Data Warehouse Service

Management Guide

Issue	01
Date	2024-03-13

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Contents

1 Process for 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.

Process Description

Figure 1-1 Process for using GaussDB(DWS)

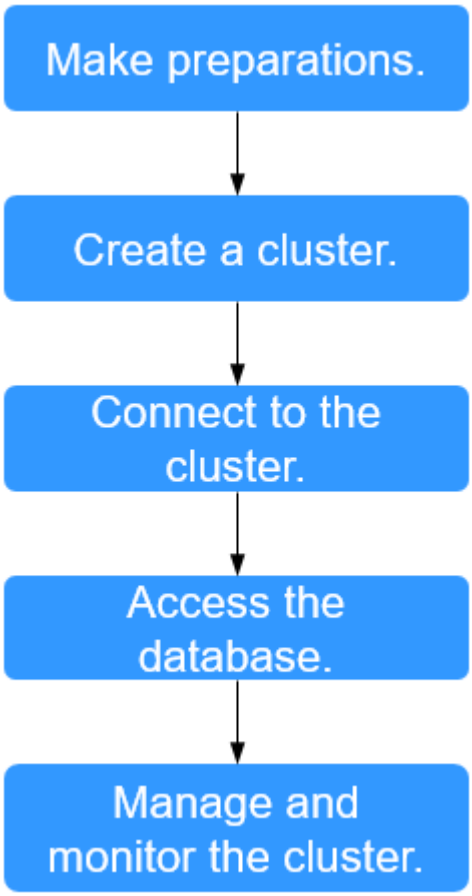


Table 1-1 Process description

Process	Task	Description	Operation Instruction
Make preparations.	-	Before using GaussDB(DWS), select an open port on your firewall as the database port of your data warehouse cluster.	**Preparations**

Process	Task	Description	Operation Instruction
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) 2.0 Cluster** • **Creating a GaussDB(DWS) 3.0 Cluster**
Connect to the cluster.	-	After the data warehouse cluster is successfully 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) management console.	**Methods of Connecting to a 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.	Manage the cluster.	View the cluster status, modify cluster configurations, add cluster tags, and scale out, restart, and delete the cluster.	**Managing clusters**
	Manage the snapshot.	Create snapshots to back up and restore the cluster.	**Snapshots**

Process	Task	Description	Operation Instruction
	Perform O&M and monitoring.	View the running status and performance of the cluster through monitoring, log auditing, event notification, and resource load management.	• **Monitoring Clusters Using Cloud Eye** • **Event Notifications Overview** • **Audit Logs** • **Resource Management**

2 Preparations

Before using Huawei Cloud GaussDB(DWS), make the following preparations:

- **Registering a Public Cloud Account**
- **Determining the Cluster Ports**

Registering a Public Cloud Account

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

1. Open the official public cloud website (**https://www.huaweicloud.com/eu/**) 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.

Determining the Cluster Ports

- When creating a GaussDB(DWS) cluster, you need to specify a port for SQL clients or applications to access the cluster.
- If your client is behind a firewall, you need an available port so that you can connect to the cluster and perform query and analysis from the SQL client tool.
- If you do not know an available port, contact the network administrator to specify an open port on your firewall. The ports supported by GaussDB(DWS) range from 8000 to 30000.
- After a cluster is created, its port number cannot be changed. Ensure that the port specified is available.

3 Creating or Deleting a Cluster

3.1 Accessing the GaussDB(DWS) Management Console

Scenario

This section describes how to log in to the GaussDB(DWS) management console and use GaussDB(DWS).

Procedure

Step 1 Log in to the Huawei Cloud management console.

Step 2 Click  in the upper left corner of the console homepage to expand the **Service List**, and choose **Analytics** > **GaussDB(DWS)**.

Step 3 Choose **Analytics** > **GaussDB(DWS)** to enter the GaussDB(DWS) management console.

----End

3.2 Creating a Dedicated Cluster

3.2.1 Creating a GaussDB(DWS) 2.0 Cluster

To use Huawei Cloud GaussDB(DWS), create a data warehouse cluster first.

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

NOTE

- To balance loads, achieve high availability, and avoid single-node faults, if no ELB is bound during cluster creation, you can bind an ELB on the cluster details page after the cluster is created. For details, see **Associating and Disassociating ELB**.
- 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**.

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**.
- Ensure that the number of available nodes meets the following conditions. Otherwise, the cluster cannot be created.

  The number of nodes that can be used by a user depends on the product type you select.A hybrid data warehouse cluster (standalone mode) 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 **Cluster > Dedicated Cluster** page.

Creating a Cluster

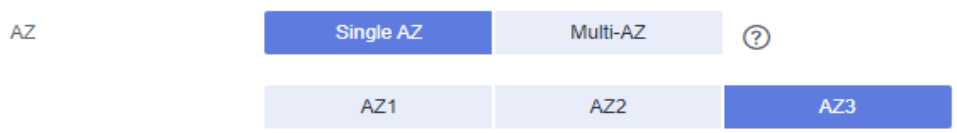
Step 1 Go to the **page for creating a data warehouse cluster**.

Step 2 Select **Region**.

Table 3-1 Region parameters

Parameter	Description	Example Value
Region	Select the actual region where the cluster nodes run.	-
AZ	Select an AZ associated with the cluster region. For more information, see **Regions and AZs**.	-

Step 3 Select an AZ. You can select **Single AZ** or **Multi-AZ** as required.



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 multi-AZ cluster supports only GaussDB(DWS) 2.0 standard data warehouses.
- 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 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.
- After a cluster is created, its type cannot be changed. For details about the differences between product types, see **Data Warehouse Types**.

Table 3-2 Node configuration parameters

Parameter	Description	Example Value
Resource	Product type. It can be: • **Standard data warehouse**: It can analyze hot and cold data and is highly cost-effective. Its storage and computing resources are not limited, and can be elastically scaled and billed per use. It is suitable for the converged analysis that requires integrated databases, warehouses, marts, and lakes. It is most suitable for OLAP workloads. • **Stream data warehouse**: It provides efficient time series computing and IoT analysis capabilities based on the standard data warehouse and supports correlation between real-time and historical data. The compression ratio can reach 40:1. It can be used for IoT real-time analysis. • **Hybrid data warehouse**: It provides high-concurrency, high-performance, and low-latency transaction processing capabilities at low costs based on large-scale data query and analysis capabilities. The data warehouse can be used to process HTAP hybrid loads, and can be deployed in standalone or cluster mode.	Standard

Parameter	Description	Example Value
	NOTE • A hybrid data warehouse can be deployed in cluster or standalone mode. – Cluster deployment: If the name of the selected node flavor contains **h** (for example, **dwsx2.h.4xlarge.4.c6**), the hybrid data warehouse can be deployed in cluster mode. You can deploy multiple nodes, scale nodes, and manage resource pools. – Standalone deployment: If the name of the selected node flavor contains **h1** (for example, **dwsx2.h1.xlarge.2.c6**), the hybrid data warehouse only supports standalone deployment, which does not provide HA capabilities. The storage cost can be reduced by half. A standalone data warehouse 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.	
Compute Resource	It can be: • ESC: Scalable, reliable, and high-throughput virtual block storage is provided in a distributed architecture. This ensures that data can be quickly migrated and restored if any data replica is unavailable, preventing data from being lost because of a single hardware fault. Backup and restoration can be performed on ECSs and EVS disks. You can configure automatic backup policies for them.	-
Storage Type	It can be: • **Cloud SSD** • Local SSD **NOTE** Local SSD disks do not support disk scale-out. For more information, see **Disk Types and Performance**.	-

Parameter	Description	Example Value
CPU Architecture	The CPU architecture includes: • **x86** • **Kunpeng** **NOTE** The only difference between the x86 and Kunpeng architectures lies in the underlying architecture, of which the application layer is unaware. The same SQL syntax is used. If x86 servers are sold out when you create a cluster, select the Kunpeng architecture.	-
Node Flavor	Select the desired node flavor based on service requirements. Each node flavor displays 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**.	dws2.m6.4xlarge.8
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.	-
Cold storage	You are advised to store cold data in OBS, which is 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

Parameter	Description	Example Value
Total	Displays the total capacity of a cluster. 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 5 Click **Next: Configure Network**.

Step 6 Configure the network.

Figure 3-1 Network parameters

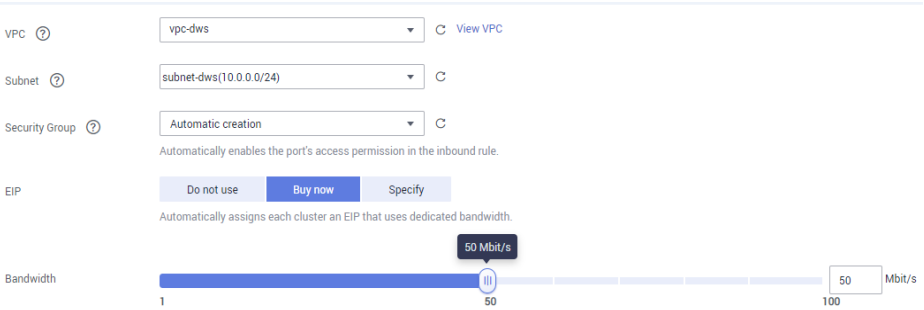


Table 3-3 Network parameters

Parameter	Description	Example Value
VPC	Specify a virtual private network for nodes in a cluster to isolate networks of different services. 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 that satisfies your needs. 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.	vpc-dws

Parameter	Description	Example Value
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 the  button 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.. – **Protocol**: **TCP** – **Port**: **8000**. Use the database port number when you create the cluster for receiving GaussDB(DWS) client connections. – **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
	The security group of a cluster cannot be changed but can be modified. For details, see **Modifying a Security Group**.	
Public Network Access	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**: The EIP is not required. If GaussDB(DWS) is used in the production environment, bind GaussDB(DWS) to ELB first, and then bind GaussDB(DWS) to an EIP on the ELB page. • **Buy Now**: Users specify the bandwidth of the EIP and the system automatically assigns an EIP that exclusively uses bandwidth to each cluster so that users can use the EIP 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 **Create EIP** to go to the **Elastic IP** page and create an EIP that satisfies your needs. You can set the bandwidth as needed. **NOTE** • 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 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

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, bind GaussDB(DWS) to ELB first, and then bind GaussDB(DWS) to an EIP on the ELB page. • **Specify**: Specify an ELB to be bound to the cluster. If no available load balancers are displayed in the drop-down list, click **Create ELB** to go to the Elastic Load Balance page and create a load balancer as needed.	Specify
Bandwidth	When **EIP** is set to **Buy now**, you need to specify the bandwidth of the EIP, which ranges from 1 Mbit/s to 100 Mbit/s.	50Mbit/s

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

Step 8 Configure cluster parameters.

Table 3-4 Cluster parameters

Parameter	Description	Example Value
Cluster Name	Set the name of the data warehouse cluster. The cluster name contains 4 to 64 case-insensitive characters and must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed. **NOTE** If the cluster name cannot be changed on the console, contact technical support.	dws-demo
Cluster Version	Displays the version of the database instance installed in the cluster. The figure 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	Set the database administrator name. The administrator username 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	Set the password of the database administrator account. The password complexity requirements are as follows: • Consists of 12 to 32 characters. • Cannot be the username or the username spelled backwards. • Must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-) • Passes the weak password check. **NOTE** Change the password regularly and keep it secure.	-
Confirm Password	Enter the database administrator password again.	-
Database Port	Specify 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 6** 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.	-

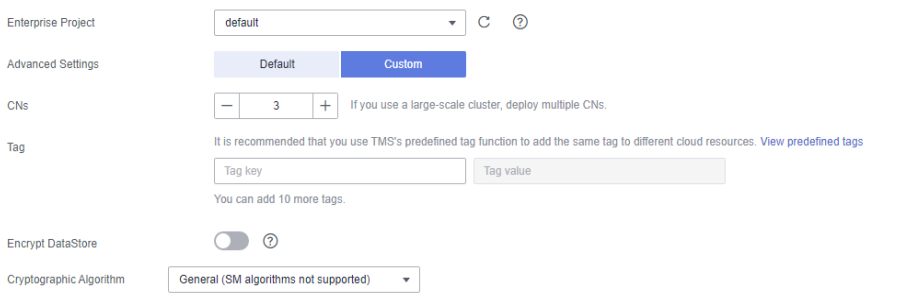
Step 9 Configure the enterprise project to which the cluster belongs. You can configure this parameter only when the Enterprise Project Management service is enabled. The default value is **default**.

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

Figure 3-2 Custom advanced parameters



- **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 3-5**. By default, no tag is added to the cluster.

  For more information about tags, see **Overview**.

  Table 3-5 Tag parameters

Parameter	Description	Example Value
Key	You can perform the following operations: – 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 36 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 perform the following operations: – 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 43 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**

   indicates that database encryption is disabled. This function is disabled by default.

   indicates that database encryption is enabled. 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.

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.
- Snapshots created after the database encryption function is enabled cannot be restored using open APIs.

- Method 1: Select a key name.
- Method 2: Enter the key ID. Enter the key ID used for authorizing the current tenant..

  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**.

Step 11 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.

For details about quotas, see **What Is the User Quota?**

Step 12 Click **Pay Now**.

After the submission is successful, the creation starts. Click **Back to Cluster List** to go back to the **Clusters** page. The initial status of the cluster is **Creating**. Cluster creation takes some time. 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 **Associating and Disassociating ELB**.

----End

3.2.2 Creating a GaussDB(DWS) 3.0 Cluster

GaussDB(DWS) 3.0 uses the cloud-native, cost-effective architecture with decoupled storage and computing. It supports hot and cold data analysis, elastic scaling of storage and computing, unlimited computing power and capacity, and pay-per-use pricing. It is applicable to OLAP analysis scenarios.

This section describes how to create a GaussDB(DWS) 3.0 cluster on the GaussDB(DWS) management console.

NOTE

To balance loads, achieve high availability, and avoid single-node faults, if no ELB is bound during cluster creation, you can bind an ELB on the cluster details page after the cluster is created. For details, see **Associating and Disassociating ELB**.

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**.
- Ensure that the number of available nodes meets the following conditions. Otherwise, the cluster cannot be created.
  - The number of available nodes is greater than or equal to 3. You can view the number of available nodes on the **Cluster > Dedicated Cluster** page.

Creating a Cluster

Step 1 Go to the **page for creating a data warehouse cluster**.

Step 2 Select a region and an AZ.

Step 3 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.

Figure 3-3 Configuring node parameters

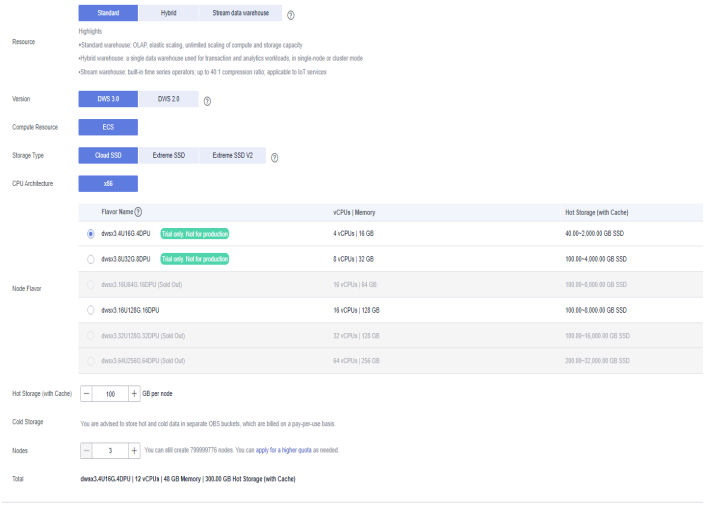


Table 3-6 Node configuration parameters

Parameter	Description	Example Value
Resource	The options are as follows: • **Standard data warehouse**: It can analyze hot and cold data and is highly cost-effective. Its storage and computing resources are not limited, and can be elastically scaled and billed per use. It is suitable for the converged analysis that requires integrated databases, warehouses, marts, and lakes. It is most suitable for OLAP workloads.	Standard
Version	• DWS 3.0 • **DWS 2.0**	DWS 3.0
Compute Resource	It can be: • ESC: Scalable, reliable, and high-throughput virtual block storage is provided in a distributed architecture. This ensures that data can be quickly migrated and restored if any data replica is unavailable, preventing data from being lost because of a single hardware fault. Backup and restoration can be performed on ECSs and EVS disks. You can configure automatic backup policies for them.	-
Storage Type	It can be: • **Cloud SSD**	-

Parameter	Description	Example Value
CPU Architecture	The following CPU architectures can be selected: • **X86** • **Kunpeng** **NOTE** The only difference between the x86 and Kunpeng architectures lies in the underlying architecture, of which the application layer is unaware. The same SQL syntax is used. If x86 servers are sold out when you create a cluster, select 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**.	-
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 Storage	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 4 Click **Next: Configure Network**.

Step 5 Configure the network.

Figure 3-4 Network parameters

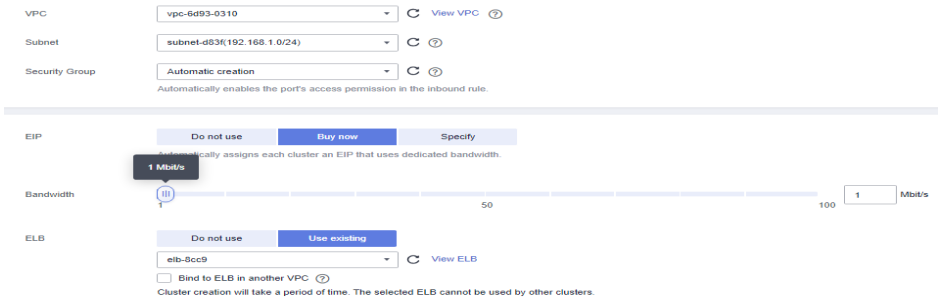


Table 3-7 Network parameters

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.	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. You can select an existing group and retry. ● Manual creation You can also log in to the to manually create a security group. Then, go back to the page for creating data warehouse clusters, click the  button 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.. – **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**: Users specify the bandwidth of the EIP and the system automatically assigns an EIP that exclusively uses bandwidth to each cluster so that users can use the EIP 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 **Create EIP** to go to the **Elastic IP** page and create an EIP as needed. The bandwidth can be customized. **NOTE** • 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

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 load balancers are displayed in the drop-down list, click **Create ELB** to go to the Elastic Load Balance page and create a load balancer as needed.	Specify
Bandwidth	Specifies the EIP bandwidth. The value ranges from 1 Mbit/s to 100 Mbit/s. This parameter is mandatory if **EIP** is set to **Buy now**.	50Mbit/s

Step 6 Click **Next: Configure Advanced Settings**.

Step 7 Configure cluster parameters.

Table 3-8 Cluster parameters

Parameter	Description	Example Value
Cluster Name	Set the name of the data warehouse cluster. The cluster name contains 4 to 64 case-insensitive characters and must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed. **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 version in the screenshot 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	Set the database administrator name. The username must meet the following requirements: • Consists of lowercase letters, digits, or underscores. • Starts with a lowercase letter or an underscore. • Contains 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	Set the password of the database administrator account. The password complexity requirements are as follows: • Contains 12 to 32 characters. • Cannot be the username or the username spelled backwards. • Must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-) • Passes 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 6** 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.	-

Parameter	Description	Example Value
Time Zone	You can set the time zone for the tenant cluster, including the system OS time zone and cluster data warehouse time zone.	-

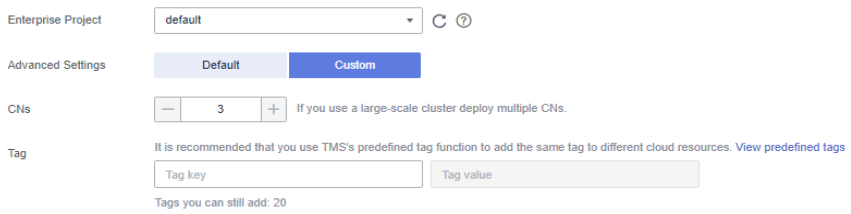
Step 8 Select the enterprise project of the cluster. You can configure this parameter only when the Enterprise Project Management service is enabled. The default value is **default**.

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

Figure 3-5 Custom advanced settings



- **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.

- **Tags**

  A tag is a key-value pair used to identify a cluster. For details about the keys and values, see **Table 3-9**. By default, no tag is added to the cluster.

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

Table 3-9 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 the *Tag Management Service User Guide*. – Enter a tag key in the text box. A tag key can contain a maximum of 36 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 43 characters, which can be an empty string. It cannot start or end with a space. The value cannot contain the following characters: =*<>\,|/	value01

Step 10 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.

For details about quotas, see **What Is the User Quota?**

Step 11 Click **Next**.

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

3.3 Purchasing a Discount Package

GaussDB(DWS) also supports discount packages. You can make a one-off payment according to the purchased service duration. The service duration ranges from one month to three years. It is economical and recommended for long-term users.

This section describes how to purchase a GaussDB(DWS) discount package.

NOTE

- A discount package is paid in advance by month or year. If you plan to use GaussDB(DWS) for a long term, nodes in a discount package are more cost-effective than nodes billed on a pay-per-use basis.
- A discount package is a billing concept. After you purchase a discount package, no cluster will be automatically created. Go to the GaussDB(DWS) management console to create a cluster if no cluster exists.
- A discount package takes effect from the date of purchase and automatically expires when the subscription period ends. This period is fixed even if you do not run a GaussDB(DWS) cluster. To reduce costs, create a cluster immediately after purchasing a discount package, or purchase the package after creating a cluster.
- When a discount package expires or you unsubscribe from the package, the system will automatically charge you on a pay-per-use basis (by hour). The service will not be interrupted as long as your account balance is sufficient.
- Storage unit conversion: 1 PB = 1024 TB; 1 TB = 1024 GB; 1 GB = 1024 MB; 1 MB = 1024 KB

Purchasing a Discount Package

Step 1 Go to the **discount package purchase** page.

Step 2 On the displayed page, select a region.

Table 3-10 Region parameters

Parameter	Description	Example Value
Region	Select the AZ for the cluster nodes to run. Discount packages in different regions are isolated.	EU-Dublin

Step 3 (Optional) If you select **Cold storage**, configure the specifications, quantity, and required duration. The estimated price will be displayed. The cold storage package is applicable to scenarios where cold storage has been used or will be used.

Table 3-11 Parameters

Parameter	Description	Example Value
Cold storage	Total Storage Capacity (GB) **NOTE** Multiple packages can be used together. If used resources exceed the package size, the excess part will be charged on a pay-per-use basis.	-
Purchase Quantity	Number of packages.	3
Validity Period	Validity period of a package. Before the package expires, the cold data usage can be deducted from the package first. Excess usage will be charged on a pay-per-use basis.	-

Step 4 (Optional) If you select **OBS hot storage**, configure the specifications, quantity, and required duration. The estimated price will be displayed. The DWS 3.0 storage package is applicable to scenarios where DWS 3.0 has been used or will be used.

Table 3-12 Parameters

Parameter	Description	Example Value
DWS 3.0 storage	Package size. **NOTE** Multiple packages can be used together. If used resources exceed the package size, the excess part will be charged on a pay-per-use basis.	-
Purchase Quantity	Quantity	3
Validity Period	Validity period of a package. Before the package expires, the GaussDB(DWS) 3.0 storage can be deducted from the package first. Excess usage will be charged on a pay-per-use basis.	-

Step 5 Click **Next** to switch to the **Confirm** page.

Step 6 Confirm the order information and click **Pay Now**.

Step 7 Select a payment method and complete the payment as prompted.

After the payment is successful, the order takes effect about 5 minutes later. You can use the purchased package only after the order takes effect. You can choose **Billing Center** > **My Orders** in the upper right corner of the console to go to the **My Orders** page and view the order status.

No cluster will be automatically created after you purchase a discount package. If you have not created a cluster, go to the GaussDB(DWS) management console to create one. For details, see **Creating a Cluster**.

After the order takes effect, if you create a data warehouse cluster with the same flavor as the package cluster flavor you purchased, the cluster is automatically associated with the discount package. The nodes within the discount package range will not be charged during the validity period and the extra nodes will be charged on the pay-per-use basis.

----End

3.4 Yearly/Monthly Billing

If you want to use a pay-per-use cluster for a long time, you can change its billing mode to yearly/monthly. This section describes the following operations:

- **From Pay-per-use to Yearly/Monthly**
- **From Yearly/Monthly to Pay-per-use**
- **Renewing a Yearly/Monthly Subscription**
- **Unsubscribing from a Yearly/Monthly Subscription**

NOTE

- Pay-per-use: a postpaid billing mode suitable in scenarios where clusters will be billed based on usage duration. You can provision or delete clusters at any time.
- **Yearly/Monthly**: a prepaid billing mode, in which a cluster is billed based on the purchase period. This mode is more cost-effective than the pay-per-use mode and applies if the resource usage period can be estimated.

  If the current console does not support this billing mode, contact technical support.

From Pay-per-use to Yearly/Monthly

Prerequisites

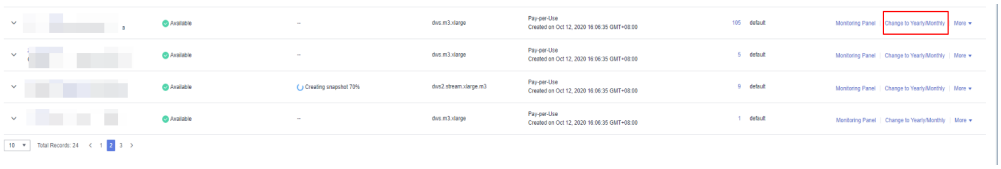
The billing mode of the cluster is pay-per-use.

Procedure

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

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters will be displayed by default.

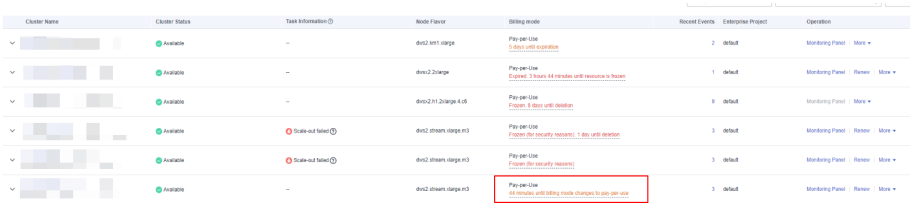
Step 3 In the **Operation** column of a cluster, click **Change to Yearly/Monthly**.



Step 4 On the CBC page, set **Renew Duration**, configure **Auto-Renew** as needed, and click **Confirm**.

Step 5 Confirm the information and click **Pay** to pay for the order.

Step 6 Return to the cluster list and click  . The billing mode of the pay-per-use cluster will change to yearly/monthly.



----End

From Yearly/Monthly to Pay-per-use

Prerequisites

- The cluster billing mode is yearly/monthly.
- The pay-per-use billing mode will take effect after the original yearly/monthly subscription has expired.

  NOTE

  Yearly/Monthly clusters cannot be changed to pay-per-use clusters within the grace period and retention period.

Procedure

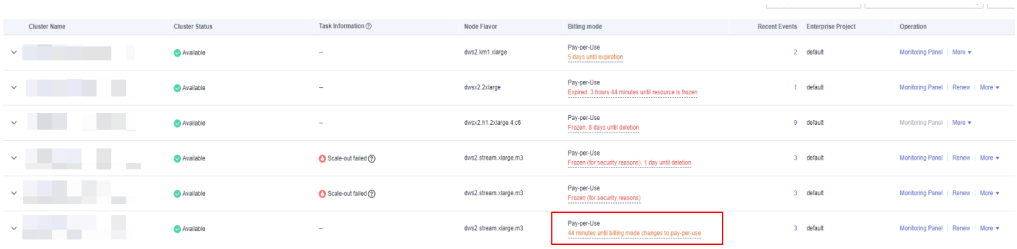
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

Step 3 In the cluster list, locate the row that contains the target cluster, choose **More** > **Change to Pay-per-use** in the **Operation** column.

Step 4 On the displayed page, click **Change to Pay-per-Use**.

Step 5 Return to the cluster list and click  . The billing mode of the yearly/monthly cluster will change to pay-per-use after the yearly/monthly subscription expires.



----End

Renewing a Yearly/Monthly Subscription

Prerequisites

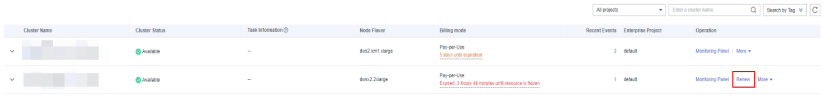
The cluster billing mode is yearly/monthly.

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 a cluster, click **Renew**.



Step 4 The CBC renewal page is displayed. Confirm the information and pay for the order.

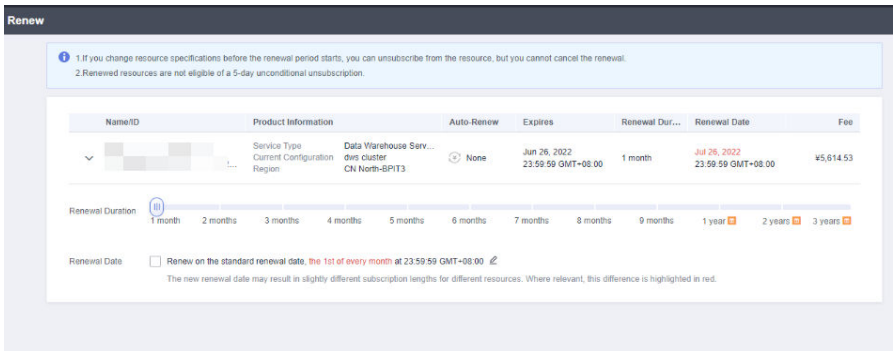
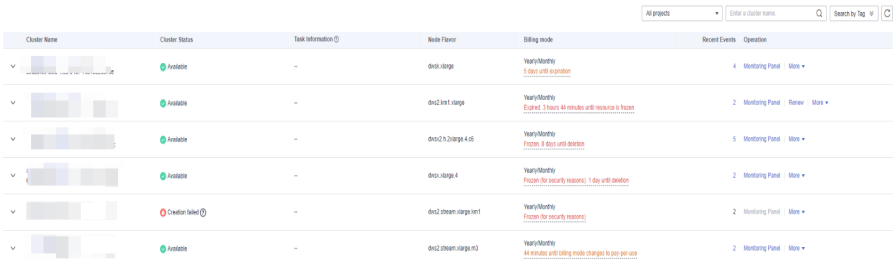


Table 3-13 Renewal parameters

Parameter	Description
Renewal Duration	Select the renewal duration.
Renewal Date	Select a renewal data. By default, a subscription expires on the last day of every month. You can choose whether to change the expiry date to the first day of every month. **NOTE** The new renewal date may extend the subscription of some resources based on the current subscription.

Step 5 Return to the cluster list and click  to refresh it.



----End

Unsubscribing from a Yearly/Monthly Subscription

Prerequisites

The cluster billing mode is yearly/monthly.

NOTE

- Yearly/Monthly clusters cannot be unsubscribed from during the grace period or retention period. You can release these clusters on the **Renewals** page of the Billing Center.
- A yearly/monthly cluster is frozen during the retention period. Snapshots are frozen together with the cluster and will be automatically deleted after the retention period expires.
- A yearly/monthly cluster that has been unsubscribed from cannot be restored. The user data and automated snapshots in the cluster are automatically deleted and cannot be accessed anymore. The manual snapshots of a yearly/monthly cluster will not be deleted when you unsubscribe from the cluster.

Precautions

To change the billing mode of a yearly/monthly cluster to pay-per-use before a renewal period takes effect, you can only unsubscribe from the cluster, but cannot cancel the renewal.

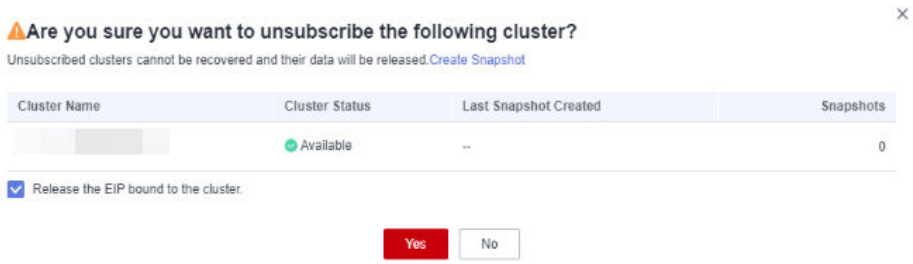
Procedure

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

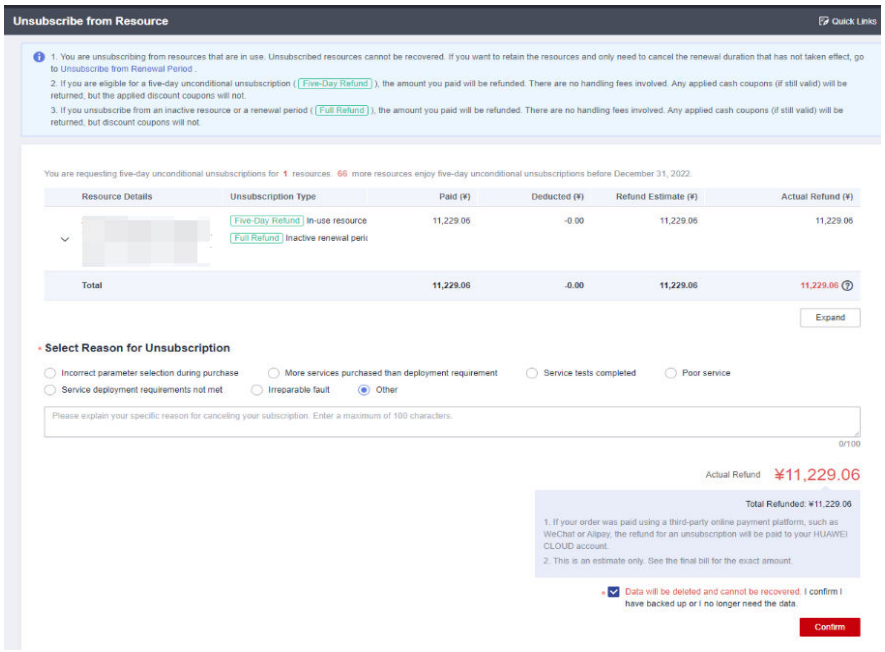
Step 2 Choose **Clusters** > **Dedicated Clusters**. All clusters are displayed by default.

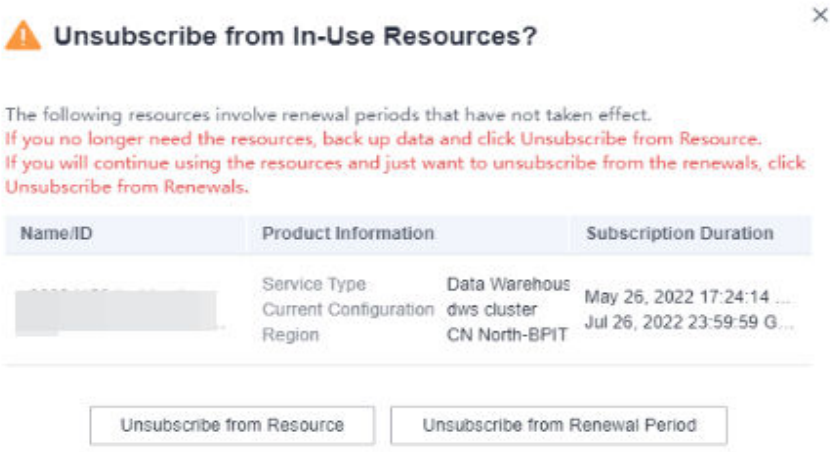
Step 3 In the **Operation** column of a cluster, choose **More** > **Unsubscribe**.

Step 4 In the displayed dialog box, click **Yes**.



Step 5 On the CBC unsubscription page, select a reason for unsubscription, and click **Confirm**. In the displayed dialog box, click **Unsubscribe from Resource**. After the order is submitted, the page will be automatically refreshed.





----End

3.5 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.
- A cluster that is read-only or being scaled cannot be deleted. You can delete it only after the scaling is complete or the read-only state is canceled.
- 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.

Deleting a Cluster

Step 1 Log in to the GaussDB(DWS) management 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, you can click **Create Snapshot**. In the dialog box that is displayed, enter the snapshot name and click **OK** to create a snapshot for the cluster to be deleted. After the snapshot is created, go back to the **Clusters** > **Dedicated Clusters** page to delete the cluster.
- Resource
  - 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.
  - Automated Snapshots
  - 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

4 Cluster Connection

4.1 Methods of Connecting to a Cluster

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.

The procedure for connecting to a cluster is as follows:

1. **Obtaining the Cluster Connection Address**
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:
   - 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 the Data Studio GUI Client to Connect to a Cluster**
     - **Using DAS to Connect to a 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 Cluster Connection Address

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

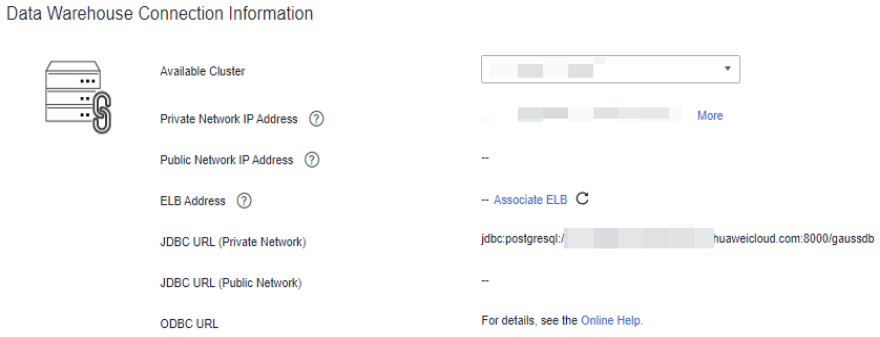
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 In the navigation tree on the left, choose **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



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) management 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

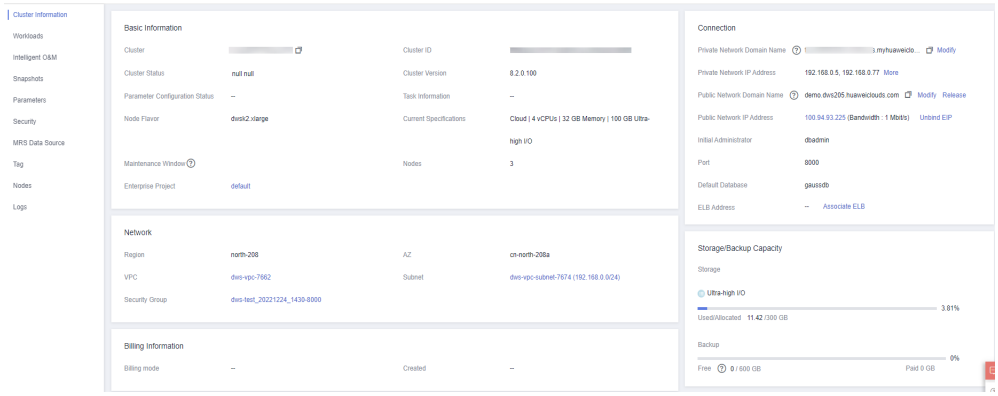


Table 4-1 Connection

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. **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 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 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.
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.

Parameter	Description
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 DAS to Connect to a Cluster

GaussDB(DWS) supports page login (WebSQL). This function depends on Data Admin Service (DAS). Currently, database management, SQL operations, and operation audit are supported. To connect to a cluster in this way, you need to enter the database username and password. You can view metadata and run SQL statements after connection.

NOTE

- Only cluster 8.0.1 and later versions support the cluster login function. The hybrid data warehouse (standalone) does not support this function.
- By default, only Huawei Cloud accounts or users with **DWS Administrator** permissions can log in to clusters. IAM users in the account do not have the permission by default. An IAM user needs to be authorized by a user who has the permission.
- Supported regions: Huawei Cloud regions where DAS is available
- If your cluster is in arrears, this function may be unavailable. Please top up your account in time.

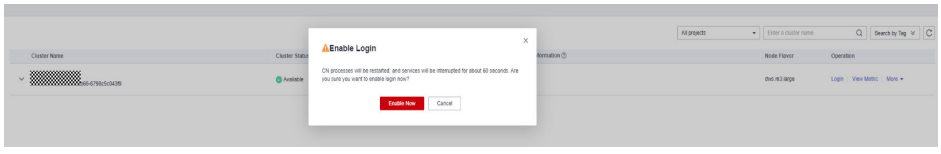
Enabling the Login Function

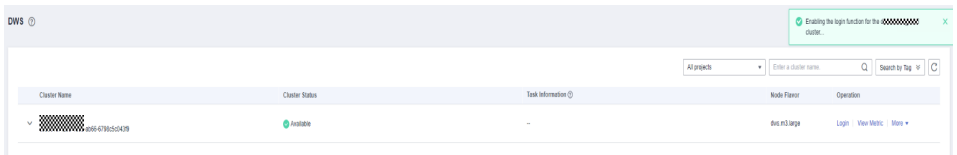
Step 1 Log in to the GaussDB(DWS) management console.

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

Step 3 In the cluster list, click **Log In** in the **Operation** column of a cluster.

Step 4 If the login function is not enabled, the **Enable Login** dialog box is displayed. Click **Enable** and click **Enable Now**.





> **NOTE**
>
> CN processes will be restarted, and services will be interrupted for about 60 seconds. You are advised to perform this operation in a proper time window.

----End

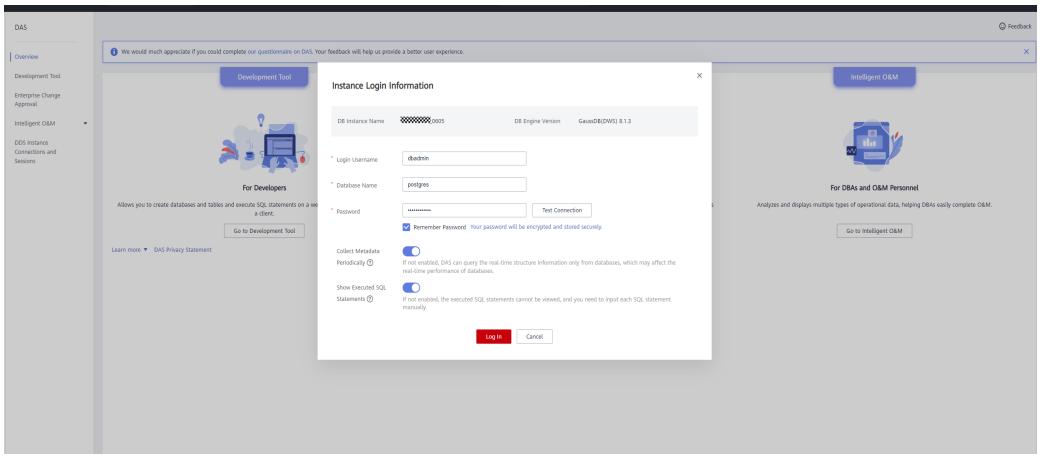
Logging In

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

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

Step 3 In the cluster list, click **Login** in the **Operation** column of a cluster.

Step 4 After you are redirected to the DAS page, enter the login username, database name, and password, and enable the scheduled collection and SQL execution history functions.



> **NOTE**

- You are advised to enable **Collect Metadata Periodically**. If it is disabled, DAS obtains only the structured data from databases in real time, and the performance of databases is affected.

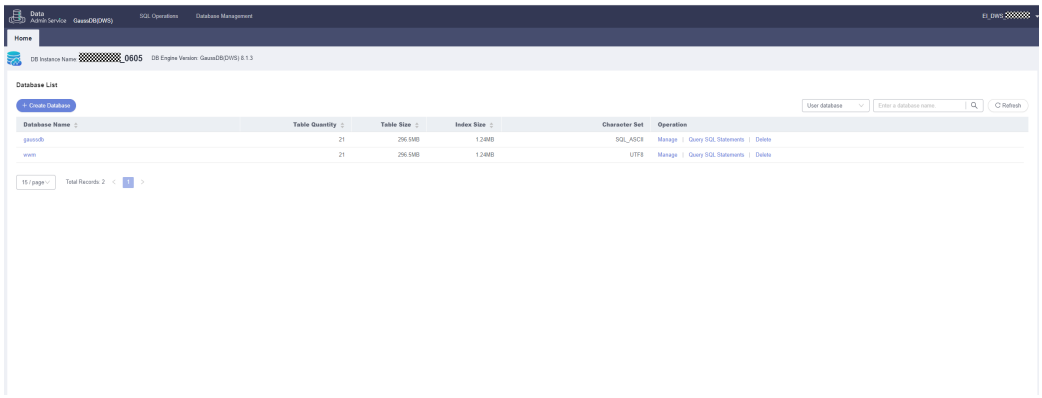
  The collection time cannot be customized. Once **Collect Metadata Periodically** is enabled, DAS automatically collects metadata at 20:00 every day (UTC time). If you are not using a UTC time, convert the time according to your local time zone. You can also click **Collect Now** to collect metadata at any time you want.

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

  Note: SQL statements that contain sensitive keywords are not recorded. Sensitive keywords include (case-insensitive): "create user", "password", "grant", "revoke", "create login", "sp_addrole", "sp_droprole", "sp_addlogin", "sp_grantdbaccess", "sp_addrolemember", "sp_revokedbaccess", "sp_password", "sp_droplogin", "create role", "dblink_connect", "gs_encrypt_aes128", "gs_decrypt_aes128", "gs_encrypt", "gs_decrypt", "gs_hash", "gs_extend_library", "exec_on_extension", "exec_hadoop_sql", "secret_access_key", "dli_secret_access_key", "filepath", "username", "digest", "hmac", "crypt", "pgp_sym_encrypt", "pgp_sym_encrypt_bytea", "pgp_sym_decrypt", "pgp_sym_decrypt_bytea", "pgp_pub_encrypt", "pgp_pub_encrypt_bytea", "pgp_pub_decrypt", "pgp_pub_decrypt_bytea", "pgp_key_id", "encrypt", "decrypt", "encrypt_iv" and "decrypt_iv".

Step 5 Click **Test Connection**. If a message is displayed indicating connection successful, continue with the operation. If a message is displayed indicating connection failed and the failure cause is provided, make modifications based on the error message.

Step 6 Click **Log In**. The database page will be displayed.



----End

Operation Audit

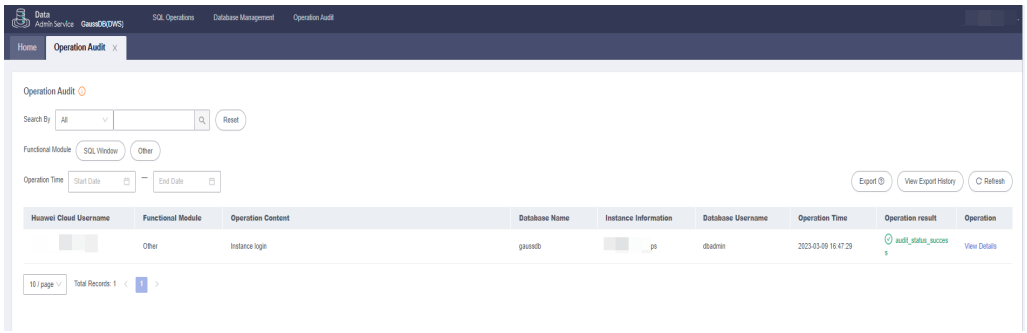
Step 1 Log in to the GaussDB(DWS) management console.

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

Step 3 In the cluster list, click **Log In** in the **Operation** column of a cluster.

Step 4 Switch to the DAS page and log in. For details, see **Logging In**.

Step 5 Choose **Operation Audit** form the navigation pane. Check the login, logout, creation, deletion, and query records on DAS. These records include the Huawei username for logging in to the DAS, function modules, operation content, database names, cluster information, login database names, operation time, and execution status.

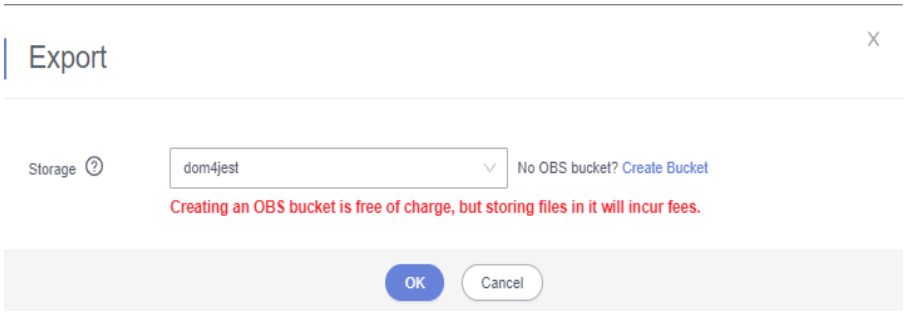


----End

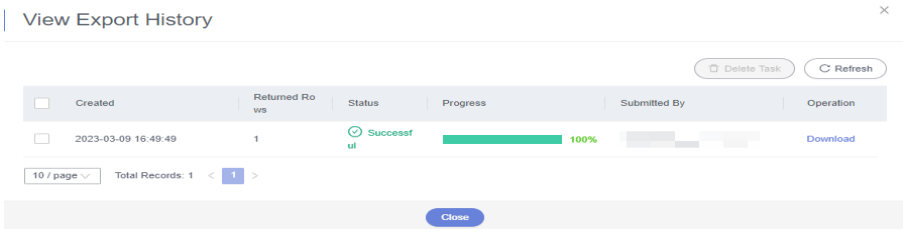
Exporting Audit Records

The export function allows you to filter data in the audit list and export it to a CSV file in asynchronous mode. That is, you need to save the file to OBS and then download it to a local PC.

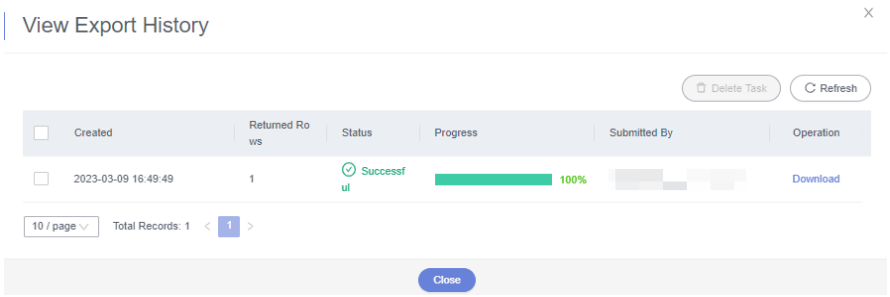
Step 1 Open the **Operation Audit** page. For details, see **Operation Audit**. Click **Export** in the upper right corner to add an asynchronous export task.



Step 2 Click **View Export History** in the upper right corner of the list to view the asynchronous export progress. Wait until the task is complete.



Step 3 In the **View Export History** dialog box, locate the row that contains the target task and click **Download** in the **Operation** column to download the CSV file to the local PC.

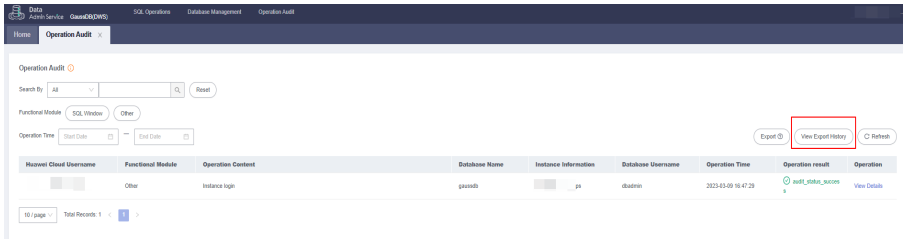


----End

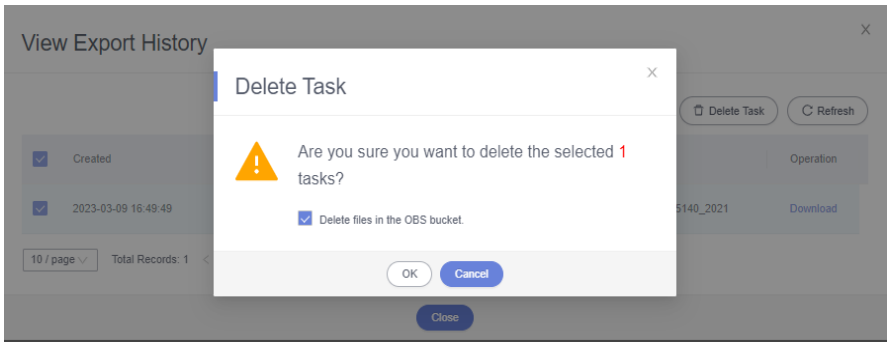
Deleting Export Records

Step 1 Open the **Operation Audit** page. For details, see **Operation Audit**.

Step 2 Click **View Export History** in the upper right corner of the list.



Step 3 Select export tasks and click **Delete Task** in the upper right corner of the list. You can determine whether to delete files in the OBS bucket at the same time.



----End

4.4 Using the Data Studio GUI Client to Connect to a 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 Cluster Connection Address**.
- 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) management console.

Step 3 Click **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.

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.

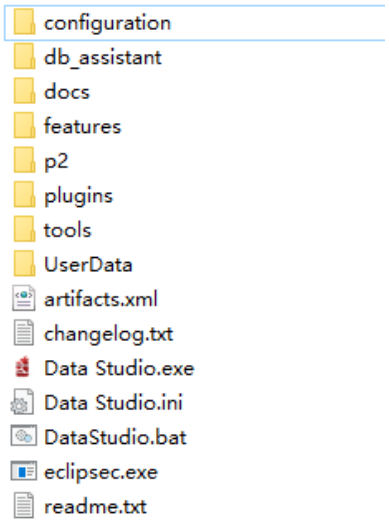
Table 4-2 Data Studio download links

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
	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 4-3**.

Figure 4-3 Starting the client

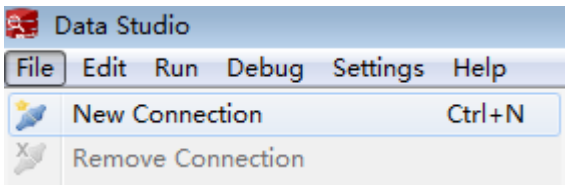


NOTE

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 4-4**.

Figure 4-4 New connection



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

Table 4-3 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

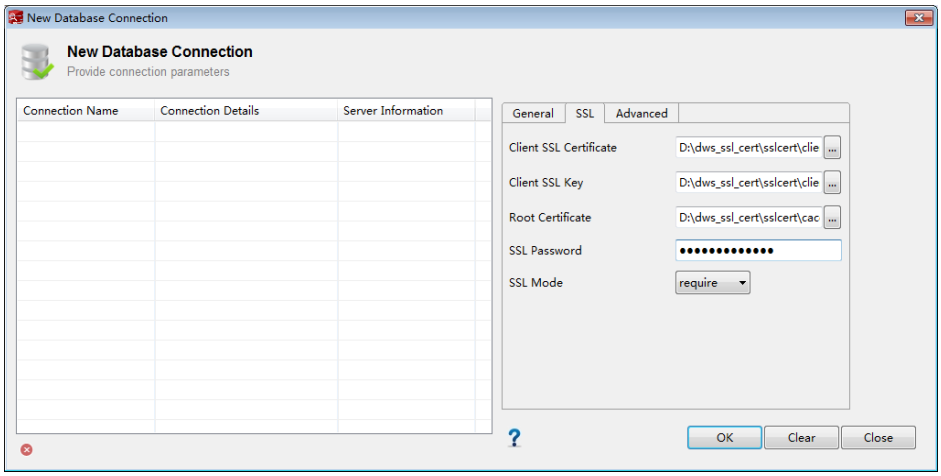
Field	Description	Example Value
Username	Username for connecting to the database	-
Password	Password for logging in to the database to be connected	-
Save Password	Select an option from the drop-down list: • **Current Session Only**: The password is saved only in the current session. • **Do Not Save**: 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.	-

When **Enable SSL** is selected, download the SSL certificate and decompress it by referring to **Downloading SSL Certificate**. Click the **SSL** tab and configure the following parameters:

Table 4-4 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.
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 4-5 Configuring SSL parameters

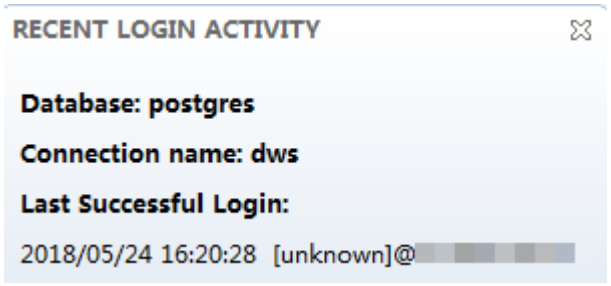


Step 9 Click **OK** to establish the database connection.

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.

Figure 4-6 Successful login



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

4.5 Using the gsql CLI Client to Connect to a Cluster

4.5.1 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.

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 when you start an 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. For details, see **Accessing the GaussDB(DWS) Management Console**.

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

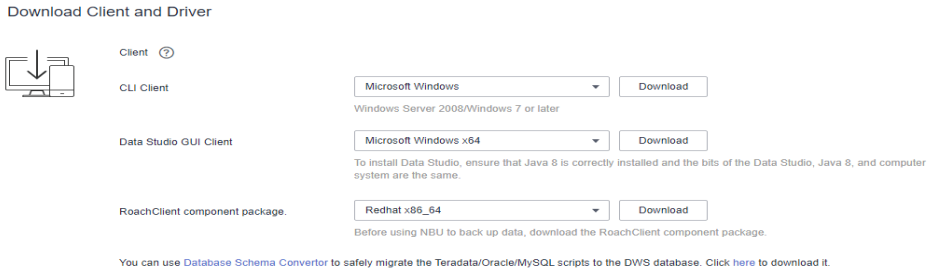
Step 3 Select the GaussDB(DWS) client of the corresponding version from the drop-down list of **gsql CLI Client**.

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

Table 4-5 gsql download links

OS Type	Applicable OS	Download Link	Verification File
Microsoft Windows	Microsoft 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
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

Figure 4-7 Downloading a gsql client



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.
- Select **Microsoft Windows** from the Windows **gsql** package drop-down list. You will get the 32-bit and 64-bit executable binary files.

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 4-6 Files and folders in the Linux gsql tool package

File or Folder	Description
bin	This folder contains the executable files of gsql on Linux, including the 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 4-7 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

4.5.2 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:

- The ECS and data warehouse cluster 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 to RHEL 7.6
    - CentOS 6.4 to CentOS 7.4

    - EulerOS 2.3
  - The **SUSE x86_64** client can be used on the following OSs:
    - SLES 11.1 to SLES 11.4
    - SLES 12.0 to SLES 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.

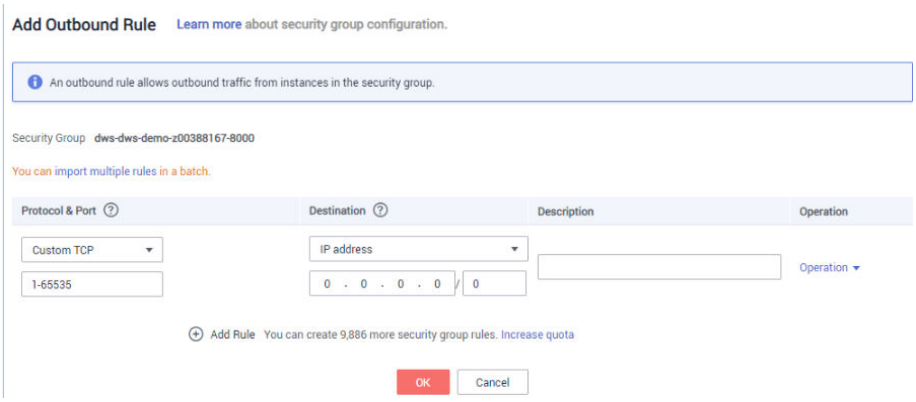
  For details about VPC operations, see **VPC and Subnet** in the *Virtual Private Cloud User Guide*.
- If the client accesses the cluster using the public network address, ensure that both the created ECS and GaussDB(DWS) cluster have an EIP.

  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/Application**: The value must contain **TCP**, for example, **TCP** and **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 4-8 Outbound rule

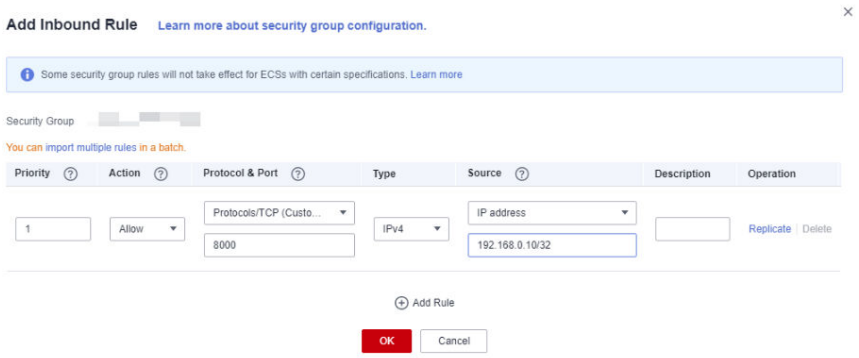


- The security group rules of the data warehouse cluster must ensure that GaussDB(DWS) can receive network access requests from clients.

  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/Application**: The value must contain **TCP**, for example, **TCP** and **All**.
  - **Port**: Set this parameter to the database port that provides services in the data warehouse cluster, for example, **8000**.
  - **Source**: The IP address set here must contain the IP address of the GaussDB(DWS) client server, for example, **192.168.0.10/32**.

Figure 4-9 Inbound rule



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.

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 GaussDB(DWS) 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 Cluster Connection Address**. 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**. 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.eu-dublin.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 details about how to obtain the AK and SK, see **Creating Access Keys (AK and SK)**. Then, 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

4.5.3 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 4-10 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.

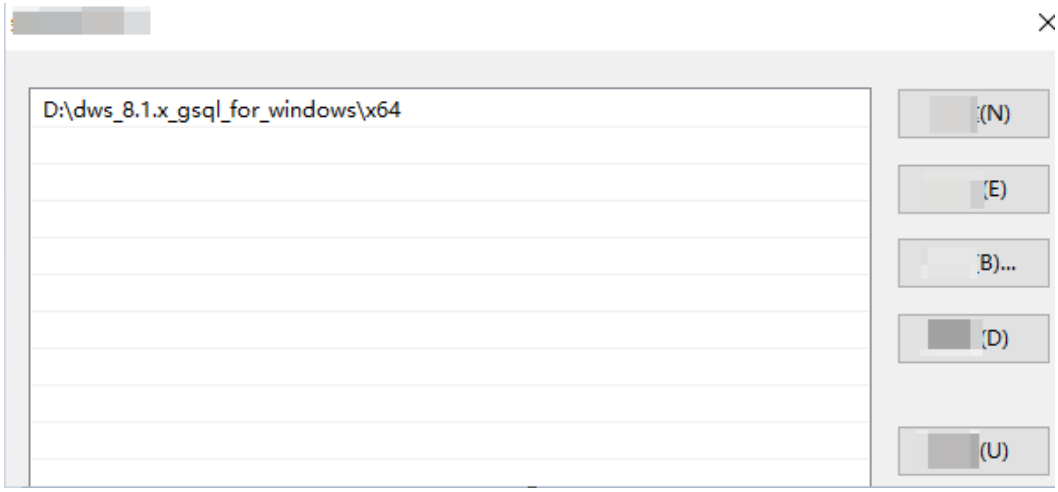
Step 4 Set environment variables. For a 32-bit OS, select the **x86** folder. For a 64-bit OS, select the **x64** folder.

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 4-11 Configuring Windows environment variables



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 Cluster Connection Address**. 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 **\!** metacommand 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
```

gsql Command Reference

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

4.5.4 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

- After you have modified the security parameters and the modifications take effect, the cluster may be restarted, which makes the cluster unavailable temporarily.
- 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) management 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

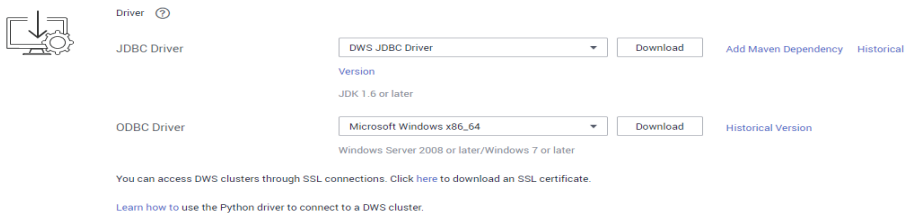
Configuring Digital Certificate Parameters Related to SSL Authentication on the gsql Client

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) management console. In the navigation pane, choose **Client Connections**.

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

Figure 4-12 Downloading an SSL certificate



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. Different authentication modes require different

client environment variables. 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 4-8** shows the differences between these two modes. You are advised to use bidirectional authentication for security purposes.

Table 4-8 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 link spoofing, 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 4-9**.

NOTE

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

Table 4-9 Client parameters

Environment Variable	Description	Value Range
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 Range
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) management 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) management 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 4-10 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.

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

4.6.1 Development Specifications

If the connection pool mechanism is used during application development, the following specifications must be met. Otherwise, connections in the connection pool have statuses, which will affect the correctness of subsequent operations on 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.

4.6.2 JDBC Version Description

Version 8.3.0.202

- New features:

  Added the tcpKeepAlive configuration,

  which takes effect only when **tcpKeepAlive** is set to **true**.

  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:

  1. An error is reported when **reWriteBatchedInserts** is used to insert data in batches.

  2. "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.

4.6.3 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

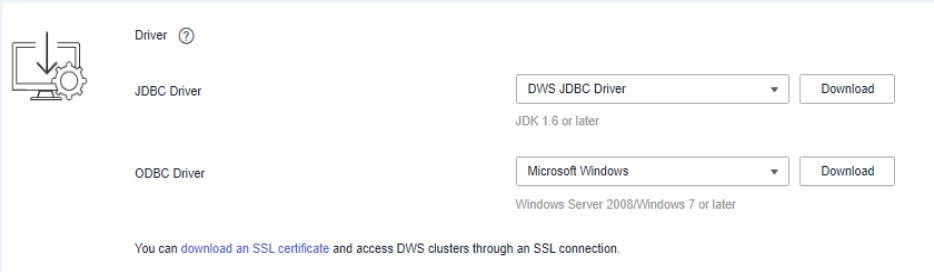
Downloading the JDBC or ODBC Driver

Step 1 Log in to the GaussDB(DWS) management 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 4-13 Downloading the driver



- **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.

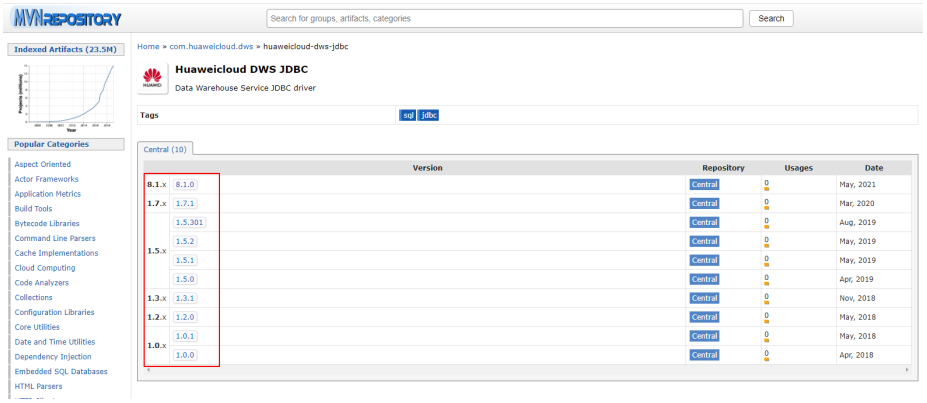
Table 4-11 JDBC driver download address

Driver	Download Link	Verification File
DWS JDBC Driver	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

Method 2:

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

Figure 4-14 Maven page



In the list shown in **Figure 4-14**, 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 4-15 Maven dependency

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>
```


- **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 4-12 ODBC driver download address

Applicable OS	Download Link	Verification File
Microsoft 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
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

4.6.4 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.

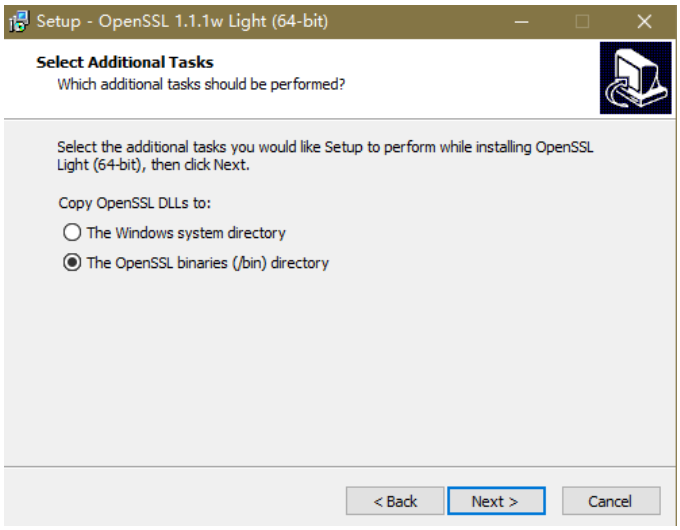
Step 1 Determine whether you want to use the SSL mode to connect to the GaussDB(DWS) cluster.

- 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**.

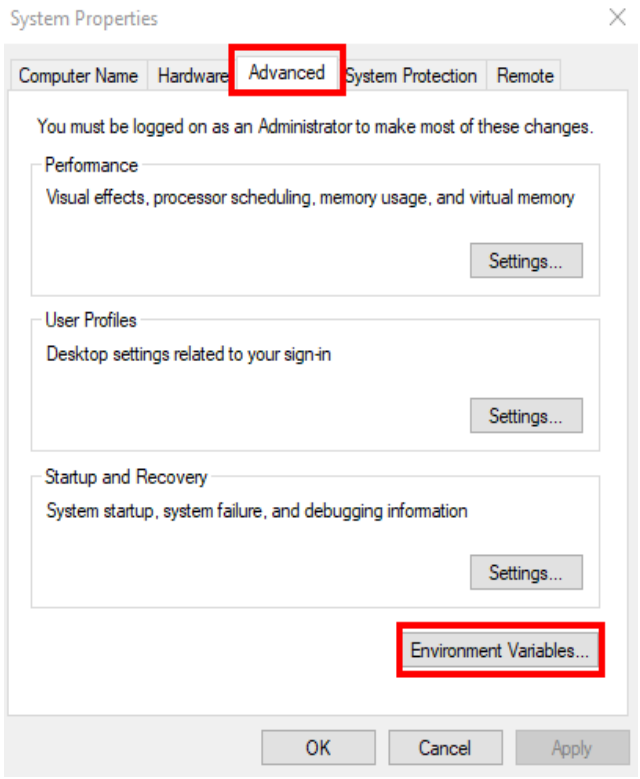
Step 2 (Optional) On Linux, use WinSCP to upload the downloaded SSL certificate file to the Linux environment.

Step 3 Configure the certificate to enable SSL connection.

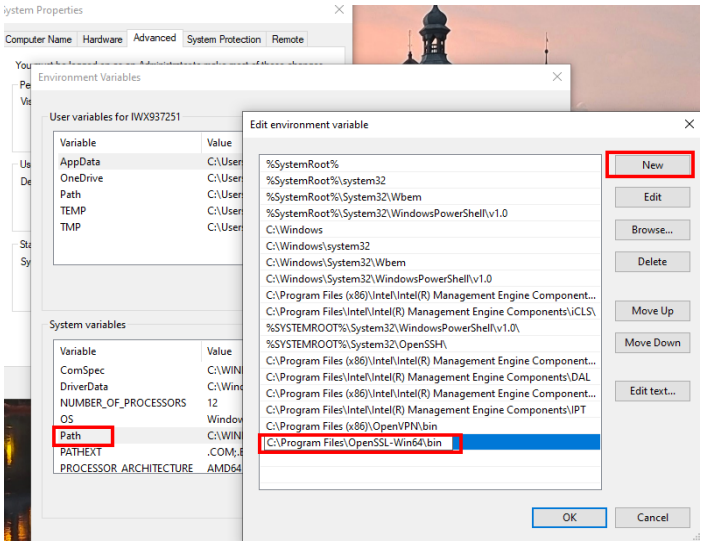
1. Download the OpenSSL toolkit for Windows at **https://slproweb.com/products/Win32OpenSSL.html**. OpenSSL 3.0.0 is currently not supported. Download Win64 OpenSSL v1.1.1w Light instead.
2. Double-click the installation package **Win64OpenSSL_Light-1_1_1w.exe** and install it to the default path on 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 successful.



