Cloud Data Warehouse for PostgreSQL

Operation Guide

Product Documentation

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Operation Guide

Managing Cluster

Creating Cluster

Last updated：2024-02-02 15:17:07

Overview

This document describes how to create a cluster in the CDWPG console.

Directions

1. Log in to the CDWPG console and select the cluster list.
2. Click **Create** to enter the purchase page and configure the parameters as prompted.

Billing Mode: Monthly subscription and pay-as-you-go billing are supported.

Region: Guangzhou, Shanghai, Beijing, and Singapore regions are supported currently.

AZ: Different AZs are not interconnected.

Network: Configure the VPC and subnet of the CDWPG cluster as instructed in VPC.For access from other subnets or public network, see Applying for Public IP and Creating IP Allowlist.

Cluster Name: Used to distinguish between clusters.

Cluster Version: Versions supported by the cluster.

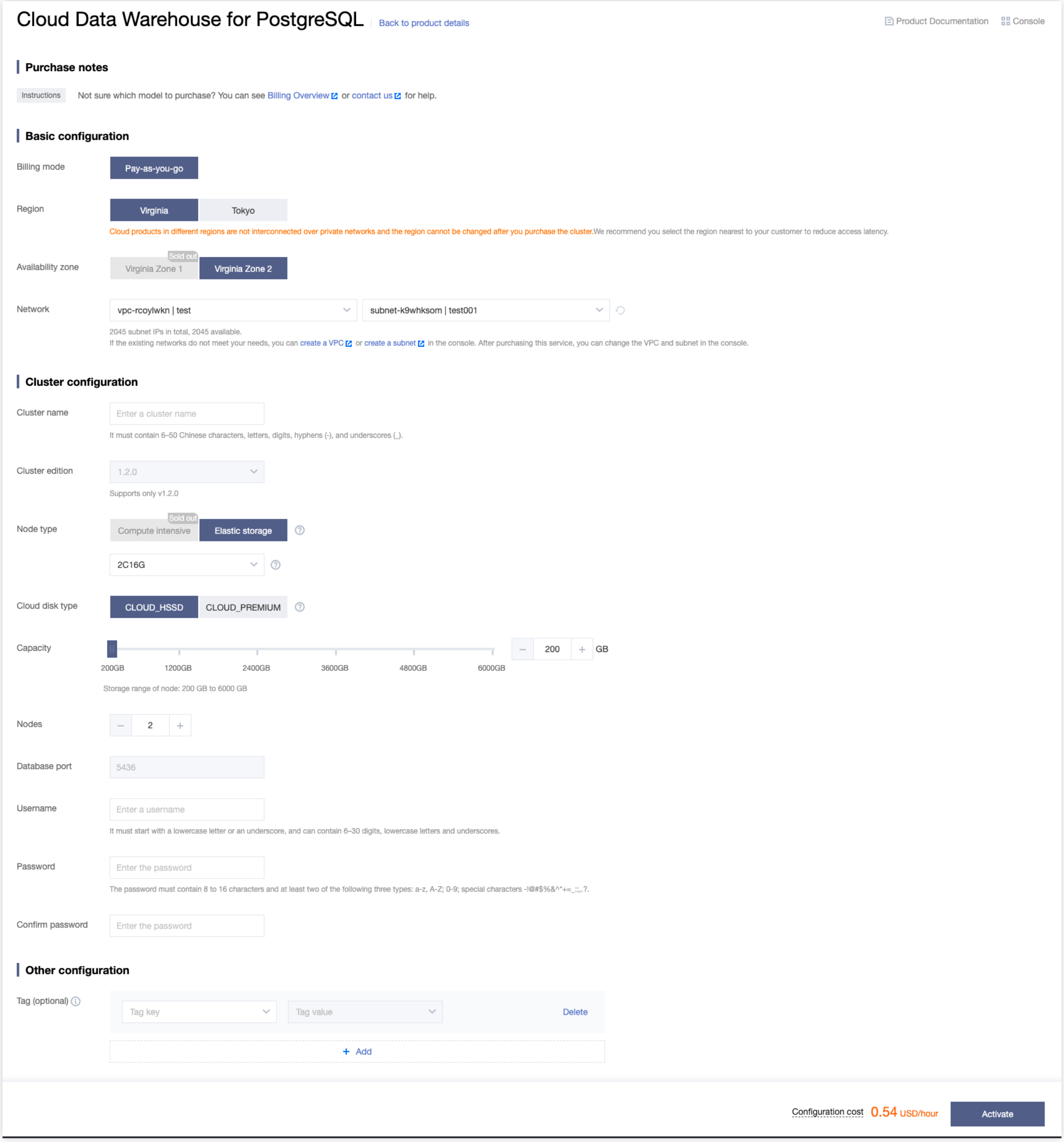
Node Type: nc.large, nc.4xlarge, nc2.large, nc2.4xlarge, ns.large, and ns2.large are currently available. For more information, see Billing Overview.

Number of Nodes: 2–50 nodes are supported. Generally, two or more (50 at maximum) nodes are required.

Database Port: 5436, which cannot be modified currently.

Username: Username to log in to the CDWPG cluster, which is the cluster admin account and cannot be changed after creation.

Password: Password to log in to the CDWPG cluster, which can be changed in the console.



Cloud Data Warehouse for PostgreSQL | Back to product details

Purchase notes

Instructions Not sure which model to purchase? You can see Billing Overview or contact us for help.

Basic configuration

Billing mode: Pay-as-you-go

Region: Virginia | Tokyo

Cloud products in different regions are not interconnected over private networks and the region cannot be changed after you purchase the cluster.We recommend you select the region nearest to your customer to reduce access latency.

Availability zone: Virginia Zone 1 (Sold out) | Virginia Zone 2

Network: vpc-rcoylwkn | test; subnet-k9whksom | test001

2045 subnet IPs in total, 2045 available.
If the existing networks do not meet your needs, you can create a VPC or create a subnet in the console. After purchasing this service, you can change the VPC and subnet in the console.

Cluster configuration

Cluster name: Enter a cluster name

It must contain 6–50 Chinese characters, letters, digits, hyphens (-), and underscores (_).

Cluster edition: 1.2.0

Supports only v1.2.0

Node type: Compute intensive (Sold out) | Elastic storage

2C16G

Cloud disk type: CLOUD_HSSD | CLOUD_PREMIUM

Capacity: 200 GB

Storage range of node: 200 GB to 6000 GB

Nodes: 2

Database port: 5436

Username: Enter a username

It must start with a lowercase letter or an underscore, and can contain 6–30 digits, lowercase letters and underscores.

Password: Enter the password

The password must contain 8 to 16 characters and at least two of the following three types: a-z, A-Z; 0-9; special characters -!@#$%&^*+=_:;,.?.

Confirm password: Enter the password

Other configuration

Tag (optional): Tag key | Tag value

Configuration cost 0.54 USD/hour

3. After completing the configuration, click **Buy Now** or **Activate** to create the cluster.

4. Return to the cluster list and use the cluster after its status changes to **Running**.

Cluster Information

Last updated：2024-02-02 15:17:07

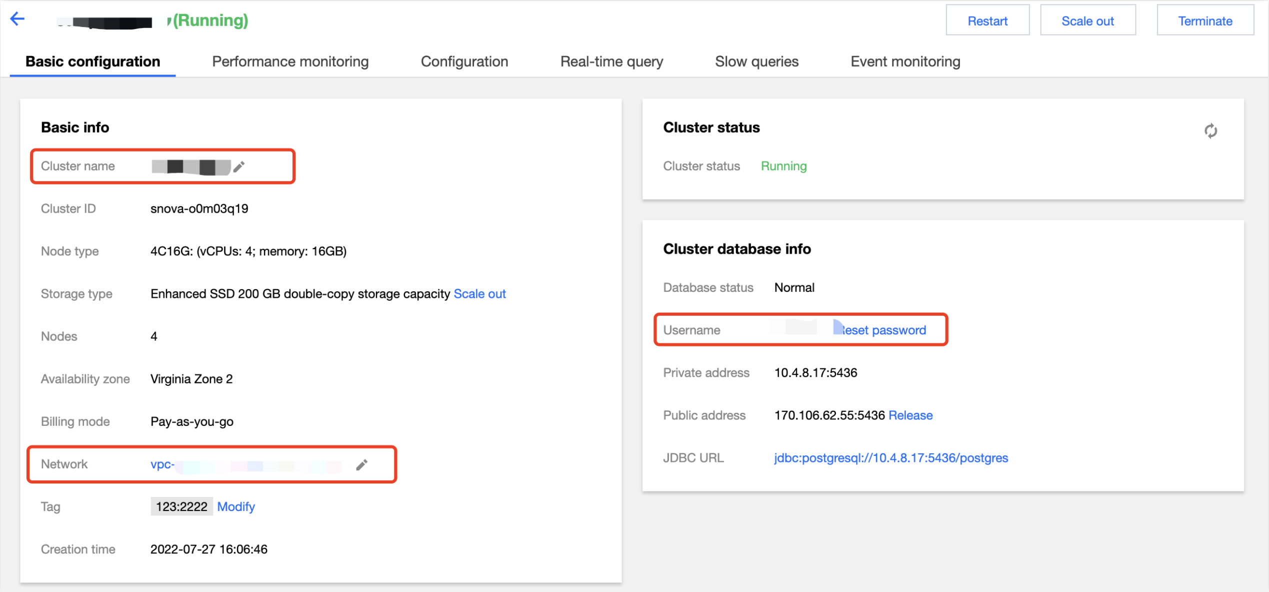
This document describes how to modify the basic information and billing mode of a CDWPG cluster in the console, as well as view its performance monitoring data, real-time queries, historical queries, and event monitoring data.

Basic Information

On the cluster list page, enter the **Basic Configuration** tab of a cluster, where you can set the cluster name, network, and admin user/password.

Note:

Changing the network address will change the cluster access connection, so you need to change the call address simultaneously.



[redacted](Running)

Basic info

Cluster name	[redacted]
Cluster ID	snova-o0m03q19
Node type	4C16G: (vCPUs: 4; memory: 16GB)
Storage type	Enhanced SSD 200 GB double-copy storage capacity Scale out
Nodes	4
Availability zone	Virginia Zone 2
Billing mode	Pay-as-you-go
Network	vpc-[redacted]
Tag	123:2222 Modify
Creation time	2022-07-27 16:06:46

Cluster status

Cluster status	Running

Cluster database info

Database status	Normal
Username	[redacted] Reset password
Private address	10.4.8.17:5436
Public address	170.106.62.55:5436 Release
JDBC URL	jdbc:postgresql://10.4.8.17:5436/postgres

Billing Information

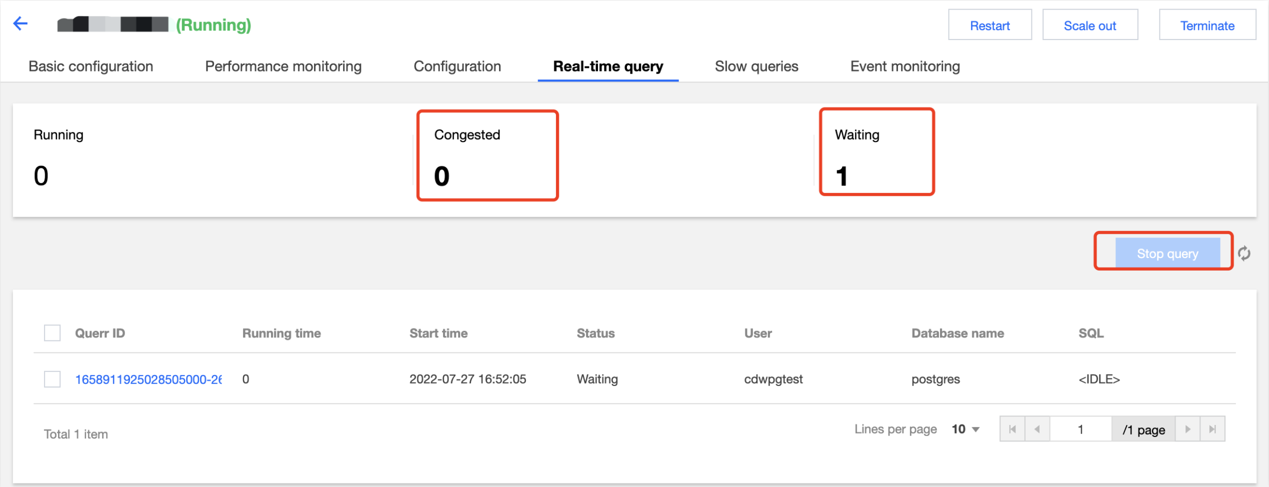
On the **Basic Configuration** tab, you can switch from pay-as-you-go to monthly subscription billing mode for the cluster and enable **Auto-Renewal** as needed.

