Cloud Service Engine

User Guide

Issue 01

Date 2025-09-19

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 Before You Start

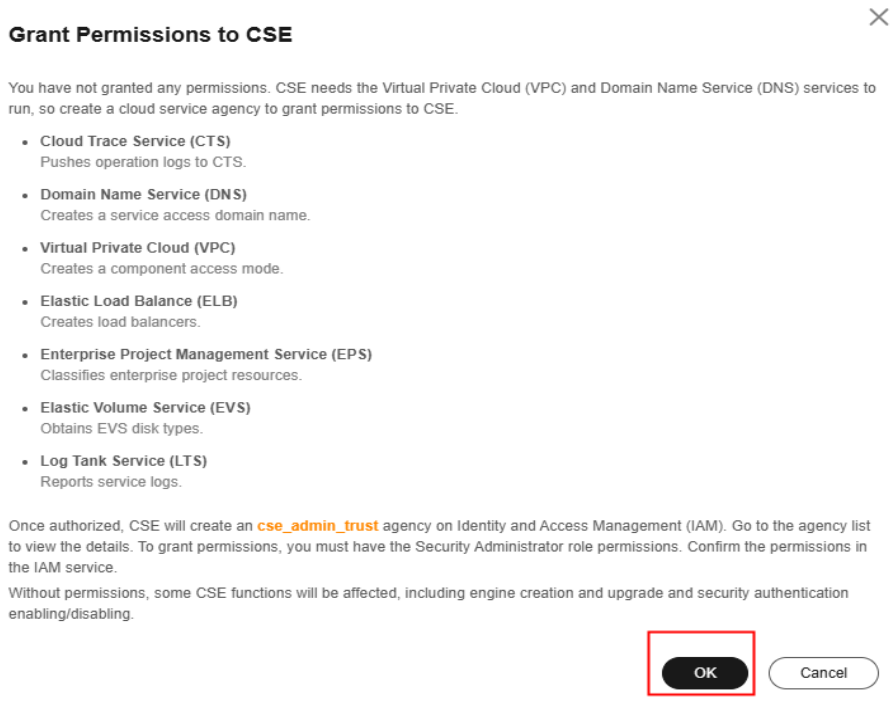
To use CSE, ensure that the following conditions are met:

1. You have **registered a HUAWEI ID and enabled Huawei Cloud services**.
2. The current login account has been authorized to use CSE. For details, see **Creating a User and Granting CSE Permissions**.

Restrictions

CSE needs the Virtual Private Cloud (VPC) and Domain Name Service (DNS) services to run. If you have not granted any permissions, create a cloud service agency to grant permissions to CSE. That is, click **OK** in the **Grant Permissions to CSE** dialog box. CSE will create an agency named cse_admin_trust on IAM. Go to the agency list to view the details. You must have the Security Administrator role. Without permissions, some CSE functions will be affected, including engine creation and upgrade and security authentication enabling/disabling.

Figure 1-1 Creating an agency



2 Creating a User and Granting CSE Permissions

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

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own security credentials for access to CSE resources.
- Grant only the permissions required for users to perform a task.
- Entrust a Huawei Cloud account or cloud service to perform professional and efficient O&M on your CSE resources.

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

This section describes the procedure for granting permissions (see **Figure 2-1**).

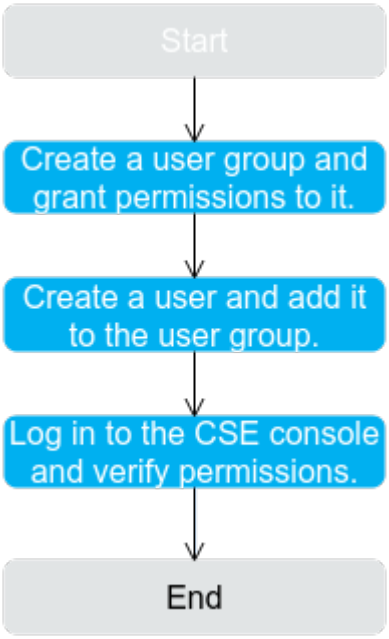
Prerequisites

Before assigning permissions to user groups, you should learn about CSE policies and select the policies based on service requirements.

- For details about system permissions supported by CSE, see **Permissions**.
- To grant permissions for other services, learn about all permissions supported by IAM by referring to **System-defined Permissions**.

Process Flow

Figure 2-1 Process of granting CSE permissions



1. Create a user group and grant permissions to it.

   Create a user group on the IAM console, and grant the **CSE ReadOnlyAccess** policy to the group.