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**.



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.



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*

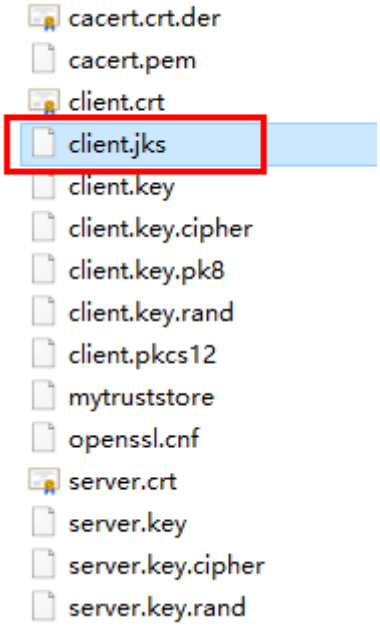
   Enter the client private key password **Gauss@MppDB**. Then enter and confirm the self-defined private key password.

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**.

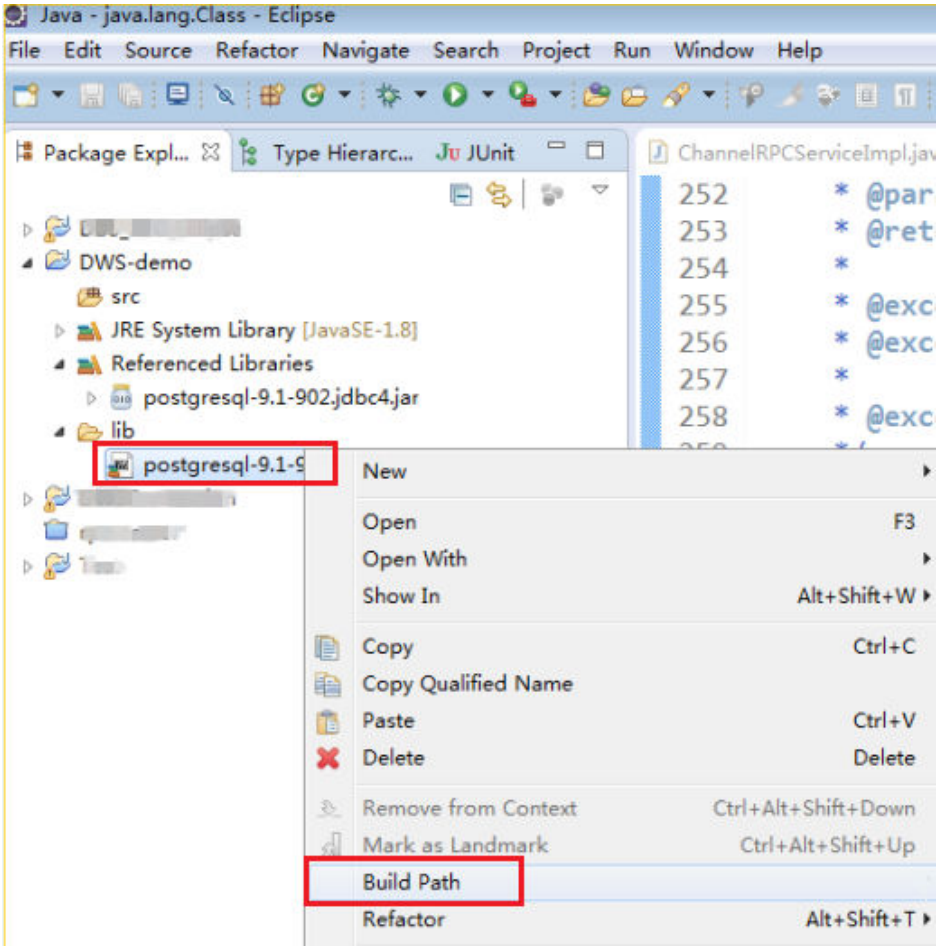


Step 4 Download the driver package **dws_8.1.x_jdbc_driver.zip** and decompress it. There will be two JDBC drive JAR packages, **gsjdbc4.jar** and **gsjdbc200.jar**. Use either of them as required.

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 4-16 Referencing a JAR file



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** (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**.
  - 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 4-13 Database connection parameters

Parameter	Description
url	Specifies the database connection descriptor, which can be viewed on the management console. For details, see **Obtaining the Cluster Connection Address**. 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**.
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" + '/' + "postgresgaussdb";
    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();
    }
```

```
 }
}
```

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

Context

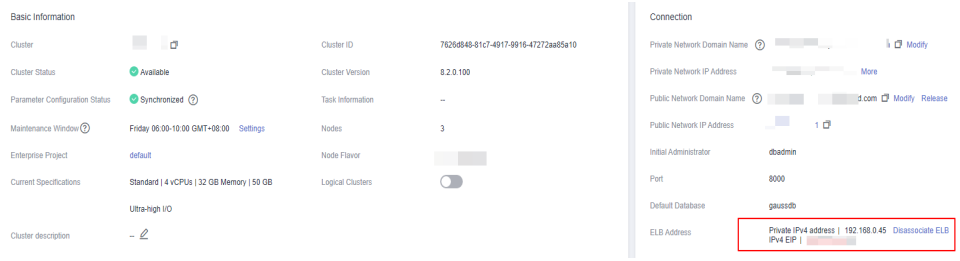
If you use JDBC to connect to only one CN in the cluster, this CN may be overloaded and other CN resources wasted. It also incurs single-node failure risks.

To avoid these problems, you can use JDBC to connect to multiple CNs. Two modes 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.
- 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.



Step 2 Configure the 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.

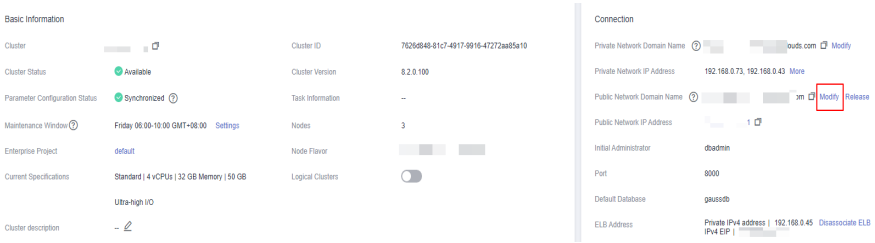
```
private static final String USER_NAME = "dbadmin";
private static final String PASSWORD = "password";
// jdbc:postgresql://ELB_IP:PORT/dbName"
private static final String URL = "jdbc:postgresql://100.95.153.169:8000/gaussdb";
private static Properties properties = new Properties();
static {
    properties.setProperty("user", 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: 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.

```
<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

4.6.6 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**.
- Stream data warehouses do not support the connection to a cluster in IAM authentication mode.

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.

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.

----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 4-14 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:eu-dublin-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. ● **eu-dublin-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. ● **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.

Parameter	Description
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;
}
```


4.6.7 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]
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 Cluster Connection Address**.

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 can choose to automatically or manually deploy the certificate based on your needs.

Automatic deployment:

Double-click the **sslcert_env.bat** file. The certificate is automatically deployed 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:

1. Create a new folder named **postgresql** in the **%APPDATA%** directory.
2. 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**.
3. 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.

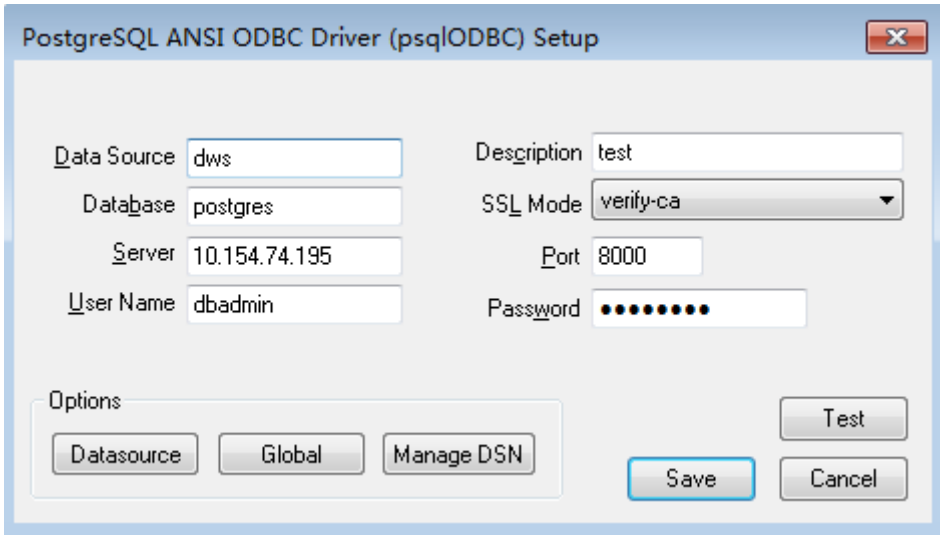
- In a 32-bit OS, open **C:\Windows\System32\odbcad32.exe**.

  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 4-17 Configuring a data source to be connected to



   You can view the values of **Server** 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 Cluster Connection Address**.

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

4.7 Using the Third-Party Function Library psycopg2 of Python to Connect to a Cluster

After creating a data warehouse cluster and using the third-party function library 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 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 set **password_encryption_type** to **1**. For details, see **Modifying Database Parameters**.

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 Cluster Connection Address**.
- You have installed the third-party function library 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, 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

psycopg2 is a PostgreSQL-based client interface, and its functions are not fully supported by GaussDB(DWS). For details, see **Table 4-15**.

NOTE

The following APIs are supported based on Python 3.8.5 and psycopg 2.9.1.

Table 4-15 psycopg2 APIs supported by DWS

Class Name	Usage	Function/Member Variable	Yes	Remarks
connections	basic	cursor(*name=None*, *cursor_factory=None*, *scrollable=None*, *withhold=False*)	Y	-
		commit()	Y	-
		rollback()	Y	-

Class Name	Usage	Function/Member Variable	Yes	Remarks
		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	-
	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	-

Class Name	Usage	Function/Member Variable	Yes	Remarks
		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.
	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	-

Class Name	Usage	Function/Member Variable	Yes	Remarks
		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.
	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	-

Class Name	Usage	Function/Member Variable	Yes	Remarks
		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	-
	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 Run the following command to connect to the cluster using the third-party function library 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. 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()
```

4.8 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 set **password_encryption_type** to **1**. For details, see **Modifying Database Parameters**.

  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 Cluster Connection Address**.
- 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 4-16**.

NOTE

The following APIs are supported based on Python 3.8.5 and PyGreSQL 5.2.4.

Table 4-16 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	-
	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	-

PyGreSQL		Yes	Remarks
	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	-
	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	-

PyGreSQL		Yes	Remarks
	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**.
	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	-

PyGreSQL		Yes	Remarks
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	-
	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	-

PyGreSQL		Yes	Remarks
	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	-
	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	-

PyGreSQL		Yes	Remarks
	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	-
	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**.

PyGreSQL		Yes	Remarks
	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	-
	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	-

PyGreSQL		Yes	Remarks
	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	-
	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

4.9 Managing 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 4-17 Number of supported connections

Parameter	Number of CN Connections	Number of DN Connections
max_connections	800	Max (Number of vCPU cores/Number of DNs on a single node x 120 + 24, 5000)

NOTE

- The policies of **max_pool_size** and **max_prepared_transactions** are the same as those of **max_connections**.
- For details about CNs and DNs, see **Logical Cluster Architecture**.

Viewing the Maximum Number of Connections

Step 1 Use the SQL client tool to connect to the database in a cluster.

Step 2 Run the following command:

```
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)
```

----End

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 4-18**.

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 4-18 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)`

View the maximum number of sessions connected to a specific database.	Run the following command to view the upper limit of connections used by the database:
	`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 the **gaussdb** database is not limited.
	`datname | datconnlimit`
	`----------+--------------`
	`gaussdb |  -1`
	`(1 row)`

View the number of session connections that have been used by a database.	Run the following command to view the number of session connections that have been used by the database:
	`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 the **gaussdb** database.
	`count`
	`-------`
	`1`
	`(1 row)`

Description	Command
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

5 Monitoring and Alarms

5.1 Dashboard

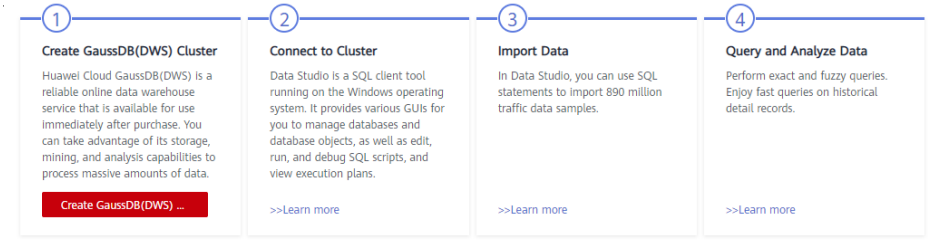
There are two types of Dashboard pages, **Dashboard Without Cluster Overview** and **Dashboard With Cluster Overview**. The dashboard page displayed on your console is determined by whether you have purchased any cluster or not.

Dashboard Without Cluster Overview

The dashboard page consists of the following modules:

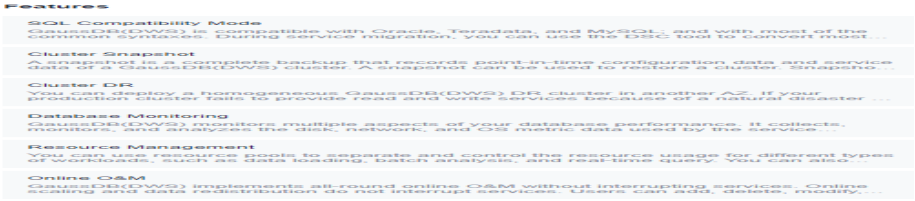
- Process

  This module describes how to create a cluster, connect to a cluster, and import sample data from OBS to GaussDB(DWS). You can click **Learn more** to check more information.

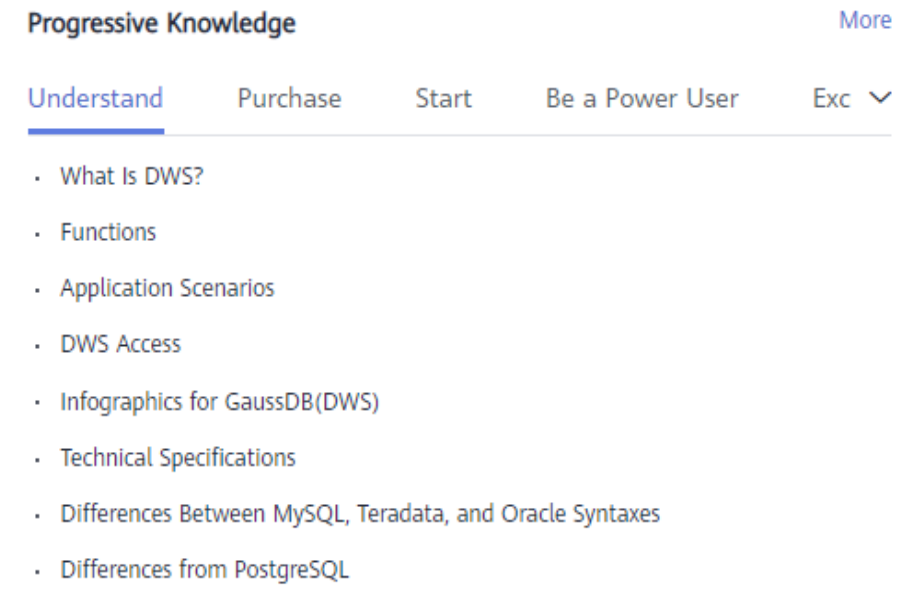
- Features

  This module describes multiple powerful functions of GaussDB(DWS), including SQL compatibility, cluster snapshot, cluster disaster recovery, database monitoring, resource management, and online O&M. You can quickly learn how they work and use them as required.

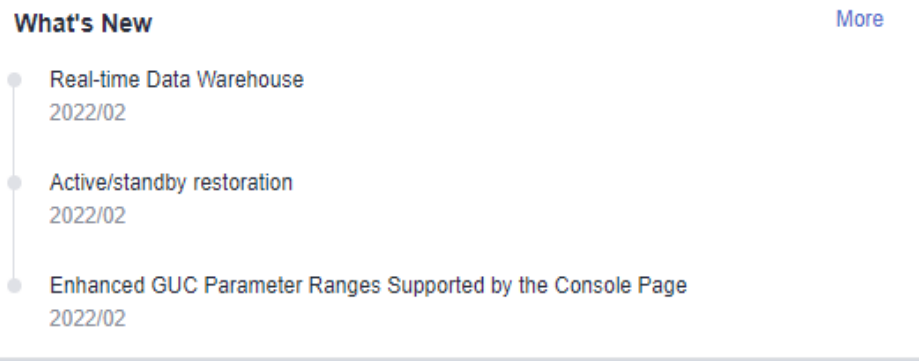
- Progressive Knowledge

  This module helps you start as a beginner and become an expert in the real-time, secure, reliable enterprise-grade GaussDB(DWS) data warehouse.



- What's New

  By default, the latest three features of GaussDB(DWS) are displayed. You can click **More** to check more information.

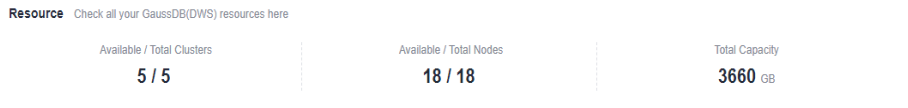


Dashboard With Cluster Overview

The dashboard page contains Progressive Knowledge, What's New, Features, and the following modules:

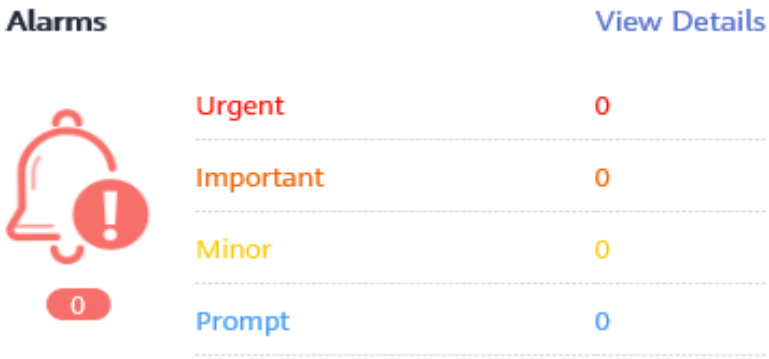
- Resources

  In the Resources module, you can view the number of available resources, including **Available/Total Clusters**, **Available/Total Nodes**, and **Total Capacity**.



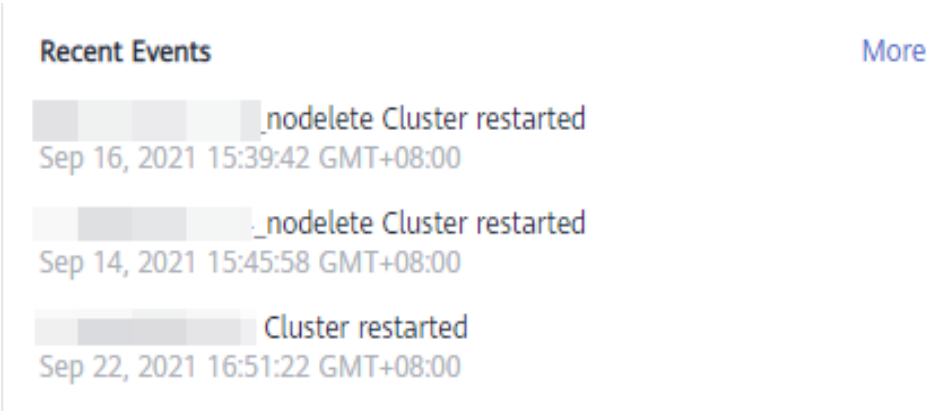
- Alarms

  Alarms are classified by severity: **Urgent**, **Important**, **Minor**, **Prompt**. For details, see **Alarms**.

- Recent Events

  Events are change records of user cluster status. Events can be triggered by user operations or cluster status changes. For details, see **Event Notifications**.

- Main cluster metrics:
  - **Cluster CPU Usage**
  - **Cluster Memory Usage**
  - **Cluster Disk Usage**

- For details about Progressive Knowledge, see **Progressive Knowledge**.
- For details about product changes, see **What' New**.

- For details about GaussDB(DWS) features, see **Features**.

5.2 Databases Monitoring (DMS)

5.2.1 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.
- The hybrid 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) 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, choose **Monitoring Panel**. The database monitoring page is displayed.

----End

5.2.2 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 5-1 describes GaussDB(DWS) monitoring metrics.

Table 5-1 GaussDB(DWS) monitoring metrics

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
Cluster Overview	Cluster Status	Status of a cluster.	Normal/ Abnormal / Degraded	30s
	Nodes	Number of available nodes and total number of nodes (Available/Total) in a cluster.	≥ 0	60s
	CNs	Number of CNs in a cluster.	≥ 0	60s
	Databases	Number of created databases in a cluster.	≥ 0	90s
Resource Consumption	CPU Usage	Average real-time CPU usage of all nodes in a cluster.	0% to 100%	30s
	Memory Usage	Average real-time memory usage of all nodes in a cluster.	0% to 100%	30s
	Disk Usage	Average real-time disk usage of all nodes in a cluster.	0% to 100%	30s
	Disk I/O	Average real-time disk I/O of all nodes in a cluster.	≥ 0 KB/s	30s
	Network I/O	Average real-time network I/O of all NICs in a cluster.	≥ 0 KB/s	30s
Top 5 Time-Consuming Queries	Query ID	ID of a query, which is automatically generated by the database.	≥ 0	180s
	SQL Statement	Query statement executed by a user.	String	180s
	Execution Time	Execution time of a query statement (unit: ms).	≥ 0 ms	180s
Top 5 Queries with Most Data Written to Disk	Query ID	ID of a query, which is automatically generated by the database.	≥ 0	180s
	SQL Statement	Query statement executed by a user.	String	180s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Data Written to Disk	Data to be written to disks after a user runs a statement (unit: MB).	≥ 0 MB	180s
Cluster Resource Metrics	CPU Usage	Average CPU usage of all nodes in a cluster.	0% to 100%	30s
	Memory Usage	Average memory usage of all nodes in a cluster.	0% to 100%	30s
	Disk Usage	Average usage of all disks in a cluster.	0% to 100%	30s
	Disk I/O Usage	Average I/O usage of all disks in a cluster.	0% to 100%	30s
	Network I/O Usage	Average I/O usage of all NICs in a cluster.	0% to 100%	30s
Key Database Metrics	Cluster Status	Cluster running status.	Normal/ Degraded / Abnormal	30s
	Cluster Abnormal CNs	Number of abnormal CNs in the cluster	≥ 0	60s
	Cluster Read-only	Whether the cluster is in the read-only state	Yes/No	30s
	Concurrent Sessions	Number of concurrent sessions in a cluster within a specified period.	≥ 0	30s
	Concurrent Queries	Number of concurrent queries in a cluster within a specified period.	≥ 0	30s
Node Monitoring-Overview	Node Name	Name of a node in a cluster.	String	30s
	CPU Usage	CPU usage of a host.	0% to 100%	30s
	Memory Usage	Memory usage of a host.	0% to 100%	30s
	Average Disk Usage (%)	Disk usage of a host.	0% to 100%	30s
	IP Address	Service IP address of a host.	String	30s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Disk I/O	Disk I/O of a host (unit: KB/s)	≥ 0 KB/s	30s
	TCP Protocol Stack Retransmission Rate	Retransmission rate of TCP packets per unit time.	0% to 100%	30s
	Status	Running status of a host	Online/ Offline	30s
Node Monitoring-Disks	Node Name	Name of a node in a cluster.	String	30s
	Disk Name	Name of a disk on a host.	String	30s
	Disk Capacity	Disk capacity of the host (unit: GB)	≥ 0 GB	30s
	Disk Usage	Disk usage of a host.	0% to 100%	30s
	Disk Read Rate	Disk read rate of the host (unit: KB/s)	≥ 0 KB/s	30s
	Disk Write Rate	Disk write rate of the host (unit: KB/s)	≥ 0 KB/s	30s
	I/O Wait Time (await, ms)	Average waiting time for each I/O request (unit: ms)	≥ 0 ms	30s
	I/O Service Time (svctm, ms)	Average processing time for each I/O request (unit: ms)	≥ 0 ms	30s
	I/O Utility (util, %)	Disk I/O usage of a host.	0% to 100%	30s
Node Monitoring-Network	Node Name	Name of a node in a cluster.	String	30s
	NIC Name	Name of the NIC on a host.	String	30s
	NIC Status	NIC status.	up/down	30s
	NIC Speed	Working rate of a NIC, in Mbit/s.	≥ 0	30s
	Received Packets	Number of received packets of a NIC.	≥ 0	30s
	Sent Packets	Number of sent packets of a NIC.	≥ 0	30s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Lost Packets Received	Number of received lost packets of a NIC.	≥ 0	30s
	Receive Rate	Number of bytes received by a NIC per unit of time (KB/s).	≥ 0 KB/s	30s
	Transmit Rate	Number of bytes sent by a NIC per unit of time (unit: KB/s)	≥ 0 KB/s	30s
Database Monitoring	Database Name	Name of the database created by a user in a cluster.	String	60s
	Usage	Used capacity of the current database (unit: GB).	≥ 0 GB	86400s
	Users	Number of users in the current database.	≥ 0	30s
	Sessions	Number of sessions in the current database.	≥ 0	30s
	Applications	Number of applications in the current database.	≥ 0	30s
	Queries	Number of active queries in the current database.	≥ 0	30s
	Scanning Rows	Number of rows returned by the full table scan query in the current database.	≥ 0	60s
	Index Query Rows	Number of rows returned by the index query in the current database.	≥ 0	60s
	Inserted Rows	Number of rows inserted in the current database.	≥ 0	60s
	Updated Rows	Number of rows updated in the current database.	≥ 0	60s
	Deleted Rows	Number of rows deleted from the current database.	≥ 0	60s
	Executed Transactions	Number of transaction executions on the current database.	≥ 0	60s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Transaction Rollbacks	Number of transactions in the current database that have been rolled back.	≥ 0	60s
	Deadlocks	Number of deadlocks detected in the current database.	≥ 0	60s
	Temporary Files	Number of temporary files created in the current database.	≥ 0	60s
	Temporary File Capacity	Size of temporary files written by the current database, in GB.	≥ 0	60s
Performance Monitoring	Cluster CPU Usage	Average CPU usage of all nodes in a cluster.	0% to 100%	30s
	Cluster Memory Usage	Average memory usage of all nodes in a cluster.	0% to 100%	30s
	Cluster Disk Usage	Average disk usage of all nodes in a cluster.	0% to 100%	30s
	Cluster Disk I/O	Average I/O of all disks in a cluster.	0% to 100%	30s
	Cluster Network I/O	Average I/O of all NICs in a cluster.	0% to 100%	30s
	Cluster Status	Historical trend of the cluster status.	Normal/ Abnormal / Degraded	30s
	Cluster Read-only	Historical trend of the cluster read-only status change trend.	Yes/No	30s
	Cluster Abnormal CNs	Historical trend of the number of abnormal CNs in the cluster.	≥ 0	60s
	Cluster Abnormal DNs	Historical trend of the number of abnormal DNs in the cluster.	≥ 0	60s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Cluster CPU Usage of DNs	Average CPU usage of all DNs in a cluster.	0% to 100%	60s
	Cluster Sessions	Historical trend of the number of sessions in a cluster.	≥ 0	30s
	Cluster Queries	Historical change trend of the number of queries in the cluster.	≥ 0	30s
	Cluster Deadlocks	Historical trend of the number of deadlocks in a cluster.	≥ 0	60s
	Cluster TPS	Average number of transactions per second of all databases in a cluster. Formula: (delta_xact_commit + delta_xact_rollback)/ current_collect_rate	≥0	60s
	Cluster QPS	Average number of concurrent requests per second of all databases in a cluster. Formula: delta_query_count/ current_collect_rate	≥ 0	60s
	Database Sessions	Historical trend of the number of sessions on a single database in a cluster.	≥ 0	30s
	Database Queries	Historical trend of the number of queries on a single database in a cluster.	≥ 0	30s
	Database Inserted Rows	Historical trend of the number of rows inserted into a single database in a cluster.	≥ 0	60s
	Database Updated Rows	Historical trend of the number of updated rows in a single database in a cluster.	≥ 0	60s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Database Deleted Rows	Historical trend of the number of deleted rows in a single database in a cluster.	≥ 0	60s
	Database Capacity	Historical trend of the capacity in a single database in a cluster.	≥ 0	86400s
Live Session	Session ID	ID of the current session (query thread ID).	String	30s
	User Name	Name of the user who executes the current session.	String	30s
	Database Name	Name of the database connected to the current session.	String	30s
	Session Duration	Duration of the current session (unit: ms).	≥ 0 ms	30s
	Application Name	Name of the application that creates the current session.	String	30s
	Queries	Number of SQL statements executed in the current session.	≥ 0	30s
	Latest Query Duration	Duration for executing the previous SQL statement in the current session.	≥ 0 ms	30s
	Client IP Address	IP address of the client that initiates the current session.	String	30s
	Connected CN	Connected CN of the current session.	String	30s
	Session Status	Execution status of the current session.	Running/ Idle/Retry	30s
Real-Time Query	Query ID	Query ID of a current query statement, which is a unique identifier allocated by the kernel to each query statement.	String	30s
	User Name	Name of the user who submits the current query statement.	String	30s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Database Name	Name of the database corresponding to the current query statement.	String	30s
	Application Name	Name of the application corresponding to the current query statement.	String	30s
	Resource Pool	Name of the resource pool for the current query statement.	String	30s
	Submitted	Timestamp when the current query statement is submitted.	String	30s
	Blocking Time	Waiting time before the current query statement is executed, in ms.	≥ 0	30s
	Execution Time	Execution time of the current query statement, in ms.	≥ 0	30s
	CPU Time	Total CPU time spent by the current query statement on all DNs, in ms.	≥ 0	30s
	CPU Time Skew	CPU time skew of the current query statement among all DNs.	0% to 100%	30s
	Statement	Query statement that is being executed.	String	30s
	Connected CN	Name of the CN that submits the current query statement.	String	30s
	Client IP Address	IP address of the client that submits the current query statement.	String	30s
	Lane	Lane where the current query statement is located.	Fast Lane/ Slow Lane	30s
	Query Status	Query status of the statement that is being executed.	String	30s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Session ID	Session ID of the current query statement, which is a unique identifier allocated by the kernel to each client connection.	String	30s
	Queuing Status	Status of the current query execution in the database, indicating whether the query is queued in the resource pool.	Yes/No	30s
Historical Query	Query ID	Query ID of a query statement, which is a unique identifier allocated by the kernel to each query statement.	String	180s
	User Name	Name of the user who submits a query statement.	String	180s
	Application Name	Application name corresponding to a query statement.	String	180s
	Database Name	Name of the database corresponding to a query statement.	String	180s
	Resource Pool	Name of the resource pool for the current query statement.	String	180s
	Submitted	Timestamp when a query statement is submitted.	String	180s
	Blocking Time	Waiting time before the query statement is executed, in ms.	≥ 0	180s
	Execution Time	Execution time of the query statement, in ms.	≥ 0	180s
	CPU Time	Total CPU time spent by the query statement on all DNs, in ms.	≥ 0	180s
	CPU Time Skew	CPU time skew of a query statement executed on all DNs.	0% to 100%	180s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Statement	Query statements to be parsed	String	180s
Slow Instance Monitoring	Slow Instance	Number of slow instances detected at the current time point.	≥ 0	240s
	Detected	Time when a slow instance is detected for the first time.	String	240s
	Node Name	Name of the node where the slow instance is deployed.	String	240s
	Instance	Name of an instance.	String	240s
	Slow Node Detections (within 24 hours)	Number of times that a slow instance is detected within 24 hours.	≥ 0	240s
Resource Pool Monitoring	Resource Pool	Name of a resource pool in a cluster.	String	120s
	CPU Usage	Real-time CPU usage of a resource pool.	0% to 100%	120s
	CPU Resource	CPU usage quota of a resource pool.	0% to 100%	120s
	Real-Time Concurrent Short Queries	Simple concurrency in a resource pool.	≥ 0	120s
	Concurrent Short Queries	Quota for simple concurrency in a resource pool.	≥ 0	120s
	Real-Time Concurrent Queries	Real-time complex concurrency in a resource pool.	≥ 0	120s
	Query Concurrency	Quota for complex concurrency in a resource pool.	≥ 0	120s
	Storage	Storage resource quota of a resource pool.	≥ 0	120s
	Disk Usage	Disk usage of a resource pool.	0% to 100%	120s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Memory	Memory quota of a resource pool.	≥ 0	120s
	Memory Usage	Memory usage of a resource pool.	0% to 100%	120s
Queries Waiting in a Resource Pool	User	Name of the user of waiting queries	String	120s
	Application	Name of the application to be queried.	String	120s
	Database	Name of the database to be queried.	String	120s
	Queuing Status	Execution status of a query in the database (CCN/CN/DN).	String	120s
	Wait Time	Waiting time for a waiting query (unit: ms).	≥ 0 ms	120s
	Resource Pool	Resource pool of the waiting query.	String	120s
	Statement	Query statement for the waiting status.	String	120s
Circuit Breaking Queries	Query ID	Query ID of the circuit breaking query statement.	String	120s
	Query Statement	Query statement for the circuit breaking status.	String	120s
	Blocking Time	Blocking time before the query statement triggers circuit breaking, in ms.	≥ 0	120s
	Execution Time	Execution time before the query statement triggers circuit breaking, in ms.	≥ 0	120s
	CPU Time	Average CPU time consumed by each DN before the query statement triggers circuit breaking, in ms.	≥ 0	120s
	CPU Skew	Skew rate of CPU time consumed by each DN before the query statement triggers circuit breaking.	0% to 100%	120s

Monitored Object	Metric	Description	Value Range	Monitoring Period (Raw Data)
	Exception Handling	Handling method after the query statement triggers circuit breaking.	Abort/ Degrade	120s
	Status	Circuit breaking handling status of a query statement.	Executing / Completed	120s
SQL Tuning	Query ID	IP address of the current query (query logic ID).	String	180s
	Database	Name of the database where the current query is executed.	String	180s
	Schema Name	Name of the current query schema.	String	180s
	User Name	Name of the user who performs the query.	String	180s
	Client	Name of the client that initiates the current query.	String	180s
	Client IP Address	IP address of the client that initiates the current query.	String	180s
	Running Time	Execution time of the current query, in ms.	≥ 0	180s
	CPU Time	CPU time of the current query, in ms.	≥ 0	180s
	Scale-Out Started	Start time of the current query.	Timestamp	180s
	Completed	End time of the current query.	Timestamp	180s
	Details	Details about the current query.	String	180s
INODE	Inode Usage	Disk inode usage.	0% to 100%	30s
SCHEMA	Schema Usage	Database schema usage.	0% to 100%	3600s

5.2.3 Cluster Overview

Cluster Overview

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, click **Cluster Overview**.

On the page that is displayed, you can view the cluster status, real-time resource consumption, top SQL statements, 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.

NOTE

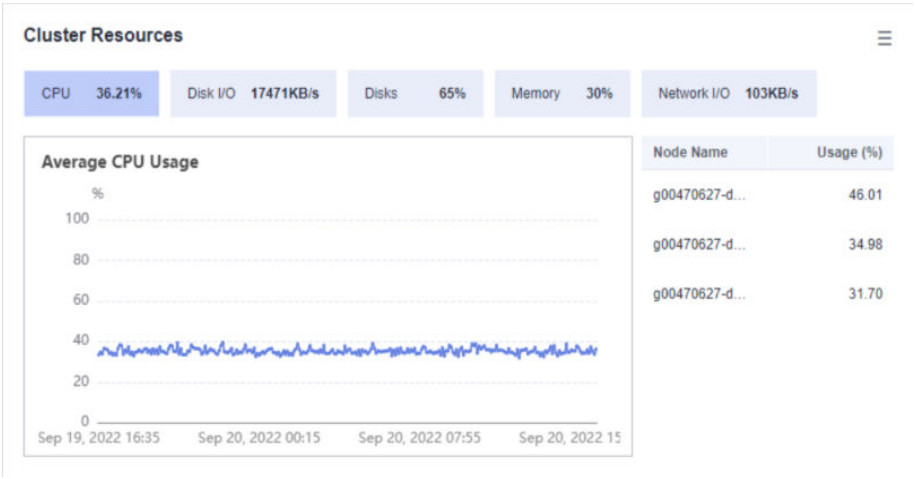
The OBS usage details are displayed in GaussDB(DWS) 3.0 cluster information.

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 **Alarms**.

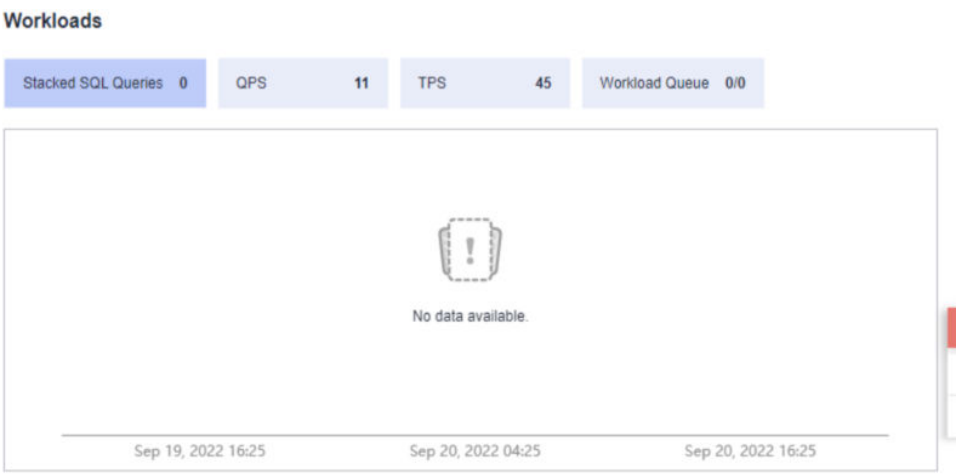
Cluster Resources

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**.



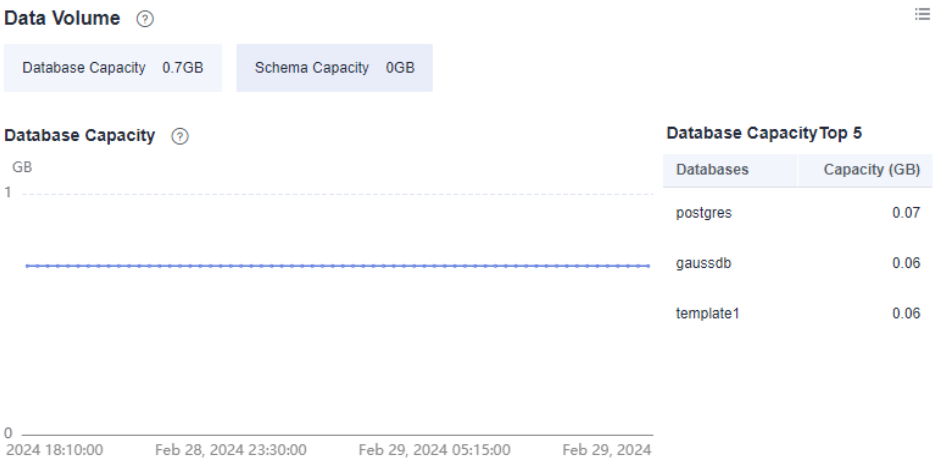
Workloads

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.



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**.

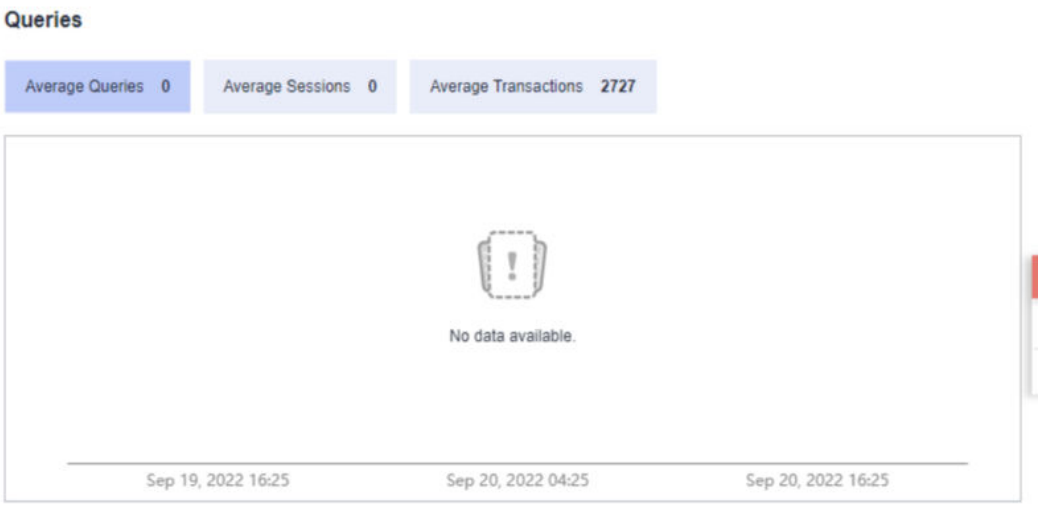


NOTE

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.



5.2.4 Monitoring

5.2.4.1 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.

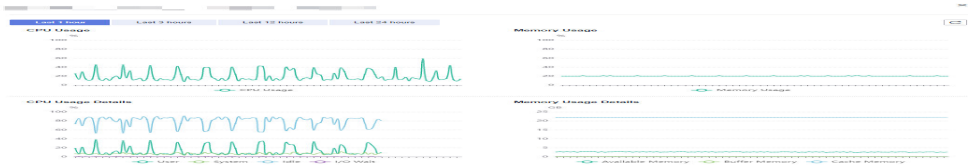
----End

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.

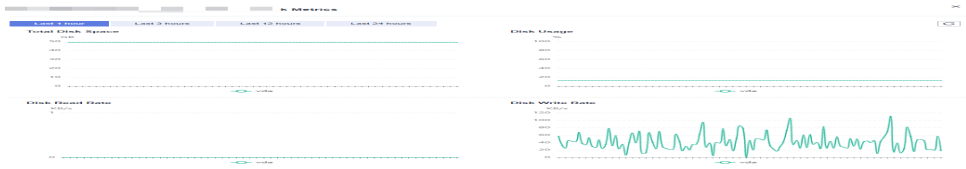


Disks

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]#
```

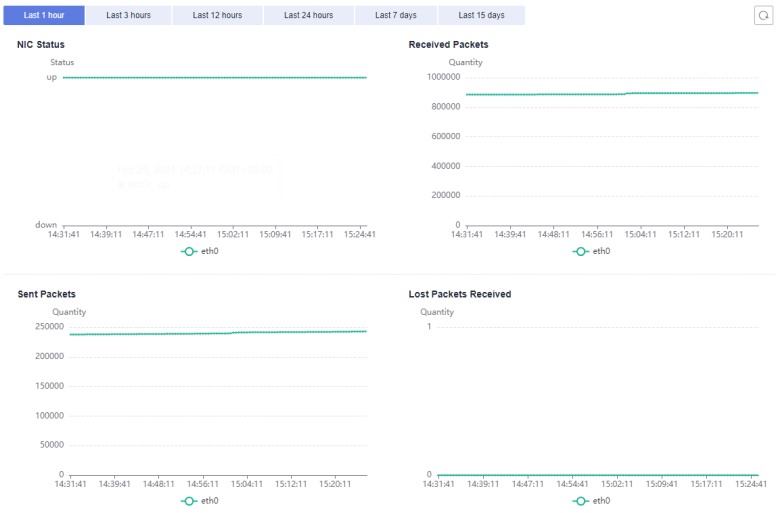
/dev/sda4: Used(5757444) + Available(540228616) != Total(569616888)

- **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.



5.2.4.2 Performance Monitoring

Performance Monitoring

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

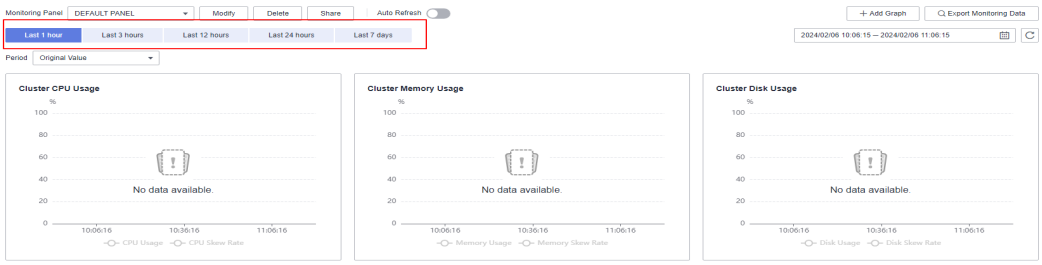
Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **Monitoring** > **Performance Monitoring**. The **Performance Monitoring** page displays the resource consumption trends of clusters, databases, and nodes.

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.



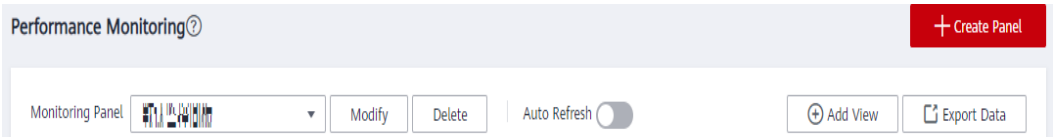
----End

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.

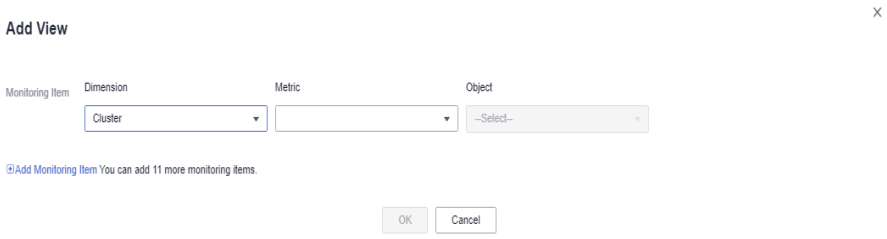
- Modifying a monitoring panel: You can click **Modify** to change the name of a monitoring panel.
- Deleting a monitoring panel: You can click **Delete** to delete a monitoring panel. The default monitoring panel cannot be deleted.
- Sharing a monitoring panel: You can click **Share** to share a monitoring panel. The recipients can view the panel but cannot modify it.



Adding a Monitoring View

Currently, DMS provides monitoring views for clusters, databases, and nodes. You can click **Add View** to add a monitoring view as required. The monitoring metrics are as follows:

- Cluster: CPU Usage, Memory Usage, Disk Usage, Disk I/O, Network I/O, Status, Abnormal CNs, Read-only, Sessions, Queries, Deadlocks, Abnormal DNs, CPU Usage of DNs, TPS, and QPS
- Database metrics: query waiting queue length, number of sessions, number of queries, number of inserted rows, number of updated rows, number of deleted rows, and capacity.
- Node: CPU usage, memory usage, average disk usage, disk I/O, TCP retransmission rate, network I/O, total disk space, disk usage, disk read rate, disk write rate, disk I/O wait time, disk I/O service time, disk I/O usage, NIC status, number of received packets, number of sent packets, number of lost packets received, receive rate, and transmit rate.



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.

5.2.4.3 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.

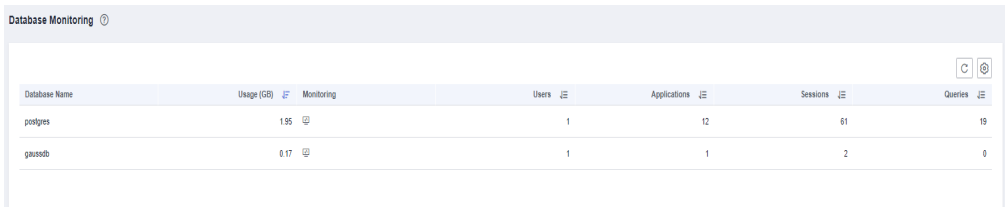
----End

Database Resource Consumption

You can select a database to view its resource usage. For details about the metrics, see **Monitoring Metrics**. including:

- Database Name
- Usage (GB)
- Monitoring
- Users
- Applications
- Sessions
- Queries
- Inserted Rows
- Updated Rows
- Deleted Rows
- Deadlocks
- Temporary Files

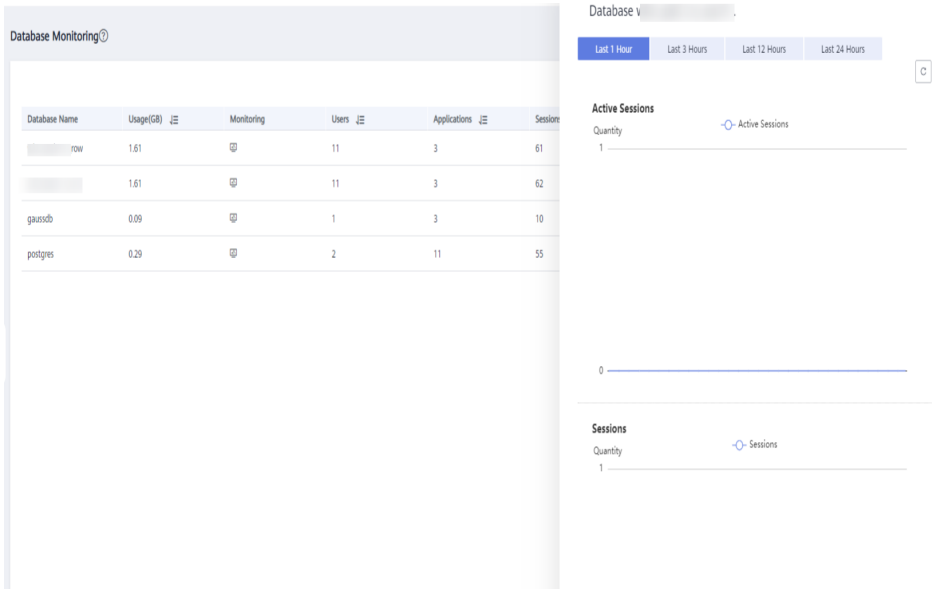
- Temporary File Capacity



Database Trend Monitoring

In the **Monitoring** column of a database, click to view the performance indicators of the database, including:

- Capacity
- Sessions
- Queries



5.2.4.4 Real-Time Queries

Going to the Real-time Query Page

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, 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 the **Monitoring Collection** tab, 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

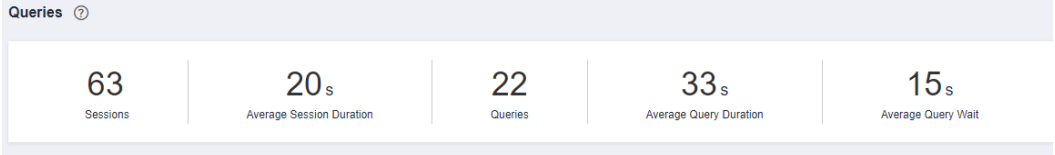
You need to set GUC parameters before viewing data on the monitoring page. If GUC parameters are not set, real-time or historical query may be unavailable. However, if this parameter is set, the cluster performance may deteriorate. Therefore, you need to balance the settings of related parameters. The following table describes recommended settings. For details about how to modify parameters, see **Modifying Database Parameters**. **Setting GUC Parameters** provides parameter details.

Table 5-2 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	1000MB	1000MB

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.



Checking Live Sessions

On the **Sessions** tab, you can browse the real-time information about all running queries,

- Session ID
- Username
- Session duration
- Application name
- QueryBand
- Client IP address
- Connected CN
- Session status. It can be:
  - **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.
- Start time
- Lock mode
- Lock holding status
- Locked object
- Query SQL
- Lock wait
- Current query duration
- Current query start time

  NOTE

  - You can click a session ID to view the queries in the current session. For details, see **Viewing Historical 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

On the **Queries** tab, you can browse all the queries that are running in a specified time period, including:

- Query ID
- Username
- Application name
- Database name
- Resource pool
- Submission time

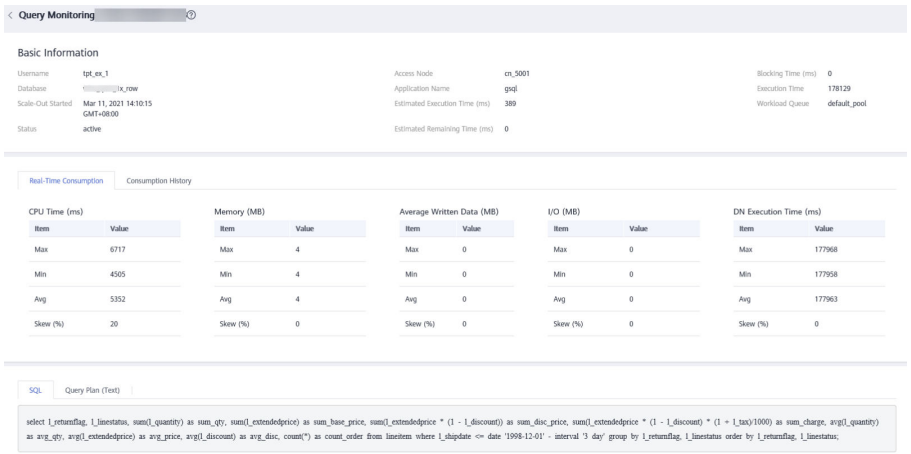
- Blocking time (ms)
- Execution time (ms)
- Statement
- Connected CN
- Client IP address
- Lane
- Query status. It can be:
  - **idle**: The backend is waiting for new client commands.
  - **active**: The backend is executing queries.
  - **idle in transaction**: The backend is in a transaction, but 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.
- Session ID
- Statement status

  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.



5.2.4.5 Historical Queries

Going to the Historical Query Page

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** > **History**.

All the historical queries in the current cluster will be displayed.

----End

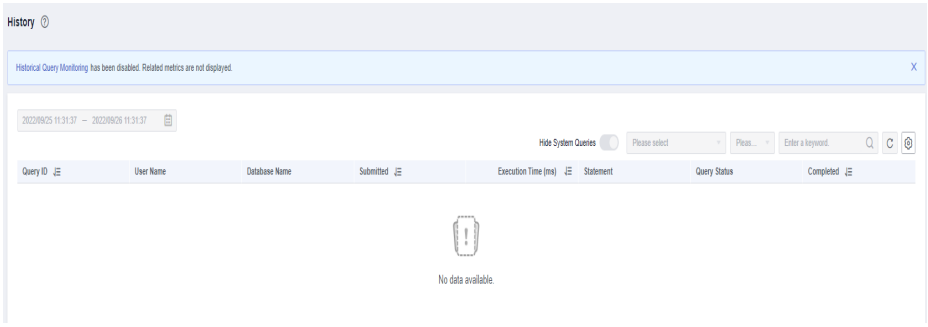
NOTE

- Historical queries can be viewed only in clusters of version 8.1.2 and later.
- To enable historical query monitoring, choose **Settings** > **Monitoring**, click the **Monitoring Collection** tab, 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, including:

- Query ID
- Username
- Application name
- Database name
- Resource pool
- Submission time
- Blocking time (ms)
- Execution time (ms)
- CPU time (ms)
- CPU time skew (%)
- Statement
- Connected CN
- Client IP address
- Query status
- Completion time
- Estimated execution time (ms)
- Cancellation reason

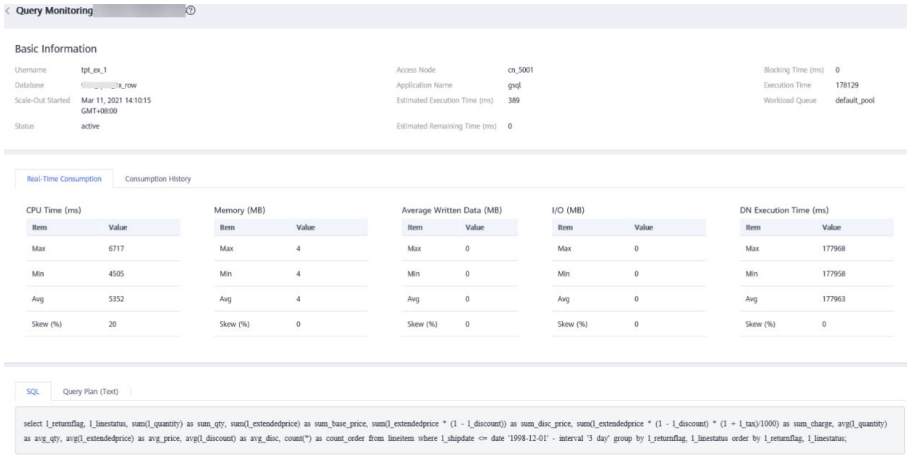


NOTE

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, including the basic information of query statements, real-time and historical resource consumption, SQL description, and query plan.



5.2.4.6 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**.

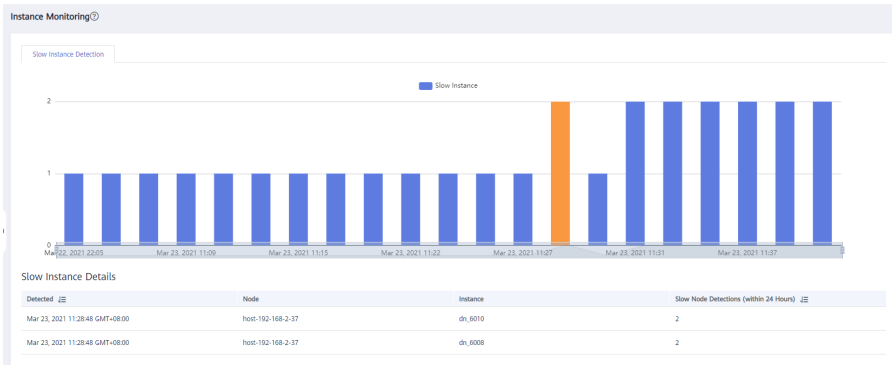
On the **Instance Monitoring** page, you can view the real-time and historical information about detected slow instances.

----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.



NOTE

If the period of an instance exceeds 240 seconds, the instance is reported as a slow instance.

5.2.4.7 Resource Pool Monitoring

Accessing the Resource Monitoring Page

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** > **Resource Pool Monitoring**.

You can check the real-time statistics and resource consumption history about resource pools.

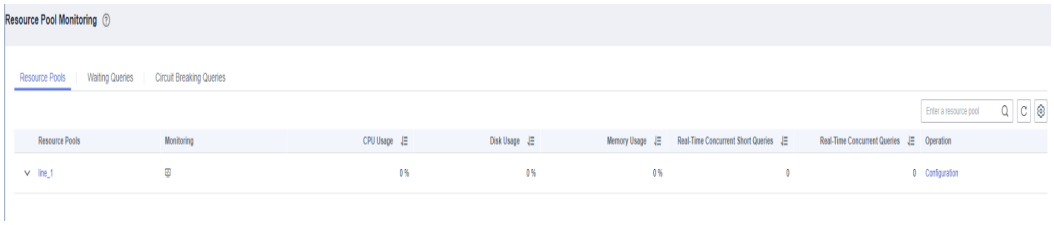
----End

Resource Pool

You can check user-defined resource pools, real-time and historical resource consumption, and the resource quotas of resource pools.

- **Resource Pool**: Resource pool name.

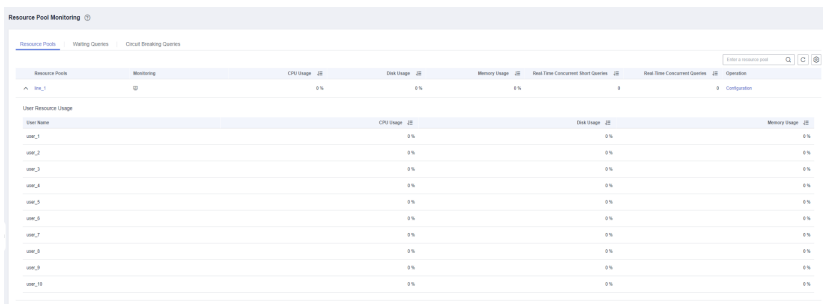
- **Monitoring**: You can click the monitoring icon to display the historical consumption trends of resources such as the CPU, memory, and disk.
- **CPU Usage (%)**: real-time CPU usage of a resource pool
- **CPU Share (%)**: CPU usage share of a resource pool
- **Real-Time Concurrent Short Queries**: number of concurrent simple queries in a resource pool. Concurrent simple queries are not controlled by the resource pool.
- **Concurrent Short Queries**: 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.
- **Query Concurrency**: quota of complex concurrent queries in a resource pool
- **Storage (MB)**: 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
- **Operation**



User Resource Usage

Click the arrow next to a resource pool name to expand resource usage details.

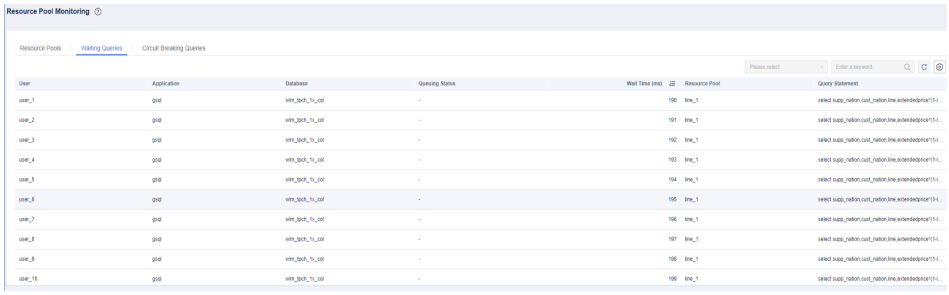
- **User Name**: name of a user in the current resource pool
- **CPU Usage** (%): real-time CPU usage of a user
- **CPU Resource**: number of CPU cores used
- **Storage Resource** (MB): storage space used by a user
- **Disk Usage** (%): disks used by a user
- **Memory Resource** (MB): memory used by a user
- **Memory Usage** (%): percentage of memory used by a user



Queries Waiting in a Resource Pool

You can view the queries waiting in a resource pool in real time to check workload status.

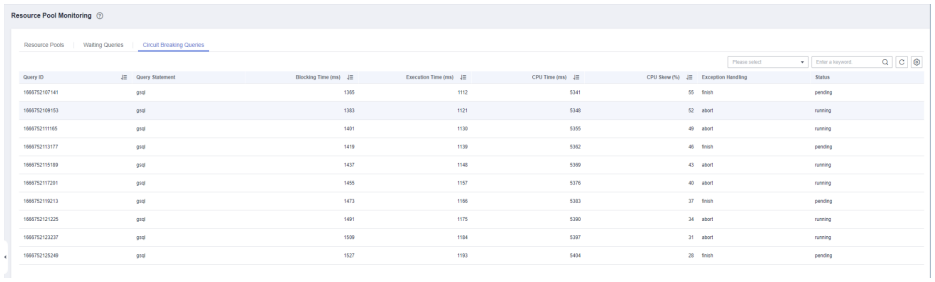
- **User**: user name of a query statement
- **Application**: application name of a query statement
- **Database**: name of the database to which a query statement is connected
- **Queuing Status**: queuing status of a query statement in a resource pool
- **Wait Time**: waiting time before a query statement is executed, in ms
- **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, in ms
- **Execution Time (ms)**: execution time of a circuit breaking statement, in ms
- **CPU Time (ms)**: CPU time consumed by a circuit breaking statement, in ms
- **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



5.2.5 Utilities

5.2.5.1 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) management console.

Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

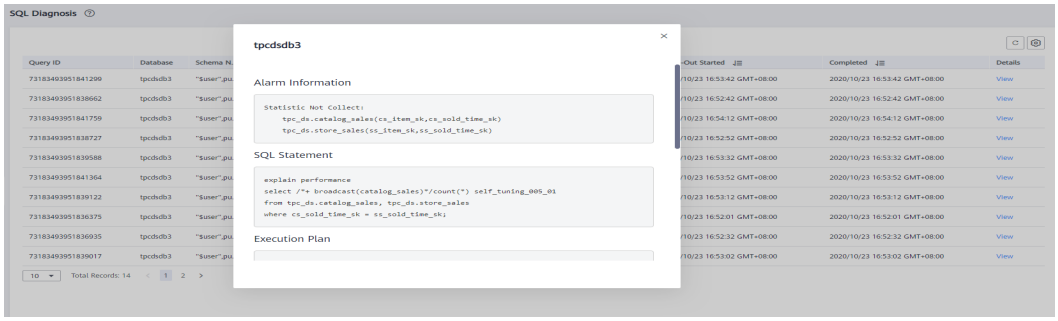
Step 3 In the **Operation** column of the target cluster, click **Monitoring Panel**.

Step 4 In the navigation pane on the left, choose **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:

- Diagnosis Type
- Alarm Information
- SQL Statement
- Execution Plan



----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: **100000**
  - 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: **off**
  - 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.

5.2.5.2 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 following functions are supported:

- **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.

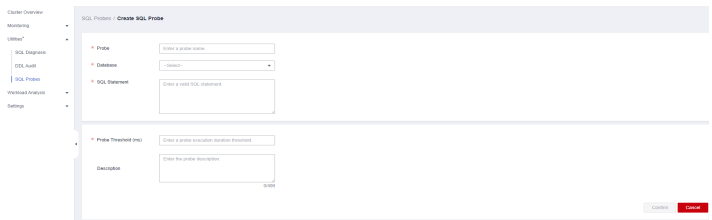
Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

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.



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.

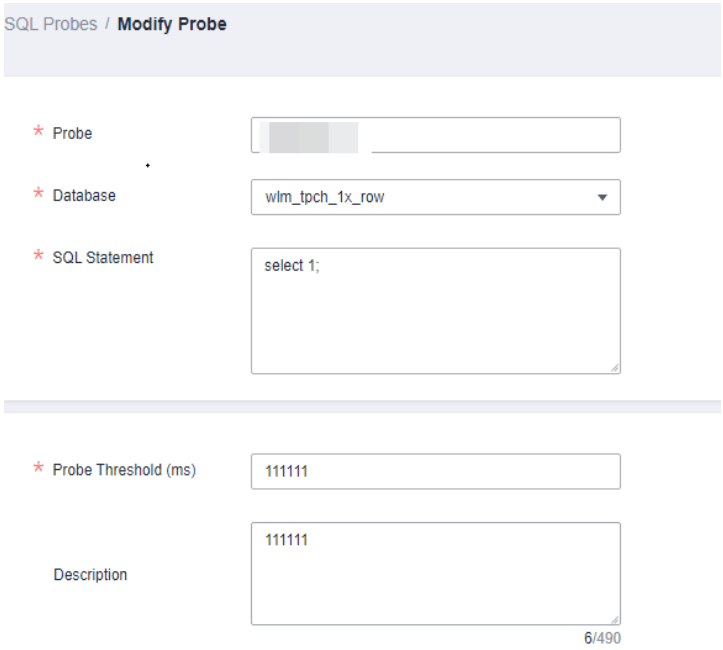
Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to be monitored.

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

5.2.5.3 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 data collection period of the table skew and dirty page checks can be configured on the **Monitoring Collection** page. Frequent data 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?**.

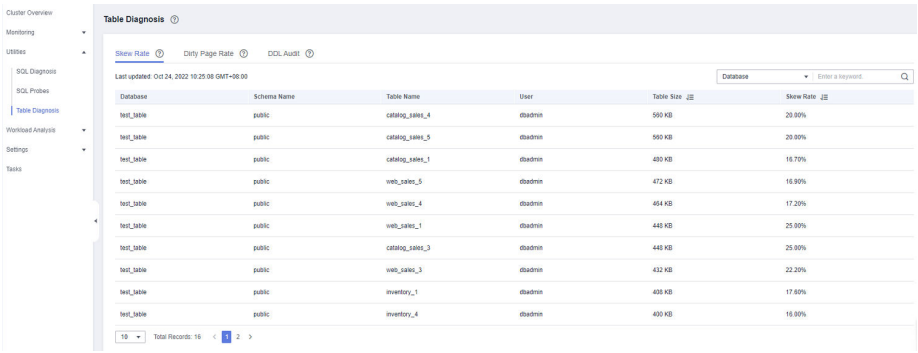
Procedure

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

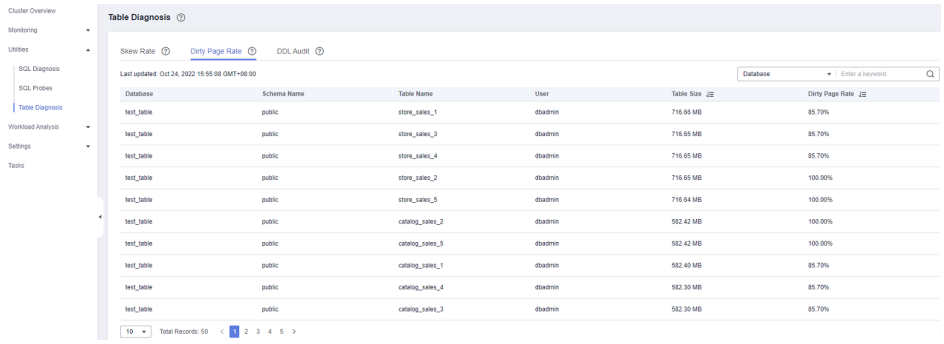
Procedure

Step 1 Log in to the GaussDB(DWS) 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 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

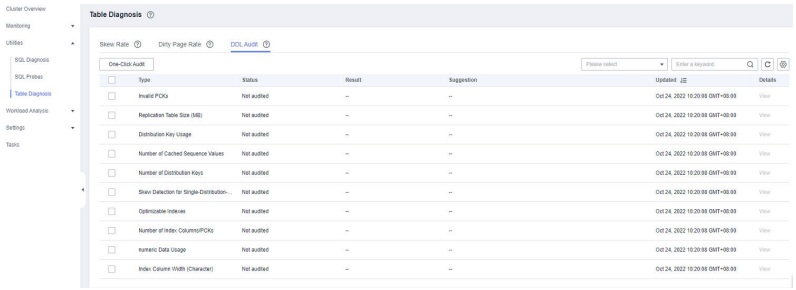
Viewing and Exporting DDL Audit Results

Step 1 Log in to the GaussDB(DWS) 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 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 5-3**.

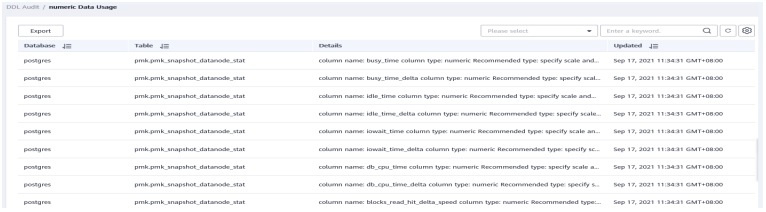
Table 5-3 Audit items

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.