Performance Monitoring

On the **Performance Monitoring** tab, you can view various the metrics in node and cluster dimensions.

Real-Time Query

On the **Real-Time Query** tab, you can view ongoing queries, including running, blocked, and waiting queries.



[redacted] (Running)

Running 0

Congested 0

Waiting 1

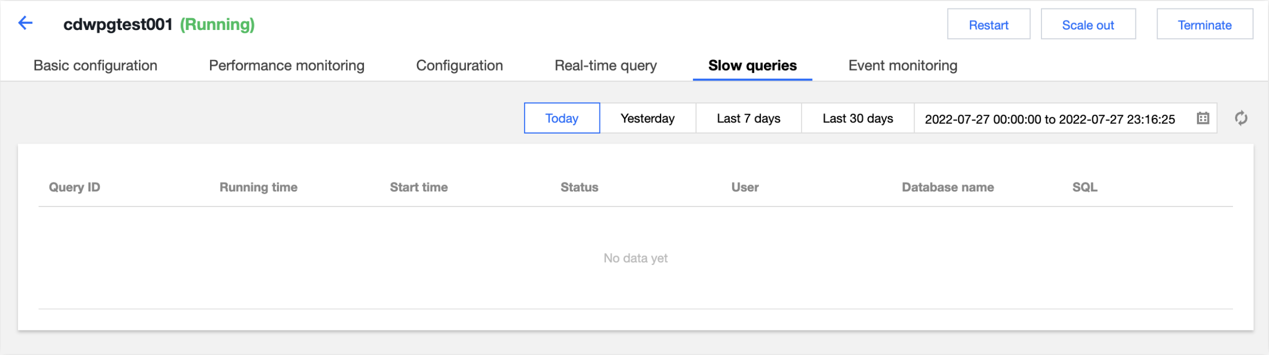
□	Querr ID	Running time	Start time	Status	User	Database name	SQL
□	1658911925028505000-2(	0	2022-07-27 16:52:05	Waiting	cdwpgtest	postgres	<IDLE>

Total 1 item Lines per page 10 ▾ 1 /1 page

You can click **Terminate Query** to terminate a query in the list.

Historical Query

On the **Historical Query** tab, you can view queries that have been completed or terminated.



cdwpgtest001 (Running)

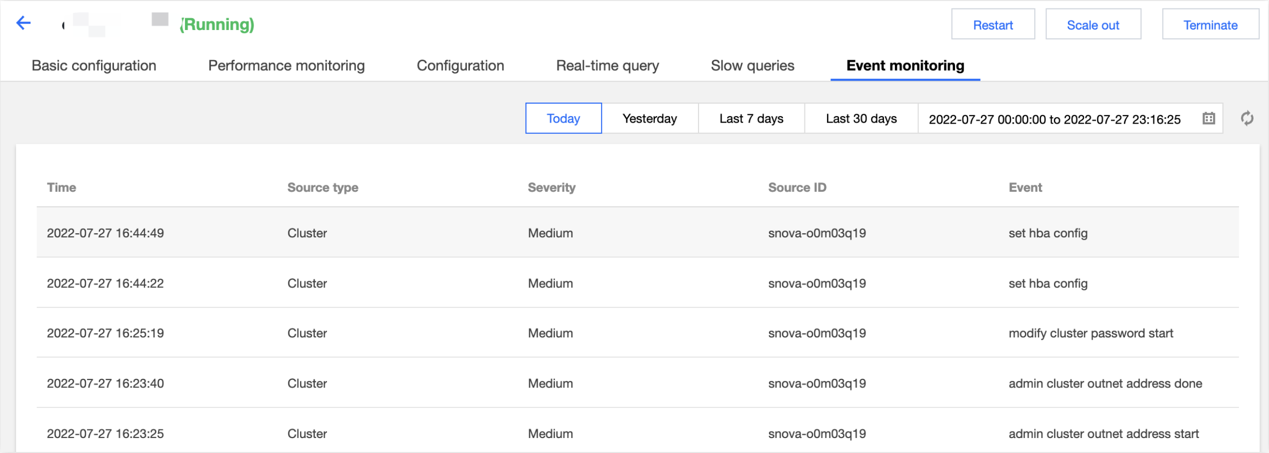
2022-07-27 00:00:00 to 2022-07-27 23:16:25

Query ID	Running time	Start time	Status	User	Database name	SQL

No data yet

Event Monitoring

On the **Event Monitoring** tab, you can query sensitive operation events in the cluster.



[redacted] (Running)

2022-07-27 00:00:00 to 2022-07-27 23:16:25

Time	Source type	Severity	Source ID	Event
2022-07-27 16:44:49	Cluster	Medium	snova-o0m03q19	set hba config
2022-07-27 16:44:22	Cluster	Medium	snova-o0m03q19	set hba config
2022-07-27 16:25:19	Cluster	Medium	snova-o0m03q19	modify cluster password start
2022-07-27 16:23:40	Cluster	Medium	snova-o0m03q19	admin cluster outnet address done
2022-07-27 16:23:25	Cluster	Medium	snova-o0m03q19	admin cluster outnet address start

Scaling Cluster

Last updated：2024-02-02 15:17:07

Overview

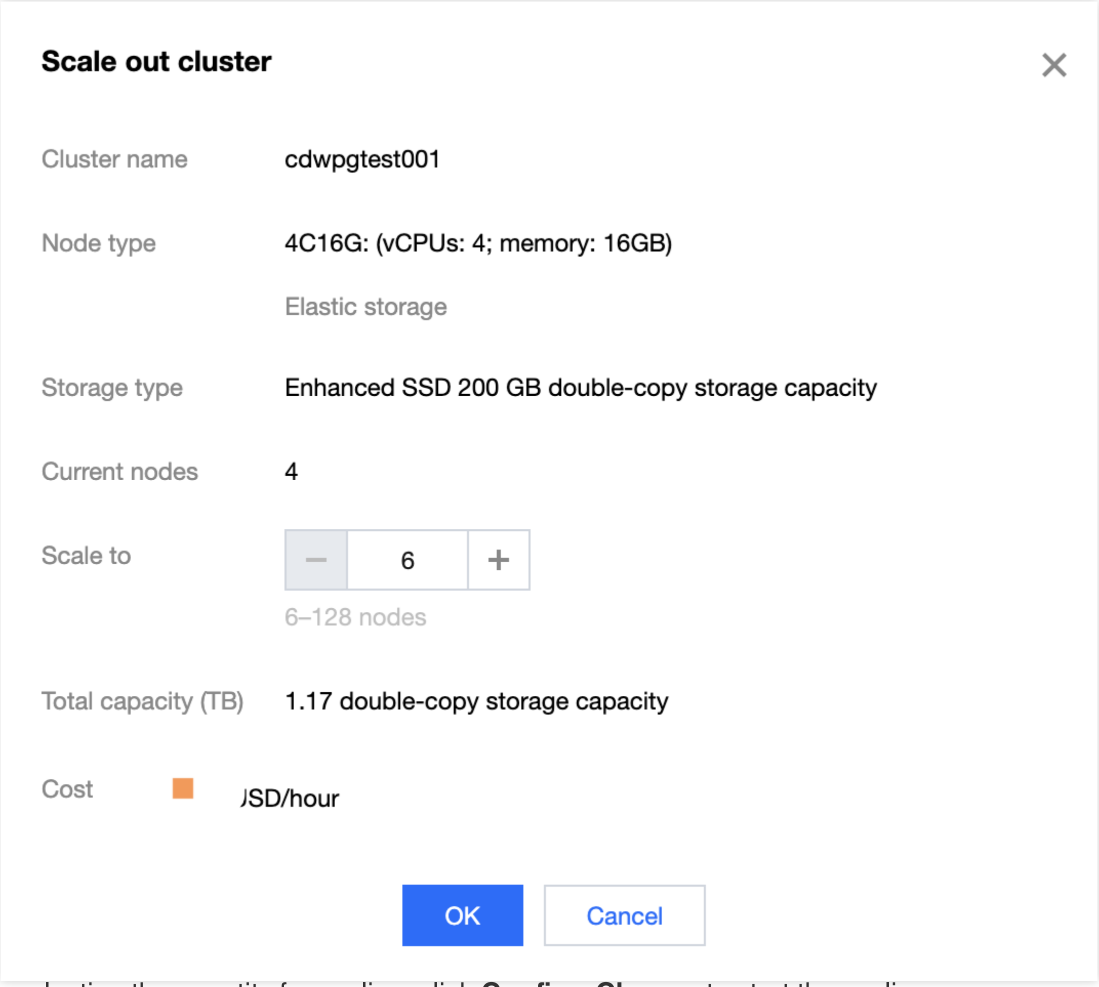
When computing or storage resources of a cluster reach a bottleneck, you can increase system resources by scaling.

Directions

1. Click **Manage** in the **Operations** column in the cluster list to enter the cluster details page.
2. Click **Scale** in the top-right corner to enter the scaling page and select the scaling information as needed.

Note:

The number of nodes entered is that after scaling, which must be greater than the current quantity by two or more. The value of the total capacity changes as the number of nodes changes.



Scale out cluster

Cluster name: cdwpgtest001

Node type: 4C16G: (vCPUs: 4; memory: 16GB)

Elastic storage

Storage type: Enhanced SSD 200 GB double-copy storage capacity

Current nodes: 4

Scale to: 6

6–128 nodes

Total capacity (TB): 1.17 double-copy storage capacity

Cost: [redacted]JSD/hour

3. After selecting the quantity for scaling, click **Confirm Change** to start the scaling process.

4. Return to the cluster list, where the cluster during scaling is in the **Scaling** status. The time it takes to scale is subject to the existing data volume in the cluster. After successful scaling, the cluster will return to the **Running** status.

Managing IP Allowlist and Blocklist

Last updated：2024-02-02 15:17:07

Overview

If you want to access CDWPG clusters through CVM instances in other subnets or servers on the public network other than the CVM instances in the specified subnets, you need to add their IPs to the allowlist.
An IP range can be specified when an IP is added to the allowlist. Add one or more IPs in the IP range to the blocklist if you want to block the access to CDWPG clusters from them.

Note:

The blocklist takes precedence over the allowlist.