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

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

3. Log in to CSE and verify permissions.

   Log in to the CSE console as the created user, and verify that it has the read-only permission for CSE.

   - In **Service List**, choose **Cloud Service Engine**. On the console, choose **ServiceComb** > **Buy Exclusive Microservice Engine**. If a message is displayed indicating insufficient permissions, the **ReadOnlyAccess** policy has taken effect.
   - Choose any other service in **Service List**. If a message is displayed indicating insufficient permissions to access the service, the **ReadOnlyAccess** policy has taken effect.

CSE Custom Policies

Custom policies can be created as a supplement to the system-defined policies of CSE.

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

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Create a JSON policy or edit an existing one.

For details, see **Creating a Custom Policy**. The following section contains examples of common CSE custom policies.

Example Custom Policies

This procedure creates a policy that an IAM user is prohibited to create and delete a microservice engine.

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "cse:*:*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cse:engine:create",
                "cse:engine:delete"
            ],
            "Effect": "Deny"
        }
    ]
}
```

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

You can verify your granted permissions using the console or REST APIs.

The following uses the custom policy as an example to describe how to verify that a user is not allowed to create microservice engines on the console.

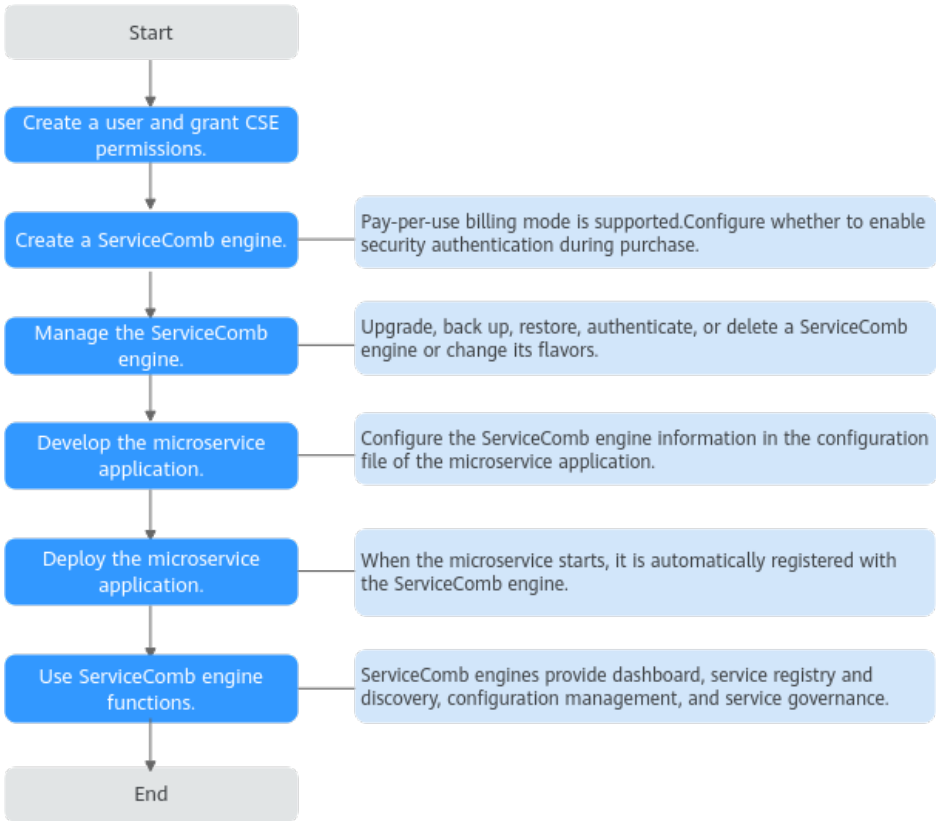
1. Log in to Huawei Cloud as an **IAM user**.
   - **Tenant name**: Name of the account used to create the IAM user
   - **IAM username** and **IAM user password**: Username and password specified during IAM user creation using the **Tenant name**
2. Create a microservice engine on the CSE console. If error 403 is returned, the permissions are correct and are in effect.

3 ServiceComb Engines

3.1 ServiceComb Engine Overview

The ServiceComb engine provides service registry, service governance, and configuration management. It allows you to quickly develop microservice applications and implement high-availability O&M, and supports multiple languages, multiple runtime systems, and Spring Cloud and Apache ServiceComb Java Chassis (Java chassis) frameworks.

The process of using a ServiceComb engine is as follows:



1. **Create a user and grant CSE permissions**.

2. **Create a ServiceComb engine**.
3. Upgrade, back up, restore, authenticate, or delete a ServiceComb engine or change its flavors by referring to **Managing ServiceComb Engines**.
4. Configure the ServiceComb engine information in the configuration file of the developed microservice application, including the configuration center, registry center, dashboard, and governance policy. For details, see **Developing Microservice Applications** and **Connecting Microservice Applications**.
5. Deploy the microservice. When the microservice starts, it is automatically registered with the ServiceComb engine. For details, see **Deploying Microservice Applications**.
6. Manage microservices using the functions provided by the ServiceComb engine. For details, see **Viewing Microservice Running Metrics Through the Microservice Dashboard** to **System Management**.

3.2 Creating a ServiceComb Engine

This section describes how to create a ServiceComb engine.

Restrictions

- When a ServiceComb engine is in use, do not disable the enterprise project. Otherwise, the engine will not be displayed in the engine list, affecting normal use.
- The VPC cannot be changed once the engine is created.
- By default, a maximum of five ServiceComb engines can be created for each project. To create more, **submit a service ticket** to increase the quota. For details about projects, see **Project**.

Prerequisites

- The login account has the permission to create a microservice engine. For details, see **Creating a User and Granting CSE Permissions**.
- A ServiceComb engine runs on a VPC. Before creating a ServiceComb engine, ensure that VPCs and subnets are available. For details, see **Creating a VPC with a Subnet**.