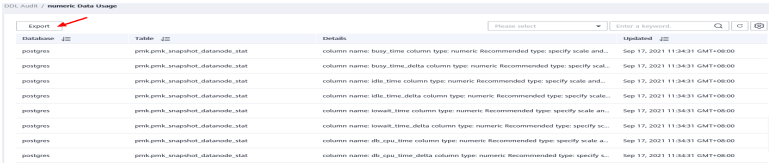
Item	Description
Invalid PCKs (invalidPck)	Do not create invalid PCK columns. In 8.1.1 and later versions, the cluster can filter and compare data of the char, int8, int2, int4, text, bpchar, varchar, date, time, timestamp and timestamptz types. If a column of an unsupported data type is used as a PCK, 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

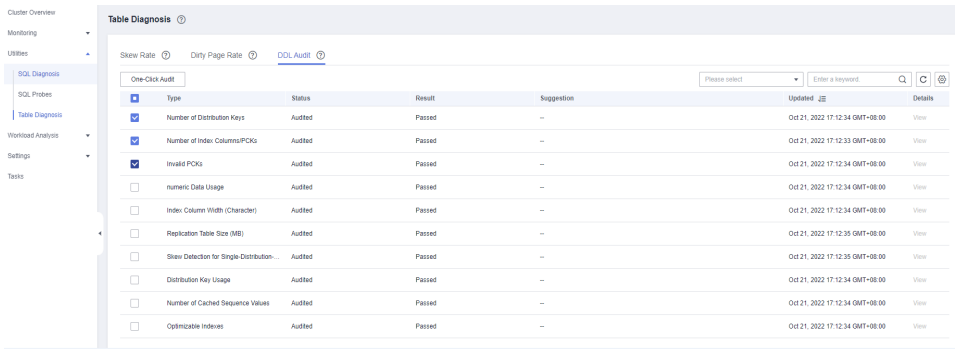
Manually Auditing DDL Items

Step 1 Log in to the GaussDB(DWS) 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, 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

5.2.6 Workload Analysis

5.2.6.1 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.

5.2.6.2 Workload Snapshots

You can check the basic information about the cluster workload snapshots, manually create a snapshot, and configure snapshot parameters.

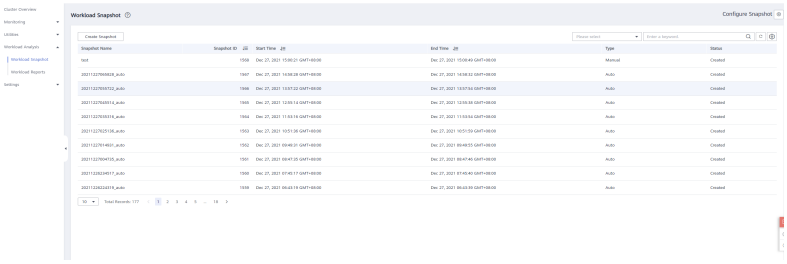
Checking Workload Snapshots

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

Step 2 On the **Cluster** > **Dedicated Cluster** 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

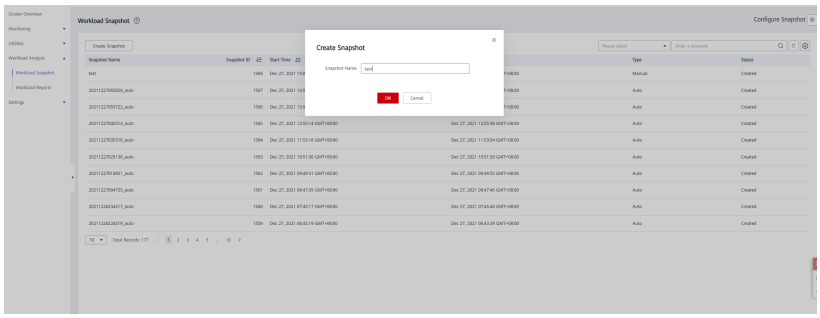
Step 1 Log in to the GaussDB(DWS) management console.

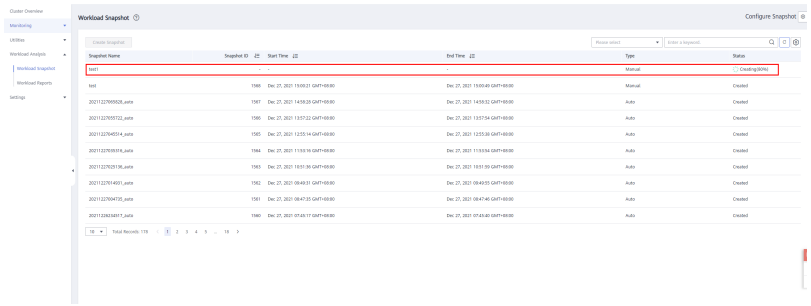
Step 2 On the **Cluster** > **Dedicated Cluster** page, locate the cluster for which you want to perform workload analysis.

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

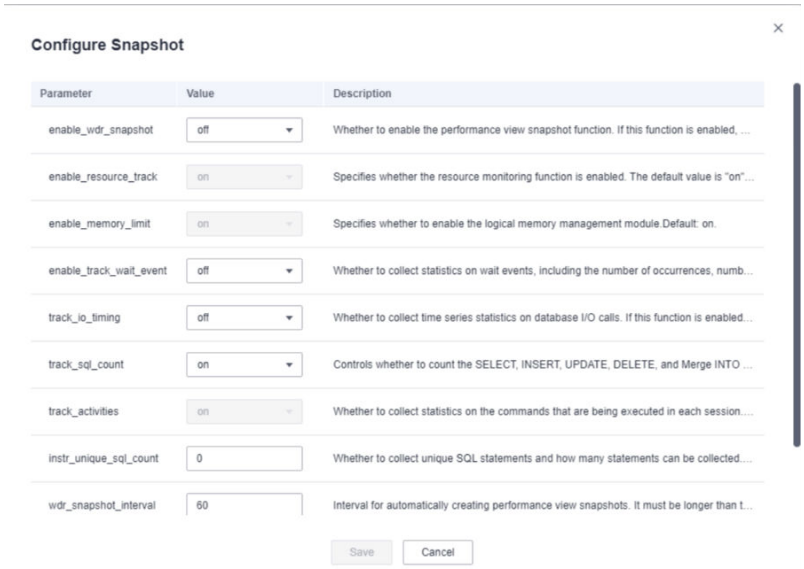
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Cluster** > **Dedicated Cluster** 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 5-4**.



----End

Table 5-4 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.

5.2.6.3 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 **Cluster** > **Dedicated Cluster** 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 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 **Cluster** > **Dedicated Cluster** page, locate the cluster for which you want to perform workload analysis.

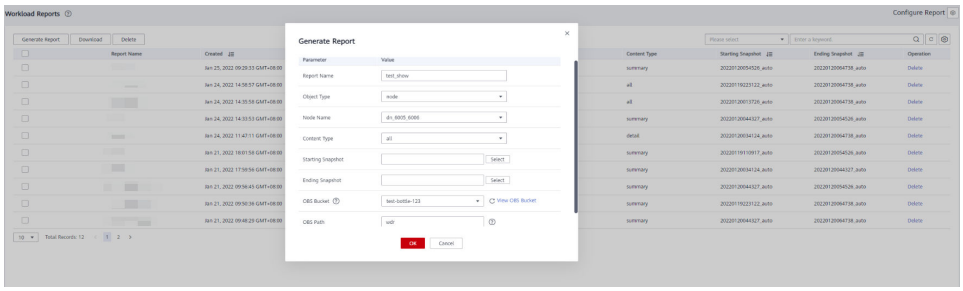
Step 3 In the **Operation** column of the cluster, choose **Monitoring Panel**. The database monitoring page is displayed.

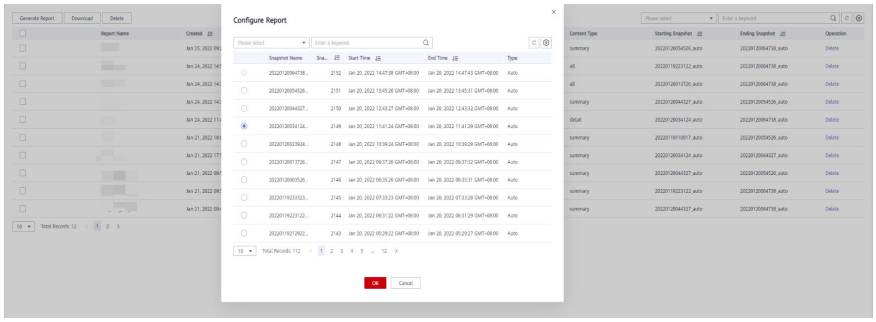
Step 4 In the navigation pane, choose **Workload Analysis** > **Workload Reports**.

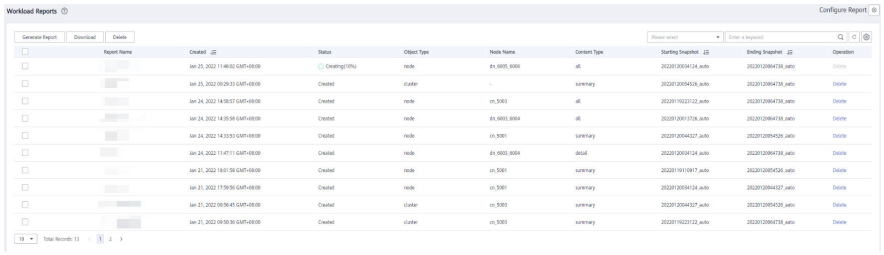
Step 5 Click **Generate Report**. In the displayed dialog box, configure the following parameters and click **OK**:

- **Report Name**: Enter a unique report name. The name can contain a maximum of 100 characters, including digits, letters, and underscores (_).)
- **Object Type**. Its value can be:
  - **node**: The performance data of a specified node will be provided.
  - **cluster**: The performance data of the entire cluster will be provided.
- **Node Name**: Select a node.
- **Content Type**. Its value can be:
  - **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.
- **Starting Snapshot**: Select a snapshot.

- **Ending Snapshot**: Select a snapshot.
- **OBS Bucket**: Select a bucket to store reports.
- **OBS Path**: A storage directory. Multiple levels of directories can be separated by slashes (/). The value cannot start with a slash (/). Up to 50 characters are allowed.







NOTE

The time of the starting snapshot start must be earlier than that of the ending snapshot.

----End

Downloading Workload Reports in Batches

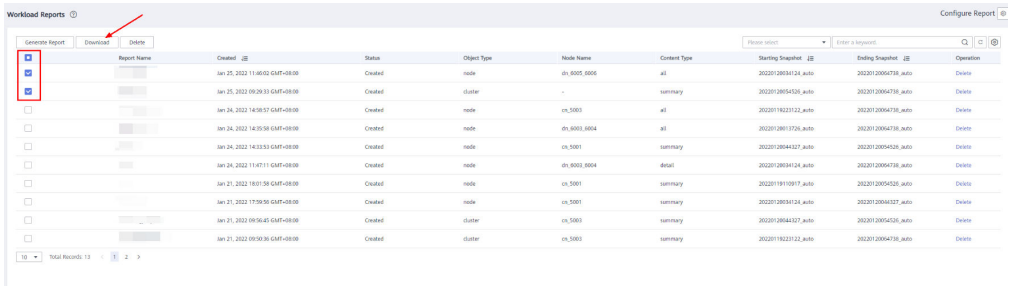
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Cluster** > **Dedicated Cluster** 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 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

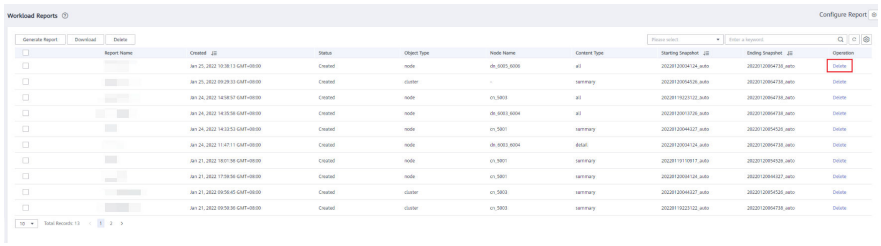
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Cluster** > **Dedicated Cluster** 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 Reports**.

Step 5 Select reports and click **Delete**.



----End

Deleting a Workload Diagnosis Report

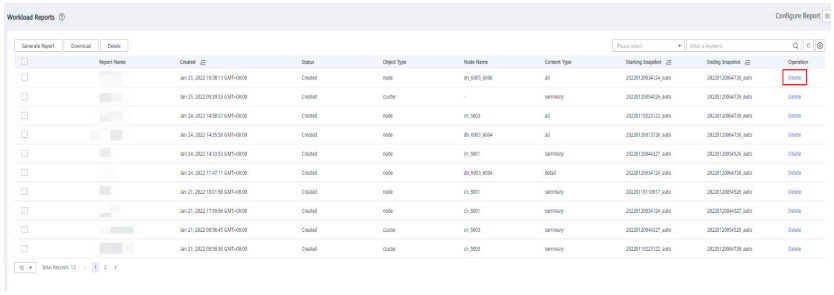
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Cluster** > **Dedicated Cluster** 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 Reports**.

Step 5 Click **Delete** in the **Operation** column of a report to delete the report record and file.



----End

Configuring Workload Report Parameters

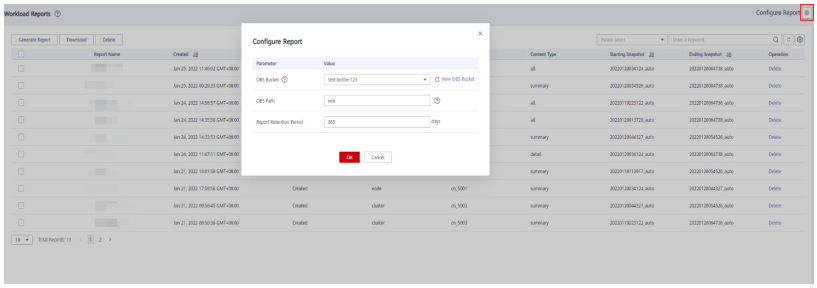
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 On the **Cluster** > **Dedicated Cluster** 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 Reports**.

Step 5 Click **Configure Report** in the upper right corner. In the displayed dialog box, set the report retention period and OBS parameters.



----End

5.2.7 Settings

The **Monitoring** page displays the collection period and data aging 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.

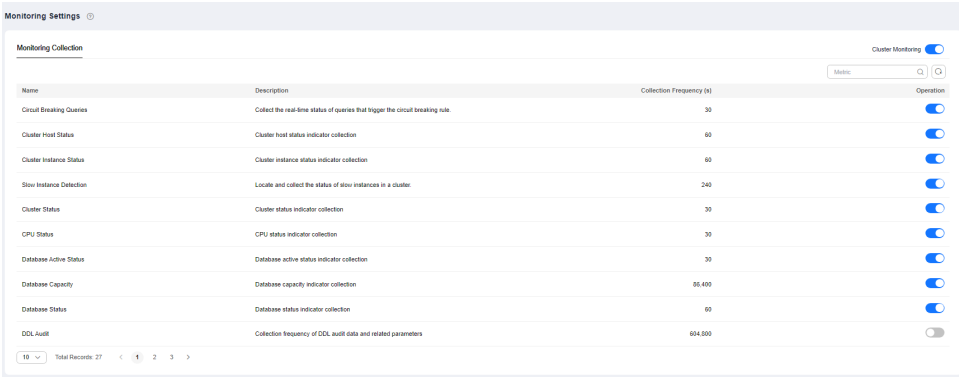
Monitoring Collection

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, choose **Monitoring Panel**. The database monitoring page is displayed.

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.



----End

5.2.8 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.

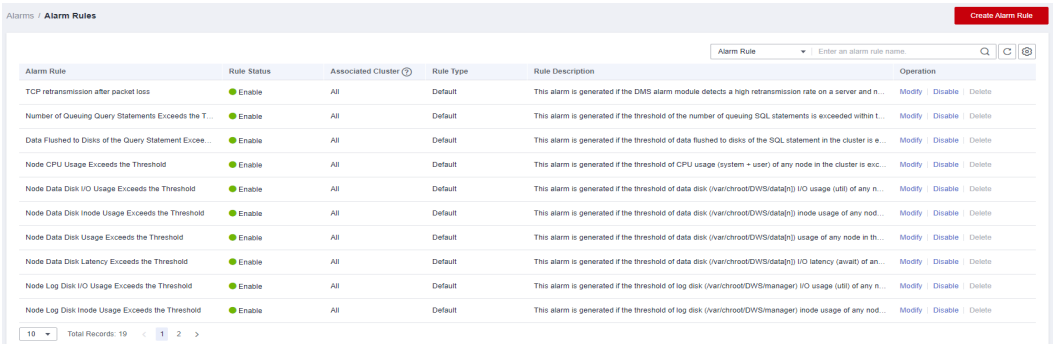
Procedure

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



----End

5.2.9 Typical Scenarios

5.2.9.1 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 **Details** to view SQL optimization suggestions.

Step 4 Optimize the SQL statement based on suggestions.

----End

5.2.9.2 Top Time-Consuming SQL Statements Viewing

Symptom

Time-consuming SQL statements exist.

Troubleshooting Process

On the **Top 5 Time-Consuming Queries** page directed from the **Cluster Overview** page, record the change of top 5 time-consuming queries.

Analyze the frequency of top 5 queries to locate slow queries.

Troubleshooting Procedure

Step 1 On the **Cluster Overview** page, click and view the **Top5 Time-Consuming Queries** page.

Step 2 Find the IDs of time-consuming queries and query the pid field (session_id) in the database view **PGXC_WLM_SESSION_STATISTICS**.

Step 3 On the **Session Monitoring** page, locate the session_id and kill the time-consuming SQL statement.

----End

5.3 Monitoring Clusters Using 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 5-5 describes GaussDB(DWS) monitoring metrics.

Table 5-5 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. It the ratio of the amount of data that already exists in the cache to the total amount of requested data. 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	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
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

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
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
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

Metric ID	Name	Description	Value Range	Monitored Object	Monitoring Period (Raw Data)
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) management 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 left navigation pane of the Cloud Eye management console, choose **Dashboard** > **Panels**.

Step 2 On the page that is displayed, click **Create Panel**. In the displayed dialog box, enter the name and click **OK**.

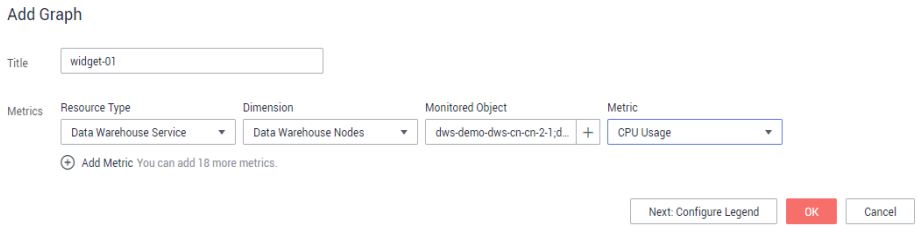
Step 3 Click **Add Graph** in the upper right corner.

Step 4 In the displayed dialog box, configure the title and monitoring metrics.

NOTE

You can add multiple monitoring metrics by clicking **Add Metric**.

Figure 5-1 Adding a graph



The following describes how to set parameters if you want to compare CPU usage of two nodes.

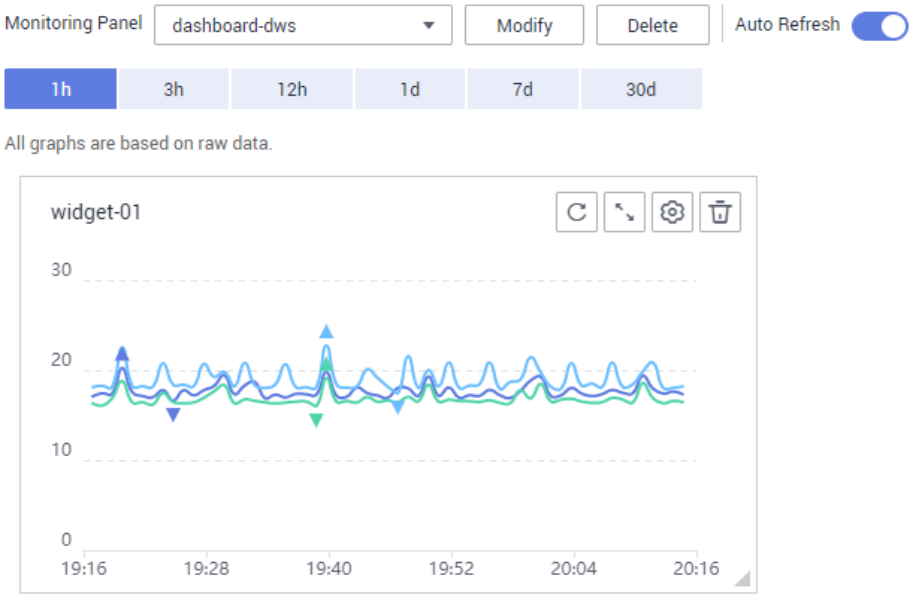
Table 5-6 Configuration example

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 5 Click **OK**.

Then you can view the corresponding monitoring graph on the **Panels** page. Move the cursor to the graph and click in the upper right corner to zoom in the graph and view detailed metric comparison data.

Figure 5-2 Viewing the monitoring graph



----End

Creating Alarm Rules

Setting GaussDB(DWS) alarm rules allows you to customize the monitored objects and notification policies and determine the running status of your GaussDB(DWS) at any time.

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) management 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.

Figure 5-3 Selecting the object to be monitored

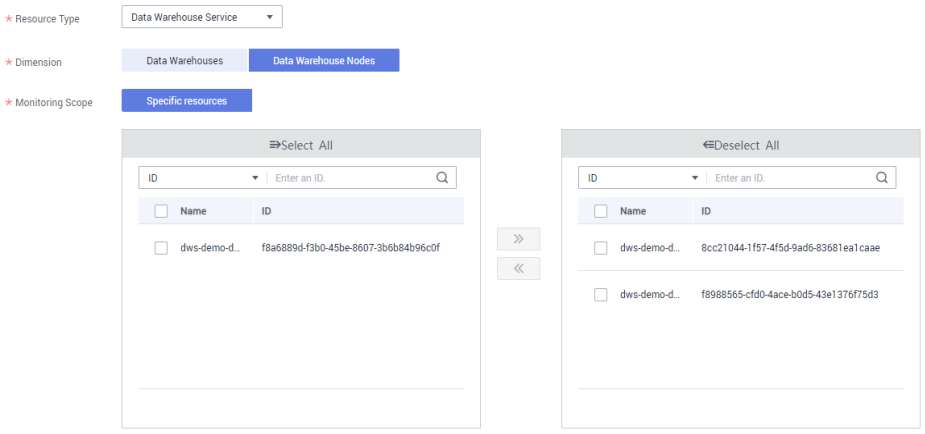


Figure 5-4 Setting the alarm policy

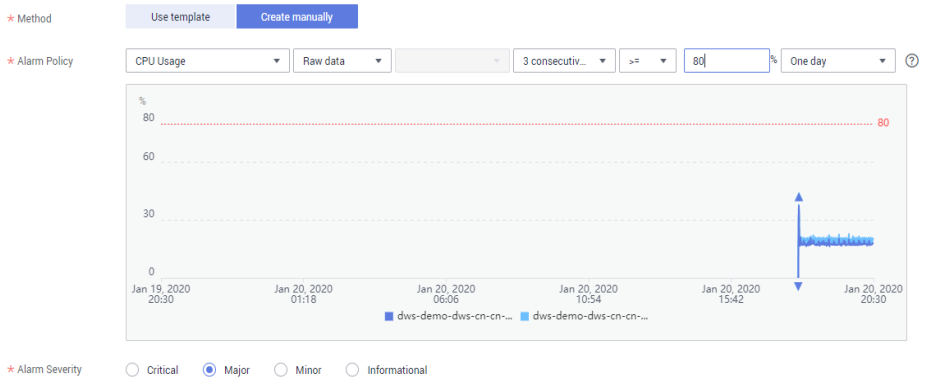


Table 5-7 Configuring alarm parameters

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

Parameter	Description	Example Value
Monitoring Scope	Resource scope to which an alarm rule applies. Select **Specific resources** and select one or more monitoring objects. Select the ID of the cluster instance or node you have created. Click  to synchronize the monitoring objects to the right pane.	Specific resources
Method	Select **Use template** or **Create manually** as required. – If no alarm template is available, set **Method** to **Create manually** and configure related parameters to create an alarm rule. – If you have available alarm rule templates, set **Method** to **Use template**, so that you can use a template to quickly create alarm rules.	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 5-5** 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.

Figure 5-5 Configuring alarm notifications

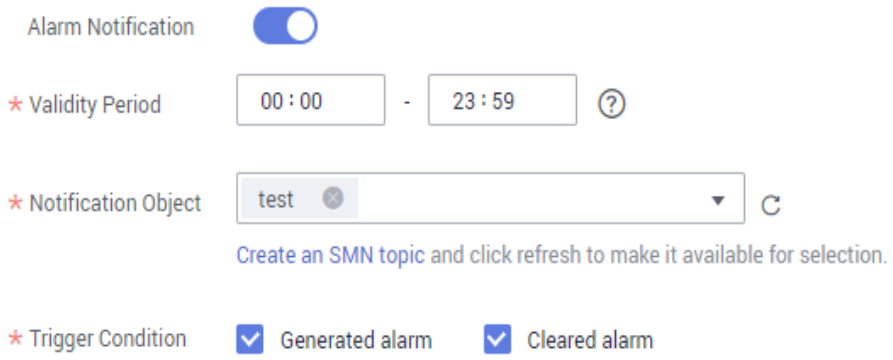


Table 5-8 Configuring alarm notifications

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.	-
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 details about how to configure OBS storage transfer, see "Viewing Alarm History > Configuring OBS Data Storage" in the ***Cloud Eye User Guide***.

5.4 Alarms

5.4.1 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 supports only the database kernel of 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) management console.

Step 2 In the navigation pane on the left, click **Alarms**.

Step 3 On the page that is displayed:

- **Existing Alarm Statistics**

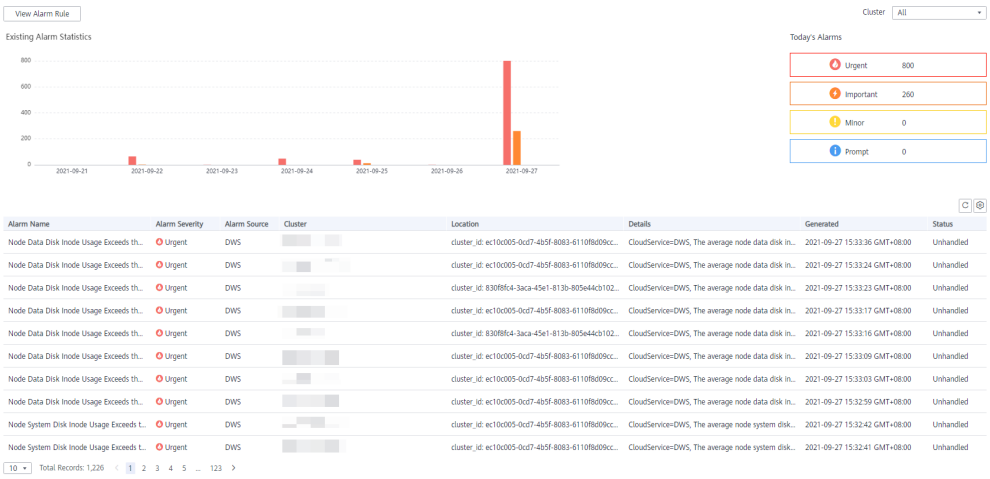
  Statistics of the existing alarms in the past seven days are displayed by alarm severity in a bar chart. In this way, you can see clearly the number and category of the alarms generated in the past week.

- **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, cluster name, location, description, generation date, and status.



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 5-9 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.
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.

Type	Name	Severity	Description
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	Node Data Disk Inode Usage Exceeds the Threshold	Urgent: > 95%; important: > 90%	This alarm is generated if the threshold of data disk (**/var/chroot/DWS/data***[n]*) inode 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]*) inode usage is lower than the threshold and the constraint is not met.
Default	Data Flushed 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 cluster default 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	The packet loss retransmission rate on the cluster network exceeds the threshold.	Urgent	This alarm is generated if the DMS alarm module detects a high retransmission rate on a server and no alarm suppression conditions are met. If the retransmission rate decreases, the alarm will be automatically cleared.
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.
Custom	*Name of the user-defined threshold alarm*	*User-defined alarm severity*	*Alarm description*

5.4.2 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

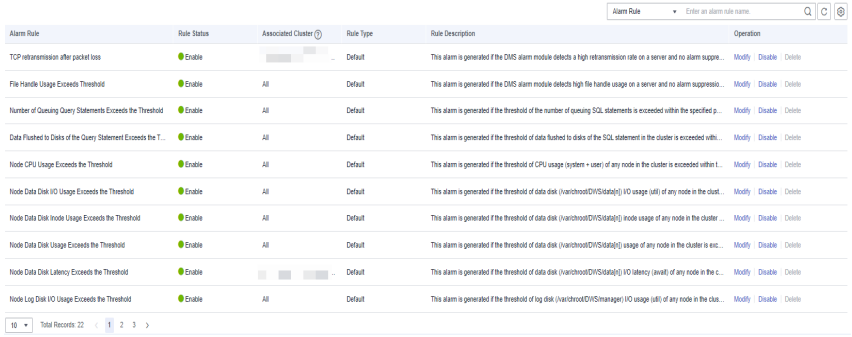
After a cluster is migrated, to monitor alarms of the new cluster, change the cluster bound to the alarm rule to the new cluster. You can also create an alarm rule for the new cluster.

Viewing Alarm Rules

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

Step 2 In the navigation pane on the left, choose **Alarms**.

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.



----End

Modifying an Alarm Rule

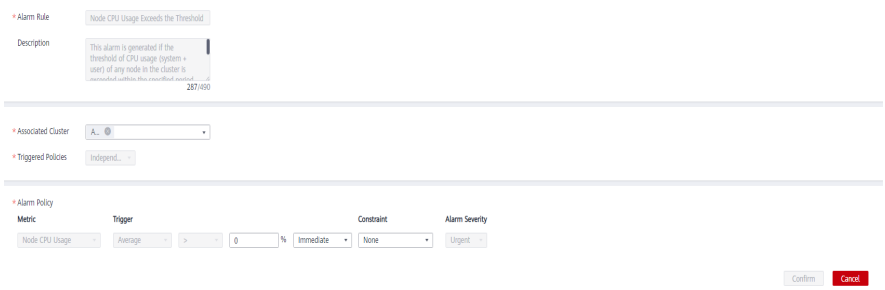
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 In the navigation pane on the left, click **Alarms**.

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.

- **Alarm rule name**: The rule name contains 6 to 64 characters (letters, digits, Chinese characters, slashes) and must start with a non-digit character.
- **Description**
- **Associated Cluster**: From the drop-down list, select the current tenant's clusters to which the alarm rule applies.
- **Triggered Policies**
  - **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.
- **Alarm Policy**
  - **Metric**: GaussDB(DWS) monitoring metric, which is the data source used by the alarm engine for threshold determination.
  - **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**.



> **NOTE**
>
> 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.

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

----End

Creating an Alarm Rule

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

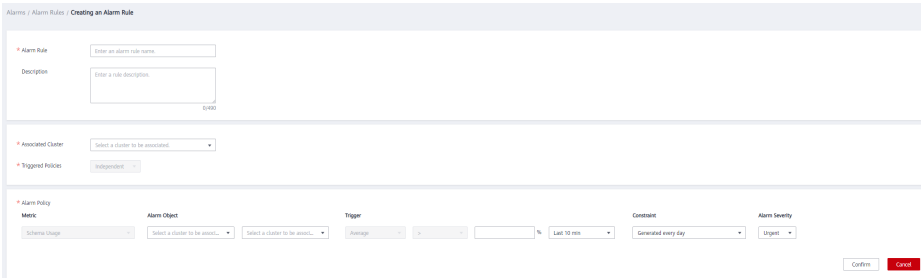
Step 2 In the navigation pane on the left, click **Alarms**.

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, clusters associated with the rule, and alarm policy.

- **Alarm rule name**: The rule name contains 6 to 64 characters (letters, digits, Chinese characters, slashes) and must start with a non-digit character.
- **Description**
- **Associated Cluster**: From the drop-down list, select the current tenant's clusters to which the alarm rule applies.
- **Triggered Policies**
  - **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.
- **Alarm Policy**
  - **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**.

Figure 5-6 Creating an alarm rule



NOTE

Currently, only alarm rules of schema usage metrics can be created on GaussDB(DWS).

----End

5.4.3 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 **Alarms** > **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.

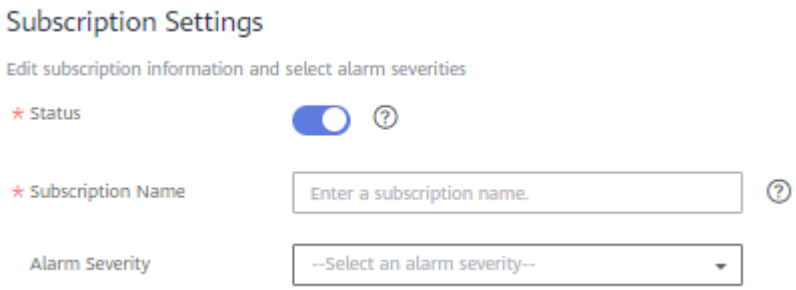
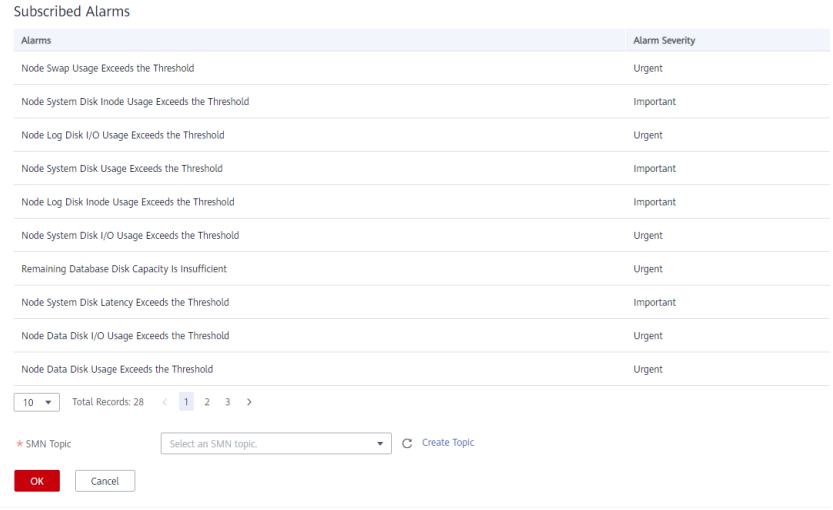


Table 5-10 Subscription parameters

Parameter	Description
Status	Whether to enable the alarm subscription. indicates that the alarm subscription is enabled. indicates that the alarm subscription is disabled. 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.
Alarm Severity	Severity of the alarm you want to subscribe to: **Urgent**, **Important**, **Minor**, or **Prompt**

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** to switch to the SMN console page. For details, .



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** as the service that can publish messages to this topic.

Step 6 Confirm the information and click **OK**.

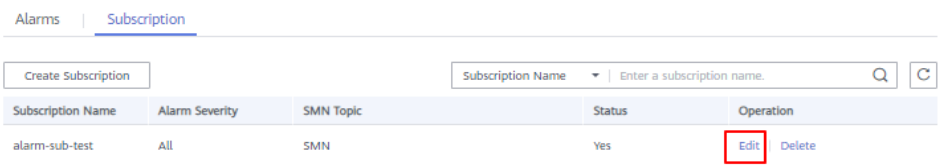
----End

Modifying a Subscription

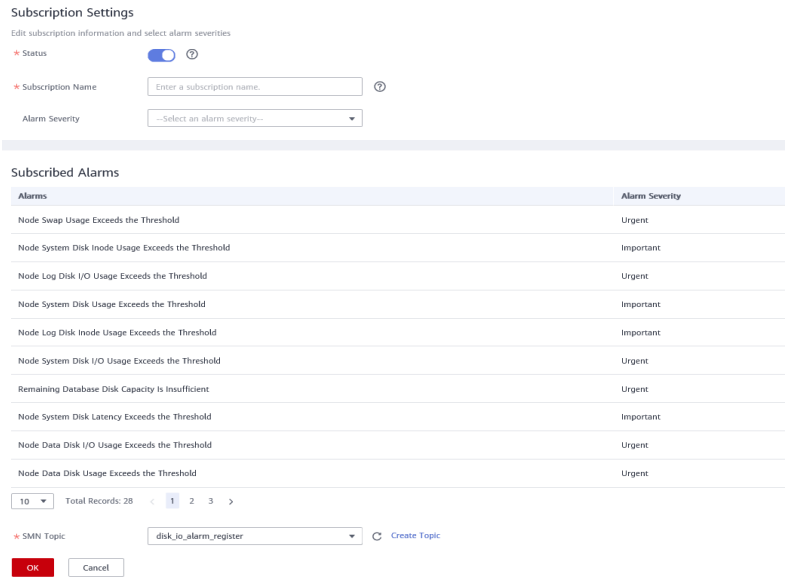
Step 1 Log in to the GaussDB(DWS) management console.

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

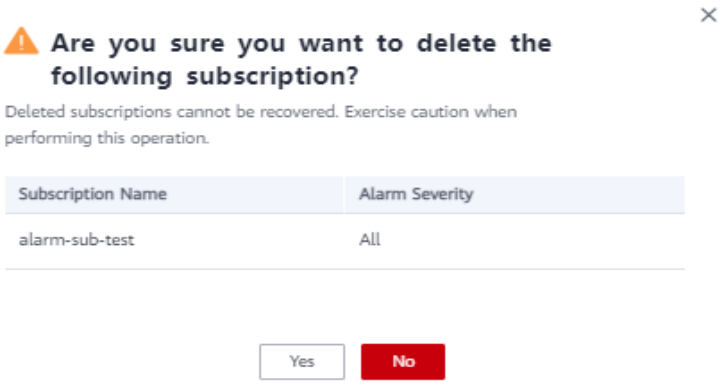
----End

Deleting a Subscription

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

Step 2 In the navigation pane on the left, choose **Alarms** > **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

5.4.4 Alarm Handling

5.4.4.1 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 Severity	Auto Clear
DWS_2000000017	Critical	Yes

Alarm Parameters

Parameter	Description
Alarm Source	Indicates the name of the system for which the alarm is generated and the detailed alarm type.
Cluster Name	ID of the cluster for which the alarm is generated
Location Information	ID and name of the cluster for which the alarm is generated
Detail Information	**CloudService** indicates the cloud service for which the alarm is generated, including the service name, resource ID, resource name, first alarm time, and formatted alarm information. Example: CloudServiceDWS, resourceId=xxxx-xxxx-xxxx-xxxx, resourceIdName=test_dws, first_alarm_time:2023-01-11:19:02:09. The average number of query statements queuing in cluster test_dws within 10 minutes is 30, which exceeds the threshold 10.
Generated	Time when an alarm is generated.
Status	Indicates the status of the current alarm.

Impact on the System

SQL queries are blocked. As a result, the execution time is too long.

Possible Causes

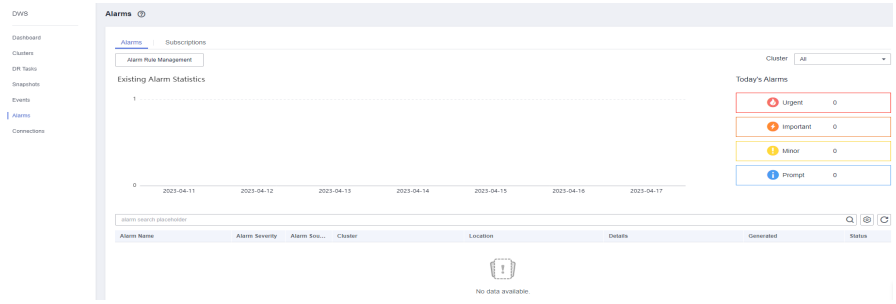
The number of queuing query statements during service execution exceeds the alarm threshold.

Handling Procedure

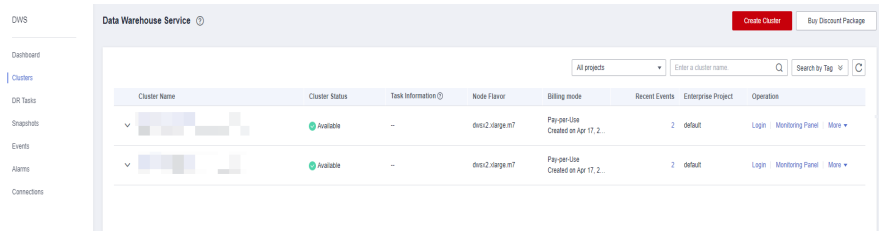
Check whether the current queuing jobs in the cluster are normal.

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

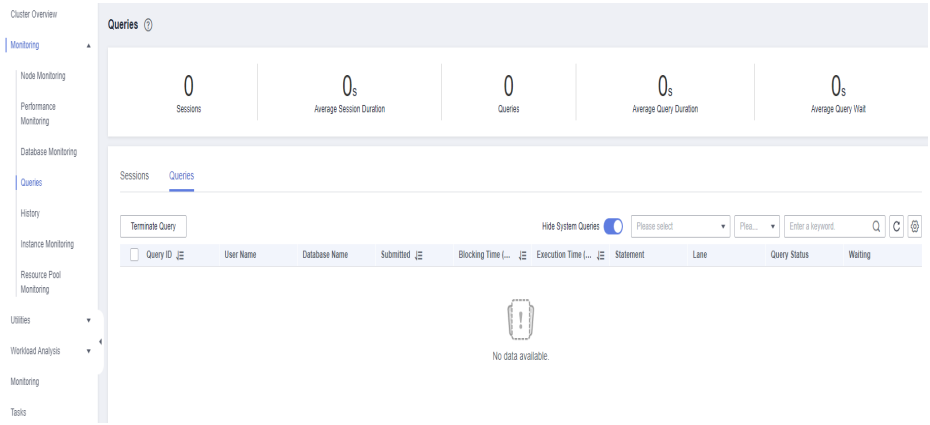
Step 2 On the **Alarms** page, select the current cluster from the cluster selection drop-down list in the upper right corner and view the alarm information of the cluster in the last seven days. Locate the name of the cluster that triggers the alarm based on the location information.



Step 3 On the **Cluster** > **Dedicated Cluster** 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.



NOTE

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.

5.4.4.2 DWS_2000000016 Data Spilled to Disks for a 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 Severity	Auto Clear
DWS_2000000016	Critical	Yes

Parameters

Parameter	Description
Source	Indicates the name of the system for which the alarm is generated and the detailed alarm type.
Cluster Name	ID of the cluster for which the alarm is generated
Location Information	ID and name of the cluster for which the alarm is generated
Other Information	**CloudService** indicates the cloud service for which the alarm is generated, including the service name, resource ID, resource name, database name, username connecting to the backend, and query ID. **first_alarm_time** indicates the time when the alarm is generated for the first time. **query statement** indicates the query statement that triggers the alarm, along with the amount of data spilled to disks caused by the query statement. Example: CloudService=DWS,resourceId: *xxxx-xxxx-xxxx-xxxx*, resourceIdName: *test_dws*, db_name: *test_db*, user_name: *test_user*, query_id: *756942385413326696*, first_alarm_time: *2022-12-30:12:42:77*. query statement (ID= *756942385413326696*) *select num,value,cnt,rk,cnt/sumcnt as ratio,sum(ratio) over (over by rk) as cumuratio from...*; The result set is spilled to disks, and the spill size is *15* GB.
Time	Indicates the time when the alarm was generated.
Status	Status of an alarm.

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.

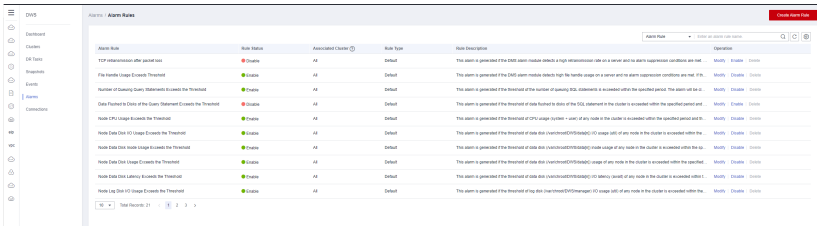
Handling Procedure

Step 1 Check whether the execution plan is poor in performance.

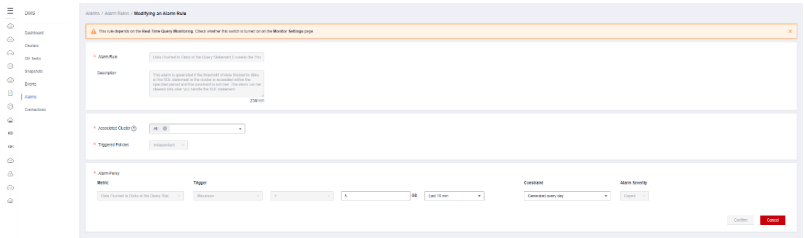
1. Obtain the SQL statement from the additional information of the alarm, run the **ANALYZE** statement on the involved tables. Run the SQL statement again, and check whether the amount of data spilled to disks decreases.
2. If there is no obvious effect, run the **EXPLAIN PERFORMANCE** command to view the actual execution information of the alarm SQL statement. For details, see **SQL Execution Plan**. Based on the execution information, if both the estimated memory usage (operator memory) and peak memory are large, for example, greater than 20% of **max_process_memory**, you need to optimize the query. For details, see **Optimization Process**.

Step 2 Check whether the alarm configuration is proper.

1. Return to the GaussDB(DWS) management console and choose **Alarms** > **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.



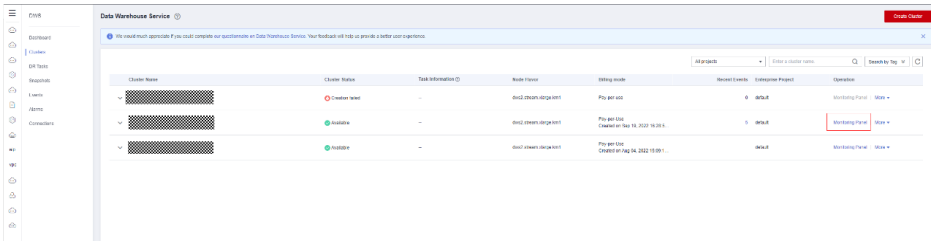
3. If the cluster disk capacity is high, you can increase the alarm reporting threshold. It is recommended that the alarm reporting threshold be less than or equal to 5% of the capacity of a single data disk.

CAUTION

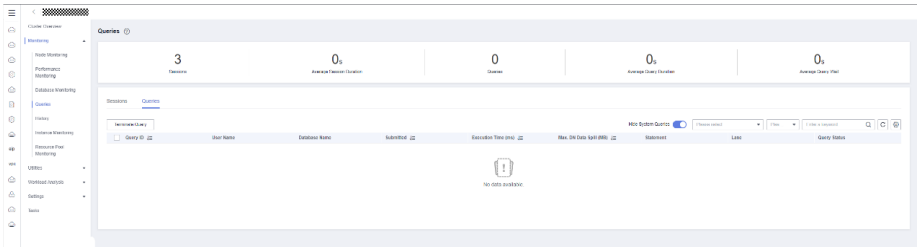
If the threshold is too large, data spilled to a disk may cause disk usage alarms or even the cluster to be read-only. If the data disk usage is close to or exceeds 80%, you are advised to clear unnecessary data when adjusting the threshold. For details about the GUI configuration, see **Alarm Rules**.

Step 3 Kill the SQL statements that cause large data spills.

1. Return to the GaussDB(DWS) management console.
2. On the **Cluster** > **Dedicated Cluster** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.

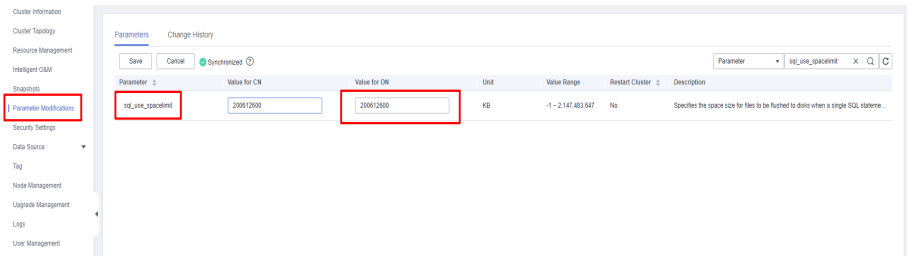


3. Choose **Monitoring** > **Queries**. Click  to see the data spill in the **Max. DN Data Spill (MB)** column.



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.



----End

Alarm Clearance

This alarm is automatically cleared when data spill drops down to a low level.

5.4.4.3 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 Severity	Auto Clear
DWS_2000000001	Critical	Yes

Parameters

Parameter	Description
Source	Indicates the name of the system for which the alarm is generated, for example, GaussDB(DWS).
Cluster Name	Indicates the cluster for which the alarm is generated.
Location Information	Includes ID and name of the cluster for which the alarm is generated, and ID and name of the instance for which the alarm is generated, for example, cluster_id: *xxxx-xxxx-xxxx-xxxx*, cluster_name: *test_dws*, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*.
Detail Information	Detailed information about the alarm, including the cluster, instance, and threshold information. Example: CloudService=DWS, resourceId= *xxxx-xxxx-xxxx-xxxx*, resourceIdName=test_dws, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*, host_name: *host-192-168-1-122*, first_alarm_time: *2022-01-30 10:30:00*; The average CPU usage of the node within 10 minutes is *90.54*%, which exceeds the threshold *90*%.
Generated	Time when an alarm is generated.
Status	Indicates the status of the current alarm.

Impact on the System

If the CPU usage is high for a long time, service processes may respond slowly or become unavailable.

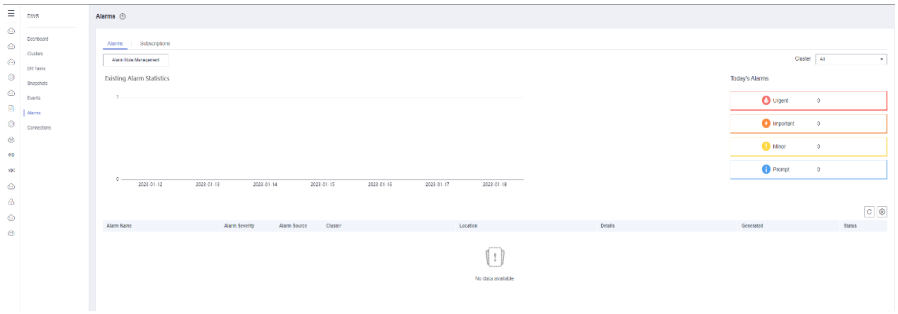
Possible Causes

- Complex services occupy a large number of CPU resources.
- The CPU configuration of the cluster is too low to meet service requirements.

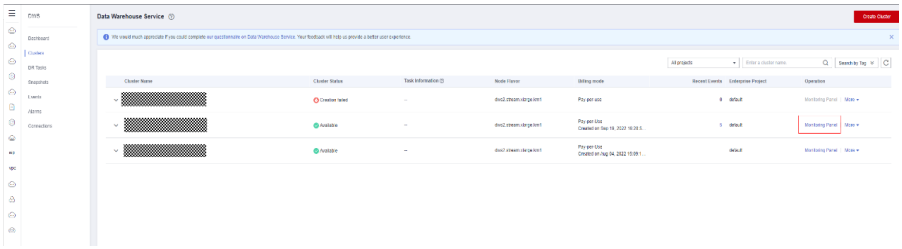
Handling Procedure

Step 1 Check the CPU usage of each node.

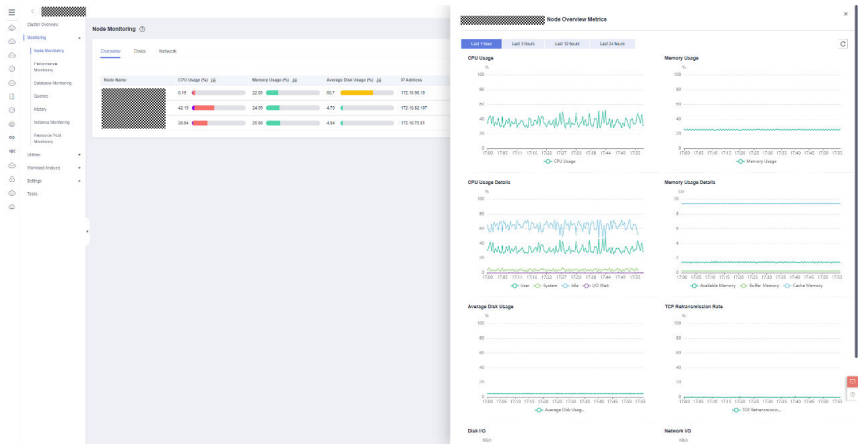
1. Log in to the GaussDB(DWS) console.
2. On the **Alarms** page, in the cluster selection drop-down list in the upper right corner, select the cluster for which the alarm is generated, view the alarm information of the cluster in the last seven days, and locate the name of the node for which the alarm is generated based on the location information.



3. On the **Cluster** > **Dedicated Cluster** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.



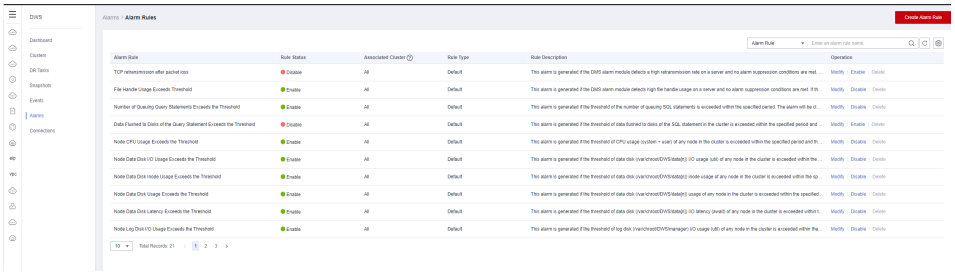
4. Choose **Monitoring** > **Node Monitoring** > **Overview** to view the CPU usage of each node in the current cluster. Click  on the right to view the CPU performance metrics in the last 1, 3, 12, or 24 hours and see whether there is a sharp increase in the CPU usage.



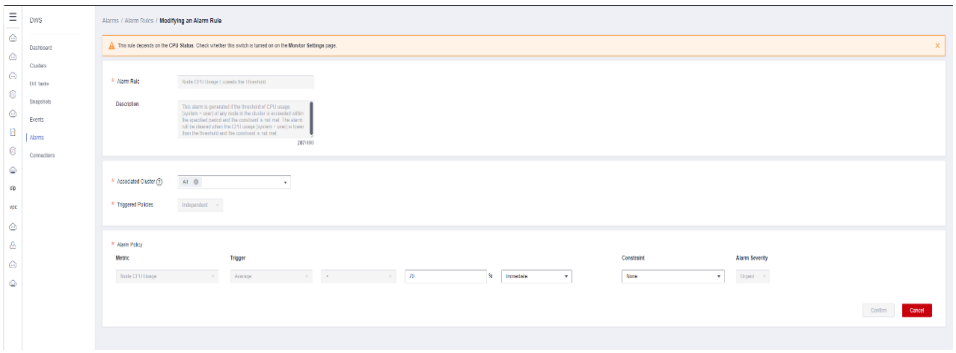
- If the CPU usage frequently increases and then returns to normal in a short period of time, it indicates that the CPU usage temporarily spikes during service execution. In this case, you can adjust the alarm threshold through **Step 2** to reduce the number of reported alarms.
- If the CPU usage remains 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 Alarms > Alarm Rules.



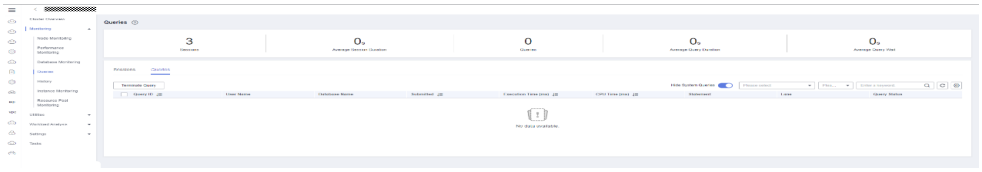
2. Locate the row that contains the **Node CPU Usage Exceeds the Threshold**, and click **Modify** in the **Operation** column. The **Modifying an Alarm Rule page** is displayed.



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.
2. After confirming with the service side, select the query ID to be stopped and click **Stop Query**.



----End

Alarm Clearance

After the CPU usage decreases, the alarm is automatically cleared.

5.4.4.4 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).

Alarm Attributes

Alarm ID	Alarm Severity	Auto Clear
DWS_2000000009	Critical	Yes

Alarm Parameters

Parameter	Description
Alarm Source	Indicates the name of the system for which the alarm is generated, for example, GaussDB(DWS).
Cluster Name	Indicates the cluster for which the alarm is generated.

Parameter	Description
Location Information	Includes ID and name of the cluster for which the alarm is generated, and ID and name of the instance for which the alarm is generated, for example, cluster_id: *xxxx-xxxx-xxxx-xxxx*, cluster_name: *test_dws*, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*.
Detail Information	Detailed information about the alarm, including the cluster, instance, disk, and threshold information. Example: CloudService=DWS, resourceId= *xxxx-xxxx-xxxx-xxxx*, resourceIdName=test_dws, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*, host_name: *host-192-168-1-122*, disk_name: */dev/vdb*, first_alarm_time: *2022-01-30 10:30:00*; The log disk I/O usage of the node within 10 minutes is *90.54*%, exceeding the threshold *90*%.
Generated	Time when an alarm is generated.
Status	Indicates the status of the current alarm.

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

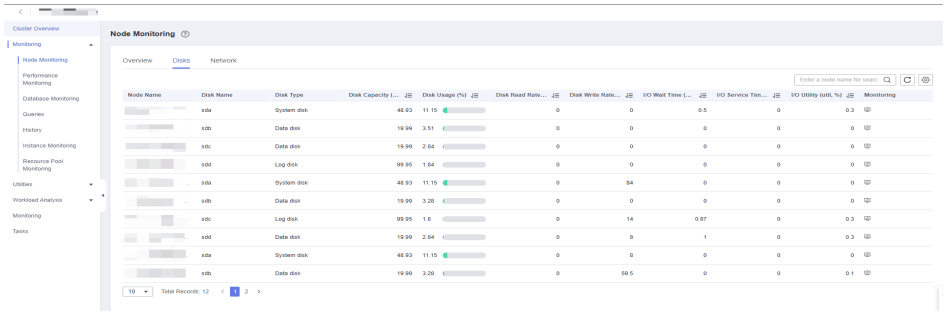
- A large number of read or write operations are performed during peak hours.
- A large amount of data spills to disks due to the execution of complex statements.
- Data is scanned by the Scan operator.

Handling Procedure

Step 1 On the **Clusters** > **Dedicated Clusters** page, locate the row that contains the target cluster and click **Monitoring** in the **Operation** column.

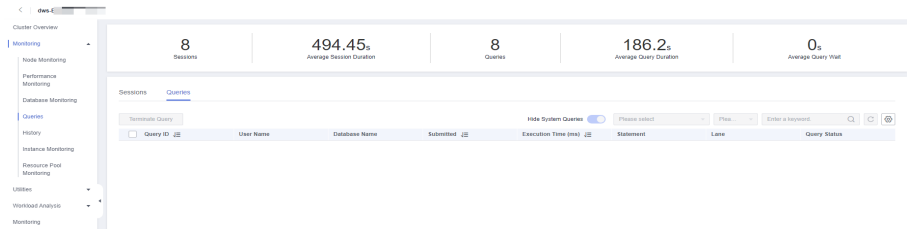
Step 2 In the navigation pane on the left, choose **Monitoring** > **Node Monitoring**. On the **Node Monitoring** page, view the data disk I/O usage and disk I/O rate.

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 tree on the left 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.

5.4.4.5 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 Severity	Auto Clear
DWS_2000000006	Critical/Major	Yes

Parameters

Parameter	Description
Source	Name of the system for which the alarm is generated, for example, GaussDB(DWS).
Cluster Name	Cluster for which the alarm is generated.
Location Information	IDs and names of the cluster and instance for which the alarm is generated, for example, cluster_id: *xxxx-xxxx-xxxx-xxxx*, cluster_name: *test_dws*, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*.
Detail Information	Detailed information about the alarm, including the cluster, instance, disk, and threshold information. Example: CloudService=DWS, resourceId: *xxxx-xxxx-xxxx-xxxx*, resourceIdName: *test_dws*, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-2-1*, host_name: *host-192-168-1-122*, disk_name: */dev/vdb*, first_alarm_time: *2022-11-26 11:14:58*; The average data disk usage of the node within 10 minutes is *84*%, which exceeds the threshold *80*%.
Generated	Time when an alarm is generated.
Status	Status of the current alarm.

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.

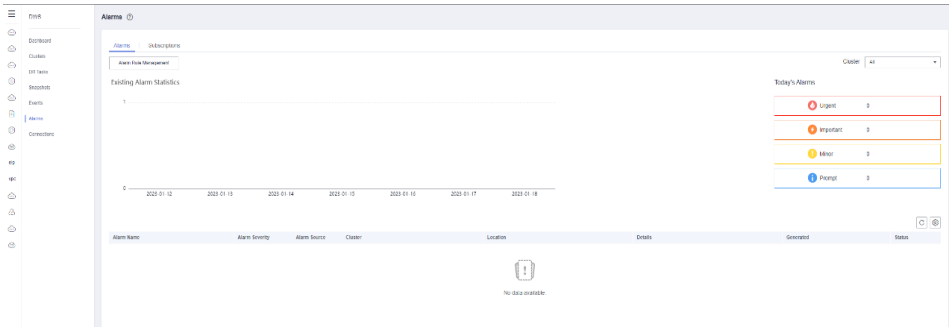
Possible Causes

- 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.

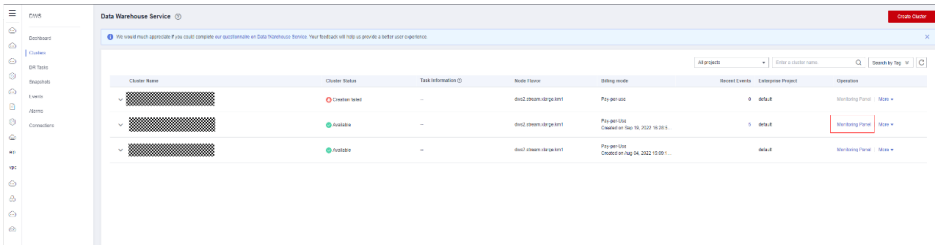
Handling Procedure

Step 1 Check the disk usage of each node.

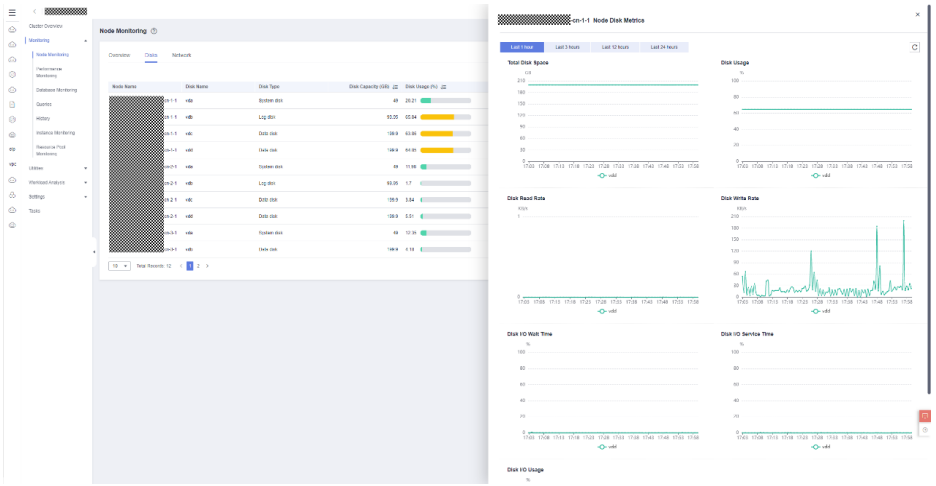
1. Log in to the GaussDB(DWS) console.
2. On the **Alarms** page, select the current cluster from the cluster selection drop-down list in the upper right corner and view the alarm information of the cluster in the last seven days. Locate the name of the node for which the alarm is generated and the disk information based on the location information.



3. On the **Cluster** > **Dedicated Cluster** 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.



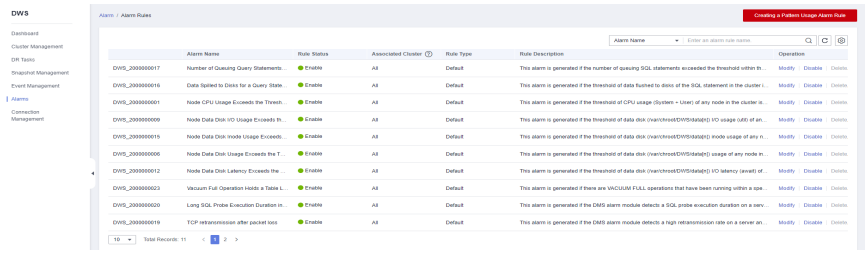
   - If the data disk usage frequently increases and then returns to normal in a short period of time, it indicates that the disk usage temporarily spikes due to service execution. In this case, you can adjust the alarm threshold through **Step 2** to reduce the number of reported alarms.
   - If the usage 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.

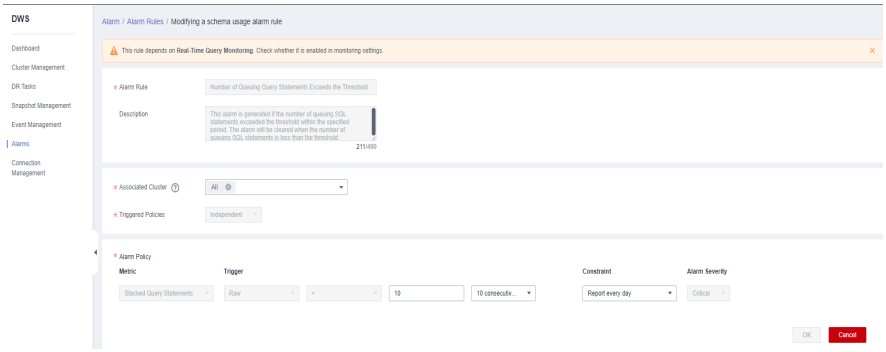
- If the usage of more than half of the data disks in the cluster exceeds 70%, the data volume in the cluster is large. In this case, refer to **Step 4** to clear data or perform **Disk Capacity Expansion**.
- If the difference between the highest and lowest data disk usage in the cluster exceeds 10%, refer to **Step 5** to handle data skew.

Step 2 Check whether the alarm configuration is proper.

1. Return to the GaussDB(DWS) management console and choose **Alarms** > **Alarm Rule**.



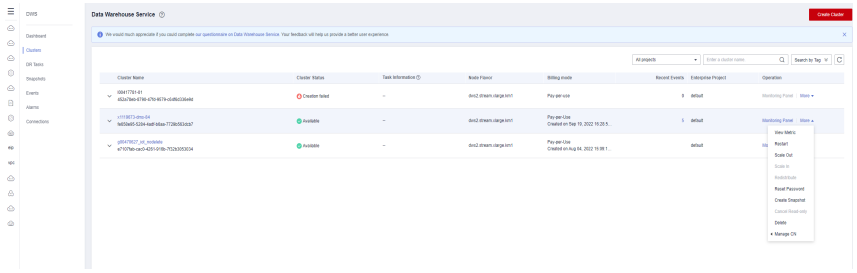
2. Locate the row that contains **Node Data Disk Usage Exceeds the Threshold** and click **Modify** in the **Operation** column. On the **Modifying an Alarm Rule** page, view the configuration parameters of the current alarm.



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) management console, click **Clusters** > **Dedicated Clusters**, locate the row that contains the abnormal cluster, and choose **More** > **Cancel Read-Only** in the **Operation** column.



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. Standard data warehouse + SSD cloud disk, stream data warehouse, and hybrid data warehouse: See **Disk Capacity Expansion**.

b. Standard data warehouse + SSD local disk and old standard data warehouse (disk scale-out is not supported): 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.

5.4.4.6 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 Severity	Auto Clear
DWS_2000000012	Major	Yes

Alarm Parameters

Parameter	Description
Alarm Source	Indicates the name of the system for which the alarm is generated, for example, GaussDB(DWS).
Cluster Name	Indicates the cluster for which the alarm is generated.
Location Information	Includes ID and name of the cluster for which the alarm is generated, and ID and name of the instance for which the alarm is generated, for example, cluster_id: *xxxx-xxxx-xxxx-xxxx*, cluster_name: *test_dws*, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*.
Detail Information	Detailed information about the alarm, including the cluster, instance, disk, and threshold information. Example: CloudService=DWS, resourceId= *xxxx-xxxx-xxxx-xxxx*, resourceIdName=test_dws, instance_id: *xxxx-xxxx-xxxx-xxxx*, instance_name: *test_dws-dws-cn-cn-1-1*, host_name: *host-192-168-1-122*, disk_name: */dev/vdb*, first_alarm_time: *2022-01-30 10:30:00*. The data disk I/O usage of the node within 10 minutes is *90.54* %, exceeding the threshold *90* %.
Generated	Time when an alarm is generated.
Status	Indicates the status of the current alarm.

Impact on the System

High disk latency will slow down the data read/write speed, causing the cluster performance to deteriorate.

Possible Causes

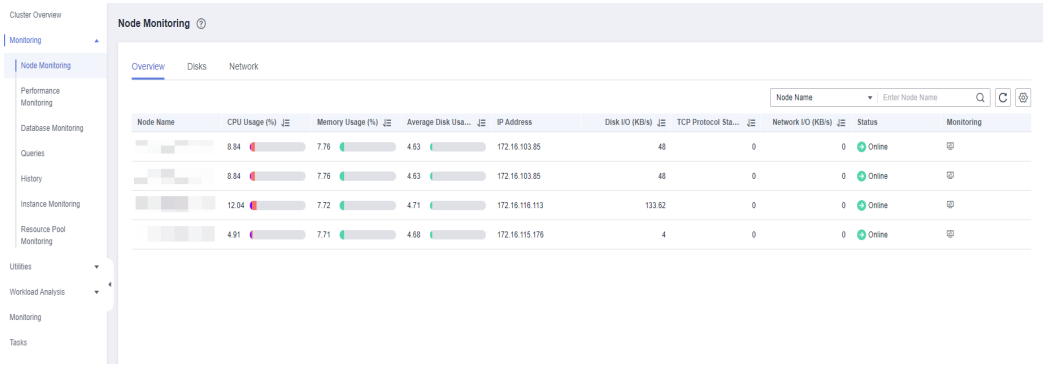
The database is in peak hours and there are a large number of read and write requests.

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.

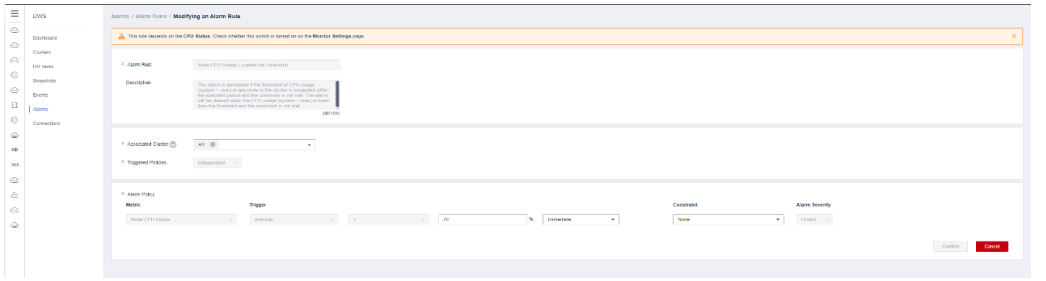
Step 2 In the navigation pane on the left, choose **Monitoring** > **Node Monitoring**. On the **Node Monitoring** page, view the CPU usage, disk usage, and memory usage.

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 Click **Alarms**, switch to the **Alarms** tab page, and click **Alarm Rule Management** in the upper left corner.

Step 4 Locate the row that contains **Node Data Disk Latency Exceeds the Threashold**, 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.