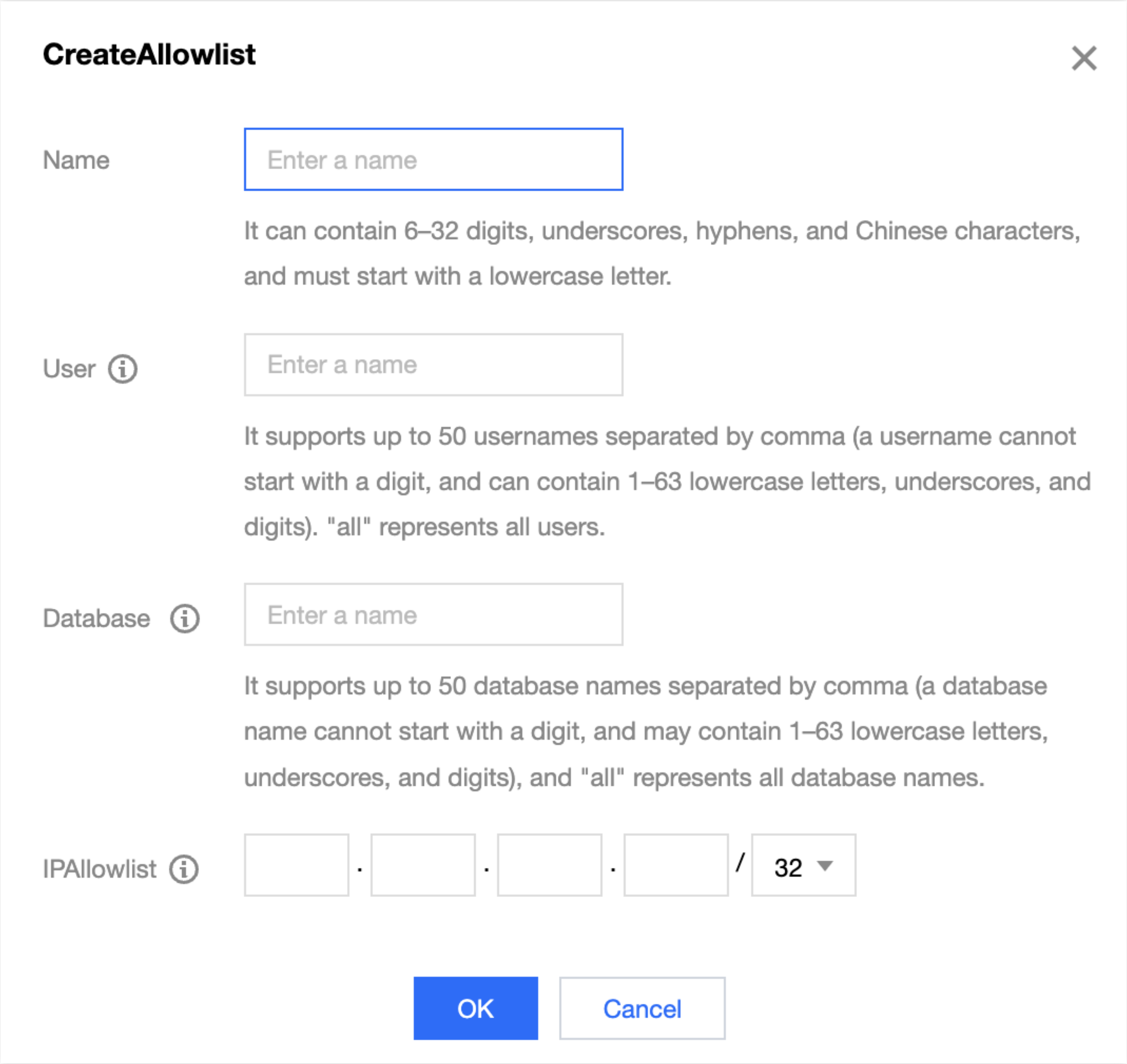
Prerequisites

Before using the IP allowlist, make sure that you have applied for a public IP for a CDWPG cluster; otherwise, you will not be able to access the cluster from servers other than CVM instances in the specified subnets even if their addresses are added to the allowlist.

Directions

Managing IP allowlist

1. Click **Manage** in the cluster list to enter the cluster details page.
2. Select **Configure** > **Access Allowlist**.
3. Click **Create Allowlist** and enter the IP, username, database name, and CIDR block as required.



CreateAllowlist

Name

Enter a name

It can contain 6–32 digits, underscores, hyphens, and Chinese characters, and must start with a lowercase letter.

User

Enter a name

It supports up to 50 usernames separated by comma (a username cannot start with a digit, and can contain 1–63 lowercase letters, underscores, and digits). "all" represents all users.

Database

Enter a name

It supports up to 50 database names separated by comma (a database name cannot start with a digit, and may contain 1–63 lowercase letters, underscores, and digits), and "all" represents all database names.

IPAllowlist . . . / 32

4. Click **OK** to allow the IP range specified by the CIDR block. Then, you can access the CDWPG cluster from a server at the IP or IP range.

Note:

When an IP address in the allowlist is no longer needed, select it in the allowlist and click **Delete** to remove it from the allowlist.

Managing IP blocklist

1. Select **Configure** > **Access Blocklist**.
2. Click **Create Blocklist** and enter the IP, username, database name, and CIDR block as required.
3. Click **OK** to block the IP range specified by the CIDR block. Then, you cannot access the CDWPG cluster from a server at the IP or IP range.

Note:

To unblock an IP or IP range, select it in the blocklist and click **Delete** to remove it from the blocklist.

Applying for Public IP

Last updated：2024-02-02 15:17:07

Overview

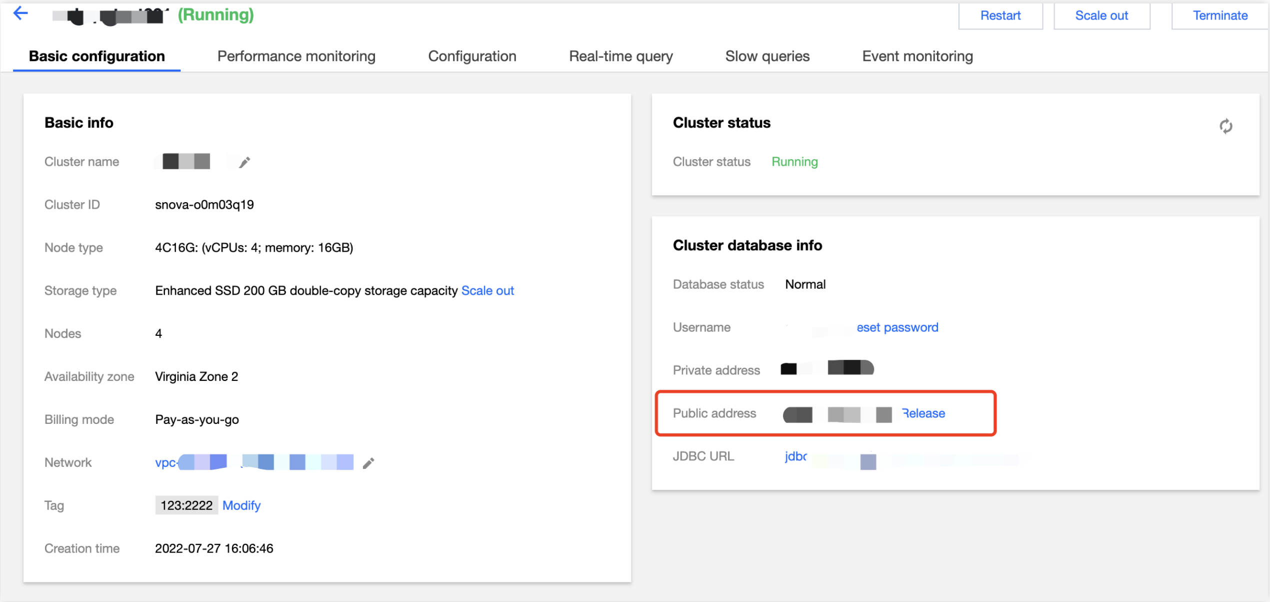
If you want to access a CDWPG cluster through a CVM instance in another subnet or a server on the public network other than the CVM instance in the specified subnet, you need to apply for a public IP for the cluster.

Note:

The CDWPG cluster can be accessed at its public IP. However, after applying for one, you need to add it to the allowlist for actual access as instructed in IP Allowlist.

Directions

1. Click **Manage** in the cluster list to enter the cluster details page.
2. Select the **Basic Configuration** page and click **Apply for Public IP**. A public IP will be generated for access to the CDWPG cluster.



[redacted] (Running)

Basic info

Cluster name [redacted]

Cluster ID snova-o0m03q19

Node type 4C16G: (vCPUs: 4; memory: 16GB)

Storage type Enhanced SSD 200 GB double-copy storage capacity Scale out

Nodes 4

Availability zone Virginia Zone 2

Billing mode Pay-as-you-go

Network vpc-[redacted]

Tag 123:2222 Modify

Creation time 2022-07-27 16:06:46

Cluster status

Cluster status Running

Cluster database info

Database status Normal

Username [redacted] eset password

Private address [redacted]

Public address [redacted] Release

JDBC URL jdbc [redacted]

Note:

When the public IP is no longer needed, click **Release Public IP** to delete it.

Managing Resource Queue

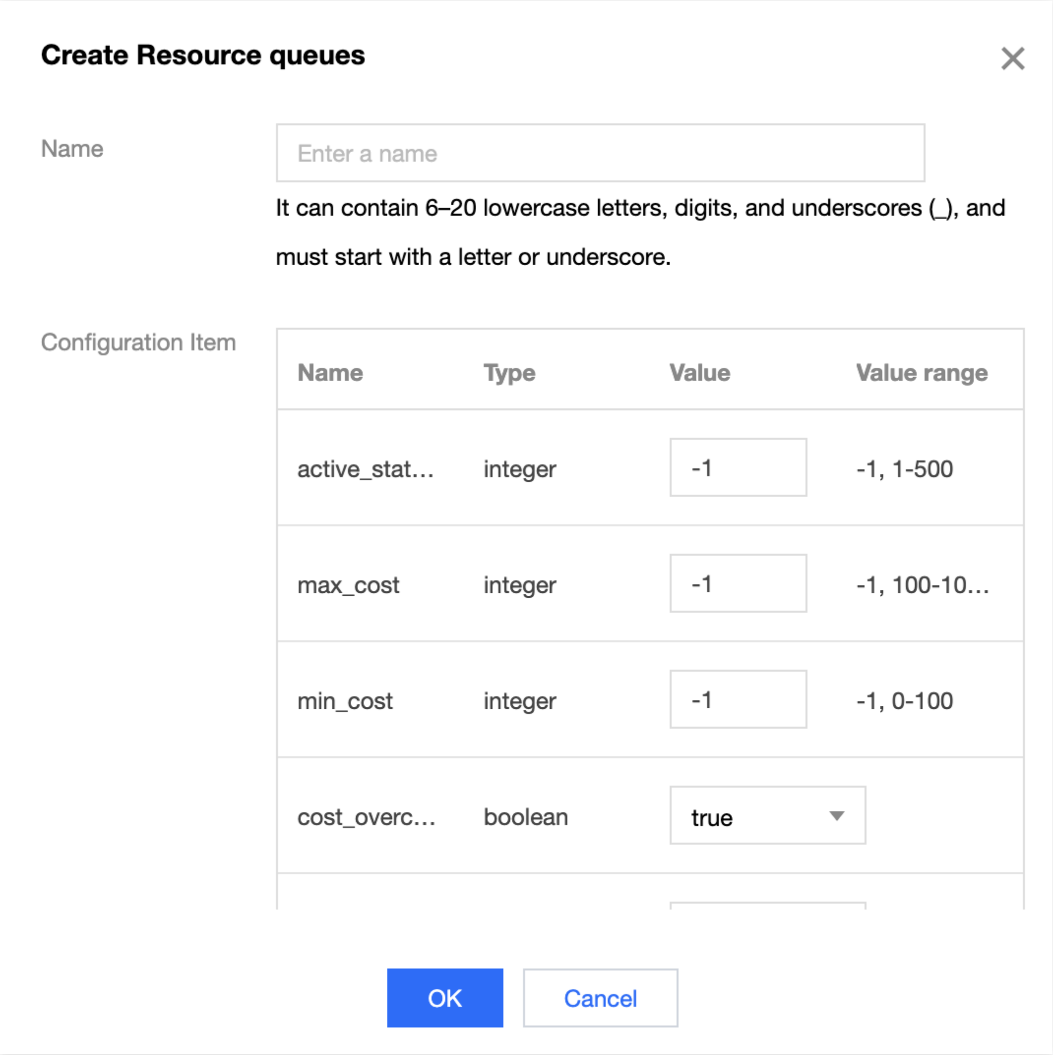
Last updated：2024-02-02 15:17:07

Overview

When you use CDWPG, a single complex query may consume too many resources and affect other queries or computing. You can utilize a resource queue to limit system resource usage by a single user or query statement.

Directions

1. Click **Manage** in the cluster list to enter the cluster details page.
2. Select **Configuration** > **Resource Queue**, click **Create Resource Queue**, and configure parameter values.
3. Click **OK**. After the resource queue is created, you can specify it to limit resources during querying.



Create Resource queues

Name: Enter a name

It can contain 6–20 lowercase letters, digits, and underscores (_), and must start with a letter or underscore.

