Data Warehouse Service

API Reference

Issue 01

Date 2022-06-16

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are trademarks 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 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 Before You Start

1.1 Overview

Welcome to GaussDB(DWS) API reference. GaussDB(DWS) is a fully-managed and enterprise-class cloud data warehouse service. It is O&M-free and supports online scale-out and multi-source data loading. It is compatible with the PostgreSQL ecosystem and helps enterprises efficiently analyze and monetize datasets online.

This document describes how to use application programming interfaces (APIs) to perform operations, on data warehouse clusters and snapshots, such as creating, querying, and deleting. For details about all supported operations, see **API Overview**.

If you plan to access GaussDB(DWS) using an API, ensure that you are familiar with GaussDB(DWS) concepts. For details, see **Service Overview** in the *Data Warehouse Service User Guide*.

1.2 API Calling

DWS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see **Calling APIs**.

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the endpoints of all services, see **Regions and Endpoints**.

1.4 Basic Concepts

- Account

  This account has full access to all cloud services and resources associated with it. It can be used to reset user passwords and grant user permissions. The domain should not be used directly to perform routine management. For

security purposes, create users and grant them permissions for routine management.

- Users

  An IAM user is created using an account for cloud services. Each IAM user has its own identity credentials (password and access keys).

  The account name, username, and password are required for API authentication.

- Region

  A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.

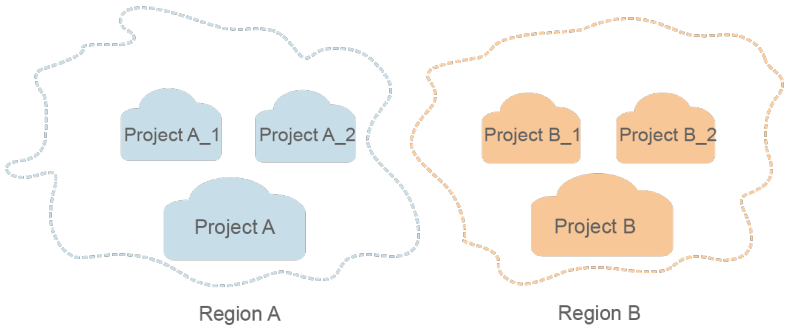
- AZ

  An AZ contains one or more physical data centers. Each AZ has independent power and network devices. Within an AZ, computing, network, storage, and other resources are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to support cross-AZ high-availability systems.

- Item

  Projects group and isolate resources (including compute, storage, and network resources) across physical regions. A default project is provided for each service region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and apply for resources in the subprojects. IAM users can then be assigned permissions to access only specific resources in the subprojects.

  Figure 1-1 Project isolating model



- Enterprise project

  Enterprise projects group and manage resources across regions. Resources in enterprise projects are logically isolated from each other. An enterprise project can contain resources of multiple regions, and resources can be added to or removed from enterprise projects.

2 API Overview

You can use the following GaussDB(DWS) functions.

Type	API	Description
Cluster management APIs	Creating a Cluster	Creates a cluster.
	Querying the Cluster List	Queries and displays the cluster list.
	Querying Cluster Details	Queries cluster details.
	Querying the Supported Node Types	Queries all node types supported by GaussDB(DWS).
	Deleting a Cluster	Deletes a cluster.
	Restarting a Cluster	Restarts a cluster.
	Scaling Out Clusters	Scales out a cluster.
	Resetting a Password	Resets the cluster's administrator password.
Snapshot management APIs	Creating a Snapshot	Creates snapshots for a specified cluster.
	Querying the Snapshot List	Queries the snapshot list.
	Querying Snapshot Details	Uses the snapshot ID to query the snapshot details.
	Deleting a Manual Snapshot	Deletes a specified snapshot.
	Restoring a Cluster	Restores a cluster using its snapshot.

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for obtaining a user token as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. You can obtain it from **Regions and Endpoints**. For example, the endpoint of IAM in the **ru-moscow-1a** region is **iam.ru-moscow-1.myhuaweicloud.com**.
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the **resource-path** of the API used to obtain a user token is **/v3/auth/tokens**.

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of "*Parameter name=Parameter value*". For example, **?limit=10** indicates that a maximum of 10 data records will be displayed.

NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server:

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to obtain a user token, the request method is POST. The request is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
```

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-type**, which specifies the request body type.

For details about common request headers, see **Table 3-3**.

Table 3-3 Common request header fields

Field	Description	Mandatory	Example
x-sdk-date	Time when the request is sent. The time is in **YYYYMMDD'T'HHMMSS'Z'** format. The value is the current GMT time of the system.	No	20150907T101459Z
Host	Server information of the resource being requested. The value can be obtained from the URL of the service API. The value is in the format of *hostname[:port]*. If the port number is not specified, the default port is used. The default port number for HTTPS is **443**.	No	code.test.com or code.test.com:443
Content-Type	Request body MIME type. You are advised to use the default value **application/json**. For APIs used to upload objects or images, the value can vary depending on the flow type.	Yes	application/json
Content-Length	Length of the request body. The unit is byte.	No	3495
X-Project-id	Project ID. Obtain the project ID by following the instructions in **Obtaining a Project ID**.	No	e9993fc787d94b6c886cbaa340f9c0f4

Field	Description	Mandatory	Example
X-Auth-Token	User token. The user token is a response to the API used to obtain a user token. This API is the only one that does not require authentication. The **X-Subject-Token** value contained in the returned message header is the token.	No This parameter is mandatory for token-based authentication.	The following is part of an example token: MIIPAgYJKoZIhvcNAQc-Co...ggg1BBIINPXsidG9rZ
X-Language	Request language.	No	en_us

The API used to obtain a user token does not require authentication. Therefore, only the **Content-type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://{{endpoint}}/v3/auth/tokens
Content-type: application/json
```

Request Body

The body of a request is often sent in a structured format (JSON or XML) as specified in the **Content-type** header field. The request body transfers content except the request header. If the request body contains Chinese characters, these characters must be coded in UTF-8.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to obtain a user token, the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace ***user_name***, ***domainname*** (account name), ******** (login password), and ***xxxxxxxxxxxxxxxxx*** (project ID) with the actual values. Obtain the project ID from **Regions and Endpoints** from the administrator.

NOTE

The **scope** parameter specifies where a token takes effect. In the example, the token takes effect only on the resources specified by the project. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see Obtaining a User Token.

```
POST https://{{endpoint}}/v3/auth/tokens
Content-type: application/json
```



```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "user_name",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
        "scope": {
            "project": {
                "id": "xxxxxxxxxxxxxxxxx"
            }
        }
    }
}
```

If all data required for the API request is available, you can send the request to call the API through **curl**, **Postman**, or coding. In the response to the API used to obtain a user token, **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

Calling an API can be authenticated using tokens.

Token-based Authentication

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to request headers to get permissions for calling the API.

> **NOTE**
>
> The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

When calling the API to obtain a user token, you must set **auth.scope** in the request body to **project**.

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "user_name",
                    "password": "********",
                    "domain": {
                        "name": "domainname"
                    }
                }
            }
        },
```



```
        "scope": {
            "project": {
                "name": "xxxxxxxx"
            }
        }
    }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see **Status Code**.

For example, if status code **201** is returned for calling the API used to obtain a user token, the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **content-type**.

Figure 3-1 shows the response header fields for the API used to obtain a user token. The **x-subject-token** header field is the desired user token. You can use this token to authenticate the calling of other APIs.

Figure 3-1 Header fields of the response to the request for obtaining a user token

Response Body

A response body conveys information other than the response header and is usually sent in a structured format (for example, JSON or XML) defined by the response header parameter **content-type**.

The following is part of the response body for the API used to obtain a user token.

```
{
    "token": {
        "expires_at": "2019-02-13T06:52:13.855000Z",
        "methods": [
            "password"
        ],
        "catalog": [
            {
                "endpoints": [
                    {
                        "region_id": "ru-moscow-1",
.....
```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
    "error_msg": "The format of message is error",
     "error_code": "AS.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Getting Started

This section describes how to use GaussDB(DWS) APIs to manage clusters. The procedure of the management clusters is as follows:

1. Call the API in **Authentication** to obtain the user token, which will be put into the request header for authentication in a subsequent request.
2. Call the API in **Querying the Supported Node Types** to obtain the supported node types.
3. Call the API in **Creating a Cluster** to create a cluster.
4. Call the API in **Querying the Cluster List** to obtain the cluster information.
5. Call the API in **Querying Cluster Details** to view cluster details.
6. Call the API in **Creating a Snapshot** to create a snapshot.
7. Call the API in **Querying Snapshot Details** to check whether the snapshot is successfully created.
8. Call the API in **Restoring a Cluster** to restore a cluster using its snapshot.
9. Call the API in **Deleting a Manual Snapshot** to delete an unwanted snapshot.
10. Call the API in **Deleting a Cluster** to delete a finished or unwanted cluster.

Prerequisites

- You have created a VPC, subnet, and security group and obtained their IDs. For details, see **Creating a VPC**.
- You have obtained the **IAM endpoint** and **GaussDB(DWS) endpoint**
- You have obtained the project ID. For details, see **Obtaining a Project ID**.

Managing a Cluster

The following values are examples (replace them based on the actual situation).

- IAM endpoint: **iam_endpoint**
- GaussDB(DWS) endpoint: **dws_endpoint**
- VPC ID: **219ab8a0-1272-4049-a383-8ad0b770fa11**
- Subnet ID: **d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec**

- Security group ID: **12e3c23a-8710-4b75-95e4-5c8d7f68ef3c**
- Project ID: **9bc552e6-19af-4326-800d-281a92984636**

Perform the following operations to manage the cluster.