5.4.4.7 DWS_2000000023 Vacuum Full Operation That Holds A Table Lock Exceeds the Threshold

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 Severity	Auto Cleared
DWS_2000000023	Important	Yes

Alarm Parameters

Parameter	Description
Alarm Source	Name of the system for which the alarm is generated, for example, GaussDB(DWS).
Cluster Name	Cluster for which the alarm is generated.
Location Info	ID and name of the cluster for which the alarm is generated. Example: **cluster_id:** *xxxx-xxxx-xxxx-xxxx*,**cluster_name:** *test_dws*
Detail Information	Detailed information about the alarm, including the cluster and threshold information. Example: **CloudService=DWS, resourceId:** *xxxx-xxxx-xxxx-xxxx*, **resourceIdName:** *test_dws*, **first_alarm_time:** *2022-11-26 11:14:58*; *The VACUUM FULL operation [query_id] in the cluster takes more than 20 minutes.*
Generated	Time when an alarm is generated.
Status	Status of the current alarm.

Impact on the System

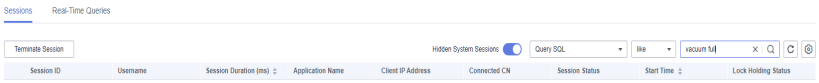
Other operations cannot the table. As a result, workloads cannot be executed.

Possible Causes

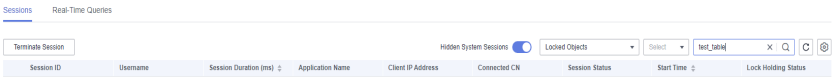
There is a VACUUM FULL operation that holds a table lock for a long time in the cluster.

Handling Procedure

Step 1 In the navigation pane of the monitoring panel, choose **Monitoring** > **Queries**. In the session list, set the search criteria to **LIKE** and search for the keyword **vacuum full**.



Step 2 Check whether there is a table lock waiting for VACUUM FULL to complete by querying the locked object.



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.

5.4.4.8 DWS_2000000020 SQL Probe of the Cluster Usage Exceeds the Threshold

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 Severity	Auto Cleared
DWS_2000000020	Critical	Yes

Alarm Parameters

Parameter	Description
Alarm Source	Name of the system for which the alarm is generated, for example, GaussDB(DWS).
Cluster Name	Cluster for which the alarm is generated.
Location Info	ID and name of the cluster for which the alarm is generated. Example: **cluster_id:** *xxxx-xxxx-xxxx-xxxx*,**cluster_name:** *test_dws*

Parameter	Description
Detail Information	Detailed information about the alarm, including the cluster and threshold information. Example: **CloudService=DWS, resourceId:** *xxxx-xxxx-xxxx-xxxx*, **resourceIdName:** *test_dws*, **first_alarm_time:** *2022-11-26 11:14:58*; *The test_dws cluster exceeds twice the SQL probe threshold. The number of SQL probes in the cluster exceeds the threshold:'select xxx from xxxx'.*
Generated	Time when an alarm is generated.
Status	Status of the current alarm.

Impact on the System

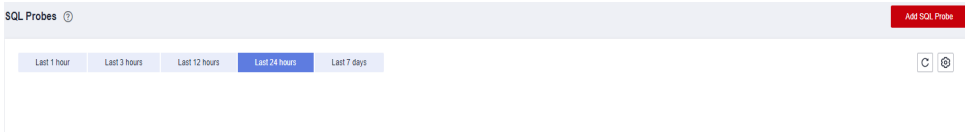
The cluster performance deteriorates or the cluster is faulty.

Possible Causes

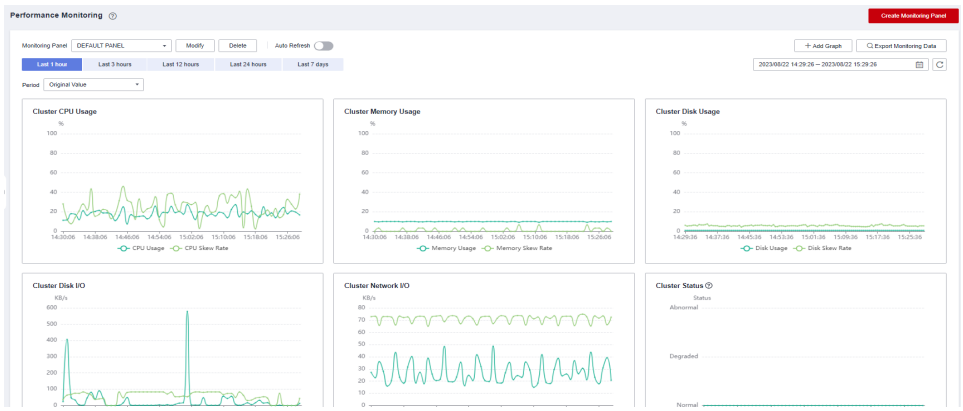
The service load of the cluster is high or the cluster is faulty. As a result, the execution of the SQL probe becomes slow.

Handling Procedure

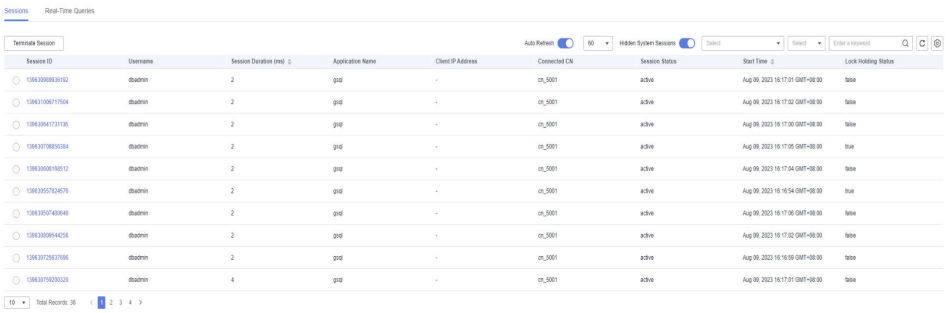
Step 1 In the navigation pane of the monitoring panel, choose **Utilities** > **SQL Probes**. Check SQL probe execution.

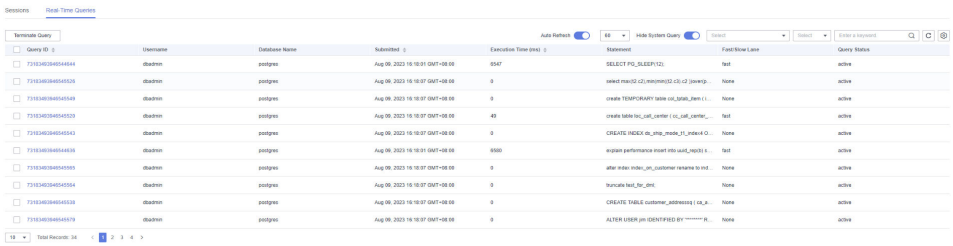


Step 2 In the navigation pane, choose **Monitoring** > **Performance Monitoring**. Check the monitoring metrics such as the CPU usage, disk usage, and memory usage to determine whether the workloads are high or any metric is abnormal.



Step 3 In the navigation pane, choose **Monitoring** > **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.

5.4.4.9 DWS_2000000018 Queue Congestion in the Cluster Default Resource Pool

Description

GaussDB(DWS) uses **Resource Pool** to control memory, I/O, and CPU resources, manages and allocates resources based on task priorities, and manages user services loads. When resources are insufficient, some SQL statements have to queue up to wait for other statements to be executed.

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 Severity	Auto Clear
DWS_2000000018	Critical	Yes

Parameters

Parameter	Description
Source	Name of the system for which the alarm is generated and the detailed alarm type.
Cluster Name	ID of the cluster for which the alarm is generated
Location Information	ID and name of the cluster for which the alarm is generated
Alarm Information	**CloudService** indicates the cloud service for which the alarm is generated, including the service name, resource ID, resource name, first alarm time, and formatted alarm information. Example: CloudServiceDWS, resourceId=xxxx-xxxx-xxxx-xxxx, resourceIdName=test_dws, first_alarm_time:2023-01-11:19:02:09. The default resource pool of cluster **test_dws** are blocked within in the past 20 minutes.
Time	Time when the alarm was generated.
Status	Current status of an alarm.

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

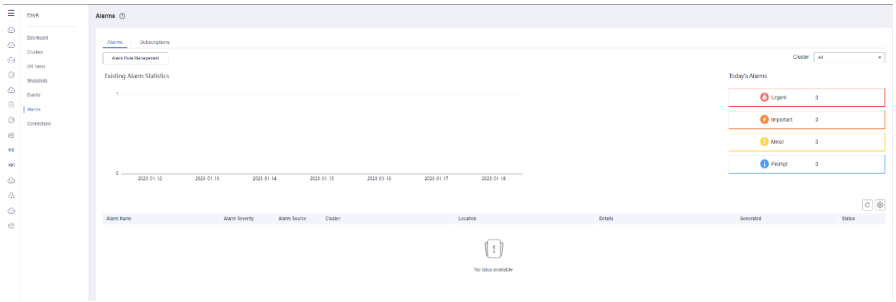
- The estimated query memory usage is too large. As a result, the accumulated estimated memory usage exceeds the upper limit of the dynamic available memory, causing CCN queuing.
- Competition for public resources such as CPU and I/O deteriorates the performance of running queries.

Handling Procedure

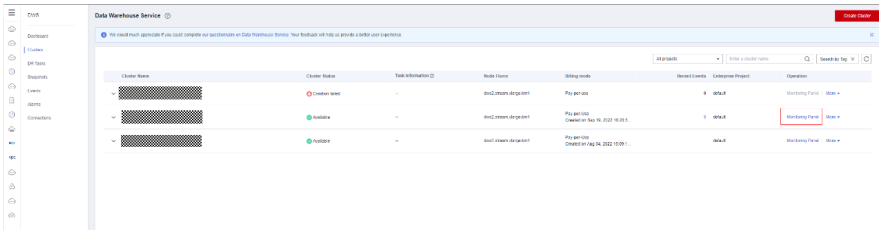
Step 1 Check whether the queue is caused by too large estimated memory.

Step 2 Check whether the available memory of the cluster is normal.

1. Log in to the GaussDB(DWS) console.
2. On the **Alarms** page, 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.

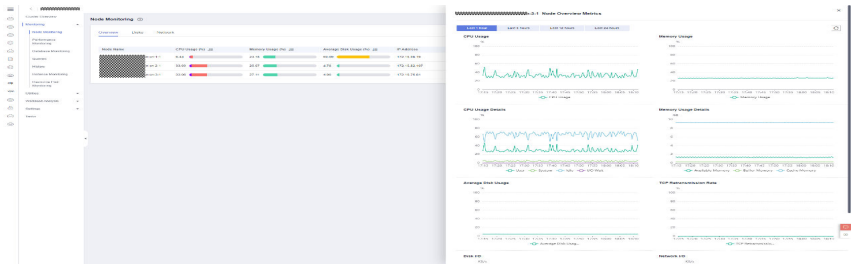


3. On the **Cluster** > **Dedicated Cluster** page, locate the row that contains the cluster for which the alarm is generated and click **Monitoring Panel** in the **Operation** column.



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,
```

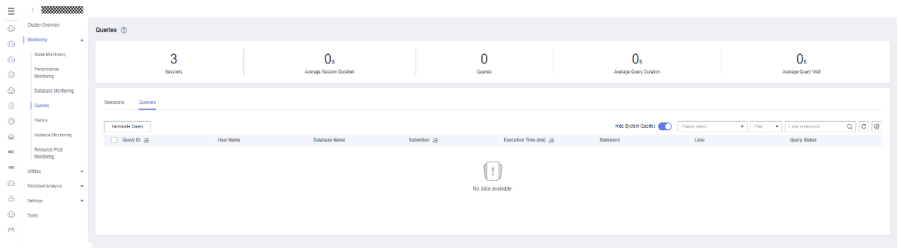


```
  SUM(CASE WHEN a.enqueue = 'waiting in global queue' THEN 1 ELSE 0 END) AS global_wait,
  SUM(CASE WHEN s.lane= 'fast' AND a.state = 'active' AND (a.enqueue IS NULL OR a.enqueue = 'no waiting queue') THEN 1 ELSE 0 END) AS fast_run,
  SUM(CASE WHEN s.lane= 'fast' AND a.enqueue = 'waiting in respool 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** > **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**.

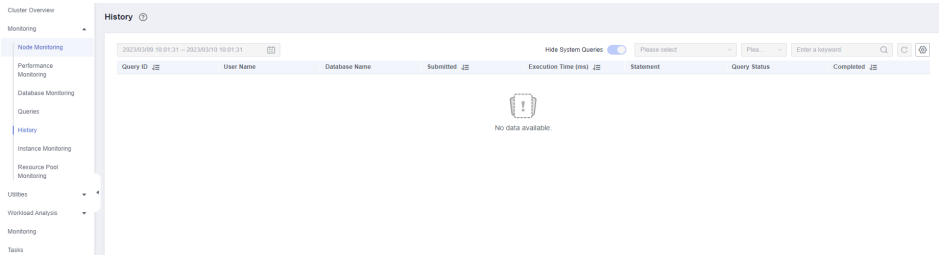
----End

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.



5.5 Event Notifications

5.5.1 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 5-11 Events whose **Event Source Category** is **Cluster**

Event Type	Event Name	Event Severity	Event
Management	createClusterFail	Warning	Failed to create the cluster.
Management	createClusterSuccess	Normal	Cluster created successfully.
Management	createCluster	Normal	Cluster creation started.
Management	extendCluster	Normal	Cluster scale-out started.
Management	extendClusterSuccess	Normal	Cluster scaled out successfully.
Management	extendClusterFail	Warning	Failed to scale out the cluster.
Management	deleteClusterFail	Warning	Failed to delete the cluster.

Event Type	Event Name	Event Severity	Event
Management	deleteClusterSuccess	Normal	Cluster deleted successfully.
Management	deleteCluster	Normal	Cluster deletion started.
Management	restoreClusterFail	Warning	Failed to restore the cluster.
Management	restoreClusterSuccess	Normal	Cluster restored successfully.
Management	restoreCluster	Normal	Cluster restoration started.
Management	restartClusterFail	Warning	Failed to restart the cluster.
Management	restartClusterSuccess	Normal	Cluster restarted successfully.
Management	restartCluster	Normal	Cluster restarted.
Management	configureMRSExtDataSources	Normal	Configuration of MRS external data source for the cluster started.
Management	configureMRSExtDataSourcesFail	Warning	Failed to configure the MRS external data source for the cluster.
Management	configureMRSExtDataSourcesSuccess	Normal	MRS external data source configured successfully for the cluster.
Management	deleteMRSExtDataSources	Normal	Deletion of MRS external data source for the cluster started.
Management	deleteMRSExtDataSourcesFail	Warning	Failed to delete the MRS external data source for the cluster.
Management	deletedMRSExtDataSourcesSuccess	Normal	MRS external data source deleted successfully for the cluster.
Management	bindEipToCluster	Normal	Bound an EIP to the cluster.

Event Type	Event Name	Event Severity	Event
Management	bindEipToClusterFail	Warning	Failed to bind an EIP to the cluster.
Management	unbindEipToCluster	Normal	Unbound an EIP from the cluster.
Management	unbindEipToCluster-Fail	Warning	Failed to unbind an EIP from the cluster.
Management	refreshEipToCluster	Normal	Refreshed the cluster's EIP.
Management	refreshEipToCluster-Fail	Warning	Failed to refresh the cluster's EIP.
Management	dmsCreateWDRSuc-cessfully	Normal	Generating the workload report...
Management	failedToCreateWDR	Warning	Failed to generate the workload report.
Management	dmsDeleteWDRSuc-cessfully	Normal	Workload report deleted.
Management	failedToDeleteWDR	Warning	Failed to delete a workload report.
Management	dmsUpdateWDRCon-figSuccessfully	Normal	Workload report parameters updated.
Management	failedToUpdateWDR-Config	Warning	Failed to update workload report parameters.
Management	dmsCreateWorkloadS napshotSuccessfully	Normal	Creating the workload snapshot...
Management	failedToCreateWor-kloadSnapshot	Warning	Failed to create the workload snapshot.
Security	resetPasswordFail	Warning	Failed to reset the password.
Security	resetPasswordSuccess	Normal	Password of the cluster reset successfully.
Security	updateConfiguration	Normal	Updating security parameters of the cluster started.
Security	updateConfiguration-Fail	Warning	Failed to update security parameters of the cluster.

Event Type	Event Name	Event Severity	Event
Security	updateConfiguration-Success	Normal	Security parameters of the cluster updated successfully.
Monitoring	repairCluster	Normal	The node is faulty. Repairing the cluster starts.
Monitoring	repairClusterFail	Warning	Failed to repair the cluster.
Monitoring	repairClusterSuccess	Normal	Cluster repaired successfully.

- The following table lists the events whose **Event Source Category** is **Snapshot**.

  Table 5-12 Events whose **Event Source Category** is **Snapshot**

Event Type	Event Name	Event Severity	Event
Management	deleteBackup	Normal	Snapshot deleted successfully.
Management	deleteBackupFail	Warning	Failed to delete the snapshot.
Management	createBackup	Normal	Snapshot creation started.
Management	createBackupSuccess	Normal	Snapshot created successfully.
Management	createBackupFail	Warning	Failed to create the snapshot.

5.5.2 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, click **Event Management**.

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 5-7 Subscription Settings

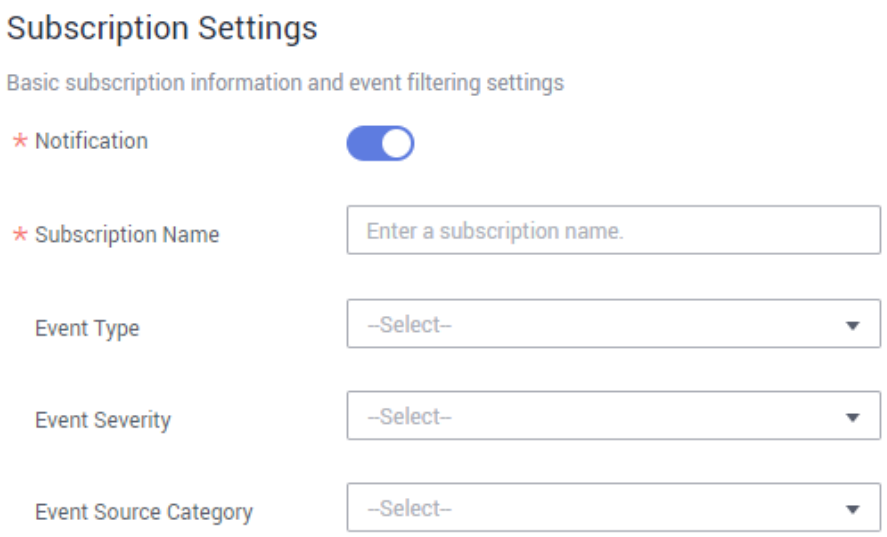


Table 5-13 Subscription parameters

Parameter	Description
Notification	Enable or disable event subscription.  indicates that event subscription is enabled.  indicates that event subscription is disabled. This function is enabled by default. After the function 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.

Parameter	Description
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*.

Figure 5-8 Creating a topic

Step 6 Click **OK** to complete the subscription.

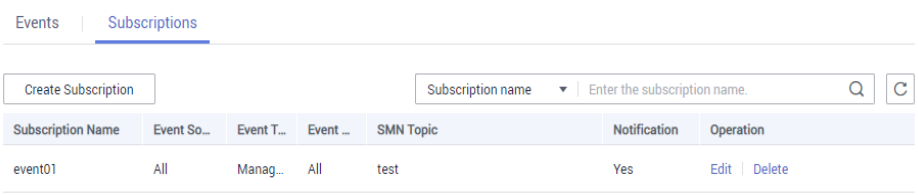
----End

Modifying the Subscription

Step 1 On the **Event Management** page of the GaussDB(DWS) management console, click the **Subscription** tab.

Step 2 In the **Operation** column of the row containing the specified subscription, click **Edit** to enter the **Edit Subscription** page.

Figure 5-9 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 On the **Event Management** page of the GaussDB(DWS) management console, click the **Subscription** tab.

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

5.5.3 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.

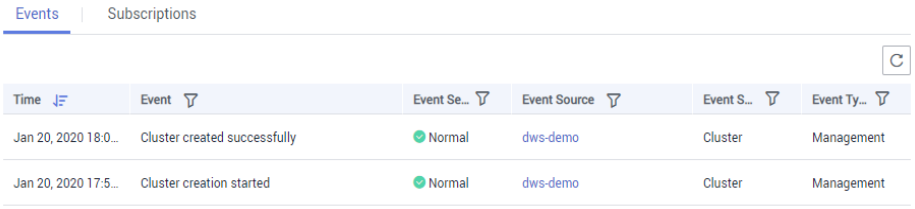
Step 2 In the navigation tree on the left, click **Events**.

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 filter the events by clicking  next to a field (except **Time**) and selecting the criteria.

Figure 5-10 Event page



----End

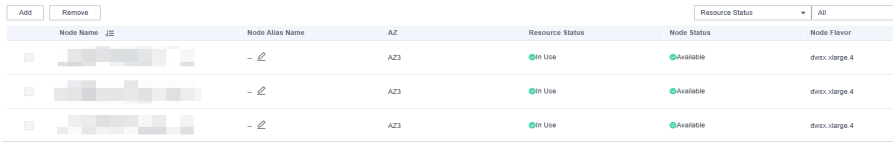
6 Specifications Change and Scaling

6.1 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.

To modify an alias, click  next to it.



NOTE

- This feature is supported only in 8.1.1.200 or later cluster versions.
- The hybrid 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. For example, if 180 more BMS nodes are needed, add them in three batches (60 for each batch). If some nodes fail to be added, add them again. After all the 180 nodes are successfully added, use the nodes for cluster scale-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.
- A node that is successfully added, usually for scale-out purposes, is called an idle node. It starts incurring charges once added. You are advised to add

nodes only when necessary and use them for scale-out in a timely manner once they are added.

- 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) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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**.

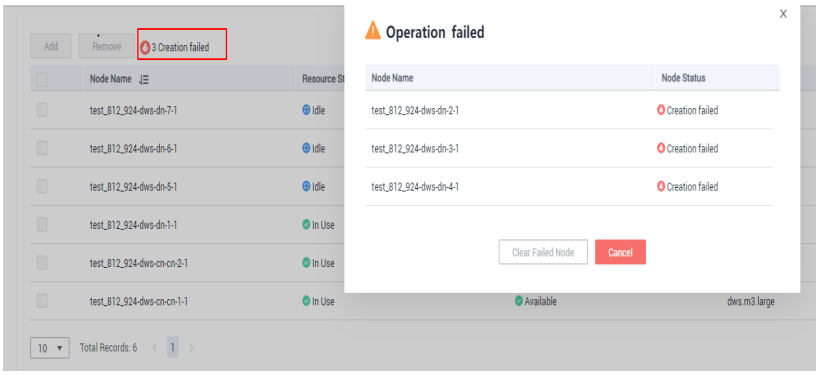
Step 4 Click **Add Node**, enter the number of nodes to be added, and select **I agree**. Click **Next: Confirm**.

Step 5 Click **Submit**. The nodes will start to be added, as shown in the following figure.

----End

NOTE

The nodes that fail to be added will be automatically rolled back and recorded in the displayed list, as shown in the following figure.



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.

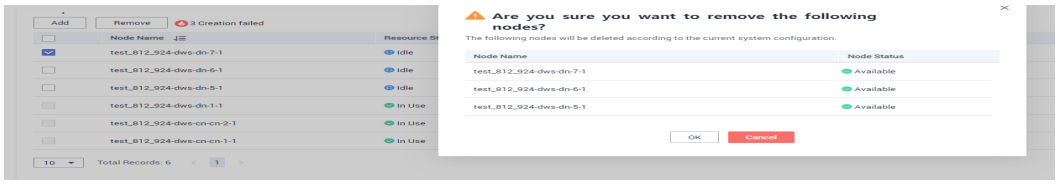
Procedure

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

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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 nodes to be removed, click **Remove**, and click **Yes** to submit the task.



Step 5 The nodes that are successfully removed will not be displayed on the **Nodes** page.

----End

6.2 Scaling Nodes

6.2.1 Scaling Out a Cluster

When you need more compute and storage resources, add more nodes for cluster scale-out on the management console.

NOTE

- When you scale out the standard data warehouse cluster, use the same storage specifications as the cluster.
- Nodes cannot be added to a hybrid data warehouse (standalone).

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.

Impact on the System

- Before the scale-out, exit 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.
- During the scale-out, functions such as cluster restart, scale-out, snapshot creation, database administrator password resetting, and cluster deletion are disabled.
- During an offline scale-out, the cluster automatically restarts. Therefore, the cluster stays **Unavailable** for a period of time. After the cluster is restarted, the status becomes **Available**. After scale-out, the system dynamically redistributes user data among all nodes in the cluster.
- 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 and database objects are checked. Do not create or delete databases or tablespaces in this period, or the cluster may fail to be locked.
- 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 database fails to be rolled back 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.
- In the cloud native 9.0.2 scale-out scenario, if the number of buckets allocated to each DN is not between [3, 20], automatic scaling is 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** 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.
- 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) management 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** > **Scale Node** > **Scale Out**. The scale-out page is displayed.

Step 4 Specify the number of nodes to be added.

- 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, subnet, 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**, **Scale Online** will be disabled, **Auto Redistribution** will be enabled, and **Redistribution Mode** will be **Offline** by default.
- If you choose **Custom**, you can configure the following advanced configuration parameters for online 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 Job**: If you enable online scale-out, you can configure automatic job termination.
- **Time Before 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 32. The default value is 4.
- **Redistribution Mode**: It can be set to **Online** or **Offline**. After confirming that the information is correct, click **OK** in the displayed dialog box.

Step 6 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**. In the last phase of scale-out, 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 or BMS nodes first by referring to **Adding Nodes** for a large-scale cluster, and scale out the cluster with 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.
- 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) management console.

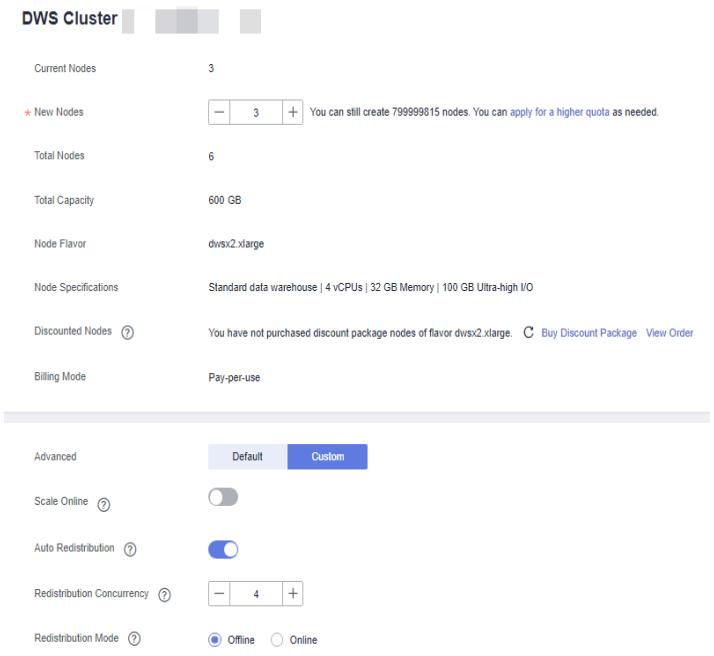
Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Scale Out**.

If there are idle nodes in the cluster, the system displays a message asking you whether to add nodes.

Step 4 Configure the scale-out and redistribution parameters as required. For details, see **Scaling Out a Cluster**.

Then click **Next: Confirm**.



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

----End

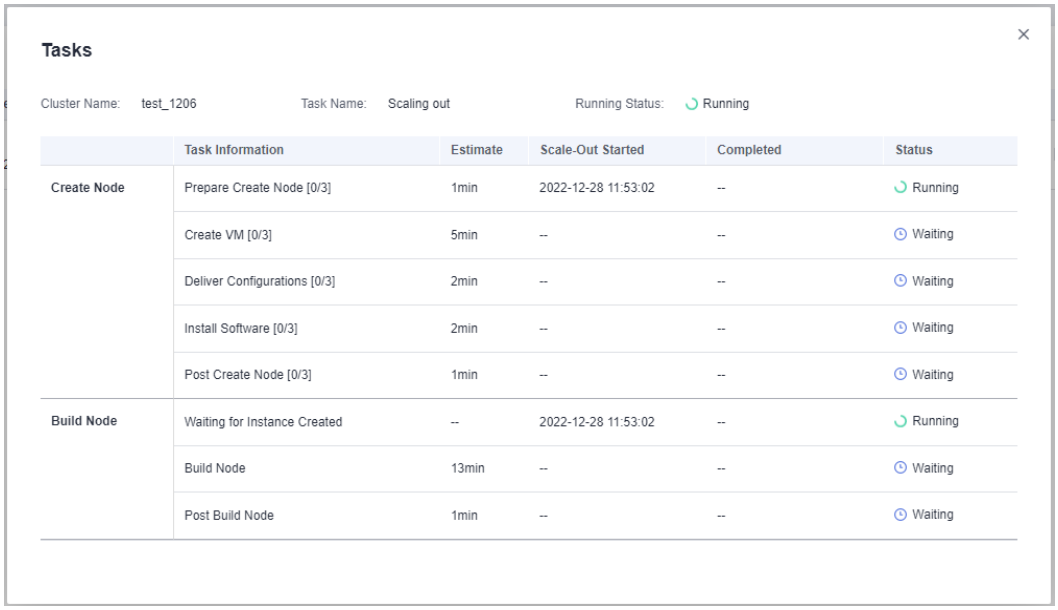
Viewing Scaling Details

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



----End

6.2.2 Cluster Redistribution

6.2.2.1 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.

Currently, **offline redistribution** and **online redistribution** are supported. 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 cluster versions.
- Offline scheduling is not supported in 8.2.0 or later.
- 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 **Configure advanced parameters**).
- 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

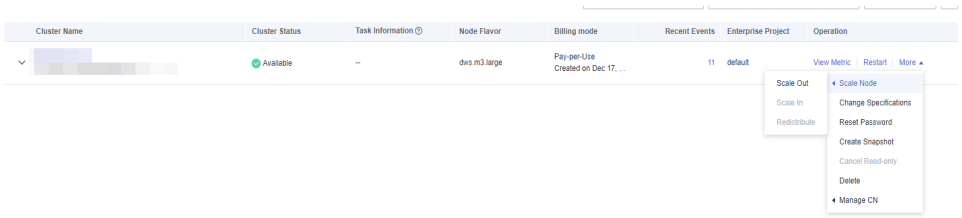
- In offline redistribution mode, the database does not support DDL and DCL operations. Tables that are being redistributed support only simple DQL operations.
- During table redistribution, a shared lock is added to tables. All insert, update, and delete operations as well as DDL operations on the tables 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

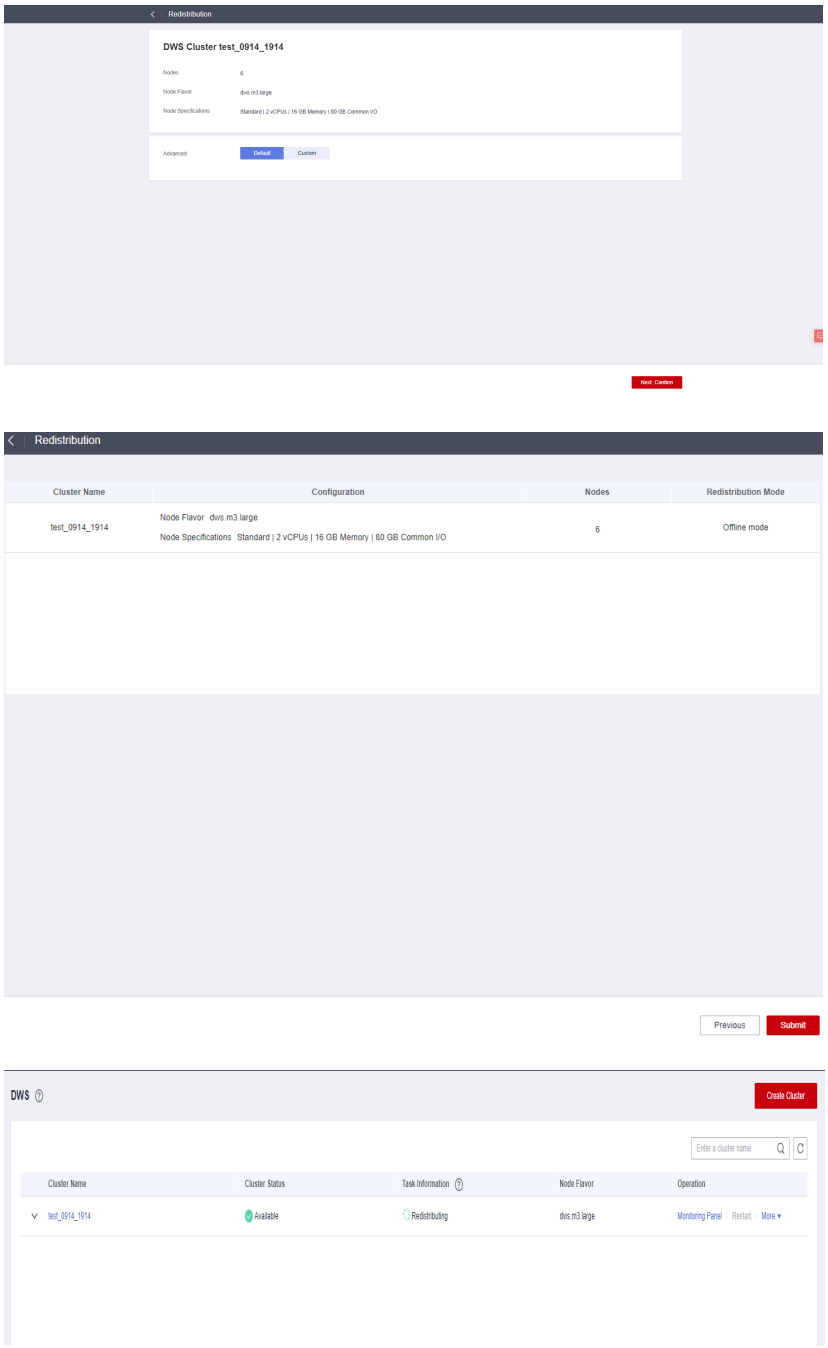
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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, 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**

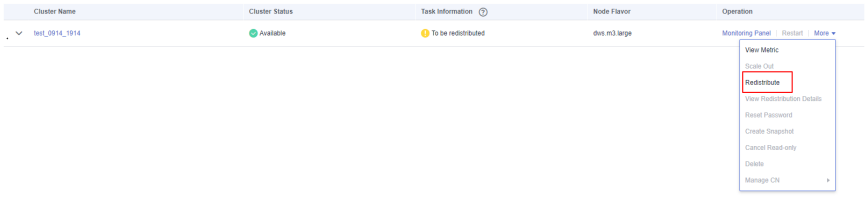
  - Join queries across node groups
  - Local table renaming, schema modification, **DROP**, **TRUNCATE**, **TRUNCATE-PARTITION**
- The following operations cannot be performed on tables that are being redistributed:
  - Run **ALTER TABLE** statements (except for **TRUNCATE PARTITION**), including adding or deleting columns or partitions.
  - Create, modify, or delete indexes.
  - Run **VACUUM FULL** or **CLUSTER** 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

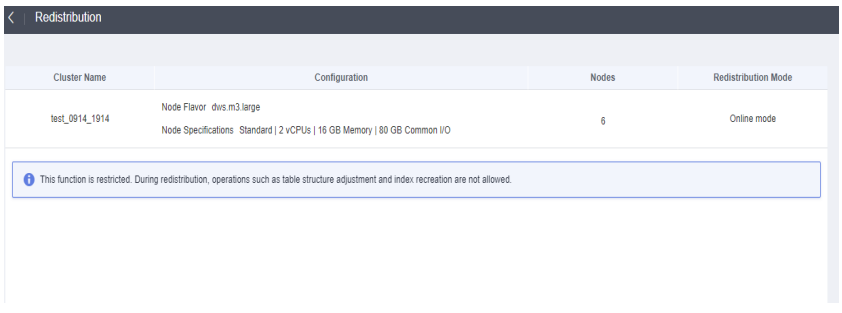
Step 1 Log in to the GaussDB(DWS) management console.

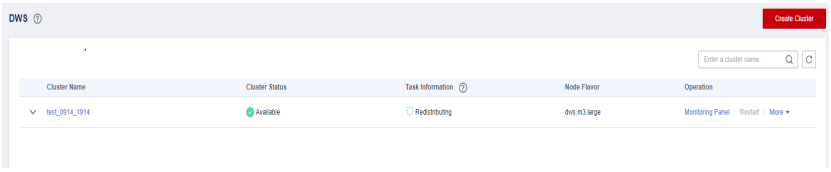
Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

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

6.2.2.2 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

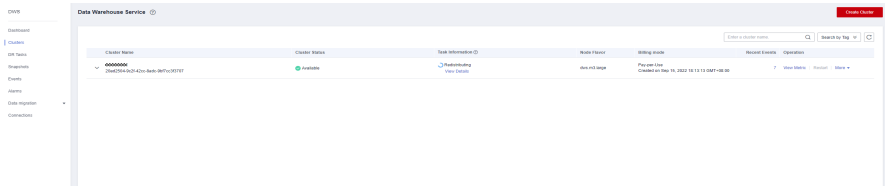
You can check redistribution details only if the cluster is being redistributed, failed to be redistributed, or is suspended. There may be a delay in the statistics update.

Procedure

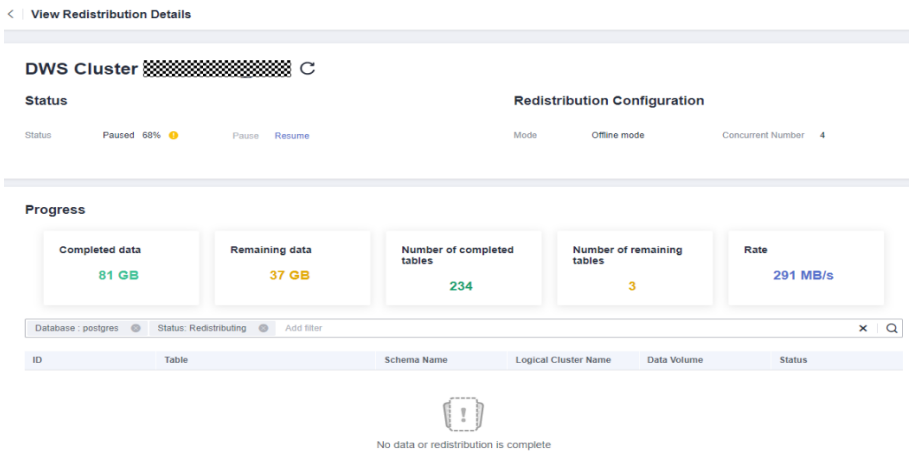
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

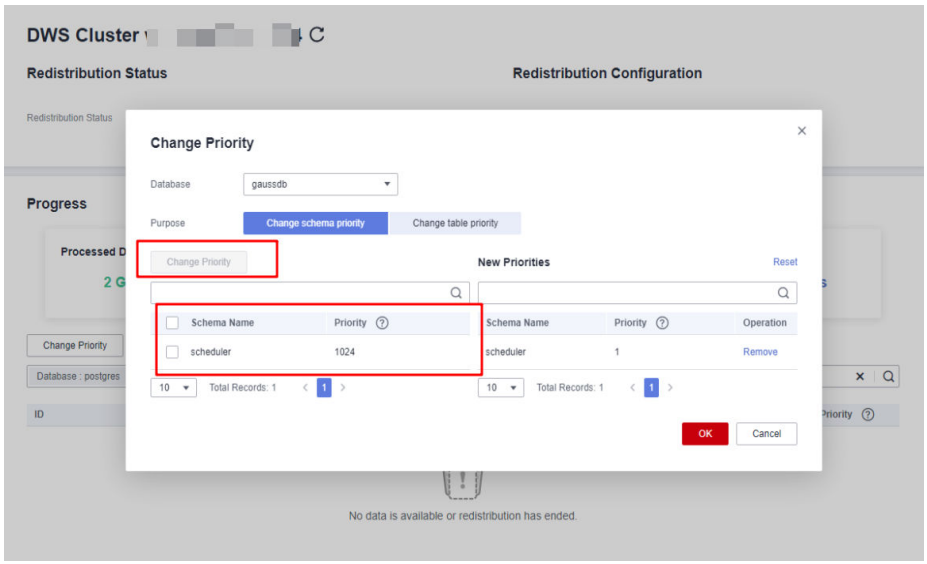
Step 3 In the **Task Information** column of a cluster, click **View Details**.

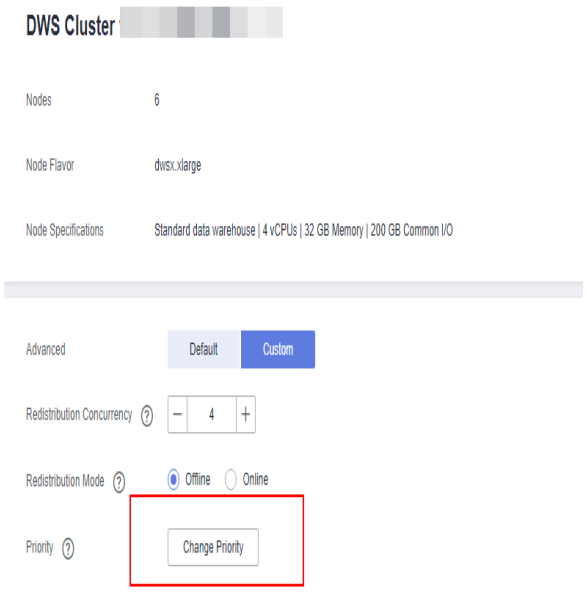


Step 4 Check the redistribution status, configuration, progress, and redistribution details of all the tables in a specified database. Specify a database that and can be searched by table redistribution status and table name. If all the tables in a database have completed redistribution, no data will 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.

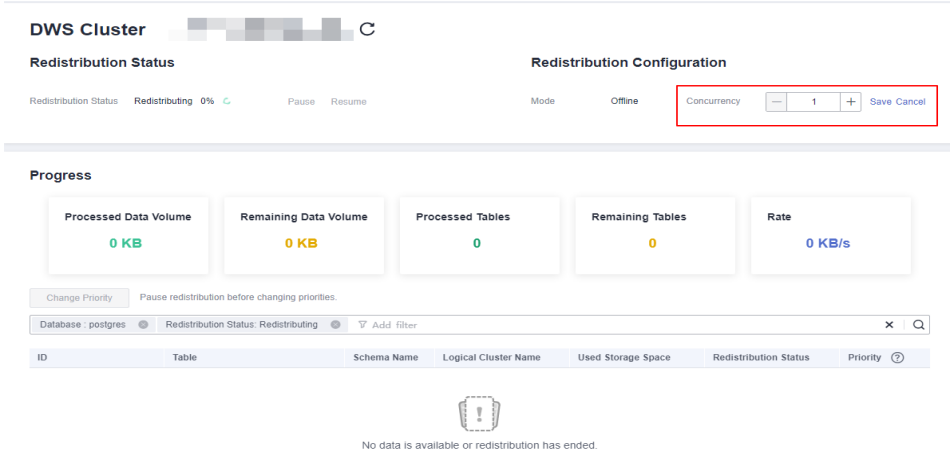




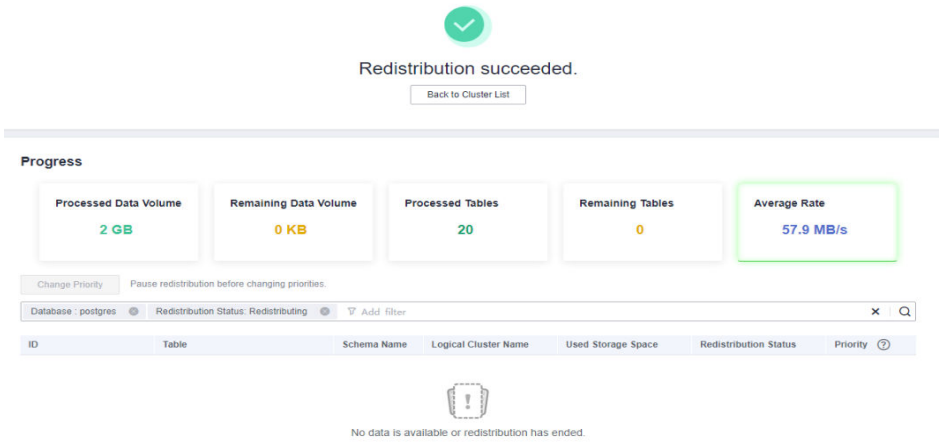
Step 6 The number of concurrent redistribution tasks can be adjusted during redistribution.

NOTE

Cluster 8.1.0 and earlier versions do not support dynamic adjustment. To change redistribution concurrency, suspend redistribution first.



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.



----End

6.2.3 Scaling In a Cluster

You can scale in your clusters on the console to release unnecessary computing and storage resources provided by GaussDB(DWS).

NOTE

- By default, scaled in nodes are charged by quantity.
- When you scale in a standard data warehouse cluster, you can only modify the same storage specifications as used by the cluster.
- A hybrid data warehouse (cluster mode) cannot be scaled in to a standalone cluster.

Impact on the System

- Before the scale-in, exit 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.
- Before scale-in, ensure that the skew rate does not exceed 10%. There is no general requirement for the dirty page rate. However, for a large table whose size is greater than 50 GB, ensure that the dirty page rate does not exceed 20% to 30%.
- In a cluster that is being scaled in, the following functions are disabled: 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.
- During offline scale-in, if a node is deleted while DDL statements are executed (to create a schema or function), these statements may report errors, because the DN cannot be found. In this case, you simply need to 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.
- 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.

    NOTE

    To check the maximum size of a single table, use the following inspection tool:

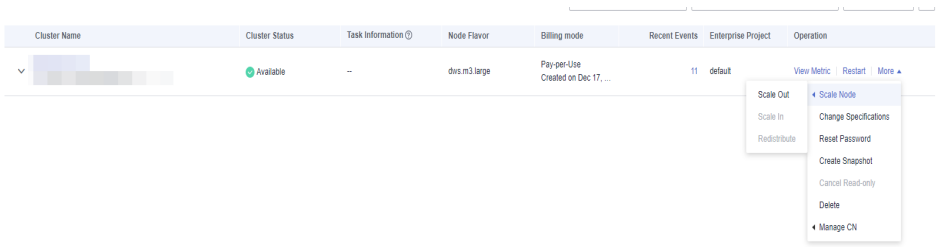
    ```
    gs_check -i CheckBiggestTable -L
    ```
- Automatic removal of faulty CNs is disabled during the scale-in and is enabled after the scale-in is complete.

Procedure

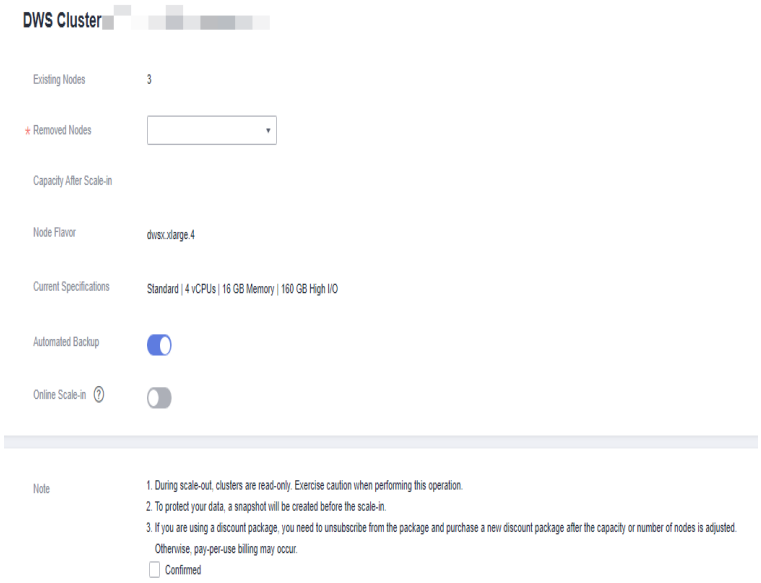
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters** > **Dedicated Clusters**.

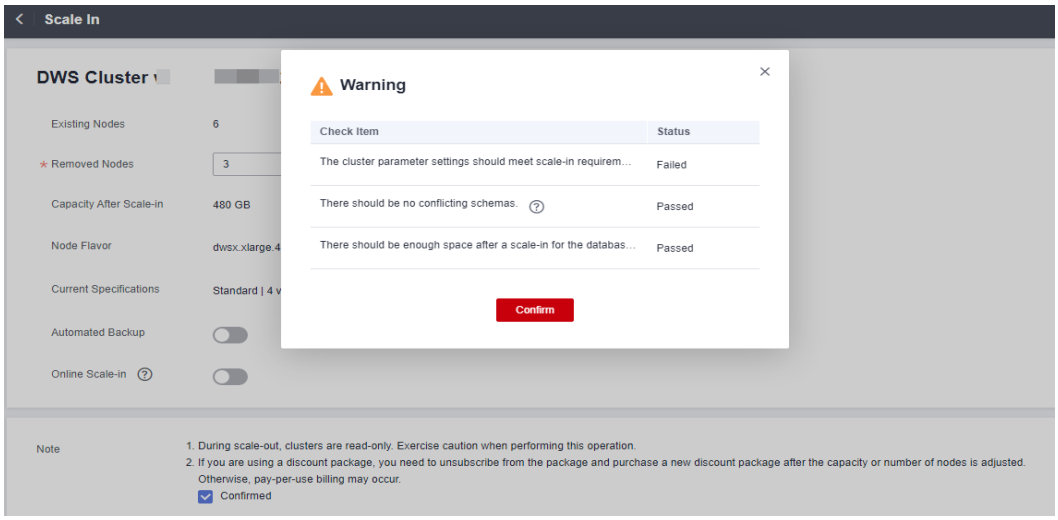
Step 3 In the **Operation** column of the target cluster, choose **More** > **Scale Node** > **Scale In**.



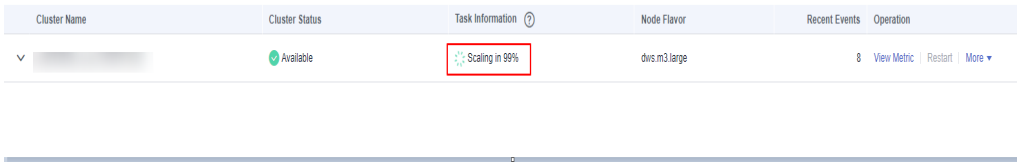
Step 4 The scale-in page is displayed. You can select the number of nodes to be scaled in. The automated backup function is enabled by default.



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.



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.



----End

NOTE

- 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.

6.3 Changing Specifications

6.3.1 Changing the Node Flavor

If you only need to cope with occasional service peaks or only increase computing capabilities, you are advised to modify cluster specifications instead of adding nodes. Before a service peak, you can modify cluster specifications to quickly improve computing capabilities. After the service peak, you can quickly reduce cluster configurations to minimize costs. For more information, see **Supported node flavors**.

NOTE

- Only cluster versions 8.1.1.300 and later support elastic flavor change. For an earlier version, contact technical support to upgrade it first.
- Currently, specifications can be modified only for offline clusters. The modification takes about 10 minutes.

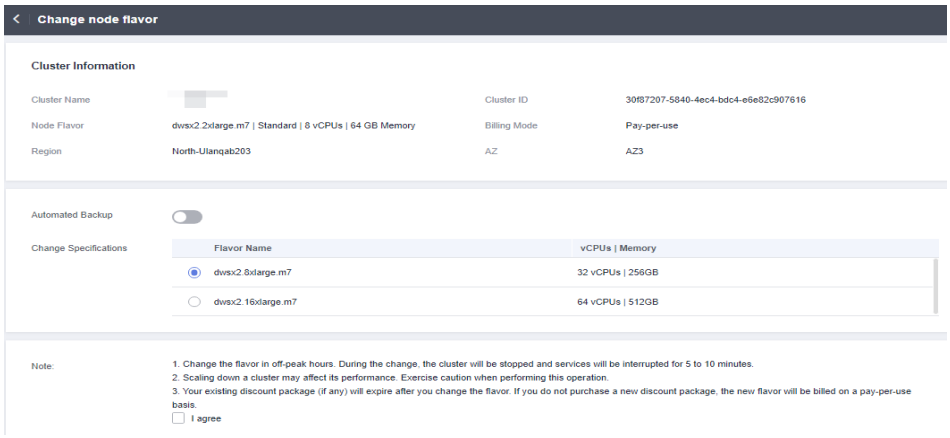
Procedure

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 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.



NOTICE

Decreasing the flavor of a cluster is to select a target flavor that is lower than the current flavor of the cluster. This operation may affect the cluster performance. Therefore, evaluate service impact before performing this operation.

Step 5 Click **Next: confirm**.

Step 6 Return to the cluster list. The cluster status will change to **Changing node flavor**. Wait for about 10 minutes.

----End

Supported Flavors

Table 6-1 Supported node flavors

Current Flavor Name	Target Flavor Name
dwsk2.xlarge	dwsk2.2xlarge, dwsk2.4xlarge, dwsk2.12xlarge, dwsk2.8xlarge
dwsk2.2xlarge	dwsk2.12xlarge, dwsk2.8xlarge, dwsk2.4xlarge
dwsk2.4xlarge	dwsk2.2xlarge, dwsk2.8xlarge, dwsk2.12xlarge
dwsk2.8xlarge	dwsk2.2xlarge, dwsk2.4xlarge, dwsk2.12xlarge
dwsk2.12xlarge	dwsk2.2xlarge, dwsk2.4xlarge, dwsk2.8xlarge
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
dwsk2.h.2xlarge.4.kc1	dwsk2.h.8xlarge.4.kc1, dwsk2.h.12xlarge.4.kc1, dwsk2.h.xlarge.4.kc1, 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, 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, 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, dwsk2.h.12xlarge.4.kc1
dwsk2.h1.12xlarge.4.kc1	dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.8xlarge.4.kc1, dwsk2.h1.2xlarge.4.kc1
dwsk2.h1.2xlarge.4.kc1	dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.8xlarge.4.kc1, dwsk2.h1.12xlarge.4.kc1
dwsk2.h1.4xlarge.4.kc1	dwsk2.h1.8xlarge.4.kc1, dwsk2.h1.12xlarge.4.kc1, dwsk2.h1.2xlarge.4.kc1
dwsk2.h1.8xlarge.4.kc1	dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.12xlarge.4.kc1, dwsk2.h1.2xlarge.4.kc1

Current Flavor Name	Target Flavor Name
dwsk2.h1.xlarge.2.kc1	dwsk2.h1.2xlarge.4.kc1, dwsk2.h1.4xlarge.4.kc1, dwsk2.h1.8xlarge.4.kc1, dwsk2.h1.12xlarge.4.kc1
dwsx2.xlarge	dwsx2.2xlarge, dwsx2.4xlarge, dwsx2.8xlarge, dwsx2.16xlarge
dwsx2.2xlarge	dwsx2.4xlarge, dwsx2.8xlarge, dwsx2.16xlarge
dwsx2.4xlarge	dwsx2.2xlarge, dwsx2.8xlarge, dwsx2.16xlarge
dwsx2.8xlarge	dwsx2.2xlarge, dwsx2.4xlarge, dwsx2.16xlarge
dwsx2.16xlarge	dwsx2.2xlarge, dwsx2.4xlarge, dwsx2.8xlarge
dwsx2.xlarge.m7	dwsx2.2xlarge.m7, dwsx2.4xlarge.m7, dwsx2.8xlarge.m7, dwsx2.16xlarge.m7
dwsx2.2xlarge.m7	dwsx2.4xlarge.m7, dwsx2.8xlarge.m7, dwsx2.16xlarge.m7
dwsx2.4xlarge.m7	dwsx2.2xlarge.m7, dwsx2.8xlarge.m7, dwsx2.16xlarge.m7
dwsx2.8xlarge.m7	dwsx2.2xlarge.m7, dwsx2.4xlarge.m7, dwsx2.16xlarge.m7
dwsx2.16xlarge.m7	dwsx2.2xlarge.m7, dwsx2.4xlarge.m7, dwsx2.8xlarge.m7
dwsx2.xlarge.m7n	dwsx2.2xlarge.m7n, dwsx2.8xlarge.m7n, dwsx2.16xlarge.m7n
dwsx2.2xlarge.m7n	dwsx2.8xlarge.m7n, dwsx2.16xlarge.m7n
dwsx2.8xlarge.m7n	dwsx2.2xlarge.m7n, dwsx2.16xlarge.m7n
dwsx2.16xlarge.m7n	dwsx2.2xlarge.m7n, dwsx2.8xlarge.m7n
dwsx2.h.xlarge.4.c6	dwsx2.h.2xlarge.4.c6, dwsx2.h.4xlarge.4.c6, dwsx2.h.8xlarge.4.c6, dwsx2.h.16xlarge.4.c6
dwsx2.h.2xlarge.4.c6	dwsx2.h.4xlarge.4.c6, dwsx2.h.8xlarge.4.c6, dwsx2.h.16xlarge.4.c6
dwsx2.h.4xlarge.4.c6	dwsx2.h.8xlarge.4.c6, dwsx2.h.16xlarge.4.c6, dwsx2.h.2xlarge.4.c6
dwsx2.h.8xlarge.4.c6	dwsx2.h.4xlarge.4.c6, dwsx2.h.16xlarge.4.c6, dwsx2.h.2xlarge.4.c6
dwsx2.h.16xlarge.4.c6	dwsx2.h.2xlarge.4.c6, dwsx2.h.4xlarge.4.c6, 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, dwsx2.h.2xlarge.4.c7

Current Flavor Name	Target Flavor Name
dwsx2.h.2xlarge.4.c7	dwsx2.h.4xlarge.4.c7, dwsx2.h.8xlarge.4.c7, dwsx2.h.16xlarge.4.c7
dwsx2.h.4xlarge.4.c7	dwsx2.h.2xlarge.4.c7, dwsx2.h.8xlarge.4.c7, dwsx2.h.16xlarge.4.c7
dwsx2.h.8xlarge.4.c7	dwsx2.h.16xlarge.4.c7, dwsx2.h.2xlarge.4.c7, 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, 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, dwsx2.h.16xlarge.4.c7n
dwsx2.h.2xlarge.4.c7n	dwsx2.h.4xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, dwsx2.h.16xlarge.4.c7n
dwsx2.h.4xlarge.4.c7n	dwsx2.h.2xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, dwsx2.h.16xlarge.4.c7n
dwsx2.h.8xlarge.4.c7n	dwsx2.h.16xlarge.4.c7n, dwsx2.h.2xlarge.4.c7n, dwsx2.h.4xlarge.4.c7n
dwsx2.h.16xlarge.4.c7n	dwsx2.h.4xlarge.4.c7n, dwsx2.h.8xlarge.4.c7n, 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, dwsx2.h1.4xlarge.4.c6
dwsx2.h1.2xlarge.4.c6	dwsx2.h1.4xlarge.4.c6, dwsx2.h1.8xlarge.4.c6, dwsx2.h1.16xlarge.4.c6
dwsx2.h1.4xlarge.4.c6	dwsx2.h1.2xlarge.4.c6, dwsx2.h1.8xlarge.4.c6, dwsx2.h1.16xlarge.4.c6
dwsx2.h1.8xlarge.4.c6	dwsx2.h1.16xlarge.4.c6, dwsx2.h1.4xlarge.4.c6, dwsx2.h1.2xlarge.4.c6
dwsx2.h1.16xlarge.4.c6	dwsx2.h1.4xlarge.4.c6, dwsx2.h1.2xlarge.4.c6, dwsx2.h1.8xlarge.4.c6
dwsx2.h1.xlarge.2.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, dwsx2.h1.16xlarge.4.c7, dwsx2.h1.2xlarge.4.c7
dwsx2.h1.16xlarge.4.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, dwsx2.h1.2xlarge.4.c7
dwsx2.h1.2xlarge.4.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, dwsx2.h1.16xlarge.4.c7
dwsx2.h1.4xlarge.4.c7	dwsx2.h1.2xlarge.4.c7, dwsx2.h1.8xlarge.4.c7, dwsx2.h1.16xlarge.4.c7
dwsx2.h1.8xlarge.4.c7	dwsx2.h1.4xlarge.4.c7, dwsx2.h1.2xlarge.4.c7, dwsx2.h1.16xlarge.4.c7

Current Flavor Name	Target Flavor Name
dwsx2.h1.xlarge.2.c7n	dwsx2.h1.2xlarge.4.c7n, dwsx2.h1.4xlarge.4.c7n, dwsx2.h1.8xlarge.4.c7n, dwsx2.h1.16xlarge.4.c7n
dwsx2.h1.2xlarge.4.c7n	dwsx2.h1.16xlarge.4.c7n, dwsx2.h1.4xlarge.4.c7n, dwsx2.h1.8xlarge.4.c7n
dwsx2.h1.4xlarge.4.c7n	dwsx2.h1.8xlarge.4.c7n, dwsx2.h1.16xlarge.4.c7n, dwsx2.h1.2xlarge.4.c7n
dwsx2.h1.8xlarge.4.c7n	dwsx2.h1.4xlarge.4.c7n, dwsx2.h1.16xlarge.4.c7n, dwsx2.h1.2xlarge.4.c7n
dwsx2.h1.16xlarge.4.c7n	dwsx2.h1.2xlarge.4.c7n, dwsx2.h1.4xlarge.4.c7n, dwsx2.h1.8xlarge.4.c7n
dwsx2.rt.xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.4xlarge.m7, dwsx2.rt.8xlarge.m7, dwsx2.rt.16xlarge.m7
dwsx2.rt.2xlarge.m7	dwsx2.rt.4xlarge.m7, dwsx2.rt.8xlarge.m7, dwsx2.rt.16xlarge.m7
dwsx2.rt.4xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.8xlarge.m7, dwsx2.rt.16xlarge.m7
dwsx2.rt.8xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.4xlarge.m7, dwsx2.rt.16xlarge.m7
dwsx2.rt.16xlarge.m7	dwsx2.rt.2xlarge.m7, dwsx2.rt.4xlarge.m7, dwsx2.rt.8xlarge.m7
dwsk2.rt.xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.4xlarge.km1, dwsk2.rt.8xlarge.km1, dwsk2.rt.12xlarge.km1
dwsk2.rt.2xlarge.km1	dwsk2.rt.4xlarge.km1, dwsk2.rt.8xlarge.km1, dwsk2.rt.12xlarge.km1
dwsk2.rt.4xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.8xlarge.km1, dwsk2.rt.12xlarge.km1
dwsk2.rt.8xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.4xlarge.km1, dwsk2.rt.12xlarge.km1
dwsk2.rt.12xlarge.km1	dwsk2.rt.2xlarge.km1, dwsk2.rt.4xlarge.km1, dwsk2.rt.8xlarge.km1
dwsx2.rt.xlarge.m7n	dwsx2.rt.2xlarge.m7n, dwsx2.rt.8xlarge.m7n, dwsx2.rt.16xlarge.m7n
dwsx2.rt.2xlarge.m7n	dwsx2.rt.8xlarge.m7n, dwsx2.rt.16xlarge.m7n
dwsx2.rt.8xlarge.m7n	dwsx2.rt.2xlarge.m7n, dwsx2.rt.16xlarge.m7n
dwsx2.rt.16xlarge.m7n	dwsx2.rt.2xlarge.m7n, dwsx2.rt.8xlarge.m7n

6.3.2 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.
- Currently, the stream cluster does not support changing all specifications.
- Currently, GaussDB(DWS) 3.0 clusters do 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) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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**.

- For the **Node Flavor** parameter, select a flavor.

  NOTE

  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.

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

6.3.3 Disk Capacity Expansion of an EVS Cluster

Context

In conventional scaling, compute and storage resources are coupled. If a company scales out disks, it has to add unnecessary CPUs and memory at the same time. The scaling takes a long time and interrupts services. Disk capacity expansion can quickly increase storage without service interruption. You can increase disk space without having to stop services.

NOTE

- Disk capacity expansion can be performed only for standard data warehouses using SSD, hybrid data warehouses, or stream data warehouses. 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

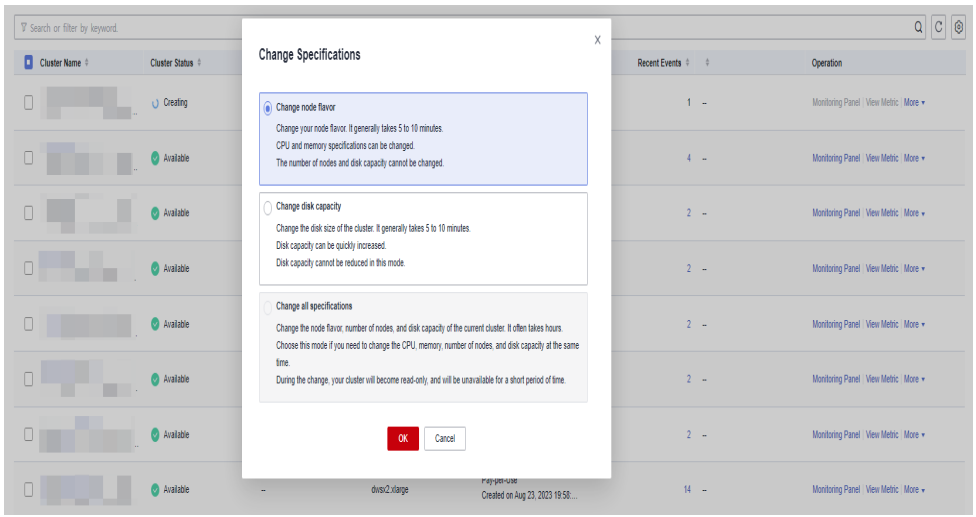
- Hot storage disks cannot be scaled down.
- Scale up hot data storage during off-peak hours.
- If the cluster is in the read-only state, a message will be displayed after you click **Expand Disk Capacity**. After you start expansion, wait until it is completed and the cluster changes to the available state.

Procedure

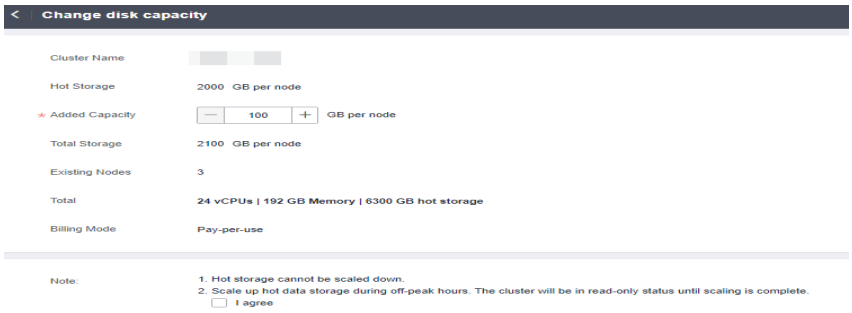
Step 1 Log in to the GaussDB(DWS) management console.

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**.



NOTE

Hot Storage is changed to **Hot Storage (with Cache)** for GaussDB(DWS) 3.0 clusters.

Step 5 Confirm the settings and click **Submit**.

Step 6 Return to the cluster list and check the disk capacity expansion progress.

----End

7 Backup and Disaster Recovery

7.1 Snapshots

7.1.1 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

NOTE

- 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 as per 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.
  - 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.
- The hybrid data warehouse (standalone) does not support snapshots.
- Only GaussDB(DWS) 3.0 clusters of 9.0.2 and later versions support the snapshot function.
- 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%.

7.1.2 Manual Snapshots

7.1.2.1 Creating a Manual Snapshot

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.
- 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) management console.

Step 2 In the navigation pane, choose **Snapshots**. Click **Create Snapshot** in the upper right corner. Alternatively, choose **More** > **Create Snapshot** in the **Operation** column.

Step 3 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 4 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

7.1.2.2 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

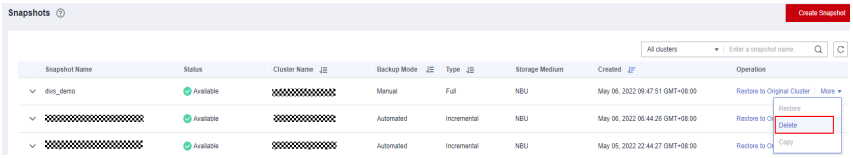
Deleted snapshots cannot be recovered. Exercise caution when performing this operation.

Procedure

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

Step 2 In the navigation pane, choose **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

7.1.3 Automated Snapshots

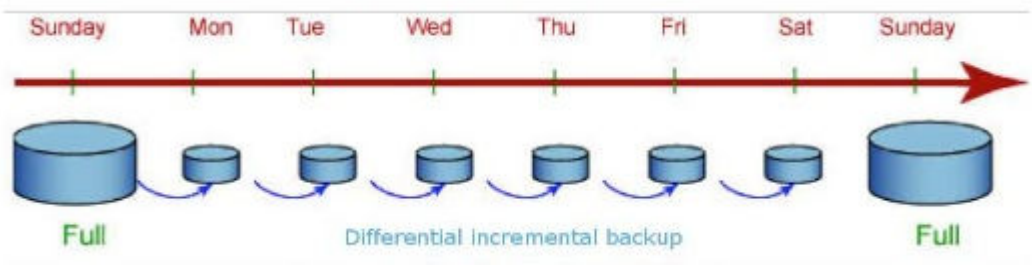
7.1.3.1 Automatic Snapshot Overview

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, when the deletion of the snapshot will not hinder incremental data backup and restoration.

Figure 7-1 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. 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 3 days. The system deletes the snapshot at the end of the retention period. 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**.

7.1.3.2 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) management console.

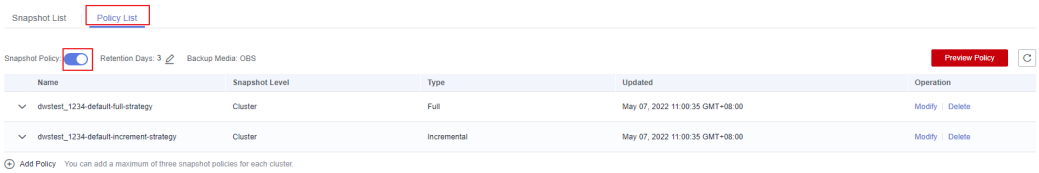
Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

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 **Policy List**. All policies of the current cluster are displayed on the **Policy List** page. Toggle on **Snapshot Policy**.

-  indicates that the policy is enabled (default). The default retention period is three days.
-  indicates that the automatic snapshot function is disabled.

Figure 7-2 Policy list



Step 5 After this function is enabled, you can set the retention mode for automated snapshots. For more information, see **Table 7-1**.

Table 7-1 Automated snapshot parameters

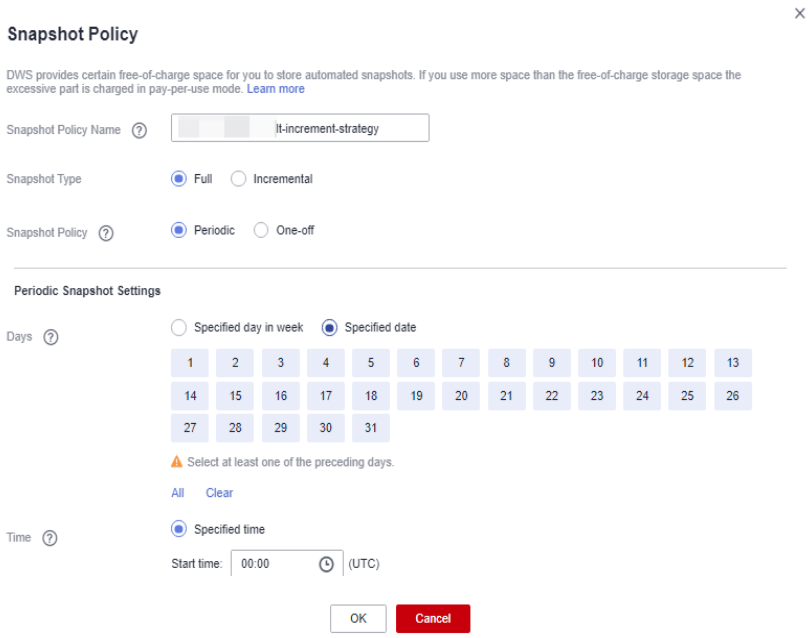
Parameter	Description
Retention Days	Retention days of the snapshots that are automatically created. The value ranges from 1 to 31 days. **NOTE** Snapshots that are automatically created cannot be deleted manually. The system automatically deletes these snapshots when their retention duration exceeds the threshold.