Configuration Item

Name	Type	Value	Value range
active_stat…	integer	-1	-1, 1-500
max_cost	integer	-1	-1, 100-10…
min_cost	integer	-1	-1, 0-100
cost_overc…	boolean	true	

Terminating Cluster

Last updated：2024-02-02 15:17:07

Overview

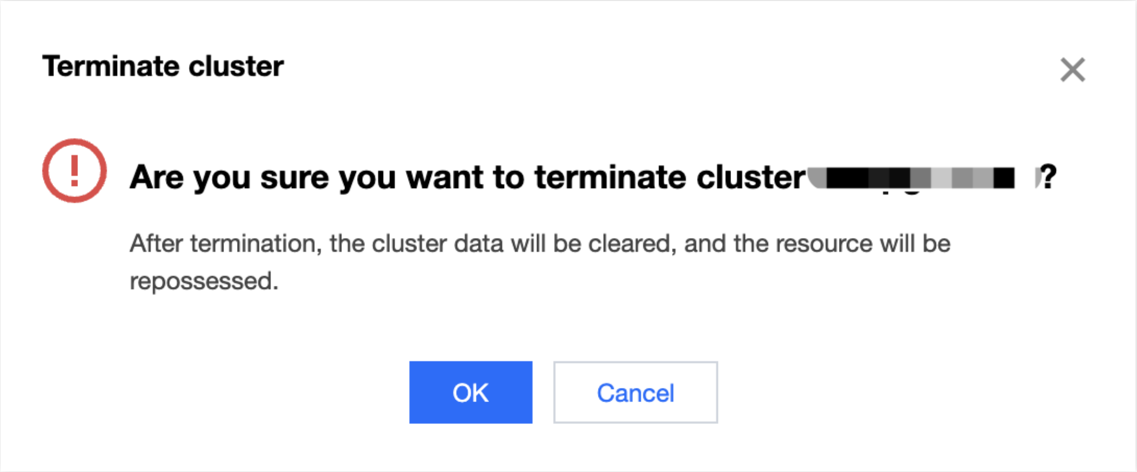
When you no longer need a pay-as-you-go cluster, you can delete it in the CDWPG console.

Note:

Proceed with caution when deleting a cluster, as all the data will be deleted.

Directions

1. Click **Manage** in the cluster list to enter the cluster details page.
2. Click **Terminate** in the top-right corner to enter the termination page.



Terminate cluster

Are you sure you want to terminate cluster [redacted]?

After termination, the cluster data will be cleared, and the resource will be repossessed.

3. Click **Confirm**, and the pay-as-you-go cluster will enter the **Deleting** status.

Note:

Pay-as-you-go and monthly subscribed clusters are terminated on different pages.

A pay-as-you-go cluster is terminated after confirmation. Once confirmed, the termination will start, and data cannot be recovered.

You will be redirected to the order refund page to terminate a monthly subscribed cluster. After the refund is confirmed, the cluster will be isolated and then retained for seven days, during which you can renew it to use it again. Note that you can only renew it once. If the cluster is not renewed, data will be cleared after seven days.

Accessing Data Warehouse

Connecting to Database

Last updated：2024-02-02 15:17:07

By default, only CVM instances in the same VPC subnet can access a CDWPG cluster. If you need to access it over the public network, apply for a public IP.

After creating a cluster, you need to use the database client to connect to the database before you can use the database services. Connect to the database with the client tool psql as instructed below.

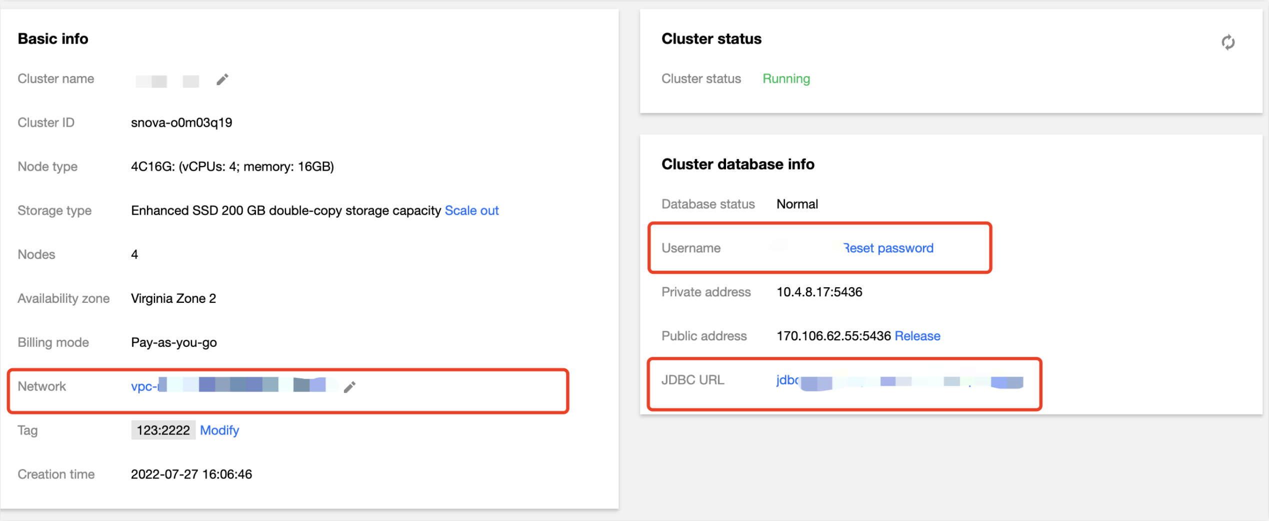
1. Get the cluster access address: Connect to the database through the IP and port number in the JDBC URL of the cluster.
2. Connect to the cluster database: Install the client and connect to the cluster database.

Prerequisites

1. You have obtained the password of the database admin account for the CDWPG cluster. The password is the one set when the cluster is created.
2. You have obtained the IP, port number, VPC, and subnet information to access the created CDWPG cluster.

Getting Cluster Access Address and Local Network Information

Select the corresponding cluster with the details as shown below. Get the information of the `vpc-aejsd98p` VPC and `subnet-83knqldq` subnet. The IP to access CDWPG is `10.0.6.10` , the port number is `5432` , and the login account is `lambuser` .



Basic info

Cluster name [redacted]

Cluster ID snova-o0m03q19

Node type 4C16G: (vCPUs: 4; memory: 16GB)

Storage type Enhanced SSD 200 GB double-copy storage capacity Scale out

Nodes 4

Availability zone Virginia Zone 2

Billing mode Pay-as-you-go

Network vpc-[redacted]

Tag 123:2222 Modify

Creation time 2022-07-27 16:06:46

Cluster status

Cluster status Running

Cluster database info

Database status Normal

Username [redacted] Reset password

Private address 10.4.8.17:5436

Public address 170.106.62.55:5436 Release

JDBC URL jdbc[redacted]

Connecting to Cluster Database on Command Line

Select a CVM instance in the obtained `vpc-aejsd98p` VPC and `subnet-83knqldq` subnet or purchase one if there is none. Log in to the instance and run the following command to install the PostgreSQL client.

```
yum install -y postgresql.x86_64
```

Run the following SQL command and enter the password set during cluster creation to log in.

```
psql -h10.0.6.10  -p5432  -dpostgres  -Ulambuser
```

Connecting to Database with JDBC

Get the JDBC officially provided by PostgresSQL here.

Or, add the following configuration to the `pom.xml` file:

```
<dependencies>
        <dependency>
            <groupId>org.postgresql</groupId>
            <artifactId>postgresql</artifactId>
            <version>42.2.2</version>
        </dependency>
</dependencies>
```

Sample code

```
package com.qcloud.snova_conn;

import java.io.InputStream;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.Statement;
import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.List;
import java.util.Properties;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;
import com.yammer.metrics.core.Meter;

public class SnovaConn {
    /*
     * args: vip vport user pwd
     */
    public static void main(String[] args) throws ClassNotFoundException, SQLExcept

        if (args.length < 4){
                 System.out.println("args err");
                 return;
        }

        String vip =  args[0];
        String vport =  args[1];
        String userName =  args[2];
        String userPwd =  args[3];

        System.out.printf("vip:%s, vport:%s, userName:%s, userPwd:%s\\n",vip, vport
        String jdbcUrl = "jdbc:postgresql://" + vip+":"+vport+"/maxluo";
        System.out.printf("jdbcUrl:%s \\n",jdbcUrl);

        Class.forName("org.postgresql.Driver");
        Connection snova = DriverManager.getConnection(jdbcUrl,userName,userPwd);
        Statement st = snova.createStatement();
        ResultSet rs = st.executeQuery("select * from test;");
        while (rs.next()) {
                System.out.print(rs.getString(1));
                System.out.print("\\n");
        }
        rs.close();
        st.close();
    }
}
```

`pom.xml` configuration

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/200
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd
<modelVersion>4.0.0</modelVersion>

<groupId>com.qcloud</groupId>
<artifactId>snova-conn</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>jar</packaging>

<name>snova-conn</name>
<url>http://maven.apache.org</url>
```

```
<properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
</properties>

<dependencies>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>5.1.40</version>
    </dependency>
    <dependency>
        <groupId>org.postgresql</groupId>
        <artifactId>postgresql</artifactId>
        <version>42.2.2</version>
    </dependency>
    <dependency>
        <groupId>com.microsoft.sqlserver</groupId>
        <artifactId>mssql-jdbc</artifactId>
        <version>6.4.0.jre8</version>
    </dependency>
    <dependency>
        <groupId>com.yammer.metrics</groupId>
        <artifactId>metrics-core</artifactId>
        <version>2.2.0</version>
    </dependency>
    <dependency>
        <groupId>ch.qos.logback</groupId>
        <artifactId>logback-classic</artifactId>
        <version>1.1.9</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <artifactId>maven-assembly-plugin</artifactId>
                <configuration>
                <descriptorRefs>
                <descriptorRef>jar-with-dependencies</descriptorRef>
                </descriptorRefs>
                </configuration>
            <executions>
                <execution>
                <id>make-assembly</id>
                <phase>package</phase>
                <goals>
                <goal>single</goal>
                </goals>
                </execution>
            </executions>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-jar-plugin</artifactId>
            <configuration>
                <excludes>
                    <exclude>*.properties</exclude>
                    <exclude>*.xml</exclude>
                    <exclude>*.json</exclude>
                    <exclude>*.sh</exclude>
                </excludes>
            </configuration>
        </plugin>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-dependency-plugin</artifactId>
            <executions>
                <execution>
                    <id>copy-dependencies</id>
                    <phase>package</phase>
                    <goals>
                        <goal>copy-dependencies</goal>
                    </goals>
                    <configuration>
                        <type>jar</type>
                        <includeTypes>jar</includeTypes>
                        <outputDirectory>
                            ${project.build.directory}/lib
                        </outputDirectory>
                    </configuration>
                </execution>
            </executions>
        </plugin>
    </plugins>
</build>
</project>
```

Then, use Maven to package the `jar` file and upload the `jar` package to the CVM instance (any instance in the VPC subnet where the CDWPG cluster resides).

Run the following command to install JDK.

```
yum install java
```

Run the following command:

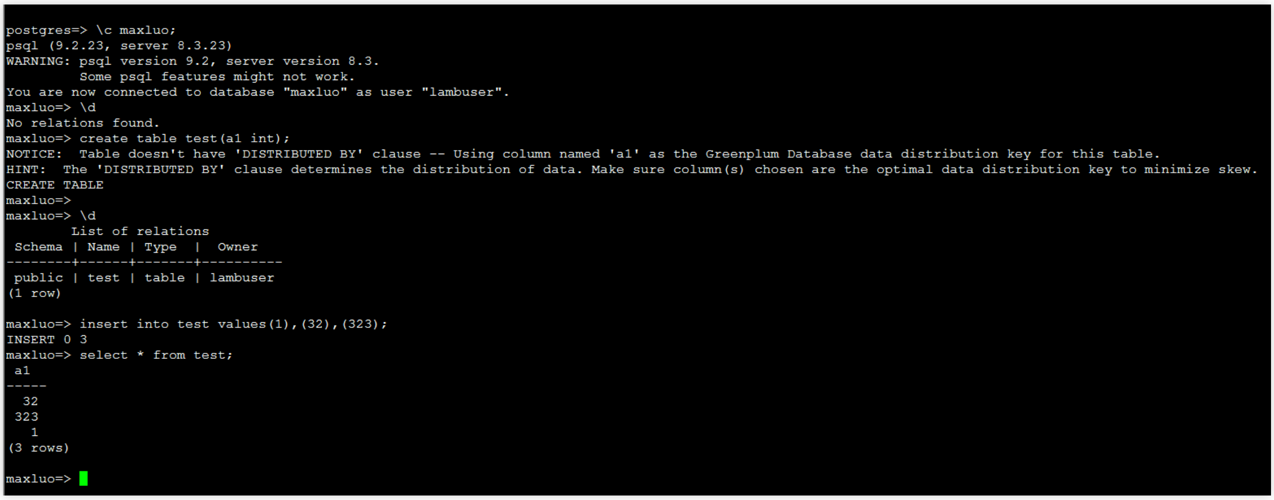
```
java -cp  snova-conn-0.0.1-SNAPSHOT-jar-with-dependencies.jar  com.qcloud.snova_con
```

Note:

The VIP and port number are the address to access the CDWPG cluster, and the username and password are those entered during cluster creation as detailed above.

Create a database and data table on the command line and insert a certain amount of data.

The query result is as shown below, where you can read the data in the `test` table in the `maxluo` database created previously.