Step 1 Before calling other APIs, call the API in **Authentication** to obtain the token and set it as an environment variable.

```
curl -H "Content-type:application/json" https://{iam_endpoint}/v3/auth/tokens -X POST -d '{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "name": "testname",
                    "domain": {
                        "name": "testname"
                    },
                    "password": "Passw0rd"
                }
            }
        },
        "scope": {
            "project": {
                "name": "ru-moscow"
            }
        }
    }
}' -v -k
```

1. Obtain the value of **X-Subject-Token** from the response header as follows. **X-Subject-Token** indicates the token.

   ```
   X-Subject-Token:MIidkgYJKoZIhvcNAQcCoIidgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
   ```

2. Run the following command to set the token as an environment variable:

   export Token={X-Subject-Token}

   X-Subject-Token is the token obtained in the preceding step.

   ```
   export Token=MIidkgYJKoZIhvcNAQcCoIidgzCCA38CAQExDTALBglghkgBZQMEAgEwgXXXXX...
   ```

Step 2 Call the API in **Querying the Supported Node Types** to obtain the supported node types.

```
curl -X GET -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://
{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/node_types -v -k
```

The request response is as follows:

```
status CODE 200
{
    "node_types": [
        {
            "spec_name": "dws.d1.xlarge",
            "id": "ebe532d6-665f-40e6-a4d4-3c51545b6a67",
            "detail": [
                {
                    "type": "vCPU",
                    "value": "4"
                },
                {
                    "value": "1675",
                    "type": "LOCAL_DISK",
                    "unit": "GB"
                },
                {
                    "type": "mem",
                    "value": "32",
```



```
                "unit": "GB"
            }
        ]
    },
    {
        "spec_name": "dws.m1.xlarge.ultrahigh",
        "id": "ebe532d6-665f-40e6-a4d4-3c51545b4f71",
        "detail": [
            {
                "type": "vCPU",
                "value": "4"
            },
            {
                "value": "512",
                "type": "SSD",
                "unit": "GB"
            },
            {
                "type": "mem",
                "value": "32",
                "unit": "GB"
            }
        ]
    }
]
}
```

Step 3 Call the API in **Creating a Cluster** to create a cluster.

The examples for configuring the cluster are as follows:

- Cluster name: **dws-demo**
- Administrator username: **dbadmin**
- Administrator password: **Dws2017demo!**
- Port: **8000**
- Node type: dws.d1.xlarge
- Number of nodes: **3**
- Elastic IP (EIP): **auto_assign**

```
curl -X POST -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
   "node_type": "dws.d1.xlarge",
      "number_of_node": 3,
      "subnet_id": "d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec",
      "security_group_id": "12e3c23a-8710-4b75-95e4-5c8d7f68ef3c",
      "vpc_id": "219ab8a0-1272-4049-a383-8ad0b770fa11",
      "port": 8000,
      "name": "dws-demo",
      "user_name": "dbadmin",
      "user_pwd": "Dws2017demo!",
      "public_ip": {
         "public_bind_type": "auto_assign"
      }
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters -v -k
```

If status code **200** is returned, the request for creating a cluster is successfully sent.

Step 4 Call the API in **Querying the Cluster List** to obtain the cluster information.

```
curl -X GET -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://
{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters -k –v
```

The request response is as follows:

```
{
    "clusters": [
```



```
    {
    "id": "7ba031f6-81f4-4670-ad20-c490b91877e5",
    "status": "AVAILABLE",
    "sub_status": "NORMAL",
    "task_status": null,
    "action_progress": null,
    "node_type":  "dws.d1.xlarge",
    "subnet_id": "d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec",
    "security_group_id": "12e3c23a-8710-4b75-95e4-5c8d7f68ef3c",
    "number_of_node": 3,
    "availability_zone": "ru-moscow-1a",
    "port": 8000,
    "name": "dws-demo",
    "version": "1.1.0",
    "vpc_id": "219ab8a0-1272-4049-a383-8ad0b770fa11",
    "user_name": "dbadmin",
    "public_ip": {
       "public_bind_type": "auto_assign",
       "eip_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
    },
    "public_endpoints": [
       {
          "public_connect_info": "10.0.0.8:8000",
          "jdbc_url": "jdbc:postgresql://10.0.0.8:8000/<YOUR_DATABASE_name>"
       }
    ],
    "endpoints": [
       {
          "connect_info": "192.168.0.10:8000",
          "jdbc_url": "jdbc:postgresql://192.168.0.10:8000/<YOUR_DATABASE_name>"
       },
       {
          "connect_info": "192.168.0.12:8000",
          "jdbc_url": "jdbc:postgresql://192.168.0.12:8000/<YOUR_DATABASE_name>"
       }
    ] ,
    "updated": "2018-01-15T12:50:06",
    "created": "2018-01-15T12:50:06"
    }
  ]
}
```


- If **status** is **CREATING**, the cluster is being created. If **status** is **AVAILABLE**, the cluster is successfully created.
- The UUID of cluster **dws-demo** is **7ba031f6-81f4-4670-ad20-c490b91877e5**. Record the UUID for subsequent use.

Step 5 Call the API in **Querying Cluster Details** to view cluster details.

```
curl -X GET -H "Content-type:application/json" -H "X-Auth-Token:$Token"
 https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters/7ba031f6-81f4-4670-ad20-c490b91877e5 -k -v
```

The request response is as follows:


```
{
  "cluster": {
    "id": "7ba031f6-81f4-4670-ad20-c490b91877e5",
    "status": "AVAILABLE",
    "name": "dws-demo",
    "updated": "2018-01-15T12:50:06",
    "created": "2018-01-15T12:50:06",
    "user_name": "dbadmin",
    "sub_status": "NORMAL",
    "task_status": null,
    "action_progress": null,
    "node_type":  "dws.d1.xlarge",
    "node_type_id": "5ddb1071-c5d7-40e0-a874-8a032e81a697",
    "subnet_id": "d23ef2e9-8b90-49b3-bc4a-fd7d6bea6bec",
```



```
        "security_group_id": "12e3c23a-8710-4b75-95e4-5c8d7f68ef3c",
        "number_of_node": 3,
        "availability_zone": "ru-moscow-1a",
        "port": 8000,
        "vpc_id": "219ab8a0-1272-4049-a383-8ad0b770fa11",
        "public_ip": {
            "public_bind_type": "auto_assign",
            "eip_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
        },
        "public_endpoints": [
        {
                "public_connect_info": "10.0.0.8:8000",
                "jdbc_url": "jdbc:postgresql://10.0.0.8:8000/<YOUR_DATABASE_name>"
         }
         ],
        "endpoints": [
        {
                "connect_info": "192.168.0.10:8000",
                "jdbc_url": "jdbc:postgresql://192.168.0.10:8000/<YOUR_DATABASE_name>"
        },
        {
                "connect_info": "192.168.0.12:8000",
                "jdbc_url": "jdbc:postgresql://192.168.0.12:8000/<YOUR_DATABASE_name>"
        }
         ],
        "version": "1.1.0",
        "maintain_window": {
            "day": "Wed",
            "start_time": "22:00",
            "end_time": "02:00"
        },
        "tags": null,
        "parameter_group": {
             "id": "157e9cc4-64a8-11e8-adc0-fa7ae01bbebc",              "name": "Default-Parameter-Group-
dws ",              "status": "In-Sync"
        }
    }
}
```


public_endpoints and **endpoints** can be queried from the response. After the cluster is successfully created, you can use **public_endpoints** or **endpoints** to access the cluster from an external source.

Step 6 Call the API in **Creating a Snapshot** to create a snapshot.

Create snapshot **snapshotForDemoCluster** for cluster **dws-demo**.

```
curl -X POST -H "Content-type:application/json" -H "X-Auth-Token:$Token" -d '{
    "snapshot": {
        "name": "snapshotForDemoCluster",
        "cluster_id": "7ba031f6-81f4-4670-ad20-c490b91877e5",
        "description": "Snapshot description"
    }
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots -k -v
```

The request response is as follows:


```
{
  "snapshot": {
      "id": "2a4d0f86-67cd-408a-8b66-017454fb7793"
  }
}
```


If status code **200** is returned, the request for creating a snapshot is successfully sent. Record **id** so that the ID can be used when you query the snapshot details later.

Step 7 Call the API in **Querying Snapshot Details** to check whether the snapshot is successfully created.


```
curl -X GET -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://
{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots/
2a4d0f86-67cd-408a-8b66-017454fb7793 -k -v
```

If the snapshot status in the response is **AVAILABLE**, the snapshot is successfully created. If the snapshot status is **CREATING**, the snapshot is being created.

```
{
    "snapshot": {
        "id": "2a4d0f86-67cd-408a-8b66-017454fb7793",
        "name": "snapshotForDemoCluster",
        "description": "Snapshot description",
        "started": "2018-01-18T13:59:23Z",
        "finished": "2018-01-18T13:01:40Z",
        "size": 500,
        "status": "AVAILABLE",
        "type": "MANUAL",
        "cluster_id": "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
    }
}
```

Step 8 Call the API in **Restoring a Cluster** to restore a cluster using its snapshot.

Restore snapshot **snapshotForDemoCluster** to new cluster **dws-restore**.

```
curl -X POST -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
    "restore": {
        "name": "dws-restore"
    }
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots/
2a4d0f86-67cd-408a-8b66-017454fb7793/actions -v -k
```

If status code **200** is returned, the cluster is successfully restored. You can check the cluster restoration status by performing operations in **Querying Cluster Details**.