Step 6 After automated snapshot is enabled, you can configure its parameters. For more information, see **Table 7-2**.

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-time**, as shown in the following figures.
  - **Periodic**: Specify the days for every week/month and the exact time on the days.



WARNING

Choosing the days in red (29th/30th/31st) may skip some monthly backups.

- **One-time**: Specify a day and the exact time on the day.

- Incremental snapshots can be set only to **Periodic**, as shown in the first figure below.
  - 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.

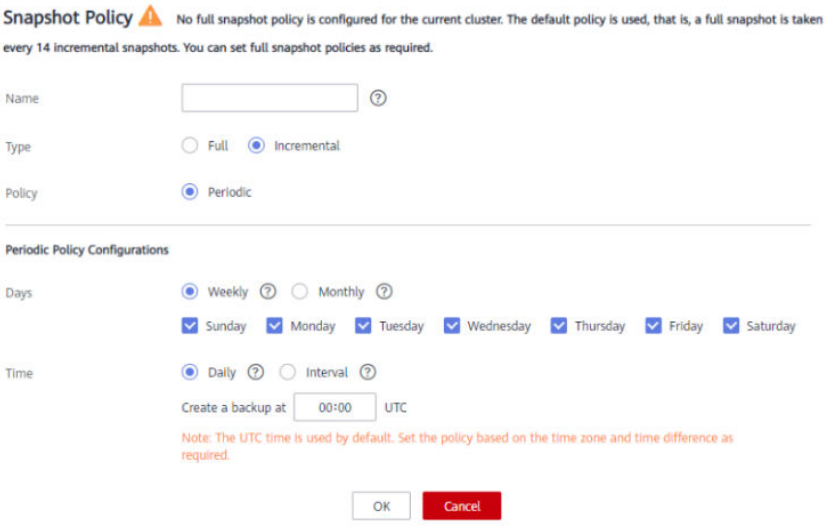


Table 7-2 Snapshot policy parameters

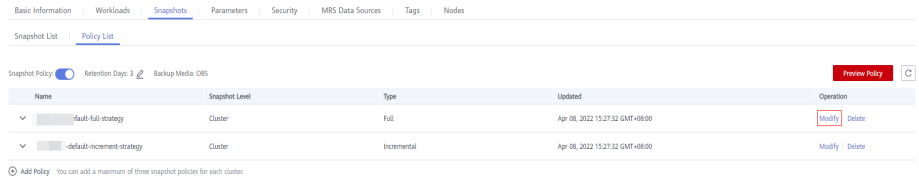
Parameter	Description
Name	The policy name must be unique, consist of 4 to 92 characters, and start with a letter. It is case-insensitive and can contain only letters, digits, hyphens (-), and underscores (_).
Type	You can choose either full or incremental snapshots. **NOTE** • A full snapshot is created 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** If the incremental data is large and the execution period is long, the backup will be slow. In this case, increase the backup frequency.

Step 7 Click **OK**.

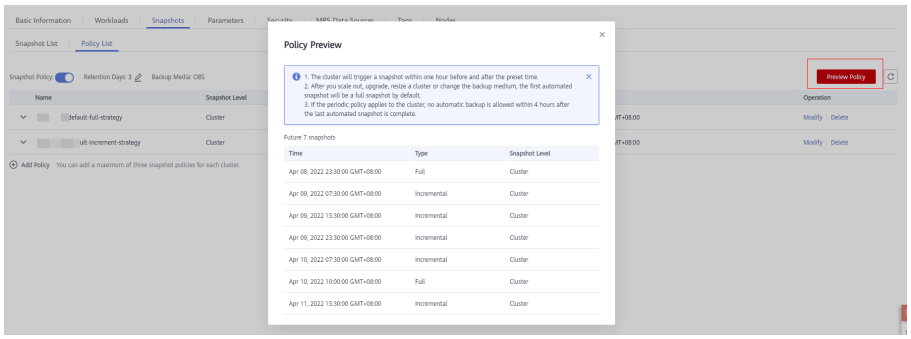
NOTE

A maximum of three snapshot policies can be set for a cluster.

Step 8 (Optional) To modify an automated snapshot policy, click **Modify** in the **Operation** column.



Step 9 (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 14 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

7.1.3.3 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) management console.

Step 2 In the navigation pane, choose **Snapshots**.

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**.

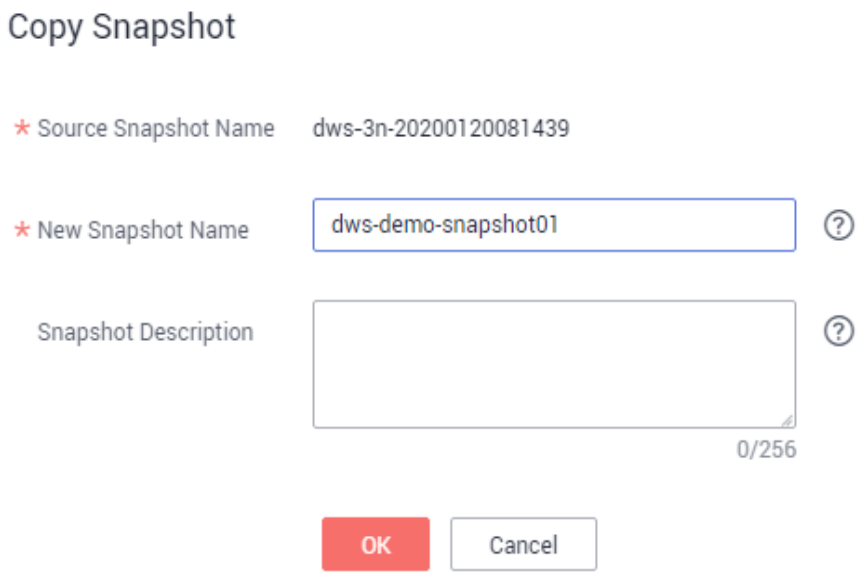
- **New Snapshot Name**: Enter a new snapshot name.

  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: !<>'=&"

Figure 7-3 Copying a snapshot



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

7.1.3.4 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.

7.1.4 Viewing Snapshot Information

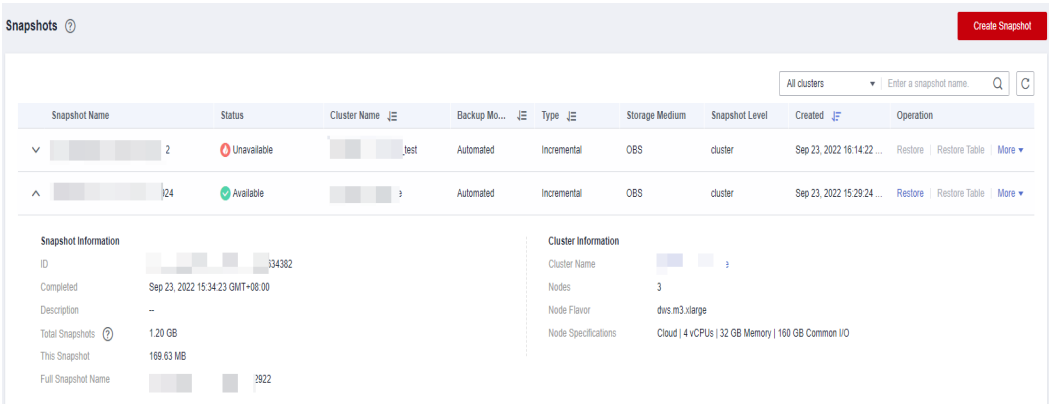
This section describes how to view snapshot information on the **Snapshots** page.

Viewing Snapshot Information

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

Step 2 In the navigation pane on the left, choose **Snapshots**.

In the snapshot list, all snapshots are displayed by default. Click next to the snapshot name to check the snapshot details.



Step 3 You can view the **Snapshot Name**, **Snapshot Status**, **Cluster Name**, **Backup Mode**, **Snapshot Type**, **Storage Medium**, 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 7-3 describes snapshot status.

Table 7-3 Snapshot status

Status	Description
Available	Indicates that the existing snapshot works properly.

Status	Description
Creating	Indicates that a snapshot is being created.
Unavailable	Indicates that the existing snapshot cannot provide services.

The following table describes the backup modes.

Table 7-4 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.

The following table describes the snapshot types.

Table 7-5 Type

Type	Description
Full	The snapshot is a full backup.
Incremental	The snapshot is an incremental backup.

The following table describes the snapshot media.

Table 7-6 Storage media

Storage Medium	Description
OBS	The created snapshot is an OBS snapshot and the backup data is stored on the OBS server.

----End

7.1.5 Restoration Using a Snapshot

7.1.5.1 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.
- The parameters after cluster-level restoration are the same as those before backup. When restoring data to a new cluster, ensure that the flavor of the new cluster is the same as that of the original cluster. If the flavor of the new cluster is smaller, the restoration may fail.

7.1.5.2 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

- Currently, you can only use the snapshots stored in OBS to restore data to a new cluster.
- 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.
- If cold and hot tables are used, snapshots cannot be used to restore cold data to a new 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

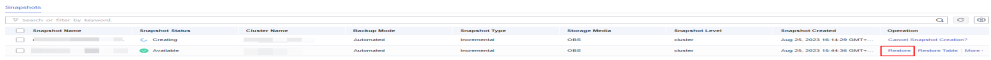
- The resources required for restoring data to a new cluster do not exceed your available resource quota.
- The snapshot is in the **Available** state.

Procedure

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

Step 2 In the navigation pane, choose **Snapshots**. All snapshots are displayed by default.

Step 3 In the **Operation** column of a snapshot, click **Restore**.

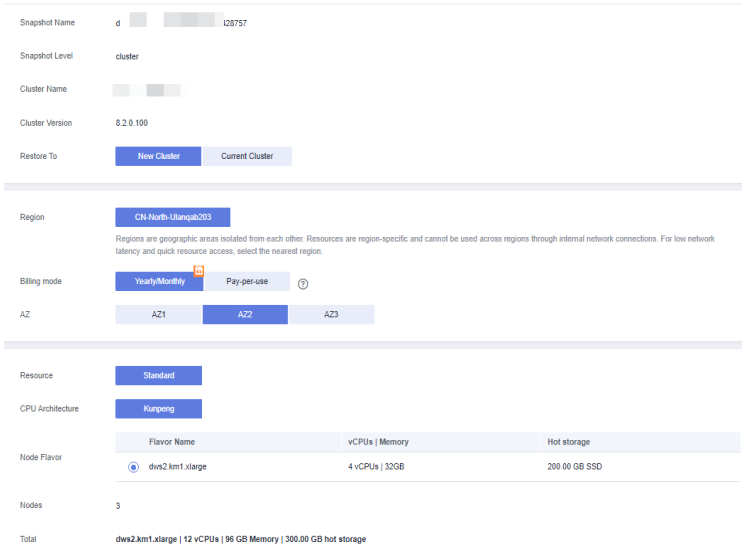


Step 4 On the **Restore Snapshot** page, configure the parameters of the new cluster, as shown in the following figure.

1. Restore to a single-AZ cluster.
2. Restore to a multi-AZ cluster.

   NOTE

   - Only clusters later than 8.2.0.100 can be restored to a multi-AZ cluster.
   - Currently, the multi-AZ cluster supports only GaussDB(DWS) 2.0 standard data warehouses.
   - 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 7-7**. By default, other parameters are the same as those in the snapshot. For details, see **Table 7-2**.

Table 7-7 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
Advanced settings	If **Custom** is selected, configure the following parameters: • **Tag**: If encryption is enabled for the original cluster, you can configure a key name.

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.

For details about quotas, see **What Is the User Quota?**

----End

7.1.5.3 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.

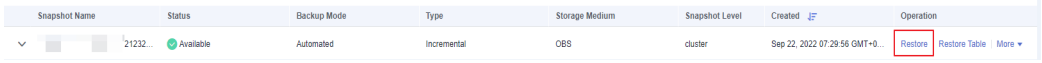
Procedure

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

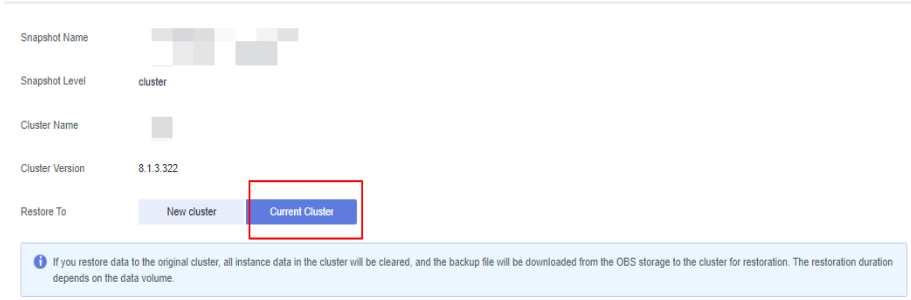
Step 2 In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 3 Click the name of a cluster and choose **Snapshots**.

Step 4 Click **Restore**.



Step 5 Restore the snapshot to the current cluster.



NOTE

If you use a snapshot to restore data to the original cluster, the cluster will be unavailable during the restoration.

----End

7.1.6 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

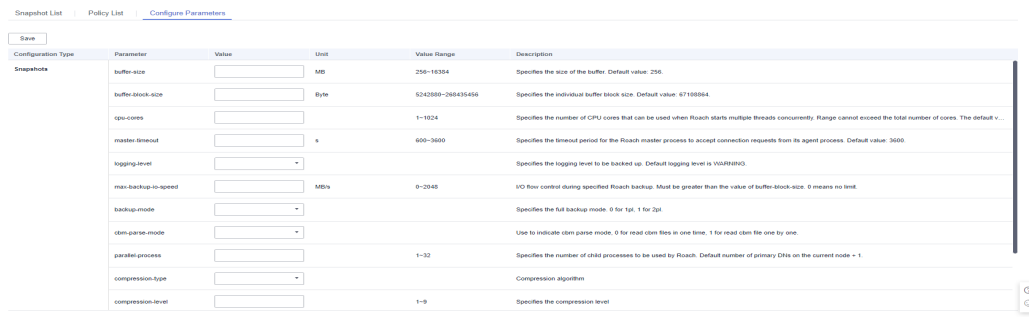
Step 1 Log in to the GaussDB(DWS) management console.

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 7-8**.



Step 6 Click **Save**.

----End

Snapshot parameters

Table 7-8 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.
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

Parameter	Type	Description	Default Value
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
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.

Parameter	Type	Description	Default Value
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
restore-by-insert	Restoration parameter	Fine-grained restoration mode. If this parameter is specified for a fine-grained restoration, the **INSERT** statement will be used to restore the target table. Otherwise, the **ALTER** statement will be used.	ALTER

7.1.7 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

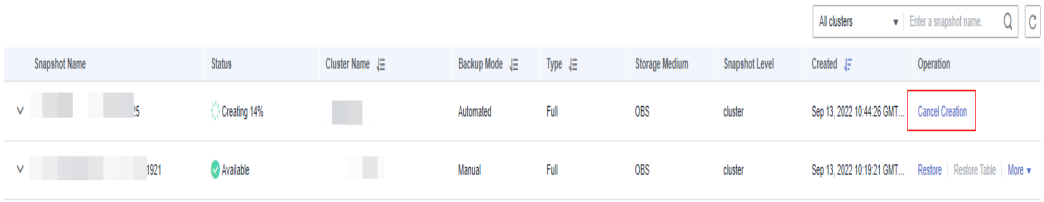
Only the snapshots in the **Creating** state can be stopped. A snapshot creation task that just started or is about to complete cannot be stopped.

Procedure

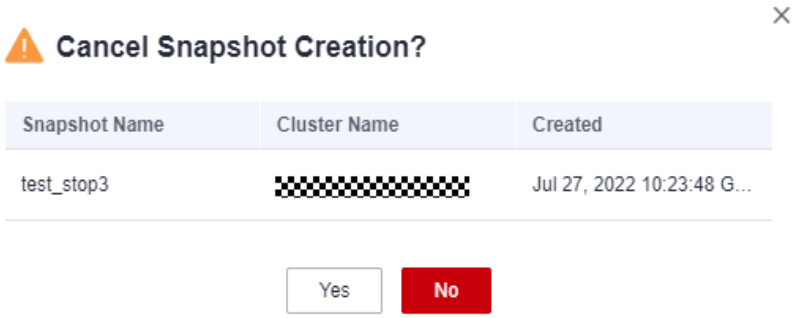
Step 1 Log in to the GaussDB(DWS) management console.

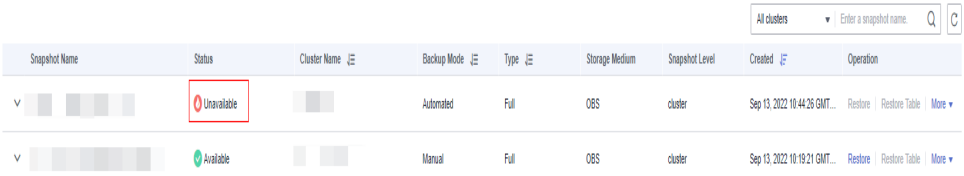
Step 2 In the navigation pane on the left, choose **Snapshots**.

In the **Operation** column of a snapshot that is being created, and click **Cancel Creation**.



Step 3 In the dialog box that is displayed, click **Yes** to stop the snapshot. The snapshot state will change to **Unavailable**.





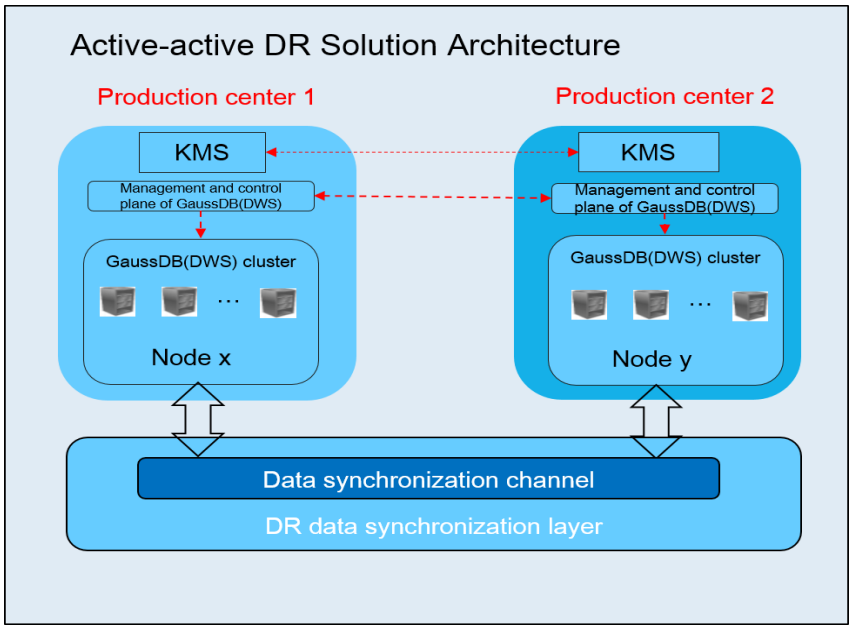
----End

7.2 Cluster DR

7.2.1 DR Overview

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.



NOTE

- Intra-region DR is supported only in cluster version 8.1.1 and later.
- The hybrid data warehouse (standalone) does not support disaster recovery.
- GaussDB(DWS) 3.0 clusters and multi-AZ clusters do not support the DR function.

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.

7.2.2 Creating a DR Task

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) management console.

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

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.

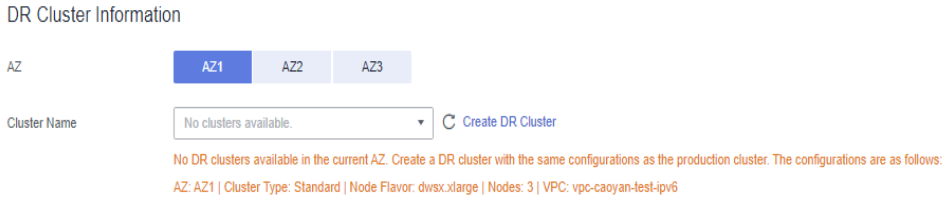
Step 6 Configure the DR cluster.

- 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.

- After you select an AZ for the DR cluster, homogeneous DR clusters will be displayed. If no DR cluster is available, create a cluster with the same configurations 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

7.2.3 Viewing DR Information

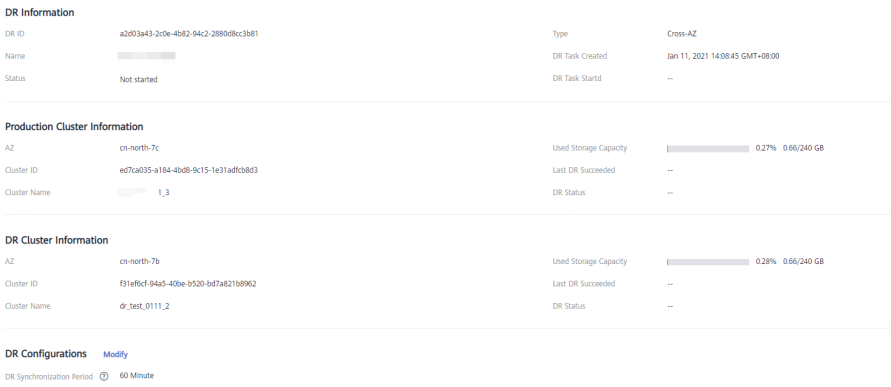
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 In the navigation pane on the left, choose **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 creation 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**: Users can view and modify the DR synchronization period.



----End

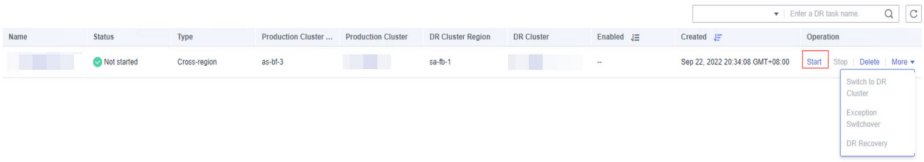
7.2.4 DR Management

Starting a DR Task

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

Step 2 In the navigation pane on the left, choose **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

Stopping the DR Task

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

Step 2 In the navigation pane on the left, choose **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

Switching to the DR Cluster

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

Step 2 In the navigation pane on the left, choose **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.
- Recovery Point Object (RPO) refers to the point in time to which a system and data must be restored after a disaster occurs. Its value varies by cluster status.
  - 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

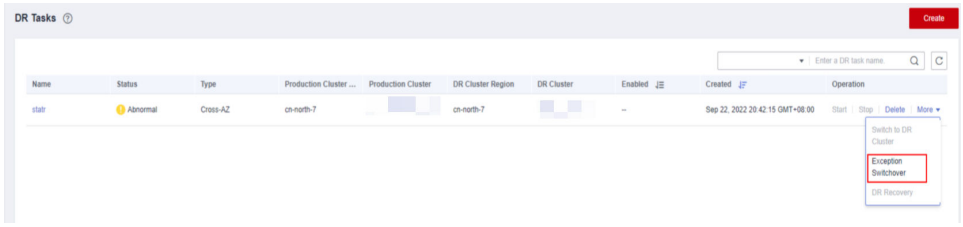
The production cluster is unavailable, the DR cluster is normal, and the DR status is **Abnormal**.

Procedure

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

Step 2 In the navigation pane on the left, choose **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

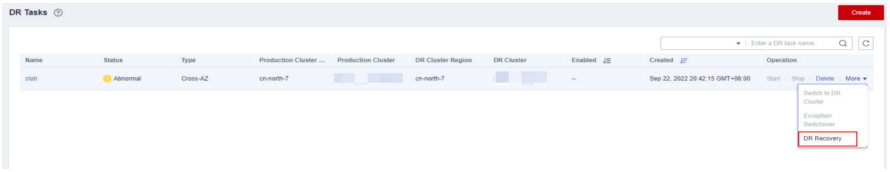
After abnormal switchover, if you have confirmed that the original production cluster was recovered, you can perform a switchback.

Procedure

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

Step 2 In the navigation pane on the left, choose **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

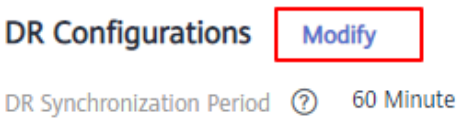
Updating DR Configurations

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

Step 2 In the navigation pane on the left, choose **DR Tasks**.

Step 3 In the DR list, click the DR name to go to the DR information page.

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.

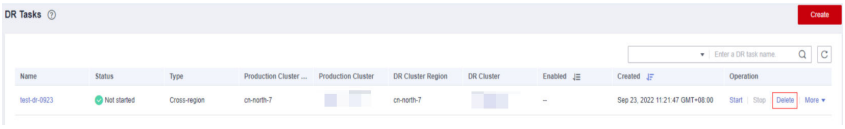
----End

Deleting DR Tasks

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

Step 2 In the navigation pane on the left, choose **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

7.2.5 Mutually Exclusive DR Cases

Case 1: How Do I Scale out a Cluster in the DR State?

Step 1 Log in to the GaussDB(DWS) management 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 If the **Task Information** is other than **DR not started**, delete the DR task. For details, see **Deleting DR Tasks**.

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 a DR Task**.

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

8 Intelligent O&M

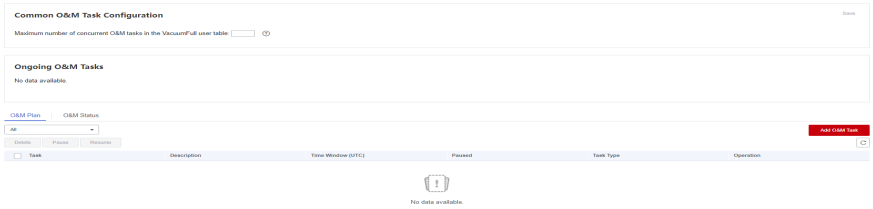
8.1 Overview

Intelligent O&M helps GaussDB(DWS) users with O&M tasks. With this feature, you can specify the proper time window and number of tasks to execute based on the cluster workload. Besides, Intelligent O&M can adjust task execution policies according to service changes in a timely manner to reduce the impact on 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.

As shown in the figure below, the **Intelligent O&M** page consists of the following parts:

- Common configuration of O&M tasks: Currently, you can only configure **Maximum number of concurrent O&M tasks in the VacuumFull user table**. This configuration takes effect on all the VACUUM FULL tasks of user tables.
- 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.).
  - 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.



NOTE

- This feature is supported only in 8.1.3 or later.
- The intelligent O&M function is not supported in hybrid data warehouses (standalone mode).
- 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.

8.2 O&M Plans

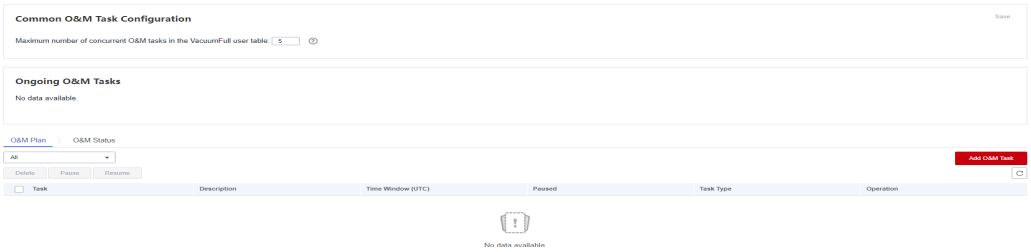
Setting the Common Configurations of O&M Tasks

Step 1 Log in to the GaussDB(DWS) management 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

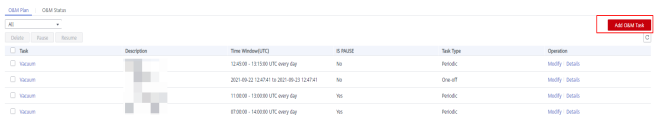
Adding an O&M Plan

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

Step 2 Click the name of the target cluster.

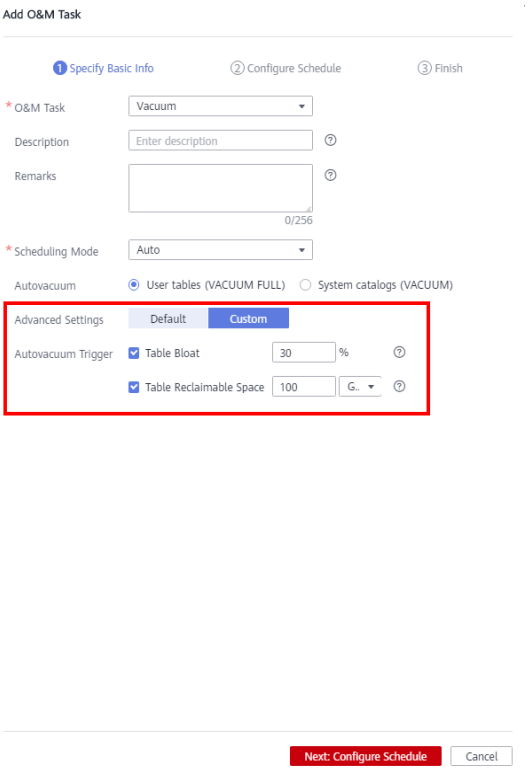
Step 3 In the navigation pane, choose **Intelligent O&M**.

Step 4 In the **O&M Plan** area, click **Add O&M Task**.

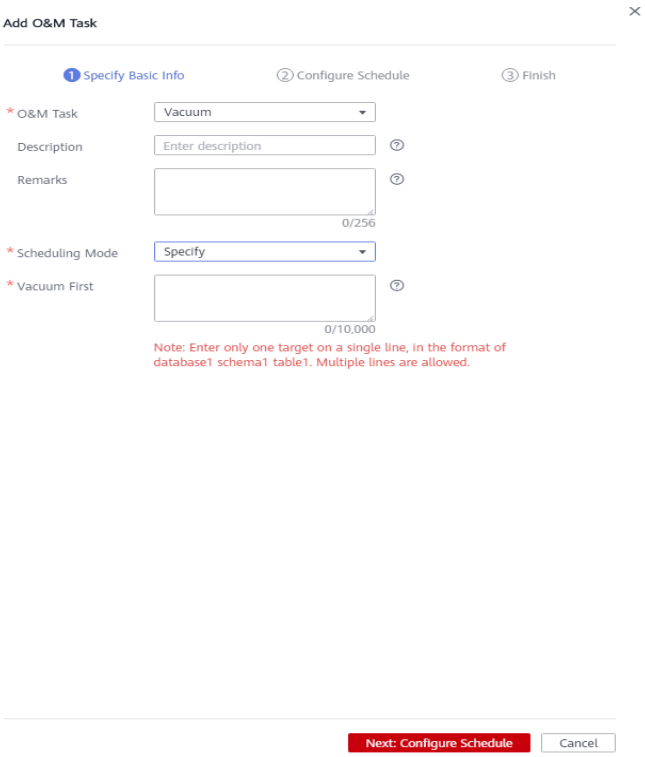


Step 5 On the right panel displayed, configure the O&M task.

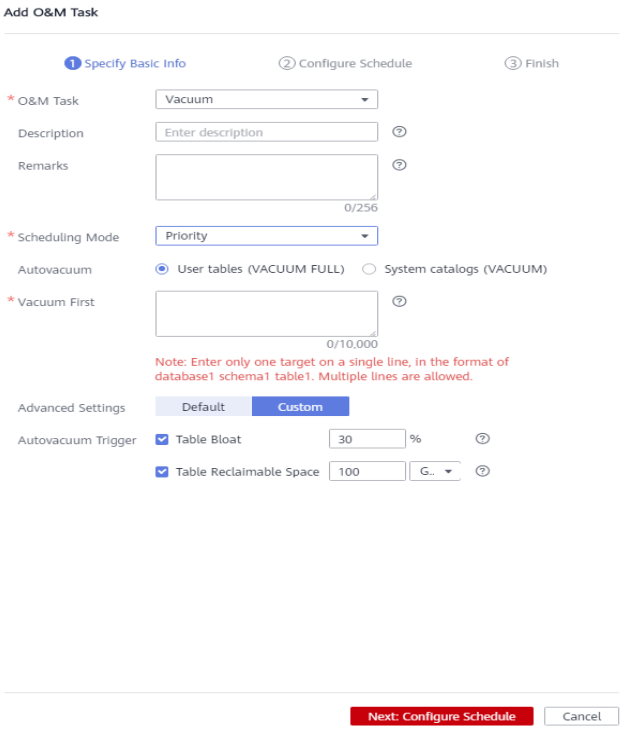
- **O&M Task**: Only **Vacuum** is currently supported.
- Description: This O&M task helps users periodically run the **VACUUM** command to free up space.
- **Remarks**
- **Scheduling Mode**
  - **Auto**: Intelligent O&M scans the database in a specified time window, and automatically delivers table-level 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.



> **CAUTION**
>
> 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.

- **Autovacuum**: **System catalogs (VACUUM)** or **User tables (VACUUM FULL)**.
  - 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.

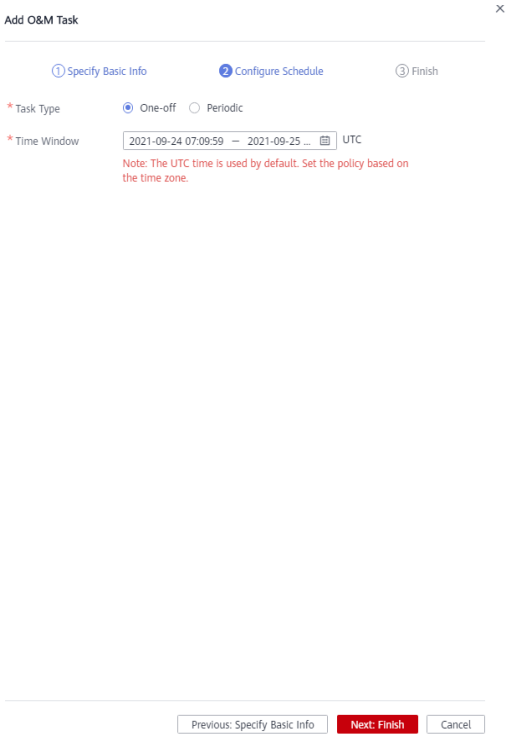
- **Vacuum First**: Configure the preferential vacuum targets. Enter only one target on a single line, in the format of *database1 schema1 table1*. Separate the names with spaces on each line and multiple lines are allowed.
- **Advanced**: If you select **Custom**, you can configure the autovacuum triggers, including the table bloat and table reclaimable space.

  If you select **Default**, **Table Bloat** defaults to **80%** and **Table Reclaimable Space** defaults to **100 GB**.

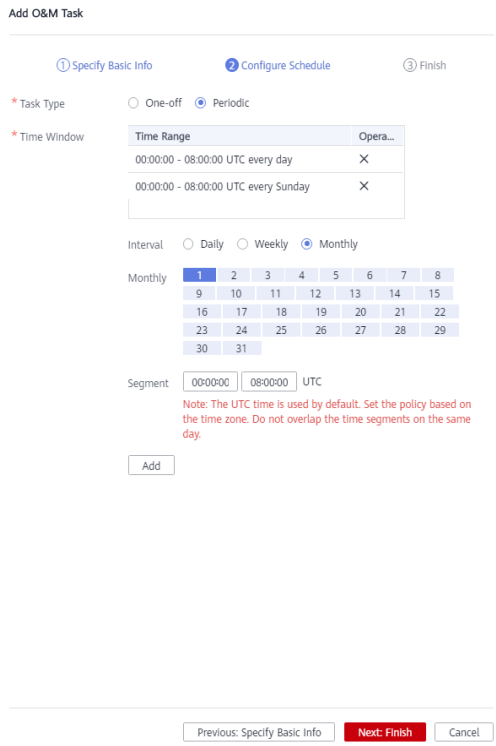
  > **NOTE**
  >
  > VACUUM bloat rate: After frequent UPDATE and DELETE operations are performed in a database, the deleted or updated rows are logically deleted from the database, but actually still exist in tables. Before VACUUM is complete, such data is still stored in disks, causing table bloat. If the bloat rate reaches the percentage threshold set in an O&M task, VACUUM will be automatically triggered.

Step 6 Click **Next: Configure Schedule** and configure the O&M task schedule.

- **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

- When configuring the time window for autovacuum O&M tasks, avoid service peak hours. Otherwise, deadlocks may occur between autovacuum tasks and 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

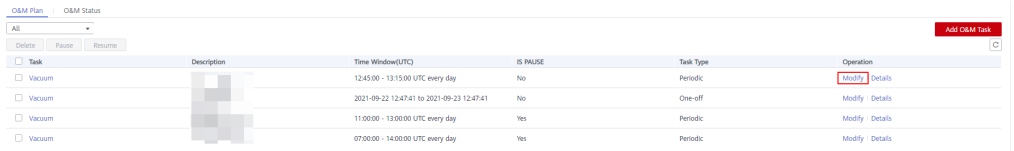
Modifying an O&M Plan

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

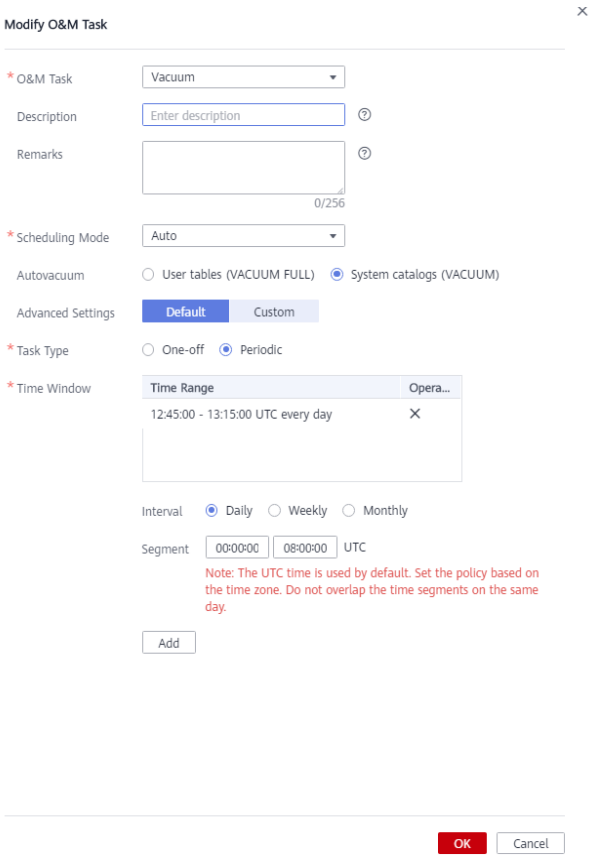
Step 2 Click the name of the target cluster.

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.



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

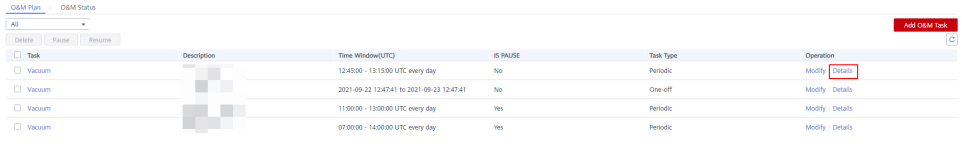
Viewing O&M Task Details

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



Step 5 The **O&M Task Details** panel is displayed for you to check the information.

----End

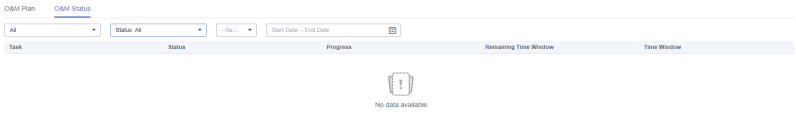
8.3 O&M Status

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



Step 5 Click the name of a specified O&M task to view the status details.

- **O&M Task**: **Vacuum**
- **Status**:
  - Waiting
  - Running
  - Finished
  - Canceled
- **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 Cluster Management

9.1 Modifying Database Parameters

After a cluster is created, you can modify the cluster's database parameters as required. On the GaussDB(DWS) management 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**.

Prerequisites

You can modify parameters only when no task is running in the cluster.

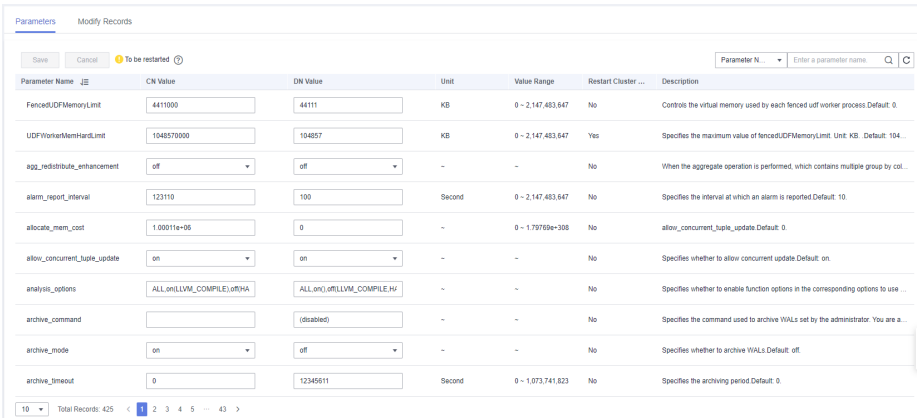
Modifying Parameters

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

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

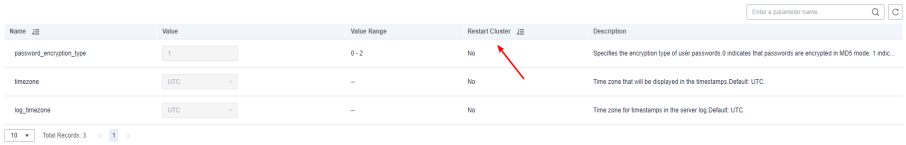
Step 3 In the cluster list, find the target cluster and click the cluster name. The **Cluster Information** page is displayed.

Step 4 Click the **Parameters** tab and modify the parameter values. Then click **Save**.



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.



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:

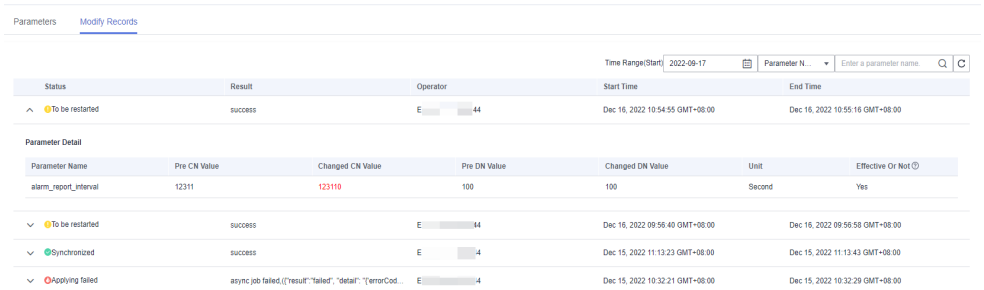
Procedure

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

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

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

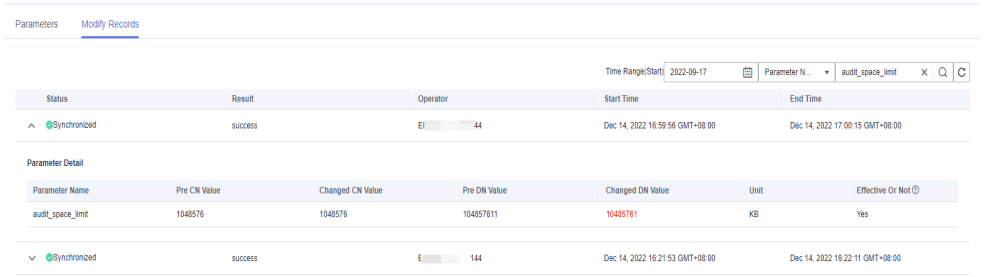
Step 4 Click the **Modify Records** tab.



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

Parameter Description

The following table describes part of the database parameters. You can search for and check more parameters by following the instructions in **Modifying Parameters**.

NOTE

The default values of the following parameters are for reference only. For more information, see **Setting GUC Parameters**.

9.2 Checking the Cluster Status

On the **Clusters** > **Dedicated Clusters** page of the GaussDB(DWS) management 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) management console. In the navigation tree on the left, 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.

Enter the cluster name in the search box, and click to search for a cluster. Alternatively, in the **All projects** drop-down list above the cluster list, select the target project. Click to refresh the cluster list and billing mode. You can also click **Search by Tag** to search for clusters based on the criteria. 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 9-1** describes the cluster list parameters.

Table 9-1 Cluster list parameters

Parameter	Description
Cluster Name	Cluster name specified when a cluster is created. **NOTE** After a cluster is created, its name cannot be changed.
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.
Billing Mode	Cluster billing mode. • In pay-per-use mode, the cluster creation time is displayed.
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 DAS to Connect to a Cluster**. • **Monitoring Panel**: For details, see **Databases Monitoring (DMS)**. • **More** – **View Metric**: For details, see **Monitoring Clusters Using Cloud Eye**. – **Restart**: Click **Restart** to restart a cluster. For details, see **Cluster Restart**. – **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**. – **View Scaling Details**: For details, see **Viewing Redistribution Details**. – **Expand Disk Capacity**: For details, see **Disk Capacity Expansion of an EVS Cluster**. – **Reset Password**: For details, see **Resetting a Password**. – **Create Snapshot**: For details, see **Manual Snapshots**. – **Cancel Readonly**: For details, see **Removing the Read-only Status**. – **Delete**: Click **Delete** to delete a cluster. For details, see **Deleting a Cluster**. – **Change node flavor**: For details, see **Changing the Node Flavor**. – **Manage CN**: For details, see **Managing CNs**.

Cluster Status

Table 9-2 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, 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	A cluster goes into this state when it has been frozen. 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**.

Cluster Task Information

Table 9-3 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.
Scaling out	Indicates that a cluster is being scaled out.
Scale-out failed	Indicates that a cluster fails to be scaled out.

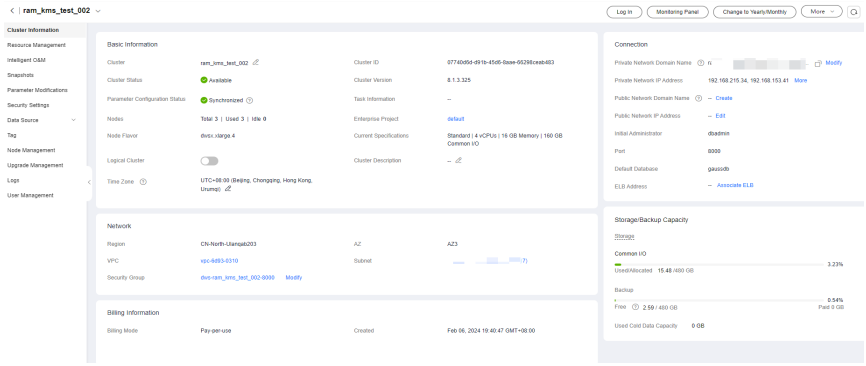
Status	Description
Expanding disk capacity	Indicates that disk capacity is being expanded.
Disk expansion failed	Indicates that disk capacity fails to be expanded.
Associating ELB	Indicates that ELB is being associated.
Failed to associate ELB	Indicates that ELB fails to be associated.
Disassociating 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. • 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.

9.3 Viewing Cluster Details

Log in to the GaussDB(DWS) management 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 9-1 Cluster Details



On the **Cluster Information** page, you can view the following information:

- **Basic Information**: **Table 9-4** lists the related parameters.
- **Connection**: **Table 9-5** describes the parameters.
- **Network**: **Table 9-6** lists the related parameters.
- **Billing Information**: **Table 9-7** describes the parameters.
- **Storage/Backup Capacity**: **Table 9-8** describes the parameters.
- **O&M Account**: **Table 9-9** describes the related parameters.
- **Data Encryption Information**: **Table 9-10** lists the related parameters.

  NOTE

  You can view this module if you enable the data encryption function when creating a cluster.

Table 9-4 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.

Parameter	Description
Cluster ID	ID of the cluster.
Cluster Version	Cluster version information.
Created	Time when the cluster was created.
Node Flavor	Node flavor of the cluster.
Maintenance Window	Maintenance window 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 Database Parameters**.

Table 9-5 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. **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 Access Domain Names**.

Parameter	Description
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 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.
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 9-6 Network

Parameter	Description
Region	Current working zone of the cluster.
AZ	AZ selected during cluster creation.

Parameter	Description
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.
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.
Security Group	Security group selected during cluster creation. The security group of a cluster cannot be changed but can be modified. Modifying an existing security group rule: Click the security group name to go to the security group details page.

Table 9-7 Billing information

Parameter	Description
Billing Mode	Billing mode. • Pay-per-use
Created	Time when a pay-per-use or yearly/monthly cluster is created.

Table 9-8 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.

Parameter	Description
OBS Hot Data Used Capacity (used for GaussDB(DWS) 3.0 only)	OBS hot data capacity used by GaussDB(DWS) 3.0.
Cold Data Used 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 9-9 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.
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 9-10 Data encryption information

Parameter	Description
Key Name	Indicates the database encryption key of the cluster when **Encrypt DataStore** is enabled.

Parameter	Description
Last Key Rotation Time	Indicates the time when the last encryption key is rotated when **Encrypt DataStore** is enabled.

9.4 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.

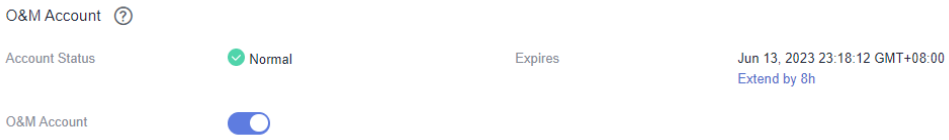
NOTE

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:

1. Enable or disable the O&M account.
2. Check the O&M account status.
3. Check and extend the validity period of the O&M account.

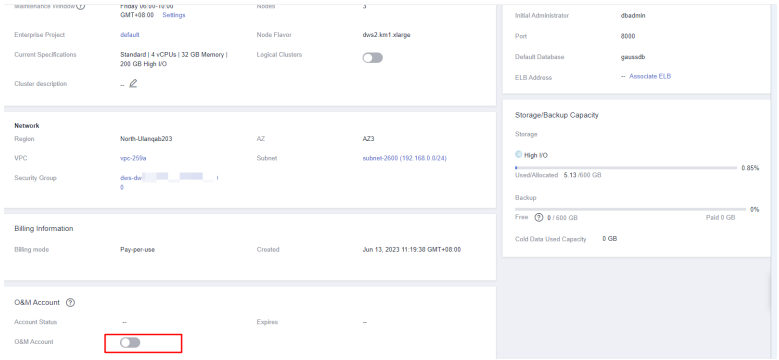


Enabling the O&M Account

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



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.

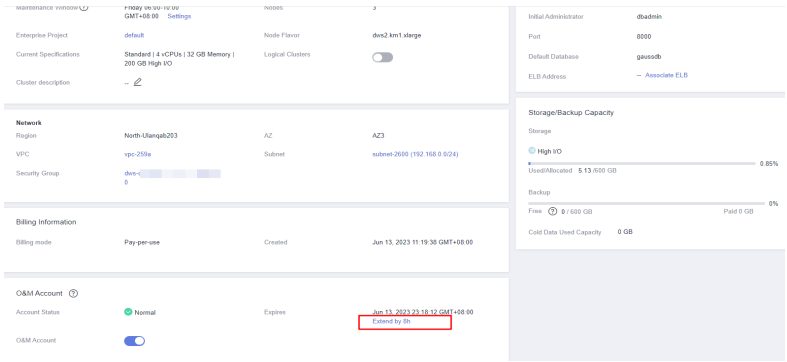
----End

Extending the Validity Period

Step 1 Log in to the GaussDB(DWS) management 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

9.5 Managing 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.

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.

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:

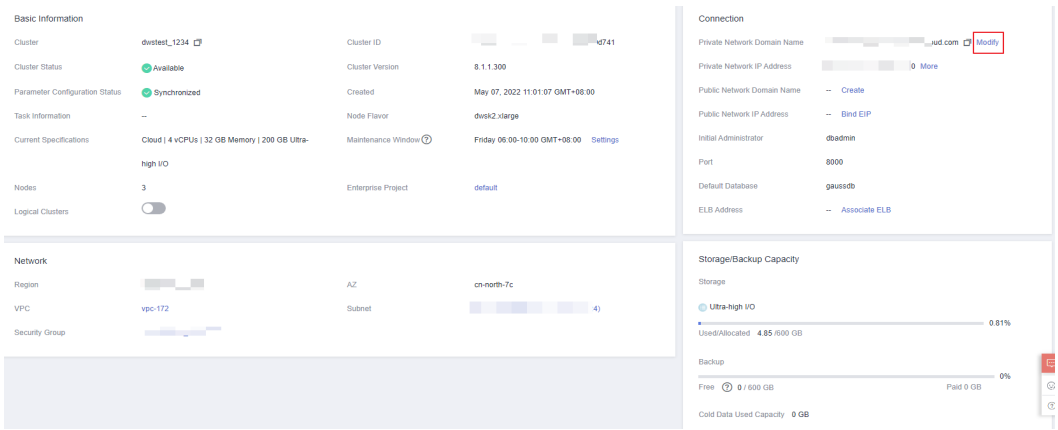
Step 1 Log in to the GaussDB(DWS) management console.

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.

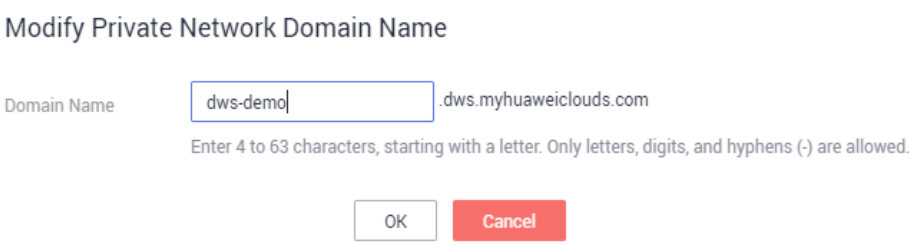
Step 4 In the **Connection** area, click **Modify** next to the automatically generated **Private Network Domain Name**.

Figure 9-2 Viewing the private network domain name



Step 5 In the **Modify Private Network Domain Name** dialog box, enter the target domain name and click **OK**.

Figure 9-3 Modifying the private network domain name



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) management 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.

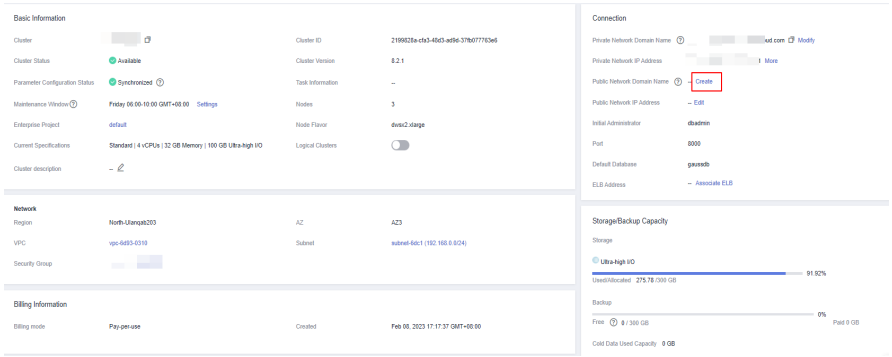
Step 5 In the **Edit Elastic IP** dialog box, select an EIP from the drop-down list to bind it to a specified CN.

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.

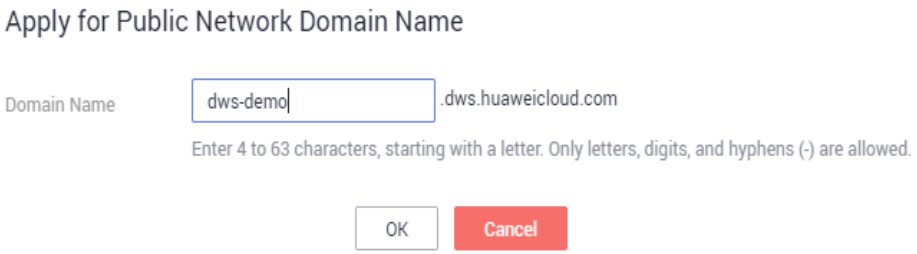
Step 6 In the **Connection** area, click **Create** next to **Public Network Domain Name** to create a public network domain name for the cluster.

Figure 9-4 Creating a public network domain name



Step 7 In the **Apply for Public Network Domain Name** dialog box, enter the target domain name and click **OK**.

Figure 9-5 Applying for a public network domain name



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:

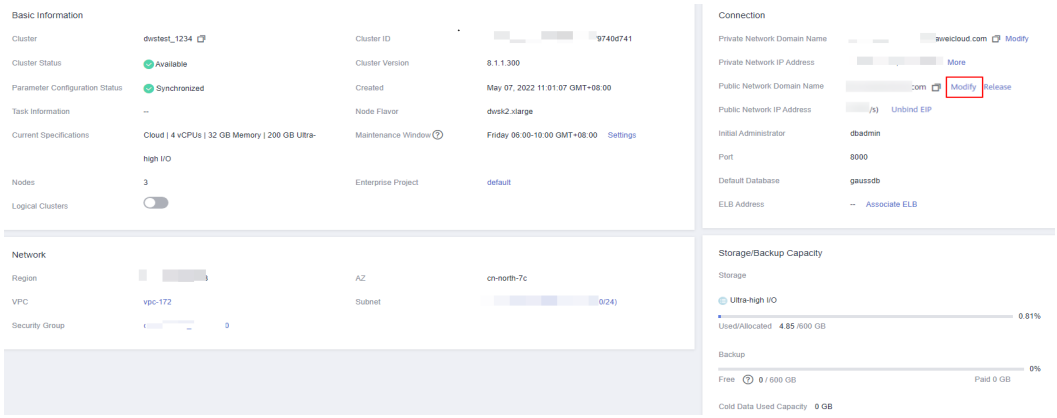
Step 1 Log in to the GaussDB(DWS) management console.

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.

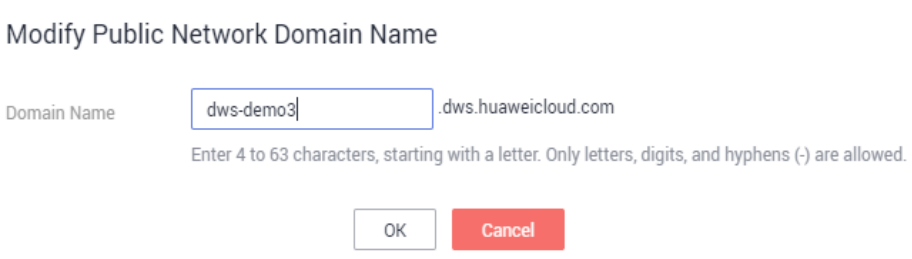
Step 4 Click **Modify** next to the **Public Network Domain Name** in the **Connection** area.

Figure 9-6 Modifying the public network domain name



Step 5 In the **Modify Public Network Domain Name** dialog box, enter the target domain name and click **OK**.

Figure 9-7 Modifying the public network domain name



----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:

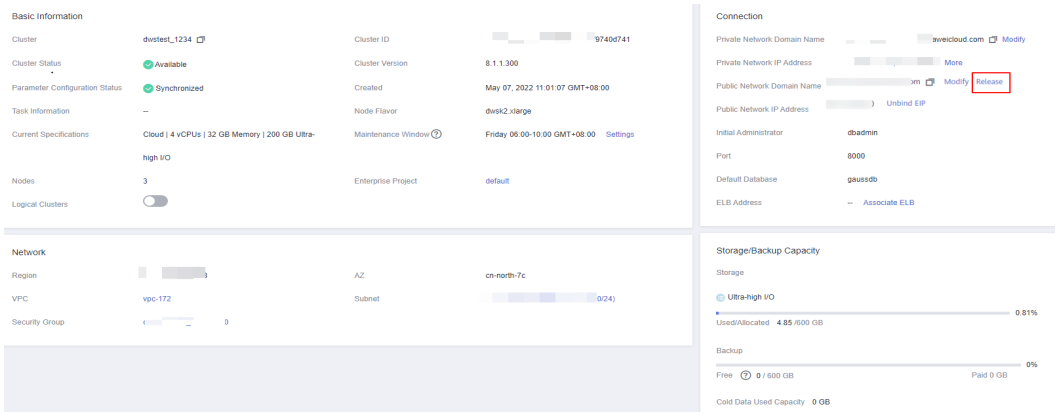
Step 1 Log in to the GaussDB(DWS) management console.

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 9-8 Releasing a public network domain name



Step 5 In the **Release Domain Name** dialog box, click **Yes**.

----End

9.6 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.

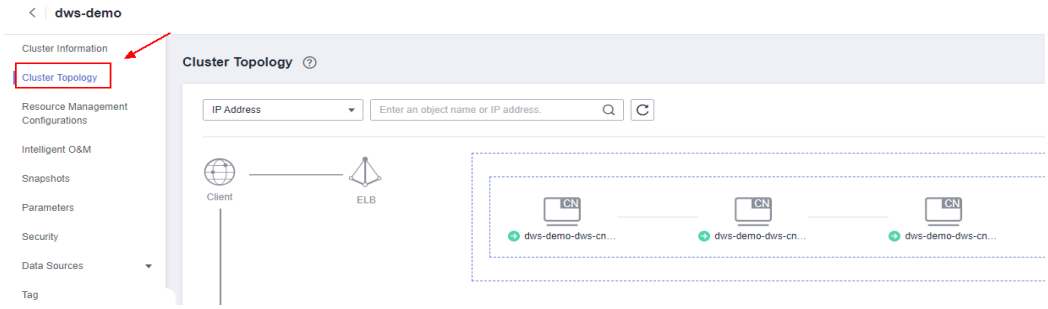
Viewing the Cluster Topology

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

Step 2 In the cluster list, click the name of a cluster.

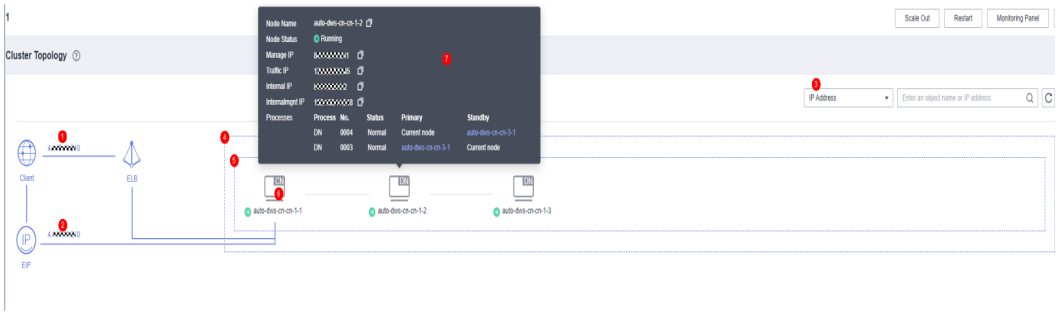
Step 3 On the **Cluster Details** page, click the **Cluster Topology** tab.

Step 4 In the upper part 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



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. A ring. Each ring occupies a line. An icon in a ring indicates a node.
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 9-11 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 **Associating and Disassociating ELB**.
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 9-12 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 9-13 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.

9.7 Managing Tags

9.7.1 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 two types of 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.

9.7.2 Tag Management

This section describes how to search for clusters based on tags and how to add, modify, and delete tags.

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 9-9 Adding a tag to a cluster

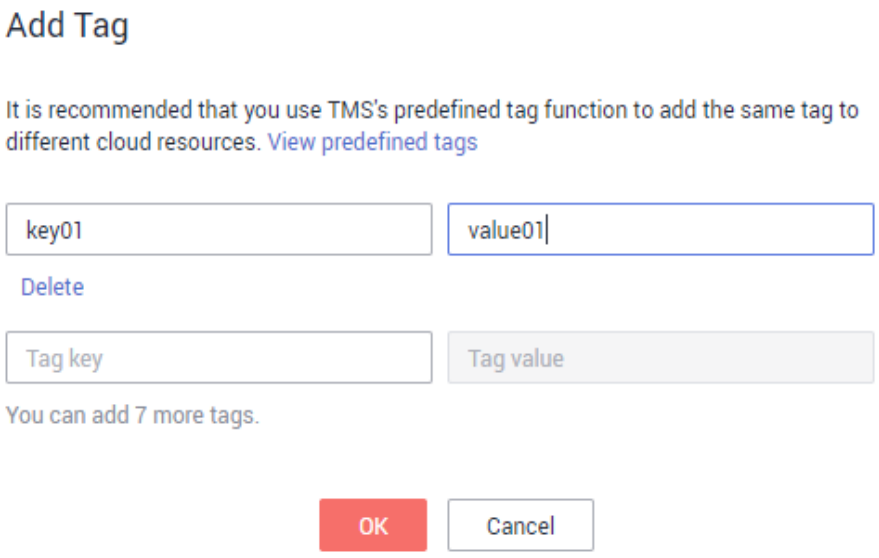


Table 9-14 Tag parameters

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. • Enter a tag key in the text box. The tag key can contain a maximum of 128 characters and cannot be an empty string. It cannot start with **_sys_**. 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. The tag key can contain a maximum of 255 characters and cannot be an empty string. Only letters, digits, spaces, and the following characters are allowed: _ . : = + - @	value01

Step 4 Click **OK**.

----End

Searching for Clusters Based on Tags

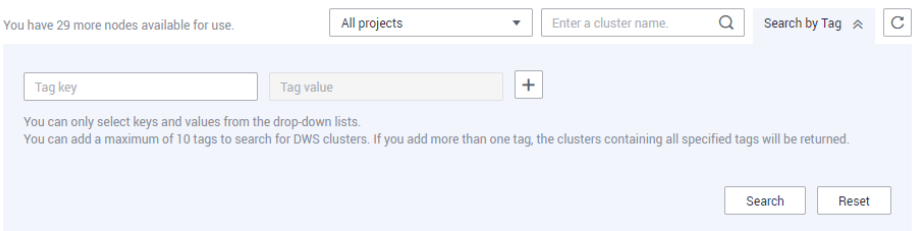
You can quickly locate a tagged cluster using tags.

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

Step 2 Choose **Cluster** > **Dedicated Cluster**.

Step 3 Click **Search by Tag** on the upper right of the cluster list to expand the tab page.

Figure 9-10 Search by Tag



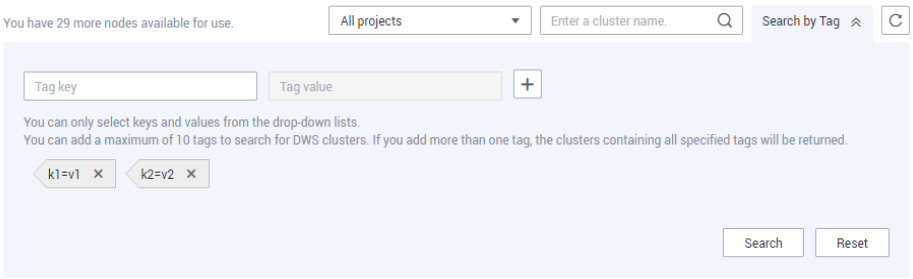
Step 4 In the **Search by Tag** area, click the **Tag Key** text box to select a tag key from the drop-down list and then click the **Tag Value** text box to select the corresponding tag value.

You can only enter a tag key or value that exists 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 + to add the selected tag to the area under the text boxes.

- Select another tag in the text boxes and click + to generate a tag combination for cluster search. You can add a maximum of 10 tags to search for data warehouse clusters. If you specify more than one tag, clusters containing all the specified tags will be displayed.
- To delete an existing tag, click × next to the tag.
- You can click **Reset** to clear all added tags.

Figure 9-11 Adding the tag key and value



Step 6 Click **Search**. The target cluster will be displayed in the cluster list.

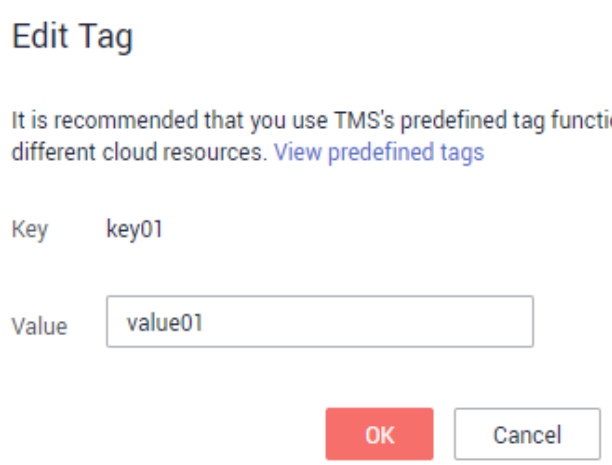
----End

Modifying a Tag

Step 1 On the **Clusters** > **Dedicated Clusters** page, click the name of the cluster for which a tag is to be modified, and click the **Tags** tab.

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.

Figure 9-12 Editing a tag



Step 3 Enter the new key value in the **Value** text box.

Step 4 Click **OK**.

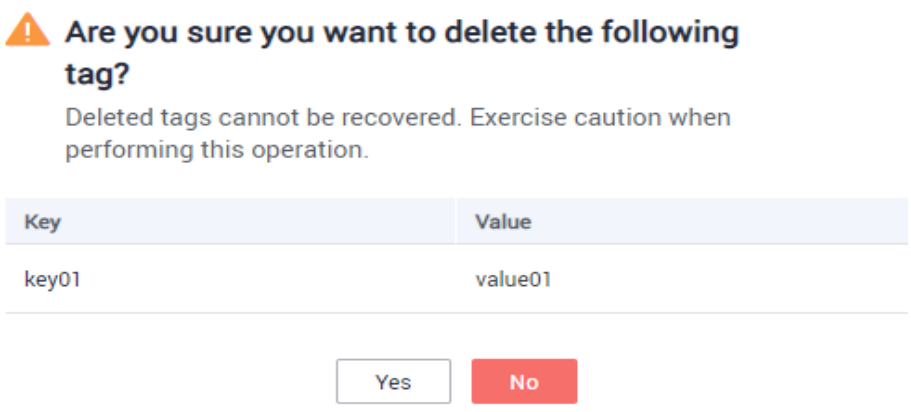
----End

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 the **Tags** tab.

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.

Figure 9-13 Deleting a tag



Step 3 Click **Yes** to delete the tag.

----End

9.8 Managing Enterprise Projects

An enterprise project is a cloud resource management mode. Enterprise Management provides users with comprehensive management in cloud-based

finance. Unlike common management consoles that feature independent control and configuration of cloud products, the Enterprise Management console is oriented to resource management. It helps enterprises with cloud-based management in finance in 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) 2.0 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.

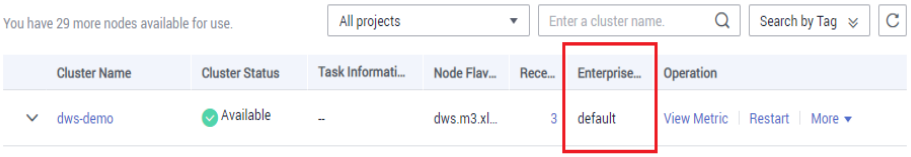
When you delete a cluster, the association between the cluster and its enterprise project is automatically deleted.

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.

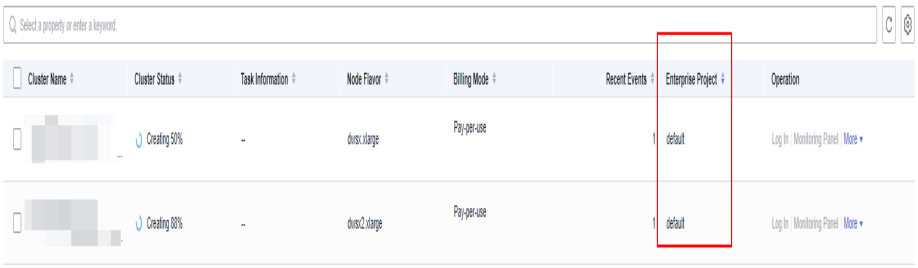
- In the cluster list on the **Clusters** page, view the enterprise project to which the cluster belongs.