```
postgres=> \c maxluo;
psql (9.2.23, server 8.3.23)
WARNING: psql version 9.2, server version 8.3.
         Some psql features might not work.
You are now connected to database "maxluo" as user "lambuser".
maxluo=> \d
No relations found.
maxluo=> create table test(a1 int);
NOTICE:  Table doesn't have 'DISTRIBUTED BY' clause -- Using column named 'a1' as the Greenplum Database data distribution key for this table.
HINT:  The 'DISTRIBUTED BY' clause determines the distribution of data. Make sure column(s) chosen are the optimal data distribution key to minimize skew.
CREATE TABLE
maxluo=>
maxluo=> \d
        List of relations
 Schema | Name | Type  |  Owner
--------+------+-------+----------
 public | test | table | lambuser
(1 row)

maxluo=> insert into test values(1),(32),(323);
INSERT 0 3
maxluo=> select * from test;
 a1
-----
  32
 323
   1
(3 rows)

maxluo=> 
```

Managing User Permission

Last updated：2024-02-02 15:17:07

Role Overview

Roles in CDWPG are used to manage access to databases. A role can be either a user (a specific database user) or a group (a group of database users). Roles can own database objects (such as tables and views) and assign access to these objects to other roles.

When creating a cluster, you will be prompted to set an initial username and password. The initial user will be the "admin user" with permissions to create users, create databases, and log in. After the cluster is created, you can connect to the database as the "admin user". Generally, the admin user has the maximum privileges, which means that this account should be used by as few users as possible. To this end, you can use the admin user to create other users and grant required permissions to them. For authorization directions, see User group and Object permission management. You can also create databases and other objects as instructed in Defining Database. To log in to a database, see Connecting to Database.

Creating User

A role can be either a user or a group. Usually, a user role (referred to as "user" hereinafter) has the permission to log in to CDWPG databases and initialize sessions. Therefore, when creating a user, you must grant them the `LOGIN` permission; for example:

```
CREATE role jsmith with LOGIN;
```

With the above operation, a user with the `LOGIN` permission is created, who can connect to databases. In addition to `LOGIN` , CDWPG also has the following permissions to manage user access, which can be granted during role creation with the `CREATE ROLE` statement.

Permission Value	Purpose	Default Value
SUPERUSER &Iota; NOSUPERUSER	Superuser permission. Only superusers can create other superusers	NOSUPERUSER

CREATEDB &Iota; NOCREATEDB	Creates databases	NOCREATEDB
CREATEROLE &Iota; NOCREATEROLE	Creates and manages roles	NOCREATEROLE
INHERIT &Iota; NOINHERIT	Determines the permissions a user inherits from the group to which the user belongs	INHERIT
LOGIN &Iota; NOLOGIN	Connects to databases, which is granted to users but not groups	NOLOGIN
CONNECTION LIMIT	Limits the number of concurrent connections to a database. -1 means no limit	-1
CREATEEXTTABLE &Iota; NOCREATEEXTTABLE	Creates external tables	NOCREATEEXTTABLE
PASSWORD	Sets the password during user creation	None
VALID UNTIL 'timestamp'	Password expiration time	None
RESOURCE QUEUE 'name'	The name of the resource queue to which the created query is scheduled after a user establishes a connection	pg_default

In addition to granting permissions to users when creating them, you can also grant permissions afterwards by using the `ALTER ROLE` syntax as shown below:

```
ALTER role jsmith with CREATEROLE;
```

User group

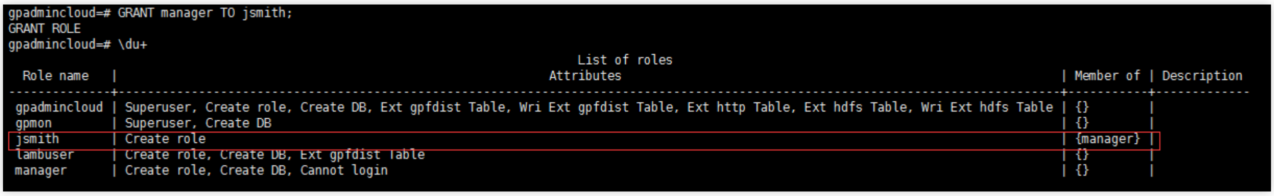
A group (i.e., user group) is a special role that is not granted the `LOGIN` permission but a combination of permissions that are frequently used together. In this way, permissions can be granted to or revoked from a user as a whole.

You can create a group that is granted a combination of permissions by using the following statement.

```
Create role, Create DB, Cannot login;
```

You can also easily add users to or remove them from the group with the `GRANT TO` or `REVOKE FROM` statement respectively. Users added to the group will inherit the group's permissions.

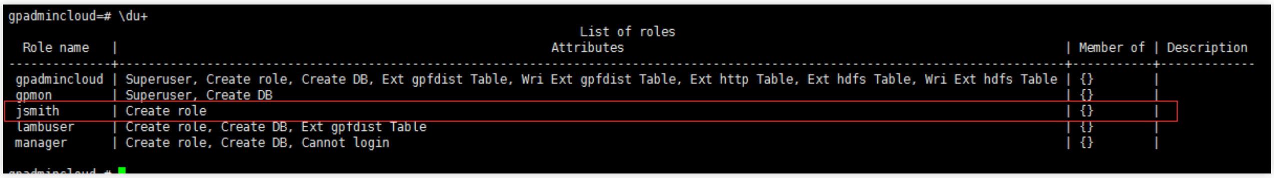
Sample `GRANT TO` statement:



```
gpadmincloud=# GRANT manager TO jsmith;
GRANT ROLE
gpadmincloud=# \du+
                                                                  List of roles
  Role name   |                                                  Attributes                                                  | Member of | Description
--------------+--------------------------------------------------------------------------------------------------------------+-----------+------------
 gpadmincloud | Superuser, Create role, Create DB, Ext gpfdist Table, Wri Ext gpfdist Table, Ext http Table, Ext hdfs Table, Wri Ext hdfs Table | {}        |
 gpmon        | Superuser, Create DB                                                                                         | {}        |
 jsmith       | Create role                                                                                                  | {manager} |
 lambuser     | Create role, Create DB, Ext gpfdist Table                                                                    | {}        |
 manager      | Create role, Create DB, Cannot login                                                                         | {}        |
```

. The `jsmith` user belongs to the `manager` group.

Sample `REVOKE FROM` statement:



```
gpadmincloud=# \du+
                                                                  List of roles
  Role name   |                                                  Attributes                                                  | Member of | Description
--------------+--------------------------------------------------------------------------------------------------------------+-----------+------------
 gpadmincloud | Superuser, Create role, Create DB, Ext gpfdist Table, Wri Ext gpfdist Table, Ext http Table, Ext hdfs Table, Wri Ext hdfs Table | {}        |
 gpmon        | Superuser, Create DB                                                                                         | {}        |
 jsmith       | Create role                                                                                                  | {}        |
 lambuser     | Create role, Create DB, Ext gpfdist Table                                                                    | {}        |
 manager      | Create role, Create DB, Cannot login                                                                         | {}        |

gpadmincloud=#
```

. The `jsmith` user no longer belongs to the `manager` group.

Object Permission Management

When an object (database, table, schema, function, etc.) is created, it must belong to an owner, which is usually the user who runs the object creation statement. Initially, only the owner has all the permissions to manipulate the object; for example:

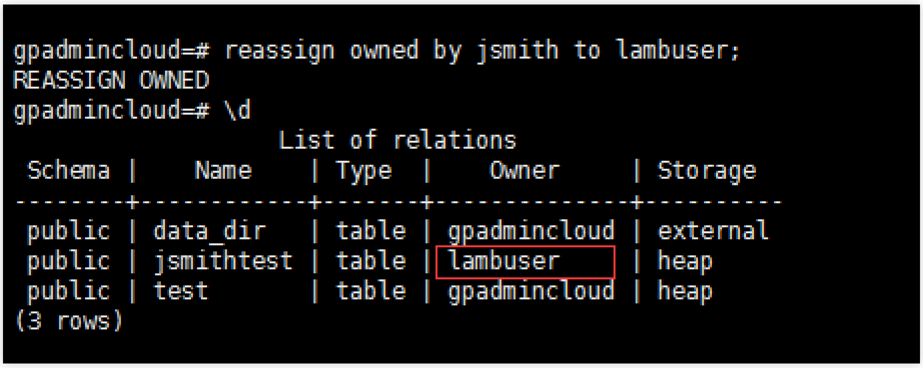
```
GRANT INSERT ON test TO  jsmith;
```

You can grant the `INSERT` permission of `test` to the `jsmith` user with the above statement and revoke it with `REVOKE FROM` .

Similarly, you can transfer all the objects owned by a user to another user with the `REASSIGN OWNED` statement as shown below:

```
SET ROLE jsmith;   // Switch to the `jsmith` user.
CREATE TABLE jsmithtest (age int, id int); // Create a table
SET ROLE gpadmincloud;        // Switch back to the superuser
reassign owned by jsmith to lambuser;   // Transfer all the objects owned by `jsmit
```

The objects owned by a superuser cannot be transferred to other users, because some of the objects also belong to the system. Therefore, you need to create a table as a non-superuser.



```
gpadmincloud=# reassign owned by jsmith to lambuser;
REASSIGN OWNED
gpadmincloud=# \d
                    List of relations
 Schema |    Name    | Type  |     Owner     | Storage
--------+------------+-------+---------------+----------
 public | data_dir   | table | gpadmincloud | external
 public | jsmithtest | table | lambuser      | heap
 public | test       | table | gpadmincloud | heap
(3 rows)
```

Complete the transfer of object ownership from `jsmith` to `lambuser`.

Defining Database

Last updated：2024-02-02 15:17:07

Database Creation and Management

You can create database objects in CDWPG.

1. To create a database for your own use, create a user, grant authorization, and log in as instructed in Managing User Permissions. Then, use the `CREATE DATABASE` statement to create the database. Before database creation, make sure that the logged-in user has `CRAETEROLE` permissions. For more information on permissions, see Managing User Permissions. A sample database can be created as follows:

```
CREATE DATABASE test;
```

All databases can be listed via `\\l` .

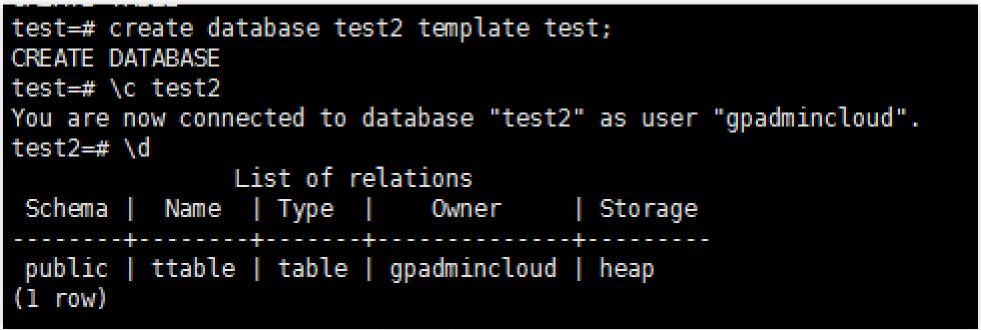
2. A database is usually created based on a template database, and the default template is empty. Any objects in the template will be in the newly created database. You can also specify a template for creation. For example, create a table in the `test` database first by using the following statement.