- If the engine is created using an account with the minimum permission for creating engines, for example, **cse:engine:create** in the **fine-grained permission dependencies of microservice engines**, the default VPC security group cse-engine-default-sg needs to be preset by the primary account and the rules listed in **Table 3-1** need to be added. For details, see **Adding a Security Group Rule**.

Table 3-1 cse-engine-default-sg rules

Direction	Priority	Policy	Protocol and Port	Type	Source Address
Inbound	1	Allow	ICMPv6: All	IPv6	::/0

Direction	Priority	Policy	Protocol and Port	Type	Source Address
	1	Allow	TCP: 30100–30130	IPv6	::/0
	1	Allow	All	IPv6	cse-engine-default-sg
	1	Allow	All	IPv4	cse-engine-default-sg
	1	Allow	TCP: 30100–30130	IPv4	0.0.0.0/0
	1	Allow	ICMP: all	IPv4	0.0.0.0/0
Outbound	100	Allow	All	IPv4	0.0.0.0/0
	100	Allow	All	IPv6	::/0

Creating a ServiceComb Engine

Step 1 Go to the **Buy Exclusive ServiceComb Engine** page.

Step 2 Set parameters according to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Billing Mode	Billing mode. Currently, **Pay-per-use** is supported.
*Enterprise Project	Select the project where the ServiceComb engine is located. You can search for and select the required enterprise project from the drop-down list. Enterprise projects let you manage cloud resources and users by project. An enterprise project can be used after it is created and enabled. For details, see **Enabling the Enterprise Project Function**. By default, **default** is selected. After a ServiceComb engine is created, you can remove ServiceComb engine resources out of the current enterprise project and into a new enterprise project. For details, see **Removing Resources from an Enterprise Project** and **Adding Resources to an Enterprise Project**.
*Instances	Select the microservice instance quota. You can select the number of instances based on the number of microservice instances to be hosted. ServiceComb engine instances with different microservice instances are rewarded with corresponding configuration items and the maximum number of microservice versions supported.

Parameter	Description
*Engine Type	Select a ServiceComb engine type. If the engine type is cluster, the engine is deployed in cluster mode and supports host-level DR.
*Name	Enter a ServiceComb engine name. The name contains 3 to 24 characters, including letters, digits, and hyphens (-), and starts with a letter but cannot end with a hyphen (-). The name cannot be **default**.
*AZ	Availability zone. Select one or three AZs for the engine based on the number of AZs in the environment. • Select one AZ to provide host-level DR. • Select three AZs to provide AZ-level DR. **NOTE** • The AZs in one region can communicate with each other over an intranet. • Multiple AZs enhance DR capabilities.
*Network	Select a VPC and subnet to provision logically isolated, configurable, and manageable virtual networks for your engine. • To use a created VPC, search for and select a VPC created under the current account from the drop-down list. • To use a new VPC, click **Create VPC** in the drop-down list. For details, see **Creating a VPC with a Subnet**.
Description	Click  and enter the engine description. The description can contain 0 to 255 characters.
Tag	Tags are used to identify cloud resources. When you have multiple cloud resources of the same type, you can use tags to classify them based on usage, owner, or environment. Click ⊕ **Add Tag**. In the **Add Tag** dialog box, enter a tag key and value. For details about tag naming rules, see **Managing Tags**. In the **Add Tag** dialog box, you can click ⊕ **Add Tag** to add multiple tags at a time, or click ⊖ next to a tag to delete the tag.

Parameter	Description
Authentication Mode	The exclusive ServiceComb engine with security authentication enabled provides the system management function using the role-based access control (RBAC) through the microservice engine console. • Select **Enable security authentication**: 1. Determine whether to enable **Authenticate Programming Interface**. After it is enabled, you need to add the corresponding account and password to the microservice configuration file. Otherwise, the service cannot be registered with the engine. After it is disabled, you can register the service with the engine without configuring the account and password in the microservice configuration file, which improves the efficiency. You are advised to disable this function when accessing the service in a VPC. 2. Enter and confirm the password of user **root**. Keep the password secure. • Select **Disable security authentication**: Disable security authentication. You can enable it after the instance is created.