  Figure 9-14 Viewing the enterprise project

- 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 9-15 Viewing the enterprise project

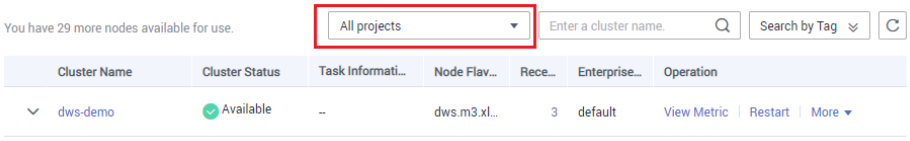


- When querying the resource list of a specified project on the Enterprise Management console, you can also query the GaussDB(DWS) resources.

Searching for Clusters by Enterprise Project

Log in to the GaussDB(DWS) management console, choose **Clusters** > **Dedicated Clusters**, click **All projects** above the cluster list, and select the required project name from the drop-down list to view all clusters associated with the project.

Figure 9-16 Search by enterprise projects



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.

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 GaussDB(DWS) cluster, configure the VPC to which the cluster belongs. To obtain the VPC list, add action **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.

----End

9.9 Managing Clusters That Fail to Be Created

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

Checking the Cause of a Creation Failure

Step 1 Log in to the GaussDB(DWS) management console. In the navigation pane on the left, choose **Clusters > Dedicated Cluster**. The **Dedicated Clusters** page is displayed.

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 creation failure.

For details about the error codes, see "Error Code Reference". 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) management console. In the navigation pane on the left, choose **Clusters > Dedicated Cluster**. The **Dedicated Clusters** page is displayed.

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

Step 3 (Optional) If the cluster is bound with an EIP during creation, click **Release the EIP bound with the cluster** to release the EIP.

Step 4 In the dialog box that is displayed, click **Yes** to delete the cluster.

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

9.10 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 details about how to solve this problem, see **High Disk Usage and Read Only Status**.

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.

Impact 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.

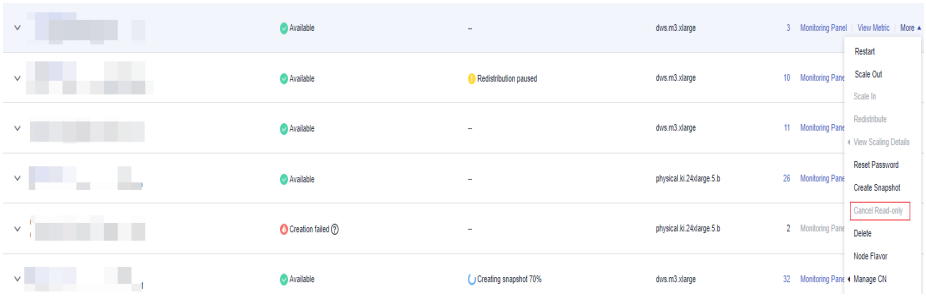
Removing Read-only Status

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

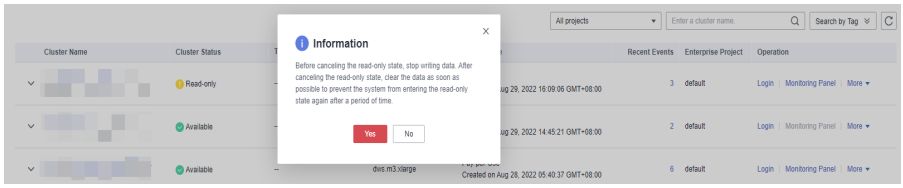
Step 2 Choose **Cluster** > **Dedicated Cluster**.

All clusters are displayed by default.

Step 3 In the **Operation** column of the target cluster, choose **More** > **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

9.11 Performing a Primary/Standby Switchback

Context

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.

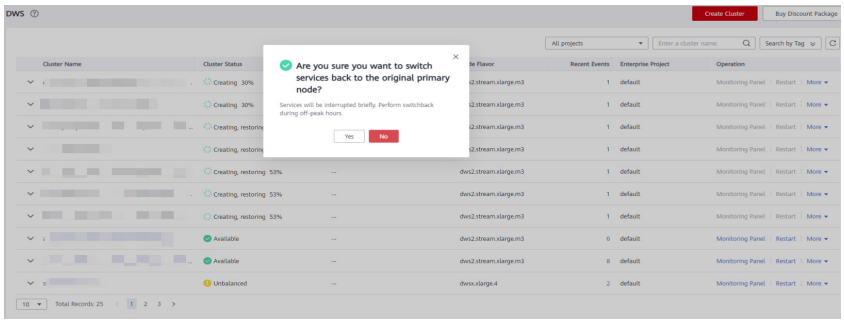
Procedure

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

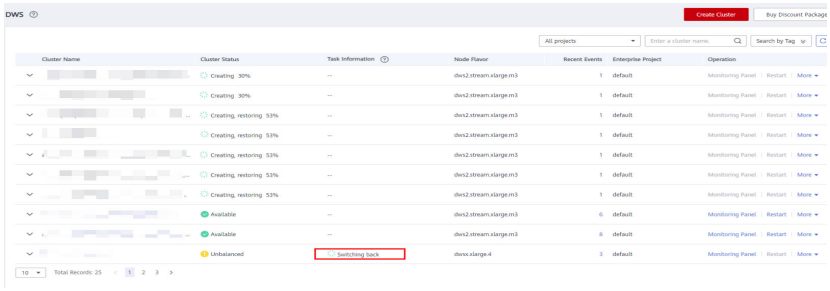
Step 2 On the **Clusters > Dedicated Clusters** page, locate the cluster in unbalanced state.

Step 3 In the **Cluster Status** column of the cluster, click **Fix** under **Unbalanced**.

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

9.12 Cluster Restart

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 **Monitoring Clusters Using 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) management console.

Step 2 Choose **Clusters** > **Dedicated Cluster**.

Step 3 In the **Operation** column of the cluster to be restarted, click **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

9.13 Resetting a Password

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 Database Parameters**.

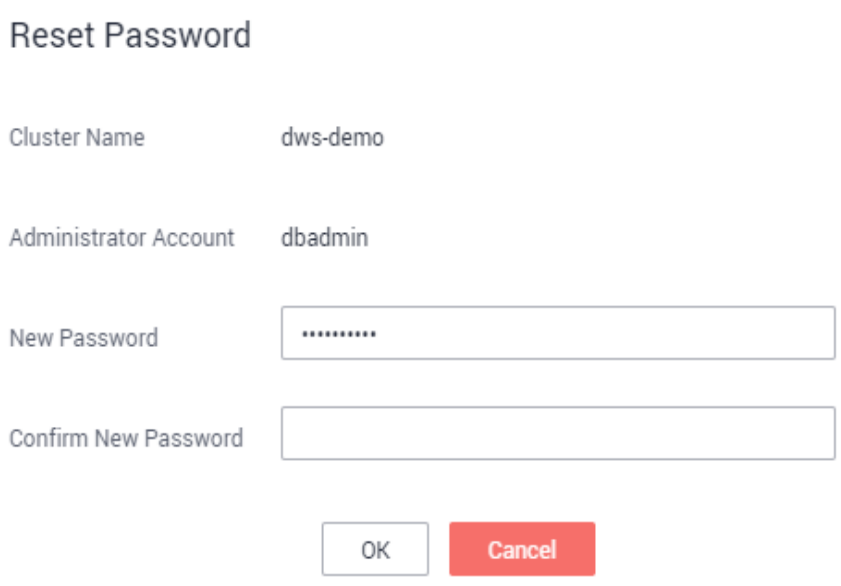
Resetting a Password

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

Step 2 Choose **Cluster** > **Dedicated Cluster**.

Step 3 In the **Operation** column of the target cluster, choose **More** > **Reset Password**.

Figure 9-17 Password resetting



Step 4 On the displayed **Reset Password** page, set a new password, confirm the password, and then click **OK**.

The password complexity requirements are as follows:

- Contains 12 to 32 characters.
- Cannot be the username or the username spelled backwards.
- Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\=-)

- Passes 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

9.14 Cluster Upgrade

By default, you do not need to manually upgrade a GaussDB(DWS) cluster. To upgrade a cluster on the console, see **Delivering a Cluster Upgrade Task on the Console**.

GaussDB(DWS) will notify you of any cluster O&M operation by sending SMS messages. Exercise caution when performing operations on the cluster during the O&M period.

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**.

If the upgrade affects the current query requests or service running, contact technical support for emergency handling.

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.

Upgrading a Cluster

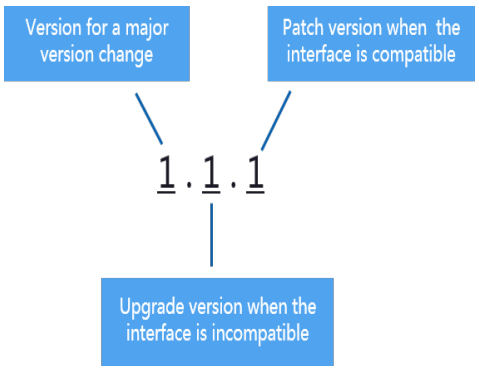
By default, you do not need to care about GaussDB(DWS) cluster patching or upgrading because GaussDB(DWS) will handle version upgrade automatically. After GaussDB(DWS) is upgraded, it will automatically upgrade the cluster to the latest version. During the upgrade, the cluster will be restarted and cannot provide services for a short period of time.

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 does not affect the original cluster data or specifications.

The following figure shows the cluster version.

Figure 9-18 Version description



- **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.

Delivering a Cluster Upgrade Task on the Console

Prerequisites

For clusters 8.1.1 or later, you need to deliver cluster upgrade operations on the console.

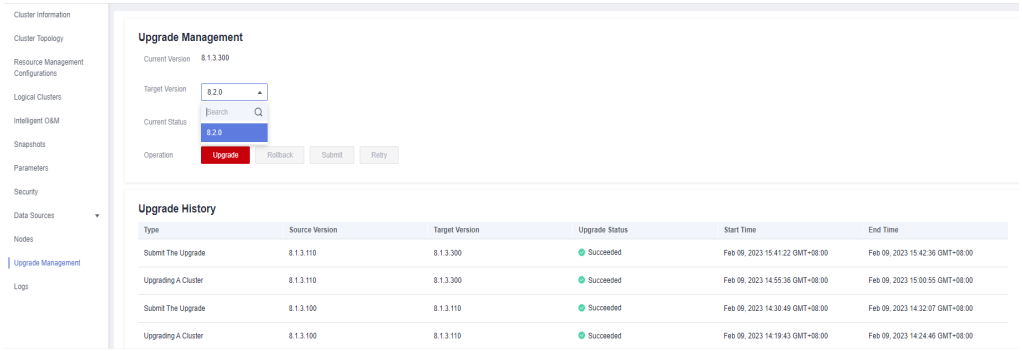
Procedure

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

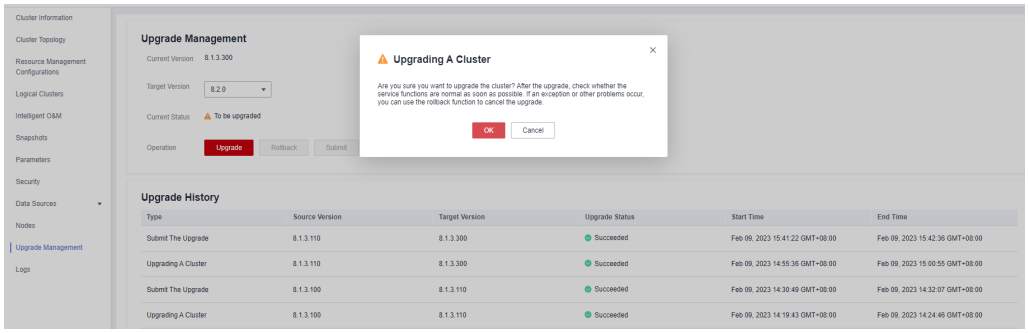
Step 2 In the cluster list, click the name of a cluster.

Step 3 In the navigation pane, choose **Upgrade Management**.

Step 4 On the **Upgrade Management** page, select a version from the **Target Version** drop-down list.



Step 5 Click **Upgrade**. Click **OK** in the displayed dialog box.

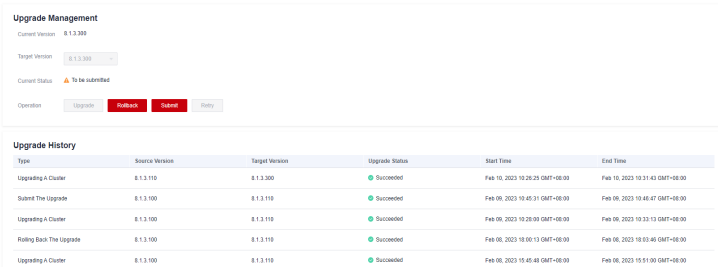


Step 6 Check whether the cluster is successfully upgraded.

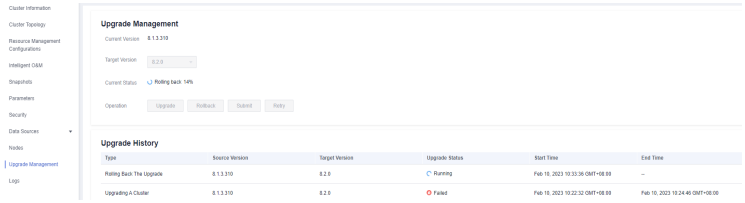
- If the cluster version is 8.1.3 or later, the cluster enters the service observation period after the upgrade is complete. If you have verified your services, click **Submit** on the **Upgrade Management** page to complete the cluster upgrade. If you find your cluster performance affected by the upgrade, you can click **Rollback** to roll back the upgrade.

  NOTE

  - In versions earlier than 8.1.3, there is neither rollback nor submission button after the upgrade is complete.
  - If you do not submit the upgraded version, there will be a **wlm** thread which occupies the system storage space and affects the performance.



- If the cluster upgrade fails, click **Rollback** to roll back to the original cluster version, or click **Retry** to deliver the upgrade again.



----End

9.15 Associating and Disassociating ELB

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 .

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.
- For load balancing and high availability purposes, and to prevent single CN failures, a cluster must be bound to 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) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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.

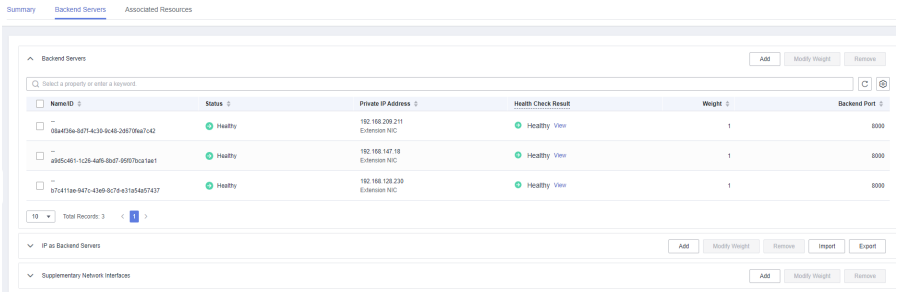
Step 4 On the **Basic Information** page that is displayed, click **Associate ELB** and select the ELB name. If no load balancer exists, create one on the ELB management console. Then refresh the GaussDB(DWS) page and associate ELB with the cluster.

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.



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.

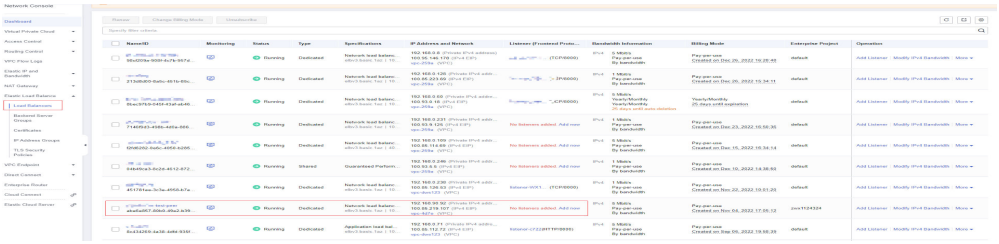
----End

Prerequisites for Binding an ELB to a Cluster Across VPCs

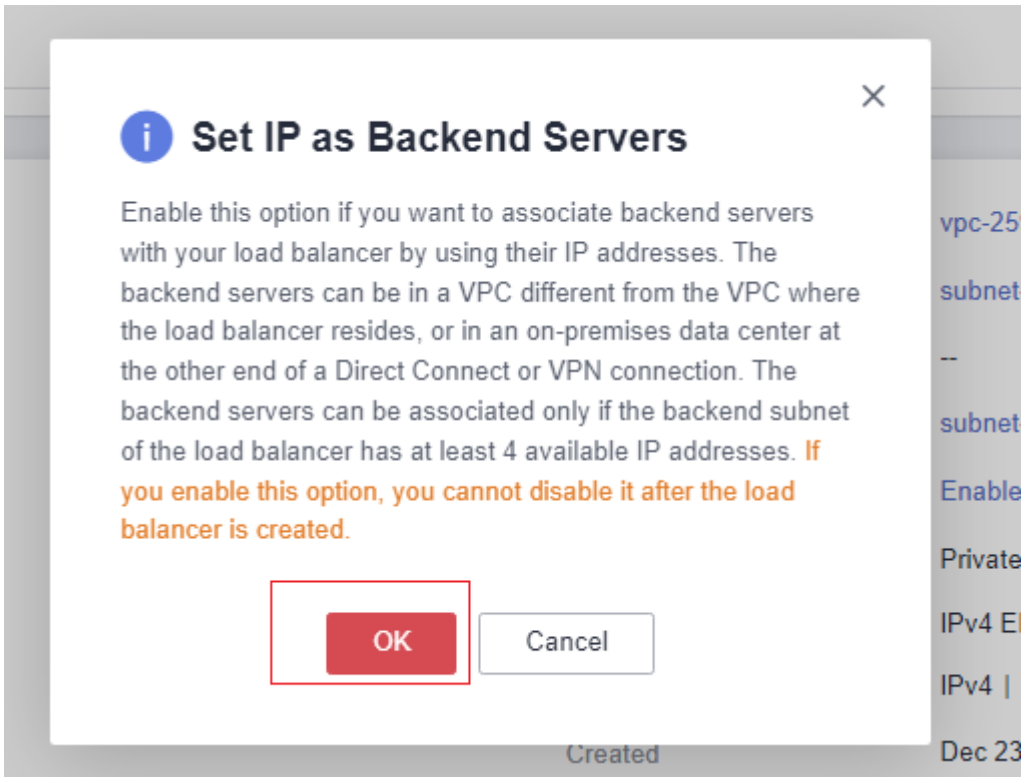
Enabling ELB for a cross-VPC backend server

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.



Step 3 On the **Summary** page, enable **IP as a Backend**, confirm the information, and click **OK**.



Step 4 Check the VPC and subnet segment.

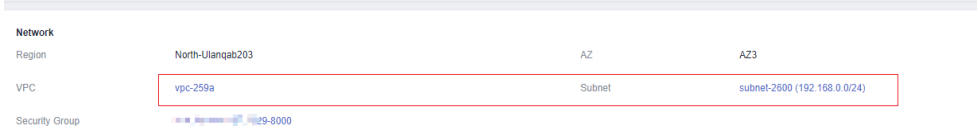
----End

Connecting the cluster VPC and the ELB VPC (by using VPC peering as an example)

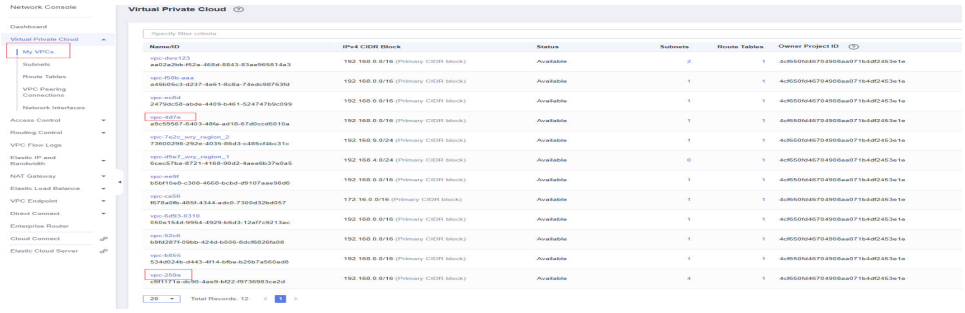
Step 1 Log in to the GaussDB(DWS) console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

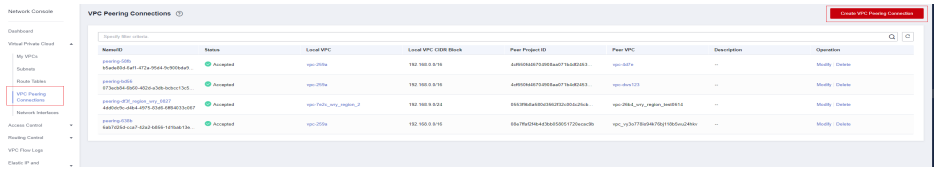
Step 3 In the cluster list, click the name of a cluster to go to the cluster details page. Check the VPC and subnet segment of the cluster.



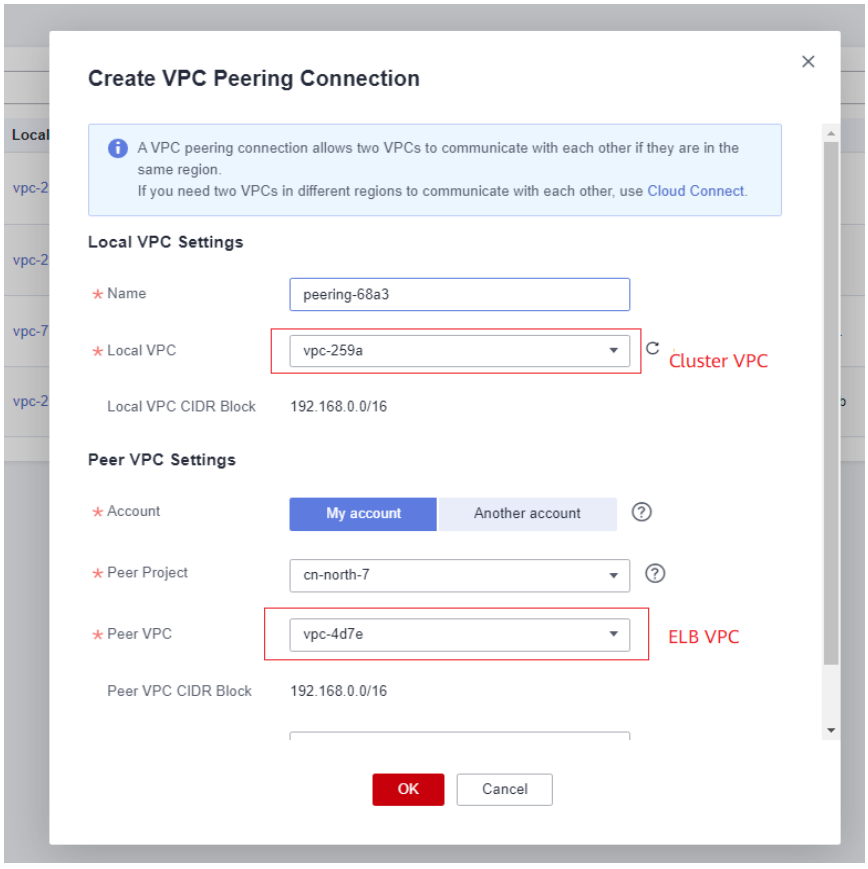
Step 4 Log in to the VPC management console. choose **My VPCs** in the navigation pane, and locate the VPC for which you want to create a VPC peering connection.



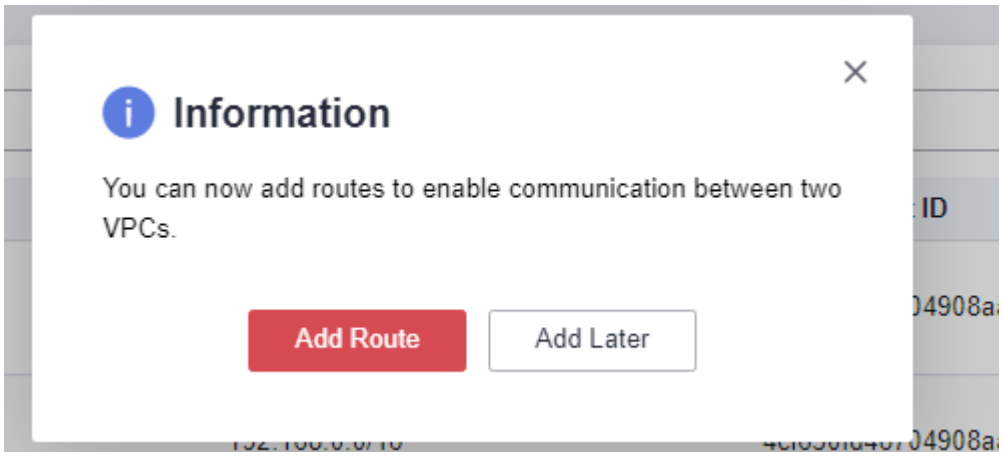
Step 5 Choose **VPC Peering Connections**. In the upper right corner of the page, click **Create VPC Peering Connection**.



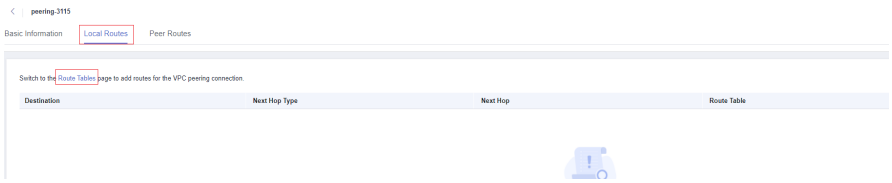
Step 6 On the displayed page, set **Local VPC** to the cluster VPC, and set **Peer VPC** to the VPC of the ELB. Confirm the settings and click **OK**.



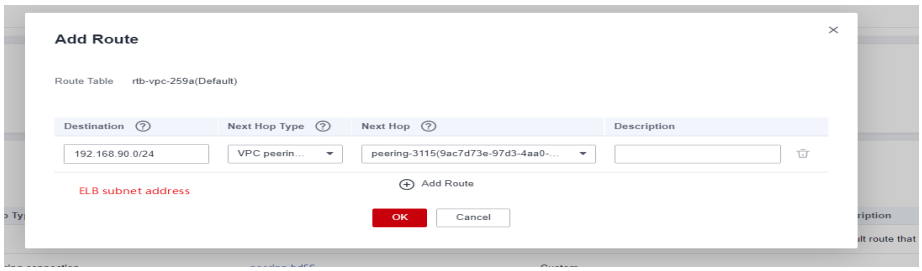
Step 7 Click **Add Route** to add the route information.



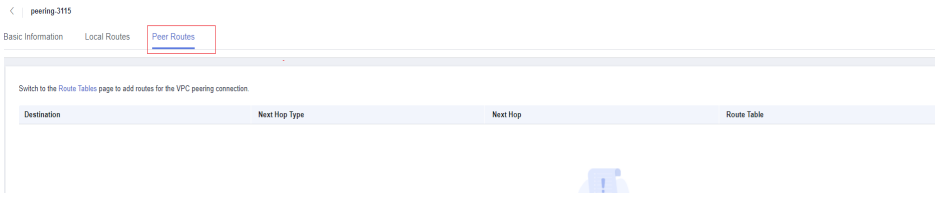
Step 8 Click the name of the created VPC peering connection. On the displayed page, click the **Local Routes** tab, click **Route Tables**, and add the route table of the cluster VPC.



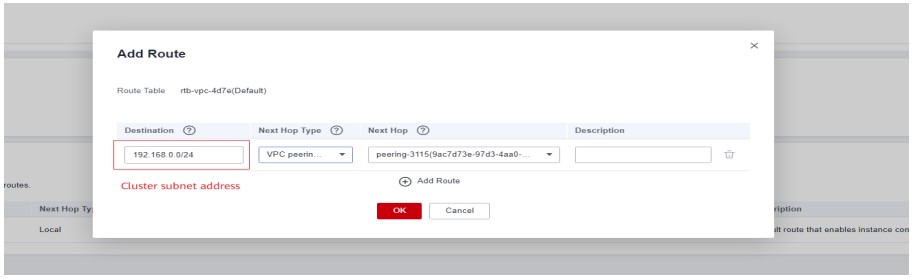
Step 9 In the local route table, set **Destination** to the subnet CIDR block of the ELB VPC, set **Next Hop Type** to **VPC peering connection**, and set **Next Hop** to the created VPC peering connection. Click **OK**.



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.



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**.



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

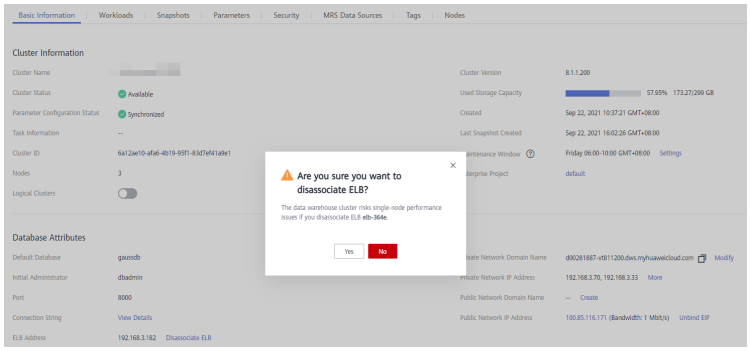
Disassociating ELB

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

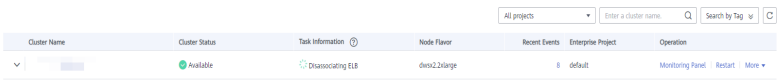
Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

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

9.16 Managing CNs

Purpose

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

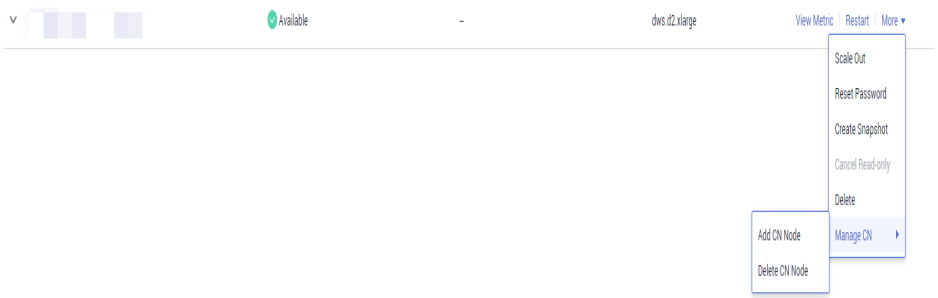
- During resource provisioning, the default number of CNs is 3. You can adjust the number of CNs based on the number of provisioned nodes. The number of CNs ranges from 2 to 20.
- Do not perform other O&M operations when adding or deleting a CN.
- Adding CNs consumes lots of CPU and I/O resources, which 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

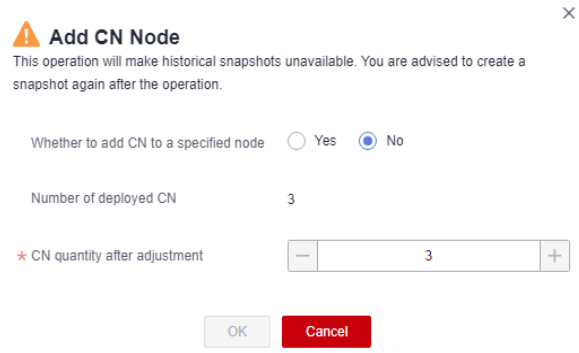
Step 1 Log in to the GaussDB(DWS) management console.

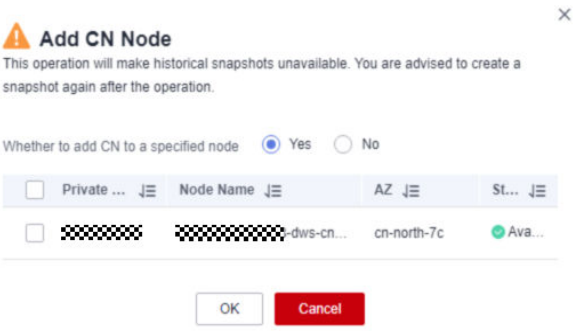
Step 2 On the **Clusters** > **Dedicated Clusters** page, locate the cluster to which you want to add CNs.

Step 3 In the **Operation** column of the specified cluster, choose **More** > **Manage CN** > **Add CN Node**.



Step 4 In the displayed dialog box, determine whether to add CNs to a specified node. If you select **No**, set the number of CNs after adjustment and click **OK**. If you select **Yes**, select a node and click **OK**.





NOTICE

- Before adding a CN, ensure that the cluster is in the **Available** or **Unbalanced** state.
- The number of CNs after adjustment cannot exceed the number of deployed CNs. It must be less than or equal to the number of nodes, and less than or equal to 20.

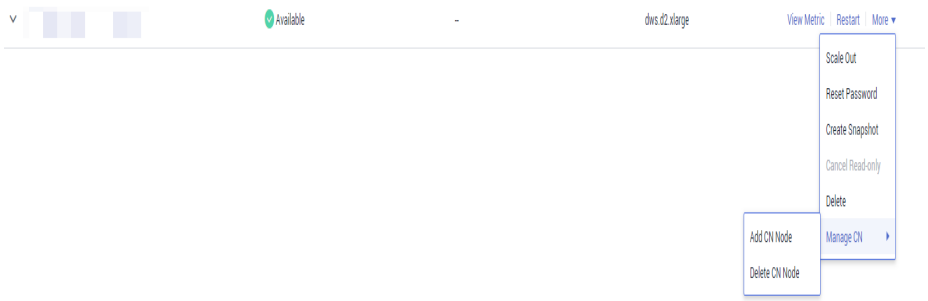
----End

Deleting CNs

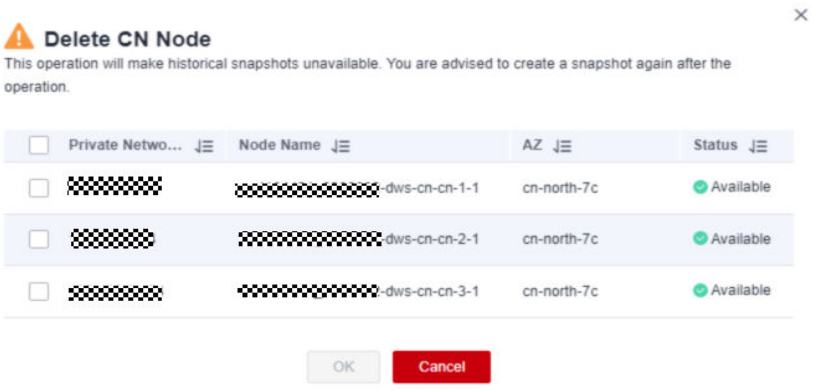
Step 1 Log in to the GaussDB(DWS) management console.

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 to be deleted and click **OK**.



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

10 Data Migration

10.1 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 and MRS to GaussDB(DWS).

NOTE

This feature is supported only in 8.2.0 or later.

10.2 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. You can create and manage clusters.

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.

NOTE

- 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 on the left, choose **Data Migration** > **Instances**.

Step 3 In the upper right corner of the page, click **Buy GDS-Kafka Instance**. Configure cluster parameters.

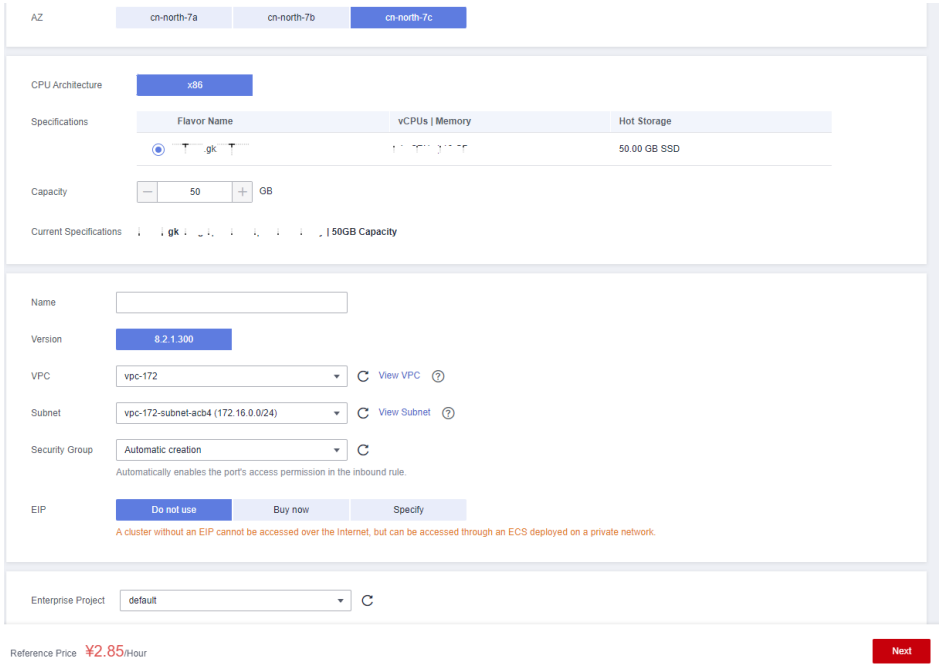


Table 10-1 Parameter description

Parameter	Description	Example Value
CPU Architecture	The following CPU architectures can be selected: • **x86** • **Kunpeng** **NOTE** The only difference between the x86 and Kunpeng architectures lies in the underlying architecture, of which the application layer is unaware. The same SQL syntax is used. If x86 servers are sold out when you create a cluster, select the Kunpeng architecture.	x86
Flavor	Select a node flavor. Each node flavor shows the vCPU, memory, and recommended application scenario.	-
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.	-

Parameter	Description	Example Value
Version	Version of the database instance installed in the cluster. The version in the screenshot is for reference only.	8.2.1.300
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.	-
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. ● **Automatically assign**: 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.	-

Parameter	Description	Example Value
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 Confirm the information and click **Submit**.

----End

Viewing Instance Details

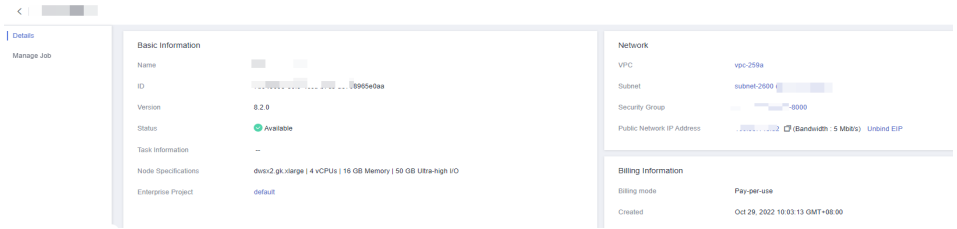
On the instance details page, you can view the basic information and network information about the cluster.

Procedure

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

Step 2 In the navigation pane on the left, choose **Data Migration** > **Instances**.

Step 3 Click the name of an instance to go to the instance details page.



----End

10.3 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.
    - 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 on the left, choose **Data Migration** > **Connections**.

Step 3 Click **Create Connection**.

Step 4 Configure connection parameters. For more information, see **Connection parameters**.

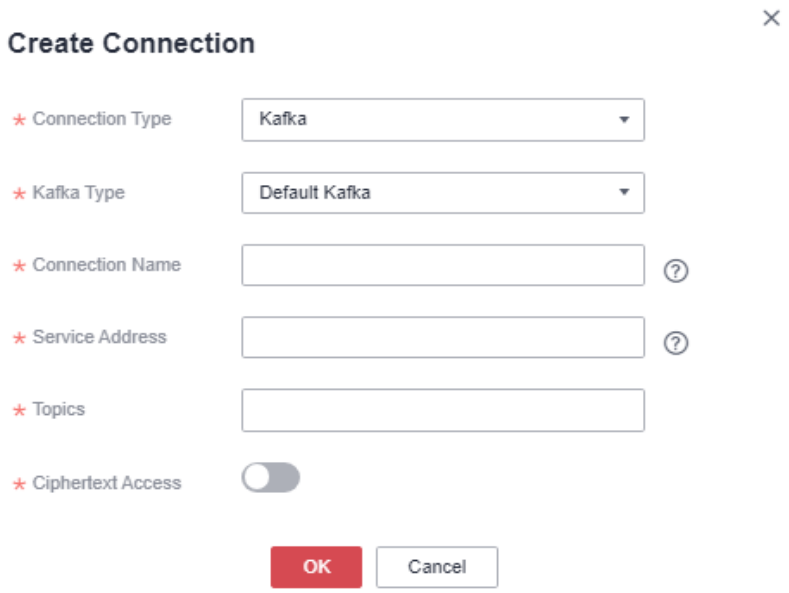


Table 10-2 Connection parameters

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

Protocol	Parameter	Mandatory	Description
	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.
	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.

Protocol	Parameter	Mandatory	Description
	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.
	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

Protocol	Parameter	Mandatory	Description
	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.
	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.

Protocol	Parameter	Mandatory	Description
	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.
	Schema	Yes	Name of a schema in the GaussDB(DWS) database.
DWS Cluster	Connection Name	Yes	Connection name, which can be customized. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
	Cluster Name	Yes	Name of a GaussDB(DWS) cluster.
	Username	Yes	Username of the database corresponding to the GaussDB(DWS) cluster.
	Password	Yes	Password of the database corresponding to the GaussDB(DWS) cluster.
	Database	Yes	Name of the database corresponding to the GaussDB(DWS) cluster.
	Schema Name	No	Schema of the database corresponding to the GaussDB(DWS) cluster.
MRS	Connection Name	Yes	Connection name, which can be customized. Only letters, numbers, underscores (_), and hyphens (-) are allowed.
	Manager Address	Yes	Address of the MRS cluster management page.
	Username	Yes	Username for logging in to the MRS cluster management page.
	Password	Yes	Password for logging in to the MRS cluster management page.

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 on the left, choose **Data Migration** > **Connections**.

Step 3 In the **Operation** column of a connection, click **Modify**.

Step 4 In the **Modify Connection** dialog box, configure **Connection Name**, **Connection Address**, **Topics**, **User**, and **Password**.

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 on the left, choose **Data Migration** > **Connections**.

Step 3 In the **Operation** column of a connection, click **Delete**.

Step 4 In the displayed dialog box, click **OK**.

----End

10.4 Table Mappings

Mapping Overview

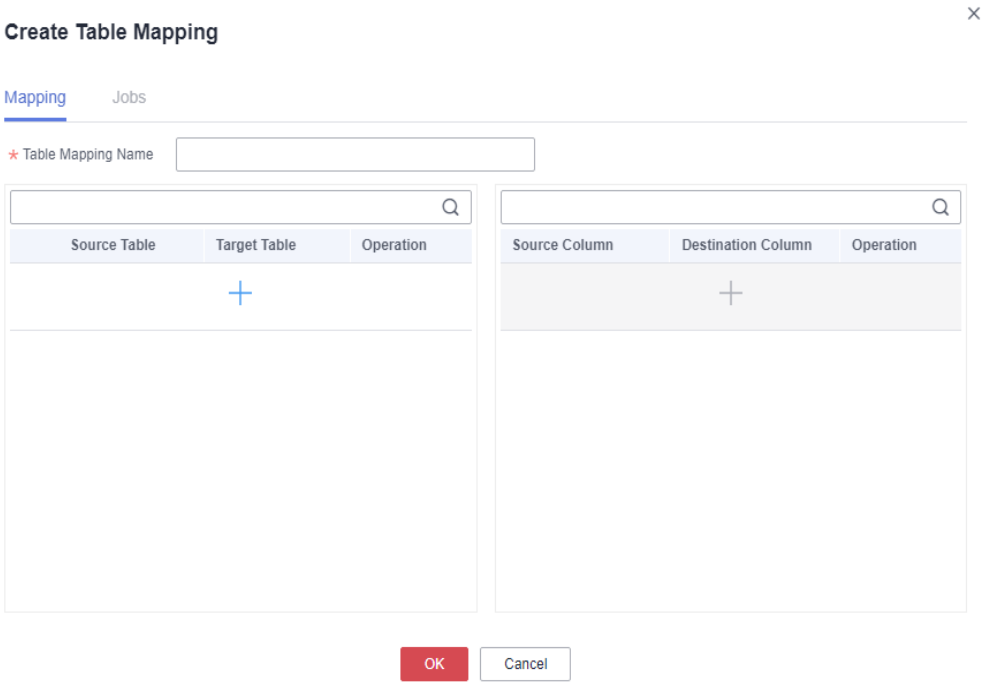
Before creating a job, you need to create a mapping to map the table structures of the source and destination databases, facilitating data migration between databases.

Creating a Table Mapping

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

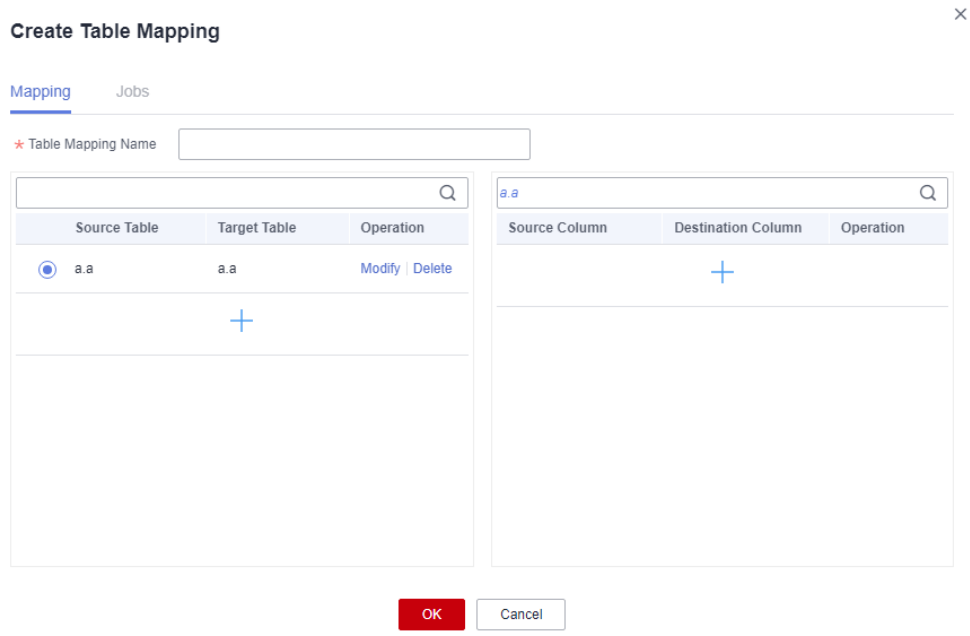
Step 2 In the navigation tree on the left, choose **Data Migration** > **Table Mappings**.

Step 3 Click **Create Table Mapping**.

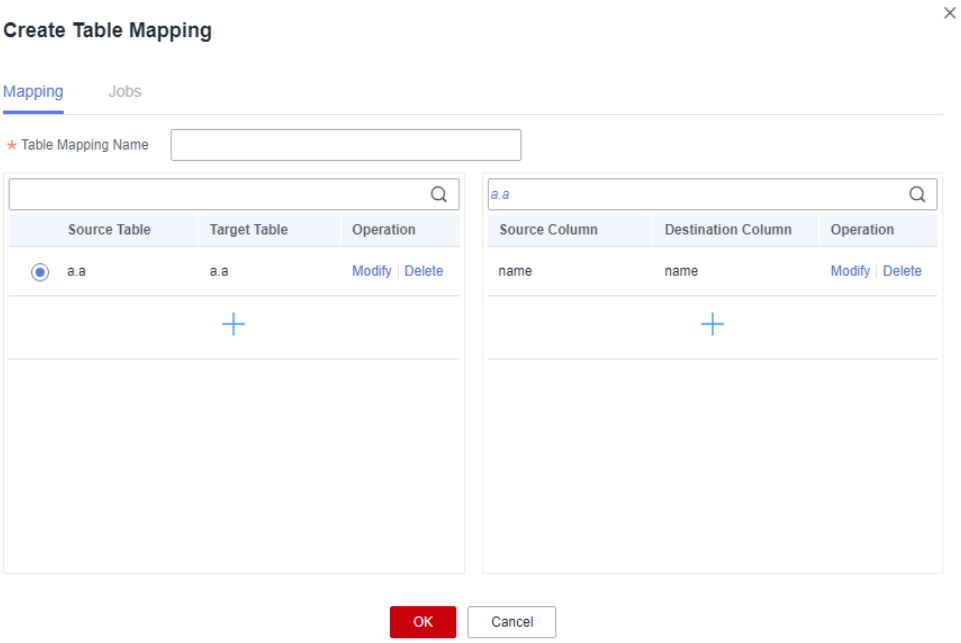


Step 4 Configure parameters.

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

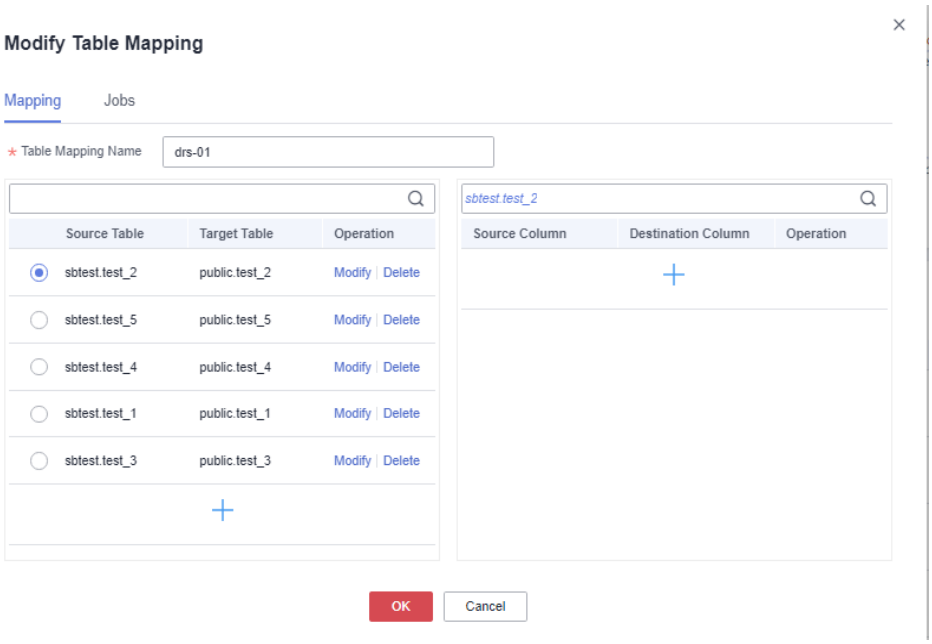
Modifying a Table Mapping

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

Step 2 In the navigation tree on the left, choose **Data Migration** > **Table Mappings**.

Step 3 In the **Operation** column of a table mapping, click **Modify**.

Step 4 In the **Modify Table Mapping** dialog box, configure **Table Mapping Name**, **Source Table**, and **Target Table**.



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

----End

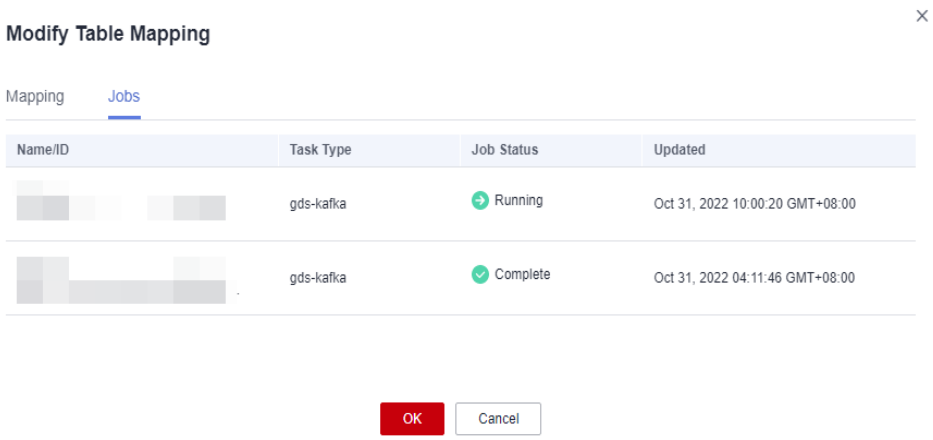
Checking a Table Mapping

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

Step 2 In the navigation tree on the left, choose **Data Migration** > **Table Mappings**.

Step 3 In the **Operation** column of a table mapping, click **Modify**.

Step 4 In the **Modify Table Mapping** dialog box, click the **Jobs** tab to view the bound jobs.



----End

Deleting a Table Mapping

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

Step 2 In the navigation tree on the left, choose **Data Migration** > **Table Mappings**.

Step 3 In the **Operation** column of a table mapping, click **Delete**.

Step 4 In the displayed dialog box, click **OK**.

----End

10.5 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.

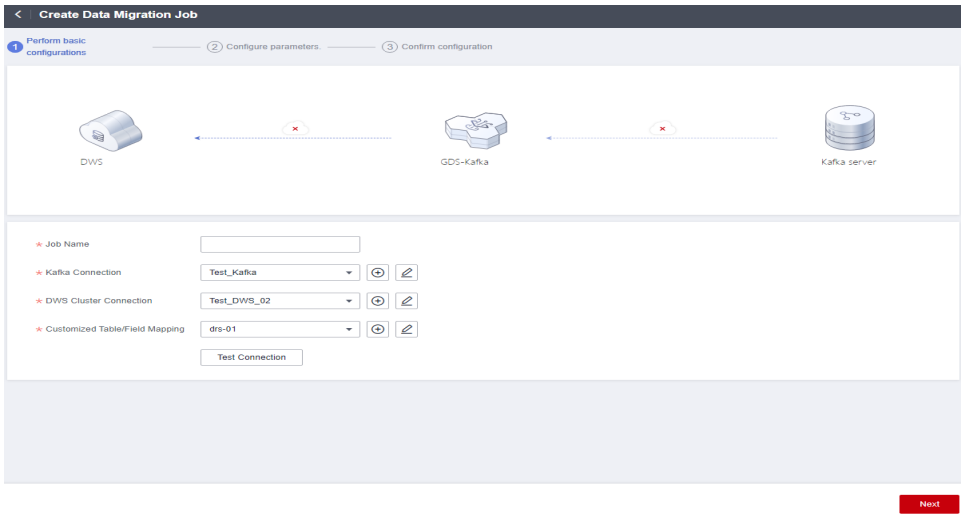
Step 2 In the navigation pane on the left, choose **Data Migration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane on the left, 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

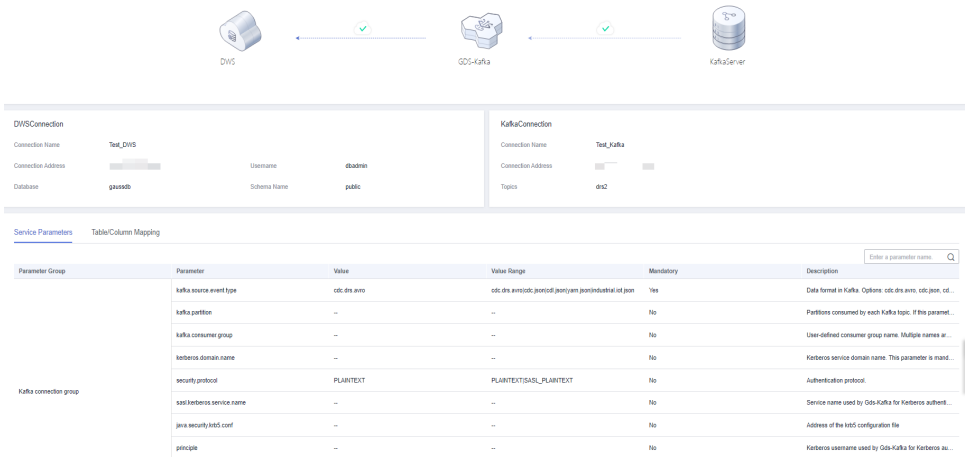
Step 1 Log in to the GaussDB(DWS) console.

Step 2 In the navigation pane on the left, choose **Data Migration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane on the left, 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.



----End

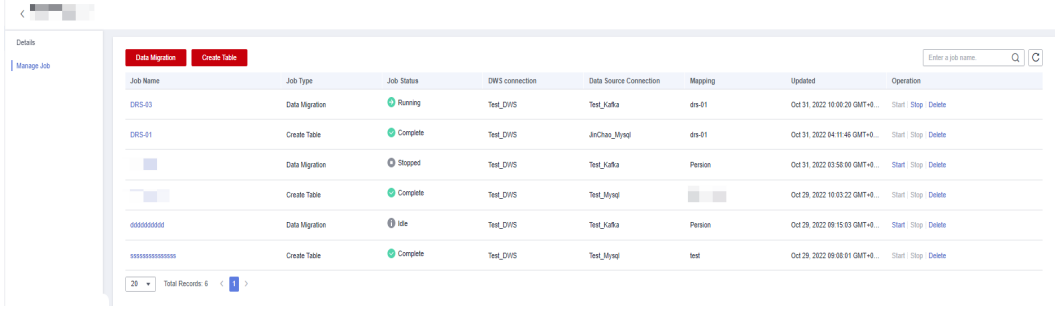
Starting a Job

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

Step 2 In the navigation pane on the left, choose **Data Migration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane on the left, 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

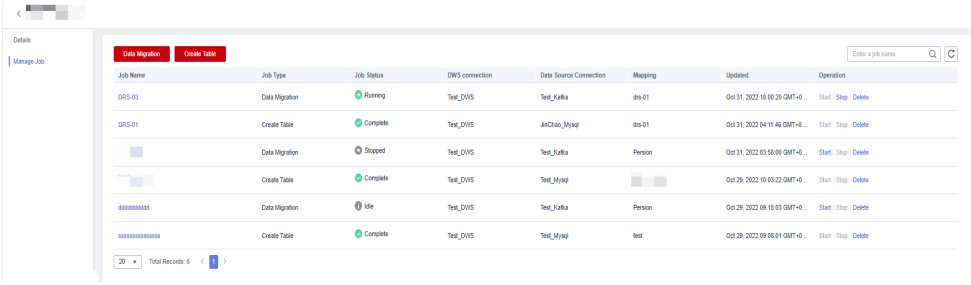
Stopping a Job

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

Step 2 In the navigation pane on the left, choose **Data Migration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane on the left, 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

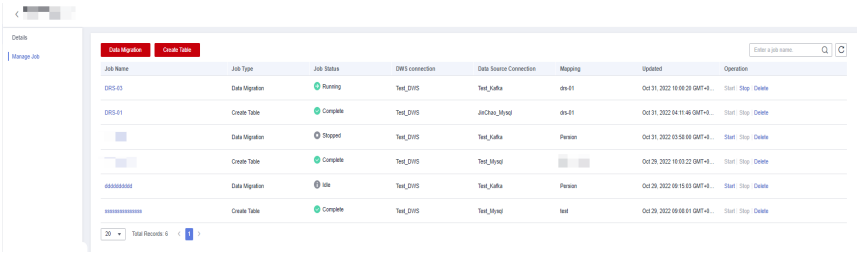
Deleting a Job

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

Step 2 In the navigation pane on the left, choose **Data Migration** > **Instances**.

Step 3 Click the name of an instance to go to the details page.

Step 4 In the navigation pane on the left, click **Manage Job**.



Step 5 In the **Operation** column of a job, click **Delete**.

Step 6 Click **OK**.

----End

10.6 GDS-Kafka Data Access

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.

1. 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**.
2. 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.

Message Formats Supported by GDS-Kafka

Table 10-3 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 10-4 GDS-Kafka import modes

Operation	app.insert.directly	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	app.insert.directly	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.

11 Cluster Log Management

Overview

Cluster logs are collected and sent to Log Tank Service (LTS). You can check or dump the collected cluster logs on LTS.

Currently, 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
- Scaling logs

NOTE

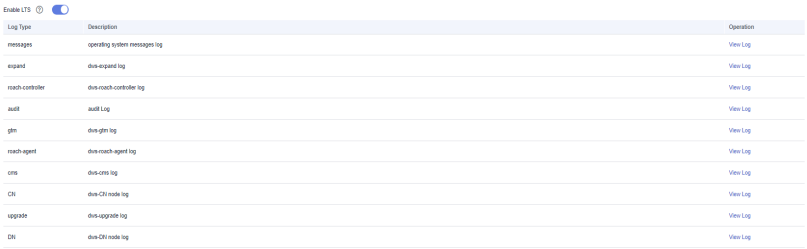
- Cluster log management depends on LTS.
- Only 8.1.1.300 and later versions support cluster log management.
- Only 8.3.0 and later versions support CMS logs, GTM logs, Roach client logs, Roach server logs, scaling logs, and upgrade logs.

Enabling LTS

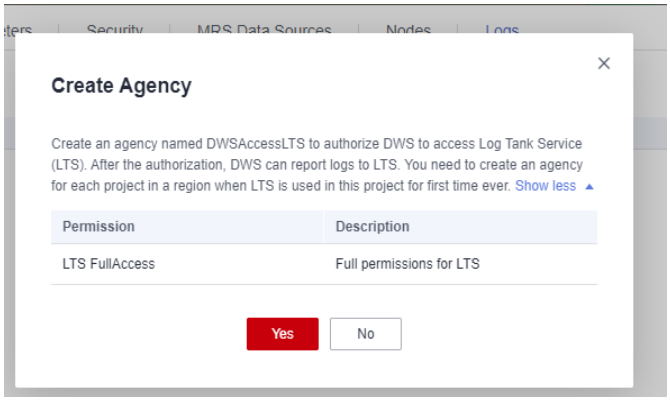
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. All clusters are displayed by default.

Step 3 Click the name of the target cluster. Choose **Logs**.



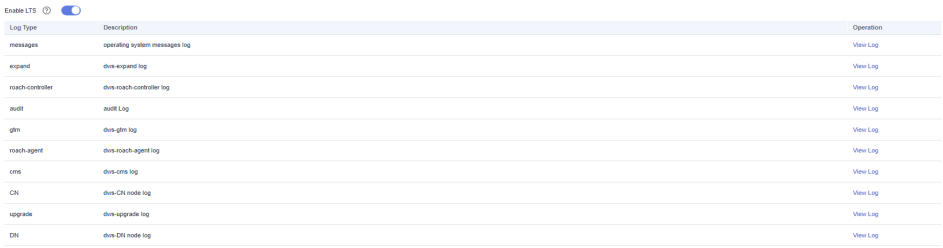
Step 4 On the **Logs** tab, enable LTS. If LTS is enabled for the first time, the following dialog box will be displayed. Confirm the information and click **Yes**.



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.

Step 5 Check the LTS status.



----End

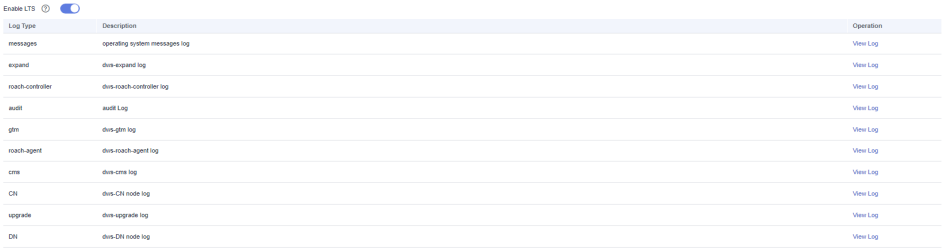
Checking Cluster Logs

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

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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.



----End

Disabling LTS

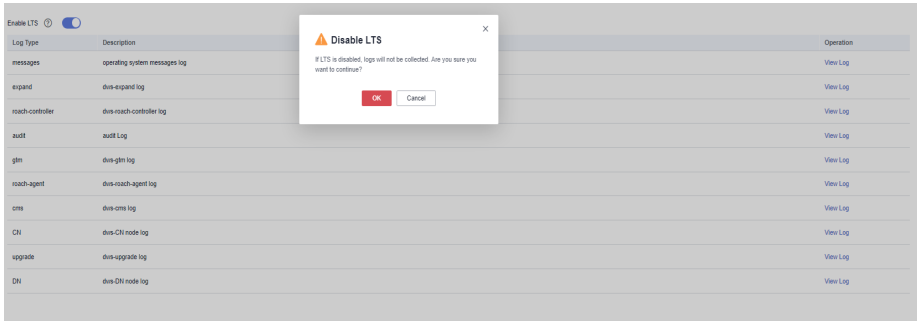
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**. 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

12 Database User Management

12.1 Managing Users

GaussDB(DWS) allows you to manage database users on the console. You can create, delete, and update database users and manage their permissions on the console.

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.

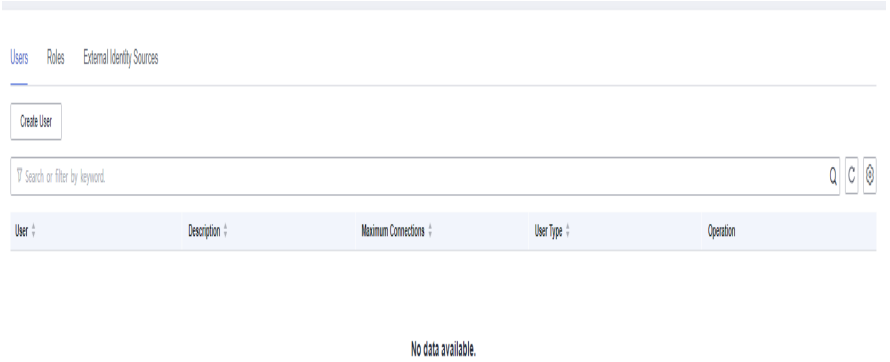
Creating a User

Step 1 Log in to the GaussDB(DWS) management 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 **User Management**.

Step 4 On the **Users** tab, click **Create User**. The user creation page is displayed.

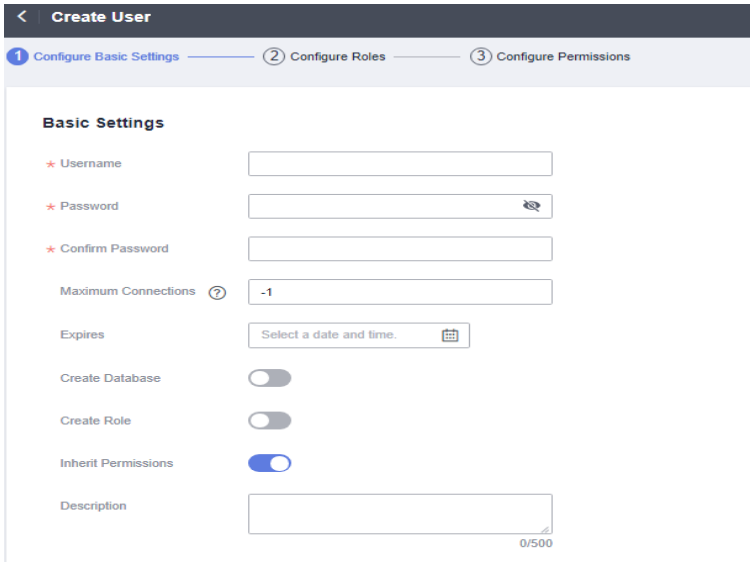


Step 5 Complete the user information as required, confirm the information, and click **Next**.

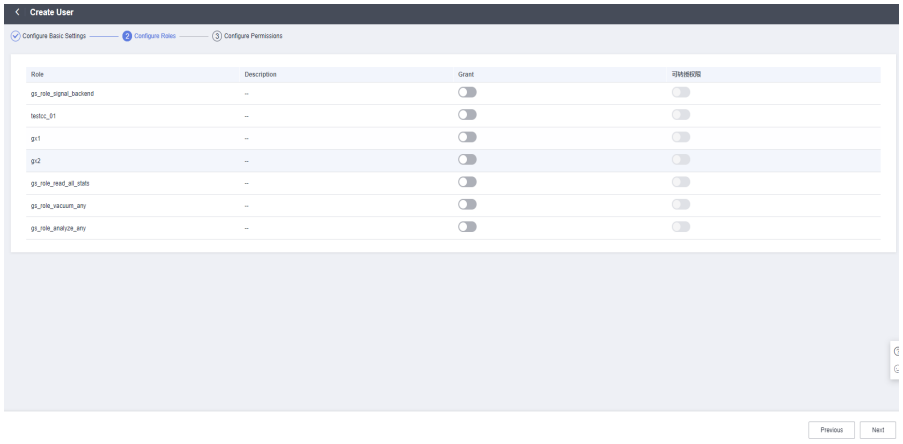
- **Username**: A username must start with a letter and can contain letters, numbers, and underscores (_). The length cannot exceed 63 characters.
- **Password**: A password must start with a letter and can contain letters, numbers, and underscores (_). The length cannot exceed 63 characters.

  NOTE

  Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\\=-)
- **Maximum Connections**: The maximum number of database connections that a user can set up. The value **-1** indicates that the number of connections is not limited.
- **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 a role inherits the permissions from its group. **This function is enabled by default. You are advised to retain this setting.**

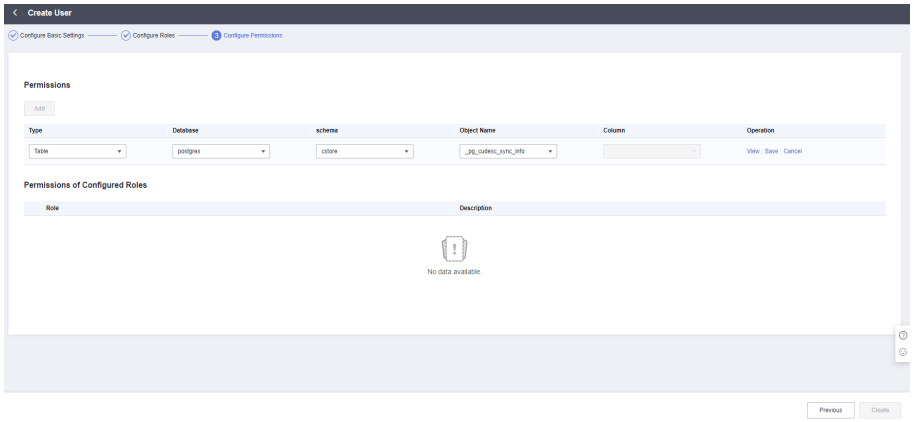


Step 6 Select the role to be granted to the user and click **Next**.



Step 7 Configure permissions not included in the roles of the user.

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 *SQL Syntax Reference*.



Step 8 After the authorization is complete, click **Create**.

----End

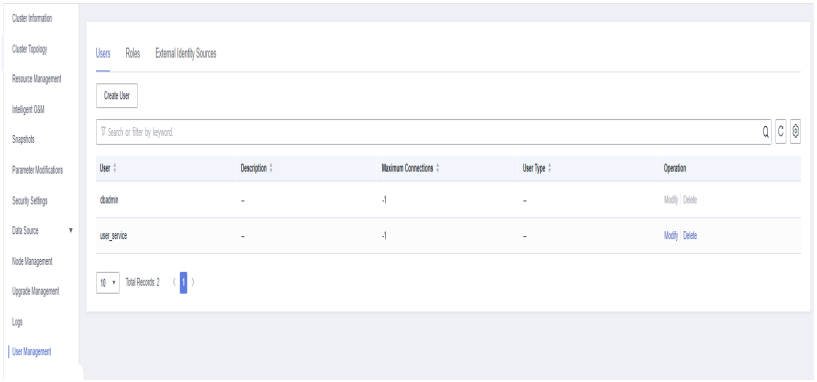
Modifying a User

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

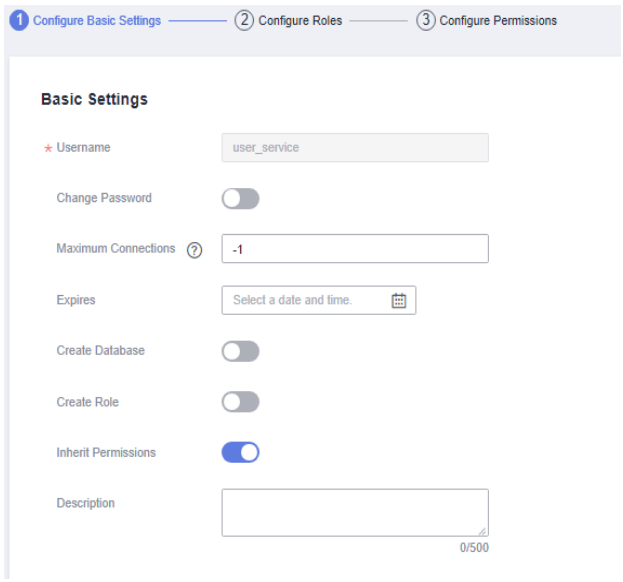
Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 In the navigation pane, choose **User Management**.

