Clearance

This alarm is automatically cleared when the data disk latency drops to a certain value.

9.4.4.5 DWS_2000000016 Data Spilled to Disks of the Query Statement Exceeds the Threshold

Description

During the execution of service queries, the database may choose to store the temporary result to the disk, which is called **operator spilling**.

GaussDB(DWS) checks the load management records of jobs being executed on CNs through the **GS_WLM_SESSION_STATISTICS** view every 60 seconds and calculates the maximum amount of data spilled to DNs.

If the number of SQL statements spilled to disks exceeds 5 GB (configurable) within 10 minutes (configurable), an alarm is reported indicating that a query statement triggers the data spill threshold. This alarm is automatically cleared when the data spill drop below the alarm conditions. For details about how to modify alarm configurations, see **Modifying Alarm Rules**.

NOTE

If blocked SQL statements that can trigger the alarm persists, the alarm is generated again after 24 hours (configurable).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000016	Management plane alarm	Urgent: > 5 GB	Operation alarm	GaussDB(DWS)	Yes

Parameters

Category	Name	Description
Location information	Name	Data Spilled to Disks of the Query Statement Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

NOTE

You can connect to the database and run the **SELECT * FROM GS_WLM_SESSION_STATISTICS** command to view the **max_spill_size** column in the view.

Impact on the System

If a large amount of data spills to disks, a large number of system I/O resources are occupied. As a result, the data disk space may be insufficient or exhausted, triggering the database to become read-only and interrupting services.

Possible Causes

- The amount of service data spilled to disks exceeds the alarm threshold.
- The performance of the SQL query plan is poor, causing a large amount of data to be imported to the memory and spilled to disks.
- Expired data is not cleared in a timely manner. As a result, too much invalid data is scanned and spilled to disks.

Handling Procedure

Step 1 **Check whether the execution plan is poor in performance.**

1. Obtain the SQL statement from the additional information of the alarm, run the **ANALYZE** statement on the involved tables. Run the SQL statement again and check whether the amount of data spilled to disks decreases.
2. If there is no obvious effect, run the **EXPLAIN PERFORMANCE** command to view the actual execution information of the alarm SQL statement. For details, see **SQL Execution Plan**. If the estimated (**operator memory**) and peak memory usage (**Peak Memory**) are high, exceeding 20% of max_process_memory, optimization of the query is necessary based on execution information. For details, see **SQL Tuning Process**.

Step 2 **Check whether the alarm configuration is proper.**

1. Log in to the GaussDB(DWS) console, choose **Monitoring** > **Alarm** and click **View Alarm Rule**.
2. Click **Modify** in the **Operation** column of the row that contains **Data Flushed to Disks of the Query Statement Exceeds the Threshold**. The **Modifying an Alarm Rule** page is displayed.
3. If the cluster disk capacity is high, you can increase the alarm reporting threshold. It is recommended that the alarm reporting threshold be less than or equal to 5% of the capacity of a single data disk.

⚠ CAUTION

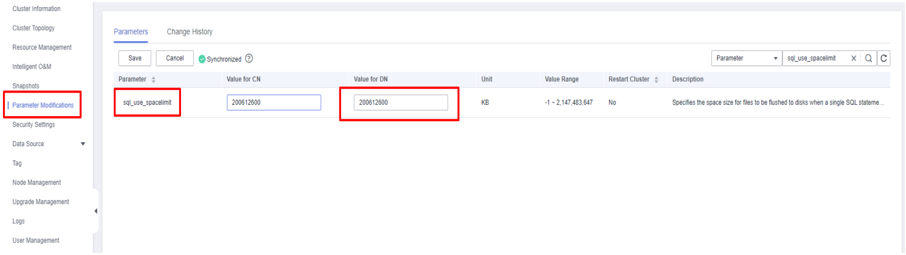
If the threshold is too large, data spilled to a disk may cause disk usage alarms or even the cluster to be read-only. If the data disk usage is close to or exceeds 80%, you are advised to clear unnecessary data when adjusting the threshold. For details about the GUI configuration, see **Alarm Rules**.

Step 3 **Kill the SQL statements that cause large data spills.**

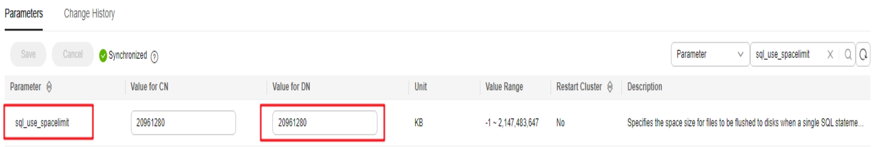
1. Return to the GaussDB(DWS) console.
2. On the **Clusters** > **Dedicated Clusters** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.
3. Choose **Monitoring** > **Queries**. Click ⚙ to see the data spill in the **Max. DN Data Spill (MB)** column.
4. After confirming with the service side, select the query ID of the query to be stopped and click **Stop Query**.
5. Adjust the database parameters for controlling the disk space of service statements. For details about the parameters, see **Statement Disk Space**

Control. For details about the procedure, see **Modifying Database Parameters**.

For example, the default value of **sql_use_spacelimit** is 10% of the total storage space of the DB instance. If the storage space is sufficient, you can increase the value. If the disk write volume of a single DN exceeds the value, GaussDB(DWS) stops the query and displays a message indicating that the disk write volume of a single DN exceeds the threshold.



Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
sql_use_spacelimit	200612600	200612600	KB	-1 ~ 2,147,483,647	No	Specifies the space size for files to be flushed to disks when a single SQL stateme...



Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
sql_use_spacelimit	20961280	20961280	KB	-1 ~ 2,147,483,647	No	Specifies the space size for files to be flushed to disks when a single SQL stateme...

----End

Alarm Clearance

This alarm is automatically cleared when data spill drops down to a low level.

9.4.4.6 DWS_2000000017 Number of Queuing Query Statements Exceeds the Threshold

Description

When real-time query monitoring is enabled, GaussDB(DWS) checks the queuing status of jobs on CNs through the **GS_WLM_SESSION_STATISTICS** view every 60 seconds by default.

This alarm is generated when the number of queuing SQL statements in the cluster exceeds 10 (configurable) within 10 minutes (configurable), and is automatically cleared when the number of queuing SQL statements drops below 10.

NOTE

If there continues to be queuing query statements more than the alarm threshold, the alarm is generated again 24 hours later (configurable).

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000017	Management plane alarm	Urgent: >10	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	Number of Queuing Query Statements Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

SQL queries are blocked. As a result, the execution time is too long.

Possible Causes

The number of queuing query statements during service execution exceeds the alarm threshold.

Handling Procedure

Check whether the current queuing jobs in the cluster are normal.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Monitoring** > **Alarm** in the navigation pane on the left, select the current cluster from the cluster selection drop-down list in the upper right corner and view the alarm information of the cluster in the last seven days. Locate the name of the cluster that triggers the alarm based on the location information.

Step 3 On the **Clusters** > **Dedicated Clusters** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.

Step 4 Choose **Monitoring** > **Queries** to view the real-time sessions and queries of the current cluster. Select the **Queries** tab to view the status of jobs being executed in the current cluster. Click ⚙ and select **Blocking Time (ms)** and **Waiting** status.

Click ↓☰ to sort the values of **Blocking Time (ms)**. You can view the information about the waiting SQL statements with the longest blocking time. If a query job is in the waiting state and the blocking time is abnormal, you can terminate the query.

Current queuing status of the statements, including:

- **Global**: global queuing.
- **Respool**: resource pool queuing.
- **CentralQueue**: queuing on the CCN
- **Transaction**: being in a transaction block
- **StoredProc**: being in a stored procedure
- **None**: not in a queue
- **Forced None**: being forcibly executed (transaction block statement or stored procedure statement are) because the statement waiting time exceeds the specified value

----End

Alarm Clearance

This alarm is automatically cleared when the number of queuing statements drops below the threshold.

9.4.4.7 DWS_2000000018 Queue Congestion in the Default Cluster Resource Pool

Description

GaussDB(DWS) uses resource pools to control memory, I/O, and CPU resources, manages and allocates resources based on task priorities, and manages user services loads. For details about resource pools, see **Resource Pool**. When resources are insufficient, some SQL statements have to queue up to wait for other statements to be executed. For details, see **CCN Queuing Under Dynamic Load Management**.

GaussDB(DWS) checks the queue in the default resource pool **default_pool** every 5 minutes. This alarm is generated when there are SQL statements that are queued up for a long time (20 minutes by default and configurable). This alarm is automatically cleared when the alarm threshold is no longer met.

NOTE

If blocked SQL statements that can trigger the alarm persists, the alarm is generated again after 24 hours (configurable).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000018	Management plane alarm	Critical	Operation alarm	GaussDB(DWS)	Yes

Parameters

Category	Name	Description
Location information	Name	Queue Congestion in the Default Cluster Resource Pool
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

When the default resource pool is blocked, all complex queries (estimated memory greater than or equal to 32 MB) associated with the default resource pool in the cluster may also be blocked. The queries in the queue are woken up only when the running queries are complete.

Possible Causes

- The estimated query memory usage is too large. As a result, the accumulated estimated memory usage exceeds the upper limit of the dynamic available memory, causing CCN queuing.
- Competition for public resources such as CPU and I/O deteriorates the performance of running queries.

Handling Procedure

Step 1 **Check whether the queue is caused by too large estimated memory.**

Rectify the fault by referring to **CCN Queuing Under Dynamic Load Management**.

Step 2 **Check whether the available memory of the cluster is normal.**

1. Log in to the GaussDB(DWS) console.
2. Choose **Monitoring** > **Alarm** in the navigation pane on the left, select the current cluster from the cluster selection drop-down list in the upper right corner and view the alarm information of the cluster in the last seven days.

Locate the name of the cluster that triggers the alarm based on the location information.

3. On the **Clusters** > **Dedicated Clusters** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.
4. Choose **Monitoring** > **Node Monitoring** > **Overview** to view the memory usage of each node in the current cluster. If you want to view the historical monitoring information about the memory usage of a node, click on the right to view the memory usage in the last 1, 3, 12, or 24 hours.

   If the cluster memory usage is low (for example, lower than 50%), the alarm may be generated because the estimated memory usage of queries is too large. In this case, perform the **Analyze** operation on related tables.

Step 3 **Check the competition of other resources.**

1. Check the CPU, I/O, and network usage of the cluster by referring to section **Step 2**.
2. If the database is fully loaded, query **Real-Time Top SQL** and kill the statements that occupy a large number of resources.

Step 4 **Check whether too many queries are submitted in a short period of time.**

1. Run the following SQL statement to query the task execution status:

```
SELECT
   s.resource_pool AS rpname, s.node_group,
   count(1) AS session_cnt,
   SUM(CASE WHEN a.enqueue = 'waiting in global queue' THEN 1 ELSE 0 END) AS global_wait,
   SUM(CASE WHEN s.lane= 'fast' AND a.state = 'active' AND (a.enqueue IS NULL OR a.enqueue = 'no
waiting queue') THEN 1 ELSE 0 END) AS fast_run,
   SUM(CASE WHEN s.lane= 'fast' AND a.enqueue = 'waiting in respool queue' THEN 1 ELSE 0 END)
AS fast_wait,
   SUM(CASE WHEN s.lane= 'slow' AND a.state = 'active' AND (a.enqueue IS NULL OR a.enqueue =
'no waiting queue') THEN 1 ELSE 0  END) AS slow_run,
   SUM(CASE WHEN s.lane= 'slow' AND (a.enqueue = 'waiting in ccn queue' OR a.enqueue = 'waiting
in respool queue') THEN 1 ELSE 0  END) AS slow_wait,
   SUM(CASE WHEN (a.enqueue IS NULL OR a.enqueue = 'no waiting queue') AND a.state = 'active'
THEN statement_mem ELSE 0 END) AS est_mem
FROM pgxc_session_wlmstat s,pgxc_stat_activity a
WHERE s.threadid=a.pid(+) AND s.attribute != 'Internal'
GROUP BY 1,2;
```

   The following is an example of the possible execution result of the SQL statement:

```
   rpname    |  node_group  | session_cnt | global_wait | fast_run | fast_wait | slow_run | slow_wait |
est_mem
--------------+--------------+-------------+-------------+----------+-----------+----------+-----------+---------
 default_pool | installation |           6 |           0 |        0 |         0 |        0 |         0 |       0
 root         | installation |           1 |           0 |        0 |         0 |        0 |         0 |       0
(2 rows)
```

- In the query result, if the value of **slow_wait** corresponding to **default_pool** is not 0, the cluster is fully loaded due to too many jobs. As a result, an alarm is generated. In this case, you can locate the row that contains the specified cluster on the console, choose **Monitoring Panel** in the **Operation** column. On the displayed page, choose **Monitoring** > **Real-Time Queries** to query the task with the longest execution time, and kill the task.

- If the alarm is frequently generated, you are advised to schedule services in off-peak hours or create new resource pools to manage system resources in a more refined manner. For details, see **Creating a Resource Pool**.

----End

Alarm Clearance

This alarm is automatically cleared when the resource pool blocking is relieved.

NOTE

To view historical blocked SQL statements, locate the row that contains the target cluster on the console, choose **Monitoring Panel** in the **Operation** column. On the displayed page, choose **Monitoring** > **History** to query the execution time of historical SQL statements.

9.4.4.8 DWS_2000000020 Long SQL Probe Execution Duration in a Cluster

Alarm Description

GaussDB(DWS) collects the execution status of the SQL probe on each node in the cluster every 30 seconds. If the execution duration of an SQL probe on a server in a cluster exceeds twice the threshold (or another user-defined value), a critical alarm is generated. If the execution duration of all SQL probes falls below the threshold, the critical alarm is cleared.

NOTE

If the SQL probe duration remains higher than the alarm reporting threshold, the alarm is generated again in 24 hours(or another user-defined value).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000020	Management plane alarm	Important	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	Long SQL Probe Execution Duration in a Cluster
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

The cluster performance deteriorates or the cluster is faulty.

Possible Causes

The service load of the cluster is high or the cluster is faulty. As a result, the execution of the SQL probe becomes slow.

Handling Procedure

Step 1 In the navigation pane of the monitoring panel, choose **Utilities** > **SQL Probes**. Check SQL probe execution.

Step 2 In the navigation pane, choose **Monitoring** > **Performance Monitoring**. Check the monitoring metrics such as the CPU usage, disk usage, and memory usage to determine whether the workloads are high or any metric is abnormal.

Step 3 In the navigation pane, choose **Monitoring** > **Real-Time Queries**. Check whether there are queries or sessions that have been running for a long time and affect cluster running. You can terminate abnormal sessions or queries.

----End

Alarm Clearance

This alarm is automatically cleared when the time consumed by an SQL probe on all servers in all clusters falls below the threshold.

9.4.4.9 DWS_2000000023 A Vacuum Full Operation That Holds a Lock for A Long Time Exists in the Cluster

Alarm Description

VACUUM FULL holds a level-8 lock on a table. If it holds the lock on a table for longer than 20 minutes (or another user-defined value), a major alarm is reported, indicating that the VACUUM FULL operation holds a lock for too long in the cluster. This major alarm is cleared when VACUUM FULL is complete.

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000023	Management plane alarm	Important	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	A Vacuum Full Operation That Holds a Table Lock for A Long Time Exists in the Cluster
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

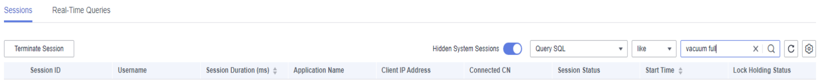
Other operations cannot the table. As a result, workloads cannot be executed.

Possible Causes

There is a VACUUM FULL operation that holds a table lock for a long time in the cluster.

Handling Procedure

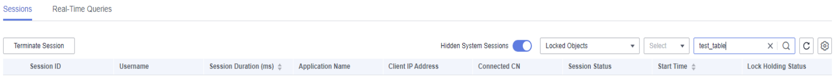
Step 1 In the navigation pane of the monitoring panel, choose **Monitoring** > **Real-Time Queries** > **Sessions**. In the session list, set the search criteria to **LIKE** and search for the keyword **vacuum full**.



Hidden System Sessions
Query SQL
like
vacuum full

Session ID	Username	Session Duration (ms)	Application Name	Client IP Address	Connected CN	Session Status	Start Time	Lock Holding Status

Step 2 Check whether there is a table lock waiting for VACUUM FULL to complete by querying the locked object.



Hidden System Sessions
Locked Objects
Select
test_table

Session ID	Username	Session Duration (ms)	Application Name	Client IP Address	Connected CN	Session Status	Start Time	Lock Holding Status

Step 3 Check whether the VACUUM FULL operation needs to be handled.

1. Check whether VACUUM FULL is a system behavior and whether it affects system functions. If VACUUM FULL does not affect other service queries, wait until it is complete. The alarm will be automatically cleared.
2. If VACUUM FULL affects normal service execution, you can find and kill related sessions on the **Real-Time Queries** tab and re-execute VACUUM FULL later.

----End

Alarm Clearance

This alarm is automatically cleared when the VACUUM FULL operation is complete.

9.4.4.10 DWS_2000000027 Memory Usage of a GaussDB(DWS) Cluster Node Exceeds the Threshold

Alarm Description

GaussDB(DWS) collects the instance memory usage of each node in a cluster every 60 seconds. If a node's instance memory usage exceeds 90% (adjustable), an alarm is reported indicating that the threshold has been surpassed. The alarm will be cleared if the average memory usage falls below 85% (5% below the reporting threshold).

NOTE

If a node's instance average memory usage consistently exceeds the alarm threshold, the alarm will be triggered again after 24 hours (adjustable).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000027	Management plane alarm	Urgent: > 90%	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	Instance Memory Usage of a Cluster Node Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

If instance memory usage remains high for an extended period, service processes may slow down or become unavailable.

Possible Causes

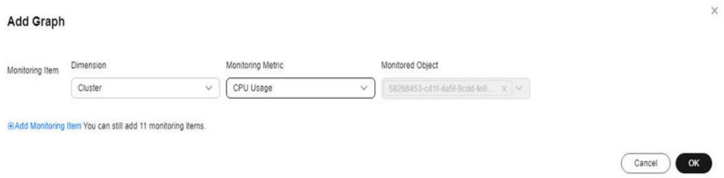
- Complex services occupy a large number of instance memory resources.
- The instance memory of the cluster is too low to meet service requirements.

Handling Procedure

Step 1 Check the memory usage of each node instance.

1. Log in to the GaussDB(DWS) console.
2. Choose **Management** > **Alarms**, select the cluster for which the alarm is generated in the cluster selection drop-down list in the upper right corner, view the alarm information of the cluster in the last seven days, and locate the name of the node for which the alarm is generated based on the location information.
3. Choose **Dedicated Clusters** > **Clusters**, locate the row that contains the cluster for which the alarm is generated, and click **Monitoring Panel** in the **Operation** column.
4. Choose **Monitoring** > **Performance** > **Monitoring View**. On the displayed page, choose an instance to see its memory usage rate. Verify the information and click **OK**.

Figure 9-19 Adding a monitoring view for instance memory usage



Add Graph

Monitoring Item

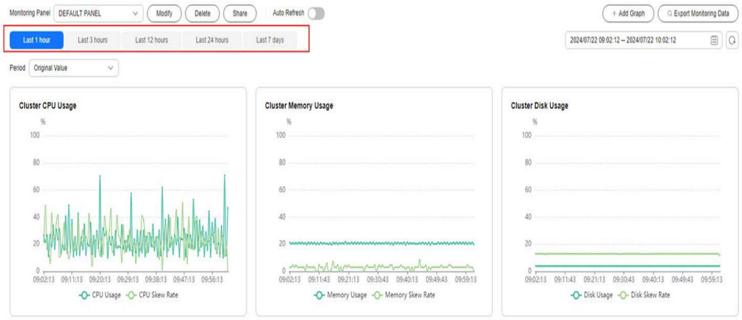
Dimension: Cluster

Monitoring Metric: CPU Usage

Monitored Object: 58268453-c41f-4a5f-9cdd-fe0...

5. You can view the memory usage of each instance in the current cluster at the page's bottom. In the upper left corner, you can check the memory usage of each instance in the last 1, 3, 12, 24 hours, or 7 days. This helps you detect any sudden increase in memory usage of any instance.

Figure 9-20 Instance memory usage monitoring view



Monitoring Panel: DEFAULT PANEL

Auto Refresh

Last 1 hour

2024/07/22 09:02:12 – 2024/07/22 10:02:12

Period: Original Value

   - If the memory usage of an instance frequently increases and then returns to normal in a short period of time, it indicates that the CPU usage temporarily spikes during service execution. In this case, you can adjust the alarm threshold to reduce the number of reported alarms.

- If the instance memory usage remains high for a long time, the cluster is overloaded. In this case, check cluster services or improve cluster specifications. For details, see Changing the Node Flavor.

Step 2 Check whether the memory usage alarm configuration of the instance is proper.

1. Choose **Management** > **Alarms** and click **View Alarm Rule**.
2. Locate the row that contains rule **Instance Memory Usage of a Cluster Node Exceeds the Threshold**, and click **Modify** in the **Operation** column. The **Modifying an Alarm Rule page** is displayed.
3. Adjust the alarm threshold and detection period. A higher alarm threshold and a longer detection period indicate a lower alarm sensitivity. For details about the GUI configuration, see Alarm Rules.

----End

Clearing an Alarm

This alarm is automatically cleared when the memory usage of the instance decreases.

9.4.4.11 DWS_2000000028 Dynamic Memory Usage of GaussDB(DWS) Cluster Nodes Exceeds the Threshold

Alarm Description

GaussDB(DWS) collects the dynamic memory usage of each node in a cluster every 60 seconds. If a node's dynamic memory usage exceeds 90% (adjustable), an alarm is reported indicating that the threshold has been surpassed. The alarm will be cleared if the average memory usage falls below 85% (5% below the reporting threshold).

NOTE

If the average dynamic memory usage of a node is always greater than the alarm threshold, the alarm is generated again 24 hours (configurable).

Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000028	Management plane alarm	Urgent: > 90%	Operation alarm	GaussDB(DWS)	Yes

Alarm Parameters

Category	Name	Description
Location information	Name	Dynamic Memory Usage of a Cluster Node Exceeds the Threshold
	Type	Operation alarm
	Generation time	Time when the alarm is generated
Other information	Cluster ID	Cluster details such as **resourceId** and **domain_id**

Impact on the System

If the usage of dynamic memory remains high for an extended period, it can cause service processes to slow down or become inaccessible.

Possible Causes

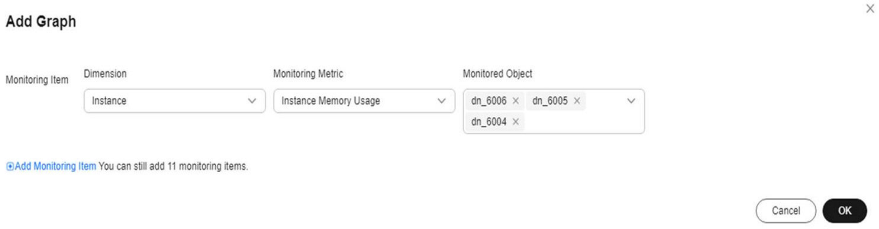
- Complex services occupy a large number of dynamic memory resources.
- The dynamic memory of the cluster is too low to meet service requirements.

Handling Procedure

Step 1 Check the dynamic memory usage of instances on each node.

1. Log in to the GaussDB(DWS) console.
2. Choose **Management** > **Alarms**, select the cluster for which the alarm is generated in the cluster selection drop-down list in the upper right corner, view the alarm information of the cluster in the last seven days, and locate the name of the node for which the alarm is generated based on the location information.
3. Choose **Dedicated Clusters** > **Clusters**, locate the row that contains the cluster for which the alarm is generated, and click **Monitoring Panel** in the **Operation** column.
4. Choose **Monitoring** > **Performance** > **Monitoring View**. On the displayed page, choose an instance to see its dynamic memory usage rate. Verify the information and click **OK**.

Figure 9-21 Adding a monitoring view for dynamic memory usage



Add Graph

Monitoring Item

Dimension: Instance

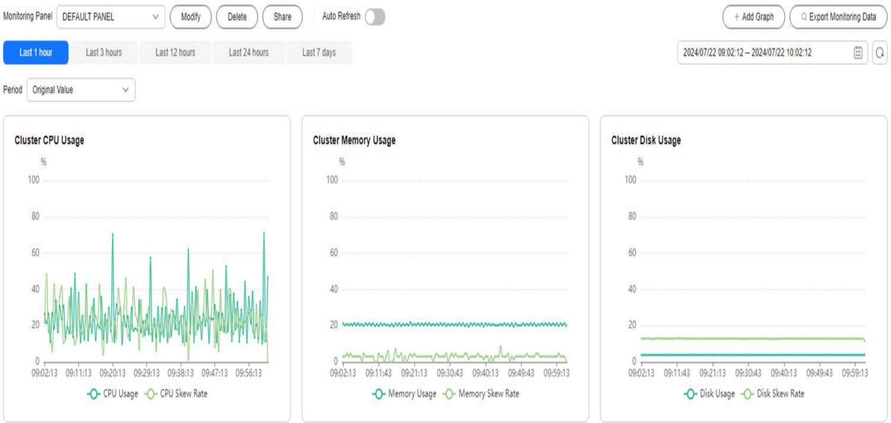
Monitoring Metric: Instance Memory Usage

Monitored Object: dn_6006 × dn_6005 × dn_6004 ×

Add Monitoring Item You can still add 11 monitoring items.

5. You can view the dynamic memory usage of each instance in the current cluster at the page's bottom. In the upper left corner, you can check the dynamic memory usage of each instance in the last 1, 3, 12, 24 hours, or 7 days. This helps you detect any sudden increase in dynamic memory usage of any instance.

Figure 9-22 Monitoring view for dynamic memory usage



Monitoring Panel DEFAULT PANEL

Period Original Value

2024/07/22 09:02:12 – 2024/07/22 10:02:12

- If the dynamic memory usage frequently increases and then returns to normal in a short period of time, it indicates that the CPU usage temporarily spikes during service execution. In this case, you can adjust the alarm threshold to reduce the number of reported alarms.
- If the dynamic memory usage remains high for a long time, the cluster is overloaded. In this case, check cluster services or improve cluster specifications. For details, see **Changing the Node Flavor**.

Step 2 Check whether the configuration of the dynamic memory usage alarm is proper.

1. Choose **Management** > **Alarms** and click **View Alarm Rule**.
2. Locate the row that contains rule **Dynamic Memory Usage of a Cluster Node Exceeds the Threshold**, and click **Modify** in the **Operation** column. The **Modifying an Alarm Rule page** is displayed.
3. Adjust the alarm threshold and detection period. A higher alarm threshold and a longer detection period indicate a lower alarm sensitivity. For details about the GUI configuration, see **Alarm Rules**.

----End

Clearing an Alarm

This alarm is automatically cleared when the memory usage of the instance decreases.

9.4.4.12 DWS_2000000029 Usage of a GaussDB(DWS) Cluster Resource Pool Exceeds the Threshold

Alarm Description

This alarm is generated by the DMS alarm module if the disk usage of the cluster resource pool goes beyond the set threshold within a specific time frame and the suppression conditions are not met. The alarm will be cleared once the disk usage of the cluster resource pool drops below the threshold.

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000029	Management plane alarm	> 90 (Critical); > 80 (Major)	Service alarm	GaussDB(DWS)	Yes

Alarm Changes

Change Type	Change Version	Description	Reason for Change
New	8.2.1.230	New alarm	New alarm

Alarm Parameters

Type	Parameter	Description
Fault Location	Cluster name	Cluster for which the alarm is generated.
	Tenant name	Name of the tenant to which the cluster belongs.
	Alarm level	Severity of the alarm.
Additional Information	Resource ID	ID of the cluster for which the alarm is generated.
	Resource name	Cluster for which the alarm is generated.
	Resource pool name	Name of the resource pool for which the alarm is generated.
	First_alarm_time	First occurrence event of an alarm, including the alarm threshold and current value.

Impact on the System

The disk space of the resource pool is insufficient, affecting service execution.

Possible Causes

The disk usage limit of the resource pool is too small.

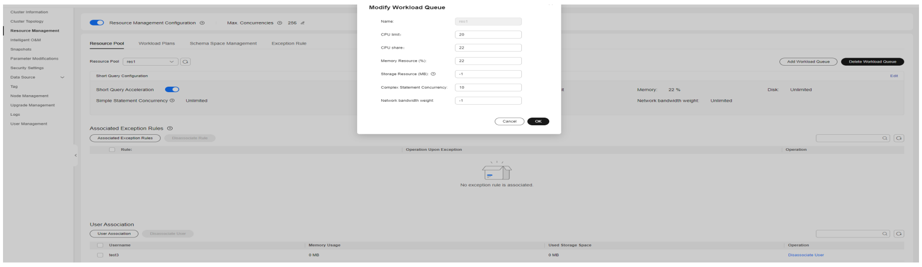
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Alarm** page, view the alarms generated in the last seven days.

Step 3 Choose **Dedicated Clusters** > **Clusters** and locate the cluster based on the cluster information in the alarm details.

Step 4 Click the cluster name. On the cluster details page that is displayed, click **Resource Management** and the **Resource Pool** page is displayed. Modify the disk configuration of the resource pool.



Modify Workload Queue

Name: res1

CPU limit: 20

CPU share: 22

Memory Resource (%): 22

Storage Resource (MB): -1

Complex Statement Concurrency: 10

Network bandwidth weight: -1

----End

Alarm Clearance

This alarm is automatically cleared after the fault is rectified.

Related Information

None

9.4.4.13 DWS_2000000030 Session Usage in a GaussDB(DWS) Cluster Exceeds the Threshold

Alarm Description

This alarm is generated by the DMS alarm module if the session usage in the cluster goes beyond the set threshold within a specific time frame and the suppression conditions are not met. The alarm will be cleared once the session usage in the cluster drops below the threshold.

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000030	Management plane alarm	> 90 (Critical); > 80 (Major)	Service alarm	GaussDB(DWS)	Yes

Alarm Changes

Change Type	Change Version	Description	Reason for Change
New	8.2.1.230	New alarm	New alarm

Alarm Parameters

Type	Parameter	Description
Fault Location	Cluster name	Cluster for which the alarm is generated.
	Tenant name	Name of the tenant to which the cluster belongs.
	Alarm level	Severity of the alarm.
Additional Information	Resource ID	ID of the cluster for which the alarm is generated.
	Resource name	Cluster for which the alarm is generated.
	First_alarm_time	First occurrence event of an alarm, including the alarm threshold and current value.

Impact on the System

The number of available sessions is insufficient, affecting service execution.

Possible Causes

The value of **max_connections** is too small.

Procedure

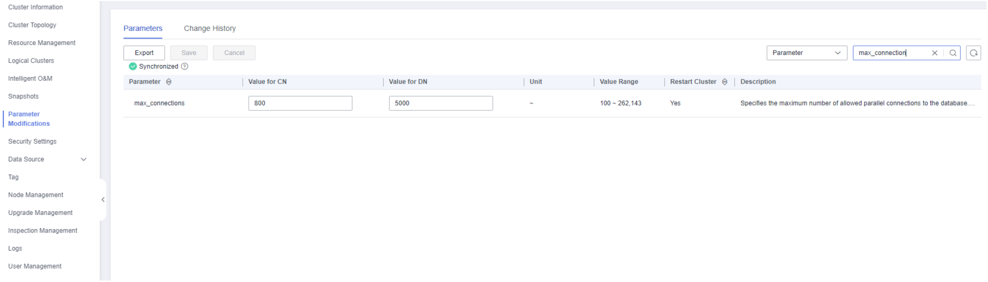
Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Alarm** page, view the alarms generated in the last seven days.

Step 3 Choose **Dedicated Clusters** > **Clusters** and locate the cluster based on the cluster information in the alarm details.

Step 4 Click the cluster name to go to the cluster details page. On the cluster details page, click **Parameter Modifications**, search for **max_connections** in the search box on the right of the parameter list, and change its value.

Figure 9-23 Modifying parameters



Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
max_connections	800	5000	~	100 ~ 262,143	Yes	Specifies the maximum number of allowed parallel connections to the database....

----End

Alarm Clearance

This alarm is automatically cleared after the fault is rectified.

Related Information

None

9.4.4.14 DWS_2000000031 Active Session Usage in a GaussDB(DWS) Cluster Exceeds the Threshold

Alarm Description

The DMS alarm module generates an alarm if the active session usage in the cluster exceeds the set threshold within a specific time frame and the suppression conditions are not met. The alarm is cleared when the DMS alarm module detects that the active session usage in the cluster is below the threshold.

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000031	Management plane alarm	> 90 (Critical); > 80 (Major)	Service alarm	GaussDB(DWS)	Yes

Alarm Changes

Change Type	Change Version	Description	Reason for Change
New	8.2.1.230	New alarm	New alarm

Alarm Parameters

Type	Parameter	Description
Fault Location	Cluster name	Cluster for which the alarm is generated.
	Tenant name	Name of the tenant to which the cluster belongs.
	Alarm level	Severity of the alarm.
Additional Information	Resource ID	ID of the cluster for which the alarm is generated.
	Resource name	Cluster for which the alarm is generated.
	First_alarm_time	First occurrence event of an alarm, including the alarm threshold and current value.

Impact on the System

The number of available sessions is insufficient, affecting service execution.

Possible Causes

The value of **max_active_statements** is too small.

Procedure

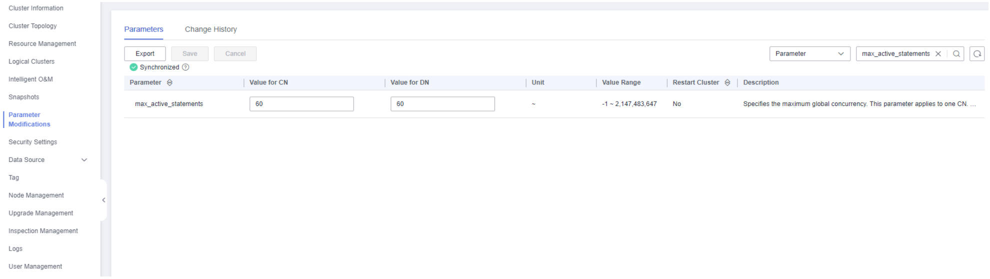
Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Alarm** page, view the alarms generated in the last seven days.

Step 3 Choose **Dedicated Clusters** > **Clusters** and locate the cluster based on the cluster information in the alarm details.

Step 4 Click the name of a cluster to view its details. On the cluster details page, click **Parameter Modifications**, search for **max_active_statements** in the search box on the right of the parameter list, and change its value.

Figure 9-24 Modifying parameters



max_active_statements

Parameter	Value for CN	Value for DN	Unit	Value Range	Restart Cluster	Description
max_active_statements	60	60	~	-1 ~ 2,147,483,647	No	Specifies the maximum global concurrency. This parameter applies to one CN. ...

----End

Alarm Clearance

This alarm is automatically cleared after the fault is rectified.

Related Information

None

9.4.4.15 DWS_2000000032 Number of Database Deadlocks in a GaussDB(DWS) Cluster Exceeds the Threshold

Alarm Description

If the number of deadlocks in the cluster database exceeds the threshold within a specific time frame and the suppression conditions are not met, the DMS alarm module will generate an alarm. The alarm will be cleared once the DMS alarm module detects that the number of deadlocks in the cluster database is below the threshold.

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000032	Management plane alarm	> 10 (Critical); > 1 (Major)	Service alarm	GaussDB(DWS)	Yes

Alarm Changes

Change Type	Change Version	Description	Reason for Change
New	8.2.1.230	New alarm	New alarm

Alarm Parameters

Type	Parameter	Description
Fault Location	Cluster name	Cluster for which the alarm is generated.
	Tenant name	Name of the tenant to which the cluster belongs.
	Alarm level	Severity of the alarm.
Additional Information	Resource ID	ID of the cluster for which the alarm is generated.
	Resource name	Cluster for which the alarm is generated.

Type	Parameter	Description
	Database name	Name of the database for which the alarm is generated.
	First_alarm_time	First occurrence event of an alarm, including the alarm threshold and current value.

Impact on the System

The connection pool is unable to allocate more connections to service requests because a large number of lock requests are causing connections to be unresponsive.

Possible Causes

Resource contention and lock actions are mutually exclusive.

Procedure

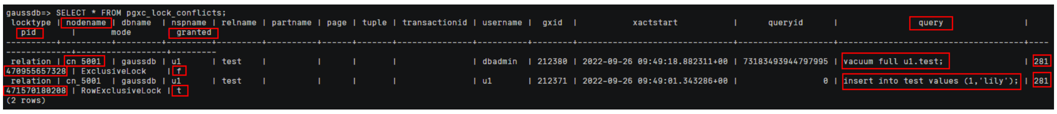
Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Alarm** page, view the alarms generated in the last seven days.

Step 3 Use gsql to connect to the cluster based on the alarm information. For details, see **Using the CLI to Connect to a GaussDB(DWS) Cluster**.

Step 4 Connect to the cluster and run the SQL statement to query the current lock conflict statement.