Step 3 Click **Buy**. The page for confirming the engine information is displayed.

Step 4 Click **Submit**. When the engine status changes to **Available**, the creation is successful.

NOTE

- It takes about 10–30 minutes to create a ServiceComb engine.
- After the ServiceComb engine is created, its status is **Available**. For details about how to check the engine status, see **Viewing ServiceComb Engine Information**.
- If the ServiceComb engine fails to be created, view the failure cause on the **Operation** page and rectify the fault. Then, you can perform the following operations:
  - In the **ServiceComb Engine Information** area, click **Retry** to create an engine again.
  - If the retry fails, delete the ServiceComb engine that fails to be created. For details, see **Deleting a ServiceComb Engine**.

----End

3.3 Managing ServiceComb Engines

3.3.1 Viewing ServiceComb Engine Information

You can click an engine to go to the engine details page and obtain the basic information, service registry and discovery address, configuration center address, instance quota, and configuration item quota.

Viewing Basic ServiceComb Engine Information

In the **ServiceComb Engine Information** area, you can view the microservice engine information as shown in **Table 3-2**.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **ServiceComb Engine Information** area, view the engine information shown in **Table 3-2**.

Table 3-2 Engine details

Item	Description
Name	Engine name entered when **creating the ServiceComb engine**. Click to copy it. Click to change an engine name. The name contains 3 to 24 characters, including letters, digits, and hyphens (-), and starts with a letter but cannot end with a hyphen.
Engine ID	Engine ID. Click to copy it.
Status	Engine status. • Creating • Available • Unavailable • Configuring • Deleting • Upgrading • Resizing • Creation failed • Deletion failed • Upgrade failed • Resizing failed • Frozen • Unknown
Version	Engine version.
Engine Type	Engine specification selected when **creating the ServiceComb engine**.
AZ	AZ selected when **creating the ServiceComb engine**.
Tag	Tags added to the ServiceComb engine. You can also click **Tag Management** and perform operations on tags as required. For details, see **Managing ServiceComb Engine Tags**.

Item	Description
Description	Engine description entered when **creating the ServiceComb engine**.

----End

Obtaining the Service Center Address of a ServiceComb Engine

The registry and discovery address is the core of service registry and discovery to implement dynamic service management. When a microservice is started, the registry center address is used to report its metadata (such as the IP address, port, service contract, and version) to the ServiceComb service center. This implements automatic service registry, avoids hard-coded addresses, and simplifies O&M. Service consumers use this address to query the list of available service instances from the registry center, select instances based on load balancing (such as weighted round robin), and dynamically detect service startup and shutdown (through heartbeats) to ensure that faulty instances are automatically removed, improving system fault tolerance.

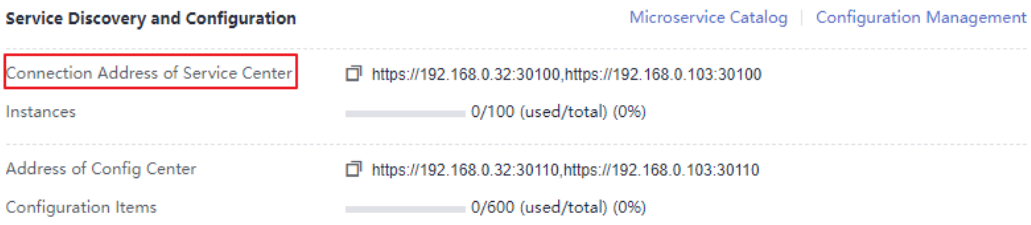
The service center address cannot be changed after the engine is created.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Service Discovery and Configuration** area, view the service center address of the microservice engine.



----End

Obtaining the Configuration Center Address of a ServiceComb Engine

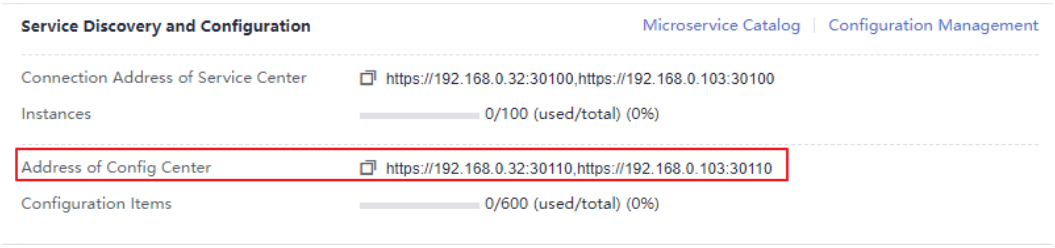
The configuration center manages microservice configurations. When a microservice is connected to the ServiceComb engine, you need to configure the configuration center address of the engine in the configuration file. ServiceComb establishes a persistent connection with the configuration center through this address. When detecting configuration changes, the configuration center pushes the changes to the microservice instance, and the configuration can be updated without restart, implementing dynamic configuration management.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Service Discovery and Configuration** area, view the configuration center address of the microservice engine.



NOTE