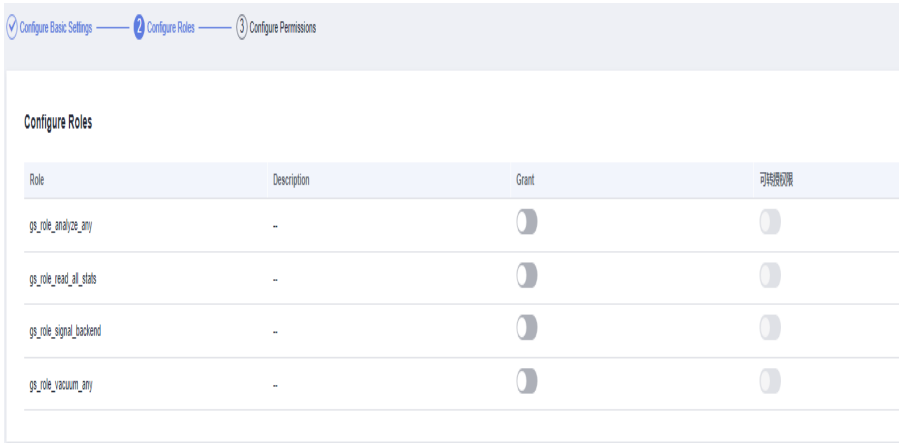
Step 4 In the user list, select a user and click **Modify**. The page for modifying user details is displayed.



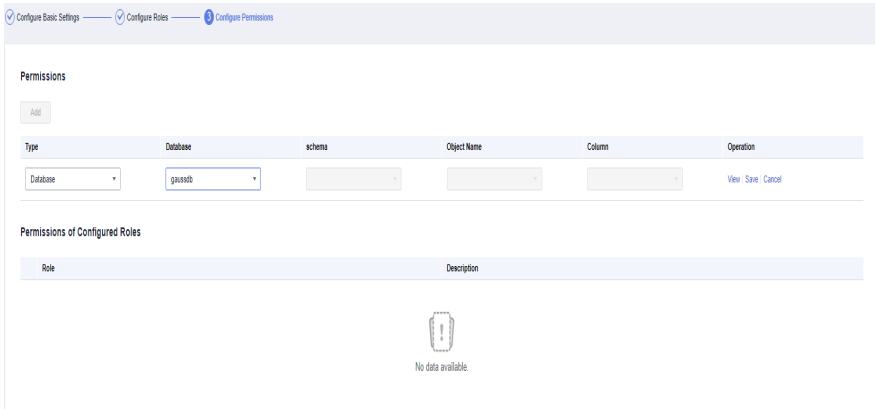
Step 5 Modify the user information. For details, see **User Information**. After confirming that the information is correct, click **Next**.



Step 6 Select the role to be granted to the user and click **Next**.



Step 7 Add or remove permissions as required.



Step 8 Confirm the permissions. Click lick **Save**.

----End

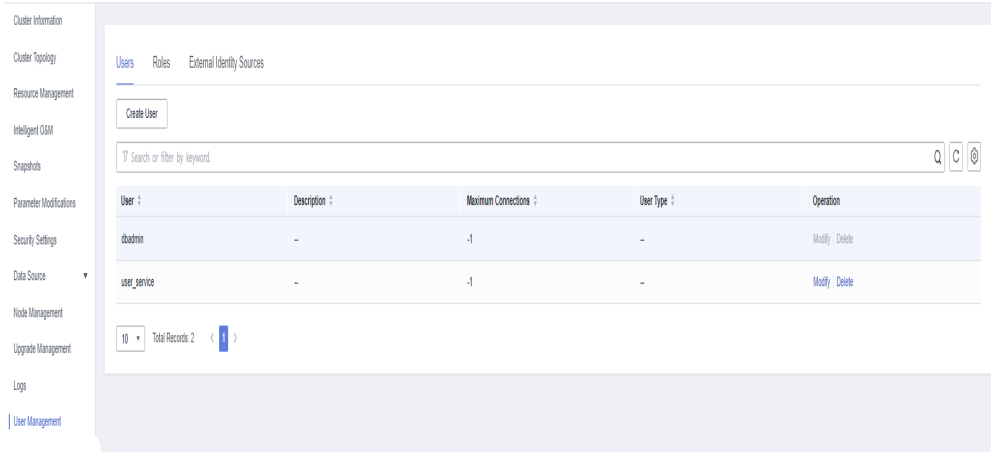
Deleting a User

Step 1 Log in to the GaussDB(DWS) management 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 **User Management**.

Step 4 Select a user from the user list and click **Delete**. A confirmation dialog box is displayed.



Step 5 Click **OK**.

NOTE

If a user has dependencies on database objects, such as tables, that have not been deleted, the user will fail to be deleted.

----End

12.2 Managing Roles

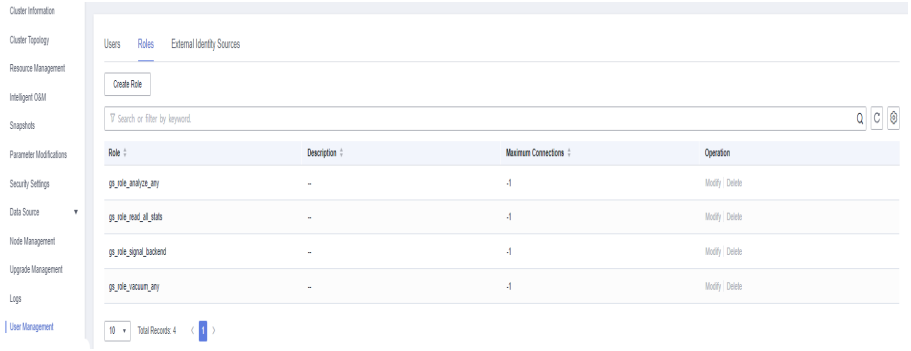
Creating a Role

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

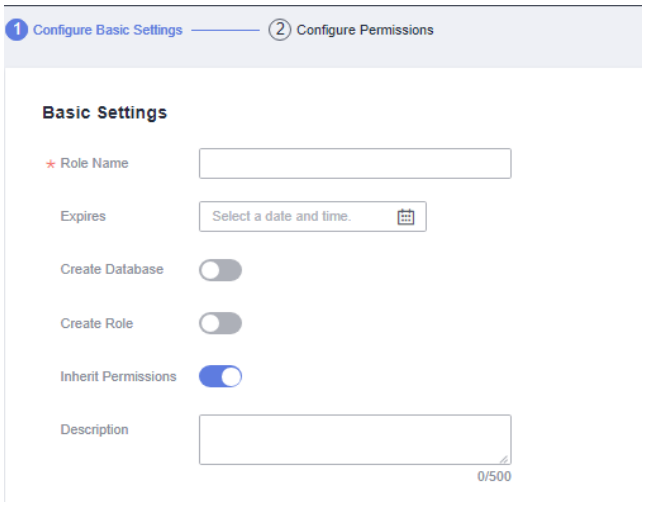
Step 3 In the navigation pane, choose **User Management**.

Step 4 Click the **Roles** tab and click **Create Role**. The role creation page is displayed.



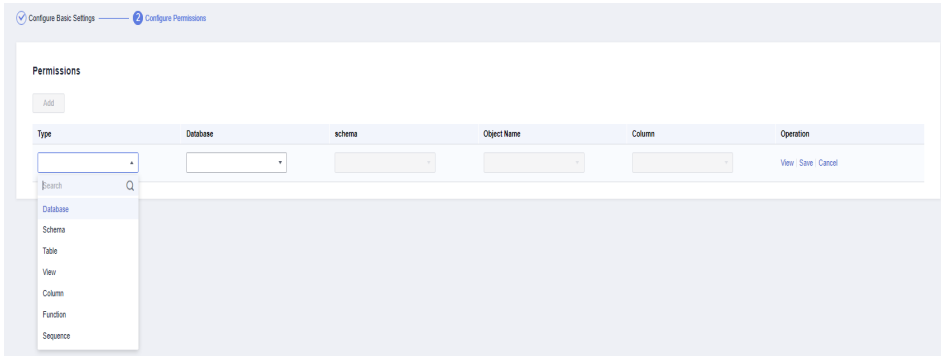
Step 5 Complete the role information, confirm the information, and click **Next**.

- **Role Name**: A username must start with a letter and can contain letters, numbers, and underscores (_). The length cannot exceed 63 characters.
- **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 a role inherits the permissions from its group. This function is enabled by default. You are advised to retain this setting.



Step 6 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 *SQL Syntax Reference*.



Step 7 After the authorization is complete, click **Create**. The role is created.

----End

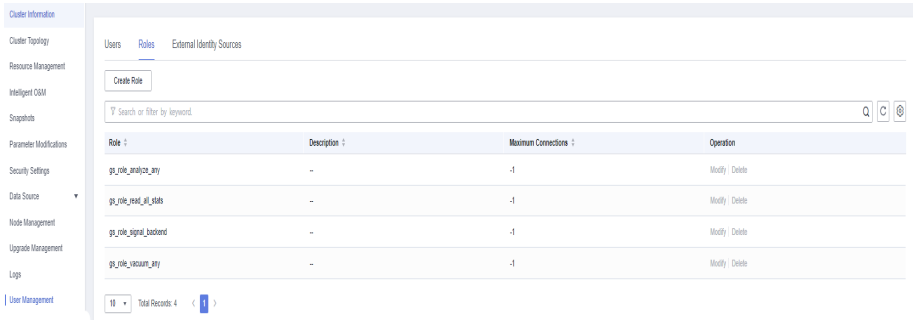
Modifying a Role

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

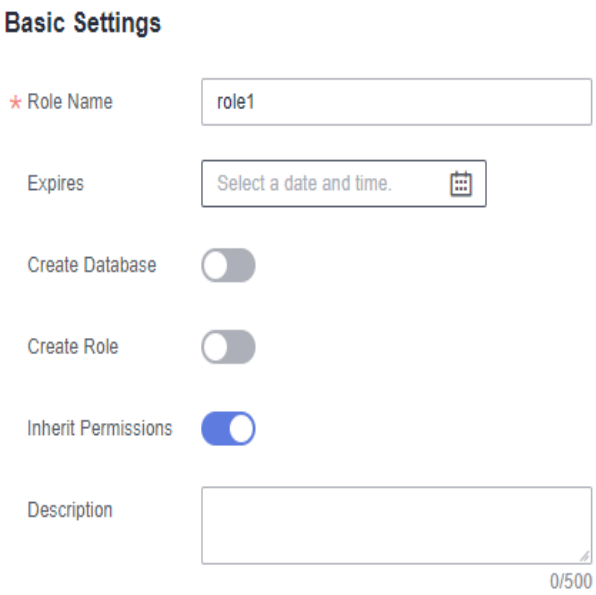
Step 3 In the navigation pane, choose **User Management**.

Step 4 In the role list, select a user and click **Modify**. The page for modifying role details is displayed.

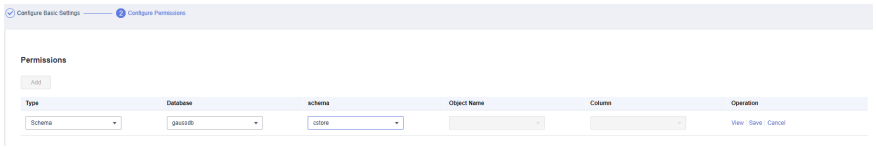


Step 5 Modify the role information, confirm the information, and click **Next**.

- **Role Name**: A username must start with a letter and can contain letters, numbers, and underscores (_). The length cannot exceed 63 characters.
- **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 a role inherits the permissions from its group. This function is enabled by default. You are advised to retain this setting.



Step 6 Add or remove permissions as required.



Step 7 Confirm the permissions. Click lick **Save**.

----End

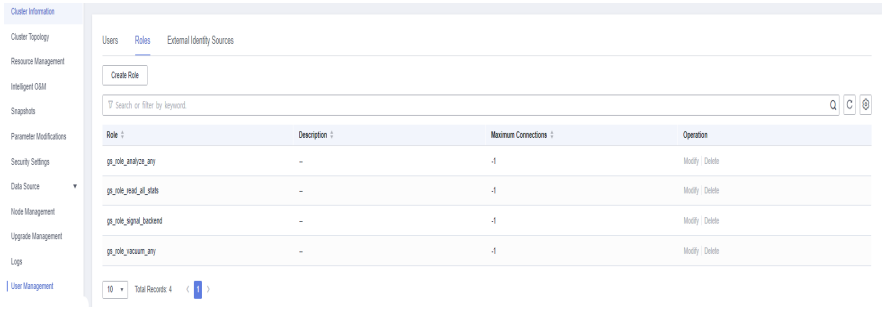
Deleting a Role

Step 1 Log in to the GaussDB(DWS) management 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 **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.

NOTE

If the role has dependencies, such as database objects, that have not been deleted, the role will fail to be deleted.

----End

13 Audit Logs

13.1 Audit Log Overview

GaussDB(DWS) provides management console audit logs and database audit logs for users to query service logs, analyze problems, and learn product security and performance status.

Management Console Audit Logs

GaussDB(DWS) uses Cloud Trace Service (CTS) to record mission-critical operations performed on the GaussDB (DWS) management 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 details about how to enable and view management console audit logs, see **Management Console Audit Logs**.

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 details about how to enable and view database audit logs, see **Configuring the Database Audit Logs** and **Viewing Database Audit Logs**.

13.2 Management Console Audit Logs

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 have enabled CTS, the system has automatically created a management tracker. Only one management tracker can be created and it cannot be deleted.

----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.

----End

Key Operations

With CTS, you can record operations associated with GaussDB(DWS) for later query, audit, and backtrack operations.

NOTE

The creation and deletion of automatic snapshots are not performed by users, therefore not recorded in audit logs.

Table 13-1 GaussDB(DWS) operations that can be recorded by CTS

Operation	Resource	Event Name
Creating/Restoring a cluster	cluster	createCluster
Deleting a cluster	cluster	deleteCluster
Scaling out a cluster	cluster	resizeCluster
Restarting a cluster	cluster	restartCluster
Creating a snapshot	backup	createBackup
Deleting a snapshot	backup	deleteBackup
Setting security parameters	configurations	updateConfigurations

Operation	Resource	Event Name
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.

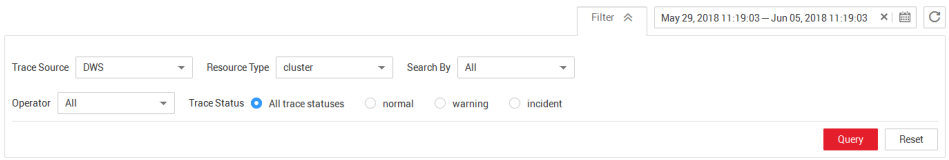
Step 2 In the navigation pane on the left, choose **Trace List**.

Step 3 In the upper right corner of the trace list, click **Filter** to set the search criteria.

The following filters are available:

- **Trace Source**, **Resource Type**, and **Search By**
  - **Trace Source**: Select **GaussDB(DWS)**.
  - **Resource Type**: Select **All resource types** or specify a resource type.
  - **Search By**: Select **All filters** or any of the following options:
    - **Trace name**: If you select this option, you also need to select a specific trace name.
    - **Resource ID**: If you select this option, you also need to select or enter a specific resource ID.
    - **Resource name**: If you select this option, you also need to select or enter a specific resource name.
- **Operator**: Select a specific operator (at user level rather than tenant level).
- **Trace Status**: Available options include **All trace statuses**, **normal**, **warning**, and **incident**. You can only select one of them.
- **Start Date** and **End Date**: You can specify the time period to query traces.

Figure 13-1 Querying traces



Step 4 Click **Query**.

Step 5 Click on the left of the trace to be queried to extend its details.

Figure 13-2 Traces

Step 6 Locate the row containing the target trace and click **View Trace** in the **Operation** column.

Figure 13-3 Viewing a trace


```
{
    "time": "Jun 01, 2018 17:44:16 GMT+08:00",
    "user": {
        "name": "DWS-hec1",
        "id": "f282510b8ca14d0fb766aa216aa0a764",
        "domain": {
            "name": "DWS-hec1",
            "id": "9e57dcaa89164a149f1b5f7130c49c52"
        }
    },
    "request": {},
    "response": {},
    "code": 500,
    "service_type": "DWS",
    "resource_type": "cluster",
    "resource_name": "wwddggmm",
    "resource_id": "ebcd6e8f-7aca-4db4-94f3-c4ae86faf9cf",
    "source_ip": "[redacted]",
    "trace_name": "createClusterConnection",
    "trace_type": "ConsoleAction",
    "api_version": "v1.0",
    "record_time": "Jun 01, 2018 17:43:48 GMT+08:00",
    "trace_id": "488975c0-6580-11e8-800d-5b47da5b8b28",
    "trace_status": "warning"
}
```


For details about the key fields in the CTS trace structure, see "Trace References > Trace Structure" and "Trace References > Example Traces" in the *Cloud Trace Service User Guide*.

----End

13.3 Database Audit Logs

13.3.1 Configuring the Database Audit Logs

Prerequisites

Database audit logs are configured on the **Security Settings** page. You can change security settings only when the cluster status is **Available** and **Unbalanced**, and **Task Information** cannot be **Creating snapshot**, **Scaling out**, **Configuring**, or **Restarting**.

Procedure

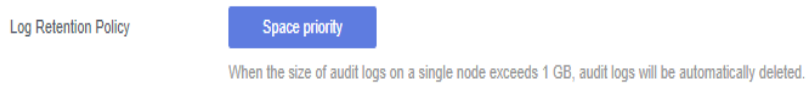
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Cluster** > **Dedicated Cluster**.

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 results are displayed.

Step 4 In the **Audit Settings** area, configure the audit log retention policy.



Space priority: Audit logs will be automatically deleted if the size of audit logs on a single node exceeds 1 GB.

CAUTION

- Clusters 1.0.0 and 1.1.0 do not support audit log retention.
- If the planned storage space of the database is limited, select **Space priority** to prevent faulty nodes or low performance caused by insufficient disk space.

Step 5 Enable the audit function for the following operations if necessary.

NOTE

Fine-grained audit items are supported in 8.1.1.100 or later.

Figure 13-4 Audit items

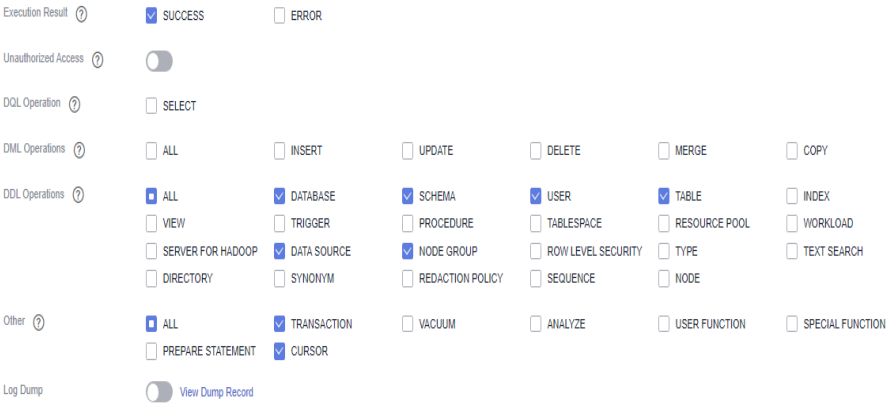


Table 13-2 describes the detailed information about the audit items.

Table 13-2 Audit items

Audit Item	Description
Unauthorized access	Specifies whether to record unauthorized operations. This parameter is disabled by default.

Audit Item	Description
DQL operations	**SELECT** operations can be selected. **NOTE** This parameter is supported by 8.1.1.100 or later.
DML operations	Specifies whether to record **INSERT**, **UPDATE**, and **DELETE** operations on tables. This parameter is disabled by default. **NOTE** The cluster supports fine-grained audit items in 8.1.1.100 or later. **COPY** and **MERGE** are added.
DDL operations	Specifies whether to record the **CREATE**, **DROP**, and **ALTER** operations of specified database objects. **DATABASE**, **SCHEMA**, and **USER** are selected by default. **NOTE** The cluster supports **TABLE**, **DATA SOURCE**, and **NODE GROUP** operations in 8.1.1.100 or later. These operations are enabled by default.
Other operations	Specifies whether to record other operations. Only the **TRANSACTION** and **CURSOR** operations are selected by default. **NOTE** • This parameter is supported by 8.1.1.100 or later. • 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 13-2**, key audit items in **Table 13-3** are enabled by default on GaussDB(DWS).

Table 13-3 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 6 Enable or disable audit log dumps.

For more information, see **Enabling Audit Log Dumps**.

Step 7 Click **Apply**.

Click . The configuration status **Applying** indicates that the configurations are being saved.

When the status changes to **Synchronized**, the configurations are saved and take effect.

----End

13.3.2 Dumping the Database Audit Logs

GaussDB(DWS) records information (audit logs) about connections and user activities in your database. With the information, you can monitor the database to ensure security and facilitate fault troubleshooting and historical operation record locating. These audit logs are stored in the database by default. You can also dump them to OBS so that users who are responsible for monitoring the database can view the logs more conveniently.

NOTE

- This function cannot be used if OBS is not available.
- 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) management 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**.

Step 4 In the **Audit Settings** area, enable **Audit Log Dump**.

 indicates that the function is enabled.  indicates that the function is disabled.

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 13-5 Enabling audit log dumps

- **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.

Wait for a moment and then refresh **Configuration Status**. When **Configuration Status** is **Synchronized**, the configuration is saved and takes effect.

----End

Modifying Audit Log Dump Configurations

After audit log dump is enabled, you can modify the dump configurations, for example, modifying the OBS bucket, path, and dump interval.

The procedure is as follows:

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

Wait for a moment and then refresh **Configuration Status**. When **Configuration Status** is **Synchronized**, the configuration is saved and takes effect.

----End

Viewing Audit Log Dumps

After audit log dump is enabled, you can view the dumped audit logs on OBS.

The procedure is as follows:

Step 1 Log in to the GaussDB(DWS) management 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 13-4 Log file fields

Field	Type	Description
begintime	timestamp with time zone	Operation start time
endtime	timestamp with time zone	Operation end time

Field	Type	Description
operation_type	text	Operation type. For details, see **Table 13-5**.
audit_type	text	Audit type. For details, see **Table 13-6**.
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
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

Table 13-5 Operation types

Operation Type	Description
audit_switch	Indicates that the operations of enabling and disabling the audit log function are audited.
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.

Operation Type	Description
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.
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.

Operation Type	Description
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 13-6 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.
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.

Parameter	Description
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.
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.

Parameter	Description
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 Dumps

You can disable audit log dumps if you do not want to dump audit logs to OBS.

The procedure is as follows:

Step 1 Log in to the GaussDB(DWS) management 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 disable audit log dump. In the navigation pane, choose **Security**.

Step 4 In the **Audit Settings** area, disable audit log dump.

 indicates that the function is disabled.

Step 5 Click **Apply**.

If **Configuration Status** is **Applying**, the system is saving the settings.

Wait for a moment and then refresh **Configuration Status**. When **Configuration Status** is **Synchronized**, the configuration is saved and takes effect.

----End

13.3.3 Viewing Database Audit Logs

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 Database Parameters**.
- 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.

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 **Viewing 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 **Cluster Connection**.

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 startime,timestamptz endtime,audit_log)
```

startime 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 startime,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

14 Cluster Security Management

14.1 Configuring Separation of Permissions

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.

GaussDB(DWS) supports role-based separation of permissions. In this way, different roles have different permissions and cluster data can be better protected.

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 GaussDB(DWS) 3.0 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**, **To be restarted**, or **Unbalanced**.
- The **Task Information** cannot be **Creating snapshot**, **Scaling out**, **Configuring**, or **Restarting**.

Procedure

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

 indicates that the function is enabled. 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.

 indicates that **Rights Separation** is disabled. **Rights Separation** is disabled by default.

Figure 14-1 Security configuration

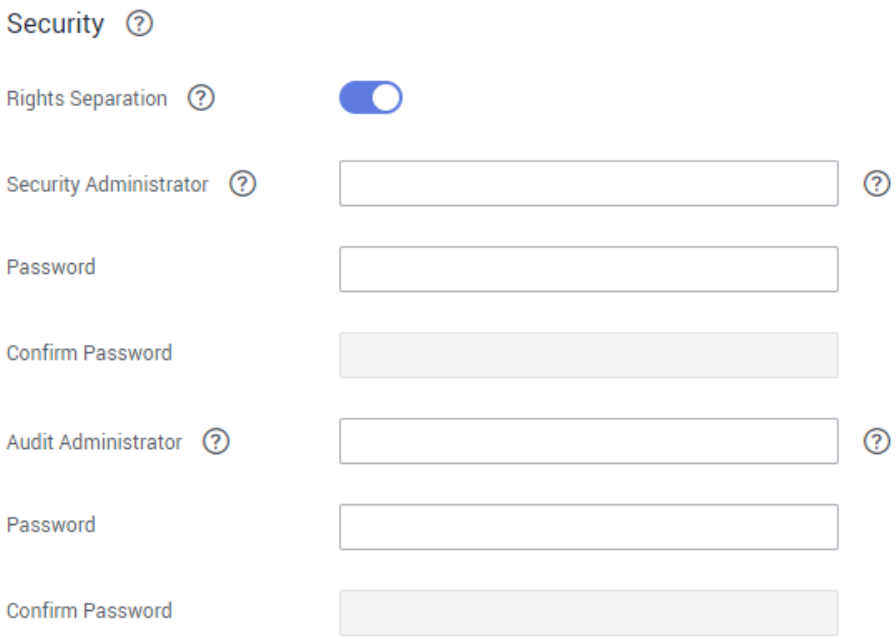


Table 14-1 Security parameters

Parameter	Description	Example Value
Security Administrator	The username must meet the following requirements: • Consists of lowercase letters, digits, or underscores. • Starts with a lowercase letter or an underscore. • Contains 6 to 64 characters.	security_admin
Password	The password complexity requirements are as follows: • Contain 12 to 32 characters. • Cannot be the username or the username spelled backwards. • Contains at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters (~!?,.:;_(){}[]/<>@#%^&*+|\\=-) • Passes the weak password check.	-
Confirm Password	Enter the password of the security administrator again.	-
Audit Administrator	The username must meet the following requirements: • Consists of lowercase letters, digits, or underscores. • Starts with a lowercase letter or an underscore. • Contains 6 to 64 characters.	audit_admin
Password	The password complexity requirements are as follows: • Contain 12 to 32 characters. • Cannot be the username or the username spelled backwards. • Must contain at least 3 of the following character types: uppercase letters, lowercase letters, digits, and special characters ~!@#%^&*()-_=+|[{}];:,<.>/? • Passes 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

14.2 Encrypting Databases

14.2.1 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 (or in reverse), you need to export all data from the unencrypted cluster and import it to a new cluster that has enabled database encryption. Database encryption is performed when data is written to GaussDB(DWS). That is, GaussDB(DWS) encrypts data when the data is written to GaussDB(DWS). If you want to query the data, GaussDB(DWS) automatically decrypts it and returns the result to you.

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

- The GaussDB(DWS) 3.0 cluster does not support database encryption.
- The database encryption function can be enabled or disabled only when a cluster is created. It cannot be enabled after a cluster is created. Once enabled, it cannot be disabled. For details, see **Encrypting the Database**.
- 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) management console.

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

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 14-2**.

Table 14-2 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.

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.

14.2.2 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. Each key rotation will update the CEK once. During the key rotation, the cluster is still in **Available** status.

Rotating Encryption Keys for Data Warehouse Clusters

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

Step 2 In the navigation tree on the left, choose **Cluster > Dedicated Cluster**.

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

14.2.3 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

By default, clusters (versions 8.1.3.325 and later, and 8.2.1.105 and later) created on the console support encryption. For old clusters later than 8.0.x, contact technical support to upgrade them.

Procedure

Step 1 Log in to the GaussDB(DWS) management console. In the navigation pane, 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.

NOTE

If **Convert to Encrypted Cluster** is not displayed, the cluster (earlier version) cannot be converted to an encrypted cluster or the cluster is already an encrypted cluster.

Step 3 In the dialog box that is displayed, select the key source and encryption algorithm to convert the cluster into an encrypted cluster.

- Method 1: Select a key name.
- Method 2: Enter the key ID. Enter the key ID used for authorizing the current tenant..

  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**.

Table 14-3 Parameter description

Parameter	Description
Key Source	You can select a key name from the key list or directly enter a key name.
Cryptographic Algorithm	The encryption algorithms are as follows: • 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

14.3 Permissions

14.3.1 Creating a User and Granting GaussDB(DWS) Permissions

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, you may skip this section.

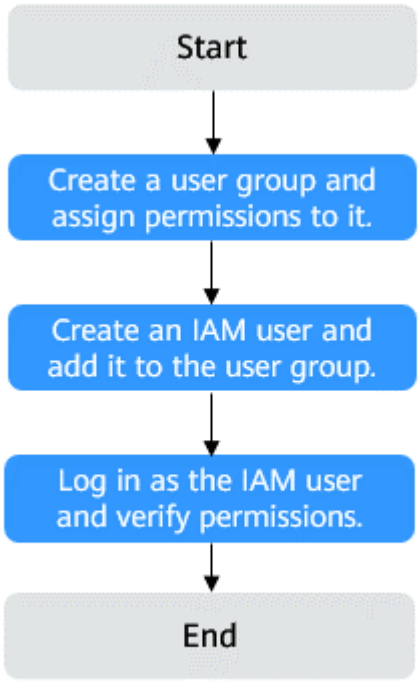
This section describes the procedure for granting permissions (see **Procedure**).

Prerequisites

Before granting permissions to a user group, familiarize yourself with the GaussDB(DWS) system permissions that can be added to the user group. For details, see **GaussDB(DWS) system permissions**.

Procedure

Figure 14-2 Procedure



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.

14.3.2 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, visit **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"
            ]
           }

       ]
}
```

14.3.3 Syntax of Fine-Grained Permissions Policies

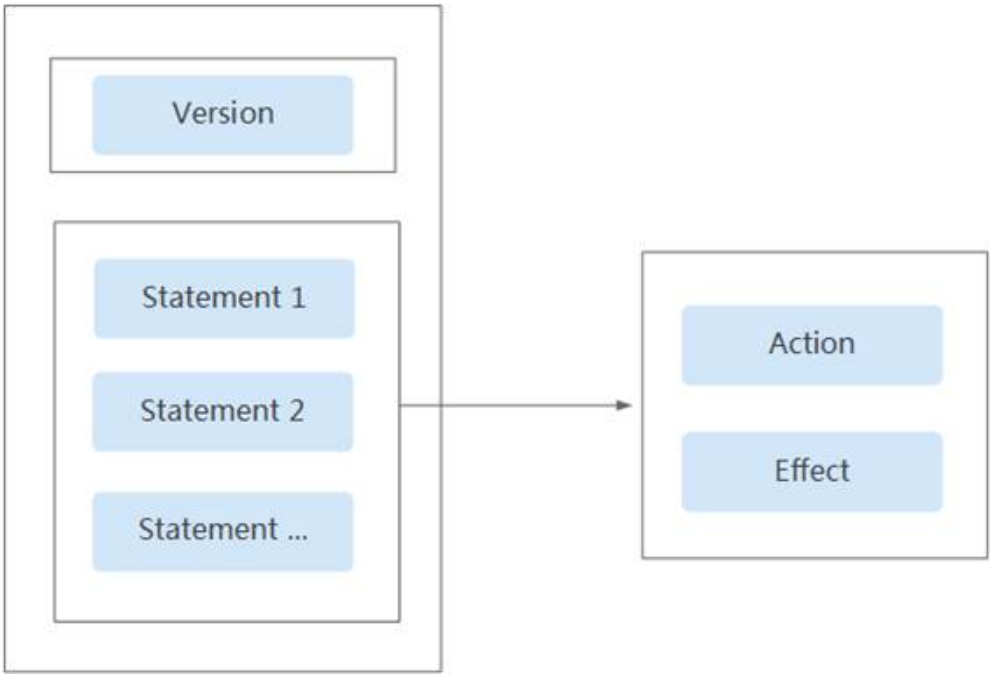
In actual services, you may need to grant different operation permissions on resources to users of different roles. The IAM service provides fine-grained access control. An IAM administrator (a user in the **admin** group) can create a custom policy containing required permissions. After a policy is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

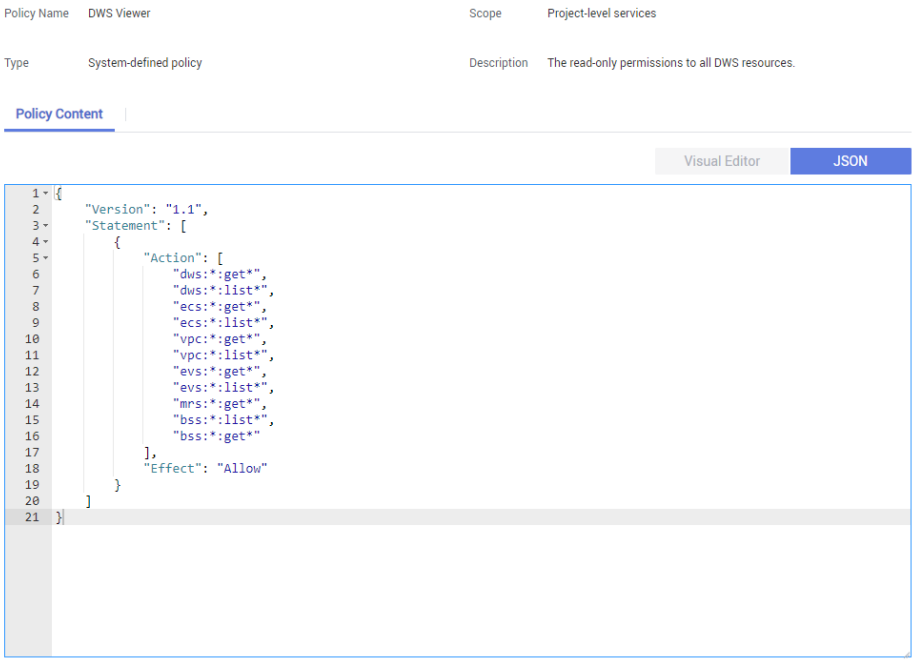
Figure 14-3 Policy structure



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 14-4 Setting the policy




```
{
    "Version": "1.1",
    "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.

- **Statement**: Permissions defined by a policy, including Effect and Action.
  - Effect

    The valid values for Effect are Allow and Deny. System policies contain only Allow statements. For custom policies containing both Allow and Deny statements, the Deny statements take precedence.
  - Action

    Permissions in the format of *Service name:Resource type:Operation*. A policy can contain one or more permissions. The wildcard (*) is allowed to indicate all of the services, resource types, or operations depending on its location in the action.

    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 RESTful API actions supported by GaussDB(DWS), see **Permissions Policies and Supported Actions**.
- **Management console operations**

  Table 14-4 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.

Table 14-4 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"	--	• Scope: – Project – Enterprise project
Obtaining the details of a cluster	"dws:cluster:getDetail"	"dws:*:get*", "dws:*:list*", "vpc:vpcs:list", "vpc:securityGroups:get"	• Scope: – Project – Enterprise project
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

Operation	Permission	Dependent Permission	Scope
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*",	• Scope: – Project – Enterprise 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 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

Operation	Permission	Dependent Permission	Scope
Querying whether DAS is supported	"dws:supportDas:list"	"dws:*:get*", "dws:*:list*", "das:*:get*", "das:*:list*",	• 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
Querying user tables	"dws:tables:list"	"dws:*:get*", "dws:*:list*",	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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
Terminating a session	"dws:dmsSession:terminate"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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
Adding a SQL probe	"dws:dmsProbe:add"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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
Terminating a query	"dws:dmsQuery:terminate"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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"	• Scope: – Project – Enterprise 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
Executing a SQL probe	"dws:dmsProbe:execute"	"dws:dmsGrpcOuter:operation"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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
Deleting an alarm record	"dws:dmsAlarmRecord:delete"	"dws:dmsQuery:list"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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
Querying intelligent O&M information	"dws:operationalTask:get"	"dws:*:get*", "dws:*:list*"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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
Resource management operations	"dws:workLoadManager:operate"	"dws:*:get*", "dws:*:list*"	• Scope: – Project – Enterprise project

Operation	Permission	Dependent Permission	Scope
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*"	• Scope: – Project
Querying event specifications	"dws:event:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying event subscriptions	"dws:eventSub:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Creating an event subscription	"dws:eventSub:create"	"dws:*:get*", "dws:*:list*",	• Scope: – Project
Updating an event subscription	"dws:eventSub:update"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Deleting an event subscription	"dws:eventSub:delete"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying alarm statistics	"dws:alarmStatistic:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying alarm details	"dws:alarmDetail:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying alarm configurations	"dws:alarmConfig:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying alarm subscriptions	"dws:alarmSub:list"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Creating an alarm subscription	"dws:alarmSub:create"	"dws:*:get*", "dws:*:list*",	• Scope: – Project

Operation	Permission	Dependent Permission	Scope
Updating an alarm subscription	"dws:alarmSub:update"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Deleting an alarm subscription	"dws:alarmSub:delete"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Delivering cluster upgrade operations (upgrade, rollback, submission, and retry)	"dws:cluster:doUpdate"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying the available upgrade paths of a cluster	"dws:cluster:getUpgradePaths"	"dws:*:get*", "dws:*:list*"	• Scope: – Project
Querying cluster upgrade records	"dws:cluster:getUpgradeRecords"	"dws:*:get*", "dws:*:list*"	• Scope: – Project

Authorization Using the Fine-Grained Permission Policy

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.
- 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.

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.

----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 14-5 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.

14.3.4 RBAC Syntax of RBAC Policies

Policy Structure

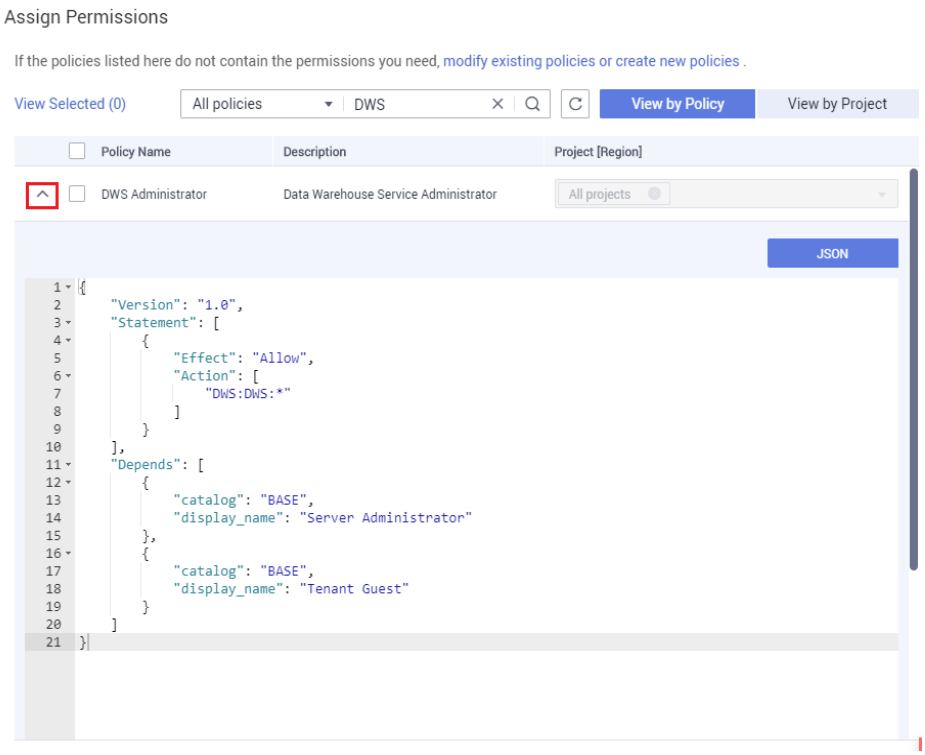
An RBAC policy consists of a Version, a Statement, and Depends.

Figure 14-6 RBAC policy structure

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 14-7 Syntax of RBAC Policies




```
{
    "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**.
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).

Parameter		Meaning	Value
	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.

14.4 Protection for Mission-Critical Operations

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: scaling out a cluster, deleting a cluster, restarting a cluster, adding a CN, and deleting a CN.

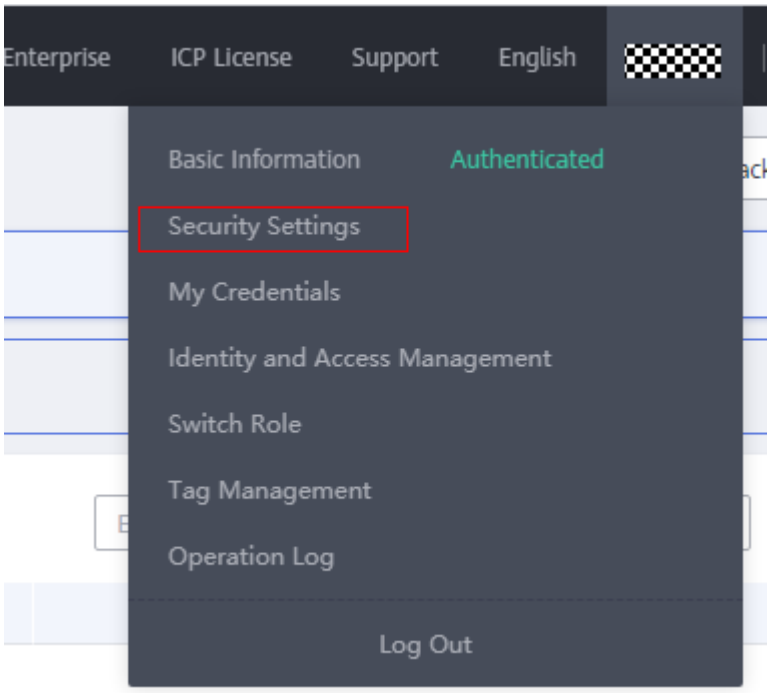
Enabling Operation Protection

Operation protection is disabled by default. To enable it, perform the following steps:

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

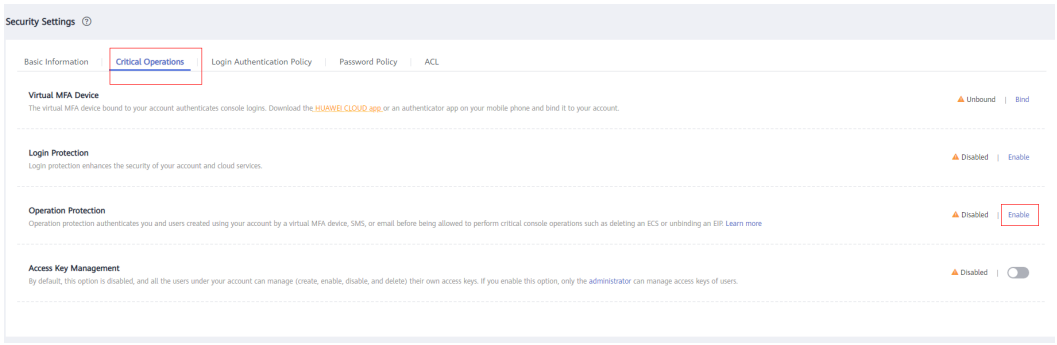
Step 2 Move the cursor to the username in the upper right corner of the page and click **Security Settings** from the drop-down list.

Figure 14-8 Security Settings



Step 3 On the **Security Settings** page, click the **Critical Operations** tab. Click **Enable** in the **Operation Protection** area.

Figure 14-9 Critical Operations



Step 4 On the **Operation Protection** page, select **Enable** to enable operation protection.

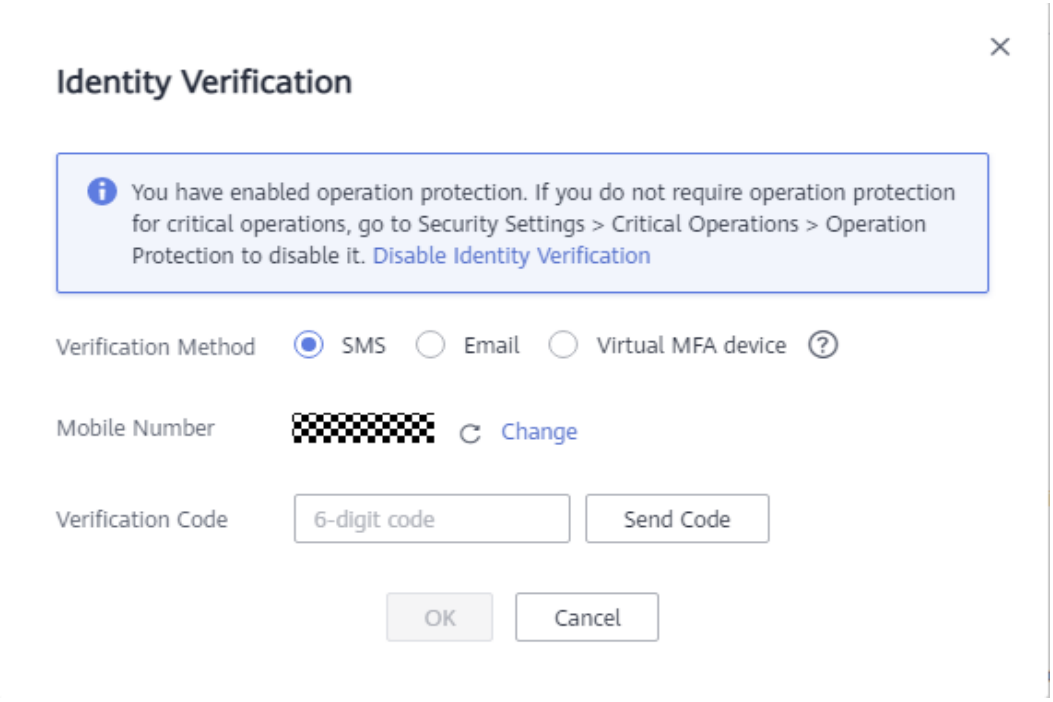
NOTE

- 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 a user is not associated with an email address, mobile number, or virtual MFA device, the user will need to associate an email address, mobile number, or virtual MFA device with their account before the user can perform any critical operations.
- Change your phone number or email address for verification in **My Account** on the management console.

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.

Figure 14-10 Identity Verification



----End

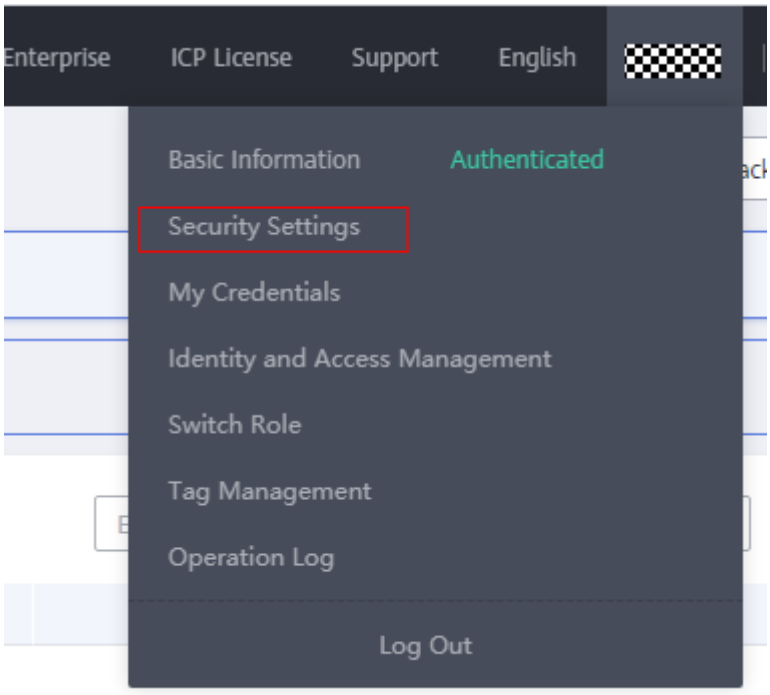
Disabling Operation Protection

To disable operation protection, perform the following steps:

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

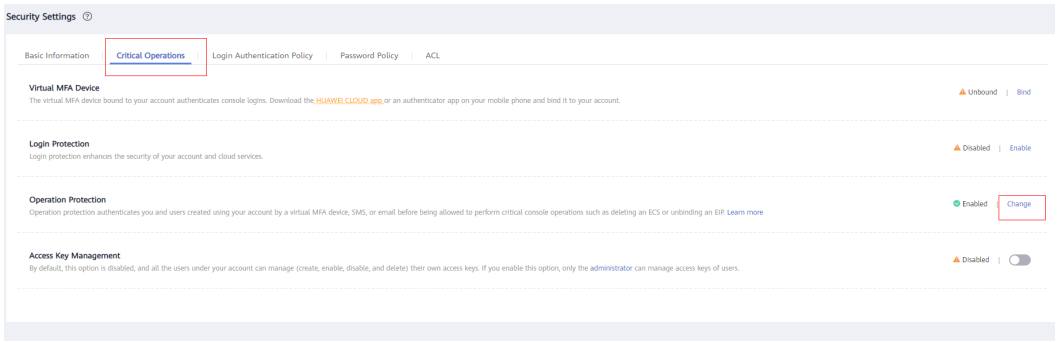
Step 2 Move the cursor to the username in the upper right corner of the page and click **Security Settings** from the drop-down list.

Figure 14-11 Security Settings



Step 3 On the **Security Settings** page, click the **Critical Operations** tab. Click **Change** in the **Operation Protection** area.

Figure 14-12 Modifying operation protection settings



Step 4 On the **Operation Protection** page, select **Disable** and click **OK**.

----End

15 Resource Management

15.1 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.

NOTICE

- This feature is supported only in 8.0 or later.
- The hybrid data warehouse (standalone) does not support resource management.

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 workload 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.

15.2 Resource Pool

15.2.1 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.

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.

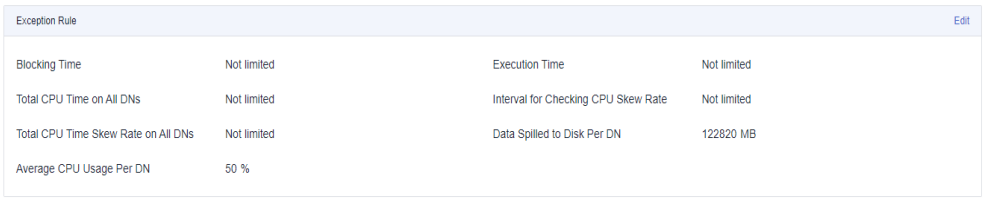
Choose either of the preceding management methods as needed. In CPU share management, CPUs can be shared and fully utilized, but resource pools are not isolated and may affect the query performance of each other. In CPU limit management, the CPUs of different resource pools are isolated, but this may result in the waste of idle resources.

NOTE

Only 8.1.3 and later versions support the CPU limit management.

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.



The following table describes exception rules.

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.

Table 15-1 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**

Parameter	Description	Value Range (0 Means No Limit)	Operation
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**
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**

15.2.2 Page Overview

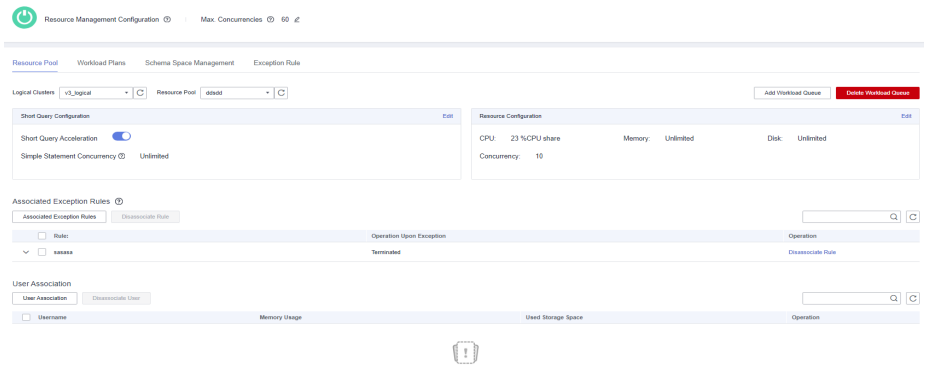
Overview

On the resource management page, you can modify global resource management configurations; add, create, and modify resource queues; add database users to a resource pool; and remove a database user from a resource pool. After a cluster is converted into a logical cluster, you can create, modify, or delete a resource pool in the logical cluster.

The page consists of the following modules:

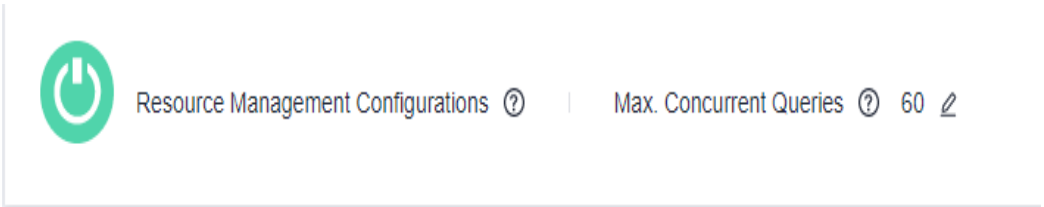
- **Enabling or Disabling Resource Management**
- **Short Query Configuration**

- Resource Configuration
- Associated Exception Rules
- User Association



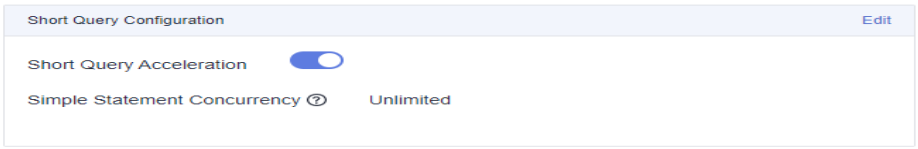
Enabling or Disabling Resource Management

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.



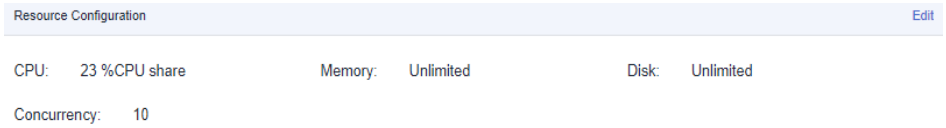
Short Query Configuration

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.



Resource Configuration

In the **Resource Configuration** area, you can view the resource configuration of the current workload queue. For example:



NOTE

Only clusters of the version 8.2.1 and later support the network bandwidth weight.

Associated Exception Rules

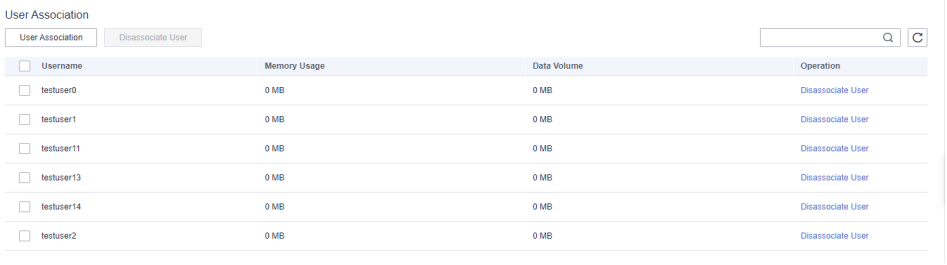
In the **Associated Exception Rules** area, you can view the exception rules associated to the current resource pool, associate new exception rules, and disassociate exception rules. For more information, see **Exception parameters**.

NOTE

Only clusters of 8.2.0 and later versions support associating exception rules. GaussDB(DWS) 3.0 does not support this function. For cluster versions earlier than 8.2.0, see **Step 7.3**.

User Association

In the **Associated User** list, you can view the associated users of the current resource pool, and the memory and disk usage of each user at the current time, as shown in the following figure.



NOTE

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.

15.2.3 Creating a Resource Pool

Step 1 Log in to the GaussDB(DWS) management 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 15-2**.

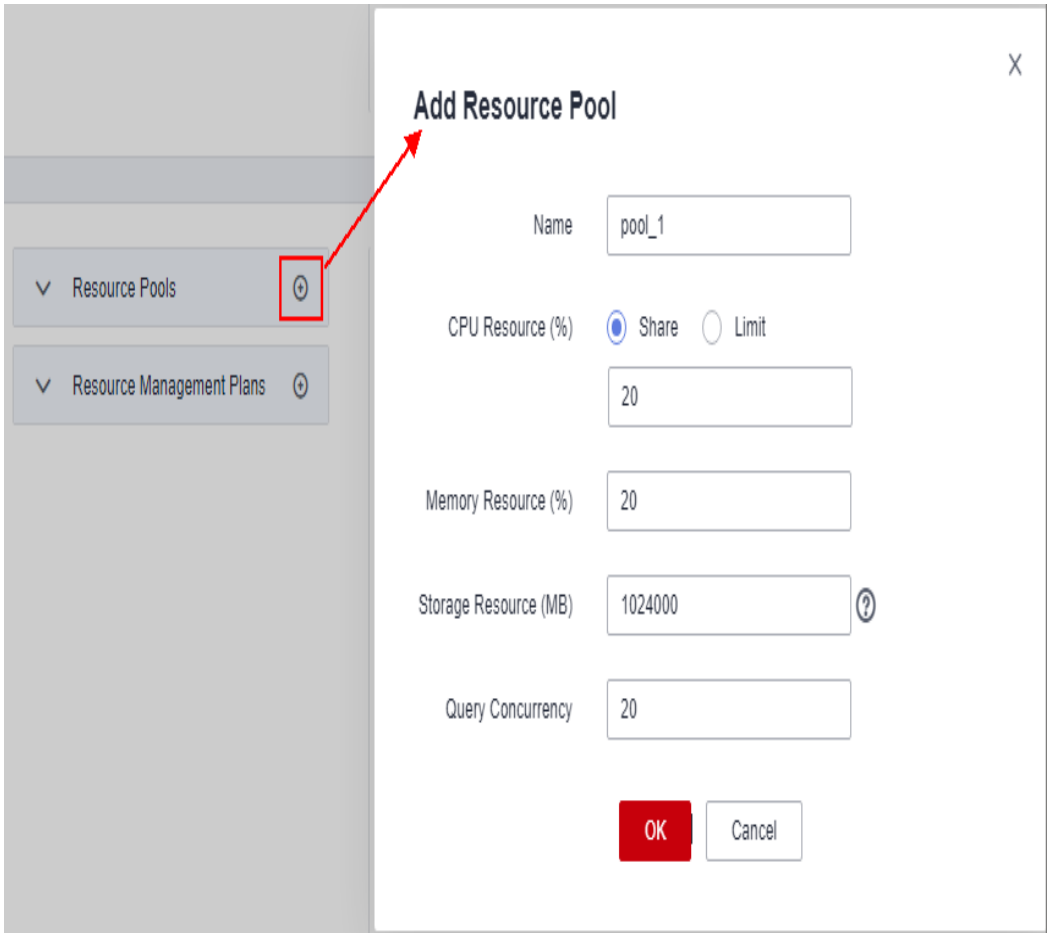


Table 15-2 Resource pool parameters

Parameter	Description	Mandatory	Default Value
Name	Resource pool name	Yes	-

Parameter	Description	Mandatory	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.	Yes	-
Memory Resource (%)	Percentage of the memory that can be used by a resource pool. **CAUTION** 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.	Yes	0 (not limited)
Storage Resource (MB)	Size of the available space for permanent tables. **CAUTION** This parameter indicates the total tablespace of all DNs in a resource pool. Available space of a single DN = Configured value/Number of DNs.	Yes	-1 (not limited)
Complex Statement Concurrency	Maximum number of concurrent queries in a resource pool. **CAUTION** 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.	Yes	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 cluster 8.2.1 and later versions support the network bandwidth weight.	Yes	-1 (not limited)

Step 6 Confirm the information and click **OK**.

----End

15.2.4 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) management 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. The following configuration areas are displayed, including **Short Query Configuration**, **Resource Configuration**, **Exception Rule**, 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
Concurrent Short Queries	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 15-3**.

Table 15-3 Resource pool parameters

Parameter	Description	Mandatory	Default Value
Name	Resource pool name	Yes	-

Parameter	Description	Mandatory	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.	Yes	-
Memory Resource (%)	Percentage of the memory that can be used by a resource pool. **CAUTION** 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.	Yes	0 (not limited)
Storage Resource (MB)	Size of the available space for permanent tables. **CAUTION** This parameter indicates the total tablespace of all DNs in a resource pool. Available space of a single DN = Configured value/Number of DNs.	Yes	-1 (not limited)
Complex Statement Concurrency	Maximum number of concurrent queries in a resource pool. **CAUTION** 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.	Yes	10

Parameter	Description	Mandatory	Default Value
Network Bandwidth Weight	Weight for network scheduling. The value is an integer ranging from 1 to 2147483647. The default value is -1. **CAUTION** Only cluster 8.2.1 and later versions support the network bandwidth weight.	Yes	-1 (not limited)

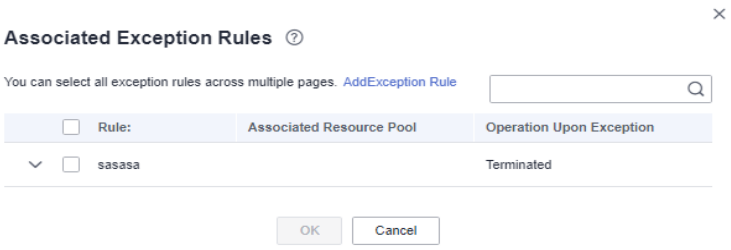
NOTE

Only 8.1.3 and later versions support the CPU limit management.

2. Click **OK**.

Step 7 Associate exception rules.

1. Associate exception rules.



2. Unbind exception rules.

NOTE

- Only clusters of 8.2.0 and later versions support associating exception rules. GaussDB(DWS) 3.0 does not support this function. For cluster versions earlier than 8.2.0, see **Step 7.1**.
- 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.

3. Modify the exception rules.

Modify rule parameters. See the following table for more information.

Table 15-4 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**

Parameter	Description	Value Range (0 Means No Limit)	Operation
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**
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**

Parameter	Description	Value Range (0 Means No Limit)	Operation
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**

NOTE

Exception rules allow you to control exceptions of jobs executed by users in a resource pool. Currently, you can configure the parameters listed in Table 15-4.

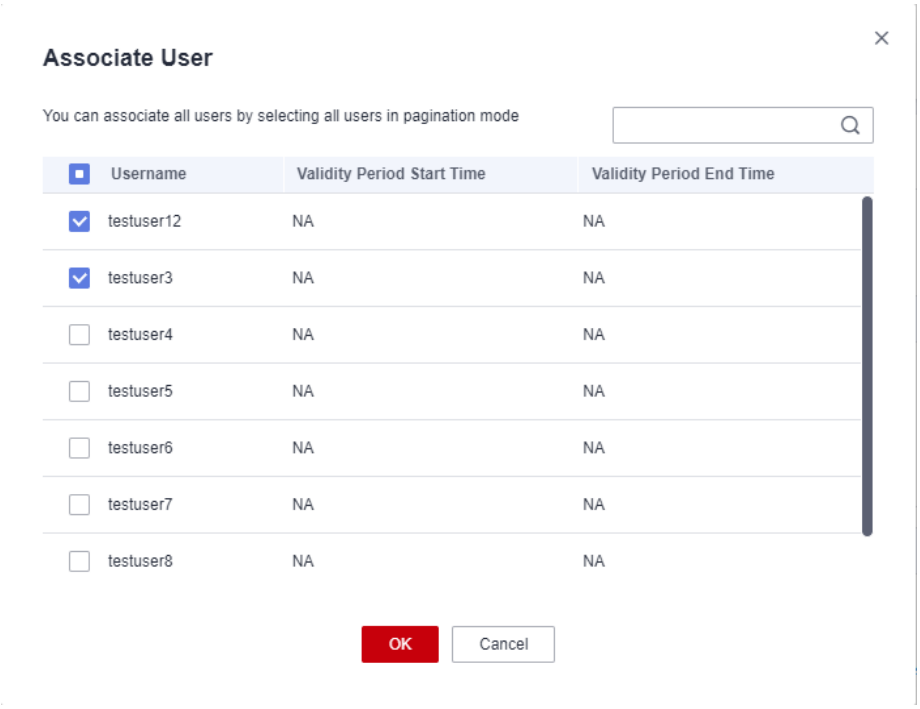
- If you select **Terminate** or **Downgrade**, you need to set the corresponding time or percentage.
- If you select **Not limited**, the corresponding execution rule does not take effect.
- Only cluster 8.2.0 and later versions support the function of modifying exception rules.

Step 8 Associate users.

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.

1. Click **Add**.
2. Select the users to be added from the current user list. You can select multiple users at a time.



3. Click **OK**.
4. To remove a user, click **Disassociate User** in the **Operation** column of the user.

----End

15.2.5 Deleting a Resource Pool

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

Step 2 Choose **Clusters**. Click the name of a cluster.

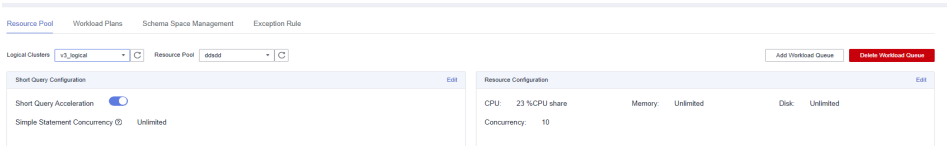
Step 3 Choose **Resource Management Configurations**.

Step 4 In the **Resource Pools** area on the left, click the name of a resource pool.

Step 5 Click **Delete Resource Pool**.

NOTE

After a resource pool is deleted, the users (if any) associated with this pool will be associated with the default resource pool instead.



----End

15.3 Resource Management Plan

15.3.1 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.

Creating a Resource Management Plan

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

Step 2 Choose **Clusters**. Click the name of a cluster.

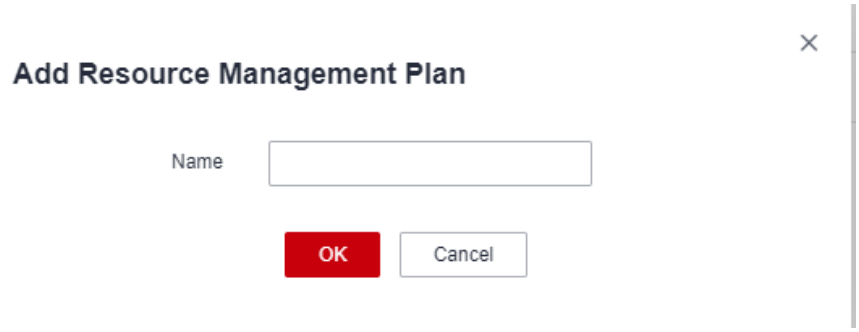
Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 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) management console.

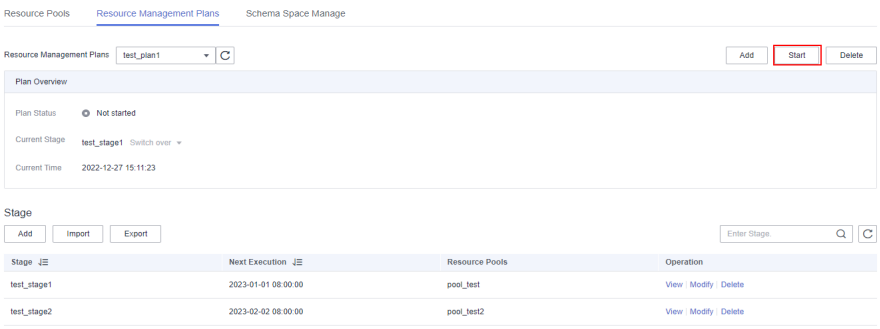
Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Enter the plan details page and click **Start** to start a resource management 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

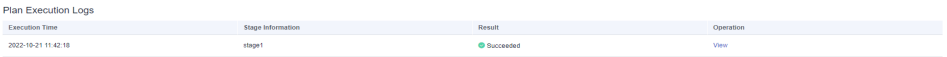
Viewing the Execution Logs of a Resource Management Plan

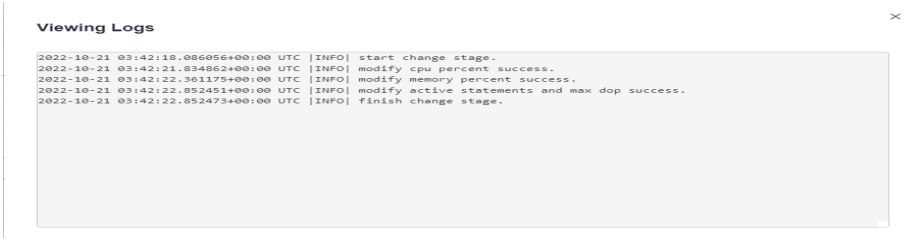
Step 1 Log in to the GaussDB(DWS) management console.

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 Go to the plan details page and view the switchover logs in the **Plan Execution Log** area.





----End

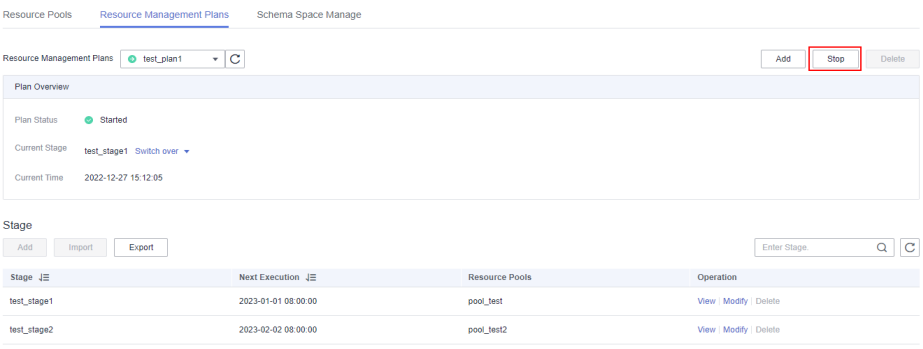
Stopping a Resource Management Plan

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 Enter the plan details page and click **Stop** to stop a resource management plan.



----End

Deleting a Resource Management Plan

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

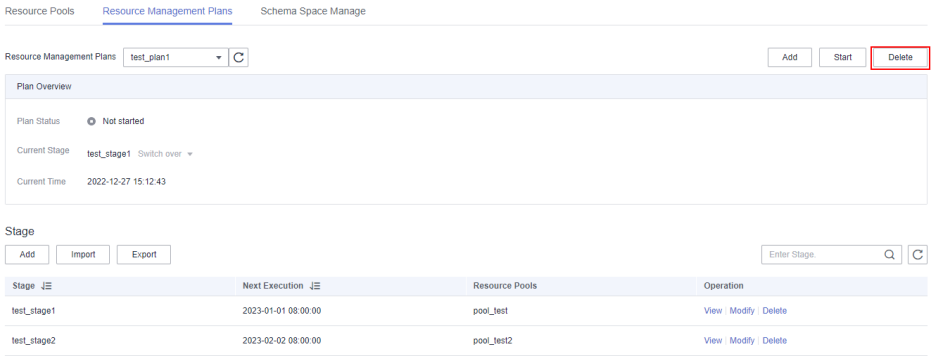
Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Go to the **Basic Information** page and click the **Resource Management** tab in the navigation pane on the left.