Step 9 Call the API in **Deleting a Manual Snapshot** to delete an unwanted snapshot.

```
curl -X DELETE -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" https://
{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/snapshots/
2a4d0f86-67cd-408a-8b66-017454fb7793 -v -k
```

If status code **202** is returned, the snapshot is successfully deleted.

Step 10 Call the API in **Deleting a Cluster** to delete an unwanted cluster.

```
curl -X DELETE -H 'Content-type:application/json;charset=utf-8' -H "X-Auth-Token:$Token" -d '{
    "keep_last_manual_snapshot":0
}' https://{dws_endpoint}/v1.0/9bc552e6-19af-4326-800d-281a92984636/clusters/7ba031f6-81f4-4670-
ad20-c490b91877e5 -v -k
```

If status code **202** is returned, the cluster is successfully deleted.

----End

5 API Description

5.1 Cluster Management APIs

A data warehouse cluster is the smallest management unit in GaussDB(DWS). A cluster is a data warehouse that runs independently. You can manage the cluster life cycle in GaussDB(DWS).

5.1.1 Creating a Cluster

Function

This API is used to create clusters.

The cluster must run in a VPC. Before creating a cluster, you need to create a VPC and obtain the VPC and subnet IDs.

This API is an asynchronous API. It takes 10 to 15 minutes to create a cluster.

URI

- URI format
  POST /v1.0/{project_id}/clusters
- Parameter description

  Table 5-1 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Message

- Sample request


```
POST /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters

{
"cluster": {
        "node_type":  "dws.d1.xlarge.ultrahigh",
        "number_of_node": 3,
        "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
        "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
        "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
        "availability_zone": "ru-moscow-1a",
        "port": 8000,
        "name": "dws-1",
        "user_name": "dbadmin",
        "user_pwd": "Password!",
        "public_ip": {
            "public_bind_type": "auto_assign",
            "eip_id": ""
        },
        "enterprise_project_id":"aca4e50a-266f-4786-827c-f8d6cc3fbada"     }
}
```

- Parameter description

Table 5-2 Request parameter description

Parameter	Mandatory	Type	Description
cluster	Yes	**CreateClusterInfo** object	Cluster object

Table 5-3 CreateClusterInfo

Parameter	Mandatory	Type	Description
availability_zone	No	String	AZ of a cluster
public_ip	No	**PublicIp** object	Public IP address. If the parameter is not specified, public connection is not used by default.
number_of_node	Yes	Integer	Number of nodes in a cluster. The value ranges from 3 to 32.
vpc_id	Yes	String	VPC ID, which is used for configuring cluster network.

Parameter	Mandatory	Type	Description
user_name	Yes	String	Administrator username for logging in to a data warehouse cluster. The administrator username must: • Consist of lowercase letters, digits, or underscores. • Start with a lowercase letter or an underscore. • Contain 1 to 63 characters. • Cannot be a keyword of the DWS database.
security_group_id	Yes	String	ID of a security group, which is used for configuring cluster network.
number_of_cn	No	Integer	Number of deployed CNs. The value ranges from 2 to the number of cluster nodes minus 1. The maximum value is **5** and the default value is **2**.
user_pwd	Yes	String	Password of the administrator for logging in to a data warehouse cluster.
enterprise_project_id	No	String	Enterprise project ID. If no enterprise project is specified for a cluster, the default enterprise project ID **0** is used.
node_type	Yes	String	Node Type
port	No	Integer	Service port of a cluster. The value ranges from 8000 to 30000. The default value is **8000**.
name	Yes	String	Cluster name, which must be unique. The cluster name must contain 4 to 64 characters, which must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.

Parameter	Mandatory	Type	Description
subnet_id	Yes	String	Subnet ID, which is used for configuring cluster network.

Table 5-4 PublicIp

Parameter	Mandatory	Type	Description
public_bind_type	Yes	String	Binding type of an EIP. The value can be one of the following: • **auto_assign** • **not_use** • **bind_existing**
eip_id	No	String	EIP ID

Response Message

- Sample response

```
{
   "cluster": {
      "id": "7d85f602-a948-4a30-afd4-e84f47471c15"
    }
}
```

- Parameter description

Table 5-5 Response parameters

Parameter	Type	Description
cluster	**Cluster** object	Cluster object

Table 5-6 Cluster

Parameter	Type	Description
id	String	Cluster ID

Status Code

- Normal

  200
- Abnormal

Table 5-7 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.2 Querying the Cluster List

Function

This API is used to query and display the cluster list.

URI

- URI format
  GET /v1.0/{project_id}/clusters
- Parameter description

Table 5-8 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request

Sample request

```
GET /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters?
datastoretype=dws&enterprise_project_id=aca4e50a-266f-4786-827c-f8d6cc3fbada
```

Response

- Sample response

```
{
    "clusters": [
    {
    "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
```



```
      "status": "AVAILABLE",
      "sub_status": "READONLY",
      "task_status": "SNAPSHOTTING",
      "action_progress": {"SNAPSHOTTING": "20%"},
      "node_type":   "dws.d1.xlarge.ultrahigh",
      "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
      "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
      "number_of_node": 3,
      "availability_zone": "ru-moscow-1a",
      "port": 8000,
      "name": "dws-1",
      "version": "1.2.0",
      "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
      "user_name": "dbadmin",
      "public_ip": {
         "public_bind_type": "auto_assign",
         "eip_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574"
       },
      "public_endpoints": [
         {
            "public_connect_info": "10.0.0.8:8000",
            "jdbc_url": "jdbc:postgresql://10.0.0.8:8000/<YOUR_DATABASE_name>"
         }
       ],
      "endpoints": [
         {
            "connect_info": "192.168.0.12:8000",
            "jdbc_url": "jdbc:postgresql://192.168.0.12:8000/<YOUR_DATABASE_name>"
         }
       ] ,
      "updated": "2016-02-10T14:28:14Z",
      "created": "2016-02-10T14:26:14Z",
      "enterprise_project_id":"aca4e50a-266f-4786-827c-f8d6cc3fbada",         "recent_event": 6
      }
   ]
}
```

- Parameter description

Table 5-9 Response parameter description

Parameter	Type	Description
clusters	Array of **ClusterInfo** objects	List of cluster objects

Table 5-10 ClusterInfo

Parameter	Type	Description
action_progress	Map<String,String>	Task information, consisting of a key and a value. The key indicates an ongoing task, and the value indicates the progress of the ongoing task. Valid key values include: • **GROWING** • **RESTORING** • **SNAPSHOTTING** • **REPAIRING** • **CREATING** The following is an example: `"action_progress": {"SNAPSHOTTING":"16%"}`
failed_reasons	Object	Cause of failure. If the parameter is left empty, the cluster is in the normal state.
availability_zone	String	AZ
endpoints	**Endpoints** object	Private network connection information about the cluster.
task_status	String	Cluster management task. The value can be one of the following: • **RESTORING** • **SNAPSHOTTING** • **GROWING** • **REBOOTING** • **SETTING_CONFIGURATION** • **CONFIGURING_EXT_DATASOURCE** • **DELETING_EXT_DATASOURCE** • **REBOOT_FAILURE** • **RESIZE_FAILURE**
public_ip	**PublicIp** object	Public IP address. If the parameter is not specified, public connection is not used by default.

Parameter	Type	Description
sub_status	String	Sub-status of clusters in the **AVAILABLE** state. The value can be one of the following: • **NORMAL** • **READONLY** • **REDISTRIBUTING** • **REDISTRIBUTION-FAILURE** • **UNBALANCED** • **UNBALANCED | READONLY** • **DEGRADED** • **DEGRADED | READONLY** • **DEGRADED | UNBALANCED** • **UNBALANCED | REDISTRIBUTING** • **UNBALANCED | REDISTRIBUTION-FAILURE** • **READONLY | REDISTRIBUTION-FAILURE** • **UNBALANCED | READONLY | REDISTRIBUTION-FAILURE** • **DEGRADED | REDISTRIBUTION-FAILURE** • **DEGRADED | UNBALANCED | REDISTRIBUTION-FAILURE** • **DEGRADED | UNBALANCED | READONLY | REDISTRIBUTION-FAILURE** • **DEGRADED | UNBALANCED | READONLY**
number_of_node	Integer	Number of nodes
recent_event	Integer	Number of events
vpc_id	String	VPC ID
created	String	Cluster creation time. The format is **ISO8601:YYYY-MM-DDThh:mm:ssZ**.
user_name	String	Username of the administrator
security_group_id	String	Security group ID
version	String	Data warehouse version
tags	**Tags** object	Tags in a cluster
node_type	String	Node type

Parameter	Type	Description
port	Integer	Service port of a cluster. The value ranges from 8000 to 30000. The default value is **8000**.
name	String	Cluster name
subnet_id	String	Subnet ID
public_endpoints	**PublicEndpoints** object	Public network connection information about the cluster. If the parameter is not specified, the public network connection information is not used by default.
id	String	Cluster ID
updated	String	Last modification time of a cluster. The format is **ISO8601:YYYY-MM-DDThh:mm:ssZ**.
status	String	Cluster status. The value can be one of the following: • **CREATING** • **AVAILABLE** • **UNAVAILABLE** • **CREATION FAILED**

Table 5-11 PublicIp

Parameter	Mandatory	Type	Description
public_bind_type	Yes	String	Binding type of an EIP. The value can be one of the following: • **auto_assign** • **not_use** • **bind_existing**
eip_id	No	String	EIP ID

Table 5-12 Tags

Parameter	Type	Description
value	String	Value. A value can contain a maximum of 43 Unicode characters, which can be null. The first and last characters cannot be spaces. Only letters, digits, hyphens (-), and underscores (_) are allowed. It cannot contain the following characters: =*<>\,|/
key	String	Key. A key can contain a maximum of 36 Unicode characters, which cannot be null. The first and last characters cannot be spaces. Only letters, digits, hyphens (-), and underscores (_) are allowed. It cannot contain the following characters: =*<>\,|/

Table 5-13 PublicEndpoints

Parameter	Type	Description
public_connect_info	String	Public network connection information
jdbc_url	String	JDBC URL of the public network

Table 5-14 Endpoints

Parameter	Mandatory	Type	Description
connect_info	Yes	String	Private network connection information
jdbc_url	Yes	String	JDBC URL on the private network. The following is the default format: jdbc:postgresql://< connect_info>/<YOUR_DATABASE_name>

Returned Value

- Normal

  200
- Abnormal

Table 5-15 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.3 Querying Cluster Details

Function

This API is used to query cluster details.