- For ServiceComb engine 1.x, the port number of the configuration center address is 30103.
- For ServiceComb engine 2.x, the port number of the configuration center address is 30110.

----End

Viewing the Instance Quota of a ServiceComb Engine

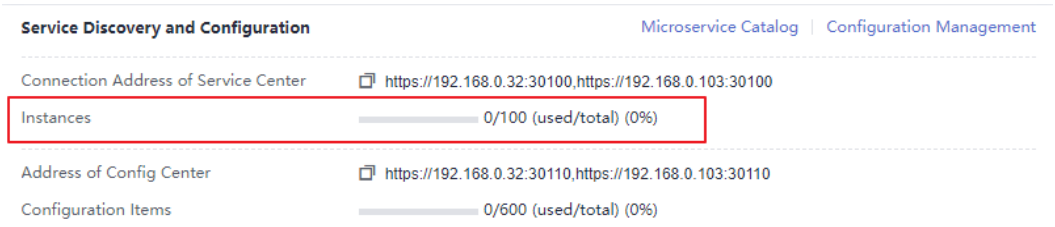
This section describes how to view the instance quota and quota usage of a ServiceComb engine.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Service Discovery and Configuration** area, view the instance quota and quota usage of the microservice engine.



----End

Viewing the Configuration Item Quota of a ServiceComb Engine

This section describes how to view the configuration item quota and quota usage of a ServiceComb engine.

NOTE

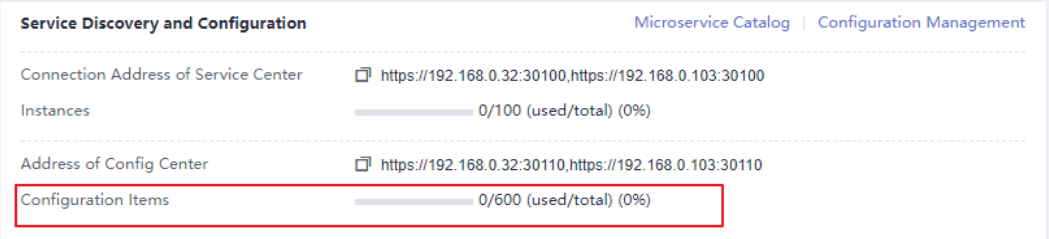
This section applies only to ServiceComb engine 2.x.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Service Discovery and Configuration** area, view the configuration item quota and quota usage of the microservice engine.



----End

3.3.2 Managing ServiceComb Engine Tags

Tags facilitate ServiceComb engine identification and management.

If your organization has configured tag policies for ServiceComb engines, add tags to engines based on the policies. If a tag does not comply with the tag policies, engine creation may fail. Contact your administrator to learn more about tag policies.

You can add tags to a ServiceComb engine when creating the engine or add tags on the details page of the created engine. Up to 20 tags can be added to an engine. Tags can be modified and deleted.

A tag consists of a tag key and a tag value. **Table 3-3** lists the tag key and value requirements.

Table 3-3 Tag naming rules

Tag	Rule
Key	• Cannot be left blank. • Must be unique for the same instance. • Contain a maximum of 128 characters. • Contain only letters, digits, spaces, and special characters (_ . : = + - @). • Cannot start with a space or _**sys**_ or end with a space.
Value	• Contain a maximum of 255 characters. • Contain only letters, digits, spaces, and special characters (_ . : = + - @).

Managing Tags

Adding or modifying tags will affect engine services for about 10 seconds. Add or modify tags during off-peak hours.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine. The details page is displayed.

Step 4 In the **ServiceComb Engine Information** area, perform the following operations in the **Tag** field as required:

- Add a tag
  a. Click **Tag Management**. The **Edit Tag** dialog box is displayed.
  b. Click **Add Tag** and enter a tag key and value in the text boxes.
  c. Click **OK**.
- Modify a tag
  a. Click **Tag Management**. The **Edit Tag** dialog box is displayed.
  b. You can modify the tag key and value in the original text boxes.
  c. Click **OK**.
- Delete a tag
  Click in the row that contains the tag to be deleted. In the dialog box that is displayed, click **OK** to delete the tag.

----End

3.3.3 Managing Public Network Access for a ServiceComb Engine

The ServiceComb engine supports public network access, which helps expand the microservice architecture from a closed environment on the intranet to an open ecosystem on the public network. This not only meets the requirements of Internet services and cross-cloud collaboration, but also improves the system elasticity, scalability, and DR capabilities through the openness of the technical architecture. In addition, based on the security and governance mechanisms, public network access is balanced between openness and controllability, providing key technical support for the business model of internal and external linkage in enterprise digital transformation.

Restrictions

- ServiceComb engines that do not have security authentication enabled do not have the authentication and authorization capabilities. Opening those engines to the public network may cause security risks and increases the system vulnerability. For example, data assets such as configurations and service information may be stolen.
- Do not use this function in a production environment or a network environment with high security requirements.

Prerequisites

An EIP has been created. For details, see **Assigning an EIP**.

Binding an EIP

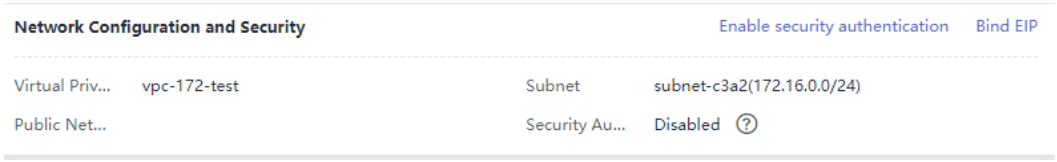
ServiceComb engines that are bound with EIPs can be accessed from the public network.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Network Configuration and Security** area, click **Bind EIP**.