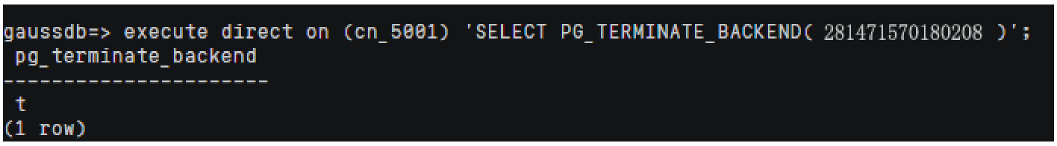
```
select * from pgxc_lock_conflicts;
```



```
gaussdb=> SELECT * FROM pgxc_lock_conflicts;
 locktype | nodename | dbname  | nspname | relname | partname | page | tuple | transactionid | username |  gxid  |          xactstart            |      queryid      |               query                 |
  pid         |       mode        | granted
----------+----------+---------+---------+---------+----------+------+-------+---------------+----------+--------+-------------------------------+-------------------+-------------------------------------+----
--------------+-------------------+---------
 relation | cn_5001  | gaussdb | u1      | test    |          |      |       |               | dbadmin  | 212380 | 2022-09-26 09:49:18.882311+00 | 73183493944797995 | vacuum full u1.test;                | 281
470955657328 | ExclusiveLock     | f
 relation | cn_5001  | gaussdb | u1      | test    |          |      |       |               | u1       | 212371 | 2022-09-26 09:49:01.343286+00 |                 0 | insert into test values (1,'lily'); | 281
471570180208 | RowExclusiveLock  | t
(2 rows)
```

Step 5 Decide whether to terminate the lock based on the statement content. To terminate the lock, run the following statement. **pid** and **nodename** are obtained from the previous step.

execute direct on (*nodename*) 'SELECT PG_TERMINATE_BACKEND(*pid*)';



```
gaussdb=> execute direct on (cn_5001) 'SELECT PG_TERMINATE_BACKEND( 281471570180208 )';
 pg_terminate_backend
----------------------
 t
(1 row)
```

----End

Alarm Clearance

This alarm is automatically cleared after the fault is rectified.

Related Information

None

9.4.4.16 DWS_2000000033 GaussDB(DWS) Cluster Database Session Usage Exceeds the Threshold

Alarm Description

The DMS alarm module will generate an alarm if the session usage of the cluster database goes over the threshold within a specific time frame and the suppression conditions are not met. The alarm will be resolved by the DMS alarm module once it detects that the session usage of the cluster database is below the threshold.

Alarm Attributes

Alarm ID	Alarm Category	Alarm Severity	Alarm Type	Service Type	Auto Cleared
DWS_2000000033	Management plane alarm	> 90 (Critical); > 80 (Major)	Service alarm	GaussDB(DWS)	Yes

Alarm Changes

Change Type	Change Version	Description	Reason for Change
New	8.2.1.230	New alarm	New alarm

Alarm Parameters

Type	Parameter	Description
Fault Location	Cluster name	Cluster for which the alarm is generated.
	Tenant name	Name of the tenant to which the cluster belongs.
	Alarm level	Severity of the alarm.
Additional Information	Resource ID	ID of the cluster for which the alarm is generated.
	Resource name	Cluster for which the alarm is generated.
	Database name	Name of the database for which the alarm is generated.

Type	Parameter	Description
	First_alarm_time	First occurrence event of an alarm, including the alarm threshold and current value.

Impact on the System

The number of available database connections is insufficient, affecting service execution.

Possible Causes

The maximum number of database connections is too small.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 On the **Alarm** page, view the alarms generated in the last seven days.

Step 3 Use gsql to connect to the cluster based on the alarm information. For details, see **Using the CLI to Connect to a GaussDB(DWS) Cluster**.

Step 4 Specify the maximum number of connections a user can have by using syntax **CONNECTION LIMIT connlimit** in the **CREATE ROLE** statement. If you need to modify this limit later, you can use the same syntax in the **ALTER ROLE** statement.

1. Use **PG_ROLES** to check the maximum number of connections of a specified user.

```
SELECT ROLNAME,ROLCONNLIMIT FROM PG_ROLES WHERE ROLNAME='role1';
 rolname | rolconnlimit
---------+--------------
 role1   |           10
(1 row)
```

2. Change the maximum number of connections a user can have.

```
ALTER ROLE role1 connection limit 20;
```

----End

Alarm Clearance

This alarm is automatically cleared after the fault is rectified.

Related Information

None

9.5 Viewing and Subscribing to GaussDB(DWS) Cluster Events

9.5.1 Event Notifications Overview

Overview

GaussDB(DWS) uses the Simple Message Notification (SMN) service to send notifications of GaussDB(DWS) events. The SMN function is only available by subscription. In a subscription, you need to specify one or more event filtering conditions. When an event that matches all filtering conditions occurs, GaussDB(DWS) sends a notification based on the subscription. The filter conditions include the **Event Type** (for example, **Management**, **Monitoring**, or **Security**), **Event Severity** (for example, **Normal** or **Warning**), and **Event Source Category** (for example, **Cluster** or **Snapshot**).

Supported Event Types and Events

Events are records of changes in the user's cluster status. Events can be triggered by user operations (such as audit events), or may be caused by cluster service status changes (for example, cluster repaired successfully or failed to repair the cluster). The following tables list the events and event types supported by GaussDB(DWS).

- The following table lists the events whose **Event Source Category** is **Cluster**.

Table 9-13 Events whose **Event Source Category** is **Cluster**

Event Type	Event Name	Event Severity	Event
Management	createClusterFail	Warning	Cluster creation failed.
Management	createClusterSuccess	Normal	The cluster is created.
Management	createCluster	Normal	Cluster creation started.
Management	extendCluster	Normal	Cluster scale-out started.
Management	extendClusterSuccess	Normal	A cluster is scaled out.
Management	extendClusterFail	Warning	Cluster scale-out failed.
Management	deleteClusterFail	Warning	Failed to delete the cluster.
Management	deleteClusterSuccess	Normal	The cluster is deleted.
Management	deleteCluster	Normal	Cluster deletion started.

Event Type	Event Name	Event Severity	Event
Management	restoreClusterFail	Warning	Cluster restoration failed.
Management	restoreClusterSuccess	Normal	The cluster is restored.
Management	restoreCluster	Normal	Cluster restoration started.
Management	restartClusterFail	Warning	Cluster restart failed.
Management	restartClusterSuccess	Normal	The cluster is restarted.
Management	restartCluster	Normal	Cluster restarted.
Management	configureMRSExtDataSources	Normal	Configuration of MRS external data source for the cluster started.
Management	configureMRSExtDataSourcesFail	Warning	The cluster's MRS external data source configurations failed.
Management	configureMRSExtDataSourcesSuccess	Normal	The MRS external data source of the cluster is configured.
Management	deleteMRSExtDataSources	Normal	Deletion of MRS external data source for the cluster started.
Management	deleteMRSExtDataSourcesFail	Warning	The deletion of the MRS external data source for the cluster failed.
Management	deletedMRSExtDataSourcesSuccess	Normal	MRS external data source is deleted.
Management	bindEipToCluster	Normal	An EIP is bound to the cluster.
Management	bindEipToClusterFail	Warning	Cluster EIP binding failed.
Management	unbindEipToCluster	Normal	The EIP is unbound from the cluster.
Management	unbindEipToClusterFail	Warning	Cluster EIP unbinding failed

Event Type	Event Name	Event Severity	Event
Management	refreshEipToCluster	Normal	The cluster's EIP is refreshed.
Management	refreshEipToCluster-Fail	Warning	Cluster EIP refreshing failed.
Management	dmsClusterMonitoringEnabledSuccessfully	Normal	DMS cluster monitoring is enabled.
Management	failedToEnableDmsClusterMonitoring	Normal	Enabling DMS cluster monitoring has failed.
Management	dmsClusterMonitoringDisabledSuccessfully	Normal	DMS cluster monitoring is disabled.
Management	failedToDisableDmsClusterMonitoring	Normal	Disabling DMS cluster monitoring has failed.
Management	dmsMetricCollectionEnabledSuccessfully	Normal	DMS collection is enabled.
Management	failedToEnableDmsMetricCollection	Normal	Enabling DMS collection has failed.
Management	dmsMetricCollectionDisabledSuccessfully	Normal	DMS collection is disabled.
Management	failedToDisableDmsMetricCollection	Normal	Disabling DMS collection has failed.
Management	dmsMetricCollectionResetSuccessfully	Normal	DMS collection is reset.
Management	failedToResetDmsMetricCollection	Normal	Failed to reset DMS collection.
Management	dmsMetricCollectionUpdatedSuccessfully	Normal	DMS collection updated successfully.
Management	failedToUpdateDmsMetricCollection	Normal	Updating DMS collection has failed.
Management	dmsMetricDataRetentionPeriodUpdatedSuccessfully	Normal	DMS collection storage time is updated.
Management	failedToUpdateTheDmsMetricDataRetentionPeriod	Normal	Updating DMS collection storage time has failed.

Event Type	Event Name	Event Severity	Event
Management	dmsSessionsTerminatedSuccessfully	Normal	DMS terminated a session.
Management	failedToTerminateDmsSessions	Normal	DMS was unable to terminate a session.
Management	dmsQueriesTerminatedSuccessfully	Normal	DMS queries terminated.
Management	failedToTerminateDmsQueries	Normal	DMS was unable to terminate a query.
Management	dmsCreateWDRSuccessfully	Normal	The DMS load report generation task was completed.
Management	failedToCreateWDR	Warning	The DMS was unable to complete the load report generation task.
Management	dmsDeleteWDRSuccessfully	Normal	DMS deleted the load report.
Management	failedToDeleteWDR	Warning	DMS was unable to delete a workload report.
Management	dmsUpdateWDRConfigSuccessfully	Normal	DMS updated the workload report parameters.
Management	failedToUpdateWDRConfig	Warning	DMS was unable to update workload report parameters.
Management	dmsCreateWorkloadSnapshotSuccessfully	Normal	DMS successfully added a workload snapshot.
Management	failedToCreateWorkloadSnapshot	Warning	DMS was unable to add a workload snapshot.
Security	resetPasswordFail	Warning	The password reset attempt was unsuccessful.
Security	resetPasswordSuccess	Normal	The cluster password has been reset.
Security	updateConfiguration	Normal	Start to update cluster security parameters.

Event Type	Event Name	Event Severity	Event
Security	updateConfiguration-Fail	Warning	Cluster security parameter update failed.
Security	updateConfiguration-Success	Normal	Cluster security parameters were updated.
Monitoring	repairCluster	Normal	The node is faulty and the cluster starts to be repaired.
Monitoring	repairClusterFail	Warning	Cluster repairing failed.
Monitoring	repairClusterSuccess	Normal	The cluster is repaired.

- The following table lists the events whose **Event Source Category** is **Snapshot**.

Table 9-14 Events whose **Event Source Category** is **Snapshot**

Event Type	Event Name	Event Severity	Event
Management	deleteBackup	Normal	The snapshot is deleted.
Management	deleteBackupFail	Warning	Snapshot deletion failed.
Management	createBackup	Normal	The snapshot is being created.
Management	createBackupSuccess	Normal	The snapshot is created.
Management	createBackupFail	Warning	Snapshot creation failed.

- The following table lists the events whose **Event Source Category** is **DR**.

Table 9-15 Events whose **Event Source Category** is **DR**.

Event Type	Event Name	Event Severity	Event
Management	beginCreateDisasterRecovery	Normal	The DR task is being created.
Management	createDisasterRecoverySuccess	Normal	The DR task is created.
Management	createDisasterRecoveryFail	Warning	DR task creation failed.
Management	beginStartDisasterRecovery	Normal	The DR task starts.
Management	startDisasterRecoverySuccess	Normal	The DR task started successfully.
Management	startDisasterRecoveryFail	Warning	The DR task was unable to start.
Management	beginStopDisasterRecovery	Normal	The DR task is being stopped.
Management	stopDisasterRecoverySuccess	Normal	DR stopped successfully.
Management	stopDisasterRecoveryFail	Warning	The DR task could not be stopped.
Management	beginSwitchoverDisasterRecovery	Normal	The DR switchover starts.
Management	switchoverDisasterRecoverySuccess	Normal	The DR switchover is successful.
Management	switchoverDisasterRecoveryFail	Warning	The DR switchover failed.
Management	beginDeleteDisasterRecovery	Normal	The DR task is being deleted.

Event Type	Event Name	Event Severity	Event
Management	deleteDisasterRecoverySuccess	Normal	The DR task is deleted.
Management	deleteDisasterRecoveryFail	Warning	The DR task deletion failed.
Management	disasterRecoveryAbnormal	Warning	The DR task runs abnormally.
Management	beginFailoverDisasterRecovery	Normal	The abnormal switchover starts.
Management	failoverDisasterRecoverySuccess	Normal	The abnormal switchover is successful.
Management	failoverDisasterRecoveryFail	Warning	The abnormal switchover fails.
Management	beginRecoveryDisaster	Normal	The disaster recovery starts.
Management	recoveryDisasterSuccess	Normal	The disaster recovery is successful.
Management	recoveryDisasterFail	Warning	The disaster recovery fails.
Management	emptyDisasterRecovery	Warning	No DR table exists in the current DR object.
Management	switchoverContinueAsFailoverDisasterRecovery	Warning	The DR switchover is degraded to an abnormal switchover.

- The following table lists the events whose event source type is data migration.

Table 9-16 Events whose event source type is data migration

Event Type	Event Name	Event Severity	Event
Data migration	dataMigrationApplication-DetectedAbnormal	Warning	The job task status is abnormal.
Data migration	dataMigrationApplication-ReturnNormal	Normal	The job task is restored.
Data migration	dataMigrationCreateApplication	Normal	Create a job task.
Data migration	dataMigrationCreateCluster	Normal	Start to create a data migration instance.
Data migration	dataMigrationCreateClusterFailed	Warning	Failed to create the data migration instance.
Data migration	dataMigrationCreateClusterSuccess	Normal	The data migration instance is created.
Data migration	dataMigrationCreateConnection	Normal	Create a connection.
Data migration	dataMigrationCreateMapping	Normal	Create a table mapping configuration.
Data migration	dataMigrationDeleteApplication	Normal	Start to delete the job task.
Data migration	dataMigrationDeleteApplicationFailed	Warning	Failed to delete the job.
Data migration	dataMigrationDeleteApplicationSuccess	Normal	The job is deleted.
Data migration	dataMigrationDeleteCluster	Normal	Start to delete the data migration instance.
Data migration	dataMigrationDeleteClusterApplication	Normal	Start to delete the job task.

Event Type	Event Name	Event Severity	Event
Data migration	dataMigrationDeleteClusterApplicationFailed	Warning	Failed to delete the job.
Data migration	dataMigrationDeleteClusterApplicationSuccess	Normal	The job is deleted.
Data migration	dataMigrationDeleteClusterFailed	Warning	Failed to delete the data migration instance.
Data migration	dataMigrationDeleteClusterSuccess	Normal	The data migration instance is deleted.
Data migration	dataMigrationDeleteConnection	Normal	Delete the connection configuration.
Data migration	dataMigrationDeleteMapping	Normal	Delete the table mapping configuration.
Data migration	dataMigrationDialsConnection	Normal	Test the connection configuration.
Data migration	dataMigrationModifyConnection	Normal	Modify the connection configuration.
Data migration	dataMigrationModifyMapping	Normal	Modify the table mapping configuration.
Data migration	dataMigrationStartApplication	Normal	Start the job task.
Data migration	dataMigrationStartApplicationFailed	Warning	Failed to start the job.
Data migration	dataMigrationStartApplicationSuccess	Normal	The job task is started.
Data migration	dataMigrationStopApplication	Normal	Start to stop the job.

Event Type	Event Name	Event Severity	Event
Data migration	dataMigrationStopApplicationFailed	Warning	Failed to stop the job.
Data migration	dataMigrationStopApplicationSuccess	Normal	The job task is stopped.

9.5.2 Subscribing to Event Notifications

After subscribing to GaussDB(DWS) event notification, you will receive notifications by text message, email, or application when management, monitoring, or security events occur in a specific cluster or snapshot.

Creating a Subscription

Step 1 Log in to the GaussDB(DWS) management console.

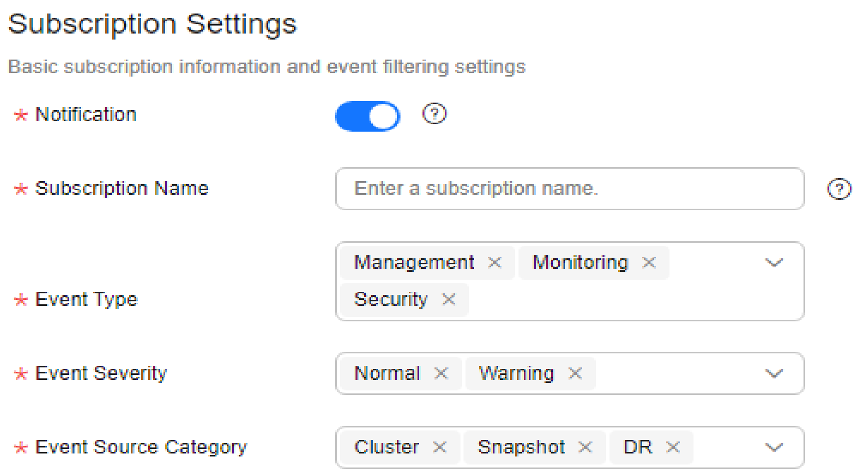
Step 2 In the navigation tree on the left, choose **Management** > **Events**.

Step 3 On the **Event Management** page, choose **Subscription** > **Create Subscription**.

Step 4 In the **Subscription Settings** area, set basic subscription information and event filtering.

The **Subscribed Event List** area displays the events filtered by the system based on the subscription settings.

Figure 9-25 Subscription Settings



Subscription Settings

Basic subscription information and event filtering settings

* Notification: on

* Subscription Name: Enter a subscription name.

* Event Type: Management, Monitoring, Security

* Event Severity: Normal, Warning

* Event Source Category: Cluster, Snapshot, DR

Table 9-17 Subscription parameters

Parameter	Description
Notification	Enable or disable event subscription. After notification is disabled, the system stops sending notifications of subscribed events but does not delete the subscription.
Subscription Name	Enter the name of a subscription. ● The name can contain letters (upper or lower case), digits, hyphens (-), and underscores (_) and must start with a letter or digit. ● The name must be between 1 and 256 characters in length.
Event Type	Select the type of the event to be subscribed. Possible values are **Management**, **Monitoring**, and **Security**.
Event Severity	Select the alarm severity of the event. Possible values are **Normal** and **Warning**.
Event Source Category	Select the event source category: cluster, snapshot,.

Step 5 Select a message notification topic from the **Message Notification Topic** drop-down list.

- The selected topic must have granted GaussDB(DWS) the permission to publish messages to the topic.

  If GaussDB(DWS) has not been authorized to publish messages to the selected topic, go to the topic management page of the SMN console to configure topic authorization. For details, see **Topic Management** > **Configuring Topic Policies** in the *Simple Message Notification User Guide*. When configuring the topic policy, select **GaussDB(DWS)** for **Services that can publish messages to this topic**.
- To create a topic, click **Create Topic**. The SMN console is displayed. For details, see **Topic Management > Creating a Topic** in the *Simple Message Notification User Guide*.

Step 6 Click **OK** to complete the subscription.

----End

Modifying the Subscription

Step 1 Choose **Management** > **Events** and click **Subscriptions**.

Step 2 In the **Operation** column of the row containing the specified subscription, click **Edit** to enter the **Edit Subscription** page.

Step 3 On the **Edit Subscription** page, set the parameters to be modified. For details, see **Step 4** to **Step 6** in section "Creating a Subscription".

----End

Deleting the Subscription

Step 1 Choose **Management** > **Events** and click **Subscriptions**.

Step 2 In the **Operation** column of the row containing the specified subscription, click **Delete**. The **Delete Subscription** dialog box is displayed.

Step 3 Click **Yes** to delete the subscription.

----End

9.5.3 Viewing Events

This section describes how to search for events that occur in a cluster or snapshot.

Step 1 Log in to the GaussDB(DWS) management console.

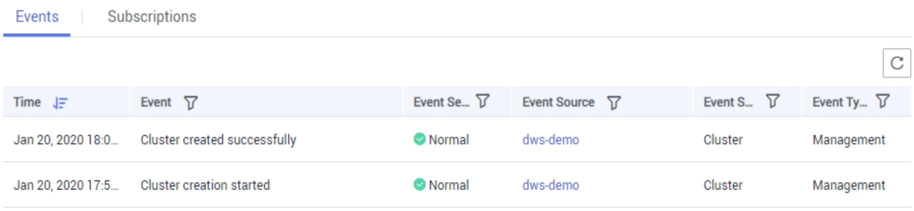
Step 2 In the navigation tree on the left, choose **Management** > **Events**.

On the **Events** tab page, all events that occur in the clusters or snapshots are displayed by default.

You can sort the events in descending or ascending order by clicking ⇕ next to **Time**.

You can search for events by time, event, event level, event source, event source type, or event type using the search box at the top of the event list.

Figure 9-26 Event page



Time	Event	Event Se...	Event Source	Event S...	Event Ty...
Jan 20, 2020 18:0...	Cluster created successfully	Normal	dws-demo	Cluster	Management
Jan 20, 2020 17:5...	Cluster creation started	Normal	dws-demo	Cluster	Management

----End

9.6 Common O&M Commands of GaussDB(DWS)

This section lists only common O&M commands. The system objects to be queried can be changed based on the site requirements. For details about the returned fields, see the description of system catalogs, system views, and system functions in the product manuals*Developer Guide*.

Viewing O&M Status

Prerequisites: The GaussDB(DWS) cluster has been connected.

- View the overall running status of the current service.
```
select coorname, usename, client_addr, sysdate-query_start as duration, state, enqueue,waiting, pid, query_id, substr(query,1,60) from pgxc_stat_activity where usename != 'Ruby' and usename != 'omm' and state = 'active' order by duration desc;
```
- View the overall concurrency of the current service.

```
select usename,coorname,enqueue,state,count(*) from pgxc_stat_activity where usename <> 'omm'
and usename <> 'Ruby' group by 1,2,3,4 order by 4,5 desc limit 30;
```

- Check the overall waiting status in the current cluster.

```
select wait_status,wait_event,count(*) as cnt from pgxc_thread_wait_status where wait_status <> 'wait
cmd' and wait_status <> 'synchronize quit' and wait_status <> 'none' and wait_status <> 'wait stream
task' group by 1,2 order by 3 desc limit 50;
```

- View the service running information of the resource pool in the current cluster (resource management and control scenario).

```
select s.resource_pool as rpname, count(1) as session_cnt,sum(case when a.state = 'active' then 1 else
0 end) as active_cnt,sum(case when s.enqueue ='global' then 1 else 0 end) as global_wait,sum(case
when s.lane = 'fast' and s.status = 'running' then 1 else 0 end) as fast_run,sum(case when s.lane =
'fast' and s.status = 'pending' and s.enqueue not in ('global','none') then 1 else 0 end) as
fast_wait,sum(case when s.lane = 'slow' and s.status = 'running' then 1 else 0 end) as
slow_run,sum(case when s.lane = 'slow' and s.status = 'pending' and s.enqueue not in ('global','none')
then 1 else 0 end) as slow_wait,sum(case when s.status = 'running' then s.statement_mem else 0
end) as est_mem from pg_catalog.pgxc_session_wlmstat s,pg_catalog.pgxc_stat_activity a where
s.threadid=a.pid(+) and s.attribute != 'internal' and s.resource_pool != 'root' group by 1;
```

- Check the dynamic memory watermark of the current cluster.

```
select a.nodename,a.memorymbytes as dynamic_used_memory,b.memorymbytes as
max_dynamic_memory, dynamic_used_memory/max_dynamic_memory*100 as used_rate  from
pgxc_total_memory_detail a join pgxc_total_memory_detail b on a.nodename=b.nodename  where
a.memorytype = 'dynamic_used_memory' and b.memorytype = 'max_dynamic_memory' order by
a.memorymbytes desc;
```

- Check the memory usage of each thread. (**Check the dynamic memory watermark of the current cluster.** Connect to the CN/DN node with high dynamic memory usage that we found to retrieve the information.)

```
select b.state, sum(totalsize) as totalsize, sum(freesize) as freesize, sum(usedsize) as usedsize from
pv_session_memory_detail a , pg_stat_activity b where split_part(a.sessid,'.',2) = b.pid group by b.state
order by totalsize desc limit 20;
```

- Check the memory used by each session in the current instance. (**Check the dynamic memory watermark of the current cluster.** Connect to the CN/DN node with high dynamic memory usage that we found to retrieve the information.)

```
select split_part(pv_session_memory_detail.sessid,'.',2) pid,pg_size_pretty(sum(totalsize))
total_size,count(*) context_count from pv_session_memory_detail group by pid order by
sum(totalsize) desc;
```

- Check the memory used by each SQL statement in the current instance. (**Check the dynamic memory watermark of the current cluster.** Connect to the CN/DN node with high dynamic memory usage that we found to retrieve the information.)

```
select sessid, contextname, level,parent, pg_size_pretty(totalsize) as total ,pg_size_pretty(freesize) as
freesize, pg_size_pretty(usedsize) as usedsize, datname,query_id, query from
pv_session_memory_detail a , pg_stat_activity b where split_part(a.sessid,'.',2) = b.pid order by
totalsize desc limit 100;
```

Emergency Recovery

⚠ CAUTION

Always confirm emergency operations that may impact services with the customer before proceeding. Never attempt to perform these operations independently.

Prerequisites: The GaussDB(DWS) cluster has been connected.

- **View the overall running status of the current service.** Obtain the PID of the statement to be scanned.

```
execute direct on(cn_name) 'select pg_cancel_backend(PID of the scanned statement)';
execute direct on(cn_name) 'select pg_terminate_backend(PID of the scanned statement)';
```

- Group scan statements together in batches, without executing the scan command.

```
select 'execute direct on(' || coorname || ') ''select pg_terminate_backend(' || pid || ')'';',sysdate-
query_start as dur, substr(query,1,60) from pgxc_stat_activity where usename != 'omm' and usename !
= 'Ruby' and state = 'active' order by dur desc limit 30;
```

- Clear idle connections.

```
clean connection to all for database xxx;    --Clear idle connections.
select * from pgxc_clean_free_conn();        --Clear pooler cache connections.
```

- Fix the CCN count by connecting to the CCN.

```
select * from pg_stat_get_workload_struct_info();  --Retain the CCN information.
select gs_wlm_node_recover(true);  --Repair CCNs.
```

- Lock abnormal users.

```
alter user usename account lock;    --Lock a user.
alter user usename account unlock;  --Unlock a user.
```

- Add a service to the blacklist.

```
select * from gs_append_blocklist(unique_sql_id);   --Add a service to the blacklist.
select * from gs_blocklist_query;   --Query the existing blacklist.
select gs_remove_blocklist(unique_sql_id);  --Remove the blacklist.
```

Service Analysis

Prerequisites: The GaussDB(DWS) cluster has been connected.

- View the waiting status of the currently executing SQL statement. Run the statement user for **viewing the overall running status of the current service** and obtain the value of the **query_id** field, which can be used as the **Actual query ID** in the following statement.

```
select * from pgxc_thread_wait_status where query_id = Actual query ID order by
node_name,wait_status,wait_event;
```

- View the running process information of the running SQL statement. Run the statement user for **viewing the overall running status of the current service** and obtain the value of the **query_id** field, which can be used as the **Actual query ID** in the following statement.

```
select * from pgxc_wlm_session_statistics where queryid = Actual query ID;
```

- View the running information of historical SQL statements. Run the statement used for **viewing the overall running status of the current service** and obtain the value of the **query_id** field, which can be used as the **Actual query ID** in the following statement.

```
select * from pgxc_wlm_session_info where queryid = Actual query ID;
```

- View the skew information of a single table.

```
select * from table_distribution('schema_name','table_name');
```

- View the dirty page rate of a single table.

```
select c.oid AS relid, n.nspname AS schemaname, c.relname, pg_stat_get_tuples_inserted(c.oid) AS
n_tup_ins, pg_stat_get_tuples_updated(c.oid) AS n_tup_upd, pg_stat_get_tuples_deleted(c.oid) AS
n_tup_del, pg_stat_get_live_tuples(c.oid) AS n_live_tup, pg_stat_get_dead_tuples(c.oid) AS n_dead_tup,
cast( (n_dead_tup / (n_live_tup + n_dead_tup + 0.0001) * 100) AS numeric(5,2)) AS dirty_page_rate
from pg_class c LEFT JOIN pg_namespace n ON n.oid = c.relnamespace where c.oid =
'schema_name.table_name'::regclass::oid;
```

- View the table definition and index information.

```
select pg_get_tabledef('schema_name.table_name');
```

- Check the table size.

```
select pg_size_pretty(pg_table_size('schema_name.table_name'));
```

- View the creation time, modification time, and last analysis time of the table.

```
select * from pg_object where object_oid='schema_name.table_name'::regclass;
```

- View details about dirty data.

```
start transaction read only;
set enable_show_any_tuples = true;
set enable_indexscan = off;
set enable_bitmapscan = off;
select ctid,xmin,xmax,pgxc_is_committed(xmin),pgxc_is_committed(xmax),oid,* from
schema_name.table_name;
select xmin,xmax,ctid, * from pgxc_node;
rollback;
```

9.7 Backing Up and Restoring a GaussDB(DWS) Cluster

9.7.1 Overview

A snapshot is a full or incremental backup of a GaussDB(DWS) cluster at a specific point in time. It records the current database data and cluster information, including the number of nodes, node specifications, and database administrator name. Snapshots can be created manually or automatically. For details, see **Manual Snapshots** and **Automated Snapshots**.

If you restore a snapshot to a new cluster, GaussDB(DWS) creates a new cluster based on the cluster information recorded in the snapshot, and then restores data from the snapshot. For more information, see **Restoring a Snapshot to a New Cluster**.

If you restore a snapshot to the original cluster, GaussDB(DWS) clears the existing data in the cluster, and then restores the database information from the snapshot to the cluster. For more information, see **Restoring a Snapshot to the Original Cluster**.

The snapshot backup and restoration rates are listed below. (The rates are obtained from the test environment with local SSDs as the backup media. The rates are for reference only. The actual rate depends on your disk, network, and bandwidth resources.)

- Backup rate: 200 MB/s/DN
- Restoration rate: 125 MB/s/DN

Constraints and Limitations

- **Backing up the cluster is essential for maintaining data reliability, especially when the service provider cannot restore data through upstream re-import. This helps prevent data loss caused by human or other factors.**
- The cluster versions that support schema-level snapshots are listed below. If the current console interface does not support this feature, contact technical support.
  - 9.1.0.100 or later
  - 8.3.0.110 or later 8.3.0.*xxx* cluster versions
  - 8.2.1.230 or later 8.2.1.2*xx* versions
- OBS snapshot storage space and billing description
  - The cluster storage is provided by GaussDB(DWS) free of charge. Cluster storage = Storage space per node x Number of nodes

  - GaussDB(DWS) provides some free-of-charge storage space for you to store snapshot data generated in cluster backup. However, if you use more space than the free-of-charge storage space, the exceeded part is charged based on OBS billing rules. For details, see the **OBS pricing details**.
- The dependency of the snapshot service is as follows:
  - The snapshot management function depends on OBS or NFS.
  - The backup device employs disk mounting mode for NFS backup media, reliant on the cloud-based SFS-Tubor service. For details, see **11.1.3.2 Automatic Snapshot Policy**.
  - Only the snapshots stored in OBS can be used to restore data to a new cluster.
- The new GaussDB(DWS) cluster created based on the snapshot must have the same configurations as the original cluster. That is, the number and specifications of nodes, memory, and disks in the new cluster must be the same as those in the original cluster.
- If you create a new cluster based on a snapshot without modifying parameters, the parameters of the new cluster will be the same as those of the snapshot.
- A storage-compute coupled data warehouse (standalone) does not support snapshots.
- Only clusters of version 9.0.2 and later support the snapshot feature. However, if there is an elastic logical cluster within the cluster, backup and restoration will not be supported. It is recommended to remove the elastic logical cluster before attempting backup and restoration.
- To perform snapshot restoration of a GaussDB(DWS) storage-compute decoupled cluster billed in hybrid mode, you can choose either yearly/monthly or pay-per-use billing. For example, if the cluster has three nodes of each billing type, restoring the cluster will change the billing mode of all six nodes to either pay-per-use or yearly/monthly.
- During snapshot creation, do not perform the **VACUUM FULL** operation, or the cluster may become read-only.
- Snapshot creation affects disk I/O performance. You are advised to create snapshots during off-peak hours.
- During the snapshot creation, some intermediate files are retained, which occupy extra disk space. Therefore, create snapshots in off-peak hours and ensure that the disk capacity usage is less than 70%.

9.7.2 Manual Snapshots

9.7.2.1 Creating a Manual Snapshot of a Cluster

Prerequisites

A cluster snapshot is a complete backup that records point-in-time configuration data and service data of a GaussDB(DWS) cluster. This section describes how to create a snapshot on the **Snapshots** page to back up cluster data.

A manual snapshot can be created at any time. It will be retained until it is deleted from the GaussDB(DWS) console. Manual snapshots are full backup data, which takes a long time to create.

NOTE

- Manual cluster snapshots can be backed up to OBS or NFS.
- To create a manual snapshot of a cluster, the cluster state must be **Available**, **To be restarted**, or **Unbalanced**. In cluster versions earlier than 8.1.3.101, you can also create a snapshot of a cluster in the **Read-only** state.

Impact on the System

If a snapshot is being created for a cluster, the cluster cannot be restarted, scaled, its password cannot be reset, and its configurations cannot be modified.

NOTE

To ensure the integrity of snapshot data, do not write data during snapshot creation.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**.

Step 3 Click **Create Snapshot** in the upper right corner. Alternatively, choose **More** > **Create Snapshot** in the **Operation** column.

Step 4 Configure the following snapshot information:

- **Cluster Name**: Select a GaussDB(DWS) cluster from the drop-down list. The drop-down list only displays clusters that are in the **Available** state.
- **Snapshot Name**: Enter a snapshot name. The snapshot name must be 4 to 64 characters in length and start with a letter. It is case-insensitive and contains only letters, digits, hyphens (-), and underscores (_).
- **Snapshot Level**: Select **cluster**.
- **Snapshot Description**: Enter the snapshot information. This parameter is optional. Snapshot information contains 0 to 256 characters and does not support the following special characters: !<>'=&"

Step 5 Click **Create**.

Task status of the cluster for which you are creating a snapshot is **Creating snapshot**. The status of the snapshot that is being created is **Creating**. After the snapshot is created, its status changes to **Available**.

NOTE

If the snapshot size is much greater than that of the data stored in the cluster, the data is possibly labeled with a deletion tag, but is not cleared and reclaimed. In this case, clear the data and recreate a snapshot. For details, see **How Can I Clear and Reclaim the Storage Space?**

----End

9.7.2.2 Creating a Manual Snapshot of a Schema

Overview

A schema snapshot is a backup of specific schemas in a GaussDB(DWS) cluster at a specific point in time. This section describes how to create a schema snapshot on the **Snapshots** page.

A manual fine-grained snapshot can be created at any time. It will be retained until it is deleted from the GaussDB(DWS) console.

NOTE

- If the current console does not support this feature, contact technical support.
- Manual schema snapshots can be backed up to OBS or NFS.
- Schema snapshots can be created only for clusters in **Available** or **Unbalanced** state.

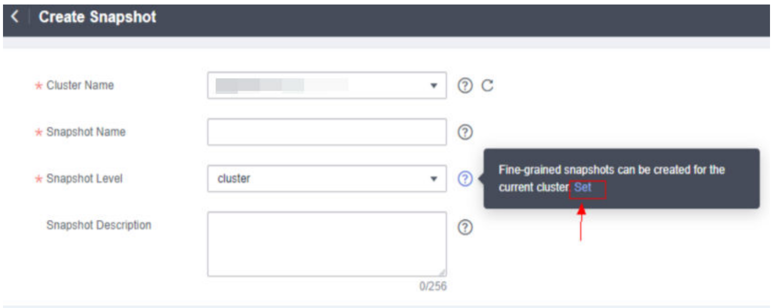
Prerequisites

Manually enable the fine-grained snapshot.

Step 1 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**.

Step 2 Click **Create Snapshot** in the upper right corner. Alternatively, choose **More** > **Create Snapshot** in the **Operation** column.

Step 3 Click ? next to **Snapshot Level** and click **Set**.



Create Snapshot

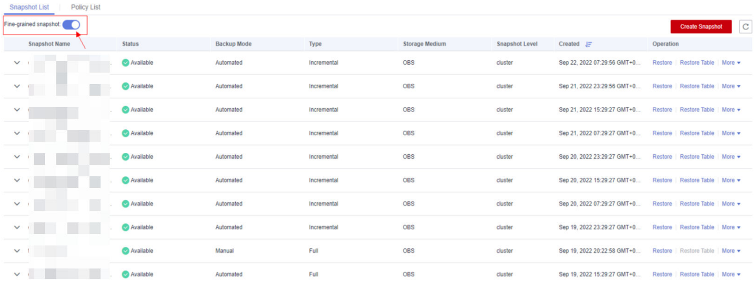
* Cluster Name	[redacted]
* Snapshot Name	
* Snapshot Level	cluster
Snapshot Description	0/256

Fine-grained snapshots can be created for the current cluster. Set

Step 4 On the **Snapshot List** page, toggle the fine-grained snapshot switch.

: enabled

: disabled



Snapshot List | Policy List

Fine-grained snapshot: ●

Create Snapshot

Snapshot Name	Status	Backup Mode	Type	Storage Medium	Snapshot Level	Created	Operation
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 22, 2022 07:29:56 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 23:29:56 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 15:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 07:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 23:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 15:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 07:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 19, 2022 23:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Manual	Full	OBS	cluster	Sep 19, 2022 20:22:58 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Full	OBS	cluster	Sep 19, 2022 15:29:27 GMT+0...	Restore | Restore Table | More

NOTE

- If the fine-grained snapshot is enabled, you can create snapshots for specific schemas.
- If the fine-grained snapshot is enabled, you can restore specific tables from automatic or manual snapshots.

----End

Impact on the System

If a snapshot is being created for a cluster, the cluster cannot be restarted, scaled, its password cannot be reset, and its configurations cannot be modified.

NOTE

To ensure the integrity of snapshot data, do not write data during snapshot creation.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**.