URI

- URI format
  ```
  GET /v1.0/{project_id}/clusters/{cluster_id}
  ```
- Parameter description

  Table 5-16 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	String	Cluster ID

Request Message

Sample request

```
GET /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/b5c45780-1006-49e3-b2d5-b3229975bbc7
```

Response Message

- Sample response
  ```
  {
     "cluster": {
  ```



```
    "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
    "status": "AVAILABLE",
    "name": "dws-1",
    "updated": "2018-02-10T14:28:14Z",
    "created": "2018-02-10T14:28:14Z",
    "user_name": "dbadmin",
    "sub_status": "READONLY",
    "task_status": "SNAPSHOTTING",
    "action_progress": {"SNAPSHOTTING": "20%"},
    "node_type":   "dws.m1.xlarge.ultrahigh",

    "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
    "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
    "number_of_node": 3,
    "availability_zone": ,
    "port": 8000,
    "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
    "public_ip": {
       "public_bind_type": "auto_assign",
       "eip_id": "85b20d7e-9etypeb2a-98f3-3c8843ea3574",
       "eip_address": "100.95.157.20"
    },
    "private_ip":["192.168.0.12","192.168.0.66"],
    "public_endpoints": [
       {
          "public_connect_info": "10.0.0.8:8000",
          "jdbc_url": "jdbc:postgresql://10.0.0.8:8000/<YOUR_DATABASE_name>"
       }
     ],
    "endpoints": [
       {
          "connect_info": "192.168.0.10:8000",
          "jdbc_url": "jdbc:postgresql://192.168.0.10:8000/<YOUR_DATABASE_name>"
       }
     ],
    "version": "1.2.0",
    "maintain_window": {
       "day": "Wed",
       "start_time": "22:00",
       "end_time": "02:00"
    },
    "resize_info" : {
       "target_node_num": "6",
       "origin_node_num": "3",
       "status": "GROWING",
       "start_time": "2018-02-14T14:28:14Z",
       "origin_node_type": "dws.m1.xlarge.ultrahigh",          "target_node_type":
"dws.d2.xlarge"       },
    "enterprise_project_id": "6a6a18fe-417a-4188-9214-75fd08c22065",       "recent_event": 6,
    "tags":[          {
          "key": "key1",          "value": "value1"          },
       {
          "key": "key2",          "value": "value2"          }
     ],

     "parameter_group": {          "id": "157e9cc4-64a8-11e8-adc0-fa7ae01bbebc",
        "name": "Default-Parameter-Group-dws ",          "status": "In-Sync"
     }
  }
}
```

- Parameter description

Table 5-17 Response parameter description

Parameter	Type	Description
cluster	**ClusterDetail** object	Cluster object

Table 5-18 ClusterDetail

Parameter	Type	Description
action_progress	Map<String,String>	The key indicates an ongoing task. The options are as follows: • **GROWING** • **RESTORING** • **SNAPSHOTTING** • **REPAIRING** • **CREATING** The value indicates the task progress.
public_ip	**PublicIp** object	Public IP address. If the parameter is not specified, public connection is not used by default.
number_of_node	Integer	Number of nodes
recent_event	Integer	Number of events
vpc_id	String	VPC ID
user_name	String	Username of the administrator
security_group_id	String	Security group ID
private_ip	Array of strings	List of private network IP addresses
parameter_group	**ParameterGroup** object	Parameter group details
node_type	String	Node type
public_endpoints	Array of **PublicEndpoints** objects	Public network connection information about the cluster. If the parameter is not specified, the public network connection information is not used by default.
id	String	Cluster ID
failed_reasons	Object	Cause of failure. If the parameter is left empty, the cluster is in the normal state.

Parameter	Type	Description
availability_zone	String	AZ
endpoints	Array of **Endpoints** objects	Private network connection information about the cluster.
task_status	String	Cluster management task. The value can be one of the following: • **RESTORING** • **SNAPSHOTTING** • **GROWING** • **REBOOTING** • **SETTING_CONFIGURATION** • **CONFIGURING_EXT_DATASOURCE** • **DELETING_EXT_DATASOURCE** • **REBOOT_FAILURE** • **RESIZE_FAILURE**
sub_status	String	Sub-status of clusters in the **AVAILABLE** state. The value can be one of the following: • **NORMAL** • **READONLY** • **REDISTRIBUTING** • **REDISTRIBUTION-FAILURE** • **UNBALANCED** • **UNBALANCED | READONLY** • **DEGRADED** • **DEGRADED | READONLY** • **DEGRADED | UNBALANCED** • **UNBALANCED | REDISTRIBUTING** • **UNBALANCED | REDISTRIBUTION-FAILURE** • **READONLY | REDISTRIBUTION-FAILURE** • **UNBALANCED | READONLY | REDISTRIBUTION-FAILURE** • **DEGRADED | REDISTRIBUTION-FAILURE** • **DEGRADED | UNBALANCED | REDISTRIBUTION-FAILURE** • **DEGRADED | UNBALANCED | READONLY | REDISTRIBUTION-FAILURE** • **DEGRADED | UNBALANCED | READONLY**

Parameter	Type	Description
created	String	Cluster creation time. The format is **ISO8601:YYYY-MM-DDThh:mm:ssZ**.
node_type_id	String	Node type ID
node_detail	**NodeDetail** object	Node details
version	String	Data warehouse version
maintain_window	**MaintainWindow** object	Cluster maintenance window
resize_info	**ResizeInfo** object	Cluster scale-out details
port	Integer	Service port of a cluster. The value ranges from 8000 to 30000. The default value is **8000**.
name	String	Cluster name
subnet_id	String	Subnet ID
updated	String	Last modification time of a cluster. The format is **ISO8601:YYYY-MM-DDThh:mm:ssZ**.
status	String	Cluster status. The value can be one of the following: • **CREATING** • **AVAILABLE** • **UNAVAILABLE** • **CREATION FAILED**

Table 5-19 NodeDetail

Parameter	Mandatory	Type	Description
name	Yes	String	Node attribute
value	Yes	String	Node attribute value

Table 5-20 PublicIp

Parameter	Mandatory	Type	Description
public_bind_type	Yes	String	Binding type of an EIP. The value can be one of the following: • **auto_assign** • **not_use** • **bind_existing**
eip_id	No	String	EIP ID

Table 5-21 PublicEndpoints

Parameter	Mandatory	Type	Description
public_connect_info	No	String	Public network connection information
jdbc_url	No	String	JDBC URL of the public network. The following is the default format: jdbc:postgresql://< public_connect_info>/ <YOUR_DATABASE_name>

Table 5-22 Endpoints

Parameter	Mandatory	Type	Description
connect_info	Yes	String	Private network connection information
jdbc_url	Yes	String	JDBC URL on the private network. The following is the default format: jdbc:postgresql://< connect_info>/ <YOUR_DATABASE_name>

Table 5-23 MaintainWindow

Parameter	Mandatory	Type	Description
day	No	String	Maintenance time in each week in the unit of day. The value can be one of the following: • **Mon** • **Tue** • **Wed** • **Thu** • **Fri** • **Sat** • **Sun**
start_time	No	String	Maintenance start time. The format is **HH:mm**. The time zone is GMT+0.
end_time	No	String	Maintenance end time. The format is **HH:mm**. The time zone is GMT+0.

Table 5-24 ResizeInfo

Parameter	Type	Description
resize_status	String	Scale-out status. The value can be either of the following: • **GROWING** • **RESIZE_FAILURE**
start_time	String	Scale-out start time. The format is **ISO8601:YYYY-MM-DDThh:mm:ss**.
target_node_num	Integer	Number of nodes after the scale-out
origin_node_num	Integer	Number of nodes before the scale-out

Table 5-25 ParameterGroup

Parameter	Mandatory	Type	Description
id	Yes	String	Parameter group ID
name	Yes	String	Parameter group name

Parameter	Mandatory	Type	Description
status	Yes	String	Cluster parameter status. The value can be one of the following: • In-Sync: synchronized • Applying: in application • Pending-Reboot: restart for the modification to take effect • Sync-Failure: application failure

Status Code

- Normal
  200
- Abnormal

Table 5-26 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.4 Querying the Supported Node Types

Function

This API is used to query the node types supported by GaussDB(DWS).