Step 5 Read the security risk prompt in the displayed dialog box and select **I understand the security risks**.

Step 6 In the **EIP** drop-down list, select the EIP to be bound. You can only select an EIP in the same enterprise project as the ServiceComb engine.

Step 7 Click **OK**.

----End

Unbinding an EIP

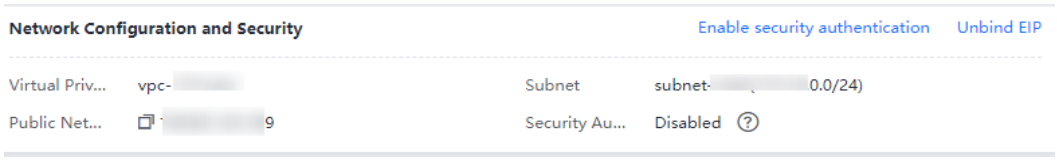
If an EIP has been bound to a ServiceComb engine, you can unbind the EIP from the engine to disable the public network access to the engine.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Network Configuration and Security** area, click **Unbind EIP**.



Step 5 In the displayed dialog box, click **OK**.

----End

3.3.4 Managing Security Authentication for a ServiceComb Engine

A ServiceComb engine may be used by multiple users. Different users must have different engine access and operation permissions based on their responsibilities and permissions. If security authentication is enabled for an exclusive ServiceComb engine, grant different access and operation permissions to users based on the roles associated with the accounts used by the users to access the engine.

For details about security authentication, see **System Management**.

You can enable or disable security authentication for the exclusive ServiceComb engine based on service requirements.

- **Enabling Security Authentication**

  If a ServiceComb engine is available with security authentication disabled, you can enable security authentication based on service requirements.

  After security authentication is enabled and programming interface authentication is also enabled, if security authentication parameters are not configured for the microservice components connected to the engine, or the security authentication account and password configured for the microservice components are incorrect, the heartbeat of the microservice components fails and the service is forced to go offline. Perform the following steps:

  - Spring Cloud: For details, see **Connecting Spring Cloud Applications to ServiceComb Engines**.
  - Java chassis: For details, see **Connecting Java Chassis Applications to ServiceComb Engines**.

- **Disabling Security Authentication**

  If a ServiceComb engine is available with security authentication enabled, you can disable security authentication based on service requirements.

  After security authentication is disabled for a microservice component, service functions of the microservice component are not affected no matter whether security authentication parameters are configured for the microservice component.

Restrictions

Currently, Java chassis and Spring Cloud support security authentication for microservices. The Java chassis version must be 2.3.5 or later, and Spring Cloud must integrate Spring Cloud Huawei 1.6.1 or later.

Enabling Security Authentication

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Network Configuration and Security** area, click **Enable security authentication**.

- If the engine version is earlier than 1.2.0, go to **Step 5**.

- If the engine version is 1.2.0 or later, go to **Step 6**.

Step 5 Upgrade the engine to 1.2.0 or later.

1. Click **Upgrade**.
2. Select **Target Version** and view the version description. Determine whether to upgrade the software to this version. Then, click **OK**.
3. Select the upgraded ServiceComb engine. In the **Network Configuration and Security** area, click **Enable security authentication**.

Step 6 On the **System Management** page, enable security authentication.

- To enable security authentication for the first time, click **Enable security authentication**.

  You need to create user **root** first. Enter and confirm the password of user **root**. Then, click **Create Now**.
- Enable security authentication again and enter the name and password of the account associated with the **admin** role in the engine.

Step 7 (Optional) Create a role based on service requirements. For details, see **Roles**.

Step 8 (Optional) Create an account based on service requirements. For details, see **Accounts**.

Step 9 On the **System Management** page, click **Enable security authentication** and configure the security settings.

- If you enable **Authenticate Console**, go to **Step 11**.

  To log in to CSE after **Authenticate Console** is enabled, determine whether to use an account and password based on the permissions of the login IAM user. The login user can only view and configure services on which the user has permission.