Step 4 Enter the plan details page and click **Delete** to delete a resource management plan.

NOTICE

You cannot delete a running resource management plan.



----End

15.3.2 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 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) management console.

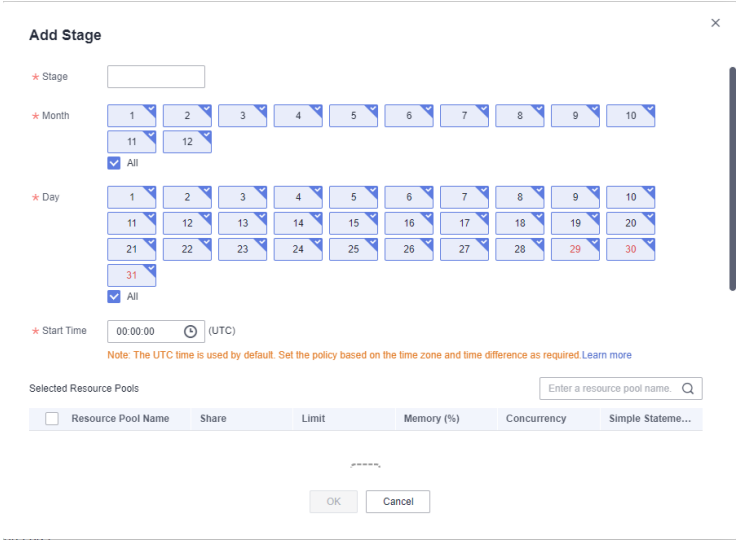
Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

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.



----End

Modifying a Resource Management Plan Stage

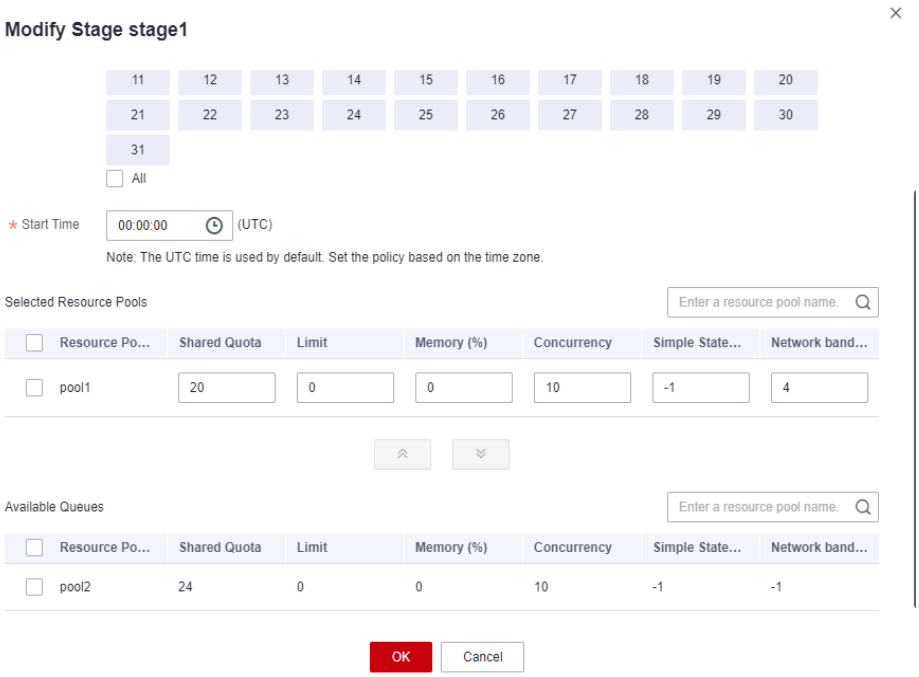
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 Go to the plan details page and click **Modify** in the **Operation** column of the target plan stage.

Step 5 Modify parameters, such as the stage changing time and resource configurations.



NOTE

Only clusters of the version 8.2.1 and later support the network bandwidth weight.

----End

Manually Changing the Resource Management Plan Stage

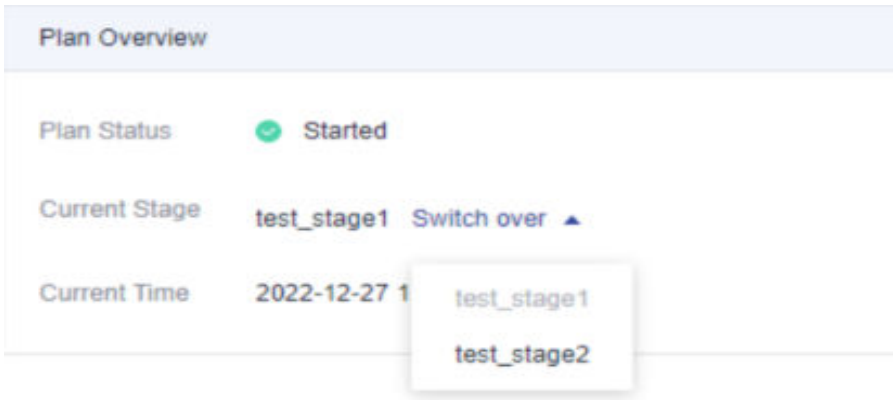
If a running plan needs to be switched to a stage in advance, you can manually do it.

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

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 Go to the plan details page, click the **Switch over** button in the plan overview area, and select a stage.



----End

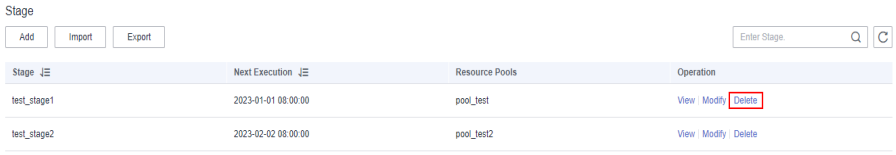
Deleting a Resource Management Plan Stage

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

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 Go to the plan details page and click **Delete** in the **Operation** column of the target plan stage.



----End

NOTE

Stages in a running resource management plan cannot be deleted.

15.3.3 Importing or Exporting a Resource Management Plan

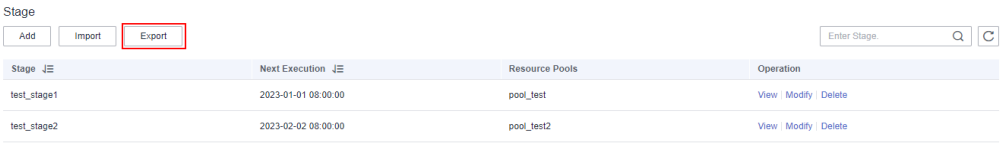
Exporting a Resource Management Plan

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

Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 Enter the plan details page and click **Export** to export a resource management plan.



----End

Importing a Resource Management Plan

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

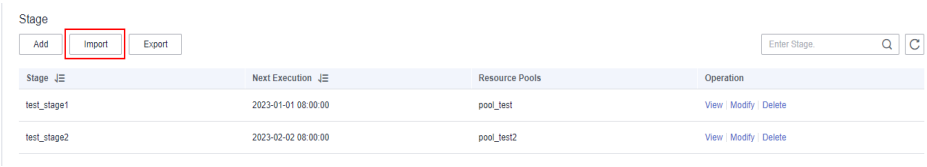
Step 2 Choose **Clusters**. Click the name of a cluster.

Step 3 Choose **Resource Management Configurations**.

Step 4 Enter the plan details page, click **Import**, and select and import a configuration file to the resource management plan.

NOTICE

- Configurations cannot be imported to a running resource management plan.
- Ensure there is a resource pool before import.



----End

15.4 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

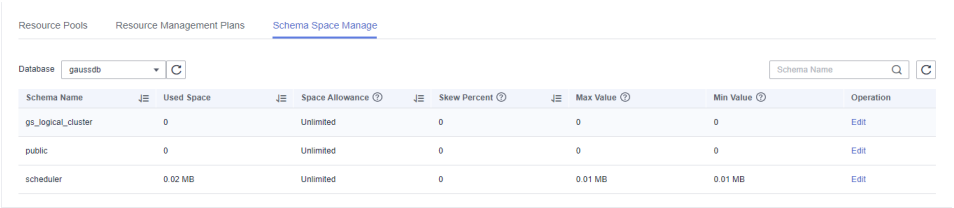
Step 1 Log in to the GaussDB(DWS) management console.

Step 2 Choose **Clusters**. Click the name of a cluster.

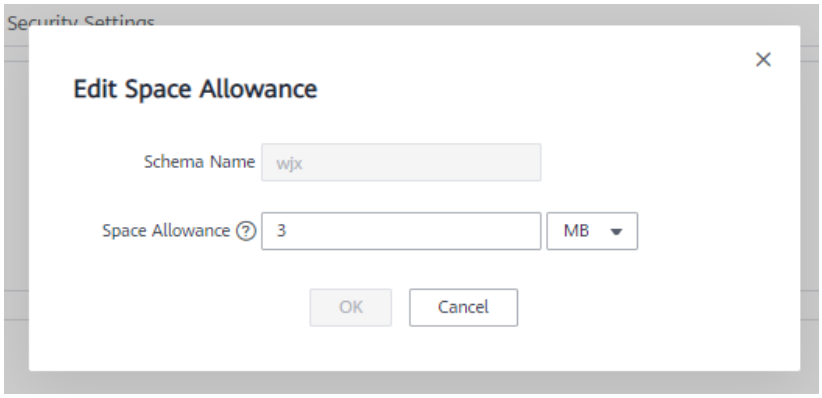
Step 3 Choose **Resource Management Configurations**.

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.



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 of a single DN = Total quota/Number of DNs. Therefore, the configured value may slightly fluctuate with the displayed value.

----End

16 Data Source Management

16.1 MRS Data Sources

16.1.1 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.

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, the hybrid data warehouse (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 in a GaussDB(DWS) 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**.

16.1.2 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 management console.

Step 2 Choose **Service List** > **Analytics** > **MapReduce Service** to enter the MRS management console and create a 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.
- MRS cluster versions 1.9.2, 2.1.0, 3.0.2-LTS, and 3.1.2-LTS are recommended.

NOTE

- 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.*, 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.

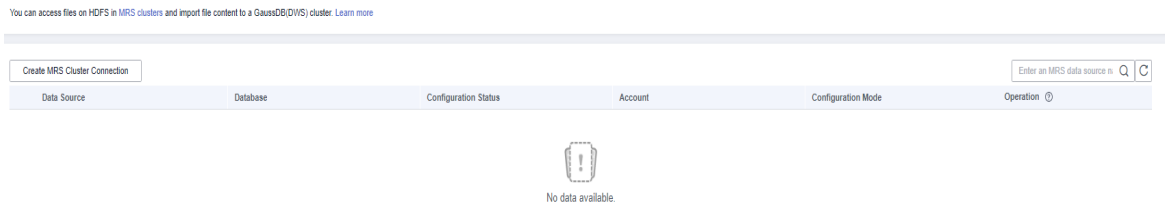
Step 3 Choose **Service List** > **Analytics** > **GaussDB(DWS)**.

Step 4 On the GaussDB(DWS) management 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**.

Figure 16-1 MRS data sources



Step 7 Click **Create MRS Cluster Connection** and configure parameters.

Figure 16-2 Selecting an MRS user and creating an MRS data source

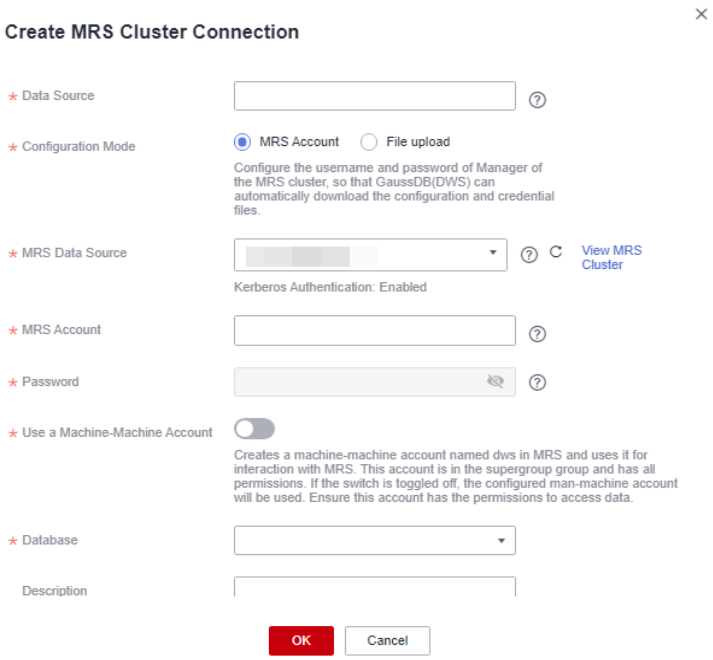


Figure 16-3 Selecting files to upload and creating an MRS data source connection

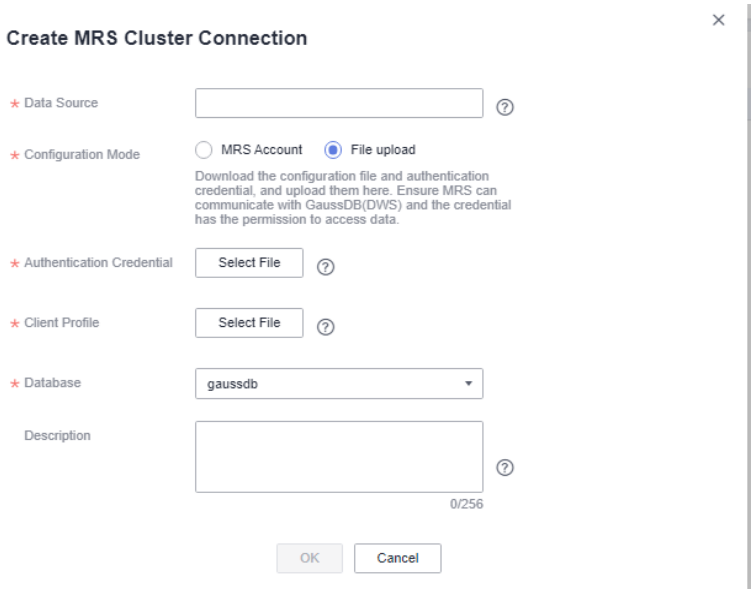


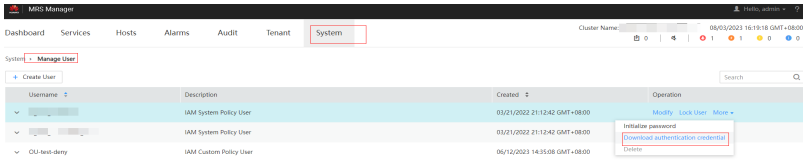
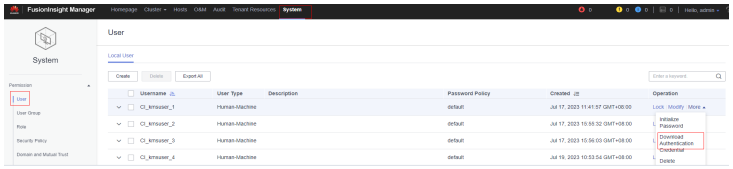
Table 16-1 MRS common connection parameters

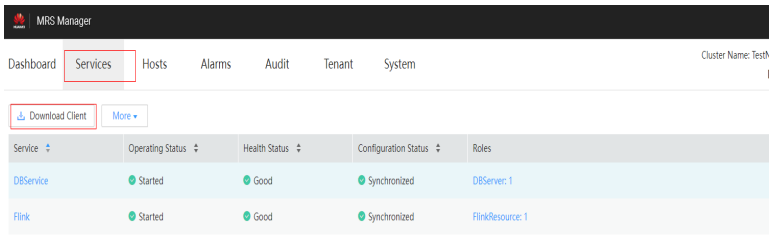
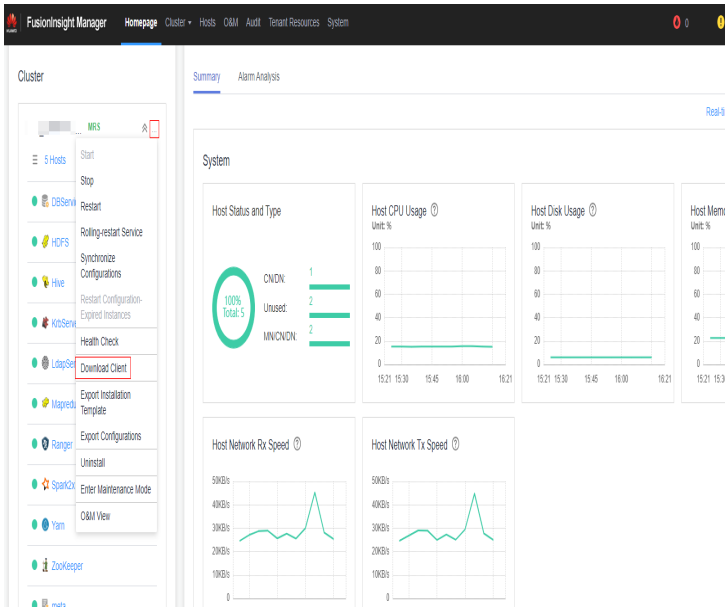
Parameter	Description
Data Source	GaussDB(DWS) database server name. It can contain 3 to 63 characters, including lowercase letters, numbers, and underscores (_), and must start with a lowercase letter.
Configuration Mode	The way in which the system obtains files. The options are as follows: • **MRS 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 16-2**. • **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 16-3**. **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 16-2 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 16-3 Parameters of the File upload mode

Parameter	Description
Authentication Credential	Keytab file of a user A credential file downloaded from Manager of the MRS cluster. File name format: **Username_Timestamp_keytab.tar** **For MRS 2.x or earlier**, choose **System** > **Manage User**. In the **Operation** column of a user, choose **More** > **Download authentication credential**.  **For MRS 3.x or later**, choose **System** > **Permission** > **User**. In the **Operation** column of a user, choose **More** > **Download Authentication Credential**. 

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**.  **For MRS 3.x or later**, choose **Homepage**. Click the **More** icon and choose **Download Client**. 

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

16.1.3 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.

Table 16-4 Parameter description

Parameter	Description
dfs.client.read.shortcircuit	Specifies whether to enable the local read function.
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.

Parameter	Description
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.
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.

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) management 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.

----End

16.2 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.

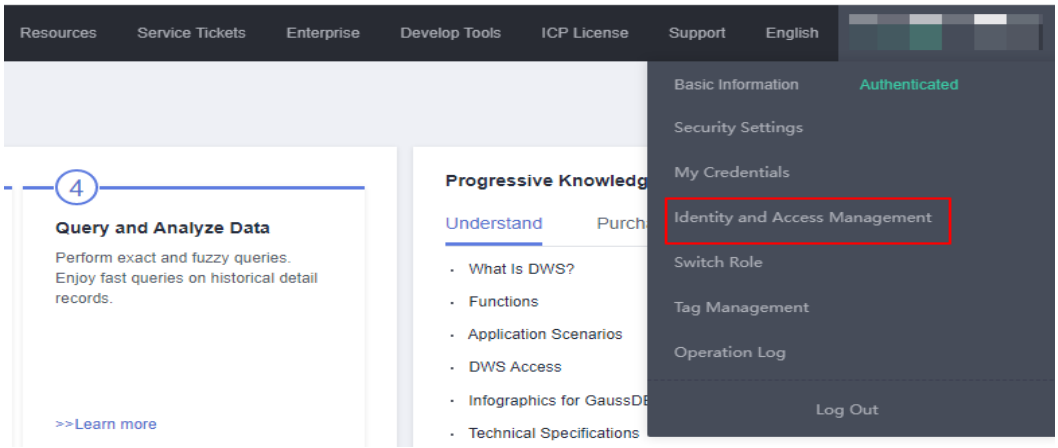
Creating an OBS Agency

Scenario

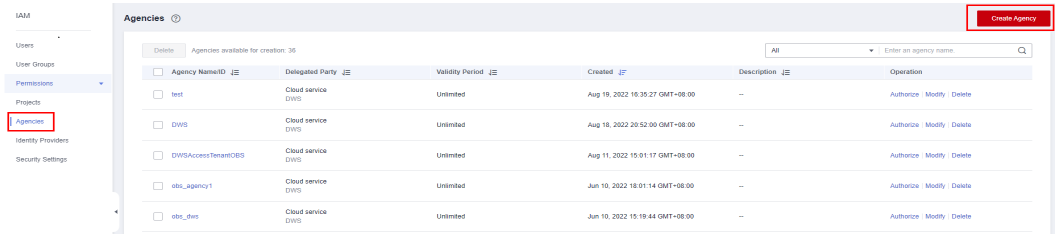
Before creating an OBS data source, create an agency that grants GaussDB(DWS) the OBS OperateAccess or OBS Administrator permission.

Procedure

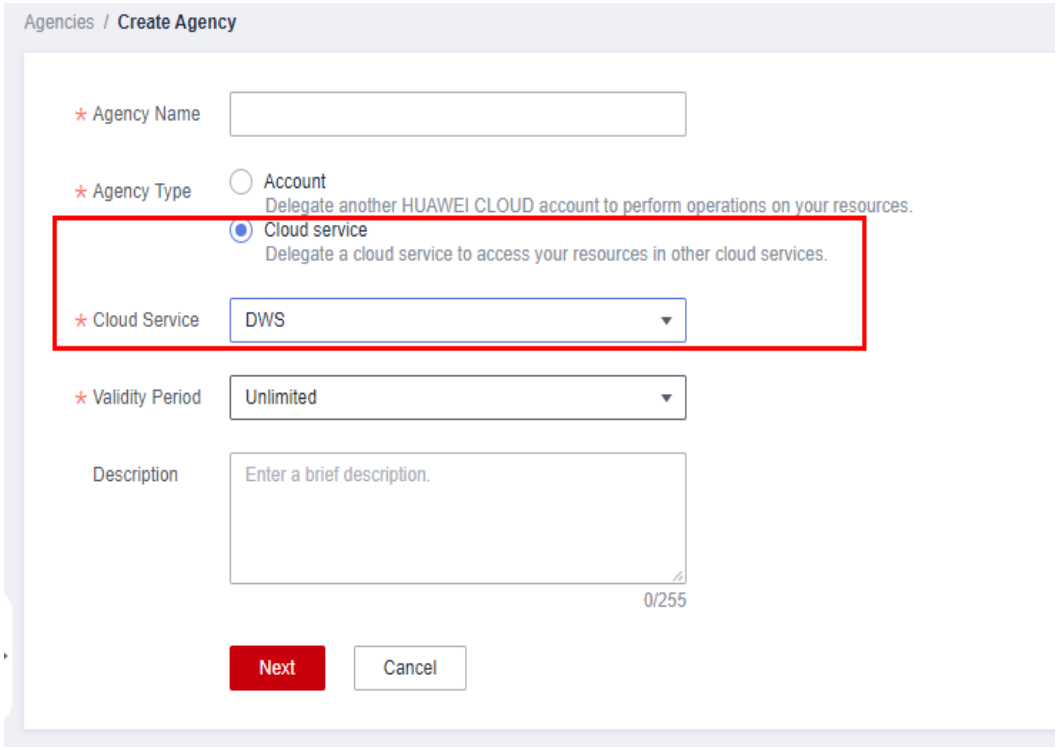
Step 1 Click your account in the upper right corner of the page and choose **Identity and Access Management**.



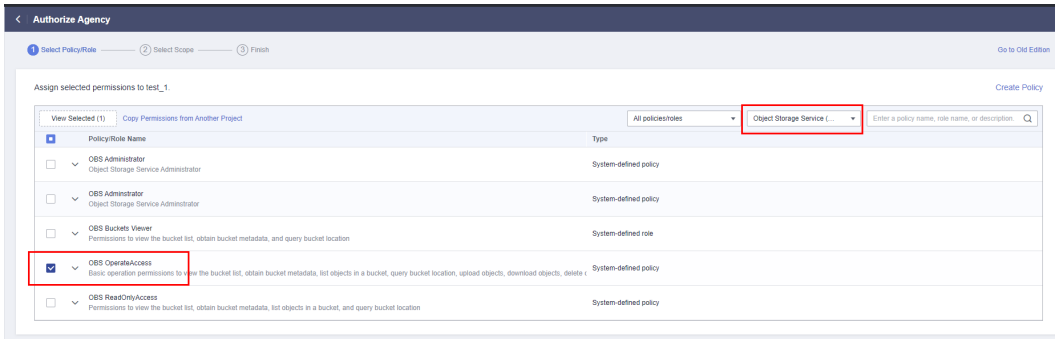
Step 2 In the navigation pane on the left, choose **Agency**. In the upper right corner, click **Create Agency**.



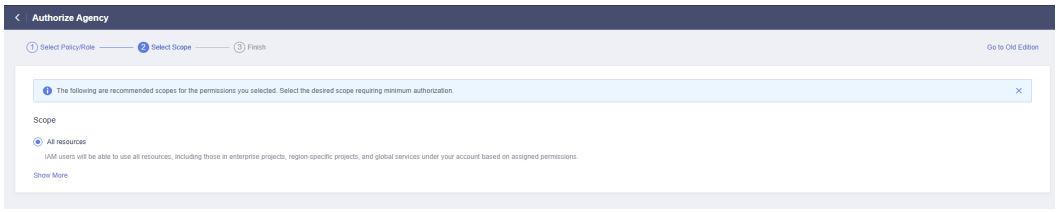
Step 3 Select **Cloud Service** and set **Cloud Service** to **DWS**.



Step 4 Click **Next** to grant the OBS OperateAccess or OBS Administrator permission to the agency.



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) management 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.

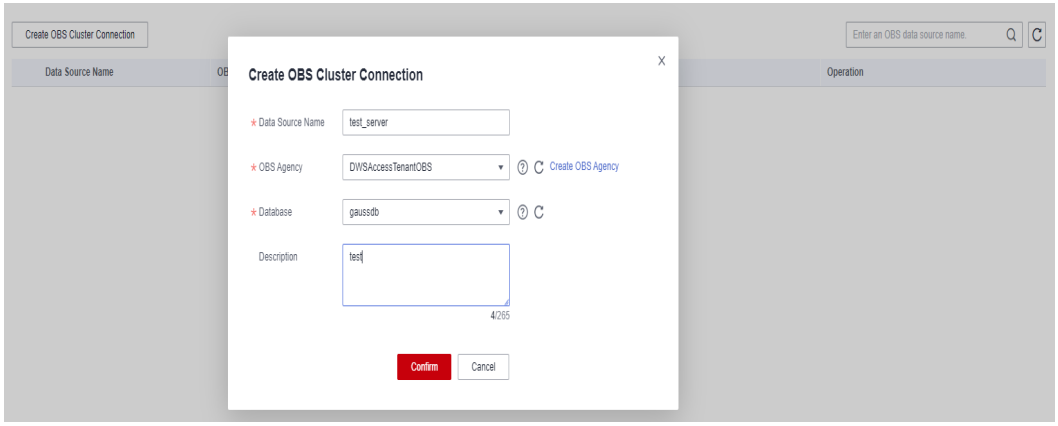


Table 16-5 OBS data source connection parameters

Parameter	Description
Data Source Name	Name of the OBS data source connection to be created
OBS Agency	Agency with the OBS OperateAccess permission to be granted to 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

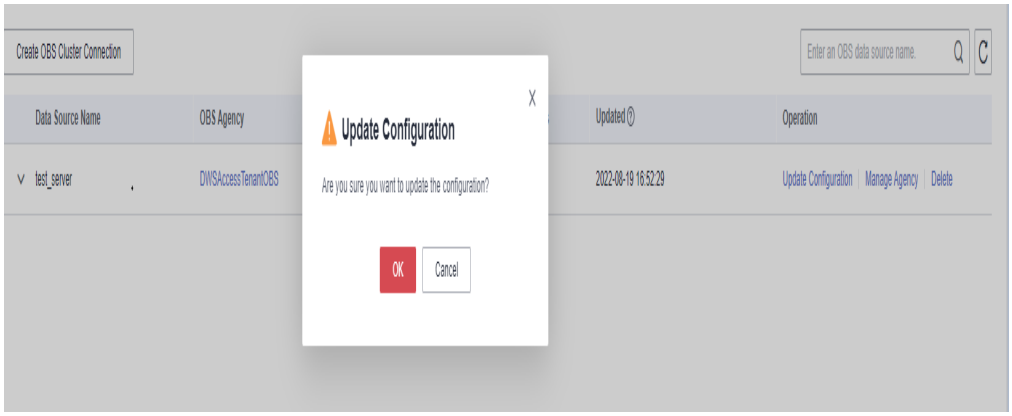
After an OBS data source connection is created, GaussDB(DWS) periodically updates the temporary agency information used by the data source. If the automatic update fails for 24 hours, the data source connection will be unavailable. To solve this problem, manually update the information on the console.

Procedure

Step 1 On the GaussDB(DWS) management 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

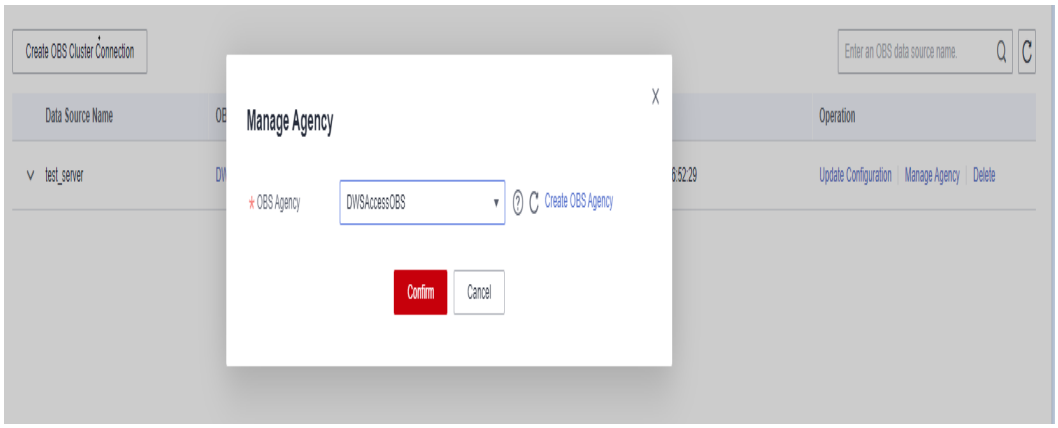
You can change the agency bound to the OBS data source.

Procedure

Step 1 On the GaussDB(DWS) management 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.

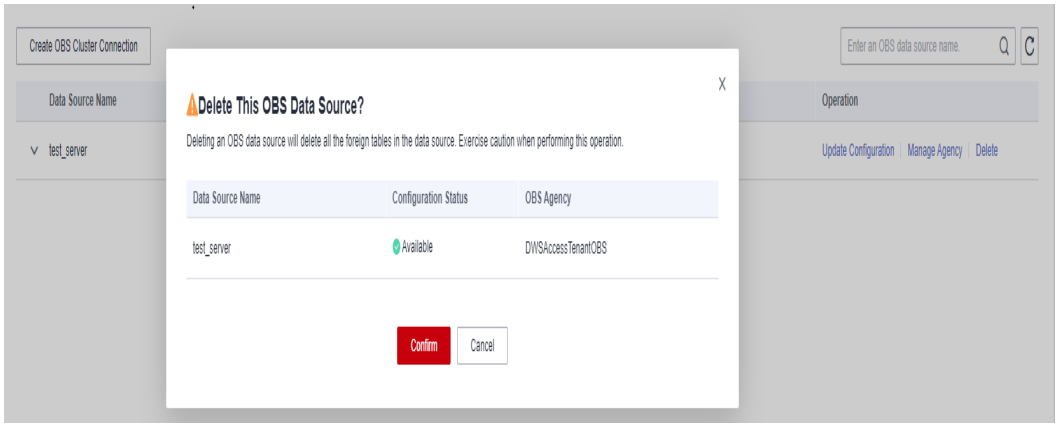
----End

Deleting an OBS Data Source

Step 1 On the GaussDB(DWS) management 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, for example, **obs_server**, on the console. The way users create and use foreign tables with an agency is the same as the way they do without an agency. For details about how to use the OBS data source, see **Importing Data from OBS**.

The following example reads data from OBS through a foreign table.

1. Create an OBS foreign table **customer_address** that does not contain partition columns. Files on **obs_server** are in ORC format and stored in **/user/obs/region_orc11_64stripe1/**.

```
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;
```

2. Query data stored in OBS by using a foreign table.

```
SELECT COUNT(*) FROM customer_address;
count
-------
20
(1row)
```


17 Managing Logical Clusters

17.1 Logical Cluster Overview

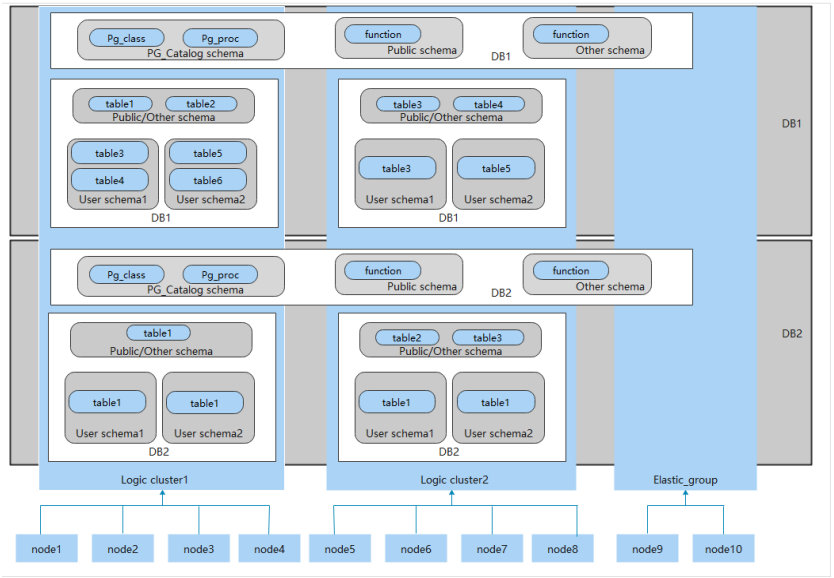
Concepts

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 17-1** 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 17-1 Relationships between logical clusters, databases, and tables



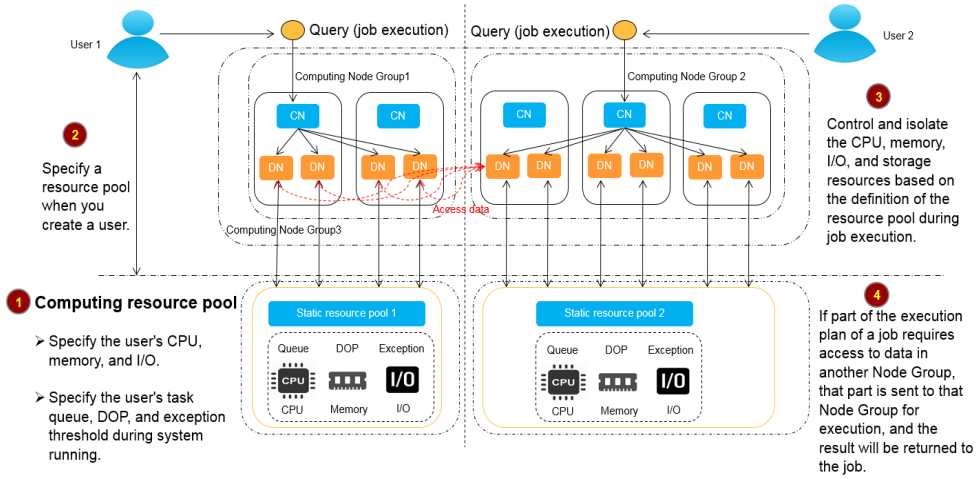
NOTE

- 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.
- The logical cluster of a GaussDB(DWS) 3.0 cluster cannot be switched if the original physical cluster contains data. Ensure the original physical cluster is empty during the switchover.

Logical Cluster Architecture

Figure 17-2 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 17-2 Logical Cluster Architecture



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:

- **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.
- **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 8.1.3.101 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.
- **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.

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.
  - GaussDB(DWS) 3.0 supports the creation of read-only logical clusters. 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 **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.

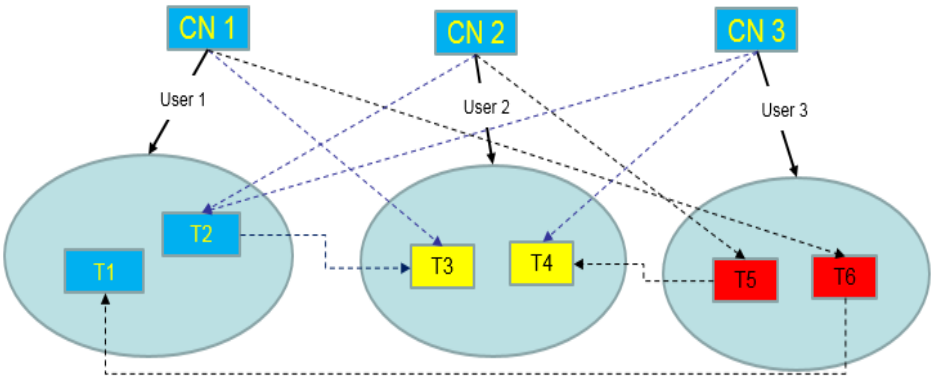
NOTE

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 17-3 Logical cluster division based on resource requirements

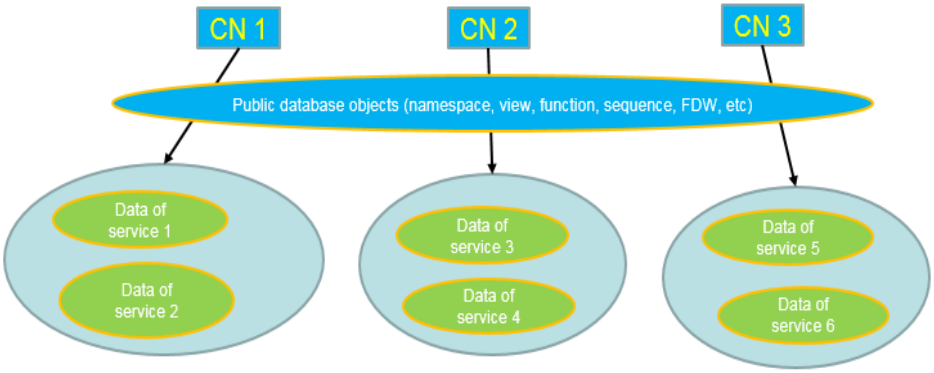


As shown in the preceding figure, data with different resource requirements is stored in different logical clusters, and different logical clusters also support mutual access. This ensures that functions are not affected while resources are isolated.

- 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 17-4 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.

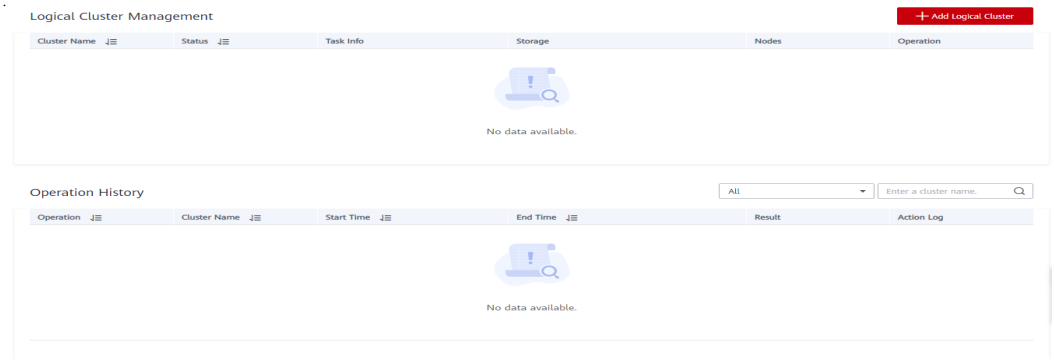
17.2 Adding Logical Clusters

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

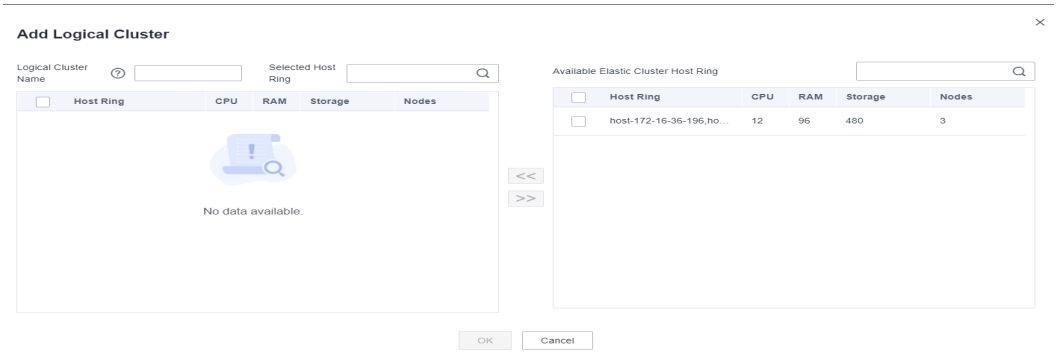
Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

Step 3 Enable **Logical Clusters**. The **Logical Clusters** menu item will be displayed in the navigation pane on the left.

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**.



----End

CAUTION

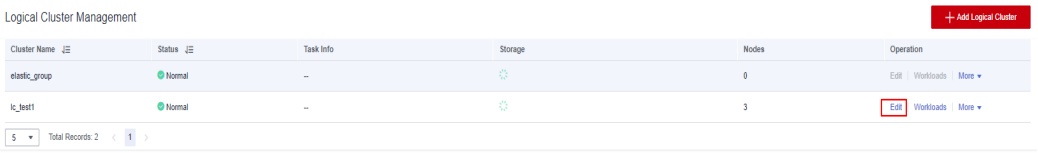
- If you access the **Logical Clusters** page for the first time, the metadata of the logical cluster created at the backend is synchronized to the frontend. After the synchronization is complete, you can view information about the logical clusters at the frontend. The logical cluster name is case sensitive. 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.

17.3 Editing Logical Clusters

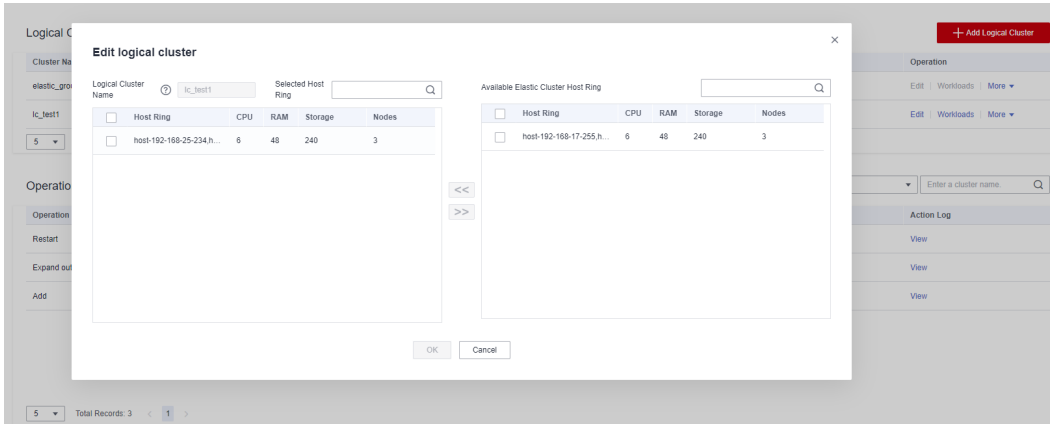
Step 1 Log in to the GaussDB(DWS) console. In the navigation pane, choose **Clusters** > **Dedicated Clusters**.

Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

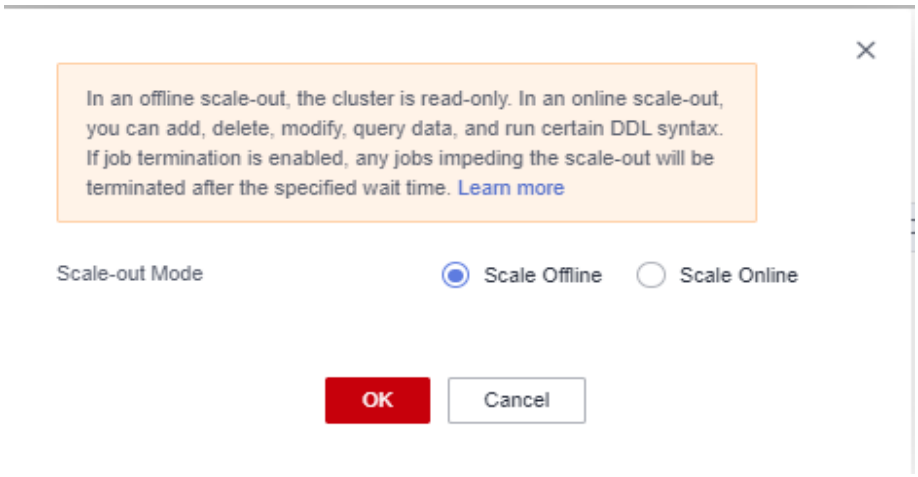
Step 3 In the navigation pane, choose **Logical Clusters** and click **Edit** in the **Operation** column of the target cluster.



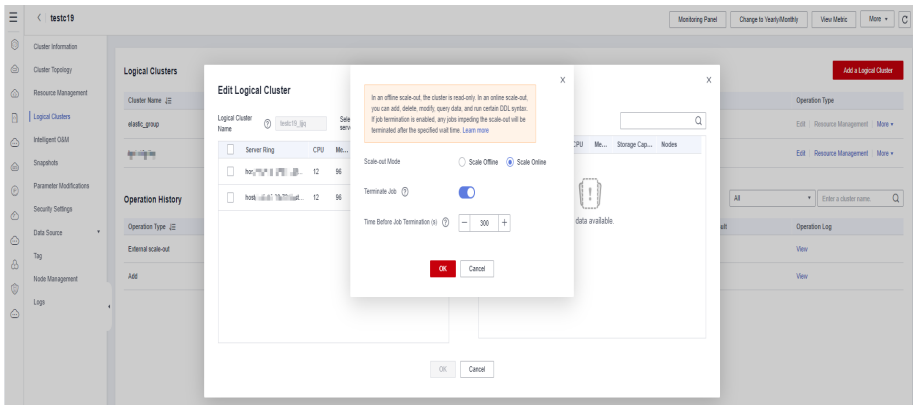
Step 4 Add a node to the logical cluster by moving the selected ring from the right to the left, or remove a node from the logical cluster by moving the selected ring from the left to the right, and click **OK**.



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. 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.



----End

NOTE

- 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.

17.4 Managing Resources (in a Logical Cluster)

Precautions

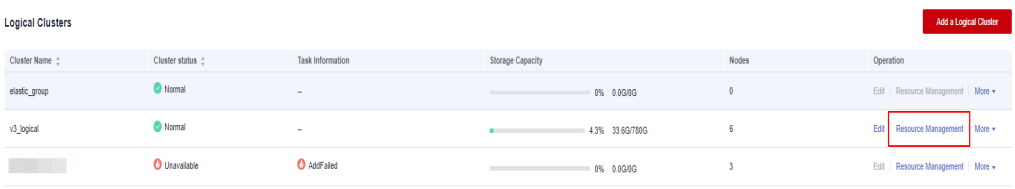
The original resource pool configuration is cleared when the cluster is converted from physical to logical. You have to add the resource pool again if you want to configure it after the conversion.

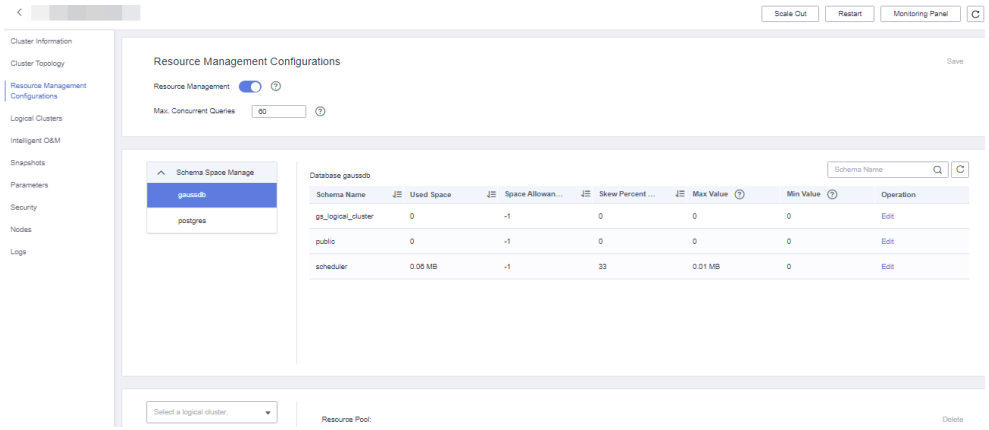
Procedure

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 **Resource Management**.





----End

17.5 Scheduling GaussDB(DWS) 3.0 Logical Cluster Creation and Deletion

Context

You can schedule the creation and deletion of logical clusters. Computing logical clusters can be created and deleted during the scheduled period of time to dynamically scale computing resources.

NOTE

- This feature supports only GaussDB(DWS) 3.0 clusters. For an earlier version, contact technical support to upgrade it first.
- In a yearly/monthly GaussDB(DWS) 3.0 cluster, nodes are automatically added when logical clusters are added as scheduled. Nodes are billed on a pay-per-use basis.
- By default, a logical cluster created using this feature is used to provide computing power. After a user is associated with the logical cluster, the user's queries are processed by this logical cluster, but table creation statements are still processed in the original logical cluster.
- 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.

Procedure

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

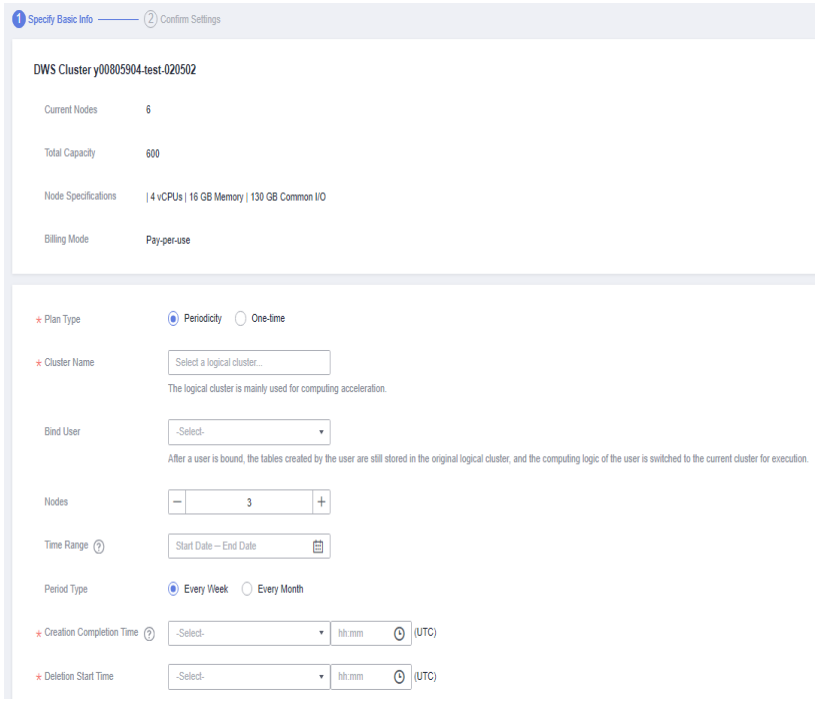
Step 2 In the cluster list, click the name of the target cluster. The **Cluster Information** page is displayed.

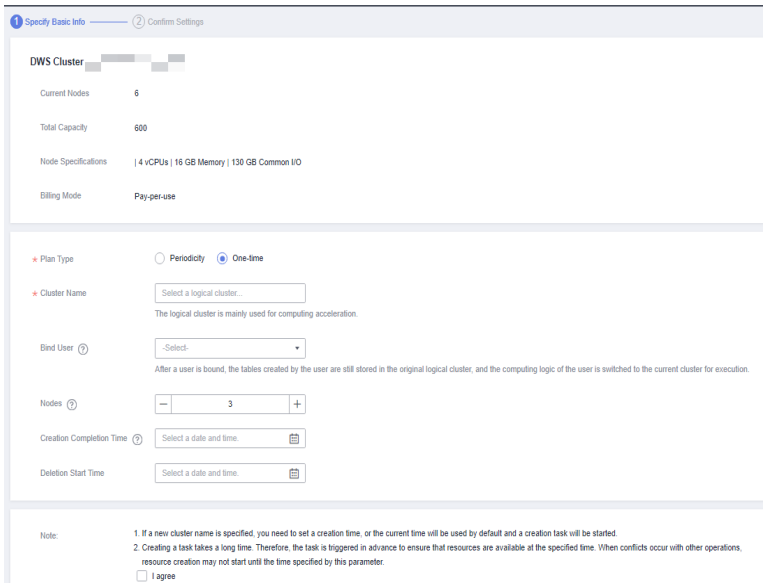
Step 3 After GaussDB(DWS) 3.0 clusters are created, they are in logical cluster mode.

- GaussDB(DWS) 3.0 cluster: In the navigation pane on the left, switch to the **Logical Clusters** page, click **Add Plan** to configure a proper scheduling plan.

Step 4 Select a plan type. It can be:

- **Periodically**: 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.
- **One-time**: The plan is executed only once in the specified period.

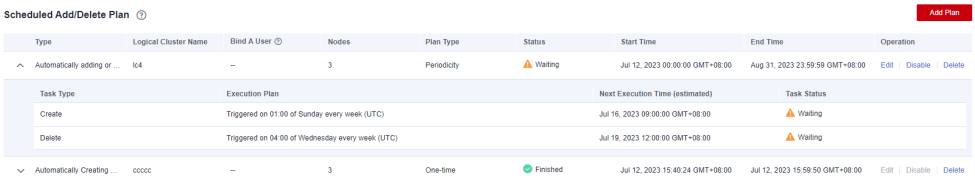




Step 5 Click **OK**. In the scheduled plan list, you can view the plan details and next execution time.

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.



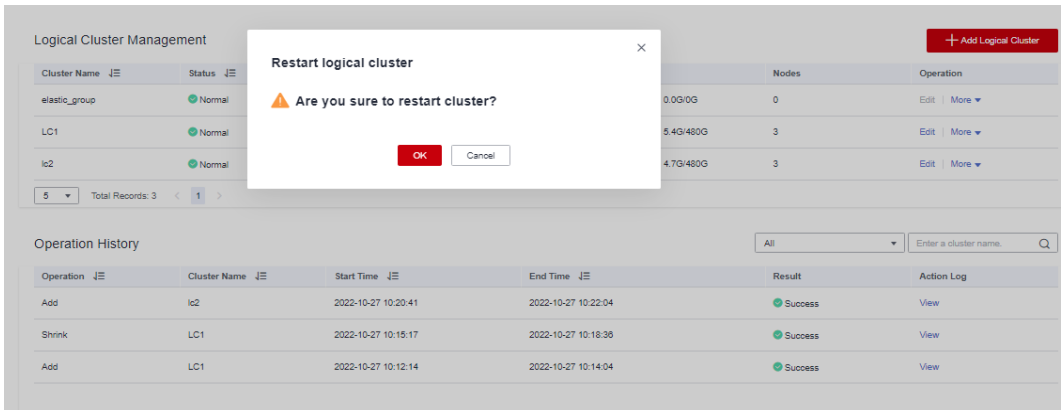
----End

17.6 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, choose **Logical Clusters**. Click **Restart** in the **Operation** column of the target cluster, and click **OK** in the displayed dialog box.



----End

17.7 Scaling Out Logical Clusters

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..

NOTE

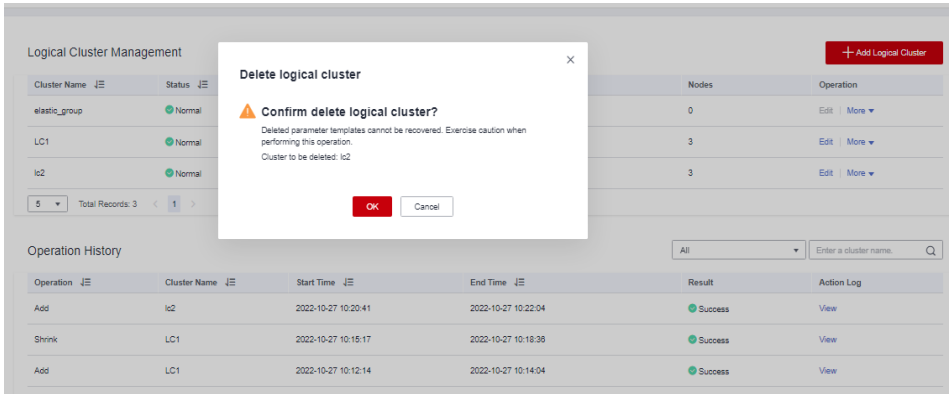
----End

17.8 Deleting 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, choose **Logical Clusters**. Click **Delete** in the **Operation** column of the target cluster, and click **OK** in the displayed dialog box.



----End

NOTICE

- The first added logical cluster cannot be deleted.
- Nodes of the deleted logical cluster are added to the elastic cluster.

17.9 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.

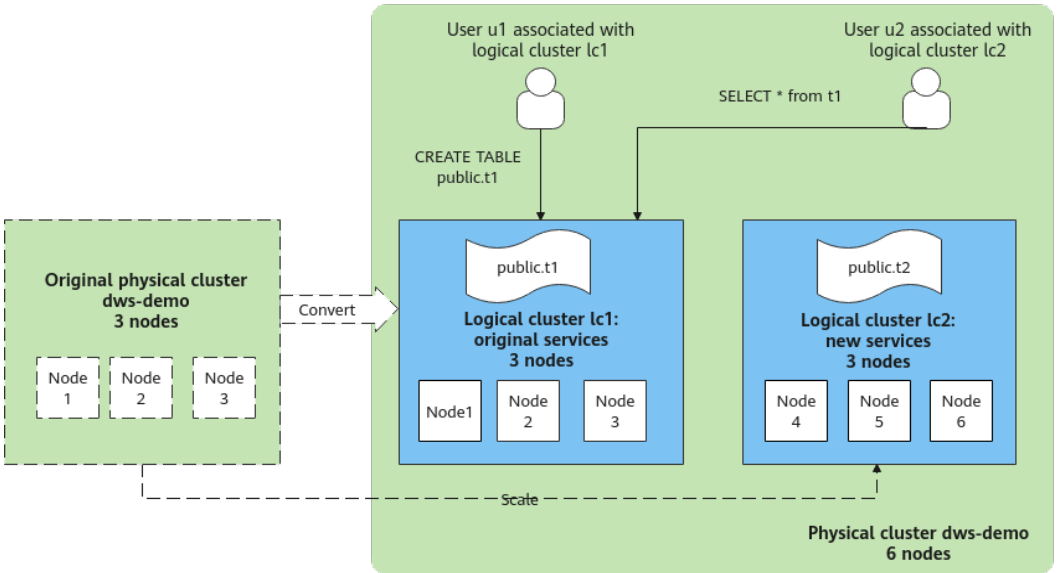
As the service scale grows, the number of services in the cluster system also increases. Creating multiple resource pools becomes less effective in controlling resource competition. GaussDB(DWS) uses the distributed architecture. and its data is distributed on multiple nodes. Each table is distributed across all DNs in the cluster, an 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 17-5 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 17-5 Accessing data across logical clusters



Creating a Cluster and Preparing Table Data

Step 1 Create a cluster. For details, see **Creating a GaussDB(DWS) 2.0 Cluster**.

Step 2 After connecting to the database, create table **t1** as the system administrator **dbadmin** and insert two data records into the table.


```
CREATE TABLE t1 (id int, name varchar(20));
INSERT INTO t1 VALUES (1,'joy'),(2,'lily');
```

----End

Converting to Logical Cluster lc1

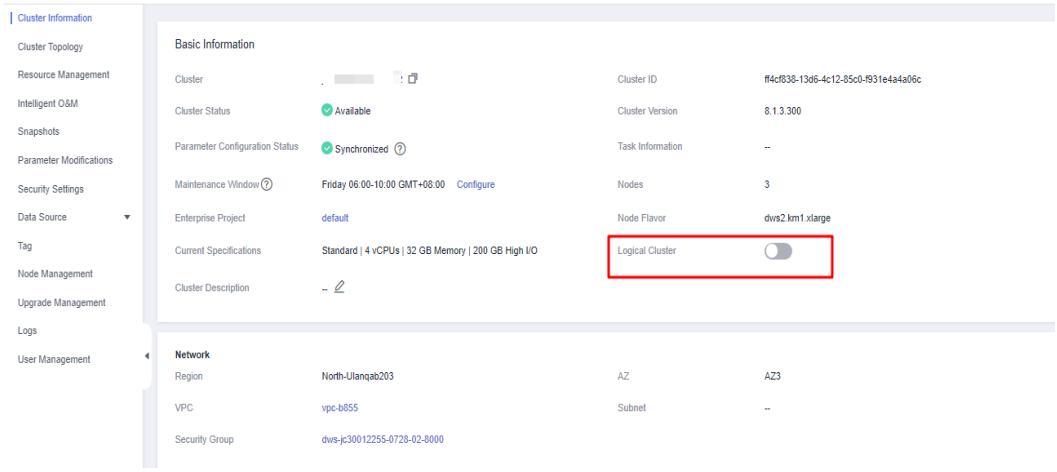
NOTICE

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.

Figure 17-6 Enabling the logical cluster function



Step 3 In the navigation pane, choose **Logical Clusters**. 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 17-7 Adding a logical cluster

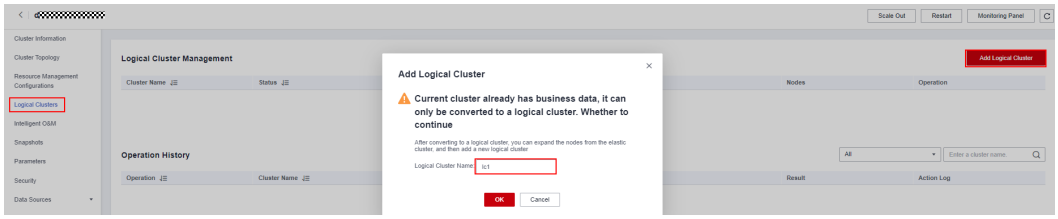
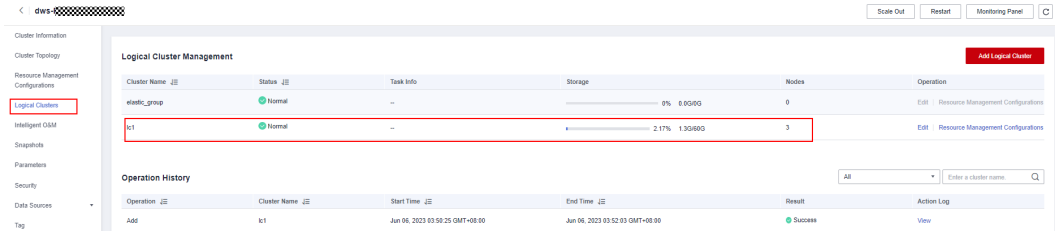


Figure 17-8 Logical cluster conversion succeeded

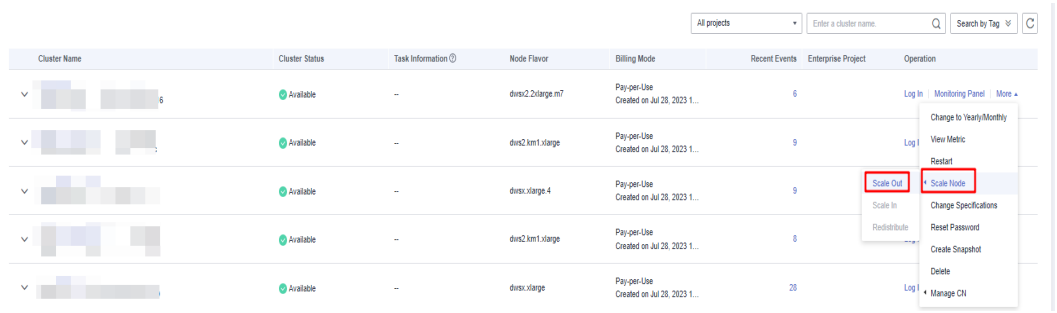


----End

Adding Nodes to the elastic_group Cluster

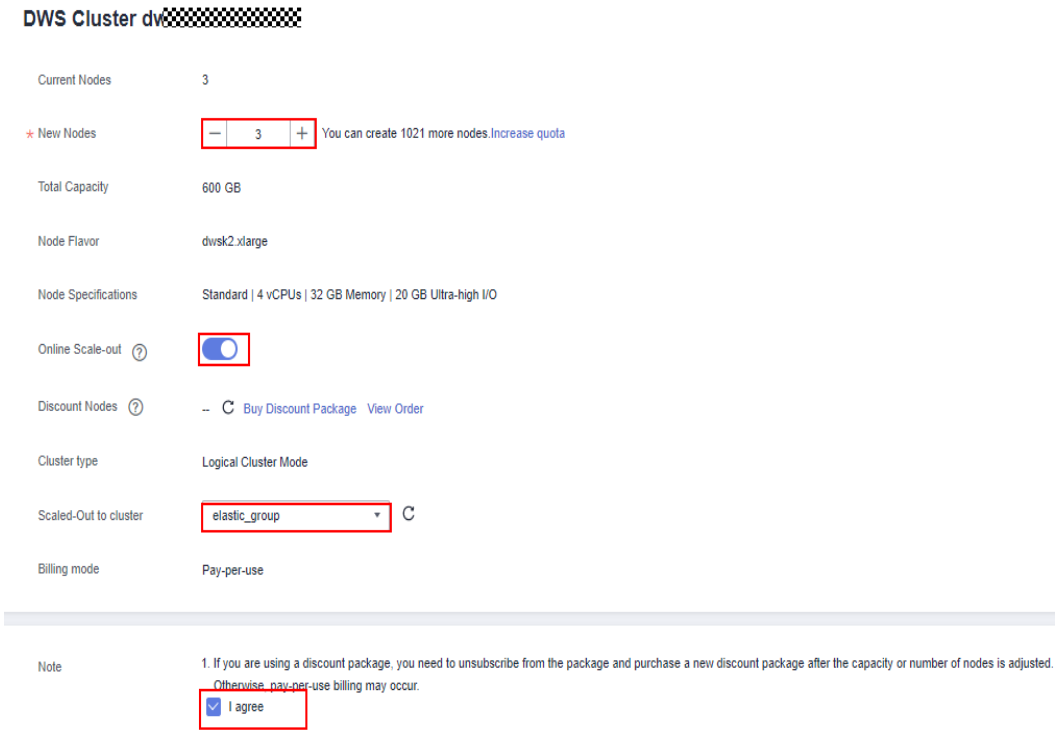
Step 1 Return to the **Cluster Management** page. In the **Operation** column of the cluster, choose **More** > **Scale Node** > **Scale Out**.

Figure 17-9 Scaling out a cluster



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**.

Figure 17-10 Scale-out process



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 17-11 Adding a logical cluster

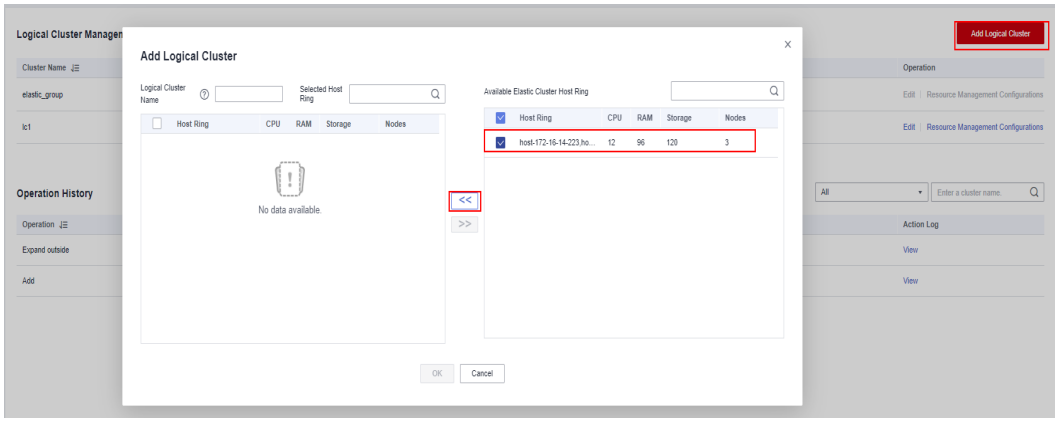


Figure 17-12 Selecting a host ring

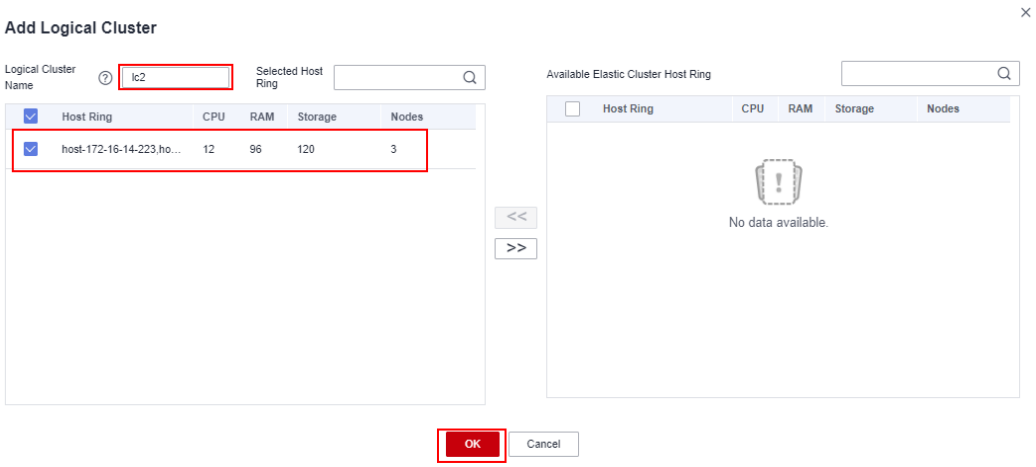
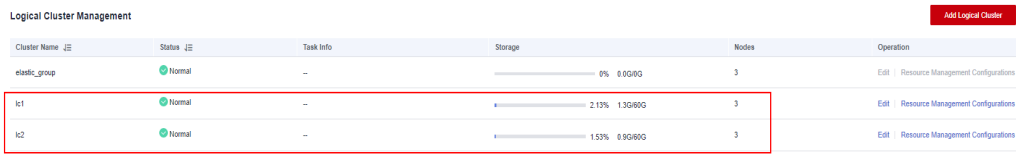


Figure 17-13 Logical cluster added



----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 **t1**:

Verify that service data can be queried after the conversion.

```
SELECT * FROM t1;
```

Step 2 Run the following statements to associate **u1** with logical cluster **lc1** and **u2** with logical cluster **lc2**, and grant all permissions of the original service table **t1** to user **u1**:

```
CREATE USER u1 NODE GROUP 'lc1' password '{password}';
CREATE USER u2 NODE GROUP 'lc2' password '{password}';
GRANT ALL ON TABLE t1 TO u1;
```

Step 3 Switch to 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**. This indicates data is isolated between logical clusters.

```
SET ROLE u2 PASSWORD '{password}';
SELECT * FROM t1;
```

```
ERROR: permission denied for node group lc1
```

Step 4 Switch back to system administrator **dbadmin** and grant the access permission of logical cluster **lc1** to user **u2**.


```
SET ROLE dbadmin PASSWORD '{password}';
GRANT USAGE ON NODE GROUP lc1 TO u2;
```

Step 5 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.

```
SET ROLE u2 PASSWORD '{password}';
SELECT * FROM t1;
```

	id	name
1	1	joy
2	2	lily

----End

17.10 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) 2.0 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;
```


	group_name
1	group_version1
2	elastic_group
3	lc1
4	lc2

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;
```

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

17.11 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 **Scheduling GaussDB(DWS) 3.0 Logical Cluster Creation and Deletion** 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) 3.0 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