```
create table ttable (age int, id int);
```

Then, create the `test2` database with `test` as the template.

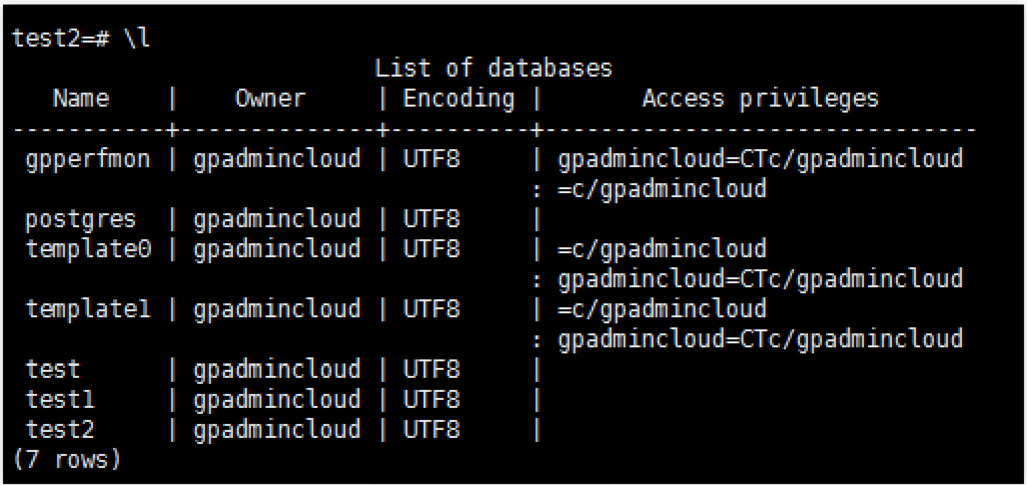
```
CREATE DATABASE test2 TEMPLATE test;
```

Switch to `test2` , and you can see that `ttable` also exists in `test2` . Therefore, avoid creating any objects in `template1` , as those in `template1` will also exist in databases created based on it. You can view all the tables in the selected database with `\\d` .



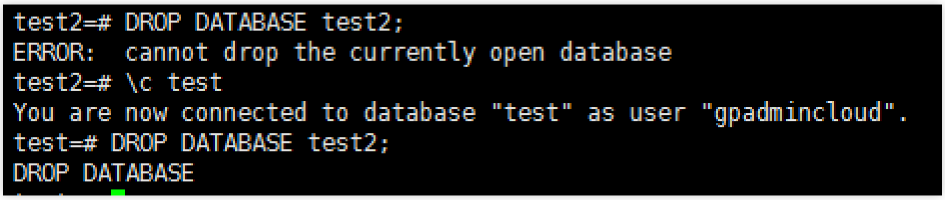
```
test=# create database test2 template test;
CREATE DATABASE
test=# \c test2
You are now connected to database "test2" as user "gpadmincloud".
test2=# \d
                List of relations
 Schema |  Name  | Type  |    Owner     | Storage
--------+--------+-------+--------------+---------
 public | ttable | table | gpadmincloud | heap
(1 row)
```

3. You can list all the databases with `\\l` .



```
test2=# \l
                          List of databases
   Name    |    Owner     | Encoding |       Access privileges
-----------+--------------+----------+-------------------------------
 gpperfmon | gpadmincloud | UTF8     | gpadmincloud=CTc/gpadmincloud
                                     : =c/gpadmincloud
 postgres  | gpadmincloud | UTF8     |
 template0 | gpadmincloud | UTF8     | =c/gpadmincloud
                                     : gpadmincloud=CTc/gpadmincloud
 template1 | gpadmincloud | UTF8     | =c/gpadmincloud
                                     : gpadmincloud=CTc/gpadmincloud
 test      | gpadmincloud | UTF8     |
 test1     | gpadmincloud | UTF8     |
 test2     | gpadmincloud | UTF8     |
(7 rows)
```

4. You can delete the database with `DROP DATABASE` . When you perform the deletion operation, make sure that the logged-in user is a superuser or a general user with the permission of database deletion. Note that the database can be deleted only if the number of its connections is 0; for example:



```
test2=# DROP DATABASE test2;
ERROR:  cannot drop the currently open database
test2=# \c test
You are now connected to database "test" as user "gpadmincloud".
test=# DROP DATABASE test2;
DROP DATABASE
```

You can see that when you select `test2` , the number of connections to `test2` must be greater than or equal to 1. You can only delete `test2` after switching to `test` .

Schema Creation and Management

In CDWPG, a schema is a logical concept for a more detailed division of the database space. Each database has a schema named `public` when it is created. Tables with the same name cannot be created in the same database unless they are in different schemas. The database system identifies a table in the form of `database.schema.table` . In addition, schemas with the same name can be created in different databases.

1. Create a schema.

```
CREATE SCHEMA testschema;
```

2. Specify the schema to create objects.

When creating tables, functions, or other objects, you can add schema prefixes to indicate that they are to be created in different schemas; if no prefixes are added, objects will be created in the `public` schema by default; for example:

```
CREATE TABLE testschema.test;
```

3. Set the priority of the schema.

```
ALTER DATABASE test set search_path to testshcema,public;
```

This statement sets the `testschema` of the `test` database to `public` mode and indicates that `testschema` has the highest priority. If no schema prefixes are added, `testschema` will be matched first.

4. Switch the schema.

```
SET search_path TO public;
```

If the current schema is `testschema`, you can switch to `public` with this statement.

5. Delete the schema.

```
DROP SCHEMA testschema;
```

Table Creation and Management

1. Set table and column constraints.

`CHECK` constraint, which specifies that a data column must satisfy a certain expression, such as:

```
CREATE TABLE products (product_no int, name text, price int CHECK(price > 0));
```

`NOT NULL` constraint, which specifies that a data column cannot be empty, such as:

```
CREATE TABLE products (product_no int NOT NULL, name text NOT NULL, price int CHECK
```

2. Set the data distribution policy.

A distributed database warehouse delivers the best performance when the amount of data stored in each node is the same. If data distribution is unbalanced, nodes with more data will spend more time completing queries, which compromises the overall query performance.

Hash distribution: You can use the `DISTRIBUTED BY` syntax to specify hash distribution when creating a table. This policy combines all the keys specified as hash distribution and determines the result of data distribution by the hash algorithm. The statement is as follows:

```
CREATE TABLE test (id int, age int) DISTRIBUTED BY (id);
```

Random distribution: You can use the `DISTRIBUTED RANDOMLY` syntax to specify random distribution when creating a table. As the name suggests, this policy determines data distribution randomly. The statement is as follows:

```
CREATE TABLE test (id int, age int) DISTRIBUTED RANDOMLY;
```

Here, you must specify one of the data distribution policies for columns with `PRIMARY KEY` or `UNIQUE` . For other columns, the first column will be used as the reference for data distribution by default, and the default data distribution policy will be hash distribution.

View Creation and Management

A view is a logical concept. Unlike a table, it has no actual corresponding data structures on the disk.

1. Create a view.

```
CREATE VIEW testview AS SELECT * FROM ttable where age=28;
```

Create the `testview` view with all rows in `ttable` that satisfy the condition "age=28".

2. Delete the view.

```
DROP VIEW testview;
```

Managing Data

Last updated：2024-02-02 15:17:07

Inserting Data

1. Insert data corresponding to the column name.

```
INSERT INTO products (name, price, product_no) VALUES ('cheese', 99, 1);
```

2. Insert data in the order of the column names defined in the table.

```
INSERT INTO products VALUES(2, 'chesse', 99);
```

3. Insert multiple data records at once.

```
INSERT INTO products VALUES (3, 'a', 1), (4, 'b', 2), (5, 'c', 3);
```

4. Import data through an external table as instructed in Using External Table.

5. Import data from TencentDB via extensions as instructed in Importing External Data.

6. Insert data by using the `COPY` command. You need to log in to the database system, select the database, create the corresponding table, and use `COPY` to insert the data from the specified `filename` into `tablename` with the specified delimiter `,` . The command is as follows:

```
COPY tablename  FROM 'filename'  WITH DELIMITER ',';
```

Updating Data

Update the data in the column corresponding to the row that satisfies the `WHERE` condition to the specified value as shown below:

```
UPDATE products SET price = 10 where product_no = 3;
```

Deleting Data

Delete the row that satisfies the `WHERE` condition as shown below:

```
DELETE FROM products where price = 3;
```

Delete all data in the table as shown below:

```
DELETE FROM products;
```

Querying Data

Access the database as instructed in Accessing Data Warehouse and query data as shown below:

1. Enter the specified database, for example, `test` :

```
\\c test;
```

2. Create the `test` table.

```
create table test(a1 int);
```

3. Insert the data.

```
insert into test values(3),(4);
```

4. Query the data.