- If you enable **Authenticate Programming Interface**, go to **Step 10**.

  After **Authenticate Programming Interface** is enabled, **Authenticate Console** is automatically enabled.

  After it is enabled, you need to add the corresponding account and password to the microservice configuration file. Otherwise, the service cannot be registered with the engine.

  After it is disabled, you can register the service with the engine without configuring the account and password in the microservice configuration file, which improves the efficiency. You are advised to disable this function when accessing the service in a VPC.

Step 10 Configure the SDK. For microservice components that have been deployed but not configured with security authentication parameters, configure the account name and password for security authentication and then upgrade the component. For details, see **Configuring the Security Authentication Account and Password for a Microservice**.

Step 11 Click **OK**.

After the ServiceComb engine is updated and the engine status changes from **Configuring** to **Available**, security authentication is enabled successfully.

----End

Disabling Security Authentication

After security authentication is disabled, accounts created on the engine will not be deleted.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Network Configuration and Security** area, click **Disable security authentication**.

Step 5 Click **OK**. After the ServiceComb engine is updated and the engine status changes from **Configuring** to **Available**, security authentication is disabled successfully.

----End

3.3.5 Configuring Backup and Restoration of a ServiceComb Engine

CSE provides the backup and restoration functions. You can back up and restore ServiceComb engine data, including microservices, contracts, configurations, and account role.

You can customize backup policies to periodically back up ServiceComb engines or manually back up ServiceComb engines.

Restrictions

- Each ServiceComb engine supports a maximum of 15 successful backups, including a maximum of 10 manual backups and a maximum of 5 automatic backups.
- The backup data will be stored for 10 days. Expired backup data will be deleted.
- The backup data will overwrite the current data of the ServiceComb engine. As a result, the microservice and service instances may be messed, and dynamic configurations may be lost. Exercise caution when performing this operation.
- If security authentication is enabled, the backup data contains the account information. You are advised to disable security authentication before restoring the backup data. Otherwise, the authentication for accessing the ServiceComb engine may fail after the restoration.

Automatic Backup

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Backup and Restoration** area, click **Automatic backup settings** and set backup parameters.

Table 3-4 Automatic backup parameters

Parameter	Description
Automatic Backup	After automatic backup is disabled, the previously set backup policy will be deleted. In this case, you need to set the backup policy again.
Backup Interval	Backup period. Interval at which the system automatically backs up data. You can select one or more days from Monday to Sunday as the backup execution date. The selected dates will trigger automatic backup. This parameter takes effect after **Automatic Backup** is enabled.
Trigger Time	Time when a backup task starts. The backup task is triggered at the specified time on all the dates of the backup interval. Only the hour is supported. This parameter takes effect after **Automatic Backup** is enabled.

Step 5 Click **OK**.

Once the backup policy is set, the backup task is triggered within one hour after the preset time.

----End

Manual Backup

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the engine whose data you want to manually back up.

Step 4 In the **Backup and Restoration** area, click **Create Manual Backup** and set backup parameters.

Table 3-5 Manual backup parameters

Parameter	Description
Name	Enter a backup task name. The name contains 3 to 24 characters, including letters, digits, and hyphens (-), and starts with a letter but cannot end with a hyphen (-). The name cannot be changed once the backup task is created.
Remarks	(Optional) Enter a description. The description can contain 0 to 255 characters.