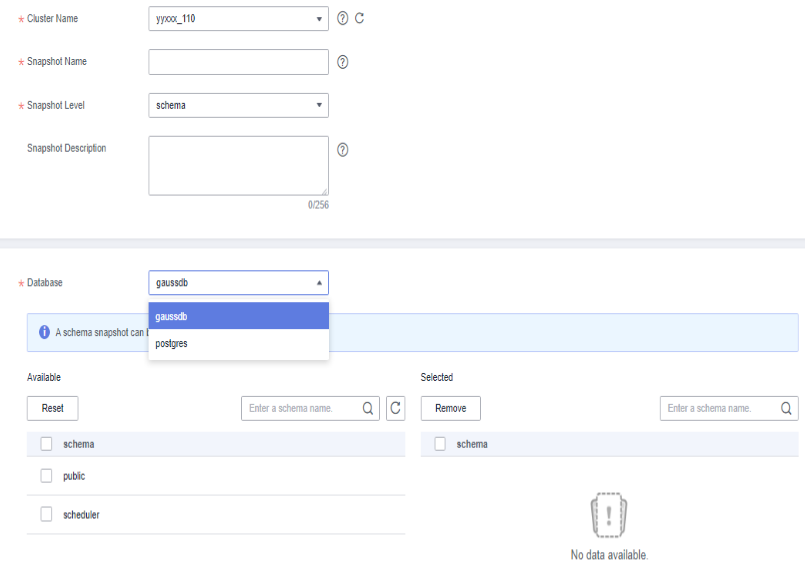
Step 3 Click **Create Snapshot** in the upper right corner. Alternatively, choose **More** > **Create Snapshot** in the **Operation** column.

Step 4 Configure the following snapshot information:

- **Cluster Name**: Select a GaussDB(DWS) cluster from the drop-down list. The drop-down list only displays clusters that are in the **Available** state.
- **Snapshot Name**: Enter a snapshot name. The snapshot name must be 4 to 64 characters in length and start with a letter. It is case-insensitive and contains only letters, digits, hyphens (-), and underscores (_).
- **Snapshot Level**: Select **schema**.
- **Snapshot Description**: Enter the snapshot information. This parameter is optional. Snapshot information contains 0 to 256 characters and does not support the following special characters: !<>'=&"

Step 5 Specify the snapshots to be backed up.

- Select a database from the **Database** drop-down list.
- In the schema list, select the schemas to be backed up. To search for a schema, enter its name in the search box in the upper right corner of the list, and click 🔍. Fuzzy search is supported.



* Cluster Name: yyxxx_110
* Snapshot Name:
* Snapshot Level: schema
Snapshot Description: 0/256

* Database: gaussdb

A schema snapshot can

NOTE

- Schemas in different databases cannot be backed up at a time.
- By default, a maximum of 50 schemas can be backed up at a time.

Step 6 Click **Create**.

The task status of the cluster for which you are creating a snapshot is **Creating snapshot**. The status of the snapshot that is being created is **Creating**. After the snapshot is created, its status becomes **Available**.

NOTE

If a snapshot is larger than the available storage space in the cluster, check whether there is data that has been marked as deleted but actually still exists in the cluster. In this case, delete such data and create a snapshot again. For details, see **How Can I Clear and Reclaim the Storage Space?**

----End

9.7.2.3 Deleting a Manual Snapshot

On the **Snapshot Management** page of the GaussDB(DWS) management console, you can delete an unwanted snapshot in the **Unavailable** state or delete an available snapshot to release the storage space.

CAUTION

Deleted snapshots cannot be recovered. Exercise caution when performing this operation.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**. All snapshots are displayed by default.

Step 3 In the **Operation** column of the snapshot that you want to delete, choose **More** > **Delete**.

NOTE

You can only delete snapshots that were manually created.

Step 4 If the information is correct, enter **DELETE** and click **OK** to delete the snapshot.

----End

9.7.3 Automated Snapshots

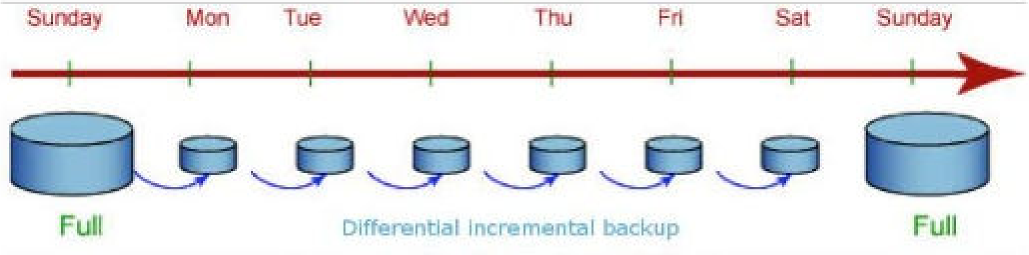
9.7.3.1 Automatic Snapshot Overview

Automated snapshots adopt differential incremental backups. The automated snapshot created for the first time is a full backup (base version), and then the system creates full backups at a specified interval. Incremental backups are generated between two full backups. The incremental backup records change based on the previous backup.

During snapshot restoration, GaussDB(DWS) uses all backups between the latest full backup and the current incremental backup to restore the cluster. Therefore, no data loss occurs.

If the retention period of an incremental snapshot exceeds the maximum retention period, GaussDB(DWS) does not delete the snapshot immediately. Instead, GaussDB(DWS) retains it until the next full backup is completed, when the deletion of the snapshot will not hinder incremental data backup and restoration.

Figure 9-27 Snapshot backup process



Automated snapshots are enabled by default when you create a cluster. If automated snapshots are enabled for a cluster, GaussDB(DWS) periodically takes snapshots of that cluster based on the time and interval you set, usually every eight hours. You can configure one or more automated snapshot policies for the cluster as required. If no full backup policy is configured, a full backup is performed every 15 incremental backups. For details, see **Configuring an Automated Snapshot Policy**.

The retention period of an automated snapshot can be set to 1 to 31 days. The default retention period is 7 days. The system deletes the snapshot at the end of the retention period. The retention period sets the duration for which users can access snapshots. If an incremental snapshot does not expire, both the incremental and full snapshots will be kept available instead of being deleted right away. Expired snapshots are hidden and can no longer be viewed by users. After all incremental snapshots expire, the hidden snapshots are physically deleted. If you want to keep an automated snapshot for a longer period, you can create a copy of it as a manual snapshot. The automated snapshot is retained until the end of the retention period, whereas the corresponding manual snapshot is retained until you manually delete it. For details about how to copy an automated snapshot, see **Copying Automated Snapshots**.

9.7.3.2 Configuring an Automated Snapshot Policy

You can select a snapshot type and set one or more automated snapshot policies for a cluster. After an automated snapshot policy is enabled, the system automatically creates snapshots based on the time, period, and snapshot type you configured.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 4 Click **Snapshots** and click **Policy List**. All policies of the current cluster are displayed on the **Policy List** page. Toggle on **Snapshot Policy**.

Step 5 (Optional) Click **Automated Snapshot** to enable the snapshot policy.

- indicates that the policy is enabled (default). The default retention period is seven days.
- indicates that the policy is disabled. Once disabled, snapshots will not be automatically created.

Step 6 After this function is enabled, you can set the retention mode and the backup device used by the current cluster for automated snapshots. For more information, see **Table 9-18**.

Table 9-18 Automated snapshot parameters

Parameter	Description
Retention Days	Retention days of the snapshots that are automatically created. The value ranges from 1 to 31 days. **NOTE** Snapshots that are automatically created cannot be deleted manually. The system automatically deletes these snapshots when their retention duration exceeds the threshold.

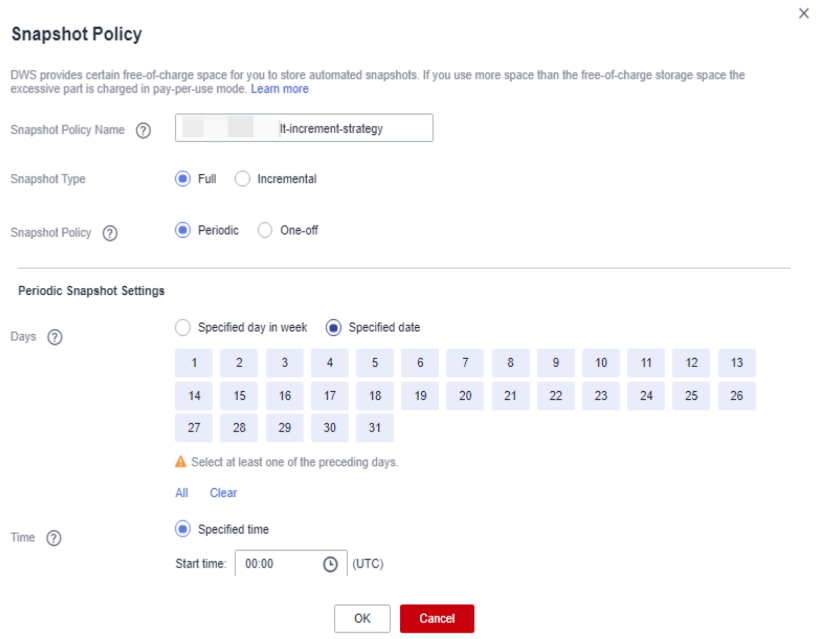
Parameter	Description
Backup Device	Select **OBS** or **NFS** from the drop-down list.
NFS Backup File System Address (NFS)	NFS shared IP address. Enter the IP address of the SFS shared path. After the mounting is successful, a mount directory is created in the **/var/chroot/nfsbackup** directory of the cluster instance by default.

Step 7 After automated snapshot is enabled, you can configure its parameters. For more information, see **Table 9-19**.

NOTE

The snapshot creation time is UTC, which may be different from your local time.

- If the snapshot type is set to **Full**, you can choose either **Periodic** or **One-off**.
  - **Periodic**: Specify the days for every week/month and the exact time on the days.



Snapshot Policy

DWS provides certain free-of-charge space for you to store automated snapshots. If you use more space than the free-of-charge storage space the excessive part is charged in pay-per-use mode. Learn more

Snapshot Policy Name: lt-increment-strategy

Snapshot Type: ● Full ○ Incremental

Snapshot Policy: ● Periodic ○ One-off

Periodic Snapshot Settings

Days: ○ Specified day in week ● Specified date

1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31

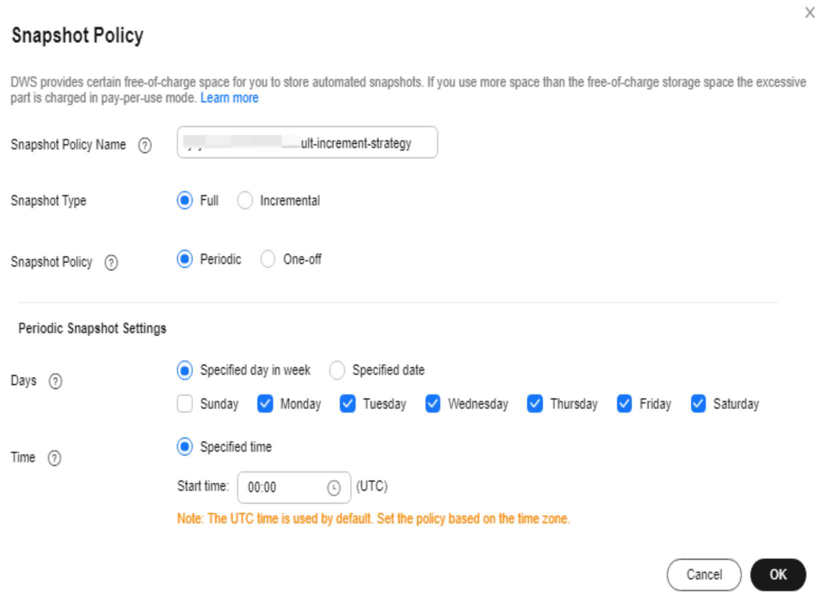
Select at least one of the preceding days.

All Clear

Time: ● Specified time

Start time: 00:00 (UTC)

Figure 9-28 Setting Snapshot Policy to Periodic



Snapshot Policy

DWS provides certain free-of-charge space for you to store automated snapshots. If you use more space than the free-of-charge storage space the excessive part is charged in pay-per-use mode. Learn more

Snapshot Policy Name: [redacted]ult-increment-strategy

Snapshot Type: ● Full ○ Incremental

Snapshot Policy: ● Periodic ○ One-off

Periodic Snapshot Settings

Days: ● Specified day in week ○ Specified date
□ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday

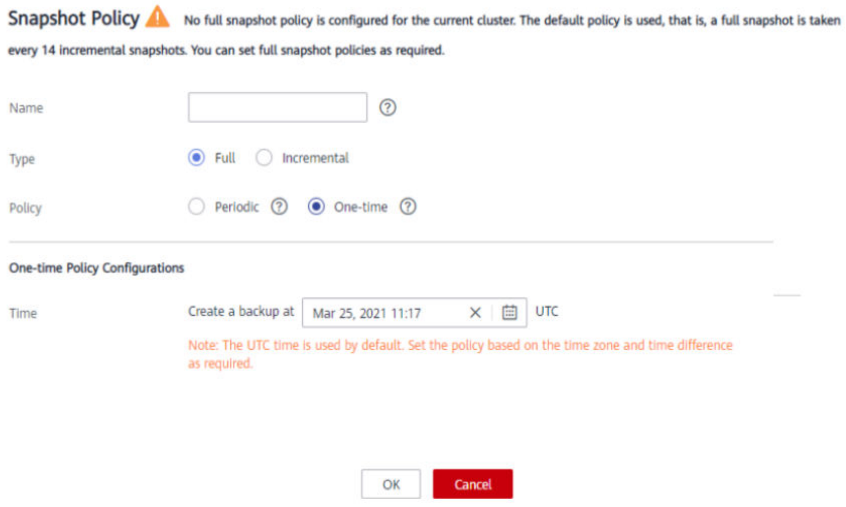
Time: ● Specified time
Start time: 00:00 (UTC)

Note: The UTC time is used by default. Set the policy based on the time zone.

WARNING

Choosing the days in red (29th/30th/31st) may skip some monthly backups. Policy and execution depend on the specific month and date you choose.

– **One-time**: Specify a day and the exact time on the day.



Snapshot Policy ⚠ No full snapshot policy is configured for the current cluster. The default policy is used, that is, a full snapshot is taken every 14 incremental snapshots. You can set full snapshot policies as required.

Name:

Type: ● Full ○ Incremental

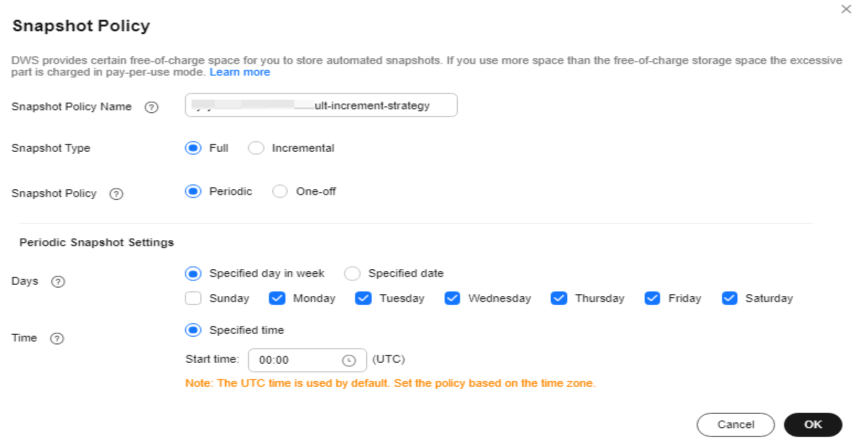
Policy: ○ Periodic ● One-time

One-time Policy Configurations

Time: Create a backup at Mar 25, 2021 11:17 UTC

Note: The UTC time is used by default. Set the policy based on the time zone and time difference as required.

Figure 9-29 Setting Snapshot Policy to One-off



Snapshot Policy

DWS provides certain free-of-charge space for you to store automated snapshots. If you use more space than the free-of-charge storage space the excessive part is charged in pay-per-use mode. Learn more

Snapshot Policy Name: [redacted]ult-increment-strategy

Snapshot Type: ● Full ○ Incremental

Snapshot Policy: ● Periodic ○ One-off

Periodic Snapshot Settings

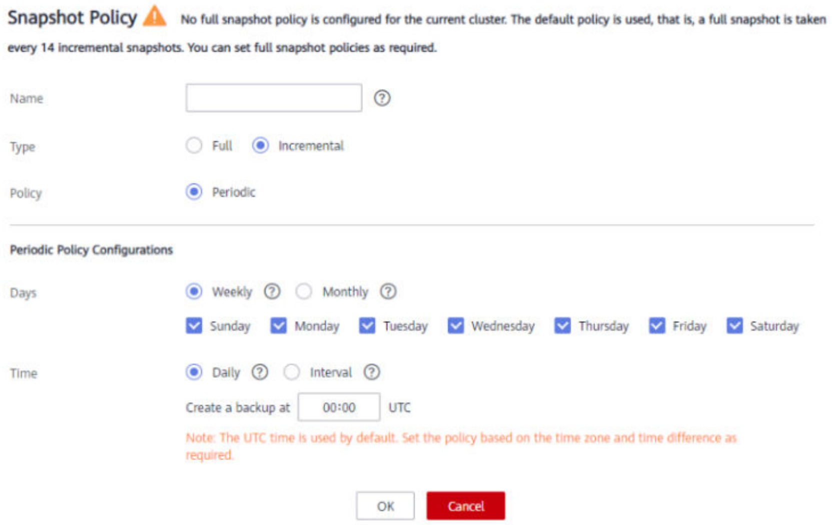
Days: ● Specified day in week ○ Specified date
□ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday

Time: ● Specified time
Start time: 00:00 (UTC)

Note: The UTC time is used by default. Set the policy based on the time zone.

- Incremental snapshots can be set only to **Periodic**.

  When configuring a periodic incremental snapshot policy, you can specify the days for every week/month and the exact time on the days. You can also specify the start time and interval for the snapshots.



Snapshot Policy ⚠ No full snapshot policy is configured for the current cluster. The default policy is used, that is, a full snapshot is taken every 14 incremental snapshots. You can set full snapshot policies as required.

Name:

Type: ○ Full ● Incremental

Policy: ● Periodic

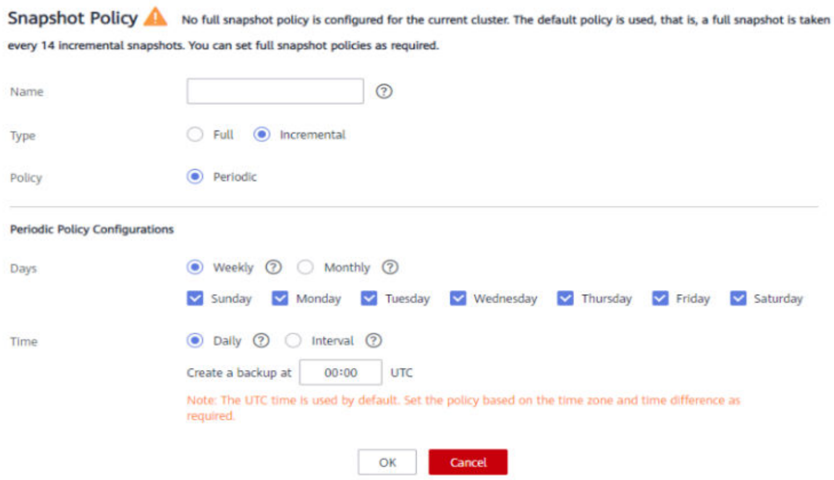
Periodic Policy Configurations

Days: ● Weekly ○ Monthly
☑ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday

Time: ● Daily ○ Interval
Create a backup at 00:00 UTC

Note: The UTC time is used by default. Set the policy based on the time zone and time difference as required.

Figure 9-30 Setting Type to Incremental



Snapshot Policy ⚠ No full snapshot policy is configured for the current cluster. The default policy is used, that is, a full snapshot is taken every 14 incremental snapshots. You can set full snapshot policies as required.

Name:

Type: ○ Full ● Incremental

Policy: ● Periodic

Periodic Policy Configurations

Days: ● Weekly ○ Monthly
☑ Sunday ☑ Monday ☑ Tuesday ☑ Wednesday ☑ Thursday ☑ Friday ☑ Saturday

Time: ● Daily ○ Interval
Create a backup at 00:00 UTC

Note: The UTC time is used by default. Set the policy based on the time zone and time difference as required.

Table 9-19 Snapshot policy parameters

Parameter	Description
Name	The policy name must be unique, consist of 4 to 92 characters, and start with a letter. It is case-insensitive and can contain only letters, digits, hyphens (-), and underscores (_).
Type	You can choose either full or incremental snapshots. **NOTE** ● A full snapshot is created after every fifteen incremental snapshots are created. ● Incremental snapshot restoration is based on full snapshots. Incremental snapshots are used to restore all data to the time point when they were created. ● An incremental snapshot records the changes made after the previous snapshot was created. A full snapshot backs up the data of an entire cluster. It takes a short time to create an incremental snapshot, and a long time to create a full snapshot. When restoring a snapshot to a new cluster, GaussDB(DWS) uses all snapshots between the latest full backup and the current snapshot.
Policy	You can choose either periodic or one-time snapshots. **NOTE** **One-time** can be selected only for full snapshots.
One-time	You can create a full snapshot at a specified time in the future. The UTC time is used.
Periodic Policy Configurations	You can create automated snapshots on a daily, weekly, or monthly basis: ● **Days**: Specify days for every week or every month. **Weekly** and **Monthly** cannot be selected at the same time. For **Monthly**, the specified days are applicable only to months that contain the dates. For example, if you select **29**, no automated snapshot will be created on February, 2022. ● **Time**: Specify the exact time on the selected days. For incremental snapshots, you can specify the start time and interval. The interval can be 4 to 24 hours, indicating that a snapshot is created at an interval of 4 to 24 hours. **NOTICE** Incremental snapshots can be set only to **Periodic**, as shown in the first figure below.

Step 8 Click **OK**.

NOTE

A maximum of three snapshot policies can be set for a cluster.

Step 9 (Optional) To modify an automated snapshot policy, click **Modify** in the **Operation** column.

Step 10 (Optional) To preview a policy, click **Preview Policy**. The next seven snapshots of the cluster will be displayed. If no full snapshot policy is configured for the cluster, the default policy is used, that is, a full snapshot is taken after every 15 incremental snapshots.

NOTICE

Implementation of the same policy varies according to operations in the cluster. For example:

- The policy preview time is for your reference only. The cluster triggers a snapshot within one hour before and after the preset time.
- The next automated snapshots after cluster scale-out, upgrade, resize, and media modification are full snapshots by default.
- If a periodic policy is used for a cluster, no automatic backup is allowed within 4 hours after the last automated snapshot is complete.
- If the time for triggering snapshots of multiple policies conflicts, the priorities of the policies are as follows: one-time > periodic > full > incremental.
- You can use any backup, full or incremental, to restore the full data of a resource.

----End

9.7.3.3 Copying Automated Snapshots

This section describes how to copy snapshots that are automatically created for long-term retention.

Copying an Automated Snapshot

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **Snapshot**.

All snapshots are displayed by default. You can copy the snapshots that were automatically created.

Step 3 In the **Operation** column of the snapshot that you want to copy, choose **More** > **Copy**.

- **New Snapshot Name**: Enter a new snapshot name.

  The snapshot name must be 4 to 64 characters in length and start with a letter. It is case-insensitive and contains only letters, digits, hyphens (-), and underscores (_).

- **Snapshot Description**: Enter the snapshot information.

  This parameter is optional. Snapshot information contains 0 to 256 characters and does not support the following special characters: !<>'=&"

Step 4 Click **OK**. The system starts to copy the snapshot for the cluster.

The system displays a message indicating that the snapshot is successfully copied and delivered. After the snapshot is copied, the status of the copied snapshot is **Available**.

NOTE

If the snapshot size is much greater than that of the data stored in the cluster, the data is possibly labeled with a deletion tag, but is not cleared and reclaimed. In this case, clear the data and recreate a snapshot. For details, see **How Can I Clear and Reclaim the Storage Space?**

----End

9.7.3.4 Deleting an Automated Snapshot

Only GaussDB(DWS) can delete automated snapshots; you cannot delete them manually.

GaussDB(DWS) deletes an automated snapshot if:

- The retention period of the snapshot ends.
- The cluster is deleted.

⚠ CAUTION

To help users restore a cluster deleted by mistake, GaussDB(DWS) provides the following policies (supported only in 8.2.0 and later) for cluster snapshots:

- If the latest snapshot is an automated snapshot, it will be retained for one day.
- If the latest snapshot is a manual snapshot, the automated snapshot of the cluster will be deleted.

9.7.4 Viewing Snapshot Information

This section describes how to view snapshot information on the **Snapshots** page.

Viewing Snapshot Information

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **Snapshots**.

In the snapshot list, all snapshots are displayed by default.

Step 3 You can view the **Snapshot Name**, **Snapshot Status**, **Cluster Name**, **Backup Mode**, **Snapshot Type**, **Storage Medium**, **Snapshot Level**, and creation time of snapshots.

You can also enter a snapshot name or cluster name in the upper right corner of the snapshot list and click 🔍 to search for the specified snapshot. GaussDB(DWS) supports fuzzy search.

Table 9-20 describes snapshot status.

Table 9-20 Snapshot status

Status	Description
Available	Indicates that the existing snapshot works properly.
Creating	Indicates that a snapshot is being created.
Unavailable	Indicates that the existing snapshot cannot provide services.

Table 9-21 lists the backup modes.

Table 9-21 Backup modes

Type	Description
Manual	Indicates the snapshot that you manually create through the GaussDB(DWS) management console or using APIs. You can delete the snapshots that are manually created.
Automated	Indicates the snapshot that is automatically created after the automated snapshot backup policy is enabled. You cannot delete the snapshots that are automatically created. The system automatically deletes the snapshots whose retention duration expires.

Table 9-22 describes the snapshot types.

Table 9-22 Type

Type	Description
Full	The snapshot is a full backup.
Incremental	The snapshot is an incremental backup.

Table 9-23 describes the snapshot media.

Table 9-23 Storage media

Storage Medium	Description
OBS	The created snapshot is an OBS snapshot and the backup data is stored on the OBS server.
NFS	The created snapshot is an NFS snapshot and the backup data is stored on the NFS server.

Table 9-24 describes the snapshot levels.

Table 9-24 Snapshot levels

Snapshot Level	Description
cluster	A backup of all the configurations and service data of a cluster at a specific point in time.
schema	A backup of all the service data of a schema at a specific point in time.

----End

Querying Snapshot Information by Table Name

Prerequisites

Only the snapshots that are created after the fine-grained snapshot function is enabled support fine-grained search.

NOTE

Only cluster versions 8.2.1.300 and later support snapshot query by table name.

Procedure

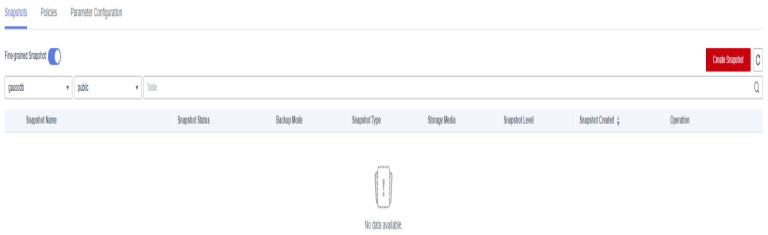
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 4 In the navigation pane, choose **Snapshots** and enable the fine-grained snapshot function. Click the advanced search button on the right.

You can set a triplet consisting of a database, a schema, and a table in a fine-grained query on the snapshot information in a specified database schema table. For details about the snapshot information, see **Step 3**.



Snapshot Name	Snapshot Status	Backup Mode	Snapshot Type	Storage Media	Snapshot Level	Snapshot Created	Operation

Table 9-25 Triplet description

Tuple Name	Description
database	Database name.

Tuple Name	Description
schema	Schema name.
table	Table name.

----End

9.7.5 Restoration Using a Snapshot

9.7.5.1 Constraints on Restoring a Snapshot

Cluster-Level Snapshot Restoration

Cluster-level restoration consists of two steps:

1. Data restoration: Restores data in the backup set to the data directory of each primary DN/CN instance in parallel.
2. Rebuilding the standby DN: After the primary DN is restored, standby DNs are rebuilt with full data in parallel.

📖 NOTE

- The restoration process takes 1.5 to 2 times longer than the backup process.
- After a cluster-level restoration, the parameters will be identical to those during the backup. The new cluster must have the same specifications as the original cluster. If any changes were made to the specifications of the original cluster, the new cluster must still match the specifications prior to the changes. If the specifications of the new cluster are smaller, the restoration may fail.

9.7.5.2 Restoring a Snapshot to a New Cluster

Scenario

This section describes how to restore a snapshot to a new cluster when you want to check point-in-time snapshot data of the cluster.

When a snapshot is restored to a new cluster, the restoration time is determined by the amount of data backed up by the snapshot. If a snapshot contains a large amount of data, the restoration will be slow. A small snapshot can be quickly restored.

Automatic snapshots are incremental backups. When restoring a snapshot to a new cluster, GaussDB(DWS) uses all snapshots between the latest full backup and the current snapshot. You can set the backup frequency. If snapshots are backed up only once a week, the backup will be slow if the incremental data volume is large. You are advised to increase the backup frequency.

NOTICE

- By default, the new cluster created during restoration has the same specifications and node quantity as the original cluster.
- Restoring data to a new cluster does not affect the services running in the original cluster.
- Fine-grained restoration does not support tables in absolute or relative tablespace.
- Logical clusters and resource pools cannot be restored to a new cluster.

Prerequisites

- The resources required for restoring data to a new cluster do not exceed your available resource quota.
- The snapshot is in the **Available** state.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**. All snapshots are displayed by default.

Step 3 In the **Operation** column of a snapshot, click **Restore**.

Step 4 On the **Restore Snapshot** page, configure the parameters of the new cluster, as shown in the following figure. **Extreme SSD** and **Extreme SSD V2** can only be selected for clusters that use ECS for computing and EVS for storage.

- Restore to a single-AZ cluster.
- Restore to a multi-AZ cluster.

NOTE

- Only clusters later than 8.2.0.100 can be restored to a multi-AZ cluster.
- This feature is only available for storage-compute coupled clusters.
- The number of AZs in the current region is greater than or equal to 3.
- The number of nodes and CNs must be a multiple of 3.
- DNs in the multi-AZ cluster must be less than or equal to 2.

You can modify cluster parameters. For details, see **Table 9-26**. By default, other parameters are the same as those in the snapshot. For details, see **Table 9-19**.

Table 9-26 Parameters for the new cluster

Category	Operation
Basic settings	Region, AZ, node flavor, cluster name, database port, VPC, subnet, security group, public access, and enterprise project

Category	Operation
Advanced settings	If **Custom** is selected, configure the following parameters: ● Backup devices: Select OBS or NFS from the drop-down list. ● Label: a key-value pair used to identify a cluster. For details about labels, see **Overview**.

Step 5 Click **Restore** to go to the confirmation page.

Step 6 Click **Submit** to restore the snapshot to the new cluster.

When the status of the new cluster changes to **Available**, the snapshot is restored.

After the snapshot is restored, the private network address and EIP (if **EIP** is set to **Buy now**) are automatically assigned.

NOTE

If the number of requested nodes, vCPU (cores), or memory (GB) exceed the user's remaining quota, a warning dialog box is displayed, indicating that the quota is insufficient and displaying the detailed remaining quota and the current quota application. You can click **Increase quota** in the warning dialog box to submit a service ticket and apply for higher node quota. Once approved, we will update your resource quota accordingly and send you a notification. For details about quotas operations, see **Quotas**.

----End

9.7.5.3 Restoring a Snapshot to the Original Cluster

Scenario

You can use a snapshot to restore data to the original cluster. This function is used when a cluster is faulty or data needs to be rolled back to a specified snapshot version.

NOTICE

- This function is supported only by clusters of version 8.1.3.200 or later.
- Snapshots whose backup device is OBS can be backed up.
- Only a snapshot in the **Available** state can be used for restoration.
- Logical clusters and resource pools cannot be restored to the current cluster.

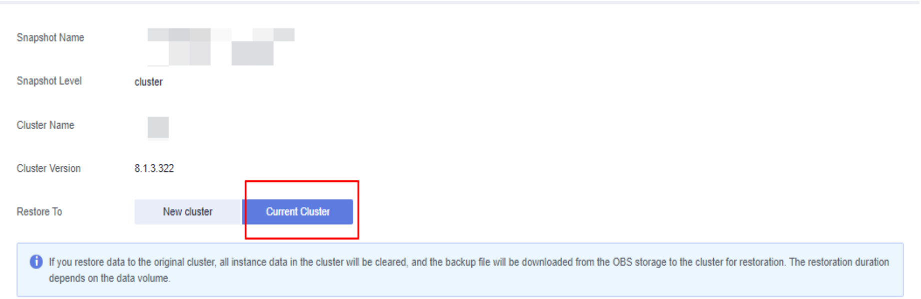
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the Cluster Information page. Then, click **Snapshots**. All snapshots are displayed by default.

Step 3 In the **Operation** column of a snapshot, click **Restore**.

Step 4 Restore the snapshot to the current cluster.



Snapshot Name [redacted]

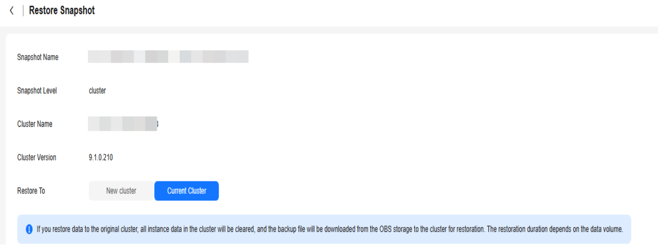
Snapshot Level cluster

Cluster Name [redacted]

Cluster Version 8.1.3.322

Restore To New cluster | Current Cluster

If you restore data to the original cluster, all instance data in the cluster will be cleared, and the backup file will be downloaded from the OBS storage to the cluster for restoration. The restoration duration depends on the data volume.



Restore Snapshot

Snapshot Name [redacted]

Snapshot Level cluster

Cluster Name [redacted]

Cluster Version 9.1.0.210

Restore To New cluster | Current Cluster

If you restore data to the original cluster, all instance data in the cluster will be cleared, and the backup file will be downloaded from the OBS storage to the cluster for restoration. The restoration duration depends on the data volume.

NOTE

If you use a snapshot to restore data to the original cluster, the cluster will be unavailable during the restoration.

----End

9.7.5.4 Restoring a Table to the Original Cluster

Scenario

You can create a table from a cluster or schema snapshot to the original cluster if the table was modified or deleted by mistake.

NOTICE

- If the current console does not support this feature, contact technical support.
- Only the tables stored in OBS can be used to restore data to the original cluster.
- Currently, only cluster- and schema-level snapshots can be used for such restoration.
- Restoration can be performed only if the snapshot and the cluster are both in the **Available** state.
- A table in a read-only cluster cannot be restored.
- Fine-grained restoration does not support tables in an absolute or relative tablespace.

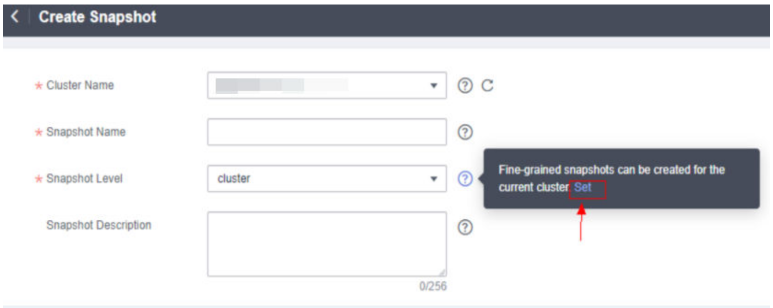
Prerequisites

Manually enable the fine-grained snapshot.

Step 1 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**.

Step 2 Click **Create Snapshot** in the upper right corner. Alternatively, choose **More** > **Create Snapshot** in the **Operation** column.

Step 3 Click ? next to **Snapshot Level** and click **Set**.



Create Snapshot

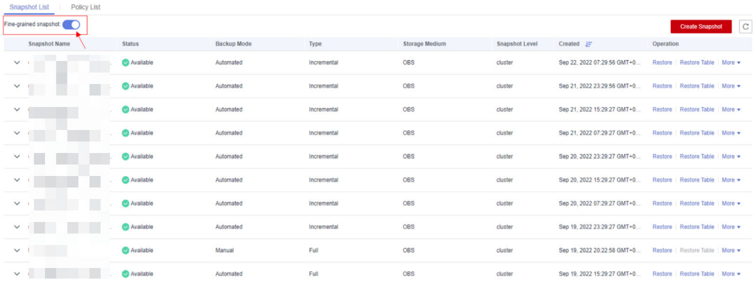
* Cluster Name	[redacted]
* Snapshot Name	
* Snapshot Level	cluster
Snapshot Description	0/256

Fine-grained snapshots can be created for the current cluster. Set

Step 4 On the **Snapshot List** page, toggle the fine-grained snapshot switch.

: enabled

: disabled



Snapshot Name	Status	Backup Mode	Type	Storage Medium	Snapshot Level	Created	Operation
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 22, 2022 07:29:56 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 23:29:56 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 15:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 07:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 23:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 15:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 07:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 19, 2022 23:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Manual	Full	OBS	cluster	Sep 19, 2022 20:22:58 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Full	OBS	cluster	Sep 19, 2022 15:29:27 GMT+0...	Restore | Restore Table | More

NOTE

- If the fine-grained snapshot is enabled, you can create snapshots for specific schemas.
- If the fine-grained snapshot is enabled, you can restore specific tables from automatic or manual snapshots.

----End

Procedure

Step 1 Log in to the GaussDB(DWS) console.