URI

- URI format
  GET /v1.0/{project_id}/node_types

Table 5-27 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Message

- Sample request

```
GET /v1.0/89cd04f168b84af6be287f71730fdb4b/node_types
```

Response Message

- Sample response

```
status CODE 200
{
   "node_types": [
      {
         "spec_name": "dws.d1.xlarge",
         "id": "ebe532d6-665f-40e6-a4d4-3c51545b6a67",
         "detail": [
            {
               "type": "vCPU",
               "value": "4"
            },
            {
               "value": "1675",
               "type": "LOCAL_DISK",
               "unit": "GB"
            },
            {
               "type": "mem",
               "value": "32",
               "unit": "GB"
            }
         ]
      },
      {
         "spec_name": "dws.m1.xlarge.ultrahigh",
         "id": "ebe532d6-665f-40e6-a4d4-3c51545b4f71",
         "detail": [
            {
               "type": "vCPU",
               "value": "4"
            },
            {
               "value": "512",
               "type": "SSD",
               "unit": "GB"
            },
            {
               "type": "mem",
               "value": "32",
               "unit": "GB"
            }
         ]
      }
   ]
}
```


- Parameter description

Table 5-28 Response parameter description

Parameter	Type	Description
node_types	Array of **NodeTypes** objects	List of node types

Table 5-29 NodeTypes

Parameter	Type	Description
detail	**Detail** object	Node type details
id	String	Node type ID
spec_name	String	Name of a node type

Table 5-30 Detail

Parameter	Type	Description
unit	String	Attribute unit
type	String	Attribute type
value	String	Attribute value

Status Code

- Normal

  200
- Abnormal

Table 5-31 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.

Returned Value	Description
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.5 Deleting a Cluster

Function

This API is used to delete clusters. All resources, including customer data, of the deleted cluster will be released. For data security reasons, create a snapshot for the cluster that you want to delete before deleting the cluster.

URI

- URI format
  ```
  DELETE /v1.0/{project_id}/clusters/{cluster_id}
  ```
- Parameter description

  Table 5-32 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	String	ID of the cluster to be deleted

Request Message

- Sample request
  ```
  DELETE /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90
  {
     "keep_last_manual_snapshot":0
  }
  ```
- Parameter description

  Table 5-33 Request parameter description

Parameter	Mandatory	Type	Description
keep_last_manual_snapshot	Yes	Integer	Number of snapshots that you want to reserve for a cluster

Response Message

Sample response

```
status CODE 202
```

Status Code

- Normal
  202
- Abnormal

Table 5-34 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.6 Restarting a Cluster

Function

This API is used to restart clusters.

URI

- URI format
  ```
  POST /v1.0/{project_id}/clusters/{cluster_id}/restart
  ```
- Parameter description

Table 5-35 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	String	ID of the cluster to be restarted

Request Message

- Sample request

```
POST /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/
restart
{
    "restart":{}
}
```

- Parameter description

Table 5-36 Request parameter description

Parameter	Mandatory	Type	Description
restart	Yes	Object	Restart flag

Response Message

Sample response

```
status CODE 200
```

Status Code

- Normal

  200

- Abnormal

Table 5-37 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.7 Scaling Out Clusters

Function

This API is used to scale out clusters.

URI

- URI format
  POST /v1.0/{project_id}/clusters/{cluster_id}/resize
- Parameter description

Table 5-38 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	String	ID of the cluster to be scaled out

Request Message

- Sample request

  scale_out sample API is as follows:

```
POST /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/
resize
{
   "scale_out":{
      "count":3
   }
}
```

- Parameter description

Table 5-39 Request parameter description

Parameter	Mandatory	Type	Description
scale_out	No	**ScaleOut** object	Scale out an object.

Table 5-40 ScaleOut

Parameter	Mandatory	Type	Description
count	Yes	Integer	Number of nodes to be added

Response Message

Sample response

```
status CODE 200
```


Status Code

- Normal
  200
- Abnormal

Table 5-41 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.1.8 Resetting a Password

Function

This API is used to reset the password of cluster administrator.

URI

- URI format
  ```
  POST /v1.0/{project_id}/clusters/{cluster_id}/reset-password
  ```
- Parameter description

Table 5-42 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
cluster_id	Yes	String	ID of the cluster whose password is to be reset

Request Message

- Sample request
  ```
  POST /v1.0/89cd04f168b84af6be287f71730fdb4b/clusters/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/reset-password
  {
  ```



```
    "new_password": "NewPassword!"
}
```

- Parameter description

Table 5-43 Request parameter description

Parameter	Mandatory	Type	Description
new_password	Yes	String	New administrator password for logging in to a data warehouse cluster A password must conform to the following rules: ● Contains 8 to 32 characters. ● Cannot be the same as the username or the username written in reverse order. ● Contains at least three types of the following: – Lowercase letters – Uppercase letters – Digits – Special characters ~!@#%^&*()-_=+|[{}];:,<.>/? ● Cannot be the same as a historical password. ● Cannot be a weak password.

Response Message

Sample response

```
status CODE 200
```

Status Code

- Normal

  200

- Abnormal

Table 5-44 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.

Returned Value	Description
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.2 Snapshot Management APIs

A GaussDB(DWS) snapshot is a complete backup of a cluster. Snapshots are stored in the storage space of Object Storage Service (OBS). A snapshot can be used to restore a cluster to a newly created one that has the same flavor. Currently, you can only restore a cluster to a new one.

5.2.1 Creating a Snapshot

Function

This API is used to create snapshots for a specified cluster.

URI

- URI format
  ```
  POST /v1.0/{project_id}/snapshots
  ```
- Parameter description

  Table 5-45 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Message

- Sample request
  ```
  POST /v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots
  {
     "snapshot": {
        "name": "snapshot-3",
        "cluster_id": "44b277eb-39be-4921-be31-3d61b43651d7",
        "description": "Snapshot-3 description"
      }
  }
  ```


- Parameter description

Table 5-46 Request parameter description

Parameter	Mandatory	Type	Description
snapshot	Yes	**Snapshot** object	Snapshot object

Table 5-47 Snapshot

Parameter	Mandatory	Type	Description
cluster_id	Yes	String	ID of the cluster for which you want to create a snapshot
name	Yes	String	Snapshot name, which must be unique and start with a letter. It consists of 4 to 64 characters, which are case-insensitive and contain letters, digits, hyphens (-), and underscores (_) only.
description	No	String	Snapshot description. If no value is specified, the description is empty.

Response Message

- Sample response

```
status CODE 200
{
    "snapshot": {
        "id": "2a4d0f86-67cd-408a-8b66-017454fb7793"
    }
}
```

- Parameter description

Table 5-48 Response parameter description

Parameter	Type	Description
snapshot	**SnapshotResp** object	Snapshot object

Table 5-49 SnapshotResp

Parameter	Type	Description
id	String	Snapshot ID

Status Code

- Normal
  200
- Abnormal

Table 5-50 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.2.2 Querying the Snapshot List

Function

This API is used to query the snapshot list.

URI

- URI format
  GET /v1.0/{project_id}/snapshots
- Parameter description

Table 5-51 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

Request Message

Sample request

```
GET /v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots
```

Response Message

- Sample response

```
status CODE 200
{
   "snapshots": [
      {
         "id": "2a4d0f86-67cd-408a-8b66-017454fb7793",
         "name": "snapshot-1",
         "description": "",
         "started": "2016-08-23T03:59:23Z",
         "finished": "2016-08-23T04:01:40Z",
         "size": 500,
         "status": "AVAILABLE",
         "type": "MANUAL",
         "cluster_id": "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
      },
      {
         "id": "4af11460-06ec-48a4-b3ad-0e3bbdcd8ab1",
         "name": "snapshot-2",
         "description": "",
         "started": "2016-08-23T18:20:00Z",
         "finished": "2016-08-23T18:22:12Z",
         "size": 500,
         "status": "AVAILABLE",
         "type": "MANUAL",
         "cluster_id": "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
      }
   ]
}
```

- Parameter description

Table 5-52 Response parameter description

Parameter	Type	Description
snapshots	Array of **Snapshots** objects	List of snapshot objects

Table 5-53 Snapshots

Parameter	Type	Description
cluster_id	String	ID of the cluster for which a snapshot is created
size	Double	Snapshot size, in GB.
name	String	Snapshot name
description	String	Snapshot description

Parameter	Type	Description
finished	String	Time when a snapshot has been created. The format is **ISO8601: YYYY-MM-DDThh:mm:ssZ**.
started	String	Time when a snapshot is being created.
id	String	Snapshot ID
type	String	Snapshot type
status	String	Snapshot status

Status Code

- Normal
  200
- Abnormal

Table 5-54 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.2.3 Querying Snapshot Details

Function

This API is used to query snapshot details by using the snapshot ID.

URI

- URI format
  GET /v1.0/{project_id}/snapshots/{snapshot_id}
- Parameter description

Table 5-55 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
snapshot_id	Yes	String	Snapshot ID

Request Message

Sample request

```
GET /v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/b5c45780-1006-49e3-b2d5-b3229975bbc7
```

Response Message

- Sample response

```
status CODE 200
{
   "snapshot": {
      "id": "2a4d0f86-67cd-408a-8b66-017454fb7793",
      "name": "snapshot-1",
      "description": "snapshot description",
      "started": "2016-08-23T03:59:23Z",
      "finished": "2016-08-23T04:01:40Z",
      "size": 500,
      "status": "AVAILABLE",
      "type": "MANUAL",
      "cluster_id": "4f87d3c4-9e33-482f-b962-e23b30d1a18c"
   }
}
```

- Parameter description

Table 5-56 Response parameter description

Parameter	Type	Description
snapshot	**SnapshotDetail** object	Snapshot object

Table 5-57 SnapshotDetail

Parameter	Type	Description
cluster_id	String	ID of the cluster for which a snapshot is created
size	Double	Snapshot size, in GB.
name	String	Snapshot name
description	String	Snapshot description

Parameter	Type	Description
finished	String	Time when a snapshot has been created. The format is **ISO8601: YYYY-MM-DDThh:mm:ssZ**.
started	String	Time when a snapshot is being created. The format is **ISO8601: YYYY-MM-DDThh:mm:ssZ**.
id	String	Snapshot ID
type	String	Snapshot type
status	String	Snapshot status

Status Code

- Normal

  200
- Abnormal

Table 5-58 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.2.4 Deleting a Manual Snapshot

Function

This API is used to delete a specified snapshot.