```
select * from test;
```

Monitoring and Alarming

Alarm Configuration

Last updated：2024-02-02 15:17:07

Background

CDWPG provides a performance monitoring dashboard where you can observe the historical and current status of the operational metrics of each node in the cluster. It also offers the alarm notification feature to keep you informed of sensitive metrics that exceed thresholds, such as the node disk usage.

Performance Monitoring

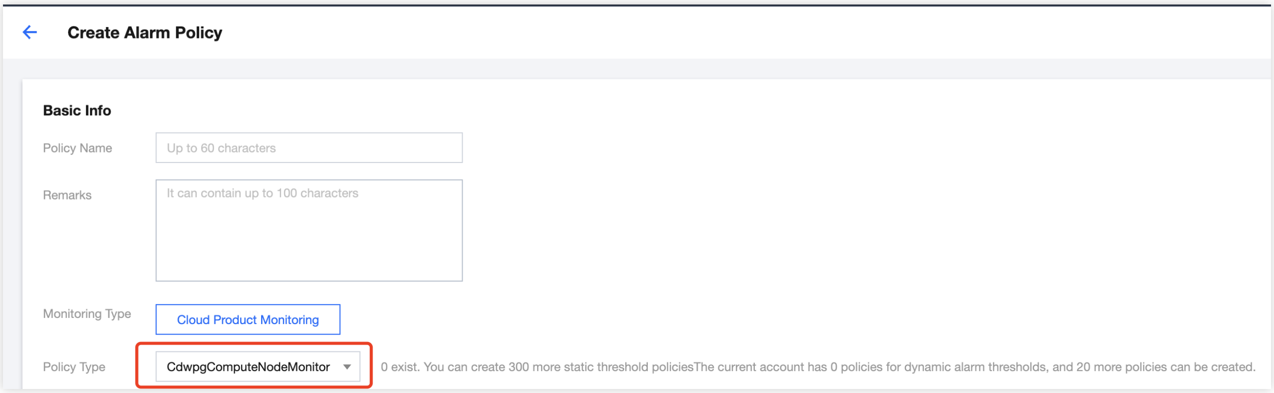
Log in to the CDWPG console and click a cluster name in the cluster list to enter the cluster details page. On the **Performance Monitoring** tab, you can view cluster metrics. If there are multiple nodes, you can select the target node in **Node Dimension**.
Currently, CDWPG metrics include the number of connections, CPU utilization, memory utilization, inbound network throughput, outbound network throughput, write IOPS, read IOPS, disk utilization, read throughput, write throughput, read latency, and write latency.

Alarm Access

CDWPG has three types of alarms, namely cluster monitoring, primary node monitoring, and compute node monitoring. These monitoring alarms are sent to users in three dimensions.

Creating alarm policy

Log in to the CM console, select **Alarm Management > Alarm Configuration > Alarm Policy**, and click **Create** to create an alarm policy. When creating the alarm policy, you can select **CDWPG - cluster monitoring**, **CDWPG - master node monitoring**, or **CDWPG - compute node monitoring** as the **Policy Type**. This document takes compute node monitoring as an example.
1. Select **CDWPG - compute node monitoring` in the **Policy Type drop-down list.



Create Alarm Policy

Basic Info

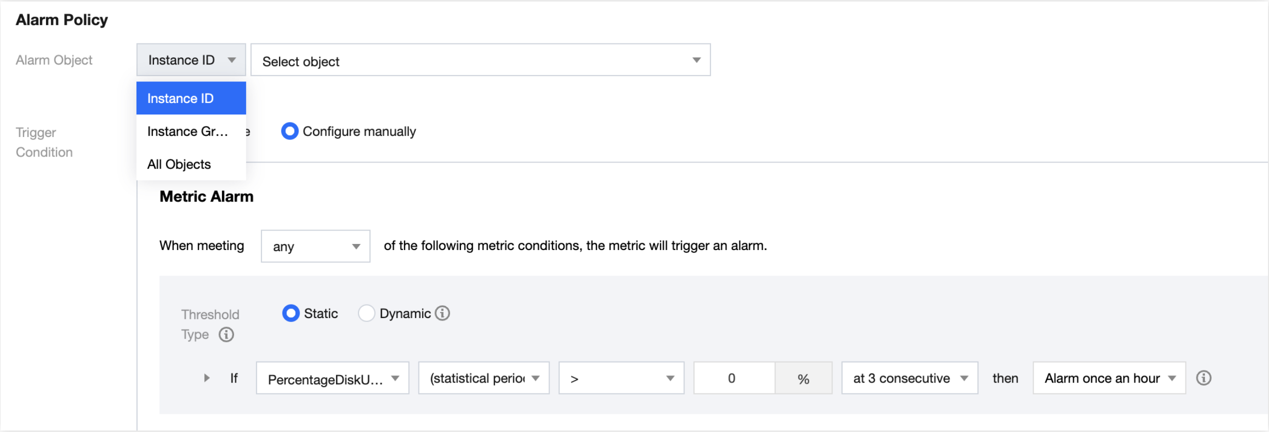
Policy Name: Up to 60 characters

Remarks: It can contain up to 100 characters

Monitoring Type: Cloud Product Monitoring

Policy Type: CdwpgComputeNodeMonitor 0 exist. You can create 300 more static threshold policiesThe current account has 0 policies for dynamic alarm thresholds, and 20 more policies can be created.

2. Set **Alarm Object** and select different groups of compute nodes in the drop-down list.



Alarm Policy

Alarm Object: Instance ID | Select object

Instance ID
Instance Gr...
All Objects

Trigger Condition: ● Configure manually

Metric Alarm

When meeting any of the following metric conditions, the metric will trigger an alarm.

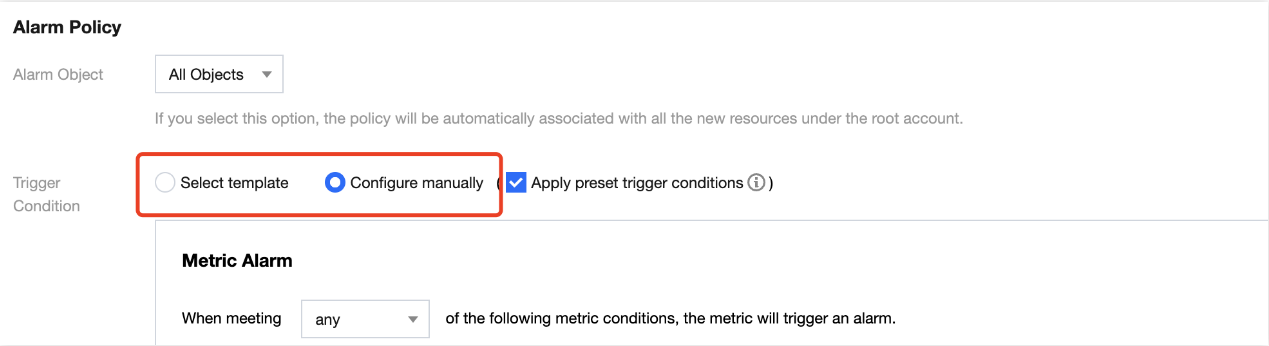
Threshold Type: ● Static ○ Dynamic

If PercentageDiskU... (statistical perio > 0 % at 3 consecutive then Alarm once an hour

3. Configure **Trigger Condition**, which can be **Select template** or **Configure manually**.

If you select **Select template**, you can click **Add Trigger Conditions** to configure an alarm threshold for each metric. Once a metric exceeds the threshold, the system will send you an alarm message. You can also click **Modify Template** to modify an existing template.

If you select **Configure manually**, you can add other metrics that you care about. Set thresholds for such metrics to configure their alarming thresholds and notification periods.



Alarm Policy

Alarm Object: All Objects

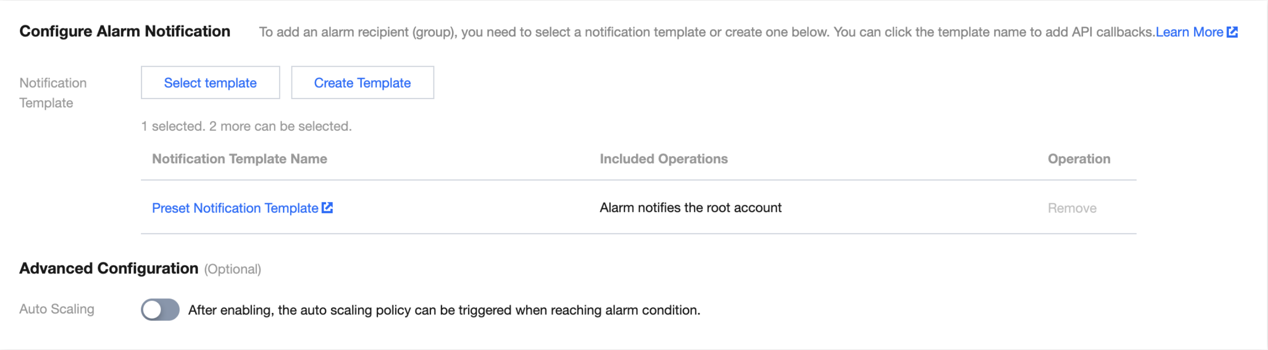
If you select this option, the policy will be automatically associated with all the new resources under the root account.

Trigger Condition: ○ Select template ● Configure manually (☑ Apply preset trigger conditions)

Metric Alarm

When meeting any of the following metric conditions, the metric will trigger an alarm.

4. To configure **Notification Template**, click **Select template** and select an existing template, or click **Create template** to create a notification template.



Configure Alarm Notification To add an alarm recipient (group), you need to select a notification template or create one below. You can click the template name to add API callbacks.Learn More

Notification Template: Select template | Create Template

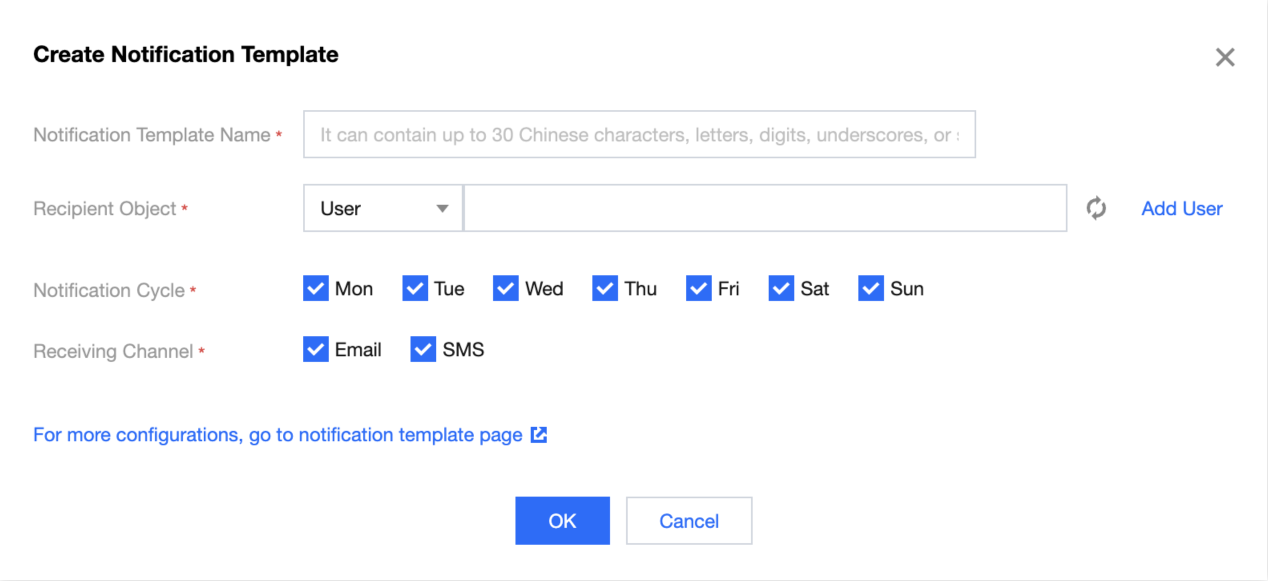
1 selected. 2 more can be selected.

Notification Template Name	Included Operations	Operation
Preset Notification Template	Alarm notifies the root account	Remove

Advanced Configuration (Optional)

Auto Scaling: After enabling, the auto scaling policy can be triggered when reaching alarm condition.

When creating a notification template, set **Recipient** to the development and operations personnel who are interested in alarming or need to handle the alarms. You can choose to notify them by email, SMS, or WeChat.



Create Notification Template

Notification Template Name *: It can contain up to 30 Chinese characters, letters, digits, underscores, or :

Recipient Object *: User | Add User

Notification Cycle *: ☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri ☑ Sat ☑ Sun

Receiving Channel *: ☑ Email ☑ SMS

For more configurations, go to notification template page

Access Management Overview

Last updated：2024-02-02 15:17:07

Concepts

Cloud Access Management (CAM) is a web-based Tencent Cloud service that helps you securely manage the access permissions of resources under your Tencent Cloud account. With CAM, you can create, manage, and terminate users (groups) and use identity and policy management to control user access to Tencent Cloud resources.

Granting Access

You can grant access permissions by specifying a user to perform a specified action on specified resources under a specified condition. Generally, the following four elements are used to describe an access policy: **principal, resource, action, and condition (optional)**.

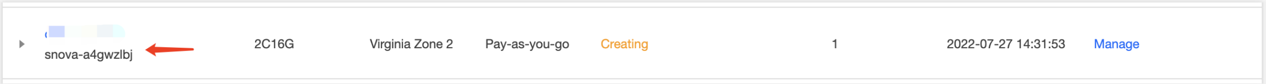
Access Authorization Elements

Tencent Cloud identity

When you sign up for a Tencent Cloud account, the system creates a root account identity for you to log in to the Tencent Cloud services. With the root account, you can use the user management feature to manage different user types, such as **collaborator, message recipient, sub-user, and role**. For specific definitions, see Glossary.

CDWPG cluster resource

CDWPG resources refer to the clusters of Cloud Data Warehouse for PostgreSQL (CDWPG), which are also the objects of access control. Usually, you can see the CDWPG identifier in the console as shown below:



[redacted] snova-a4gwzlbj	2C16G	Virginia Zone 2	Pay-as-you-go	Creating	1	2022-07-27 14:31:53	Manage

Here, `snova-28fg7yl3` is the unique cluster identifier, which can also be seen as the identifier of the CDWPG resource.

CDWPG cluster action

Cluster actions are performed by users in the CDWPG console. Basically, each action can be mapped to a TencentCloud API, such as deleting a cluster and viewing cluster details. Each action has an `action` identifier that can be used for its access control (read and write).

Principle of least privilege

You must specify the scope of the permission granted to the **specified user** for performing **what actions** and access **what resources** under **what conditions**.

Setting Policy

Last updated：2024-02-02 15:17:07

Overview

Access policies can be used to grant access to CDWPG instances. They use JSON-based access policy syntax. You can authorize specified principals to perform specified operations on specified CDWPG resources through the access policy syntax.

The access policy syntax describes the basic elements and usage of the policy. For the description of the policy syntax, see Permissions and Policies.

Elements in Access Policy

An access policy contains the following elements with basic meanings:

statement: It describes the details of one or more permissions. It contains a permission or permission set of multiple other elements such as `effect` , `action` , `resource` , and `condition` . One policy must and can have only one `statement` .

effect: It is required and describes the result of a statement. The result can be an "allow" or "explicit deny".

action: It is required and describes the allowed or denied action (operation). An operation can be an API (prefixed with "name") or a feature set (a set of specific APIs prefixed with "permid").

resource: It is required and describes the details of authorization. A resource is described in a six-segment format. Detailed resource definitions vary by product.

condition: It is optional and describes the condition for the policy to take effect. A condition consists of operator, action key, and action value. A condition value may contain information such as time and IP address. Some services allow you to specify additional values in a condition.

Element Usage

Specifying effect

If access to a resource is not explicitly granted (allowed), then it is implicitly denied. It can also be explicitly denied, which ensures that users cannot access the resource even if they are granted the access permission by other policies. Below is an example of specifying the "allow" effect:

```
"effect" : "allow"
```

Specifying action

CDWPG defines console operations that can be specified in a policy. The specified operations are divided into reading part of APIs (cdwpg:Describe*) and all APIs (cdwpg:*) according to the operation nature.

Below is an example of specifying the allowed operations:

```
"action": [
  "name/cdwpg:Describe*"
]
```

Specifying resource

The `resource` element describes one or more operation objects, such as CDWPG resources. All resources can use the following six-segment format:

```
qcs:project_id:service_type:region:account:resource
```

The parameters are as detailed below:

Parameter	Description	Required
qcs	Tencent Cloud service abbreviation, which indicates a service of Tencent Cloud	Yes
project_id	Project information, which is only used to enable compatibility with legacy CAM logic and generally can be left empty	No

service_type	Product abbreviation, which is `cdwpg` here	Yes
region	Region information	Yes
account	Root account information of the resource owner, i.e., root account UIN in the format of `uin/${OwnerUin}` , such as uin/100000000001	Yes
resource	Resource details prefixed with `cdwpg-instance`	Yes

Below is a sample six-segment CDWPG resource description:

```
"resource":["qcs::cdwpg:ap-guangzhou:uin/100000000001:cdwpg-instance/snova-73jingds
```

Specifying condition

The access policy syntax allows you to specify the condition when granting permissions, which is mainly used to set tag authentication in CDWPG. The tag condition takes effect only for clusters bound with the tag. Below is a sample tag policy:

```
"condition": {
               "for_any_value:string_equal": {
                   "qcs:tag": [
                       "jing&jingfdd"
                   ]
               }
           }
```

This statement means that the policy contains resources whose tag key is `jing` and tag value is `jingfdd`.

Use Cases

In the following case, the policy is to allow access to the resource whose cluster ID is `snova-jidnshgdsh` under UIN 1250000000 and the resources whose tag key is `testkey` and tag value is `testvalue`.

```
{
   "version": "2.0",
    "statement": [
        {
            "action": [
                "name/cdwpg:Describe*",
            ],
            "condition": {
               "for_any_value:string_equal": {
                    "qcs:tag": [
                        "testkey&testvalue"
                    ]
                }
            },
            "effect": "allow",
            "resource": [
                "qcs::cdwpg:ap-guangzhou:uin/1250000000:cdwpg-instance/snova-jidnsh
            ]
        }
    ]
}
```

Granting Policy

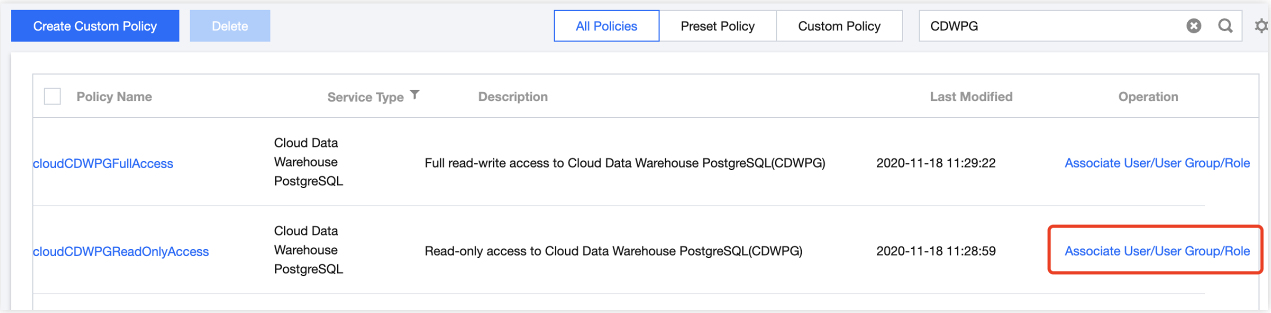
Last updated：2024-02-02 15:17:07

Preset CDWPG Policy Management

To facilitate authorizing sub-accounts, CDWPG provides two preset policies. Go to the CAM console, search for CDWPG in the top-right corner of the page, and you can see the following two policies:

Policy	Description
QcloudCDWPGFullAccess	Grants full access to CDWPG management
QcloudCDWPGReadOnlyAccess	Grants read-only access to CDWPG management

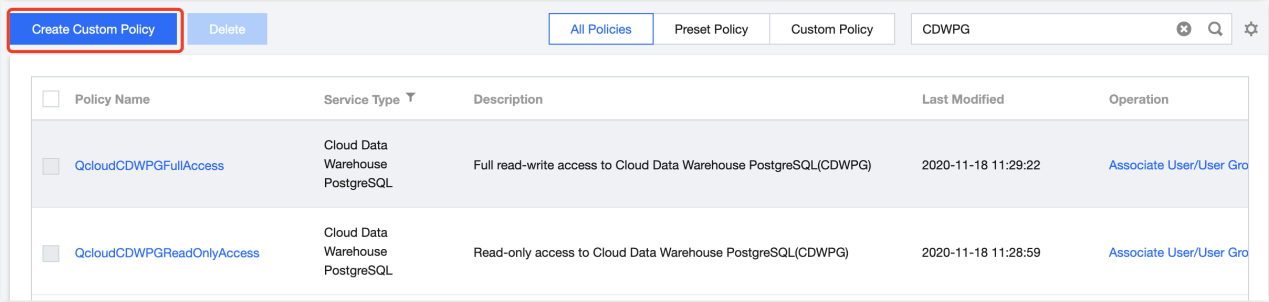
You can use the `QcloudCDWPGFullAccess` policy to grant a user permissions to create and manage CDWPG instances.
You can use the `QcloudCDWPGReadOnlyAccess` policy to grant a user permissions to query but not create, delete, or modify CDWPG clusters and resources (VPCs, security groups, and monitors).



□ Policy Name	Service Type	Description	Last Modified	Operation
cloudCDWPGFullAccess	Cloud Data Warehouse PostgreSQL	Full read-write access to Cloud Data Warehouse PostgreSQL(CDWPG)	2020-11-18 11:29:22	Associate User/User Group/Role
cloudCDWPGReadOnlyAccess	Cloud Data Warehouse PostgreSQL	Read-only access to Cloud Data Warehouse PostgreSQL(CDWPG)	2020-11-18 11:28:59	Associate User/User Group/Role

Custom CDWPG Policy

If preset policies cannot meet your needs, you can click **Create Custom Policy** to create custom policies.

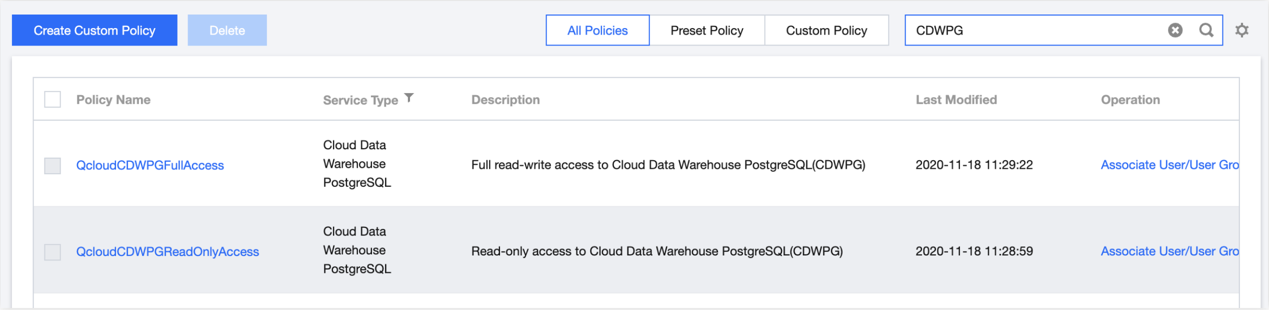


Policy Name	Service Type	Description	Last Modified	Operation
QcloudCDWPGFullAccess	Cloud Data Warehouse PostgreSQL	Full read-write access to Cloud Data Warehouse PostgreSQL(CDWPG)	2020-11-18 11:29:22	Associate User/User Gro
QcloudCDWPGReadOnlyAccess	Cloud Data Warehouse PostgreSQL	Read-only access to Cloud Data Warehouse PostgreSQL(CDWPG)	2020-11-18 11:28:59	Associate User/User Gro

For the method of custom policy creation, see Policy Settings.

Policy Authorization

A configured policy can grant permissions by associating user groups or sub-users.



Policy Name	Service Type	Description	Last Modified	Operation
QcloudCDWPGFullAccess	Cloud Data Warehouse PostgreSQL	Full read-write access to Cloud Data Warehouse PostgreSQL(CDWPG)	2020-11-18 11:29:22	Associate User/User Gro
QcloudCDWPGReadOnlyAccess	Cloud Data Warehouse PostgreSQL	Read-only access to Cloud Data Warehouse PostgreSQL(CDWPG)	2020-11-18 11:28:59	Associate User/User Gro

Resource Types Authorizable by Custom Policy

Resource-level permission can be used to specify which resources a user can manipulate. CDWPG supports certain resource-level permissions. This means that for CDWPG operations that support resource-level permission, you can control the time when a user is allowed to perform operations or to use specified resources. The following table describes the types of resources that can be authorized in CAM.

Resource Type	Resource Description Method in Authorization Policy
CDWPG	`qcs::cdwpg:$region:$account:cdwpg-instance/*` <br> `qcs::cdwpg:$region:$account:cdwpg-instance/$clusterId`

The following table describes the CDWPG API operations that currently support resource-level permissions. When setting a policy, you can enter the API operation name in the `action` field to control the individual API. You can also use `\\*` as a wildcard to set the `action` .

List of APIs supporting resource-level authorization

API Operation	Resource Path
ModifyClusterSize	Modifies the number of cluster nodes
DescribeClusters	Gets cluster details
DescribeRealtimeQuery	Gets real-time query details of a cluster
DescribeHistoryQuery	Gets historical query details of a cluster
AbortQuery	Aborts a cluster query
DescribeRealtimeQueries	Gets the list of real-time queries in a cluster
DescribeGpStatus	Gets the cluster database status
RebootCluster	Restarts a cluster
DescribeClusterStatus	Gets the cluster status
ModifyClusterSubnet	Modifies the cluster subnet
DescribeHistoryQueries	Gets the list of historical queries in a cluster
DeleteCluster	Deletes a cluster
ModifyClusterUserPassword	Resets the cluster password
ModifyClusterBasic	Renames a cluster
DescribeClustersStatistics	Gets the number of clusters
DescribeVpcLinks	Gets the VPC access link of a cluster
CreateVpcLink	Creates a VPC access link
DeleteVpcLink	Deletes a VPC access link
ExpandClusterSize	Scales a cluster
DescribeHbaConfigList	Gets the access address allowlist of a cluster
SetHbaConfigList	Modifies the access address allowlist of a cluster

DescribeClusterResourceQueueList	Queries the resource queue list of a database cluster
DescribeClustersLimit	Queries the resource limit configuration of a database cluster
HandlerResourceQueue	Manipulates a database resource queue
AdminClusterOutnetAddres	Manages the public IP
DescribeClustersNodesInfo	Gets the node information of a cluster

List of APIs not supporting resource-level authorization

For CDWPG API operations that don't support resource-level authorization, you can still authorize a user to perform them, but you must specify `*` as the `resource` element in the policy statement.

API Operation	API Description
DescribeNodeConfigInfo	Gets the node model specification information
DescribeEvents	Gets the information of all cluster events
CreateCluster	Creates a cluster
DescribeDbStatus	Gets the database status
DescribeZones	Gets AZs available for purchase
DescribeSegNodeMaxCount	Queries the maximum number of compute nodes
DescribeClusterExtend	Gets all Ops information of a cluster
DescribeResidual	Gets the resource status in a region
DescribeSpecResidual	Checks whether specific specifications are sold out
DescribeZonesResource	Gets the resource information in an AZ
DescribeValidRegionAndZones	Gets valid regions and AZs