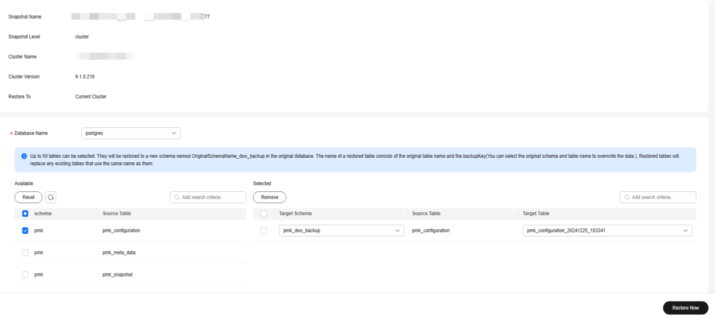
Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**. All snapshots are displayed by default.

Step 3 Locate the row that contains the target snapshot, click **Restore**.

Step 4 On the **Restore Table** page, configure the following parameters:

- **Database:** To restore a cluster snapshot, select a database. To restore a schema snapshot, select the database specified during backup. For details, see **Creating a Manual Snapshot of a Cluster** and **Creating a Manual Snapshot of a Schema**.
- **Source Schema**: Specify the schema of the table to be restored.
- **Source Table**: Specify the name of the table to be restored.
- **Destination Schema**: Specify the schema where the table is to be restored to.
- **Destination Table**: Specify the name of the new table.

Figure 9-31 Table-level restoration



Snapshot Name	[redacted]
Snapshot Level	cluster
Cluster Name	[redacted]
Cluster Version	9.1.0.210
Restore To	Current Cluster

* Database Name: postgres

Up to 50 tables can be selected. They will be restored to a new schema named OriginalSchemaName_dws_backup in the original database. The name of a restored table consists of the original table name and the backupKey(You can select the original schema and table name to overwrite the data.). Restored tables will replace any existing tables that use the same name as them.

Available

■ schema	Source Table
☑ pmk	pmk_configuration
□ pmk	pmk_meta_data
□ pmk	pmk_snapshot

Selected

□ Target Schema	Source Table	Target Table
□ pmk_dws_backup	pmk_configuration	pmk_configuration_20241225_103341

CAUTION

- A table name must meet the following GaussDB(DWS) database naming constraints: The table name is case sensitive can contain up to 63 characters. It must start with a letter or underscore (_). Letters, digits underscores (_) are allowed.
- The source table to be restored must be a table in the backup set, or the restoration will fail.
- If the target table already exists in the database, this table will be overwritten during restoration. Check the table name before starting restoration.

Step 5 Confirm the information and click **Restore**.

----End

9.7.5.5 Restoring a Table or Multiple Tables to a New Cluster

Scenario

You can create a table from a cluster or schema snapshot to the original cluster. In case of accidental deletion or operation on data in a table during service operations, you can use this function to find the latest snapshot that contains the table data and restore it to a new cluster. Compare the data between the old and new clusters without affecting the original table data, and then restore the data as needed.

NOTICE

- This function is supported only by clusters of version 9.1.0 or later. The system supports OBS.
- You can restore fine-grained snapshots of clusters from an earlier version to a new cluster of version 9.1.0, even if the versions are different.
- You can restore the fine-grained snapshot of the 9.1.0 cluster to a new heterogeneous cluster of version 9.1.0, even if the number of nodes and specifications of the old and new clusters are different.
- Only fine-grained single-table or multi-table snapshots can be restored to a new cluster.

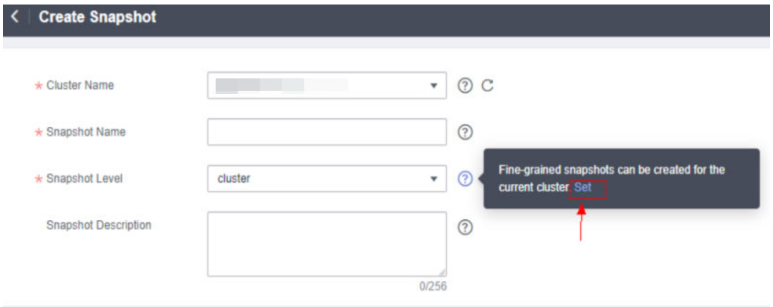
Prerequisites

Manually enable the fine-grained snapshot.

Step 1 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the Cluster Information page. Then, click **Snapshots**.

Step 2 Click **Create Snapshot** in the upper right corner. Alternatively, choose **More** > **Create Snapshot** in the **Operation** column.

Step 3 Click ⓘ next to **Snapshot Level** and click **Set**.



* Cluster Name: [redacted]

* Snapshot Name:

* Snapshot Level: cluster

Snapshot Description:

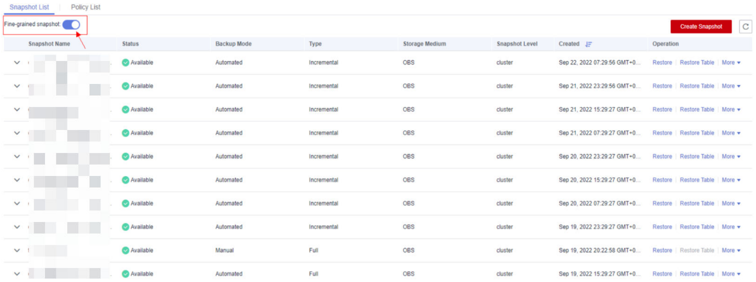
0/256

Fine-grained snapshots can be created for the current cluster. Set

Step 4 On the **Snapshot List** page, toggle the fine-grained snapshot switch.

: enabled

: disabled



Fine-grained snapshot: on

Snapshot Name	Status	Backup Mode	Type	Storage Medium	Snapshot Level	Created	Operation
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 22, 2022 07:29:56 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 23:29:56 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 15:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 21, 2022 07:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 23:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 15:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 20, 2022 07:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Incremental	OBS	cluster	Sep 19, 2022 23:29:27 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Manual	Full	OBS	cluster	Sep 19, 2022 20:22:58 GMT+0...	Restore | Restore Table | More
[redacted]	Available	Automated	Full	OBS	cluster	Sep 19, 2022 15:29:27 GMT+0...	Restore | Restore Table | More

NOTE

- If the fine-grained snapshot is enabled, you can create snapshots for specific schemas.
- If the fine-grained snapshot is enabled, you can restore specific tables from automatic or manual snapshots.

----End

Procedure

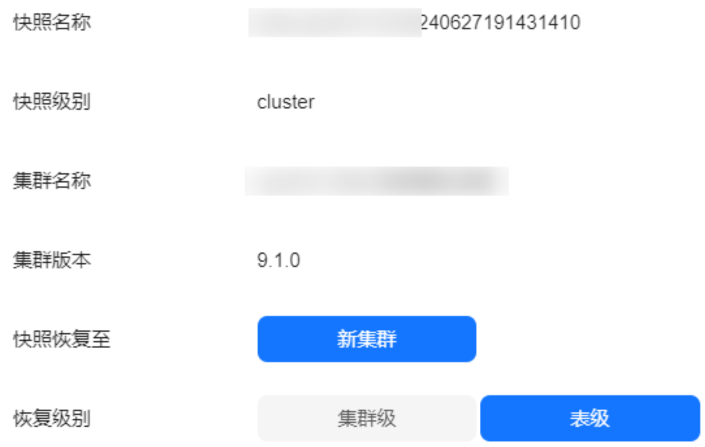
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**. All snapshots are displayed by default.

Step 3 In the **Operation** column of a snapshot, click **Restore**.

Step 4 Set the recovery level to the table level.

Figure 9-32 Table-level restoration



快照名称	[redacted]240627191431410
快照级别	cluster
集群名称	[redacted]
集群版本	9.1.0
快照恢复至	新集群
恢复级别	集群级 表级

Step 5 Select the basic information about the new cluster to be restored. For details, see **Creating a GaussDB(DWS) Storage-Compute Coupled Cluster**.

NOTE

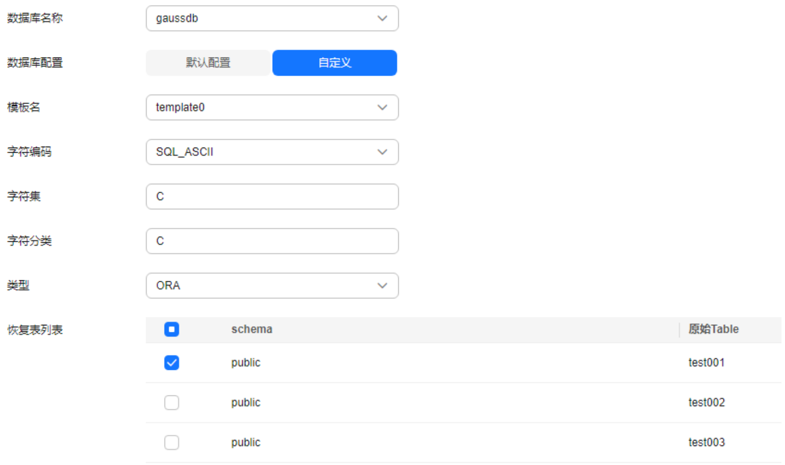
- If fine-grained heterogeneous recovery is available, you can select different node specifications and quantities for the new cluster, regardless of whether they match those of the original cluster.
- You need a cluster version of 9.1.0 or later to restore one or more tables to a new cluster.

Step 6 Select a single table or multiple tables. Select a database name from the drop-down list. If you select custom database configuration, you can adjust the following configuration parameters. If you select the default configuration, the parameters will use their default values. Once you've finished configuring, choose one or more tables in the table list to restore.

NOTE

When you restore data to a new cluster, a new database is created. If the configuration of the new database is not the same as the snapshot database, the restoration process may fail. Before restoring, make sure to review the configuration of the original database. If it differs from the default configuration, adjust it accordingly.

Figure 9-33 Custom database configuration



数据库名称: gaussdb

数据库配置: 默认配置 | 自定义

模板名: template0

字符编码: SQL_ASCII

字符集: C

字符分类: C

类型: ORA

恢复表列表:

■	schema	原始Table
☑	public	test001
□	public	test002
□	public	test003

Table 9-27 Custom database parameters

Parameter	Description	Value Range	Default Value
Template Name	Name of the template from which the database is created. GaussDB(DWS) creates a database by copying a database template. GaussDB(DWS) has two initial template databases **template0** and **template1** and a default user database **gaussdb**.	Names of existing databases, **template0**, and **template1**	template0

Parameter	Description	Value Range	Default Value
Character Encoding	● Encoding format used by the new database. The value can be a string (for example, **SQL_ASCII**) or an integer. ● By default, the encoding format of the template database is used. The encoding formats of the template databases **template0** and **template1** vary based on OS environments by default. – The **template1** database does not allow encoding customization. To specify encoding for a database when creating it, use **template0**. – To specify encoding, set **template** to **template0**.	Value range: **GBK**, **UTF8**, **Latin1**, and **SQL_ASCII**	SQL_ASCII
Character Set Support	Character set used by the new database. For example, this parameter can be set using **lc_collate = 'zh_CN.gbk'**. The use of this parameter affects the sort order applied to strings, for example, in queries with **ORDER BY**, as well as the order used in indexes on text columns. The default is to use the collation order of the template database.	Valid collation order	C
Character Classification	Character classification to use in the new database. For example, this parameter can be set using **lc_ctype = 'zh_CN.gbk'**. The use of this parameter affects the categorization of characters, for example, lower, upper and digit. The default is to use the character classification of the template database.	Valid character classification	C
Type	Compatible database type.	ORA, TD, and MySQL	ORA

Step 7 Click **Next: Confirm**.

Step 8 Confirm the information and click **Restore**.

----End

9.7.6 Configuring a Snapshot

You can configure the parameters for creating and restoring a snapshot.

NOTE

- This feature applies only to clusters of 8.2.0 or later. (For clusters of versions earlier than 8.2.0, only some parameters can be configured.)
- The parameters take effect on all the snapshot creation and restoration tasks.

Procedure

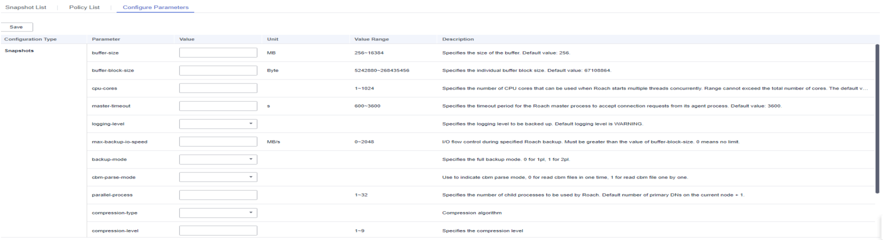
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 4 Click the **Snapshots** tab page and click **Configure Parameters**. All the configurable parameters of the current cluster will be displayed.

Step 5 Configure parameters as required. For details, see **Table 9-28**.



Configuration Type	Parameter	Value	Unit	Value Range	Description
Snapshots	buffer-size		MB	256~16384	Specifies the size of the buffer. Default value: 256.
	buffer-block-size		Byte	5242880~268435456	Specifies the individual buffer block size. Default value: 67108864.
	cpu-cores			1~1024	Specifies the number of CPU cores that can be used when Roach starts multiple threads concurrently. Range cannot exceed the total number of cores. The default v...
	master-timeout		s	600~3600	Specifies the timeout period for the Roach master process to accept connection requests from its agent process. Default value: 3600.
	logging-level				Specifies the logging level to be backed up. Default logging level is WARNING.
	max-backup-io-speed		MB/s	0~2048	I/O flow control during specified Roach backup. Must be greater than the value of buffer-block-size. 0 means no limit.
	backup-mode				Specifies the full backup mode. 0 for 1pl, 1 for 2pl.
	cbm-parse-mode				Use to indicate cbm parse mode, 0 for read cbm files in one time, 1 for read cbm file one by one.
	parallel-process			1~32	Specifies the number of child processes to be used by Roach. Default number of primary DNs on the current node + 1.
	compression-type				Compression algorithm
	compression-level			1~9	Specifies the compression level

Step 6 Click **Save**.

----End

Snapshot parameters

Table 9-28 Snapshot information

Parameter	Type	Description	Default Value
parallel-process	Backup parameter	Number of concurrent processes on each node during Roach backup. **NOTE** This parameter can be configured for clusters earlier than 8.2.0.	The value is the number of DNs on the current node.

Parameter	Type	Description	Default Value
compression-type	Backup parameter	Compression algorithm. ● zlib ● LZ4 **NOTE** This parameter can be configured for clusters earlier than 8.2.0.	LZ4
compression-level	Backup parameter	Compression level. The value range is 0 to 9. ● **0**: fast backup and no compression ● **9**: slow backup and maximum compression **NOTE** This parameter can be configured for clusters earlier than 8.2.0.	6
buffer-size	Backup parameter	Buffer size of the Roach upload media. The value range is 256 to 16,384, in MB.	256
buffer-block-size	Backup parameter	Data block size of the data file to be read by Roach. The value range is 5,242,880 to 268,435,456, in bytes.	67108864
cpu-cores	Backup parameter	Number of CPU cores that can be used when Roach starts multiple threads concurrently	1/2 of the total number of logical CPU cores on the node
master-timeout	Backup parameter	Timeout period for the communication between the Roach master and agent nodes. The value range is 600 to 3600, in seconds.	3600
max-backup-io-speed	Backup parameter	I/O flow control during Roach backup. The value range is 0 to 2048, in MB/s. The value must be greater than the value of **buffer-block-size**. The value **0** indicates no limit.	0

Parameter	Type	Description	Default Value
backup-mode	Backup parameter	Full backup mode. - **0**: phase-1 backup - **1**: phase-2 backup	0
cbm-parse-mode	Backup parameter	Incremental backup mode. - **0**: one-time CBM scan (high memory usage and high performance) - **1**: multiple CBM scans (stable memory usage and low performance)	0
parallel-process	Restoration parameter	Number of concurrent processes on each node during Roach backup. By default, the value is the number of primary DNs on the current node plus 1.	1
cpu-cores	Restoration parameter	Number of CPU cores that can be used when Roach starts multiple threads concurrently	The default value is 1/2 of the number of CPU cores.
logging-level	Restoration parameter	Log levels: - **FATAL**: Unrecoverable faults that cause the system suspension. This is the most severe level. - **ERROR**: Major errors. - **WARNING**: Exceptions. In this case, the system may continue to process tasks. - **INFO**: Notes. - **DEBUG**: Debugging details. - **DEBUG2**: Detailed debugging information, which is generally not displayed. This is the least severe level.	INFO

9.7.7 Stopping Snapshot Creation

You can stop snapshot creation on the **Snapshots** page.

NOTE

- This feature is supported only in version 8.1.3.200 and later.
- If the snapshot is ready to complete, the command for stopping the snapshot will not take effect and the snapshot will end normally.

Precautions

Only the snapshots in the **Creating** state can be stopped. A snapshot creation task that just started or is about to complete cannot be stopped.

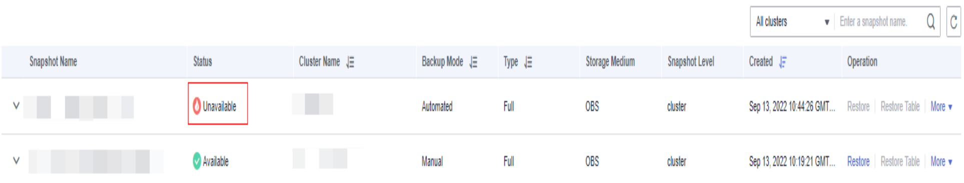
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Management** > **Snapshots**. Alternatively, in the cluster list, click the name of the target cluster to switch to the **Cluster Information** page. Then, click **Snapshots**. All snapshots are displayed by default.

Step 3 In the **Operation** column of a snapshot that is being created, and click **Cancel Creation**.

Step 4 In the dialog box that is displayed, click **Yes** to stop the snapshot. The snapshot state will change to **Unavailable**.



All clusters ▾ | Enter a snapshot name.

Snapshot Name	Status	Cluster Name	Backup Mode	Type	Storage Medium	Snapshot Level	Created	Operation
[redacted]	Unavailable	[redacted]	Automated	Full	OBS	cluster	Sep 13, 2022 10:44:26 GMT...	Restore | Restore Table | More ▾
[redacted]	Available	[redacted]	Manual	Full	OBS	cluster	Sep 13, 2022 10:19:21 GMT...	Restore | Restore Table | More ▾

----End

9.8 Scaling GaussDB(DWS) Cluster Nodes

9.8.1 Viewing Inspection Results

Context

GaussDB(DWS) allows you to inspect the cluster before making any changes like scaling, changing specifications, or upgrading. Simply click **Inspect** on the relevant page, and the system will check if the cluster's health status and metrics meet the requirements for the change. Once the inspection is passed, you can proceed with the change. If the inspection fails, you can view the inspection details page to see which items did not pass the inspection. From there, you can handle the inspection items based on the details provided. For details about the inspection standards, see **Table 9-29**.

NOTE

- This feature is supported only in cluster version 8.1.1 or later.
- If you cannot handle the failed inspection items, contact technical support engineers.

Precautions

- The inspection plug-in 8.3.1.100 or later has been installed in the cluster.
- The inspection result is valid for 24 hours, during which you can make the change operation. Once the 24-hour validity period expires, you will need to perform the inspection again.
- If the cluster has not been inspected within 24 hours before the change, inspect it before making any changes like scaling, changing specifications, or upgrading. Ensure that the inspection is passed before proceeding with the change.

Viewing Inspection Details

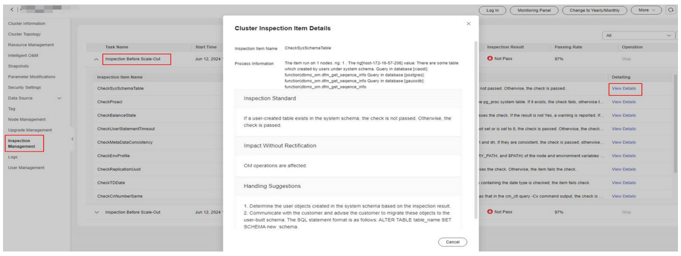
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the cluster list, click the name of the target cluster.

Step 3 On the cluster details page, click **Inspection Management**.

Step 4 Click the drop-down button next to its name to check the inspection status, execution progress, inspection result, and pass rate. For more details about these inspection items, click **View Details** in the task's row.

Figure 9-34 Viewing inspection details



Cluster Inspection Item Details

Inspection Item Name: CheckSysSchemaTable

Process Information: The item run on 1 nodes. ng: 1 . The ng[host-172-16-57-206] value: There are some table which created by users under system schema. Query in database [xiaodi]: function|dbms_om.dfm_get_seqence_info Query in database [postgres]: function|dbms_om.dfm_get_seqence_info Query in database [gaussdb]: function|dbms_om.dfm_get_seqence_info

Inspection Standard

If a user-created table exists in the system schema, the check is not passed. Otherwise, the check is passed.

Impact Without Rectification

OM operations are affected.

Handling Suggestions

1. Determine the user objects created in the system schema based on the inspection result.
2. Communicate with the customer and advise the customer to migrate these objects to the user-built schema. The SQL statement format is as follows: ALTER TABLE table_name SET SCHEMA new_schema.

NOTE

After creating an inspection task on the configuration change page, you can keep track of its progress and view details. You can even stop the inspection from that same page.

----End

Stopping an Inspection Task

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the cluster list, click the name of the target cluster.

Step 3 On the cluster details page, click **Inspection Management**.

Step 4 Locate the row that contains the inspection task and click **Stop** in the **Operation** column to stop the inspection task.

----End

Inspection Criteria

Table 9-29 Inspection criteria

Change Operation	Item	Check Criteria
Scaling operations and typical modifications	CheckTimeZone	The inspection passes if all nodes in the cluster use the same time zone, and fails if they do not.
	CheckSpaceUsage	When usage goes beyond the warning threshold (set at 70% by default), a warning is issued. If it goes beyond the NG threshold (set at 90% by default), the inspection fails. The inspection fails if the available space in the **GAUSSHOME/PGHOST/GPHOME/GAUSSLOG/tmp/data** directory is less than the threshold.
	CheckClusterState	It verifies the CM process, fenced UDF, and cluster status. If the CM process is missing, the inspection fails. A warning is issued if the fenced UDF status is **down**. The inspection passes for a normal cluster status, but fails otherwise.
	CheckEnvProfile	It checks the environment variables (**$GAUSSHOME**, **$LD_LIBRARY_PATH**, and **$PATH**) of a node. If these variables exist and are properly configured, the inspection passes. Otherwise, the inspection fails.
	CheckReadonlyMode	It checks the value of **default_transaction_read_only** on all nodes that contain CNs in the cluster. If the value is **off**, the inspection passes. Otherwise, the inspection fails.
	CheckCatchup	It checks whether the **CatchupMain** function can be found in the **gaussdb** process stack. If it cannot be found, the inspection passes. Otherwise, the inspection fails.
	CheckCollector	It checks whether the information collection is successful. If it is, the inspection passes. Otherwise, the inspection fails.

Change Operation	Item	Check Criteria
	CheckTrust	If any node is not trusted, the inspection fails. Otherwise, the inspection passes.
	CheckBalanceState	It checks the **Balanced** attribute of the cluster. If it is **Yes**, the inspection passes. A warning is displayed if it is not. If the query fails, the inspection also fails.
	CheckCnNumberSame	It checks if the number of CNs queried by running the **/opt/dws/xml/cluster.xml** command is different from that queried by running the **cm_ctl query -Cv** command. If they are different, the inspection passes. If not, the inspection fails.
	CheckCMParam	It checks if the value of **enable_transaction_read_only** is **on** and if the value of **coordinator_heartbeat_timeout** is consistent on each node. If both are true, the inspection passes.
	CheckUtilslib	It checks for the existence of the **$GAUSSHOME/utilslib** directory. If it exists, the inspection fails. If it does not exist, the inspection passes.
	CheckPgxcgroup	It checks the number of records in the **pgxc_group** table where **in_redistribution** is **Y**. If the number is **0**, the inspection passes. If the number is greater than 0, the inspection fails.
	CheckLockState	It checks whether the cluster is locked. If the cluster is not locked, the inspection passes. If it is locked, the inspection fails.
	CheckDBConnection	It checks whether the database can be connected. If it can, the inspection passes. Otherwise, the inspection fails.
	CheckGUCConsistent	It checks whether the GUC parameters of CNs and DNs are consistent. If they are consistent, the inspection passes. Otherwise, the inspection fails.
	CheckTDDate	The inspection fails if the ORC table in the TD database exists and has columns of the date type.

Change Operation	Item	Check Criteria
	CheckPgxcRedistb	If any temporary table remains in the database after data redistribution, the inspection fails.
	CheckMetaData	If metadata in the system table is consistent, the inspection passes. Otherwise, the inspection fails.
	CheckGUCSetting	If the GUC parameters in **postgresql.conf** are consistent with those in **pg_settings**, the inspection passes. Otherwise, the inspection fails.
	CheckProacl	The inspection fails if **proacl** in the **pg_proc** system table has usernames with only digits. Otherwise, it passes.
	CheckMetaDataConsistency	If the data in the system table between CN and DN is consistent, the inspection passes. Otherwise, the inspection fails.
	CheckReturnType	If the return value is invalid, the inspection fails. Otherwise, the inspection passes.
	CheckUltraWideTable	If a table with more than 996 columns exists, the inspection fails. Otherwise, the inspection passes.
	CheckDataRedisSchema	If a **data_redis** schema exists in the database and the owner name is not **redisuser**, the inspection fails. Otherwise, the inspection passes.
	CheckDiskSpaceLimited	If the disk space of the user is limited, the inspection fails. Otherwise, the inspection passes.
	CheckTableCollate	The inspection fails if the database has a PCK table or a column-store partitioned table with a **collate** field. Otherwise, it passes.
	CheckDefaultOrientation	It checks the GUC parameter. If both the database and **default_orientation** are set to row storage, the inspection passes. Otherwise, the inspection fails.
	CheckReplicatio-nUuid	The inspection passes if there is no replicated table that uses the default UUID. Otherwise, it fails.

Change Operation	Item	Check Criteria
	CheckUserStatementTimeout	If **statement_timeout** is not set or is set to **0**, the inspection passes. Otherwise, the inspection fails.
	CheckJsonb	Run the **select attrelid::regclass from pg_attribute a join pg_type t on a.atttypid = t.oid and t.typname = 'jsonb' group by 1** SQL statement. If the **jsonb** type is used, the inspection fails. Otherwise, the inspection passes.
	CheckLengthOfIndex	Run the **SELECT length(pg_get_indexdef(indexrelid)) FROM pg_index order by 1 desc limit 1** SQL statement. If the result is greater than 192 x 1024, the inspection fails. Otherwise, the inspection passes.
	CheckLengthOfTable	Run the **select c.oid from pg_class c,pg_namespace n where c.relnamespace=n.oid and relkind='r' and n.nspname not in ('cstore') and length(n.nspname||'.'||c.relname)>=64;** SQL statement. If the result is not empty, the inspection fails. Otherwise, the inspection passes.
	CheckUseWorkloadManager	Run the **show use_workload_manager** SQL statement. If the result is **on**, the inspection passes. Otherwise, the inspection fails. This inspection item is not included in version 8.1.3.320 and beyond. Meaning, it has been verified and passed in later versions.
	CheckNecessarySchema	It checks whether the necessary **schema(public)** exists. If **schema(public)** does not exist, the check fails.
	CheckCMParamConsistency	To pass this check, ensure that the parameter settings in the **cm.conf** file obtained from both the active and standby CM nodes are consistent. The check will pass if they are consistent and fail if they are not.

Change Operation	Item	Check Criteria
	CheckSQLCompatibility	In MySQL compatibility mode, redistribution of temporary tables with indexes is slow. If the **SHOW sql_compatibility** value is set to **mysql** in the service database and the **disable_including_all_mysql** enumerated value is not present in the **behavior_compat_options**, this item will not pass the check. However, if the enumerated value is included, the item will pass the check.
	CheckBinaryUpgrade	Check whether the corresponding backup file exists in the **/DWS/manager/upgrade_backup/**directory. If the backup file exists, the check fails. Otherwise, the check is passed.
	CheckColdTableSpace	Checks whether cold and hot tables exist in all databases. If yes, this item fails the check. If no, this item passes the check.
	CheckXFS	View the **/etc/os-release** file to obtain the version information. If EulerOS is used and the version is 4.19.87 or earlier, XFS bugs are involved and the check is not passed. Otherwise, the check is passed.
	CheckGTMConfigConsistency	It obtains the parameters in the configuration files of the active and standby GTMs. If the parameter settings are consistent, this item passes the check.
	CheckColversion	If there are column-store tables that are not marked as 1.0 and the current default column-store table is 2.0, this item fails the check. Otherwise, this item passes the check.
	CheckTopSqlSize	It checks the size of the **topsql** table. If the size exceeds 50 GB, the check fails.
	CheckDeltaTable	It checks the existence of the **delta** table. If the table exists, this item fails the check.
	CheckMaxDatanode	It checks the value of **comm_max_datanode**. If the value is not equal to the actual number of primary DataNodes, this item fails the check.

Change Operation	Item	Check Criteria
	CheckSSHIP	To pass this check, ensure that the parameter settings in the **cm.conf** file obtained from both the active and standby CM nodes are consistent. The check will pass if they are consistent and fail if they are not.
	CheckTimeZoneLink	Run the **ll /etc/localtime** command in the sandbox. If the file path to which the link points contains **/var/chroot**, the check is not passed.
	CheckSpecialFile	Checks whether files in the program directory **(GAUSSHOME)** contain special characters and files of non-Ruby users. If no, this item passes the check.
	CheckSysSchemaTable	If a user-created table exists in the system schema, the check fails. Otherwise, the check is successful.
Pre-upgrade health check	CheckClusterPar-ams	The cluster configuration parameters (IP address, port, and path parameters) specified in **postgresql.conf** or **pgxc_node** should match those in the static configuration file. Otherwise, the inspection fails.
	CheckCNNum	It checks the number of CNs in the cluster. If the number is greater than 2 and no more 10, the inspection passes. Otherwise, the inspection fails.
	CheckDDL	Start a transaction to delete schemas and tables. If the transaction can be committed, the inspection passes. Otherwise, the inspection fails.
	CheckTimeZone	The inspection passes if all nodes in the cluster use the same time zone, and fails if they do not.
	CheckXidEpoch	It checks the XID consumption. If the value is greater than or equal to 2 to the power of 32, the inspection fails.

Change Operation	Item	Check Criteria
	CheckCnNumberSame	It checks if the number of CNs queried by running the **/opt/dws/xml/cluster.xml** command is different from that queried by running the **cm_ctl query -Cv** command. If they are different, the inspection passes. If not, the inspection fails.
	CheckGaussVer	The inspection passes if the binary files in the **$GAUSSHOME/bin** directory on each node have identical versions.
	CheckPsort	The inspection fails if the **psort** index exists.
	CheckCatchup	It checks whether the **CatchupMain** function can be found in the **gaussdb** process stack. If it cannot be found, the inspection passes. Otherwise, the inspection fails.
	CheckClusterState	It verifies the CM process, fenced UDF, and cluster status. If the CM process is missing, the inspection fails. A warning is issued if the fenced UDF status is **down**. The inspection passes for a normal cluster status, but fails otherwise.
	CheckMetaDataConsistency	If the data in the system table between CN and DN is consistent, the inspection passes. Otherwise, the inspection fails.
	CheckDependSystemObj	If self-created objects depend on system objects, the inspection fails. Otherwise, the inspection passes.
	CheckPgKeyWords	If names of tables, columns, functions, or data types are new reserved keywords, the inspection fails. Otherwise, the inspection passes.
	CheckReadonlyMode	It checks the value of **default_transaction_read_only** on all nodes that contain CNs in the cluster. If the value is **off**, the inspection passes. Otherwise, the inspection fails.
	CheckMetaData	If metadata in the system table is consistent, the inspection passes. Otherwise, the inspection fails.

Change Operation	Item	Check Criteria
	CheckGUCSetting	If the GUC parameters in **postgresql.conf** are consistent with those in **pg_settings**, the inspection passes. Otherwise, the inspection fails.
	CheckPgxcgroup	It checks the number of records in the **pgxc_group** table where **in_redistribution** is **Y**. If the number is **0**, the inspection passes. If the number is greater than 0, the inspection fails.
	CheckCmserverStandby	If the value of **cm_server** in the cluster is **standby**, the inspection passes. Otherwise, a warning is displayed.
	CheckSpaceUsage	When usage goes beyond the warning threshold (set at 70% by default), a warning is issued. If it goes beyond the NG threshold (set at 90% by default), the inspection fails. The inspection fails if the available space in the **GAUSSHOME/PGHOST/GPHOME/GAUSSLOG/tmp/data** directory is less than the threshold.
	CheckEnvProfile	It checks the environment variables (**$GAUSSHOME**, **$LD_LIBRARY_PATH**, and **$PATH**) of a node. If these variables exist and are properly configured, the inspection passes. Otherwise, the inspection fails.
	CheckBalanceState	It checks the **Balanced** attribute of the cluster. If it is **Yes**, the inspection passes. A warning is displayed if it is not. If the query fails, the inspection also fails.
	CheckTDDate	The inspection fails if the ORC table in the TD database exists and has columns of the date type.
	CheckCatalog	If there is a custom database object in **pg_catalog**, the inspection fails. Otherwise, the inspection passes.
	CheckPgauthid	If the most significant bit of **oid** in **pg_authid** is **1**, the inspection fails. Otherwise, the inspection passes.
	CheckSysdate	If the **sysdate** view is used in tables, views, and stored procedures, the inspection fails. Otherwise, the inspection passes.

Change Operation	Item	Check Criteria
	CheckFilesNumber	If the number of **tmp** files in the **GAUSSHOME/PGHOST/GPHOME** directory is greater than 10,000, the inspection fails. Otherwise, the inspection passes.
	CheckKeyFilesExist	If the **upgrade_version**, **conf**, **control**, and **data** files exist in key directories, the inspection passes. Otherwise, the inspection fails.
	CheckReturnType	If the return value is invalid, the inspection fails. Otherwise, the inspection passes.
	CheckTrust	If any node is not trusted, the inspection fails. Otherwise, the inspection passes.
	CheckEnumGUCValue	It checks whether some parameters in **pg_postgres.conf** contain quotation marks. If single quotation marks are missing, this item fails the check.
	CheckSpecialFile	It checks whether files in the program directory **(GAUSSHOME)** contain special characters and files of non-Ruby users. If no, this item passes the check.
	CheckNecessarySchema	It checks the existence of the necessary **schemas (public)**.
	CheckUserDefinedDataType	It connects to all databases. Run the **select count(*) from pg_type t,pg_namespace n,PG_ATTRIBUTE a where t.typnamespace=n.oid and t.oid=a.atttypid and t.typname ='time_stamp' and n.nspname='information_schema' and a.atttypid> 16384;** SQL statement . If the result is empty, the check is passed.
	CheckCMParamConsistency	To pass this check, ensure that the parameter settings in the **cm.conf** file obtained from both the active and standby CM nodes are consistent. The check will pass if they are consistent and fail if they are not.

Change Operation	Item	Check Criteria
	CheckLightProxy	It checks the **enable_light_proxy** parameter. If the value is **off** and **behavior_compat_options** does not contain the enumerated value **enable_force_add_batch**, the item fails the check.
	CheckSSHIP	To pass this check, ensure that the parameter settings in the **cm.conf** file obtained from both the active and standby CM nodes are consistent. The check will pass if they are consistent and fail if they are not.
	CheckSysSchemaTable	If a user-created table exists in the system schema, the check fails. Otherwise, the check is successful.
	CheckTimeZoneLink	Run the **ll /etc/localtime** command in the sandbox. If the file path to which the link points contains **/var/chroot**, the check is not passed.

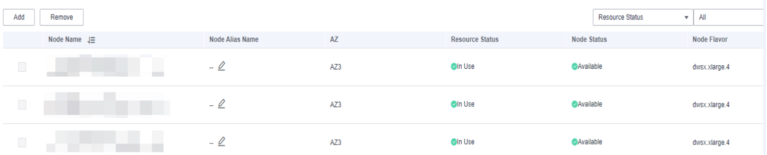
9.8.2 Managing Nodes

Overview

On the **Nodes** tab page, you can view the node list of the current cluster, add new nodes to or remove nodes from it, and view the node usage, status, flavor, and AZ.

In addition, you can click the 🖉 icon next to the text in the **Node Alias Name** column of a specified node to modify the alias of the node. If the node does not have an alias, you can add an alias for the node.

Figure 9-35 Managing Nodes



	Node Name	Node Alias Name	AZ	Resource Status	Node Status	Node Flavor
□	[redacted]	-- 🖉	AZ3	● In Use	● Available	dwsx.xlarge.4
□	[redacted]	-- 🖉	AZ3	● In Use	● Available	dwsx.xlarge.4
□	[redacted]	-- 🖉	AZ3	● In Use	● Available	dwsx.xlarge.4

NOTE

- This feature is supported only in 8.1.1.200 or later cluster versions.
- A storage-compute coupled data warehouse (standalone) does not support node management.

Adding Nodes

This function is more suited for large-scale scale-out. Nodes can be added in batches in advance without interrupting services. To add 180 nodes, add them in three batches of 60 nodes each. If any nodes fail to be added, retry adding them. Once all 180 nodes are added, use them for scaling out.

Precautions

- Nodes can be added only when no other task is running on the management side.
- The storage size of a new node must be the same as that of each of the existing nodes in the cluster.
- An added node, typically for scaling out, is referred to as an idle node. It starts incurring charges once added. You are advised to only add nodes when needed and promptly use them for scaling out.
- The anti-affinity rule dictates that the number of nodes to be added at a time must be an integer multiple of the cluster ring size. For example, if the cluster ring size is 3, the number of nodes to be added must be an integer multiple of 3.
- In the anti-affinity deployment mode, when a node is idle and fails due to power-off or other causes, it makes other nodes in its server group unavailable. In this case, you should remove and re-add the failed node.
- The anti-affinity rule dictates that, if a node fails to be added and is rolled back, other nodes that are being added in the same server group will also be rolled back.