URI

- URI format

  ```
  DELETE /v1.0/{project_id}/snapshots/{snapshot_id}
  ```
- Parameter description

Table 5-59 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
snapshot_id	Yes	String	Snapshot ID

Request Message

Sample request

```
DELETE /v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90
```

Response Message

Sample response

```
status CODE 202
```

Status Code

- Normal
  202
- Abnormal

Table 5-60 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

5.2.5 Restoring a Cluster

Function

This API is used to restore clusters using the snapshot.

URI

- URI format
  POST /v1.0/{project_id}/snapshots/{snapshot_id}/actions
- Parameter description

Table 5-61 URI parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	Project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.
snapshot_id	Yes	String	ID of the snapshot to be restored

Request Message

- Sample request

```
POST /v1.0/89cd04f168b84af6be287f71730fdb4b/snapshots/4ca46bf1-5c61-48ff-b4f3-0ad4e5e3ba90/actions
{"restore": {
        "name": "dws-1",
        "subnet_id": "374eca02-cfc4-4de7-8ab5-dbebf7d9a720",
        "security_group_id": "dc3ec145-9029-4b39-b5a3-ace5a01f772b",
        "vpc_id": "85b20d7e-9eb7-4b2a-98f3-3c8843ea3574",
        "availability_zone": "ru-moscow-1a",
        "port": 8000,
        "public_ip": {
            "public_bind_type": "auto_assign",
            "eip_id": ""
        },
        "enterprise_project_id":"aca4e50a-266f-4786-827c-f8d6cc3fbada"
    }
}
```

- Parameter description

Table 5-62 Request parameter description

Parameter	Mandatory	Type	Description
restore	Yes	**Restore** object	Object to be restored

Table 5-63 Restore

Parameter	Mandatory	Type	Description
availability_zone	No	String	AZ of a cluster. The default value is the same as that of the original cluster.

Parameter	Mandatory	Type	Description
public_ip	No	**PublicIp** object	Public IP address. If the parameter is not specified, public connection is not used by default.
port	No	Integer	Service port of a cluster. The value ranges from 8000 to 30000.
vpc_id	No	String	VPC ID, which is used for configuring cluster network. The default value is the same as that of the original cluster.
name	Yes	String	Cluster name, which must be unique. The cluster name must contain 4 to 64 characters, which must start with a letter. Only letters, digits, hyphens (-), and underscores (_) are allowed.
security_group_id	No	String	Security group ID, which is used for configuring cluster network. The default value is the same as that of the original cluster.
subnet_id	No	String	Subnet ID, which is used for configuring cluster network. The default value is the same as that of the original cluster.

Table 5-64 PublicIp

Parameter	Mandatory	Type	Description
public_bind_type	Yes	String	Binding type of an EIP. The value can be one of the following: • **auto_assign** • **not_use** • **bind_existing**
eip_id	No	String	EIP ID

Response Message

- Sample response

```
{
   "cluster": {
      "id": "7d85f602-a948-4a30-afd4-e84f47471c15"
    }
}
```

- Parameter description

Table 5-65 Response parameter description

Parameter	Type	Description
cluster	**Cluster** object	Cluster object

Table 5-66 Cluster

Parameter	Type	Description
id	String	Cluster ID

Status Code

- Normal

  200
- Abnormal

Table 5-67 Returned value description

Returned Value	Description
400 Bad Request	Request error.
401 Unauthorized	Authentication failed.
403 Forbidden	You do not have the permission to perform the operation.
404 Not Found	The requested resource was not found.
500 Internal Server Error	Internal service error.
503 Service Unavailable	The service is unavailable.

6 Application Cases

6.1 Using Postman to Call the API for Creating a Snapshot

Scenarios

Use Postman to call the API for creating a snapshot. For details, see **Calling APIs**.

Involved APIs

When this API is called, authentication is required. The following APIs are involved:

- **Authentication**: This API is used to authenticate identity to obtain the permission to operate APIs.
- **Creating a Snapshot**: This API is used to create a snapshot for a cluster.

Procedure

Step 1 Configure the environment.

NOTE

Download Postman (7.24.0 is recommended) and call the snapshot creation service.

Step 2 Call services.

1. Obtain the token for authentication.

   Create a POST request on Postman, select the Body configuration item, and enter the username, password, account name, and project ID. For details about how to obtain the project ID, see **Obtaining a Project ID**.

   NOTE

   The region where the CLOUD service for obtaining the token is located must be the same as the region where the called service is located. Otherwise, the calling fails.

2. Click **Send** in the upper right corner to send **POST https://{{endpoint}}/v3/auth/tokens**. Obtain the token value in the returned result. The token validity period is 24 hours.

3. Call the API for creating a snapshot.
   - Create a POST request on Postman, select **Headers**, add the corresponding KEY and VALUE, and copy the token value to **X-Auth-Token**. The region in the request URL must be the same as that of the service that is called. You can obtain the region from **Regions and Endpoints**.
   - Click **Body** and add the request parameter values to the request. Parameters vary by the service. For details, see **API Description**.
   - Click **Send** in the upper right corner to send **POST https://{{endpoint}}/v3/auth/tokens** and view the creation result.

     ```
     {
       "snapshot": {
           "id": "809bcf6e-4022-496d-9320-bbc700695692"
         }
     }
     ```

----End

7 Permissions Policies and Supported Actions

This section describes fine-grained permissions management for your GaussDB(DWS) service using IAM. You can skip this section if your cloud account already satisfies your needs.

By default, new IAM users do not have permissions assigned. You need to add the users to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

You can grant users permissions by using roles and policies. Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

An account has all of the permissions required to call all APIs, but IAM users must have the required permissions specifically assigned. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query the GaussDB(DWS) cluster list using an API, the user must have been granted permissions that allow the **dws:openAPICluster:list** action.

Supported Actions

DWS provides system-defined policies that can be directly used in IAM. You can also create custom policies and use them to supplement system-defined policies, implementing more refined access control. Actions supported by policies are specific to APIs. The following are common concepts related to policies:

- **Permissions**: Allow or deny operations on specified resources under specific conditions.
- **APIs**: RESTful APIs that can be called in a custom policy.

- **Actions**: Added to a custom policy to control permissions for specific operations.
- **IAM or enterprise projects**: Type of projects for which an action will take effect. Policies that contain actions supporting both IAM and enterprise projects can be assigned to user groups and take effect in both IAM and Enterprise Management. Policies that only contain actions supporting IAM projects can be assigned to user groups and only take effect for IAM. Such policies will not take effect if they are assigned to user groups in Enterprise Management.

  NOTE

  The check mark (√) indicates that an action takes effect. The cross mark (x) indicates that an action does not take effect.

GaussDB(DWS) supports the following actions that can be defined in custom policies:

- **Managing Clusters**
- **Managing Snapshots**

Managing Clusters

Permissions	APIs	Actions	IAM Project	Enterprise Project
Creating clusters	POST /v1.0/{project_id}/clusters	dws:openAPICluster:create	√	x
Querying the cluster list	GET /v1.0/{project_id}/clusters	dws:openAPICluster:list	√	x
Querying cluster details	GET /v1.0/{project_id}/clusters/{cluster_id}	dws:openAPICluster:getDetail	√	x
Querying the node type	GET /v1.0/{project_id}/node_types	dws:openAPIFlavors:get	√	x
Deleting clusters	DELETE /v1.0/{project_id}/clusters/{cluster_id}	dws:openAPICluster:delete	√	x
Restarting clusters	POST /v1.0/{project_id}/clusters/{cluster_id}/restart	dws:openAPICluster:restart	√	x

Permissions	APIs	Actions	IAM Project	Enterprise Project
Scales out a cluster.	POST /v1.0/{project_id}/clusters/{cluster_id}/resize	dws:openAPICluster:resize	√	x
Resetting the cluster's administrator password	POST /v1.0/{project_id}/clusters/{cluster_id}/reset_password	dws:openAPICluster:resetPassword	√	x

Managing Snapshots

Permissions	APIs	Actions	IAM Project	Enterprise Project
Creating snapshots	POST /v1.0/{project_id}/snapshots	dws:openAPISnapshot:create	√	x
Querying the snapshot list	GET /v1.0/{project_id}/snapshots	dws:openAPISnapshot:list	√	x
Querying snapshot details	GET /v1.0/{project_id}/snapshots/{snapshot_id}	dws:openAPISnapshot:getDetail	√	x
Deleting snapshots	DELETE /v1.0/{project_id}/snapshots/{snapshot_id}	dws:openAPISnapshot:delete	√	x
Restoring clusters	POST /v1.0/{project_id}/snapshots/{snapshot_id}/actions	dws:openAPISnapshot:restore	√	x

8 Appendix

8.1 Status Code

Table 8-1 describes the status code.

Table 8-1 Status code

Status Code	Code	Description
100	Continue	The client continues sending the request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the server.
101	Switching Protocols	Switching protocols. The target protocol must be more advanced than the source protocol. For example, the current HTTP protocol is switched to a later version.
201	Created	The request for creating a resource has been fulfilled.
202	Accepted	The request has been accepted, but the processing has not been completed.
203	Non-Authoritative Information	The server successfully processed the request, but is returning information that may be from another source.
204	NoContent	The server has successfully processed the request, but has not returned any content. The status code is returned in response to an HTTP OPTIONS request.

Status Code	Code	Description
205	Reset Content	The server has fulfilled the request, but the requester is required to reset the content.
206	Partial Content	The server has processed certain GET requests.
300	Multiple Choices	There are multiple options for the location of the requested resource. The response contains a list of resource characteristics and addresses from which the user or user agent (such as a browser) can choose the most appropriate one.
301	Moved Permanently	The requested resource has been assigned a new permanent URI, and the new URI is contained in the response.
302	Found	The requested resource resides temporarily under a different URI.
303	See Other	Retrieve a location. The response to the request can be found under a different URI and should be retrieved using a GET or POST method.
304	Not Modified	The requested resource has not been modified. When the server returns this status code, it does not return any resources.
305	Use Proxy	The requested resource must be accessed through a proxy.
306	Unused	The HTTP status code is no longer used.
400	BadRequest	Invalid request. The client should not repeat the request without modifications.
401	Unauthorized	The status code is returned after the client provides the authentication information, indicating that the authentication information is incorrect or invalid.
402	Payment Required	This status code is reserved for future use.
403	Forbidden	The server understood the request, but is refusing to fulfill it. The client should not repeat the request without modifications.
404	NotFound	The requested resource cannot be found. The client should not repeat the request without modifications.

Status Code	Code	Description
405	MethodNotAllowed	The method specified in the request is not supported for the requested resource. The client should not repeat the request without modifications.
406	Not Acceptable	The server cannot fulfill the request according to the content characteristics of the request.
407	Proxy Authentication Required	This status code is similar to 401, but indicates that the client must first authenticate itself with the proxy.
408	Request Time-out	The request timed out. The client may repeat the request without modifications at any later time.
409	Conflict	The request could not be completed due to a conflict with the current state of the resource. This status code indicates that the resource that the client attempts to create already exits, or the request fails to be processed because of the update of the conflict request.
410	Gone	The requested resource is no longer available. The status code indicates that the requested resource has been deleted permanently.
411	Length Required	The server refuses to process the request without a defined Content-Length.
412	Precondition Failed	The server does not meet one of the preconditions that the requester puts on the request.
413	Request Entity Too Large	The request is larger than that a server is able to process. The server may close the connection to prevent the client from continuing the request. If the server cannot process the request temporarily, the response will contain a Retry-After header field.
414	Request-URI Too Large	The URI provided was too long for the server to process.
415	Unsupported Media type	The server is unable to process the media format in the request.
416	Requested range not satisfiable	The requested range is invalid.
417	Expectation Failed	The server fails to meet the requirements of the Expect request-header field.

Status Code	Code	Description
422	UnprocessableEntity	The request is well-formed but is unable to be processed due to semantic errors.
429	TooManyRequests	The client has sent more requests than its rate limit is allowed within a given amount of time, or the server has received more requests than it is able to process within a given amount of time. In this case, it is advisable for the client to re-initiate requests after the time specified in the Retry-After header of the response expires.
500	InternalServerError	The server is able to receive the request but it could not understand the request.
501	Not Implemented	The server does not support the requested function.
502	Bad Gateway	The server is acting as a gateway or proxy and receives an invalid request from a remote server.
503	ServiceUnavailable	The requested service is invalid. The client should not repeat the request without modifications.
504	ServerTimeout	The request cannot be fulfilled within a given time. This status code is returned to the client only when the **Timeout** parameter is specified in the request.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

8.2 Error Code

No data is returned if an API fails to be called. You can locate the cause of error according to the error code of each API. When the calling fails, HTTP status code 4*xx* or 5*xx* is returned. The returned message body contains the specific error code and error information. If you fail to locate the cause of error, contact the customer service and technical support and provide the error code so that we can help you solve the problem as soon as possible.

Table 8-2 Error code

Error Code	Status Code	Description	Solution
DWS.0001	400	Incorrect parameter.	Enter the correct parameter according to the corresponding parameter description and try again later.
DWS.0005	500	Server failure.	Contact the customer service or try again later.
DWS.0006	400	The request is empty. Enter a request parameter.	Enter the request parameter and try again later.
DWS.0011	409	This operation cannot be performed because another operation is being performed on the instance or the instance is faulty. Please try again later.	Try again later. If the error persists, contact the customer service.
DWS.0015	403	Resource not found or permission denied.	Verify the access resources and access permissions and try again later.
DWS.0022	404	The instance does not exist or has been deleted.	Enter a correct instance and try again later.
DWS.0030	400	The number of snapshots reaches the quota.	Apply for a larger quota or delete unnecessary snapshots and try again.
DWS.0031	400	The object already exists.	Enter a correct object and try again later.
DWS.0032	403	You are not authorized to perform the operation. Check the account permission on the IAM.	Check the account permission through the IAM.
DWS.0048	400	The cluster does not support the operation.	Verify the cluster and cluster status and try again later.
DWS.1102	400	The instance name already exists.	Enter a correct instance and try again later.
DWS.1112	400	The number of instances reaches the quota.	Apply for a larger quota or delete unnecessary clusters and try again.

Error Code	Status Code	Description	Solution
DWS.2026	400	This is a weak password. Enter a strong password.	Enter a valid password according to the password description.
DWS.2067	400	The input parameter is illegal.	Enter the correct parameter according to the parameter description and try again later.
DWS.2073	400	A historical password cannot be used again.	Enter a password that is different from historical passwords.
DWS.3026	400	The backup file does not exist.	Ensure that the backup file is correct and try again later.
DWS.3027	400	The cluster does not exist.	Ensure that the cluster is correct and try again later.
DWS.3029	400	The backup file of a specific time point does not exist.	Ensure that the backup file is correct and try again later.
DWS.3030	400	The volume size of the new instance should not be less than that of the original one.	Rectify the fault according to the prompted error message and try again later.
DWS.5001	400	Invalid instance name.	Enter a valid instance name according to the instance name description and try again later.
DWS.5002	400	Invalid database type.	Enter a valid database type according to the database parameter description and try again later.
DWS.5003	400	Invalid database version.	Enter a valid database version according to the database parameter description and try again later.
DWS.5004	400	Empty datastore field.	Enter the correct datastore according to the actual situation and try again later.

Error Code	Status Code	Description	Solution
DWS.5005	400	Database type or version not supported.	Enter a valid database type and version according to the database parameter description and try again later.
DWS.5006	400	Invalid flavor.	Select the correct flavor and try again later.
DWS.5007	400	The selected flavor does not exist.	Select the correct flavor and try again later.
DWS.5013	400	Invalid password of database **root**.	Enter a valid password according to the password description.
DWS.5014	400	Invalid VPC ID.	Enter a valid ID according to the description of parameter **vpc_id**.
DWS.5015	400	Invalid subnet ID.	Enter a valid ID according to the description of parameter **subnet_id**.
DWS.5016	400	Invalid security group ID.	Enter a valid ID according to the description of the security group ID parameter.
DWS.5017	400	Invalid retention period of automated backups.	Enter a proper retention period and try again later.
DWS.5018	400	Invalid automated backup period.	Enter a proper automated backup period and try again later.
DWS.5020	400	The time when automated backup starts is invalid.	Enter a proper automated backup start time and try again later.
DWS.5021	404	The VPC does not exist or does not belong to the user.	Enter a correct VPC and try again later.
DWS.5022	404	The subnet does not exist or does not belong to the VPC.	Enter a correct subnet and try again later.
DWS.5023	404	The security group does not exist or does not belong to the VPC.	Enter a correct security group and try again later.

Error Code	Status Code	Description	Solution
DWS.5026	400	Illegal AZ.	Enter a correct AZ and try again later.
DWS.5027	400	Invalid AZ.	Enter a correct AZ and try again later.
DWS.5045	400	The current tenant is not allowed to perform this operation.	Ensure that the current tenant is the same as the tenant to which the cluster belongs and try again.
DWS.5047	400	Invalid instance quantity.	Enter a correct instance quantity.
DWS.5048	400	Invalid extended parameter.	Enter a valid extended parameter according to the corresponding description of the extended parameter.
DWS.5049	400	Invalid cluster name.	Enter a valid cluster name according to the cluster name description.
DWS.5050	409	The cluster name already exists.	Enter a correct cluster name and try again later.
DWS.5052	400	Invalid value of the AZ.	Enter a correct AZ and try again later.
DWS.5053	400	The value of locality between instances is invalid.	Enter a correct locality value and try again later.
DWS.5054	400	The parameter group ID is invalid.	Enter a correct parameter group ID and try again later.
DWS.5055	400	The flavor ID is invalid.	Enter a correct flavor ID and try again later.
DWS.5057	400	The cluster name pattern is invalid.	Enter a valid cluster name according to the cluster name description.
DWS.5059	400	Database username contains invalid characters.	Enter a valid database username according to the database username description.
DWS.5060	400	Integer type is of wrong range.	Enter a valid Integer type and try again later.
DWS.5061	400	The volume type is illegal.	Enter a valid volume type and try again later.