Procedure

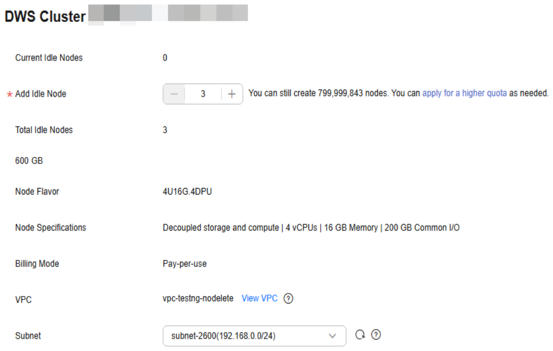
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 Click the name of the target cluster. On the **Cluster Information** page that is displayed, choose **Nodes**.

Step 4 Click **Add Node**, enter the number of idle nodes to be added, and click **Next: Confirm**. If you create a BMS cluster, contact technical support to add the cluster to the whitelist for cross-flavor scale-out. After the cluster is whitelisted, **Resource Flavor** will be displayed on the **Add Node** page.If there are not enough IP addresses in the original subnet, you can add idle nodes from other subnets.

Figure 9-36 Adding a node



DWS Cluster [redacted]

Current Idle Nodes 0

* Add Idle Node − 3 + You can still create 799,999,843 nodes. You can apply for a higher quota as needed.

Total Idle Nodes 3

600 GB

Node Flavor 4U16G.4DPU

Node Specifications Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 200 GB Common I/O

Billing Mode Pay-per-use

VPC vpc-testng-nodelete View VPC ?

Subnet subnet-2600(192.168.0.0/24)

NOTE

In yearly/monthly mode, the time remaining and the expiration time are displayed.

Step 5 After confirming that the information is correct, click **Submit**. The **Nodes** page is displayed. On this page, you can start adding nodes. Nodes that fail to be added are automatically rolled back and recorded in the failure list.

----End

Removing Nodes

Precautions

- Nodes can be removed only when no other task is running on the management side.
- Only nodes whose resource status is **Idle** can be removed. Nodes that are in use cannot be removed.
- In anti-affinity deployment, nodes are removed by cluster ring. For example, when you remove a node, other nodes in the same ring will be automatically selected and displayed.
- In a yearly/monthly cluster during the grace period or retention period, nodes cannot be deleted. You can release nodes on the **Renewals** page of the Billing Center.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 Click the name of the target cluster. On the **Cluster Information** page that is displayed, choose **Nodes**.

Step 4 On the **Nodes** page, select the node to be deleted and click **Delete**.

Step 5 (Yearly/Monthly billing mode) The page for deletion confirmation is displayed.

Step 6 Confirm the information and click **OK**. After the deletion is successful, the node is no longer displayed on the **Nodes** page.

----End

9.8.3 Scaling Nodes

9.8.3.1 Scaling Out a Cluster

When you need more compute and storage resources, add more nodes for cluster scale-out on the management console.

NOTE

- If a cluster is billed in yearly/monthly mode, new nodes in the cluster will also be billed in this mode.
- When you scale out a storage-compute coupled data warehouse cluster, use the same storage specifications as the cluster.
- Nodes cannot be added to a storage-compute coupled data warehouse (standalone).
- If you create a BMS cluster, contact technical support to add you to the whitelist for cross-flavor scale-out. After you are whitelisted, **Resource Flavor** will be displayed on the **Add Node** page.
- If the number of subnet IP addresses is insufficient, cross-subnet scale-out is allowed.

After the data in a data warehouse is deleted, the occupied disk space may not be released, resulting in dirty data and disk waste. Therefore, if you need to scale out your cluster due to insufficient storage capacity, run the **VACUUM** command to reclaim the storage space first. If the used storage capacity is still high after you run the **VACUUM** command, you can scale out your cluster. For details about the VACUUM syntax, see **VACUUM** in the *SQL Syntax Reference*.

Impact on the System

- Before the scale-out, disable the client connections that have created temporary tables because temporary tables created before or during the scale-out will become invalid and operations performed on these temporary tables will fail. Temporary tables created after the scale-out will not be affected.
- After you start a scale-out task, the cluster automatically takes a snapshot before the task begins.
- Certain cluster functions, including restarting, stopping, and starting, modifying specifications, adding or removing CNs, creating snapshots, and resetting the database administrator's password, cannot be performed while scaling out the cluster.
- During an offline scale-out, the cluster automatically restarts. Therefore, the cluster changes to **Unavailable** for a period of time. After the cluster is restarted, the status becomes **Available**. At the end of the scale-out, if you select automatic redistribution, the system dynamically redistributes user data in the cluster to all nodes. Otherwise, you need to start data redistribution.
- During offline scale-out, stop all services or run only a few query statements. During table redistribution, a shared lock is added to tables. All insert, update, and delete operations as well as DDL operations on the tables are blocked for a long time, which may cause a lock wait timeout. After a table is redistributed, you can access the table. Do not perform queries that take more than 20 minutes during the redistribution (the default time for applying for the write lock during redistribution is 20 minutes). Otherwise, data redistribution may fail due to lock wait timeout.
- In an online scale-out, during node addition, the cluster is locked Database objects are checked when the cluster is locked. To ensure that the cluster is successfully locked, avoid executing statements that create or delete databases and tablespaces while adding nodes to the cluster.
- During online scale-out, you can perform insert, update, and delete operations on tables, but data updates are still be blocked for a short period of time. Redistribution consumes lots of CPU and I/O resources, which will

greatly impact job performance. Therefore, perform redistribution when services are stopped or during periods of light load. Phase-based scale-out is also recommended: Perform high-concurrency redistribution during periods of light load, and stop redistribution or perform low-concurrency redistribution during periods of heavy load.

- If a new snapshot is created for the cluster after the scale-out, the new snapshot contains data on the newly added nodes.
- If the cluster scale-out fails, the database automatically performs the rollback operation in the background so that the number of nodes in the cluster can be restored to that before the scale-out.
  - If the rollback is successful and the cluster can be normally used, you can perform **Scale Out** again. If the scale-out still fails, contact the technical support.
  - If the rollback fails due to some exceptions, the cluster may become **Unavailable**. In this case, you cannot perform **Scale Out** or restart the cluster. Contact the technical support.
- If the number of buckets allocated to each DN in the storage-compute decoupled cluster scale-out scenario is not between [3, 20], automatic scaling will be triggered. You can view the number of buckets using the GUC parameter **table_buckets**.
  - Currently, buckets can only be scaled offline. The procedure is the same as that of the existing scaling procedure. The system automatically determines and executes the bucket scaling process.
  - During scaling, the cluster will be restarted and all connections will be closed. The restart takes several minutes.
  - After the restart is complete, the database can be read but cannot be written until data redistribution is complete.

  For example, if the number of buckets on the current node is 32 and the number of DNs in the logical cluster is 9, and the number of DNs needs to be increased to 15, as 32/15=2 (rounded down) does not fall within the range [3,20], bucket scale-out will be triggered.

Prerequisites

- The cluster to be scaled out is in the **Available**, **Read-only**, or **Unbalanced** state.
- The number of nodes to be added must be less than or equal to the available nodes. Otherwise, system scale-out is not allowed.
- To scale out a cluster as an IAM user, ensure that the IAM user has permissions for VPC, EVC, and BMS.

Scaling Out a Cluster

NOTE

- A cluster becomes read-only during scale-out. Exercise caution when performing this operation.
- The cluster will be intermittently disconnected during scale-out. Exercise caution when performing this operation.
- To ensure data security, you are advised to create a snapshot before the scale-out. For details about how to create a snapshot, see **Manual Snapshots**.
- After you start a scale-out, the system first checks for scale-out prerequisites. If your cluster fails the check, modify configurations as prompted and try again. For details, see **What Do I Do If the Scale-out Check Fails?**

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

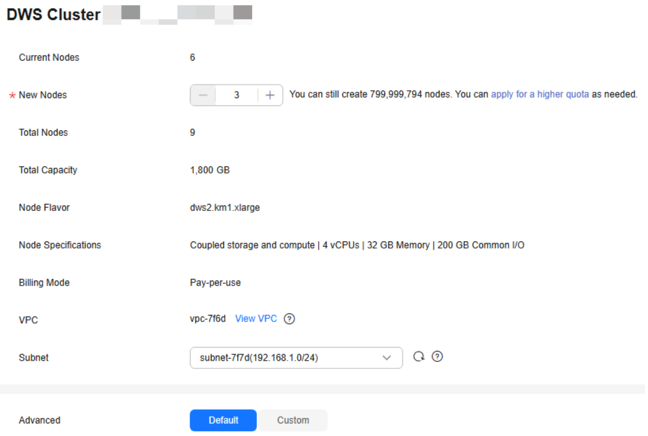
All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Scale Out**. The scale-out page is displayed.

Before scaling out the cluster, it is crucial to verify if it meets the inspection conditions. Click **Immediate Inspection** to complete the inspection and proceed to the next step only if it passes. For more information, see **Viewing Inspection Results**.

- If you create a BMS cluster, contact technical support to add you to the whitelist for cross-flavor scale-out. After you are whitelisted, **Resource Flavor** will be displayed on the **Add Node** page.
- If the IP addresses of the original subnet are insufficient, you can expand the capacity across subnets.

Figure 9-37 Scaling out a cluster



DWS Cluster [redacted]

Current Nodes	6
* New Nodes	3 You can still create 799,999,794 nodes. You can apply for a higher quota as needed.
Total Nodes	9
Total Capacity	1,800 GB
Node Flavor	dws2.km1.xlarge
Node Specifications	Coupled storage and compute | 4 vCPUs | 32 GB Memory | 200 GB Common I/O
Billing Mode	Pay-per-use
VPC	vpc-7f6d View VPC
Subnet	subnet-7f7d(192.168.1.0/24)
Advanced	Default Custom

NOTE

In yearly/monthly billing mode, the number of nodes in the discount package is not displayed. The remaining time and the expiration time are displayed.

Step 4 Specify the number of nodes to be added.

- DNs are added during scale-out. For details about how to add CNs, see **Adding or Deleting a CN in a GaussDB(DWS) Cluster**.
- The number of nodes after scale-out must be at least three nodes more than the original number. The maximum number of nodes that can be added depends on the available quota. In addition, the number of nodes after the scale-out cannot exceed 256.

  If the node quota is insufficient, click **Increase quota** to submit a service ticket and apply for higher node quota.
- Flavor of the new nodes must be the same as that of existing nodes in the cluster.
- The VPC and security group of the cluster with new nodes added are the same as those of the original cluster.
- The number of nodes to be added to a multi-AZ cluster must be a multiple of 3.

Step 5 Configure advanced parameters.

- If you choose **Default**, **Auto Redistribution** will be enabled, **Scale Online** will be disabled, and **Redistribution Mode** will be **Offline** by default.
- If you choose **Custom**, you can configure the following advanced configuration parameters for scale-out:
  - **Scale Online**: Online scale-out can be enabled. During online scale-out, data can be added, deleted, modified, and queried in the database; and some DDL syntaxes are supported. Errors will be reported for unsupported syntaxes.
  - **Terminate Blocked Job**: If you enable online scale-out, you can configure automatic job termination.
  - **Time Before Blocked Job Termination (s)**: If job termination is enabled and congestion occurs during online scale-out, the system waits for the duration you specified and then terminates congested jobs. The value can be an integer in the range 30 to 1200.

    NOTE

    Clusters of version 8.2.1.100 and later support job termination.
  - **Auto Redistribution**: Automatic redistribution can be enabled. If automatic redistribution is enabled, data will be redistributed immediately after the scale-out is complete. If this function is disabled, only the scale-out is performed. In this case, to redistribute data, select a cluster and choose **More** > **Scale Node** > **Redistribute**.
  - **Redistribution Concurrency**: If automatic redistribution is enabled, you can set the number of concurrent redistribution tasks. The value range is 1 to 200. The default value is **4**.
  - **Auto Redistribution**: Select **Online** or **Offline** as needed.

Step 6 Confirm the settings, select the confirmation check box, and click **Next: Confirm**.

Step 7 Click **Submit**.

- After you submit the scale-out application, task information of the cluster changes to **Scaling out** and the process will take several minutes.

- During the scale-out, the cluster automatically restarts. Therefore, the cluster status will stay **Unavailable** for a while. After the cluster is restarted, the status will change to **Available**.
- After the scale-out is complete, the system dynamically redistributes user data in the cluster, during which the cluster is in the **Read-only** state.
- A cluster is successfully scaled out only when the cluster is in the **Available** state and task information **Scaling out** is not displayed. Then you can use the cluster.
- If **Scale-out failed** is displayed, the cluster fails to be scaled out.

----End

Scaling Out with Idle Nodes

To ensure reliability, prepare ECS first by referring to **Adding Nodes** for a large-scale cluster, and scale out the cluster using idle nodes.

NOTE

- Disable automatic redistribution when you scale out a large-scale cluster to facilitate retries upon failures for improved reliability.
- After the scale-out is complete, manually perform **redistribution** to ensure that multiple retries can be performed in this phase.

Precautions

- A number of available nodes must be added to the cluster in advance so that idle nodes can be created and added for scale-out.
- The anti-affinity rule dictates that the number of idle nodes to be added must be an integer multiple of the cluster ring size.
- Make sure to configure the scale-out task before submitting it. This involves completing the scale-out preparation. Once done, wait for a moment.
- After you start a scale-out, the system first checks for scale-out prerequisites. If your cluster fails the check, modify configurations as prompted and try again. For details, see **What Do I Do If the Scale-out Check Fails?**

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Scale Out**.

Before scaling out the cluster, it is crucial to verify if it meets the inspection conditions. Click **Immediate Inspection** to complete the inspection and proceed to the next step only if it passes. For more information, see **Viewing Inspection Results**.

If there are idle nodes in the cluster, the system displays a message asking you whether to add nodes.

Step 4 Click the corresponding button to make scale-out preparations and wait until the preparation is complete.

Step 5 Configure the parameters as required. For details, see **Scaling Out a Cluster**.

After setting the scale-out and redistribution parameters, select the confirmation check box, and click **Next: Confirm**.

Step 6 Confirm the information and click **Submit**.

----End

Viewing Scaling Details

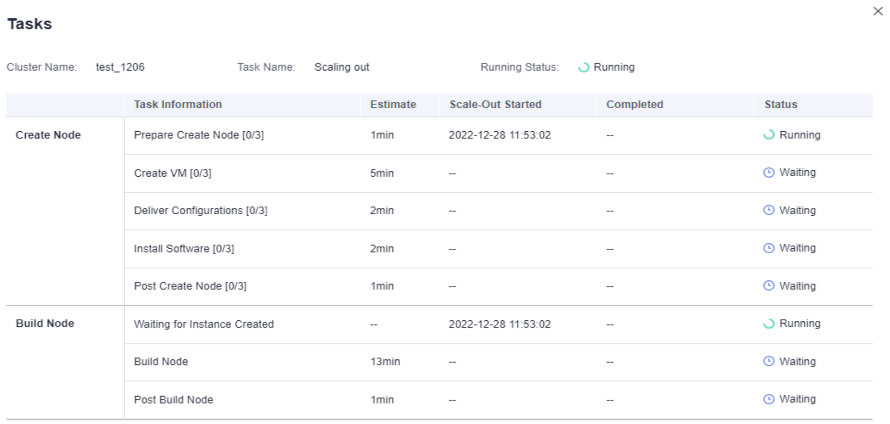
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

Step 3 In the **Task Information** column of a cluster, click **View Details**.

Step 4 Check the scale-out status of the cluster on the scaling details page.

Figure 9-38 Viewing scale-out details



Tasks

Cluster Name: test_1206 Task Name: Scaling out Running Status: Running

	Task Information	Estimate	Scale-Out Started	Completed	Status
Create Node	Prepare Create Node [0/3]	1min	2022-12-28 11:53:02	--	Running
	Create VM [0/3]	5min	--	--	Waiting
	Deliver Configurations [0/3]	2min	--	--	Waiting
	Install Software [0/3]	2min	--	--	Waiting
	Post Create Node [0/3]	1min	--	--	Waiting
Build Node	Waiting for Instance Created	--	2022-12-28 11:53:02	--	Running
	Build Node	13min	--	--	Waiting
	Post Build Node	1min	--	--	Waiting

----End

9.8.3.2 Cluster Redistribution

9.8.3.2.1 Redistributing Data

Data redistribution, where data in existing nodes is evenly allocated to the new nodes after you scale out a cluster, is a time-consuming yet crucial task that accelerates service response.

By default, redistribution is automatically started after cluster scale-out. For enhanced reliability, disable the automatic redistribution function and manually start a redistribution task after the scale-out is successful. In this way, both scale-out and redistribution can be retried upon failures.

GaussDB(DWS) supports **offline redistribution** and **online redistribution**. The default mode is offline redistribution.

Before redistribution starts or when redistribution is paused, you can set redistribution priorities for the tables that have not been redistributed by schema or table.

NOTICE

- The cluster redistribution function is supported in 8.1.1.200 or later cluster versions.
- This function can be manually enabled only when the cluster task information displays **To be redistributed** after scale-out.
- You can also select the redistribution mode when you configure cluster scale-out (see **Step 5**).
- Redistribution queues are sorted based on the relpage size of tables. To ensure that the relpage size is correct, you are advised to perform the **ANALYZE** operation on the tables to be redistributed.

Offline Redistribution

Precautions

- In offline redistribution mode, the database does not support DDL and DCL operations. Tables that are being redistributed support only simple DQL operations.
- During table redistribution, a shared lock is added to tables. All insert, update, and delete operations as well as DDL operations on the tables are blocked for a long time, which may cause a lock wait timeout. Do not perform queries that take more than 20 minutes during the redistribution (the default time for applying for the write lock during redistribution is 20 minutes). Otherwise, data redistribution may fail due to lock wait timeout.

Procedure

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Redistribute**, as shown in the following figure.

The **Redistribution** page is displayed.

Step 4 On the **Redistribute** page that is displayed, keep the default **offline** redistribution mode and click **Next: Confirm** to submit the task.

----End

Online Redistribution

Precautions

In online redistribution mode, the database supports partial DDL and DCL operations.

- Local tables that are being redistributed support insert, delete and update operations and some DDL operations:
  - **INSERT**, **DELETE**, **UPDATE**, **MERGE INTO**, **OVERWRITE**, **UPSERT**
  - Join queries across node groups

  - Local table renaming, schema modification, **DROP**, **TRUNCATE**, **TRUNCATE-PARTITION**
- The following operations cannot be performed on tables that are being redistributed:
  - Run **ALTER TABLE** statements (except for **TRUNCATE PARTITION**), including adding or deleting columns or partitions.
  - Create, modify, or delete indexes.
  - Perform **VACUUM FULL** and **CLUSTER** operations on tables.
  - Modify the sequence objects on which a column depends, including creating and modifying them. Typical statements are **CREATE** and **ALTER SEQUENCE ... OWNED BY**.
  - During the redistribution of a table with more than 996 columns, **UPDATE** and **DELETE** statements cannot be executed. **SELECT** and **INSERT** statements are allowed.
  - Database and tablespace objects cannot be created, deleted, or modified during redistribution.
  - A partition swap can be performed only if the redistribution is complete for both of the tables to be swapped. The two tables belong to different node groups and do not allow partition swap if either of them is being redistributed.

Procedure

Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Redistribute**, as shown in the following figure.

Step 4 On the **Redistribute** page that is displayed, set **Advanced** to **Custom**, set the redistribution mode to **Online mode**, and click **Next: Confirm** to submit the task.

----End

9.8.3.2.2 Viewing Redistribution Details

On the **View Redistribution Details** page, you can check the monitoring information, including the redistribution mode, redistribution progress, and table redistribution details of the current cluster. You can pause and resume redistribution, set the redistribution priority, and change the number of concurrent redistribution tasks.

NOTE

The function of viewing redistribution details is supported by 8.1.1.200 and later cluster versions. Details about the data table redistribution progress are supported only by 8.2.1 and later cluster versions.

Precautions

You can check redistribution details only if the cluster is being redistributed, failed to be redistributed, or is suspended. There may be a delay in the statistics update.

Procedure

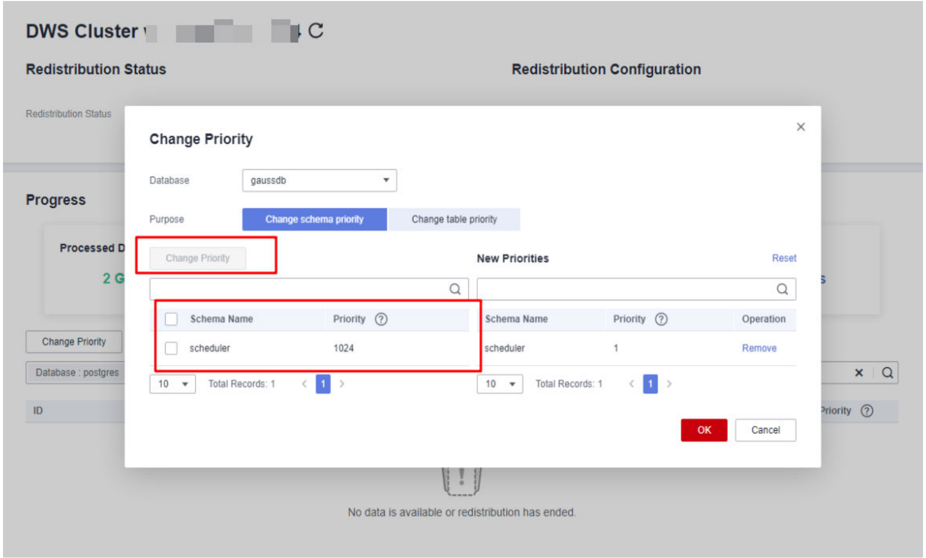
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the **Task Information** column of a cluster, click **View Details**.

Step 4 Check the redistribution status, configuration, progress, and redistribution details of all the tables in a specified database. Specify a database that and can be searched by table redistribution status and table name. If all the tables in a database have completed redistribution, no data will be displayed for the database.

Step 5 When redistribution is paused, you can set the redistribution priority (in schema or table dimension), and redistribution will be performed based on the configured redistribution sequence. You can also set the redistribution priority before the redistribution starts.



DWS Cluster [redacted]

Redistribution Status

Redistribution Configuration

Redistribution Status

Progress

Processed D

2 G

Change Priority

Database : postgres

ID

Priority

No data is available or redistribution has ended.

Change Priority

Database: gaussdb

Purpose: Change schema priority | Change table priority

Change Priority

Schema Name	Priority
scheduler	1024

10 Total Records: 1 ‹ 1 ›

New Priorities

Reset

Schema Name	Priority	Operation
scheduler	1	Remove

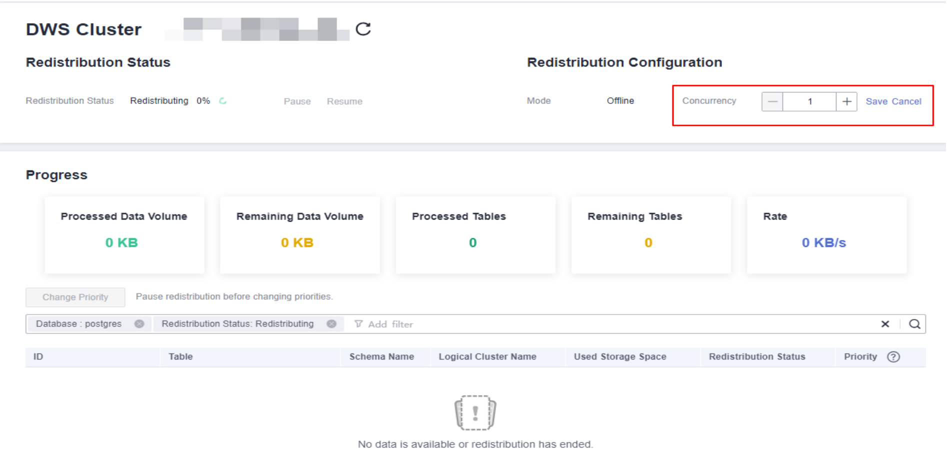
10 Total Records: 1 ‹ 1 ›

OK Cancel

Step 6 The number of concurrent redistribution tasks can be adjusted during redistribution.

NOTE

Cluster 8.1.0 and earlier versions do not support dynamic adjustment. To change redistribution concurrency, suspend redistribution first.



DWS Cluster [redacted]

Redistribution Status

Redistribution Status: Redistributing 0%

Pause Resume

Redistribution Configuration

Mode: Offline

Concurrency: 1 Save Cancel

Progress

Processed Data Volume	Remaining Data Volume	Processed Tables	Remaining Tables	Rate
0 KB	0 KB	0	0	0 KB/s

Change Priority

Pause redistribution before changing priorities.

Database : postgres

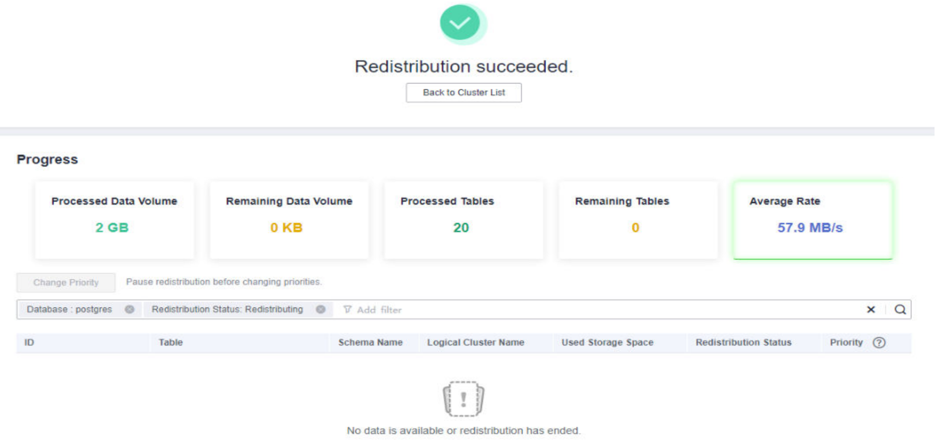
Redistribution Status: Redistributing

Add filter

ID	Table	Schema Name	Logical Cluster Name	Used Storage Space	Redistribution Status	Priority

No data is available or redistribution has ended.

Step 7 Check the redistribution progress. After the redistribution is complete, the amount of completed data, amount of remaining data, number of completed tables, number of remaining tables, and average rate during redistribution are displayed.



Redistribution succeeded.

Progress

Processed Data Volume: 2 GB

Remaining Data Volume: 0 KB

Processed Tables: 20

Remaining Tables: 0

Average Rate: 57.9 MB/s

Pause redistribution before changing priorities.

Database : postgres

Redistribution Status: Redistributing

ID	Table	Schema Name	Logical Cluster Name	Used Storage Space	Redistribution Status	Priority

No data is available or redistribution has ended.

----End

9.8.3.3 Scaling In a Cluster

You can scale in your clusters on the console to release unnecessary compute and storage resources provided by GaussDB(DWS).

NOTE

- Scale-in is supported only by pay-per-use clusters of version 8.1.1.300 and later. For clusters billed in yearly/monthly mode, the function is supported only in version 8.2.1 and later.
- By default, scaled in nodes are charged by quantity.
- When you scale in a storage-compute coupled data warehouse cluster, you can only modify the same storage specifications as used by the cluster.
- A storage-compute coupled data warehouse (cluster mode) cannot be scaled in to a standalone cluster.

Impact on the System

- Before the scale-in, close the client connections that have created temporary tables, because temporary tables created before or during the scale-in will become invalid and operations performed on these temporary tables will fail. Temporary tables created after the scale-in will not be affected.
- If you start a scale-in, an automatic snapshot will be created for the cluster before scale-in. If you do not need the snapshot, you can disable the automated backup function on the scale-in page.
- Ensure that the skew rate is below 10% before scale-in and there is no specific guideline for the dirty page rate. However, it is advisable to maintain a skew rate of 20-30% for large tables exceeding 50 GB in size.
- When scaling in a cluster, several functions are disabled, including cluster restart, cluster scale-out, snapshot creation, node management, intelligent O&M, resource management, parameter modification, security configurations, log service, database administrator password resetting, and cluster deletion.

- During offline scale-in, stop all services or run only a few query statements. During table redistribution, a shared lock is added to tables. All insert, update, and delete operations as well as DDL operations on the tables are blocked for a long time, which may cause a lock wait timeout. After a table is redistributed, you can access the table. Do not perform queries that take more than 20 minutes during the redistribution (the default time for applying for the write lock during redistribution is 20 minutes). Otherwise, data redistribution may fail due to lock wait timeout.
- During online scale-in, you can perform insert, update, and delete operations on tables, but data updates may still be blocked for a short period of time. Redistribution consumes lots of CPU and I/O resources, which will greatly impact job performance. Therefore, perform redistribution when services are stopped or during periods of light load.
- If a node is deleted while DDL statements are being executed (to create a schema or function) during online scale-in, errors may occur because the DN cannot be found. To resolve this issue, you can simply retry the statements.
- If a cluster scale-in fails, the database does not automatically roll back the scale-in operation, and no O&M operations can be performed. In this case, you need to click the **Scale In** on the console to try again.
- In the cloud native 9.0.2 scale-out scenario, if the number of buckets allocated to each DN is not between [3, 20], the system adjusts the number of buckets. You can view the number of buckets using the GUC parameter **table_buckets**.
  - Currently, the bucket scaling supports only the offline mode. The procedure is the same as that of the existing scaling procedure. The system automatically determines and executes the bucket scaling process.
  - During scaling, the cluster will be restarted and all connections will be closed. The restart takes several minutes.
  - After the restart is complete, the database can be read but cannot be written until data redistribution is complete.

Prerequisites

- The cluster is in **Available** state, is not read-only, and there is no data being redistributed in the cluster.
- A cluster configuration file has been generated, and configuration information is consistent with the current cluster configuration.
- Before the scale-in operation starts, the value of **default_storage_nodegroup** is **installation**.
- The cluster is configured in the ring mode. A ring is the smallest unit for scale-in. Four or five hosts form a ring. The primary, standby, and secondary DNs are deployed in this ring. If the current cluster has only one cluster ring, scale-in is not supported and the scale-in button is unavailable.
- The scale-in host does not contain the GTM, ETCD, or CM Server component.
- There are no CNs on the nodes to be scaled in.
- Scale-in does not support rollback but supports retry. A data redistribution failure after a scale-in does not affect services. You can complete scale-in at other appropriate time. Otherwise, unbalanced data distribution will persist for a long time.
- Before redistribution, ensure that the **data_redis** schema in the corresponding database is reserved for redistribution and that no user operation on it or its

tables is allowed. During redistribution, **data_redis** is used. After the operation is complete, the schema will be deleted. User tables (if any) in the schema will also be deleted.

- **gs_cgroup** cannot be used during scale-in.
- Before the scale-in, check the remaining capacity of the cluster. The nodes remaining in a scale-in must have sufficient space to store the data of the entire cluster. Otherwise, the scale-in cannot be properly performed.
  - The used physical disk space on each node is less than 80%.
  - All the users and roles use less than 80% of resource quota in total.
  - The estimated space usage after scale-in must be less than 80%.
  - The available space is 1.5 times larger than the maximum size of a single table.
- Automatic removal of faulty CNs is disabled during the scale-in and is enabled after the scale-in is complete.

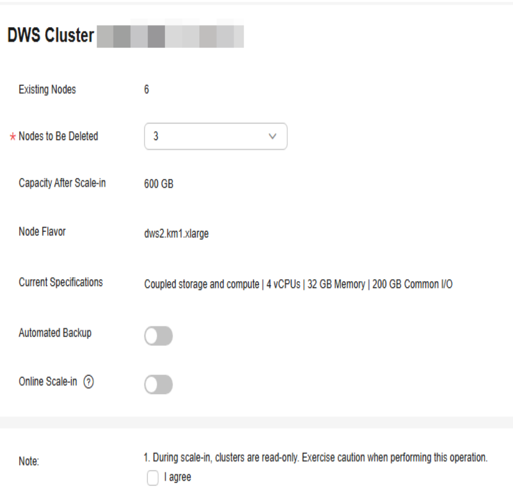
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Scale In**.

Step 4 The scale-in page is displayed. You can select the number of nodes to be scaled in. The automated backup function is enabled by default. The storage-compute decoupled cluster does not have the **Automated Backup** switch.



DWS Cluster [redacted]

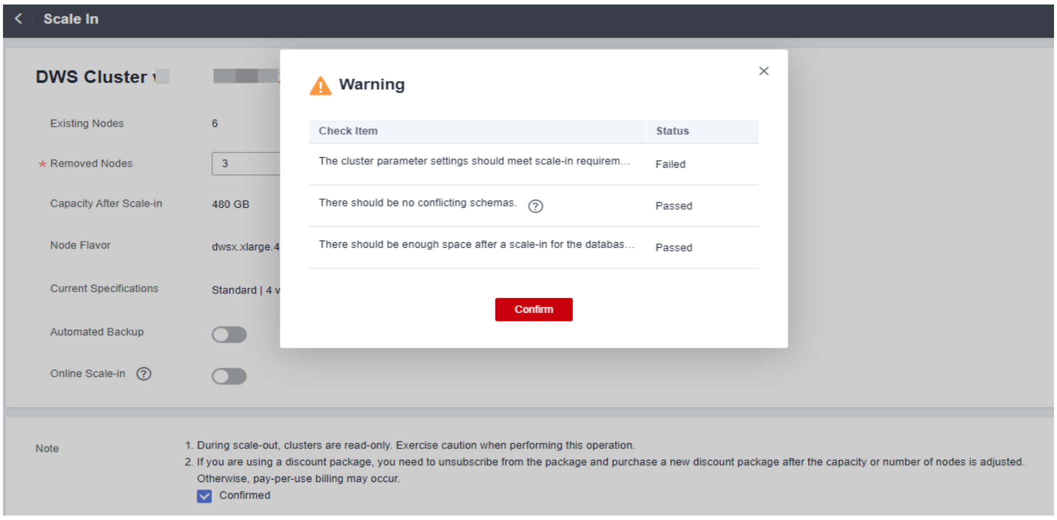
Existing Nodes	6
* Nodes to Be Deleted	3
Capacity After Scale-in	600 GB
Node Flavor	dws2.km1.xlarge
Current Specifications	Coupled storage and compute | 4 vCPUs | 32 GB Memory | 200 GB Common I/O
Automated Backup	○
Online Scale-in ⓘ	○

Note: 1. During scale-in, clusters are read-only. Exercise caution when performing this operation.
□ I agree

NOTE

Before scaling in the cluster, it is crucial to verify if it meets the inspection conditions. Click **Immediate Inspection** to complete the inspection and proceed to the next step only if it passes. For more information, see **Viewing Inspection Results**.

Step 5 Click **Next: Confirm**. The system will check the cluster status before scale-in. If your cluster fails the check, an error message will be displayed.



DWS Cluster [redacted]

Existing Nodes: 6

* Removed Nodes: 3

Capacity After Scale-in: 480 GB

Node Flavor: dwsx.xlarge.4

Current Specifications: Standard | 4 v

Automated Backup

Online Scale-in

Warning

Check Item	Status
The cluster parameter settings should meet scale-in requirem...	Failed
There should be no conflicting schemas.	Passed
There should be enough space after a scale-in for the databas...	Passed

Note

1. During scale-out, clusters are read-only. Exercise caution when performing this operation.
2. If you are using a discount package, you need to unsubscribe from the package and purchase a new discount package after the capacity or number of nodes is adjusted. Otherwise, pay-per-use billing may occur.

☑ Confirmed

Step 6 After the check is passed, click **Confirm** to return to the cluster list. The cluster status is **Scaling in**. Wait for a while.

Step 7 (Yearly/Monthly billing mode) After the cluster scale-in is complete, you will be prompted to delete idle nodes. Click **OK** to delete idle nodes.

Step 8 On the **Delete Node** page, view the resource information and click **Submit**.

Step 9 (Yearly/Monthly billing mode) On the displayed resource confirmation page, confirm the refund information and click **Submit**.

----End

NOTE

- After the scale-in of a pay-per-use cluster is complete, specified nodes will be automatically removed in the background. For a yearly/monthly cluster, you need to manually delete the nodes.
- If the cluster parameters fail the check, the scale-in will fail. To avoid this problem, ensure your parameter settings are correct.
- If schemas fail the check, the scale-in will fail. To avoid this problem, check whether any schema that conflicts with the scale-in exists.
- If the disk space fails the check, the scale-in may fail or the cluster may become read-only after the scale-in. To avoid this problem, increase your cluster disk capacity.

9.9 Changing GaussDB(DWS) Cluster Specifications

9.9.1 Using the Elastic Specification Change

Overview

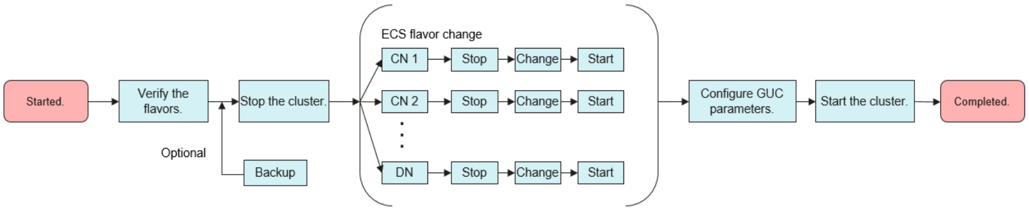
Heavy service traffic requires additional resources (such as CPU, memory, and disk resources) to support it. If the current cluster resources are insufficient, creating a new cluster with more resources may be necessary. However, this can be costly and time-consuming. Moreover, creating a cluster with many resources but low service volume can result in resource redundancy and high costs.

Elastic specification change is introduced to tackle this problem. It is ideal for scenarios where computing capabilities (CPU and memory) need to be adjusted during peak hours or when only computing capabilities need to be changed. By using elastic specification change before peak hours, the cluster's computing capability can be quickly increased. After peak hours, the cluster configuration can be reduced to minimize costs. For more information, see **Supported node flavors**.

You can modify the CPU and memory configurations of the VM nodes in the target cluster by utilizing the underlying ECS capabilities. The following figure illustrates this process.

- To prevent service disruptions, it is crucial to schedule the elastic specification change time window properly since the cluster must be stopped during the entire process.
- Changing all nodes concurrently ensures that the process will not take longer due to the number of nodes. Typically, the entire process takes around 5 to 10 minutes.

Figure 9-39 Principle of elastic specification change



NOTE

- Only cluster versions 8.1.1.300 and later support elastic specification change. For an earlier version, contact technical support to upgrade it first.
- Elastic specification change is only supported by storage-compute coupled and decoupled clusters that use ECSs and EVS disks. Clusters with local ECSs do not have this capability.

Precautions

- Decreasing the specifications of a cluster is to select the target specifications that are lower than the current specifications of the cluster. This operation may affect the cluster performance. Therefore, evaluate service impact before performing this operation.
- Make sure to check if there are enough ECS resources and tenant CPU quotas in the current region before modifying the specifications.
- You can change the specifications again if needed. In case the specifications of some nodes fail to change, you can resubmit the change task to execute the process.

Constraints and Limitations

- You can upgrade or downgrade ECS specifications of the same type. For instance, you can change from **dwsx2.2xlarge.m7** to **dwsx2.4xlarge.m7**, but not to **dwsx2.4xlarge.m6**.

- Stop the VM before changing the specifications. The specification change can only be done offline and it takes 5 to 10 minutes.
- If you choose the year/monthly billing mode (default), you can easily increase your cluster's specifications as needed. However, decreasing the specifications is not possible in this mode. To do so, you will need to switch to pay-per-use billing mode first.

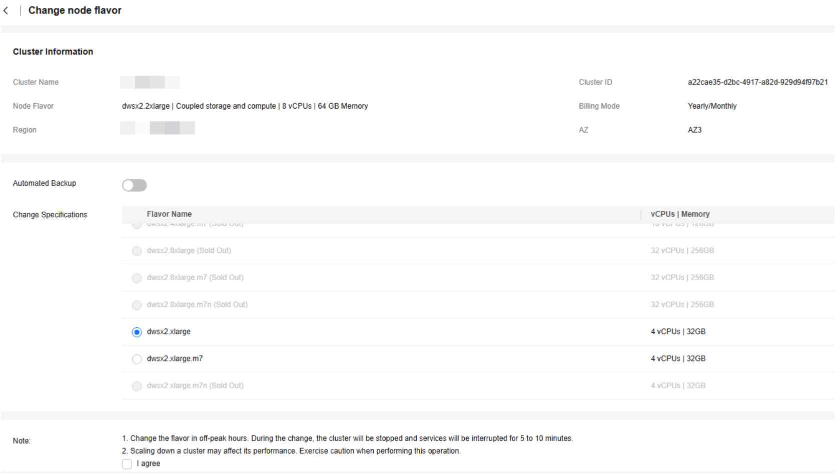
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the row of a cluster, choose **More** > **Change Flavor** in the **Operation** column and click **Change Node Flavor**.

Step 4 Configure the flavor. Enable automatic backup as needed.



Change node flavor

Cluster Information

Cluster Name	[redacted]	Cluster ID	a22cae35-d2bc-4917-a82d-929d94f97b21
Node Flavor	dwsx2.2xlarge | Coupled storage and compute | 8 vCPUs | 64 GB Memory	Billing Mode	Yearly/Monthly
Region	[redacted]	AZ	AZ3

Automated Backup: off

Change Specifications

	Flavor Name	vCPUs | Memory
○	dwsx2.4xlarge.m7 (Sold Out)	16 vCPUs | 128GB
○	dwsx2.8xlarge (Sold Out)	32 vCPUs | 256GB
○	dwsx2.8xlarge.m7 (Sold Out)	32 vCPUs | 256GB
○	dwsx2.8xlarge.m7n (Sold Out)	32 vCPUs | 256GB
●	dwsx2.xlarge	4 vCPUs | 32GB
○	dwsx2.xlarge.m7	4 vCPUs | 32GB
○	dwsx2.xlarge.m7n (Sold Out)	4 vCPUs | 32GB

Note: 1. Change the flavor in off-peak hours. During the change, the cluster will be stopped and services will be interrupted for 5 to 10 minutes.
2. Scaling down a cluster may affect its performance. Exercise caution when performing this operation.
□ I agree

Step 5 Confirm the settings, select the confirmation check box, and click **Next: Confirm**.

Step 6 Click **Submit**.

Step 7 Confirm the information and click **Submit**.

NOTE

If the billing mode is yearly/monthly, you will be redirected to the payment page.

Step 8 Return to the cluster list. The cluster status will change to **Changing node flavor**. Wait for about 5 to10 minutes.

----End

Hybrid Billing for Specifications Change in a Yearly/Monthly Billed Cluster

Prerequisites

During peak hours, nodes are automatically added to a yearly/monthly-billed cluster as scheduled. Nodes are billed on a pay-per-use basis. For details, see **Elastically Adding or Deleting a Logical Cluster**. After scale-out, the cluster uses the hybrid billing mode, that is, both the pay-per-use and yearly/monthly billing modes are used. Nodes created by creating addition or deletion plans are pay-per-use nodes, and nodes created by creating a yearly/monthly cluster are yearly/monthly nodes.

NOTE

Only storage-compute decoupled clusters support hybrid billing.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 2 Click **Create GaussDB(DWS) Cluster** in the upper right corner of the page to **create a storage-compute decoupled cluster**.

Step 3 To change the cluster size, first create an addition or deletion. Then, after nodes are added automatically, find the cluster row and select **More** > **Change Specifications** > **Change node flavor**. This opens the hybrid price page. The additional fee is the price that needs to be paid for this specification change. The configuration fee is the hourly price of the pay-per-use nodes in the cluster.

----End

Supported Specifications

Table 9-30 Supported specifications

Current Specification Name	Target Specification Name
dwsk2.xlarge	dwsk2.2xlarge, dwsk2.4xlarge, dwsk2.12xlarge, and dwsk2.8xlarge
dwsk2.2xlarge	dwsk2.12xlarge, dwsk2.8xlarge, and dwsk2.4xlarge
dwsk2.4xlarge	dwsk2.2xlarge, dwsk2.8xlarge, and dwsk2.12xlarge
dwsk2.8xlarge	dwsk2.2xlarge, dwsk2.4xlarge, and dwsk2.12xlarge
dwsk2.12xlarge	dwsk2.2xlarge, dwsk2.4xlarge, and dwsk2.8xlarge
dwsk2.h.12xlarge.4.kc1	dwsk2.h.xlarge.4.kc1, dwsk2.h.2xlarge.4.kc1, dwsk2.h.4xlarge.4.kc1, and dwsk2.h.8xlarge.4.kc1
dwsk2.h.2xlarge.4.kc1	dwsk2.h.8xlarge.4.kc1, dwsk2.h.12xlarge.4.kc1, dwsk2.h.xlarge.4.kc1, and dwsk2.h.4xlarge.4.kc1
dwsk2.h.4xlarge.4.kc1	dwsk2.h.8xlarge.4.kc1, dwsk2.h.12xlarge.4.kc1, dwsk2.h.xlarge.4.kc1, and dwsk2.h.2xlarge.4.kc1
dwsk2.h.8xlarge.4.kc1	dwsk2.h.xlarge.4.kc1, dwsk2.h.2xlarge.4.kc1, dwsk2.h.4xlarge.4.kc1, and dwsk2.h.12xlarge.4.kc1
dwsk2.h.xlarge.4.kc1	dwsk2.h.2xlarge.4.kc1, dwsk2.h.4xlarge.4.kc1, dwsk2.h.8xlarge.4.kc1, and dwsk2.h.12xlarge.4.kc1

Current Specification Name	Target Specification Name
dwsk2.h1.12xlarge.4.kc1	dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.8xlarge.4.kc1, and dwsk2.h1.2xlarge.4.kc1
dwsk2.h1.2xlarge.4.kc1	dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.8xlarge.4.kc1, and dwsk2.h1.12xlarge.4.kc1
dwsk2.h1.4xlarge.4.kc1	dwsk2.h1.8xlarge.4.kc1, dwsk2.h1.12xlarge.4.kc1, and dwsk2.h1.2xlarge.4.kc1
dwsk2.h1.8xlarge.4.kc1	dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.12xlarge.4.kc1, and dwsk2.h1.2xlarge.4.kc1
dwsk2.h1.xlarge.2.kc1	dwsk2.h1.2xlarge.4.kc1, dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.8xlarge.4.kc1, and dwsk2.h1.12xlarge.4.kc1
dwsx2.xlarge	dwsx2.2xlarge, dwsx2.4xlarge, dwsx2.8xlarge, and dwsx2.16xlarge
dwsx2.2xlarge	dwsx2.4xlarge, dwsx2.8xlarge, and dwsx2.16xlarge
dwsx2.4xlarge	dwsx2.2xlarge, dwsx2.8xlarge, and dwsx2.16xlarge
dwsx2.8xlarge	dwsx2.2xlarge, dwsx2.4xlarge, and dwsx2.16xlarge
dwsx2.16xlarge	dwsx2.2xlarge, dwsx2.4xlarge, and dwsx2.8xlarge
dwsx2.xlarge.m7	dwsx2.2xlarge.m7, dwsx2.4xlarge.m7, dwsx2.8xlarge.m7, and dwsx2.16xlarge.m7
dwsx2.2xlarge.m7	dwsx2.4xlarge.m7, dwsx2.8xlarge.m7, and dwsx2.16xlarge.m7
dwsx2.4xlarge.m7	dwsx2.2xlarge.m7, dwsx2.8xlarge.m7, and dwsx2.16xlarge.m7
dwsx2.8xlarge.m7	dwsx2.2xlarge.m7, dwsx2.4xlarge.m7, and dwsx2.16xlarge.m7
dwsx2.16xlarge.m7	dwsx2.2xlarge.m7, dwsx2.4xlarge.m7, and dwsx2.8xlarge.m7
dwsx2.xlarge.m7n	dwsx2.2xlarge.m7n, dwsx2.8xlarge.m7n, and dwsx2.16xlarge.m7n
dwsx2.2xlarge.m7n	dwsx2.8xlarge.m7n and dwsx2.16xlarge.m7n
dwsx2.8xlarge.m7n	dwsx2.2xlarge.m7n and dwsx2.16xlarge.m7n
dwsx2.16xlarge.m7n	dwsx2.2xlarge.m7n and dwsx2.8xlarge.m7n
dwsx2.h.xlarge.4.c6	dwsx2.h.2xlarge.4.c6, dwsx2.h.4xlarge.4.c6, dwsx2.h.8xlarge.4.c6, and dwsx2.h.16xlarge.4.c6
dwsx2.h.2xlarge.4.c6	dwsx2.h.4xlarge.4.c6, dwsx2.h.8xlarge.4.c6, and dwsx2.h.16xlarge.4.c6

Current Specification Name	Target Specification Name
dwsx2.h.4xlarge.4.c6	dwsx2.h.8xlarge.4.c6, dwsx2.h.16xlarge.4.c6, and dwsx2.h.2xlarge.4.c6
dwsx2.h.8xlarge.4.c6	dwsx2.h.4xlarge.4.c6, dwsx2.h.16xlarge.4.c6, and dwsx2.h.2xlarge.4.c6
dwsx2.h.16xlarge.4.c6	dwsx2.h.2xlarge.4.c6, dwsx2.h.4xlarge.4.c6, and dwsx2.h.8xlarge.4.c6
dwsx2.h.xlarge.4.c7	dwsx2.h.4xlarge.4.c7, dwsx2.h.8xlarge.4.c7, dwsx2.h.16xlarge.4.c7, and dwsx2.h.2xlarge.4.c7
dwsx2.h.2xlarge.4.c7	dwsx2.h.4xlarge.4.c7, dwsx2.h.8xlarge.4.c7, and dwsx2.h.16xlarge.4.c7
dwsx2.h.4xlarge.4.c7	dwsx2.h.2xlarge.4.c7, dwsx2.h.8xlarge.4.c7, and dwsx2.h.16xlarge.4.c7
dwsx2.h.8xlarge.4.c7	dwsx2.h.16xlarge.4.c7, dwsx2.h.2xlarge.4.c7, and dwsx2.h.4xlarge.4.c7
dwsx2.h.16xlarge.4.c7	dwsx2.h.8xlarge.4.c7, dwsx2.h.xlarge.4.c7, dwsx2.h.2xlarge.4.c7, and dwsx2.h.4xlarge.4.c7
dwsx2.h.xlarge.4.c7n	dwsx2.h.2xlarge.4.c7n, dwsx2.h.4xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, and dwsx2.h.16xlarge.4.c7n
dwsx2.h.2xlarge.4.c7n	dwsx2.h.4xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, and dwsx2.h.16xlarge.4.c7n
dwsx2.h.4xlarge.4.c7n	dwsx2.h.2xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, and dwsx2.h.16xlarge.4.c7n
dwsx2.h.8xlarge.4.c7n	dwsx2.h.16xlarge.4.c7n, dwsx2.h.2xlarge.4.c7n, and dwsx2.h.4xlarge.4.c7n
dwsx2.h.16xlarge.4.c7n	dwsx2.h.4xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, and dwsx2.h.2xlarge.4.c7n
dwsx2.h1.xlarge.2.c6	dwsx2.h1.8xlarge.4.c6, dwsx2.h1.16xlarge.4.c6, dwsx2.h1.2xlarge.4.c6, and dwsx2.h1.4xlarge.4.c6
dwsx2.h1.2xlarge.4.c6	dwsx2.h1.4xlarge.4.c6, dwsx2.h1.8xlarge.4.c6, and dwsx2.h1.16xlarge.4.c6
dwsx2.h1.4xlarge.4.c6	dwsx2.h1.2xlarge.4.c6, dwsx2.h1.8xlarge.4.c6, and dwsx2.h1.16xlarge.4.c6
dwsx2.h1.8xlarge.4.c6	dwsx2.h1.16xlarge.4.c6, dwsx2.h1.4xlarge.4.c6, and dwsx2.h1.2xlarge.4.c6
dwsx2.h1.16xlarge.4.c6	dwsx2.h1.4xlarge.4.c6, dwsx2.h1.2xlarge.4.c6, and dwsx2.h1.8xlarge.4.c6

Current Specification Name	Target Specification Name
dwsx2.h1.xlarge.2.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, dwsx2.h1.16xlarge.4.c7, and dwsx2.h1.2xlarge.4.c7
dwsx2.h1.16xlarge.4.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, and dwsx2.h1.2xlarge.4.c7
dwsx2.h1.2xlarge.4.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, and dwsx2.h1.16xlarge.4.c7
dwsx2.h1.4xlarge.4.c7	dwsx2.h1.2xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, and dwsx2.h1.16xlarge.4.c7
dwsx2.h1.8xlarge.4.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.2xlarge.4.c7, and dwsx2.h1.16xlarge.4.c7
dwsx2.h1.xlarge.2.c7n	dwsx2.h1.2xlarge.4.c7n, dwsx2.h1.4xlarge.4.c7n, dwsx2.h1.8xlarge.4.c7n, and dwsx2.h1.16xlarge.4.c7n
dwsx2.h1.2xlarge.4.c7n	dwsx2.h1.16xlarge.4.c7n, dwsx2.h1.4xlarge.4.c7n, and dwsx2.h1.8xlarge.4.c7n
dwsx2.h1.4xlarge.4.c7n	dwsx2.h1.8xlarge.4.c7n, dwsx2.h1.16xlarge.4.c7n, and dwsx2.h1.2xlarge.4.c7n
dwsx2.h1.8xlarge.4.c7n	dwsx2.h1.4xlarge.4.c7n, dwsx2.h1.16xlarge.4.c7n, and dwsx2.h1.2xlarge.4.c7n
dwsx2.h1.16xlarge.4.c7n	dwsx2.h1.2xlarge.4.c7n, dwsx2.h1.4xlarge.4.c7n, and dwsx2.h1.8xlarge.4.c7n
dwsx2.rt.xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.4xlarge.m7, dwsx2.rt.8xlarge.m7, and dwsx2.rt.16xlarge.m7
dwsx2.rt.2xlarge.m7	dwsx2.rt.4xlarge.m7, dwsx2.rt.8xlarge.m7, and dwsx2.rt.16xlarge.m7
dwsx2.rt.4xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.8xlarge.m7, and dwsx2.rt.16xlarge.m7
dwsx2.rt.8xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.4xlarge.m7, and dwsx2.rt.16xlarge.m7
dwsx2.rt.16xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.4xlarge.m7, and dwsx2.rt.8xlarge.m7
dwsk2.rt.xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.4xlarge.km1, dwsk2.rt.8xlarge.km1, and dwsk2.rt.12xlarge.km1
dwsk2.rt.2xlarge.km1	dwsk2.rt.4xlarge.km1, dwsk2.rt.8xlarge.km1, and dwsk2.rt.12xlarge.km1
dwsk2.rt.4xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.8xlarge.km1, and dwsk2.rt.12xlarge.km1

Current Specification Name	Target Specification Name
dwsk2.rt.8xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.4xlarge.km1, and dwsk2.rt.12xlarge.km1
dwsk2.rt.12xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.4xlarge.km1, and dwsk2.rt.8xlarge.km1
dwsx2.rt.xlarge.m7n	dwsx2.rt.2xlarge.m7n, dwsx2.rt.8xlarge.m7n, and dwsx2.rt.16xlarge.m7n
dwsx2.rt.2xlarge.m7n	dwsx2.rt.8xlarge.m7n and dwsx2.rt.16xlarge.m7n
dwsx2.rt.8xlarge.m7n	dwsx2.rt.2xlarge.m7n and dwsx2.rt.16xlarge.m7n
dwsx2.rt.16xlarge.m7n	dwsx2.rt.2xlarge.m7n and dwsx2.rt.8xlarge.m7n
4U32G.4DPU	8U64G.8DPU, 16U128G.16DPU, 32U256G.32DPU, 64U512G.64DPU, 96U768G.96DPU, and 128U1024G.128DPU
8U64G.8DPU	16U128G.16DPU, 32U256G.32DPU, 64U512G.64DPU, 96U768G.96DPU, and 128U1024G.128DPU
16U128G.16DPU	8U64G.8DPU, 32U256G.32DPU, 64U512G.64DPU, 96U768G.96DPU, and 128U1024G.128DPU
32U256G.32DPU	8U64G.8DPU, 16U128G.16DPU, 64U512G.64DPU, 96U768G.96DPU, and 128U1024G.128DPU
64U512G.64DPU	8U64G.8DPU, 16U128G.16DPU, 32U256G.32DPU, 96U768G.96DPU, and 128U1024G.128DPU
96U768G.96DPU	8U64G.8DPU, 16U128G.16DPU, 32U256G.32DPU, 64U512G.64DPU, 96U768G.96DPU, and 128U1024G.128DPU
128U1024G.128DPU	8U64G.8DPU, 16U128G.16DPU, 32U256G.32DPU, 64U512G.64DPU, 96U768G.96DPU, and 128U1024G.128DPU

9.9.2 Changing All Specifications

If you want to change your cluster topology or capacity but the **Change node flavor** option is grayed out, you can select **Change all specifications** to increase or decrease the nodes and their capacities on the GaussDB(DWS) console. First, you need to configure the new specifications you want, and a cluster with these specifications will be created. Then, data will be migrated from the old cluster to the new one. In case you need to restore data, a full snapshot will be taken for the old cluster, and the old cluster will be retained for a period of time.

NOTE

- To use this feature, contact technical support engineers to upgrade your version first.
- A storage-compute coupled cluster (standalone) does not support changing all specifications.
- A cluster billed in yearly/monthly mode does not support this feature.
- A storage-compute decoupled cluster does not support changing all specifications.
- The new cluster does not incur charges before the change completes. The old cluster enters the retention period and will not incur charges after the resizing completes.
- A cluster can have up to 240 nodes. The old and new clusters can have up to 480 nodes in total.
- The **Change all specifications** option do not support logical clusters.

Impact of Changing All Specifications

- Before the change, you need to exit the client connections that have created temporary tables, because temporary tables created before or during the change will become invalid and operations performed on these temporary tables will fail. The temporary tables created after the change are not affected.
- When all specifications are changed, the cluster status changes to **Read-only** during data redistribution. During the change, services may be blocked for a long time. You are advised to perform the change with the assistance of engineers to prevent services from being affected.
- After the specifications are changed, the private IP address changes, which should be updated for connection.
- After the specifications are changed, the domain name remains unchanged, and the IP address bound to the domain name is switched. During the switchover, the connection is interrupted for a short period of time. Therefore, avoid writing service statements in the switchover. If the service side uses a domain name for connection, you need to update the cache information corresponding to the domain name to prevent connection failure after the change.
- If an ELB is bound to the cluster, the connection address on the service side remains unchanged after the specifications are changed, while the internal server address of the ELB is changed to the new connection address.
- In case you need to restore data, a full snapshot will be taken for the old cluster (on condition that your cluster support snapshot creation). You can check it in the snapshot list and manually delete it if it is no longer necessary.
- During the change, the cluster is read-only, affecting intelligent O&M tasks. You are advised to start these tasks after the change or pause them before the change.

Prerequisites

- The cluster to be changed is in the **Available**, **Read-only**, or **Unbalanced** state.
- The number of nodes after resizing must be smaller than or equals the available node quotas, or the resizing fill fail.
- The total capacity of the new cluster after the change must be at least 1.2 times greater than the used capacity of the old cluster.

- To perform the change in a cluster as an IAM user, ensure that the IAM user has permissions for VPC, EVC, and BMS.

Changing All Specifications

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

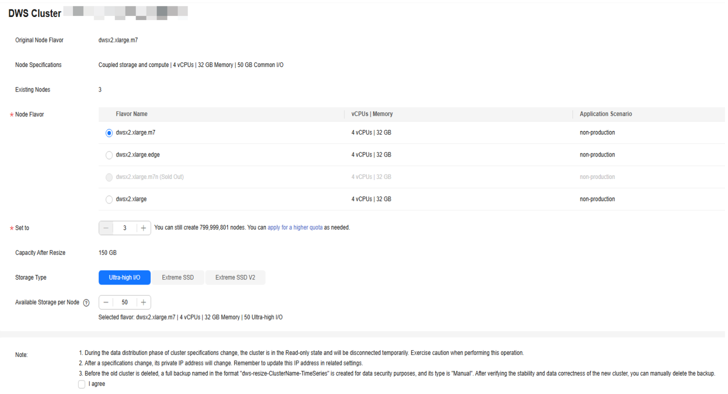
Step 3 In the row of a cluster, choose **More** > **Change Specifications** in the **Operation** column and click **Change all specifications**.

NOTE

Before the change, it is crucial to verify if it meets the inspection conditions. Click **Inspect** to complete the inspection and proceed to the next step only if it passes. For more information, see **Viewing Inspection Results**.

- For the **Node Flavor** parameter, select a flavor. The VPC, subnet, and security group of the new cluster are the same as those of the original cluster.
- For the **Set to** parameter, set the number of nodes you want for the new cluster.

Step 4 (Optional) If the cluster storage can be modified, you can set the storage type and the available storage for each node.



DWS Cluster [redacted]

Original Node Flavor: dwsx2.xlarge.m7

Node Specifications: Coupled storage and compute | 4 vCPUs | 32 GB Memory | 50 GB Common I/O

Existing Nodes: 3

* Node Flavor

	Flavor Name	vCPUs | Memory	Application Scenario
●	dwsx2.xlarge.m7	4 vCPUs | 32 GB	non-production
○	dwsx2.xlarge.edge	4 vCPUs | 32 GB	non-production
○	dwsx2.xlarge.m7n (Sold Out)	4 vCPUs | 32 GB	non-production
○	dwsx2.xlarge	4 vCPUs | 32 GB	non-production

* Set to: − 3 + You can still create 799,999,801 nodes. You can apply for a higher quota as needed.

Capacity After Resize: 150 GB

Storage Type: [✓ Ultra-high I/O] Extreme SSD Extreme SSD V2

Available Storage per Node: − 50 +

Selected flavor: dwsx2.xlarge.m7 | 4 vCPUs | 32 GB Memory | 50 Ultra-high I/O

Note:

1. During the data distribution phase of cluster specifications change, the cluster is in the Read-only state and will be disconnected temporarily. Exercise caution when performing this operation.
2. After a specifications change, its private IP address will change. Remember to update this IP address in related settings.
3. Before the old cluster is deleted, a full backup named in the format "dws-resize-ClusterName-TimeSeries" is created for data security purposes, and its type is "Manual". After verifying the stability and data correctness of the new cluster, you can manually delete the backup.

□ I agree

NOTE

Extreme SSD and **Extreme SSD V2** can only be selected for clusters that use ECS for computing and EVS for storage.

Step 5 Read the nodes and select **Confirmed**. Click **Resize Cluster Now**.

Step 6 Click **Submit**.

- After the change request is submitted, **Task Status** of the cluster changes to **Changing all specifications**. The process will take several minutes.
- During the change, the cluster automatically restarts, and **Cluster Status** is **Unavailable** for a period of time. After the restart is complete, **Cluster Status** changes to **Available**. Data is redistributed during resizing. During the redistribution, **Cluster Status** is **Read-only**.
- The resizing succeeds only when **Cluster Status** is **Available** and the **Change all specifications** task in **Task Information** is complete. Then the cluster begins providing services.

- If **Change all specifications failed** is displayed, the cluster failed to be changed.
- If change fails, and a message requiring retry is displayed when you click **Resize**, the failure is probably caused by abnormal cluster status or network problems. In this case, contact technical support to troubleshoot the problem and try again.

----End

9.9.3 Disk Capacity Expansion of an EVS Cluster

Context

As customer services evolve, disk space often becomes the initial bottleneck. In scenarios where other resources are ample, the conventional scale-out process is not only time-consuming but also resource-inefficient. Disk capacity expansion can quickly increase storage without service interruption. You can expand the disk capacity when no other services are running. If the disk space is insufficient after the expansion, you can continue to expand the disk capacity. If the expansion fails, you can expand the disk capacity again.

NOTE

- Disk capacity expansion can be performed only for storage-compute coupled data warehouses with cloud SSDs. Only version 8.1.1.203 and later are supported.
- Disk capacity can be expanded only if the cluster is in **Available**, **To be restarted**, **Read-only**, or **Node fault**, **Unbalanced** state.

Precautions

- Hot storage disks cannot be scaled down.
- Scale up hot data storage during off-peak hours.
- If the cluster is in the read-only state, a message will be displayed after you click **Expand Disk Capacity**. After you start expansion, wait until it is completed and the cluster changes to the available state.
- The added disks of a yearly/monthly cluster are billed in yearly/monthly mode by default.

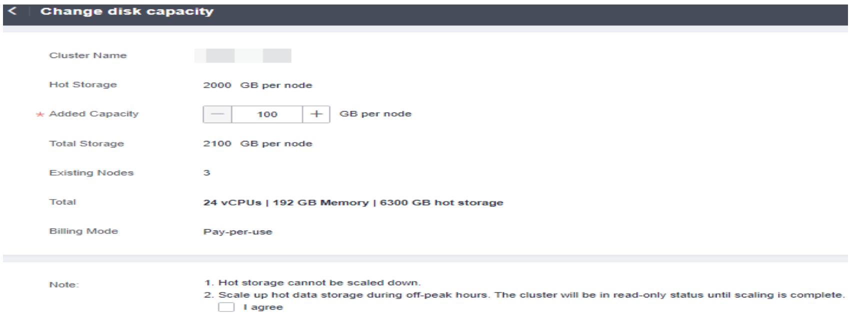
Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Change Specifications** and click **Change disk capacity**. The **Expand Disk Capacity** page is displayed.

Step 4 Set the capacity and click **Resize Cluster Now**.



Change disk capacity

Cluster Name	
Hot Storage	2000 GB per node
* Added Capacity	100 GB per node
Total Storage	2100 GB per node
Existing Nodes	3
Total	24 vCPUs | 192 GB Memory | 6300 GB hot storage
Billing Mode	Pay-per-use

Note:
1. Hot storage cannot be scaled down.
2. Scale up hot data storage during off-peak hours. The cluster will be in read-only status until scaling is complete.

□ I agree

NOTE

Hot Storage is changed to **Hot Storage (with Cache)** for storage-compute decoupled clusters.

Step 5 Confirm the settings and click **Submit**.

Step 6 Return to the cluster list and check the disk capacity expansion progress.

----End

Hybrid Billing for Disk Capacity Expansion of a Yearly/Monthly Cluster

Prerequisites

During peak hours, nodes are automatically added to a yearly/monthly-billed cluster as scheduled. Nodes are billed on a pay-per-use basis. For details, see **Elastically Adding or Deleting a Logical Cluster**. After scale-out, the cluster uses the hybrid billing mode, that is, both the pay-per-use and yearly/monthly billing modes are used. Nodes created by creating addition or deletion plans are pay-per-use nodes, and nodes created by creating a yearly/monthly cluster are yearly/monthly nodes.

NOTE

Only storage-compute decoupled clusters support hybrid billing.

Procedure

Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 Click **Create GaussDB(DWS) Cluster** in the upper right corner of the page to **create a storage-compute decoupled cluster**.

Step 3 After the cluster is created, add an addition or deletion plan. After nodes are automatically added, if you need to expand the disk capacity, locate the row that contains the cluster and choose **More** > **Change Specifications** > **Change disk capacity** in the **Operation** column. The hybrid price page is displayed. The supplementary fee is the price that needs to be paid for the disk scale-out of yearly/monthly nodes. The configuration fee is the hourly price of pay-per-use nodes in the cluster after disk scale-out.

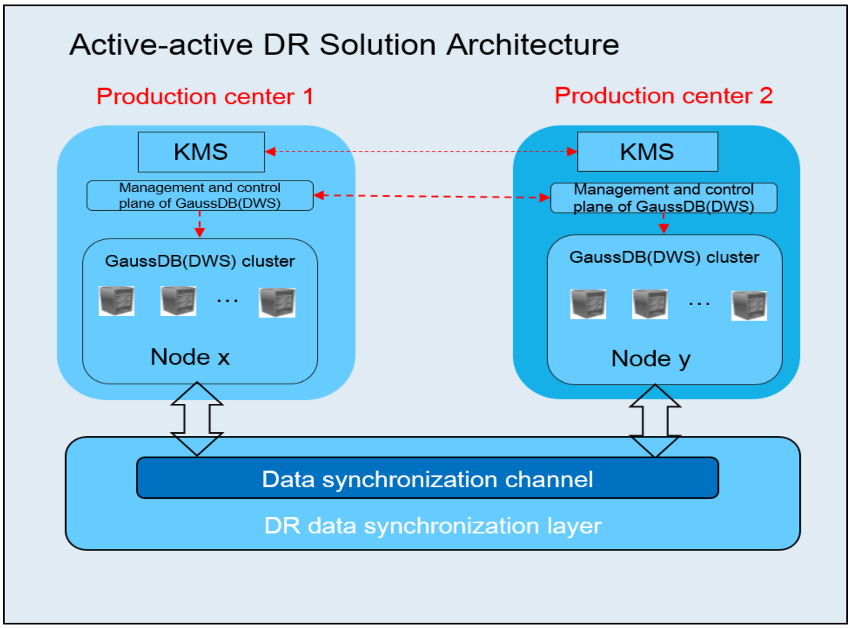
----End

9.10 GaussDB(DWS) Cluster DR Management

9.10.1 GaussDB(DWS) Cluster DR Scenarios

Overview

A homogeneous GaussDB(DWS) disaster recovery (DR) cluster is deployed in the same region. If the production cluster fails to provide read and write services due to natural disasters in the specified region or cluster internal faults, the DR cluster becomes the production cluster to ensure service continuity. The following figure shows the architecture.


Active-active DR Solution Architecture
Production center 1
Production center 2
KMS
KMS
Management and control plane of GaussDB(DWS)
Management and control plane of GaussDB(DWS)
GaussDB(DWS) cluster
GaussDB(DWS) cluster
Node x
Node y
Data synchronization channel
DR data synchronization layer

NOTE

- Intra-region DR is supported only in cluster version 8.1.1 and later.
- A storage-compute coupled data warehouse (standalone) does not support DR.
- Storage-compute decoupled clusters and multi-AZ clusters do not support DR.
- If you use a yearly/monthly package for a DR cluster, the cluster will be automatically frozen for a period of time after the package expires, and will be deleted if its subscription is not renewed in time. Be sure to renew it in a timely manner to prevent DR exceptions caused by deletion of the DR cluster.

DR Features

- Multi-form DR
  - Intra-region DR
  - Multiple data synchronization modes: synchronization layer based on mutual trust
- Low TCO
  - Heterogeneous deployment (logical homogeneity)

  - Cluster-level DR
- Visual console

  Automatic and one-click DR drills

Constraints and Limitations

- During data synchronization, a non-fine-grained DR cluster cannot provide read or write services.
- When the DR task is stopped or abnormal but the DR cluster is normal, the DR cluster can provide the read service. After the DR switchover is successful, the DR cluster can provide the read and write services.
- When the DR task is created, the snapshot function of the production cluster is normal, but that of the DR cluster is disabled. Besides, snapshot restoration of both clusters is disabled.
- Logical clusters are not supported.
- Resource pools are not supported.
- If cold and hot tables are used, cold data is synchronized using OBS.
- DR does not synchronize data from external sources.
- DR management refers to dual-cluster DR under the same tenant.
- The DR cluster and the production cluster must be logically homogeneous and in the same type and version.
- The production cluster and DR cluster used for intra-region DR must be in the same VPC.
- In intra-region DR, after services are switched over from the production cluster to the DR cluster, the bound ELB is automatically switched to the new production cluster. During the switchover, the connection is interrupted for a short period of time. Do not run service statements to write data during the switchover.
- During intra-region DR, the EIP, intranet domain name, and connection IP address of the original production cluster are not automatically switched with the cluster switchover. The EIP, domain name, or IP address used for connection in the service system need to be switched to the new cluster.

9.10.2 Creating and Starting DR for a GaussDB(DWS) Cluster

Creating an Intra-Region Cluster-Level DR Task

Prerequisites

You can create a DR task only when the cluster is in the **Available** or **Unbalanced** state.

Procedure

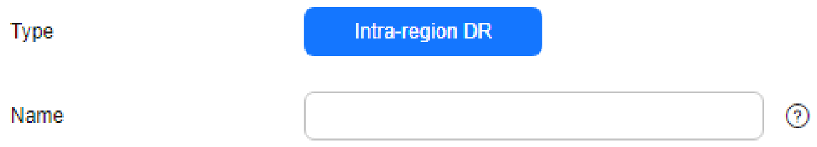
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 On the displayed page, click **Create**.

Step 4 Select the type and enter the name of the DR task to be created.

- **Type**: **Intra-region DR**
- **Name**: Enter 4 to 64 case-insensitive characters, starting with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.



Type: Intra-region DR

Name:

Step 5 Configure the production cluster.

- Select a created production cluster from the drop-down list.
- After a production cluster is selected, the system automatically displays its AZ.

Step 6 Configure the DR cluster.

- Select the AZ associated with the region where the DR cluster resides.

  NOTE

  The AZ of the DR cluster can be the same as that of the production cluster. In a 3-AZ cluster, any of the three AZs can be selected for DR.

- **Cluster Name**: Upon selecting an AZ for the DR cluster, the system will auto-filter DR clusters that fulfill the logical homogeneity criteria. Should there be no qualifying DR clusters, you can click **Create DR Cluster** to create a DR cluster with the same configuration as the production cluster.

Step 7 Configure advanced parameters. Select **Default** to keep the default values of the advanced parameters. You can also select **Custom** to modify the values.

- The DR synchronization period indicates the interval for synchronizing incremental data from the production cluster to the DR cluster. Set this parameter based on the actual service data volume.

  NOTE

  The default DR synchronization period is 30 minutes.

Step 8 Click **OK**.

The DR status will then change to **Creating**. Wait until the creation is complete, and the DR status will change to **Not Started**.

----End

Starting a DR Task

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 Click **Start** in the **Operation** column of the target DR task.

Step 4 In the dialog box that is displayed, click **OK**.

The DR status will change to **Starting**. The process will take some time. After the task is started, the DR status will change to **Running**.

NOTE

- You can start a DR task that is in the **Not started**/**Startup failed**/**Stopped** state.
- After you start the DR task, you cannot perform operations such as restoration, scale-out, upgrade, restart, node replacement, and password update, on the production cluster or DR cluster, and backup is also not allowed on the DR cluster. Exercise caution when performing this operation.
- After the DR task is started, if the DR cluster is running properly and DR recovery is in progress, the cluster will be billed.

----End

Viewing DR Information

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 In the DR list, click the name of a DR task.

On the page that is displayed, view the following information:

- **DR Information**: You can view the DR ID, DR name, DR type, DR creation time, DR start time, and DR status.
- **Production Cluster Information**: You can view the production cluster ID, cluster name, AZ, used storage capacity, cluster DR status, and the time of the latest successful DR task.
- **DR Cluster Information**: You can view the DR cluster ID, cluster name, AZ, used storage capacity, cluster DR status, and the time of the latest successful DR task.
- **DR Configuration**: You can view and modify the DR synchronization period.

----End

Updating DR Configurations

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 In the DR list, click the name of a DR task.

Step 4 In the **DR Configurations** area, click **Modify**.

NOTE

- Only DR tasks in the **Not started** or **Stopped** state can be modified.
- The new configuration takes effect after DR is restarted.

----End

Case 1: How Do I Scale out a Cluster in the DR State?

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, if **Task Information** of the cluster you want to scale out is **DR not started**, perform **Step 5** and **Step 7**.