Error Code	Status Code	Description	Solution
DWS.5062	400	The volume tag is illegal.	Enter a valid volume tag and try again later.
DWS.5063	400	Invalid Specific field.	Enter a correct Specific value and try again later.
DWS.5068	400	Invalid disk parameters.	Enter correct disk parameters and try again later.
DWS.5070	400	Flavor ref is invalid in xml.	Enter correct flavor ref and try again later.
DWS.5071	400	The volume type is invalid in XML.	Enter a correct disk type and try again later.
DWS.5072	400	The security group ID is invalid.	Enter a valid security group ID and try again later.
DWS.5074	400	The subnet does not belong to the VPC.	Enter a correct subnet and try again later.
DWS.5079	400	The backup period is not permitted.	Enter a proper backup period and try again later.
DWS.5080	400	The backup retention period is not permitted.	Enter a proper backup retention duration and try again later.
DWS.5082	400	Database user is not supported.	Enter a valid database username according to the database username description.
DWS.5084	400	Incorrect volume configuration in the XML file.	Enter a correct volume value and try again later.
DWS.5086	400	The same instance type does not meet flavor consistency requirements.	Enter the correct flavor and try again later.
DWS.5087	400	The same instance type does not meet disk consistency requirements.	Enter a correct disk and try again later.
DWS.5093	400	Flavor information is invalid.	Enter valid flavor information and try again later.
DWS.5096	400	Invalid disk quantity.	Enter valid disk quantity and try again later.

Error Code	Status Code	Description	Solution
DWS.5097	400	Region or AZ does not exist.	Enter a correct region or AZ and try again later.
DWS.5098	400	Invalid EIP binding type.	Enter a correct EIP binding type and try again later.
DWS.5106	400	Invalid instance type.	Enter a valid instance type and try again later.
DWS.5107	400	Invalid cluster mode.	Enter a valid disk mode and try again later.
DWS.5110	400	Parameter groups of instances of the same type are inconsistent.	Enter the correct parameter group and try again later.
DWS.5111	400	The backup file ID does not exist.	Enter a correct backup file ID and try again later.
DWS.5144	400	Invalid password of the cluster administrator.	Enter a valid cluster administrator password according to the description of the cluster administrator's password.
DWS.5145	400	The administrator password cannot contain the username or the reverse username.	Enter a valid administrator password as prompted and try again later.
DWS.5149	400	The snapshot ID is invalid.	Enter a valid snapshot ID according to the snapshot ID description.
DWS.5165	400	Invalid null content in the data file.	Enter the correct control content and try again later.
DWS.5166	400	Invalid noescaping parameter in text format	Enter the correct noescaping parameter and try again later.
DWS.5167	400	Invalid header parameter in the data file. The supported values are true/on and false/off.	Enter the correct header parameter and try again later.
DWS.5168	400	Invalid **fill_missing_fields** parameter in the data file. The supported values are **true/on** and **false/off**.	Enter the correct **fill_missing_fields** parameter and try again later.

Error Code	Status Code	Description	Solution
DWS.5169	400	Invalid **ignore_extra_data** parameter in the data file. The supported values are **true/on** and **false/off**.	Enter the correct **ignore_extra_data** parameter and try again later.
DWS.5170	400	Invalid number of incorrect data formats during data import.	Enter the correct number of incorrect data formats and try again later.
DWS.5171	400	Invalid username or password for logging in to the database.	Enter the correct username or password and try again later.
DWS.5172	400	The database does not exist.	Enter the correct database name and try again later.
DWS.5173	400	The schema where the database table resides does not exist.	Enter a correct schema and try again later.
DWS.5174	400	The database table does not exist.	Enter the correct database table and try again later.
DWS.5175	400	The database username cannot be blank.	Enter the username and try again later.
DWS.5176	400	The password corresponding to the database username cannot be blank.	Enter the password and try again later.
DWS.5177	400	Name of the database where the table to which data is loaded is located cannot be blank.	Enter the database name and try again later.
DWS.5178	400	Schema where the table to which data is loaded is located cannot be blank.	Enter the schema and try again later.
DWS.5179	400	Database table to which data is loaded cannot be blank.	Enter the database table and try again later.
DWS.5180	400	Invalid path for OBS file loading.	Enter a correct OBS loading path and try again later.
DWS.5181	400	Failed to load data. The error information is #failedReason#.	Rectify the fault according to the error information and try again later.
DWS.5182	400	The current job is stopped.	Select a loading job and cancel it.

Error Code	Status Code	Description	Solution
DWS.5183	400	The current job has been canceled.	Select a loading job and cancel it.
DWS.5184	400	Invalid format of the data source file. The supported file formats are CSV and TEXT.	Enter a correct data source file format and try again later. The supported file formats are CSV and TEXT.
DWS.5185	400	Invalid encoding format of the data file. The supported file formats are GBK, UTF8, Latin1, and SQL_ASCII.	Enter a correct encoding format and try again later. The supported file formats are GBK, UTF8, Latin1, and SQL_ASCII.
DWS.5186	400	Invalid field delimiter of row data in a data source file.	Enter a correct field delimiter and try again later.
DWS.5187	400	Invalid quote character for a CSV file.	Enter a correct quote character and try again later.
DWS.5188	400	Invalid escape character for a CSV file.	Enter a correct escape character and try again later.
DWS.5189	400	Failed to obtain the AK/SK.	Contact the customer service or try again later.
DWS.5191	400	The tag does not exist.	Enter the correct tag and try again later.
DWS.5194	400	The number of tags exceeds the maximum value.	Verify the number of tags for the resource and try again later.
DWS.5197	400	The tag is invalid.	Enter the correct tag and try again later.
DWS.5200	400	The flavor has been sold out. Switch to another AZ or region.	Create a cluster of another flavor or switch to another AZ or region.
DWS.5206	400	Databases of this version do not support the function.	Ensure that the interface parameters match the cluster version. If the parameters do not match the version, correct the parameters and try again.

Error Code	Status Code	Description	Solution
DWS.6000	400	Cluster creation fails.	Check the account's remaining quota. If it is greater than the amount of resources applied by the user, contact the customer service.
DWS.6001	400	Cluster scale-out failed.	Check the account's remaining quota of nodes. If it is greater than the number of nodes to be added, contact the customer service.
DWS.6002	400	Cluster restart failed.	Contact the customer service or try again later.
DWS.6003	400	Cluster restoration failed.	Check the account's remaining quota. If it is greater than the applied quota, contact the customer service.
DWS.9999	500	Request processing failed.	Contact the customer service or try again later.

8.3 Creating a VPC

Background

Before creating a cluster, you need to create a VPC to provide a secure and isolated network environment for using GaussDB(DWS).

If you have already created a VPC, you do not need to create it again.

NOTE

For details about how to create a VPC, see "Creating a VPC" in the *Virtual Private Cloud User Guide*.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Network**, click **Virtual Private Cloud**.

Step 3 On the **Virtual Private Cloud** page, click **Create VPC** to create a VPC.

Step 4 Obtain the VPC and subnet ID for subsequent use in **Creating a Cluster**.

Step 5 On the **Virtual Private Cloud** page, choose **Access Control** > **Security Groups** in the navigation tree on the left, and click **Create Security Group** to create a security group.

Step 6 Obtain the security group ID for subsequent use in **Creating a Cluster**.

----End

8.4 Obtaining a Project ID

Obtaining a Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the specified criteria.

The API used to obtain a project ID is **GET https://{*Endpoint*}/v3/projects/**. {*Endpoint*} indicates the IAM endpoint and can be obtained from **Regions and Endpoints**. For details about API authentication, see **Authentication**.

The following is an example response. The value of **id** is the project ID.

```
{
    "projects": [
        {
            "domain_id": "65382450e8f64ac0870cd180d14e684b",
            "is_domain": false,
            "parent_id": "65382450e8f64ac0870cd180d14e684b",
            "name": "ru-moscow-1a",
            "description": "",
            "links": {
                "next": null,
                "previous": null,
                "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
            },
            "id": "a4a5d4098fb4474fa22cd05f897d6b99",
            "enabled": true
        }
    ],
    "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects"
    }
}
```

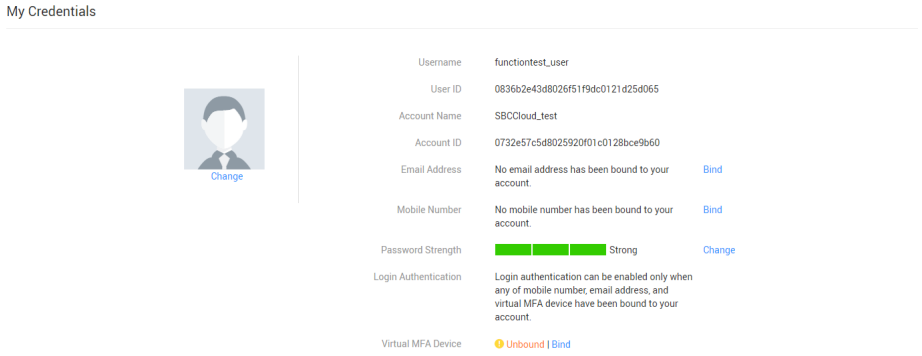
Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credential** from the drop-down list.

   On the **My Credential** page, view project IDs in the project list.

Figure 8-1 Viewing project IDs



8.5 Obtaining an Account ID

An account ID is required for some URLs when an API is called. To obtain the account ID, perform the following steps:

1. Log in to the management console.
2. Click the username and select **My Credential** from the drop-down list.
   On the **My Credential** page, view the **Account ID**.