Step 4 (Optional) If the **Task Information** is other than **DR not started**, delete the DR task. For details, see **Deleting a DR Task**.

Step 5 In the **Operation** column of the production and DR clusters, choose **More** > **Scale Out**.

Step 6 Create a DR task. For details, see **Creating and Starting DR for a GaussDB(DWS) Cluster**.

Step 7 Start the DR task. For details, see **Starting a DR Task**.

NOTE

After scale-out, the number of DNs in the production cluster must be the same as that in the DR cluster.

----End

9.10.3 Performing a DR Switchover for the GaussDB(DWS) Cluster

Switching to the DR Cluster

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 Click **Switch to DR Cluster** in the **Operation** column of the target DR task.

Step 4 In the dialog box that is displayed, click **OK**.

The DR status will change to **DR switching**.

After the switchover is successful, the DR status will change to the original status.

NOTE

- To perform a switchover when the DR cluster is running properly, click **Switch to DR Cluster**.
- You can perform a DR switchover when the DR task is in the **Running** state.
- During a switchover, the original production cluster is not available.
- The Recovery Point Object (RPO, time point to which the system and data must be recovered after a disaster occurs) in different DR switchover scenarios is described as follows:
  - Production cluster in the **Available** state: RPO = 0
  - Production cluster in the **Unavailable** state: A zero RPO may not be achieved, but data can at least be restored to that of the latest successful DR synchronization (**Last DR Succeeded**). For details, see **Viewing DR Information**.

----End

Exception Switchover

Scenario

The production cluster is unavailable, the DR cluster is normal, and the DR status is **Abnormal**.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 Choose **More** > **Exception Switchover** in the **Operation** column of the target DR task.

Step 4 In the dialog box that is displayed, click **OK**.

The **Status** will change to **Switchover in progress**.

After the switchover is successful, the DR status will change to the original status. In this procedure, the DR status will change back to **Abnormal**.

NOTE

- To perform a switchover when the DR cluster is abnormal or the production cluster is faulty, click **Exception Switchover**.
- DR exception switchover is supported only by clusters of version 8.1.2 or later.
- Before a switchover, check the latest synchronization time in the DR cluster. The DR cluster will serve as a production cluster after an abnormal switchover, but the data that failed to be synchronized from the original production cluster to the DR cluster will not exist in the DR cluster.
- If the DR type is **Cross-region DR**, the switchover can be performed only in the region where the standby cluster is located.

----End

Performing a DR Switchback

Scenario

After abnormal switchover, if you have confirmed that the original production cluster was recovered, you can perform a switchback.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 Click **DR Recovery** in the **Operation** column of a DR task.

Step 4 In the displayed dialog box, set **Synchronization Mode** to **Incremental** or **Full**.

NOTE

You are advised to set **Synchronization Mode** to **Incremental** when updating a DR creation task.

Step 5 Click **OK**.

The **Status** will change to **Recovering**.

After the DR recovery is successful, the **Status** will change to **Running**.

NOTE

- DR is supported only by clusters of 8.1.2 or later.
- During DR recovery, data in the DR cluster will be deleted, and the DR relationship will be re-established with the new production cluster.
- If the DR type is **Cross-region DR**, the recovery can be performed only in the region where the standby cluster is located.

----End

9.10.4 Stopping and Deleting DR for a GaussDB(DWS) Cluster

Stopping the DR Task

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 Click **Stop** in the **Operation** column of the target DR task.

Step 4 In the dialog box that is displayed, click **OK**.

The DR status will change to **Stopping**. The process will take some time. After the DR task is stopped, the status will change to **Stopped**.

NOTE

- Only DR tasks in the **Running** or **Stop failed** state can be stopped.
- Data cannot be synchronized after a DR task is stopped.

----End

Deleting a DR Task

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane, choose **Management** > **DR Tasks**.

Step 3 Click **Delete** in the **Operation** column of the target DR task.

Step 4 In the dialog box that is displayed, click **OK**.

The DR status will change to **Deleting**.

NOTE

- You can delete a DR task when **DR Status** is **Creation failed**, **Not started**, **Startup failed**, **Stopped**, **Stop failed**, or **Abnormal**.
- Data cannot be synchronized after a DR task is deleted, and the deleted task cannot be restored.

----End

9.11 Upgrading a GaussDB(DWS) Cluster

GaussDB(DWS) allows users to upgrade clusters on the console. For details, see **Upgrading a Cluster**.

During cluster O&M operations, GaussDB(DWS) will send SMS notifications to keep you informed. It is important to be careful when performing any operations on the cluster during this time.

If a node needs to be replaced due to a hardware fault, the repairCluster event will be triggered. You can check the event progress by **Subscribing to Event Notifications**.

During the upgrade, the cluster will be restarted and cannot provide services for a short period of time.

A cluster is charged by hour or in yearly/monthly mode as long as it is in the **Available** state, so you will not see any difference in the bills if a faulty node or system upgrade causes a short interruption, for example, 15 minutes. If such events cause major system interruption, which is a very rare case, you will not be charged for those downtime hours.

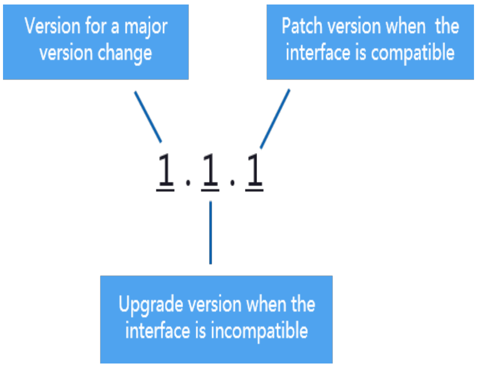
NOTE

- After a cluster is upgraded to 8.1.3 or later, it enters the observation period. During this period, you can check service status and roll back to the earlier version if necessary.
- Upgrading the cluster from 9.0.3 to 9.1.0 changes the **disk_cache_base_paths** value for the cache path, which brings the performance back to normal.
- Upgrading the cluster does not affect the original cluster data or specifications.

Upgrade Version Description

The following figure shows the cluster version.

Figure 9-40 Version description


Version for a major version change
Patch version when the interface is compatible
1.1.1
Upgrade version when the interface is incompatible

- **Service patch upgrade**: The last digit of cluster version *X.X.X* is changed. For example, the cluster is upgraded from 1.1.0 to 1.1.1.
  - Duration: The whole process will take less than 10 minutes.
  - Impact on services: During this period, if the source version is upgraded to 8.1.3 or later, online patching is supported. During the patch upgrade, you do not have to stop services, but the services will be intermittently interrupted for seconds. If the destination version is earlier than 8.1.3, services will be interrupted for 1 to 3 minutes. Therefore, you are advised to perform this operation during off-peak hours.
- **Service upgrade**: The first two digits of cluster version *X.X.X* are changed. For example, the cluster is upgraded from 1.1.0 to 1.2.0.

  - Duration: The whole process will take less than 30 minutes.
  - Impact on services: Online upgrade is supported for update to 8.1.1 or later. During the upgrade, you are not required to stop services, but services are intermittently interrupted for seconds. You are advised to perform the upgrade during off-peak hours.
- Hot patch upgrade: A hot patch upgrade involves adding a one-digit version number (in the format of **0001-9999**) to the current cluster version.
  - Duration: The upgrade of a single hot patch takes less than 10 minutes.
  - Impact on services: The hot patch upgrade will not affect services, but there is a chance that the issues resolved by the current hot patch may come back after it is uninstalled.

Upgrading a Cluster

Prerequisites

Cluster 8.1.1 and later versions allow users to deliver cluster upgrade operations on the console.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

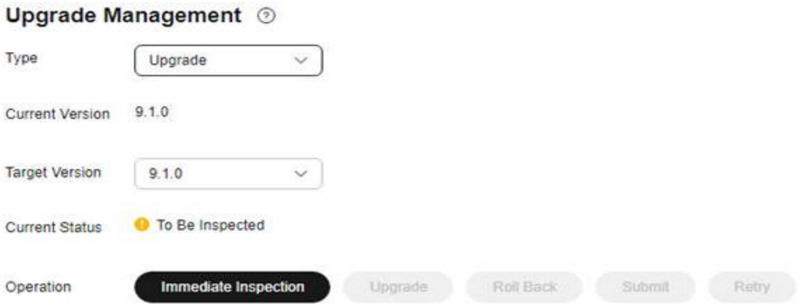
Step 2 In the cluster list, click the name of a cluster.

Step 3 In the navigation pane, choose **Upgrade Management**.

Step 4 Choose either **Upgrade** or **Hot patch** from the **Type** drop-down list depending on the type of upgrade you want to perform.

Step 5 On the **Upgrade Management** page, select a version from the **Target Version** drop-down list.

Figure 9-41 Upgrading a cluster



Upgrade Management

Type: Upgrade

Current Version: 9.1.0

Target Version: 9.1.0

Current Status: To Be Inspected

NOTE

- Before the upgrade, it is crucial to verify if it meets the inspection conditions. Click **Immediate Inspection** to complete the inspection and proceed to the next step only if it passes. For more information, see **Viewing Inspection Results**.
- DR cannot be established after a hot patch is installed in a cluster.

Step 6 Click **Upgrade**. Click **OK** in the displayed dialog box.

Step 7 Check whether the cluster is successfully upgraded.

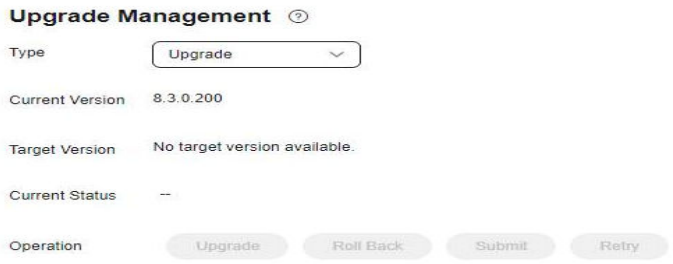
- If the cluster version is 8.1.3 or later, the cluster enters the service observation period after the upgrade is complete. If you have verified your services, click **Submit** on the **Upgrade Management** page to complete the cluster upgrade. If you find your cluster performance affected by the upgrade, you can click **Rollback** to roll back the upgrade.

  NOTE

  – If you are using a version of 8.1.3 or earlier, you will not be able to roll back or submit upgrade tasks until the cluster upgrade is finished.

  – After an upgrade task is successfully delivered, if the upgrade task is not submitted, the **wlm** thread occupies the system storage space and affects the system performance.

  Figure 9-42 Cluster upgrade success



Upgrade Management

Type: Upgrade

Current Version: 8.3.0.200

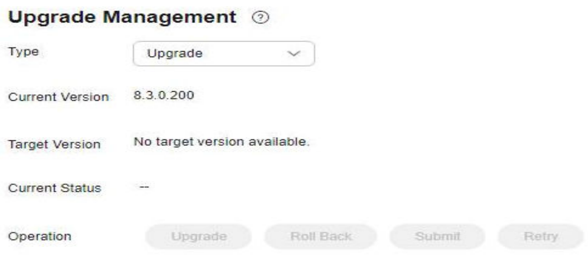
Target Version: No target version available.

Current Status: --

Operation

- If the cluster upgrade fails, click **Rollback** to roll back to the original cluster version, or click **Retry** to deliver the upgrade again.

  Figure 9-43 Cluster upgrade failure



Upgrade Management

Type: Upgrade

Current Version: 8.3.0.200

Target Version: No target version available.

Current Status: --

Operation

----End

9.12 GaussDB(DWS) Cluster Log Management

9.12.1 Log Types Supported by GaussDB(DWS) Clusters

GaussDB(DWS) provides database audit logs, management console audit logs, and other logs for users to query service logs, analyze problems, and learn product security and performance.

Database Audit Logs

If the **Security** function is enabled, GaussDB(DWS) records any DML and DDL operations performed by the database. You can locate and analyze faults based on the database audit logs, and perform behavior analysis and security auditing on historical database operations to improve GaussDB(DWS) security.

For how to enable and view database audit logs, see **Viewing GaussDB(DWS) Database Audit Logs**.

Management Console Audit Logs

GaussDB(DWS) uses Cloud Trace Service (CTS) to record mission-critical operations performed on the GaussDB(DWS) console, such as cluster creation, snapshot creation, cluster scale-out, and cluster restart. The logs can be used in purposes such as security analysis, compliance audit, resource tracing, and fault locating.

For how to enable and view management console audit logs, see **Viewing Operation Logs on the GaussDB(DWS) Console**.

Other Logs

GaussDB(DWS) interconnects with Log Tank Service (LTS). You can view collected cluster logs or dump logs on the LTS console.

The following log types are supported: CN logs, DN logs, OS messages logs, audit logs, cms logs, gtm logs, Roach client logs, Roach server logs, upgrade logs, and scale-out logs.

9.12.2 Dumping GaussDB(DWS) Database Audit Logs

GaussDB(DWS) records information (audit logs) about connections and user activities in your database. These audit logs help you monitor the database to ensure security, rectify faults, and locate historical operation records. GaussDB(DWS) saves audit logs in the database, but they can be viewed outside of it by dumping them to OBS. It is worth mentioning that the audit log dump and kernel audit log dump functions can be enabled or disabled independently. With the kernel audit log dump feature, audit logs stored in the database can be dumped directly to OBS.

NOTE

- This function cannot be used if OBS is not available.
- Only 9.1.0.100 and later versions support kernel log dump.
- Data may during cluster specifications change, CN addition, or CN deletion. You are advised to disable audit log dump during these operations.
- If a CN node is faulty, data on the CN node may be lost.
- After audit log dumping is enabled, audit logs will be dumped if the size of saved audit logs exceeds 1 GB. This may cause abnormal query results. Exercise caution when performing this operation.
- Version support for the audit log dump directory partition is as follows:
  - For 8.1.3.x clusters, only 8.1.3.322 and later versions support this feature. For 8.2.0.x clusters, only 8.2.0.106 and later versions support this feature. By default, the audit log dump directory partition is enabled and cannot be disabled.
  - To use this feature in earlier versions, contact technical support to upgrade your cluster first. Manually enable this feature after the upgrade.

Prerequisites

After a GaussDB(DWS) cluster is created, you can enable log dump for it to dump audit logs to OBS. **Before enabling audit log dump, ensure the following conditions are met:**

You have created an OBS bucket for storing the audit logs. For details, see "Managing Buckets > Creating a Bucket" in the *Object Storage Service Console Operation Guide*.

Enabling Audit Log Dumps

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

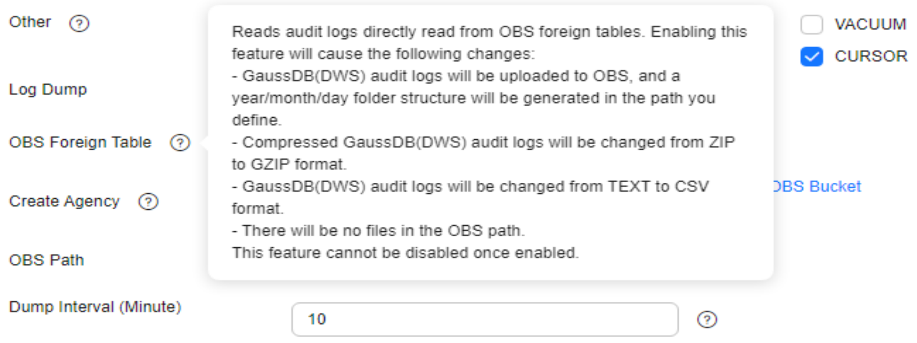
Step 3 In the cluster list, click the name of the cluster for which you want to enable audit log dump. In the navigation pane, choose **Security Settings**.

Step 4 In the **Audit Settings** area, enable **Audit Log Dump**.

When you enable audit log dump for a project in a region for the first time, the system prompts you to create an agency named **DWSAccessOBS**. After the agency is created, GaussDB(DWS) can dump audit logs to OBS.

By default, only Huawei Cloud accounts or users with **Security Administrator** permissions can query and create agencies. IAM users under an account do not have the permission to query or create agencies by default. Contact a user with that permission and complete the authorization on the current page.

Figure 9-44 Enabling Audit Log Dumps



Other ⓘ

□ VACUUM

☑ CURSOR

Log Dump

OBS Foreign Table ⓘ

Create Agency ⓘ

OBS Path

Dump Interval (Minute) 10 ⓘ

Reads audit logs directly read from OBS foreign tables. Enabling this feature will cause the following changes:
- GaussDB(DWS) audit logs will be uploaded to OBS, and a year/month/day folder structure will be generated in the path you define.
- Compressed GaussDB(DWS) audit logs will be changed from ZIP to GZIP format.
- GaussDB(DWS) audit logs will be changed from TEXT to CSV format.
- There will be no files in the OBS path.
This feature cannot be disabled once enabled.

- **OBS Foreign Table**: Audit logs can be read using OBS foreign tables during dumping. Audit logs are stored in CSV format and compressed in GZ format.
- **OBS Bucket**: Name of the OBS bucket used to store the audit data. If no OBS bucket is available, click **View OBS Bucket** to access the OBS console and create one. For details, see "Managing Buckets" > "Creating a Bucket" in the *Object Storage Service Console Operation Guide*.
- **OBS Path**: User-defined directory on OBS for storing audit files. Different directory levels are separated by forward slashes (/). The value is a string containing 1 to 50 characters, which cannot start with a forward slash (/). If the entered OBS path does not exist, the system creates one and dumps data to it.

- **Dump Interval (Minute)**: Interval based on which GaussDB(DWS) periodically dumps data to OBS. The value range is 5 to 43200. The unit is minute.

Step 5 Click **Apply**.

If **Configuration Status** is **Applying**, the system is saving the settings.

When the status changes to **Synchronized**, the configurations are saved and take effect.

----End

Enabling Kernel Audit Log Dump

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Dedicated Clusters** > **Clusters** in the navigation pane.

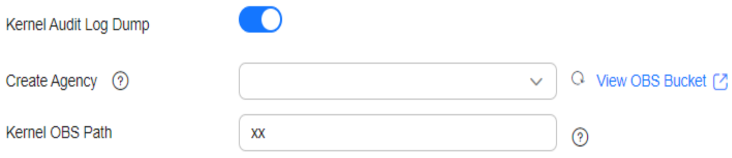
Step 3 Click the name of the cluster for which you want to enable kernel log dump. Choose **Security**.

Step 4 In the **Audit Settings** area, enable **Kernel Audit Log Dump**.

When you enable the kernel audit log dump feature for a project in a region for the first time, the system prompts you to create an agency named **DWSAccessOBS**. After the agency is created, GaussDB(DWS) can dump audit logs to OBS.

By default, only Huawei Cloud accounts or users with Security Administrator permissions can query and create agencies. IAM users under an account do not have the permission to query or create agencies by default. Contact a user with that permission and complete the authorization on the current page. For details, see **Allowing GaussDB(DWS) to Manage Resources**.

Figure 9-45 Enabling Kernel Audit Log Dump



Kernel Audit Log Dump

Create Agency:

Kernel OBS Path: xx

- **OBS Bucket**: name of the OBS bucket for storing kernel audit data. If no OBS bucket is available, click **View OBS Bucket** to access the OBS console and create one. For details, see "Managing Buckets" > "Creating a Bucket" in the *Object Storage Service Console Operation Guide*.
- **Kernel OBS Path**: user-defined directory for storing kernel logs on OBS. Different directory levels are separated by forward slashes (/). The value is a string containing 1 to 50 characters, which cannot start with a forward slash (/). If the entered OBS path does not exist, the system creates one and dumps data to it.

Step 5 Click **Apply**.

If **Configuration Status** is **Applying**, the system is saving the settings.

When the status changes to **Synchronized**, the configurations are saved and take effect.

Step 6 After the kernel audit log dump function is enabled, you can use the **pg_query_audit** function to view the dumped logs. For details, see **Using Functions to View Database Audit Logs**.

Alternatively, select the OBS bucket and folder where logs are stored to view the log files. For details, see **Step 6**.

----End

Modifying Audit Log Dump Configurations

After audit log dump is enabled, you can modify the dump configuration. For example, you can modify the OBS bucket and path for storing logs and the dump period.

The procedure is as follows:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of the cluster for which you want to modify the audit log dump configurations. In the navigation pane, choose **Security**.

Step 4 In the **Audit Settings** area, modify the **Audit Log Dump** configurations.

Step 5 Click **Apply**.

If **Configuration Status** is **Applying**, the system is saving the settings.

When the status changes to **Synchronized**, the configurations are saved and take effect.

----End

Viewing Dumped Audit Logs

After audit log dump is enabled, you can view the dumped audit logs on OBS.

To view dumped audit logs, perform the following steps:

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of the target cluster for which you want to view the log dump history. In the navigation pane, choose **Security**.

Step 4 In the **Audit Settings** area, click **View Dump Record**.

Step 5 In the **Audit Log Dump Records** dialog box, click **View OBS Bucket**. The OBS console page is displayed.

Step 6 Select the OBS bucket and folder where the logs are stored to view the log files.

You can download and decompress the files to view. The fields of audit log files are described as follows:

Table 9-31 Log file fields

Field	Type	Description
begintime	timestamp with time zone	Operation start time.
endtime	timestamp with time zone	Operation end time.
operation_type	text	Operation type. For details, see **Table 9-32**.
audit_type	text	Audit type. For details, see **Table 9-33**.
result	text	Operation result.
username	text	Name of the user who performs the operation.
database	text	Database name.
client_conninfo	text	Client connection information, that is, gsql, JDBC, or ODBC.
object_name	text	Object name.
object_details	text	Operation object details.
command_text	text	Command used to perform the operation.
detail_info	text	Operation details.
transaction_xid	text	Transaction ID.
query_id	text	Query ID.
node_name	text	Node name.
thread_id	text	Thread ID.
local_port	text	Local port.
remote_port	text	Remote port.
result_rows	text	Number of rows in the operation result.
error_code	text	Error code.

Table 9-32 operation_type: operation types

Operation Type	Description
audit_switch	Indicates that the operations of enabling and disabling the audit log function are audited.

Operation Type	Description
login_logout	Indicates that user login and log-out operations are audited.
system	Indicates that the system startup, shutdown, and instance switchover operations are audited.
sql_parse	Indicates that SQL statement parsing operations are audited.
user_lock	Indicates that user locking and unlocking operations are audited.
grant_revoke	Indicates that user permission granting and revoking operations are audited.
violation	Indicates that user's access violation operations are audited.
ddl	Indicates that DDL operations are audited. DDL operations are controlled at a fine granularity based on operation objects. Therefore, **audit_system_object** is used to control the objects whose DDL operations are to be audited. (The audit function takes effect as long as **audit_system_object** is configured, no matter whether **ddl** is set.)
dml	Indicates that the DML operations are audited.
select	Indicates that the **SELECT** operations are audited.
internal_event	Indicates that internal incident operations are audited.
user_func	Indicates that operations related to user-defined functions, stored procedures, and anonymous blocks are audited.
special_func	Indicates that special function invoking operations are audited. Special functions include **pg_terminate_backend** and **pg_cancel_backend**.
copy	Indicates that the **COPY** operations are audited.
set	Indicates that the **SET** operations are audited.
transaction	Indicates that transaction operations are audited.
vacuum	Indicates that the **VACUUM** operations are audited.
analyze	Indicates that the **ANALYZE** operations are audited.
cursor	Indicates that cursor operations are audited.
anonymous_block	Indicates that the anonymous block operations are audited.
explain	Indicates that the **EXPLAIN** operations are audited.

Operation Type	Description
show	Indicates that the **SHOW** operations are audited.
lock_table	Indicates that table lock operations are audited.
comment	Indicates that the **COMMENT** operations are audited.
preparestmt	Indicates that the **PREPARE, EXECUTE**, and **DEALLOCATE** operations are audited.
cluster	Indicates that the **CLUSTER** operations are audited.
constraints	Indicates that the **CONSTRAINTS** operations are audited.
checkpoint	Indicates that the **CHECKPOINT** operations are audited.
barrier	Indicates that the **BARRIER** operations are audited.
cleanconn	Indicates that the **CLEAN CONNECTION** operations are audited.
seclabel	Indicates that security label operations are audited.
notify	Indicates that the notification operations are audited.
load	Indicates that the loading operations are audited.

Table 9-33 audit_type parameters

Parameter	Description
audit_open/audit_close	Indicates that the audit type is operations enabling or disabling audit logs.
user_login/user_logout	Indicates that the audit type is operations and users with successful login/logout.
system_start/system_stop/system_recover/system_switch	Indicates that the audit type is system startup, shutdown, and instance switchover.
sql_wait/sql_parse	Indicates that the audit type is SQL statement parsing.
lock_user/unlock_user	Indicates that the audit type is successful user locking and unlocking.
grant_role/revoke__role	Indicates that the audit type is user permission granting and revoking.
user_violation	Indicates that the audit type is unauthorized user access operations.

Parameter	Description
ddl_*database_object*	Indicates that successful DDL operations are audited. DDL operations are controlled at a fine granularity based on operation objects. So, **audit_system_object** is used to control the objects whose DDL operations are to be audited. (The audit function takes effect as long as **audit_system_object** is configured, no matter whether **ddl** is set.) For example, **ddl_sequence** indicates that the audit type is sequence-related operations.
dml_action_insert/ dml_action_delete/ dml_action_update/ dml_action_merge/ dml_action_select	Indicates that the audit type is DML operations such as **INSERT**, **DELETE**, **UPDATE**, and **MERGE**.
internal_event	Indicates that the audit type is internal events.
user_func	Indicates that the audit type is user-defined functions, stored procedures, or anonymous block operations.
special_func	Indicates that the audit type is special function invocation. Special functions include **pg_terminate_backend** and **pg_cancel_backend**.
copy_to/copy_from	Indicates that the audit type is **COPY** operations.
set_parameter	Indicates that the audit type is **SET** operations.
trans_begin/trans_commit/ trans_prepare/ trans_rollback_to/ trans_release/trans_savepoint/ trans_commit_prepare/ trans_rollback_prepare/ trans_rollback	Indicates that the audit type is transaction-related operations.
vacuum/vacuum_full/ vacuum_merge	Indicates that the audit type is **VACUUM** operations.
analyze/analyze_verify	Indicates that the audit type is **ANALYZE** operations.
cursor_declare/cursor_move/ cursor_fetch/cursor_close	Indicates that the audit type is cursor-related operations.
codeblock_execute	Indicates that the audit type is anonymous blocks.

Parameter	Description
explain	Indicates that the audit type is **EXPLAIN** operations.
show	Indicates that the audit type is **SHOW** operations.
lock_table	Indicates that the audit type is table locking operations.
comment	Indicates that the audit type is **COMMENT** operations.
prepare/execute/deallocate	Indicates that the audit type is **PREPARE**, **EXECUTE**, or **DEALLOCATE** operations.
cluster	Indicates that the audit type is **CLUSTER** operations.
constraints	Indicates that the audit type is **CONSTRAINTS** operations.
checkpoint	Indicates that the audit type is **CHECKPOINT** operations.
barrier	Indicates that the audit type is **BARRIER** operations.
cleanconn	Indicates that the audit type is **CLEAN CONNECTION** operations.
seclabel	Indicates that the audit type is security label operations.
notify	Indicates that the audit type is notification operations.
load	Indicates that the audit type is loading operations.

----End

Disabling Audit Log Dump/Kernel Audit Log Dump

After the audit log dump or kernel audit log dump is enabled, you can disable it if you no longer need to dump audit logs or kernel audit logs to OBS.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Dedicated Clusters** > **Clusters** in the navigation pane.

Step 3 Click the name of the cluster for which you want to disable **Audit Log Dump** or **Kernel Audit Log Dump**. In the navigation pane, choose **Security Settings**.

Step 4 In the audit configuration area, toggle the audit log dump/kernel audit log dump function off.

Step 5 Click **Apply**.

If **Configuration Status** is **Applying**, the system is saving the settings.

When the status changes to **Synchronized**, the configurations are saved and take effect.

----End

9.12.3 Viewing GaussDB(DWS) Database Audit Logs

Database audit logs can be set on the **Security Settings** page of the cluster. Security configurations can be modified only for clusters in the **Available** or **Unbalanced** state. Furthermore, the target cluster should not be undergoing any node additions, specification changes, configurations, upgrades, redistribution operations, or restarts.

Prerequisites

- The audit function has been enabled by setting **audit_enabled**. The default value of **audit_enabled** is **ON**. To disable audit, set it to **OFF** by referring to **Modifying GUC Parameters of the GaussDB(DWS) Cluster**.
- The audit items have been configured. For details about how to enable audit items, see **Configuring the Database Audit Logs**.
- The database is running properly and a series of addition, modification, deletion, and query operations have been executed in the database. Otherwise, no audit result is generated.
- The audit logs of each database node are recorded separately.
- Only users with the **AUDITADMIN** permission can view audit records.

Configuring the Database Audit Logs

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

Step 3 In the cluster list, click the name of a cluster. Choose **Security**.

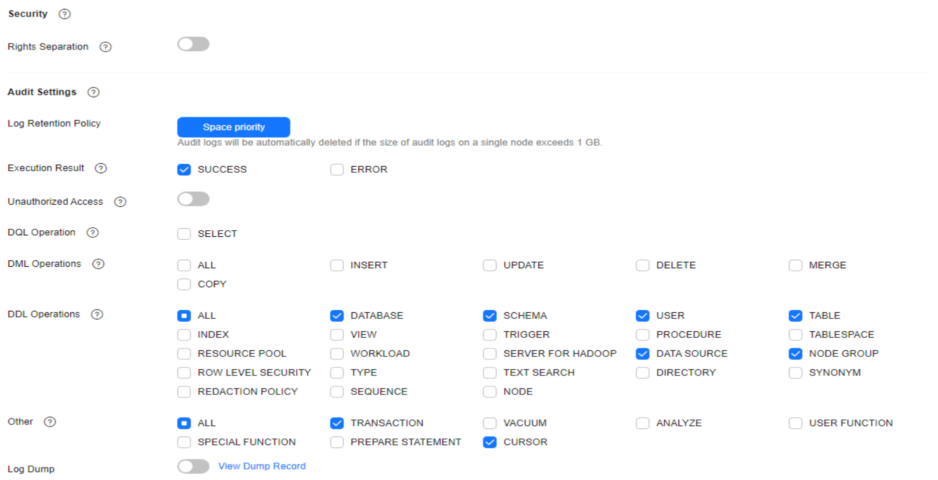
By default, **Configuration Status** is **Synchronized**, which indicates that the latest database result is displayed.

Step 4 In the **Audit Settings** area, set the audit items:

NOTE

The default audit log retention policy is space-first, which means audit logs will be automatically deleted when the size of audit logs on a single node exceeds 1 GB. This function prevents node faults or low performance caused by high disk space occupied by audit logs.

Figure 9-46 Audit items



Security

Rights Separation: off

Audit Settings

Log Retention Policy: Space priority
Audit logs will be automatically deleted if the size of audit logs on a single node exceeds 1 GB.

Execution Result: ☑ SUCCESS ☐ ERROR

Unauthorized Access: off

DQL Operation: ☐ SELECT

DML Operations: ☐ ALL ☐ INSERT ☐ UPDATE ☐ DELETE ☐ MERGE ☐ COPY

DDL Operations: ■ ALL ☑ DATABASE ☑ SCHEMA ☑ USER ☑ TABLE ☐ INDEX ☐ VIEW ☐ TRIGGER ☐ PROCEDURE ☐ TABLESPACE ☐ RESOURCE POOL ☐ WORKLOAD ☐ SERVER FOR HADOOP ☑ DATA SOURCE ☑ NODE GROUP ☐ ROW LEVEL SECURITY ☐ TYPE ☐ TEXT SEARCH ☐ DIRECTORY ☐ SYNONYM ☐ REDACTION POLICY ☐ SEQUENCE ☐ NODE

Other: ■ ALL ☑ TRANSACTION ☐ VACUUM ☐ ANALYZE ☐ USER FUNCTION ☐ SPECIAL FUNCTION ☐ PREPARE STATEMENT ☑ CURSOR

Log Dump: off

Table 9-34 describes the detailed information about the audit items.

Table 9-34 Audit items

Audit Item	Description
Unauthorized access	Whether to record unauthorized operations. This parameter is disabled by default.
DQL operations	**SELECT** operations can be selected. **NOTE** Clusters of 8.1.1.100 and later versions support the **DQL operations** audit item.
DML operations	Whether to record **INSERT**, **UPDATE**, and **DELETE** operations on tables. This parameter is disabled by default. **NOTE** 8.1.1.100 and later versions support fine-grained splitting of audit items, and the **COPY** and **MERGE** options are added.
DDL operations	Whether to record the **CREATE**, **DROP**, and **ALTER** operations of specified database objects. **DATABASE**, **SCHEMA**, and **USER** are selected by default. **NOTE** 8.1.1.100 and later versions support **TABLE**, **DATA SOURCE**, and **NODE GROUP** operations. These operations are enabled by default.

Audit Item	Description
Other operations	Whether to record other operations. Only the **TRANSACTION** and **CURSOR** operations are selected by default. **NOTE** ● 8.1.1.100 and later versions support the **Other operations** audit item. ● You are advised to select **TRANSACTION**. Otherwise, statements in a transaction will not be audited. ● You are advised to select **CURSOR**. Otherwise, **SELECT** statements in a cursor will not be audited. The Data Studio client automatically encapsulates **SELECT** statements using **CURSOR**.

Except the audit items listed in **Table 9-34**, key audit items in **Table 9-35** are enabled by default on GaussDB(DWS).

Table 9-35 Key audit items

Parameter	Description
Key audit items	Records successful and failed logins and logout.
	Records database startup, stop, recovery, and switchover.
	Records user locking and unlocking.
	Records the grants and reclaims of user permissions.
	Records the audit function of the **SET** operation.

Step 5 Enable or disable audit log dumps.

For more information, see **Enabling Audit Log Dumps**.

Step 6 Click **Apply**.

If **Configuration Status** is **Applying**, the system is saving the settings.

When the status changes to **Synchronized**, the configurations are saved and take effect.

You can click ↻ to refresh the configuration information.

----End

Viewing Database Audit Logs

Method 1: Audit logs will occupy disk space. To prevent excessive disk usage, GaussDB(DWS) supports audit log dumping. You can enable the **Log Dump** function to dump audit logs to OBS (you need to create an OBS bucket for storing audit logs first). For details about how to view the dumped logs, see **Enabling Audit Log Dumps**.

Method 2: Use the **Log** function of LTS to view or download the collected database audit logs. For details, see **Checking Cluster Logs**.

Method 3: Database audit logs are stored in the database by default. After connecting to the cluster, you can use the **pg_query_audit** function to view the logs. For details, see **Using Functions to View Database Audit Logs**.

Using Functions to View Database Audit Logs

Step 1 Use the SQL client tool to connect to the database cluster. For details, see **Connecting to a GaussDB(DWS) Cluster**.

Step 2 Use the **pg_query_audit** function to query the audit logs of the current CN. The syntax is as follows:

```
pg_query_audit(timestamptz starttime,timestamptz endtime,audit_log)
```

starttime and **endtime** indicate the start time and end time of the audit record, respectively. **audit_log** indicates the physical file path of the queried audit logs. If **audit_log** is not specified, the audit log information of the current instance is queried.

For example, view the audit records of the current CN node in a specified period.

```
SELECT * FROM pg_query_audit('2021-02-23 21:49:00','2021-02-23 21:50:00');
```

The query result is as follows:

```
        begintime          |          endtime          | operation_type | audit_type | result |  username  | database |
client_conninfo | object_name | command_text |                            detail_info                             |
transaction_xid | query_id |  node_name   |               session_id                | local_port | remote_port
---------------------------+---------------------------+----------------+------------+--------+------------+----------
+-----------------+-------------+-----------------+---------------------------------------------------------------
+-----------------+----------+--------------+-----------------------------+------------+-------------
 2021-02-23 21:49:57.76+08 | 2021-02-23 21:49:57.82+08 | login_logout   | user_login | ok     | dbadmin |
gaussdb | gsql@[local]    | gaussdb     | login db     | login db(gaussdb) successfully, the current user is:
dbadmin | 0               | 0        | coordinator1 | 140324035360512.667403397820909.coordinator1 | 27777
|
```

This record indicates that user **dbadmin** logged in to the **gaussdb** database at 2021-02-23 21:49:57.82 (GMT+08:00). After the host specified by **log_hostname** is started and a client is connected to its IP address, the host name found by reverse DNS resolution is displayed following the at sign (@) in the value of **client_conninfo**.

Step 3 Use the **pgxc_query_audit** function to query audit logs of all CNs. The syntax is as follows:

```
pgxc_query_audit(timestamptz starttime,timestamptz endtime)
```

For example, view the audit records of all CN nodes in a specified period.

```
SELECT * FROM pgxc_query_audit('2021-02-23 22:05:00','2021-02-23 22:07:00') where audit_type =
'user_login' and username = 'user1';
```

The query result is as follows:

```
        begintime           |          endtime           | operation_type | audit_type | result | username | database |
client_conninfo | object_name | command_text |                            detail_info                          | transaction_xid
| query_id |  node_name   |               session_id                      | local_port | remote_port
----------------------------+----------------------------+----------------+------------+--------+----------+----------
+-----------------+-------------+---------------+------------------------------------------------------------
+-----------------+----------+--------------+---------------------------------------------+------------+-------------
```

```
 2021-02-23 22:06:22.219+08 | 2021-02-23 22:06:22.271+08 | login_lgout    | user_login | ok     | user1    |
gaussdb  | gsql@[local]    | gaussdb     | login db     | login db(gaussdb) successfully, the current user is:
user1 | 0               | 0        | coordinator2 | 140689577342720.667404382271356.coordinator  | 27782      |
 2021-02-23 22:05:51.697+08 | 2021-02-23 22:05:51.749+08 | login_lgout    | user_login | ok     | user1    |
gaussdb  | gsql@[local]    | gaussdb     | login db     | login db(gaussdb) successfully, the current user is:
user1 | 0               | 0        | coordinator1 | 140525048424192.667404351749143.coordinator1 | 27777      |
```

The query result shows the successful login records of **user1** in to CN1 and CN2.

Step 4 Query the audit records of multiple objects.

```
SET audit_object_name_format TO 'all';
SELECT object_name,result,operation_type,command_text FROM pgxc_query_audit('2022-08-26
8:00:00','2022-08-26 22:55:00') where command_text like '%student%';
```

The query result is as follows:

```
                           object_name                            | result | operation_type
|                                                    command_text

-------------------------------------------------------------------+--------+----------------
+-------------------------------------------------------------------------------------------------------------------
--------------------------------------------
 student                                                          | ok     | ddl            | CREATE TABLE student(stuNo int,
stuName TEXT);
 studentscore                                                     | ok     | ddl            | CREATE TABLE studentscore(stuNo
int, stuscore int);
 ["public.student_view01","public.studentscore","public.student"] | ok     | ddl            | CREATE OR REPLACE
VIEW student_view01 AS SELECT * FROM student t1 where t1.stuNo in (select stuNo from studentscore t2
where t1.stuNo = t2.stuNo);
 ["public.student_view01","public.student","public.studentscore"] | ok     | dml            | SELECT * FROM
student_view01;
```

In the **object_name** column, the table, view, and base table associated with the view are displayed.

----End

9.12.4 Viewing Operation Logs on the GaussDB(DWS) Console

Enabling CTS

A tracker will be automatically created after CTS is enabled. All traces recorded by CTS are associated with a tracker. Currently, only one tracker can be created for each account.

Step 1 Log in to the management console, choose **Service List** > **Management & Governance** > **Cloud Trace Service**. The CTS management console is displayed.

Step 2 In the navigation tree on the left, click **Trackers**.

Step 3 Enable CTS.

If you are a first-time CTS user and do not have any created tackers in the tracker list, enable CTS first. For details, see **Enabling CTS** in the ***Cloud Trace Service Getting Started***.

If you have enabled CTS, the system has automatically created a management tracker. Only one management tracker can be created and it cannot be deleted. You can also manually create a data tracker. For details, see "Tracker Management" > "Creating a Tracker" in the ***Cloud Trace Service User Guide***.

----End

Disabling the Audit Log Function

If you want to disable the audit log function, disable the tracker in CTS.

Step 1 Log in to the management console, choose **Service List** > **Management & Governance** > **Cloud Trace Service**. The CTS management console is displayed.

Step 2 Disable the audit log function by disabling the tracker. To enable the audit log function again, you only need to enable the tracker.

For details about how to enable or disable a tracker, see **Disabling or Enabling a Tracker** in the *Cloud Trace Service User Guide*.

----End

Key Operations

With CTS, you can record operations associated with GaussDB(DWS) for future query, audit, and backtracking.

NOTE

- The creation and deletion of automatic snapshots are not performed by users, therefore not recorded in audit logs.
- There are many GaussDB(DWS) cluster operation events, but the table below only includes some frequently audited operations.

Table 9-36 GaussDB(DWS) operations that can be recorded by CTS

Operation	Resource	Event Name
Creating a cluster	cluster	createCluster
Deleting a cluster	cluster	deleteCluster
Performance a cluster inspection	cluster	createInspection
Stopping an inspection	cluster	AbortInspection
Scaling out a cluster	cluster	growCluster
Increasing the capacity of idle nodes	cluster	resizeWithFreeNodes
Performing cluster redistribution	cluster	redistributeCluster
Querying redistribution details	cluster	queryRedisInfo
Adding disk capacity	cluster	executeDiskExpand

Operation	Resource	Event Name
Changing cluster specifications	cluster	flavorResize
Restarting a cluster	cluster	rebootCluster
Performing a cluster switchover	cluster	activeStandySwitchover
Resetting a password	cluster	resetPassword
Restoring a cluster	cluster	repairCluster
Creating a cluster connection	cluster	createClusterConnection
Modifying a cluster connection	cluster	modifyClusterConnection
Deleting a cluster connection	cluster	deleteClusterConnection
Resizing a cluster	cluster	resizeCluster
Binding or unbinding an EIP	cluster	bindOrUnbindEIP
Creating or binding an ELB	cluster	createOrBindElb
Unbinding an ELB	cluster	unbindElb
Adding a CN	cluster	addCN
Deleting a CN	cluster	deleteCN
Upgrading a cluster	cluster	clusterUpdateMgr
Scaling in a cluster	cluster	shrinkCluster
Creating a resource management plan	cluster	addWorkloadPlan
Deleting a resource pool	cluster	deleteWorkloadQueueInfo
Creating a resource pool	cluster	addWorkloadQueueInfo
Modifying cluster GUC parameters	cluster	updateClusterConfigurations
Removing the read-only status	cluster	cancelReadonly

Operation	Resource	Event Name
Modifying a maintenance window	cluster	modifyMaintenanceWindow
Adding CN nodes in batches	cluster	batchCreateCn
Deleting CN nodes in batches	cluster	batchDeleteCn
Adding tags in batches	cluster	batchCreateResourceTag
Deleting tags in batches	cluster	batchDeleteResourceTag
Creating a logical cluster	cluster	createLogicalCluster
Deleting logical clusters	cluster	deleteLogicalCluster
Editing a logical cluster	cluster	editLogicalCluster
Restarting logical clusters	cluster	restartLogicalCluster
Converting to a logical cluster	cluster	switchLogicalCluster
Starting a cluster	cluster	startCluster
Stopping a cluster	cluster	stopCluster
Modifying the security group of a cluster	cluster	changeSecurityGroup
Changing the cluster time zone	cluster	modifyClusterTimezone
Creating a snapshot	backup	createBackup
Deleting a snapshot	backup	deleteBackup
Restoring a cluster	backup	restoreCluster
Copying snapshots	backup	copySnapshot
Deleting a snapshot policy	backup	deleteBackupPolicy

Operation	Resource	Event Name
Updating a snapshot policy	backup	updateClustersBackupPolicy
Creating a DR task	disasterRecovery	createDisasterRecovery
Deleting a DR task	disasterRecovery	deleteDisasterRecovery
Starting a DR task	disasterRecovery	startDisasterRecoveryAction
Stopping a DR task	disasterRecovery	stopDisasterRecoveryAction
Switching to the DR cluster	disasterRecovery	switchoverDisasterRecoveryAction
Performing an exception switchover	disasterRecovery	failoverDisasterRecoveryAction
Performing DR	disasterRecovery	recoveryDisaster
Updating DR configurations	disasterRecovery	updateRecoveryDisaster
Querying DR details	disasterRecovery	disasterRecoveryOperate
Setting security parameters	configurations	updateConfigurations
Creating an extended data source	dataSource	createExtDataSource
Deleting an extended data source	dataSource	deleteExtDataSource
Updating an extended data source	dataSource	updateExtDataSource
Creating an MRS data source	dataSource	createExtDataSource
Deleting an MRS data source	dataSource	deleteExtDataSource
Updating an MRS data source	dataSource	updateExtDataSource

Viewing Traces

Step 1 Log in to the management console, choose **Service List** > **Management & Governance** > **Cloud Trace Service**. The CTS management console is displayed.

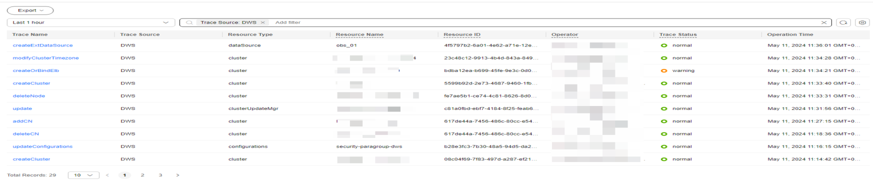
Step 2 In the navigation pane on the left, choose **Trace List**.

Step 3 Click the search box above the trace list and set the search criteria.

The following filters are available:

- **Trace Name**: If you select this option, you also need to select a specific trace name.
- **Cloud Service**: Select **GaussDB(DWS)**.
- **Resource Type**: Select **All resource types** or specify a resource type.
- **Resource Name**: If you select this option, you also need to select or enter a specific resource name.
- **Resource ID**: If you select this option, you also need to select or enter a specific resource ID.
- **Operator**: Select a specific operator (at user level rather than tenant level).
- **Event ID**: If you select this option, you also need to select or enter an event ID.
- **Trace Status**: Available options include **All trace statuses**, **normal**, **warning**, and **incident**. You can only select one of them.
- **Enterprise Project ID**: If you select this option, you also need to select or enter a specific enterprise project ID.
- **Access Key ID**: If you select this option, you also need to select or enter a specific access key ID.

Figure 9-47 Querying traces



Trace Name	Trace Source	Resource Type	Resource Name	Resource ID	Operator	Trace Status	Operation Time
createExtDataSource	DWS	dataSource	obs_01	4f5797b2-6a01-4e62-a71e-12e...	[redacted]	normal	May 11, 2024 11:36:01 GMT+0...
modifyClusterTimezone	DWS	cluster	[redacted]	23c48c12-9913-4b4d-843a-849...	[redacted]	normal	May 11, 2024 11:34:28 GMT+0...
createOrBindElb	DWS	cluster	[redacted]	bdba12ea-b699-45fe-9e3c-0d0...	[redacted]	warning	May 11, 2024 11:34:21 GMT+0...
createCluster	DWS	cluster	[redacted]	5599b92d-2e73-4687-9460-1fb...	[redacted]	normal	May 11, 2024 11:33:40 GMT+0...
deleteNode	DWS	cluster	[redacted]	fe7ae5b1-ce74-4c81-8626-8d0...	[redacted]	normal	May 11, 2024 11:33:31 GMT+0...
update	DWS	clusterUpdateMgr	[redacted]	c81a0fbd-ebf7-4184-8f25-feab6...	[redacted]	normal	May 11, 2024 11:31:56 GMT+0...
addCN	DWS	cluster	[redacted]	617de44a-7456-486c-80cc-e54...	[redacted]	normal	May 11, 2024 11:27:15 GMT+0...
deleteCN	DWS	cluster	[redacted]	617de44a-7456-486c-80cc-e54...	[redacted]	normal	May 11, 2024 11:18:36 GMT+0...
updateConfigurations	DWS	configurations	security-paragroup-dws	b28e3fc3-7b30-48a5-94d5-da2...	[redacted]	normal	May 11, 2024 11:16:15 GMT+0...
createCluster	DWS	cluster	[redacted]	08c04f69-7f83-497d-a287-ef21...	[redacted]	normal	May 11, 2024 11:14:42 GMT+0...

Total Records: 29 10 < 1 2 3 >

Step 4 Click **Query**.

Step 5 Click the name of the trace to be viewed. A window is displayed, showing the trace details.

For details about key fields of a CTS trace, see "Trace References" > "Trace Structure" and "Trace References" > "Example Traces" in ***Cloud Trace Service User Guide***.

----End

9.12.5 Viewing Other Logs of the GaussDB(DWS) Cluster

Overview

Cluster logs are collected and sent to Log Tank Service (LTS). You can check or dump the collected cluster logs on LTS.

The following log types are supported: CN logs, DN logs, OS messages logs, audit logs, cms logs, gtm logs, Roach client logs, Roach server logs, upgrade logs, and scale-out logs.

NOTE

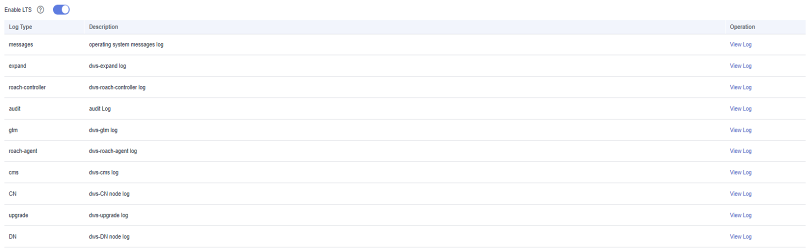
- Only 8.1.1.300 and later versions support cluster log management.
- Only 8.3.0 and later versions support CMS logs, GTM logs, Roach client logs, Roach server logs, scaling logs, and upgrade logs.

Enabling LTS

Step 1 Log in to the GaussDB(DWS) console.

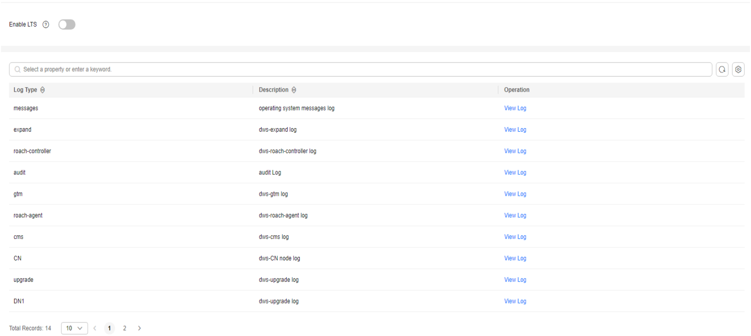
Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 Click the name of the target cluster. Choose **Logs**.



Enable LTS ⓘ ●

Log Type	Description	Operation
messages	operating system messages log	View Log
expand	dws-expand log	View Log
roach-controller	dws-roach-controller log	View Log
audit	audit Log	View Log
gtm	dws-gtm log	View Log
roach-agent	dws-roach-agent log	View Log
cms	dws-cms log	View Log
CN	dws-CN node log	View Log
upgrade	dws-upgrade log	View Log
DN	dws-DN node log	View Log



Enable LTS ⓘ ○

Select a property or enter a keyword.

Log Type	Description	Operation
messages	operating system messages log	View Log
expand	dws-expand log	View Log
roach-controller	dws-roach-controller log	View Log
audit	audit Log	View Log
gtm	dws-gtm log	View Log
roach-agent	dws-roach-agent log	View Log
cms	dws-cms log	View Log
CN	dws-CN node log	View Log
upgrade	dws-upgrade log	View Log
DN1	dws-upgrade log	View Log

Total Records: 14   10   <   1   2   >

Step 4 On the **Logs** tab, enable LTS. If LTS is enabled for the first time, the following dialog box will be displayed. Confirm the information and click **Yes**.

NOTE

- If LTS has been enabled and authorized to create an agency, no authorization is required when LTS is enabled again.
- By default, only Huawei Cloud accounts or users with **Security Administrator** permissions can query and create agencies. IAM users under an account do not have the permission to query or create agencies by default. Contact a user with that permission and complete the authorization on the current page.
- When interconnecting with LTS, you need to grant LTS-related permission policies **(LTS Admin**, **LTS Administrator**, **LTS FullAccess**, and **LTS ReadOnlyAccess**) to users.

----End

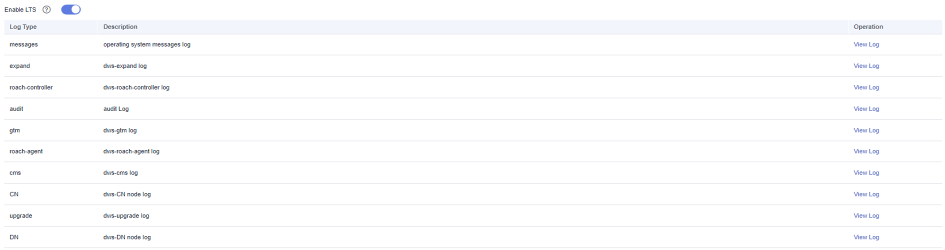
Checking Cluster Logs

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 Click the name of the target cluster. Choose **Logs**.

Step 4 On the **Logs** tab, click **View Log** in the **Operation** column of a log type to go to the Log Tank Service (LTS) page and view logs.



Enable LTS ⓘ ●

Log Type	Description	Operation
messages	operating system messages log	View Log
expand	dws-expand log	View Log
roach-controller	dws-roach-controller log	View Log
audit	audit Log	View Log
gtm	dws-gtm log	View Log
roach-agent	dws-roach-agent log	View Log
cms	dws-cms log	View Log
CN	dws-CN node log	View Log
upgrade	dws-upgrade log	View Log
DN	dws-DN node log	View Log

----End

Disabling LTS

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 Click the name of the target cluster. Choose **Logs**.

Step 4 Toggle off the LTS switch.

Step 5 Click **OK** in the dialog box.

----End

9.13 Handling Abnormal GaussDB(DWS) Clusters

Removing the Read-only Status

A cluster in read-only status does not allow write operations. You can remove this status on the management console. A cluster becomes read-only probably because of high disk usage. For how to solve this problem, see **Solution to High Disk Usage and Cluster Read-Only**.

NOTE

- The read-only status can be canceled for version 1.7.2 or later.
- In 8.2.0 and later versions, you can free up disk space by using **DROP/TRUNCATE TABLE** in a read-only cluster.

Impacts on the System

- You can cancel the read-only status only when a cluster is read-only.
- When a cluster is in read-only status, stop the write tasks to prevent data loss caused by used up disk space.
- After the read-only status is canceled, clear the data as soon as possible to prevent the cluster from entering the read-only status again after a period of time.

Procedure

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Dedicated Clusters** > **Clusters**.

All clusters are displayed by default.

Step 3 In the row containing the cluster whose cluster status is **Read-only**, click **Cancel Read-only**.

Step 4 In the dialog box that is displayed, click **OK** to confirm and remove the read-only status for the cluster.

----End

Performing a Primary/Standby Switchback

In the **Unbalanced** state, the number of primary instances on some nodes increases. As a result, the load pressure is high. In this case, the cluster is normal, but the overall performance is not as good as that in a balanced state. Restore the primary-standby relationship to recover the cluster to the available state.

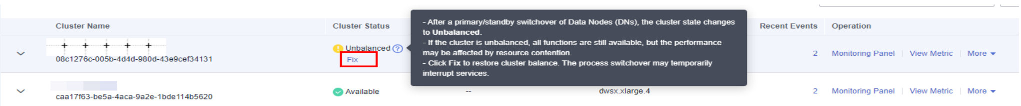
NOTE

- Only 8.1.1.202 and later versions support primary/standby cluster restoration.
- Cluster restoration interrupts services for a short period of time. The interruption duration depends on the service volume. You are advised to perform this operation during off-peak hours.

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Dedicated Clusters** > **Clusters** and locate the cluster whose load is unbalanced.

Step 3 In the **Cluster Status** column of the cluster, click **Fix** under **Unbalanced**.



- After a primary/standby switchover of Data Nodes (DNs), the cluster state changes to **Unbalanced**.
- If the cluster is unbalanced, all functions are still available, but the performance may be affected by resource contention.
- Click **Fix** to restore cluster balance. The process switchover may temporarily interrupt services.

Cluster Name	Cluster Status			Recent Events	Operation
[redacted] 08c1276c-005b-4d4d-980d-43e9cef34131	Unbalanced Fix			2	Monitoring Panel | View Metric | More
[redacted] caa17f63-be5a-4aca-9a2e-1bde114b5620	Available	--	dwsx.xlarge.4	2	Monitoring Panel | View Metric | More

Step 4 In the dialog box that is displayed, confirm that the service is in off-peak hours, and click **Yes**. A message will be displayed in the upper right corner, indicating that the switchback request is being processed.

Step 5 Check the cluster status. During the switchback, the cluster status is **Switching back**. After the switchback, the cluster status will change to **Available**.

----End

9.14 Reclaiming GaussDB(DWS) Space Using Vacuum

9.14.1 Overview

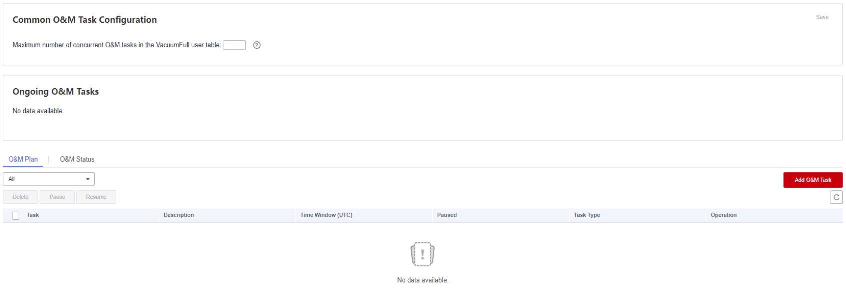
GaussDB(DWS) provides the intelligent O&M feature to help users quickly and efficiently execute O&M tasks. Intelligent O&M selects a proper time window and concurrency to complete specified tasks based on the cluster load. During O&M

tasks, intelligent O&M monitors user service changes and promptly adapts task execution policies to minimize the impact on user services. Periodic tasks and one-off tasks are supported, and you can configure the time window as required.

Intelligent O&M ensures high availability. When the cluster is abnormal, failed O&M tasks will be retried. If some steps of an O&M task cannot be completed due to an abnormal cluster, the failed steps will be skipped for cost saving.

The intelligent O&M page consists of the following parts:

- Setting the common configurations of O&M tasks
  - **Maximum number of concurrent O&M tasks in the VacuumFull user table**: applies to VacuumFull O&M tasks for each user table. You are advised to set this parameter based on the available disk space and I/O load within a specific time window. The value ranges from 1 to 24. The recommended value is **5**.
- Information about ongoing O&M tasks. (Currently, only VACUUM tasks are displayed. If disk space is insufficient because of table bloating, you can vacuum tables. For details, see **Performance Deterioration Due to Table Bloating**).
  - Frequent table creation and deletion can lead to table bloating. To free up space, you can run the **VACUUM** command on system catalogs.
  - Frequently update and delete operations can lead to table bloating. To free up space, you can run the **VACUUM** or **VACUUM FULL** command on system catalogs.
- O&M details: **O&M Plan** and **O&M Status**. **O&M Plan** displays the basic information about all O&M tasks, and **O&M Status** displays the running status.



Common O&M Task Configuration

Maximum number of concurrent O&M tasks in the VacuumFull user table:

Ongoing O&M Tasks

No data available.

☐ Task	Description	Time Window (UTC)	Paused	Task Type	Operation

No data available.

NOTE

- This feature is supported only in 8.1.3 or later.
- A storage-compute coupled data warehouse (standalone) does not support the intelligent O&M function.
- After completing the **VACUUM FULL** O&M task, the system automatically performs the **ANALYZE** operation.
- Only cluster 8.1.3 and later versions support the common configuration module for O&M tasks. For earlier versions, contact technical support to upgrade them.

9.14.2 Managing O&M Plans

Setting the Common Configurations of O&M Tasks

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click the name of the target cluster.

Step 3 In the navigation pane, choose **Intelligent O&M**.

Step 4 In the **Common O&M Task Configuration** area, configure **Maximum number of concurrent O&M tasks in the VacuumFull user table**.

NOTE

- This configuration takes effect for the VACUUM FULL O&M tasks of all user tables.
- The concurrency value range is 1 to 24. Configure it based on the remaining disk space and I/O load. You are advised to set it to 5.

----End

Adding an O&M Plan

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click the name of the target cluster.

Step 3 In the navigation pane, choose **Intelligent O&M**.

Step 4 In the **O&M Plan** area, click **Add O&M Task**.

Step 5 In the displayed **Add O&M Task** dialog box, configure basic information about the O&M task.

Table 9-37 Basic configuration items of an O&M task

Configuration Item	Description	Example
O&M Task	Vacuum (Currently, only Vacuum O&M tasks are supported.)	Vacuum
Description	Brief description of the intelligent O&M task.	This intelligent O&M task helps users periodically invoke Vacuum commands to reclaim space.
Remarks	Supplementary information.	-

Configuration Item	Description	Example
Scheduling Mode	The supports the following scheduling modes: ● **Auto**: Intelligent O&M scans the database in a specified time window, and automatically delivers table-level vacuum tasks by service load and reclaimable space of user tables. ● **Specify**: You need to specify a vacuum target. Intelligent O&M will automatically deliver a table-level vacuum task in a specified time window. ● **Priority**: You can specify the preferential vacuum targets. During the remaining time window (if any), Intelligent O&M will automatically scan other tables that can be vacuumed and deliver table-level vacuum tasks. **NOTE** You are advised to select **Specify** for **VACUUM** and **VACUUM FULL** operations. Do not perform **VACUUM FULL** on wide column-store tables. Otherwise, memory bloat may occur.	Specify

Configuration Item	Description	Example
Autovacuum	Supported: system catalog Vacuum or user table VacuumFull. ● A system catalog VACUUM transaction holds a level-5 lock (share update exclusive lock), which does not affect user services. Only the transactions on the DDL process of the system catalog are blocked. ● A user table VACUUM FULL transaction holds a level-8 lock (access exclusive lock). All the other transactions on the table are blocked until VACUUM FULL is complete. To avoid affecting services, you are advised to perform VACUUM FULL during off-peak hours. **CAUTION** During VACUUM FULL, the space usage will first increase and then decrease, because this operation requires the same space as the table to be vacuumed. (Actual table size = Total table size x (1 – dirty page rate). Ensure you have sufficient space before doing VACUUM FULL.	User table (VACUUM FULL)
Vacuum First	You can configure the preferred Vacuum target. Each row corresponds to a table. Each table is represented by the database name, schema name, and table name, which are separated by spaces.	-
Advanced settings	If you select **Custom**, you can configure the autovacuum triggers, including the table bloat and table reclaimable space. If you select **Default**, the default configuration is used. **NOTE** VACUUM bloat rate: After frequent UPDATE and DELETE operations are performed in a database, the deleted or updated rows are logically deleted from the database, but actually still exist in tables. Before VACUUM is complete, such data is still stored in disks, causing table bloat. If the bloat rate reaches the percentage threshold set in an O&M task, VACUUM will be automatically triggered.	Default configuration (table bloat rate 80% or reclaimable space 100 GB.)

Step 6 Click **Next** > **Schedule** to configure scheduling for O&M tasks.

Select an O&M type.

- **One-off**: Set the start time and end time of the task.
- **Periodic**: Select a time window type, which includes **Daily**, **Weekly**, and **Monthly**, and select a time segment. Intelligent O&M will automatically analyze the time window and deliver O&M tasks accordingly.

⚠ CAUTION

- Do not choose peak hours when configuring the time window for autovacuum O&M tasks. Otherwise, automatic Vacuum may cause a deadlock on user services.
- The number of concurrent O&M tasks (vacuum/vacuum full) ranges from 0 to 24 for user tables, and from 0 to 1 for system catalogs. The concurrency value cannot be customized, but can be automatically adjusted based on system **io_util**.
  - Two intervals for 0% to 60%
    - 0% to 30%: The concurrency value increases by 2 each time the value of **io_util** decreases by 15%.
    - 30% to 60%: The concurrency value is incremented by 1 each time the value of **io_util** decreases by 15%.
  - 60% to 70%: The concurrency value remains unchanged.
  - Above 70%: The concurrency value decreases by 1 until it reaches 0.
- The scheduler scans the expansion of column-store compression units (CUs) within the time window. If the average number of CU records in a column-store table is less than 1000, the scheduler scans the table first. The scanning of column-store CUs is not limited by table bloat or table reclaimable space.
- A maximum of 100 tables can be added to the priority list.
- The scheduler autovacuum function depends on the statistics. If the statistics are inaccurate, the execution sequence and results may be affected.
- The scheduler does not support names containing spaces or single quotation marks, including database names, schema names, and table names. Otherwise, the tables will be skipped. Priority tables whose name contains spaces or single quotation marks will also be skipped automatically.

Step 7 Click **Next: Finish**. After you confirm the information, click **Finish** to submit the request.

----End

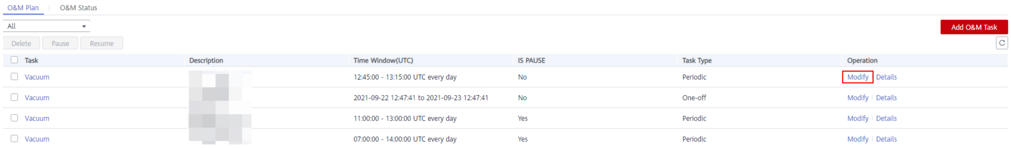
Modifying an O&M Plan

Step 1 Log in to the GaussDB(DWS) console.

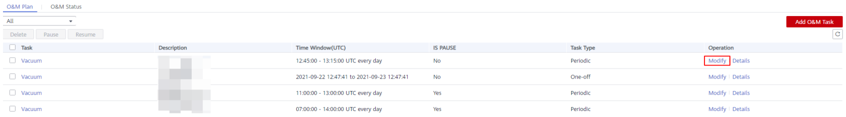
Step 2 Click the name of the target cluster.

Step 3 In the navigation pane, choose **Intelligent O&M**.

Step 4 In the **O&M Plan** area, click **Modify** in the **Operation** column of the target task.



Task	Description	Time Window(UTC)	IS PAUSE	Task Type	Operation
Vacuum	[redacted]	12:45:00 - 13:15:00 UTC every day	No	Periodic	Modify Details
Vacuum	[redacted]	2021-09-22 12:47:41 to 2021-09-23 12:47:41	No	One-off	Modify | Details
Vacuum	[redacted]	11:00:00 - 13:00:00 UTC every day	Yes	Periodic	Modify | Details
Vacuum	[redacted]	07:00:00 - 14:00:00 UTC every day	Yes	Periodic	Modify | Details



Task	Description	Time Window(UTC)	IS PAUSE	Task Type	Operation
Vacuum	[redacted]	12:45:00 - 13:15:00 UTC every day	No	Periodic	Modify Details
Vacuum	[redacted]	2021-09-22 12:47:41 to 2021-09-23 12:47:41	No	One-off	Modify | Details
Vacuum	[redacted]	11:00:00 - 13:00:00 UTC every day	Yes	Periodic	Modify | Details
Vacuum	[redacted]	07:00:00 - 14:00:00 UTC every day	Yes	Periodic	Modify | Details

Step 5 The **Modify O&M Task** panel is displayed. The configurations are similar to adding an O&M task (see **Adding an O&M Plan**).

Step 6 Confirm the modification and click **OK**.

----End

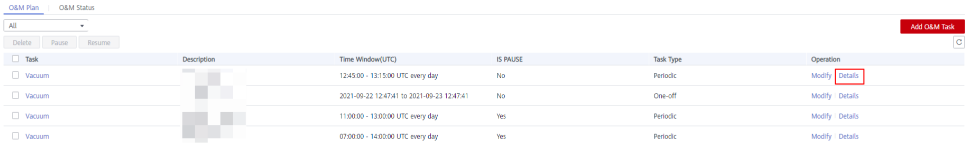
Viewing O&M Task Details

Step 1 Log in to the GaussDB(DWS) console.

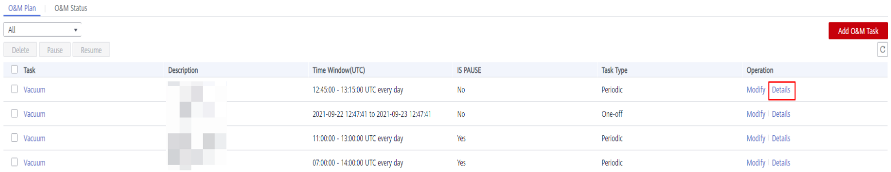
Step 2 Click the name of the target cluster.

Step 3 In the navigation pane, choose **Intelligent O&M**.

Step 4 In the **O&M Plan** area, click **Details** in the **Operation** column of the target task.



Task	Description	Time Window(UTC)	IS PAUSE	Task Type	Operation
Vacuum	[redacted]	12:45:00 - 13:15:00 UTC every day	No	Periodic	Modify Details
Vacuum	[redacted]	2021-09-22 12:47:41 to 2021-09-23 12:47:41	No	One-off	Modify | Details
Vacuum	[redacted]	11:00:00 - 13:00:00 UTC every day	Yes	Periodic	Modify | Details
Vacuum	[redacted]	07:00:00 - 14:00:00 UTC every day	Yes	Periodic	Modify | Details



Task	Description	Time Window(UTC)	IS PAUSE	Task Type	Operation
Vacuum	[redacted]	12:45:00 - 13:15:00 UTC every day	No	Periodic	Modify Details
Vacuum	[redacted]	2021-09-22 12:47:41 to 2021-09-23 12:47:41	No	One-off	Modify | Details
Vacuum	[redacted]	11:00:00 - 13:00:00 UTC every day	Yes	Periodic	Modify | Details
Vacuum	[redacted]	07:00:00 - 14:00:00 UTC every day	Yes	Periodic	Modify | Details

Step 5 The **O&M Task Details** panel is displayed for you to check the information.

----End

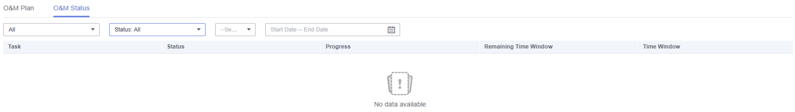
9.14.3 Viewing O&M Tasks

Step 1 Log in to the GaussDB(DWS) console.

Step 2 Click the name of the target cluster.

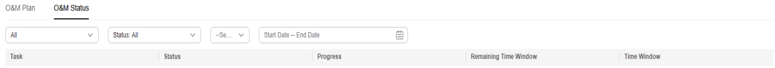
Step 3 In the navigation pane, choose **Intelligent O&M**.

Step 4 Switch to the **O&M Status** area.



Task	Status	Progress	Remaining Time Window	Time Window

No data available



Task	Status	Progress	Remaining Time Window	Time Window

Step 5 Click the name of a specified O&M task to view the status details.

- **O&M Task**: **Vacuum**
- Status: **Waiting**, **Running**, **Completed**, or **Failed**.
- **Progress**
- **Remaining Time Window**
- **Time Window** (Local Time)
- **Tables Being Vacuumed**
- **Tables to Be Vacuumed**
- **Vacuumed Tables**
- **Failed Tables**

NOTE

- A maximum of 100 tables can be displayed for each category of the tables above.
- If the cluster is read-only, the INSERT statement cannot be executed for intelligent O&M tasks. There may be tasks remaining in the **Running** status. The **Running** status in this case is a historical status, and it indicates that the task is not completed within the specified time. If you manually pause the task and the task is not scheduled, the task may remain in the **Waiting** status. In this case, cancel the cluster read-only state and contact technical support to update the task status.

----End

9.15 Authorizing a GaussDB(DWS) Cluster O&M Account

Context

If you need technical support when using a cluster, you can authorize them to use an O&M account on the GaussDB(DWS) console to access the cluster for fault locating.

NOTE

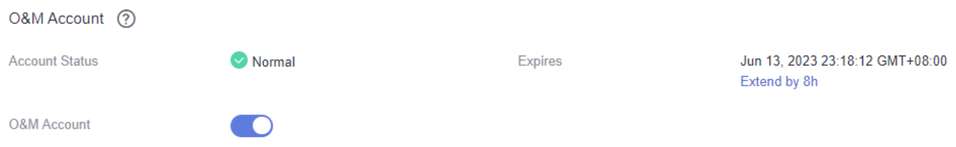
Only cluster 8.1.3.110 and later versions support O&M accounts. For earlier versions, contact technical support.

Overview

You can perform the following operations:

- Enable or disable the O&M account.
- Check the O&M account status.
- View the validity period of an O&M account and extend it as needed. (If the validity period is not extended, the account will expire automatically.)

Figure 9-48 O&M Account



O&M Account

Account Status: Normal

Expires: Jun 13, 2023 23:18:12 GMT+08:00

O&M Account

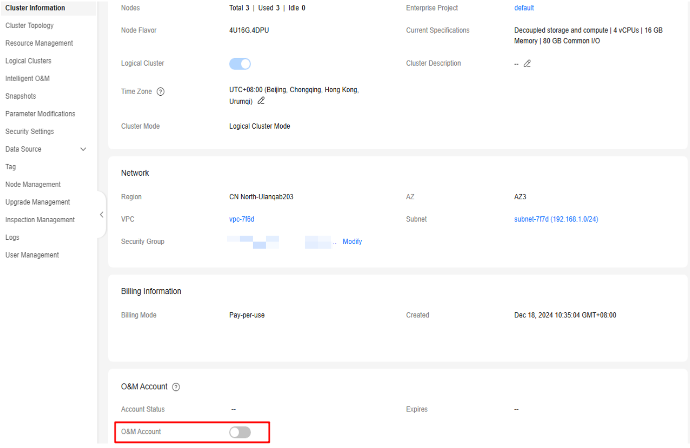
Enabling the O&M Account

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the cluster list, click the name of a cluster.

Step 3 On the cluster details page, and enable **O&M Account** in the **O&M Account** area.

Figure 9-49 Enabling O&M Account



Nodes: Total 3 | Used 3 | Idle 0

Enterprise Project: default

Node Flavor: 4U16G.4DPU

Current Specifications: Decoupled storage and compute | 4 vCPUs | 16 GB Memory | 80 GB Common I/O

Logical Cluster

Cluster Description: --

Time Zone: UTC+08:00 (Beijing, Chongqing, Hong Kong, Urumqi)

Cluster Mode: Logical Cluster Mode

Network

Region: CN North-Ulanqab203

AZ: AZ3

VPC: vpc-7f6d

Subnet: subnet-7f7d (192.168.1.0/24)

Security Group: [redacted] .. Modify

Billing Information

Billing Mode: Pay-per-use

Created: Dec 18, 2024 10:35:04 GMT+08:00

O&M Account

Account Status: --

Expires: --

O&M Account

Step 4 In the displayed dialog box, click **OK**.

Step 5 Check the created O&M account. Its name format is **om_user**_*First_eight_numbers_in_cluster_ID*.

Assign the **gs_role_analyze_any**, **gs_role_vacuum_any**, **gs_role_read_all_stats**, and **gs_role_signal_backend** roles to the account. For details, see **Preset Roles**.

NOTE

You can toggle off the switch and delete the O&M account if it is no longer needed.

----End

Extending the Validity Period

Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the cluster list, click the name of a cluster.

Step 3 On the cluster details page, click **Extend by 8h** in the **O&M Account** area.

Step 4 In the displayed dialog box, click **OK**.

- For a normal account, its validity period is extended to 8 hours later than its expiration time.
- For an expired account, its validity period is extended to 8 hours later than the current time.

----End