Step 5 Click **OK** to execute the backup task immediately. A backup task is generated in the backup task list. If the execution result changes from **Processing** to **Successful**, the manual backup is successful.

----End

Restoring Backup Data

The backup data will overwrite the current data of the ServiceComb engine. As a result, the microservice and service instances may be messed, and dynamic configurations may be lost. Exercise caution when performing this operation.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Backup and Restoration** area, click **Restore** in the **Operation** column of the row that contains the specified backup data.

1. Select **I have read and fully understand the risks**.
2. Click **OK**. To view the restoration status, click **Restoration History** in the **Backup and Restoration** area.

----End

Deleting Backup Data

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the engine whose backup data you want to delete.

Step 4 In the **Backup and Restoration** area, click **Delete** in the **Operation** column of the target backup data. In the displayed dialog box, enter **DELETE** and click **OK**.

----End

3.3.6 Upgrading a ServiceComb Engine

ServiceComb engines are created using the latest engine version. When a later version is released, you can upgrade your engine.

During upgrade, two instances are upgraded in rolling mode without service interruptions. However, one of the two access addresses may be unavailable. In this case, you need to quickly switch to the other instance. Currently, ServiceComb SDK and Mesher support instance switching. If you call the APIs of the service center and configuration center for service registry and discovery, instance switching is required.

Restrictions

- During the ServiceComb engine upgrade, the microservice and engine are intermittently disconnected, but services of running microservices are not affected. You are advised not to upgrade, restart, or change microservices when upgrading a ServiceComb engine.

- Version rollback is not supported after the upgrade.
- For details about the precautions for upgrading an exclusive ServiceComb engine from 1.x to 2.x, see **What Do I Need to Know Before Upgrading an Exclusive ServiceComb Engine?**

Upgrading a ServiceComb Engine

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click **Upgrade** in the **Operation** column of the target engine. Alternatively, click the target engine and click **Upgrade** in the **ServiceComb Engine Information** area.

Step 4 Select **Target Version** and view the version description. Determine whether to upgrade the software to this version.

Step 5 Click **OK**. If the engine status changes from **Upgrading** to **Available**, the upgrade is complete.

If the upgrade fails, click **Retry** to perform the upgrade again.

----End

3.3.7 Changing ServiceComb Engine Specifications

Specifications of exclusive ServiceComb engines can be automatically changed online. Currently, only scale-out is supported. The service will be disconnected intermittently during the change, which does not affect services. New services cannot be registered during the change.

Restrictions

The service will be disconnected intermittently during the change, which does not affect services. New services cannot be registered during the change.

Changing ServiceComb Engine Specifications

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click **More** > **Change Specifications** in the **Operation** column of the target engine. Alternatively, click the target engine and click **Change Specifications** in the **ServiceComb Engine Information** area.

Step 4 On the displayed page, select the target specifications.

Step 5 Click **Change Now**, confirm the information, and click **Submit**. If the engine status changes from **Resizing** to **Available**, the change is complete.

----End

3.3.8 Viewing ServiceComb Engine Operation Logs

In the **Operation** area, view the operation logs of a ServiceComb engine. The operation logs record user operations on the engine, which are useful for security compliance and audit.

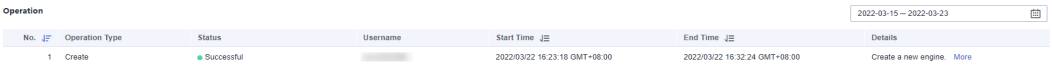
Viewing ServiceComb Engine Operation Logs

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 In the **Operation** area, view the operation logs of a ServiceComb engine.



- Click  in the upper right corner to view operation logs in a specified period.
- Click **More** in the **Details** column of a specified operation log to view details about the operation log.

----End

3.3.9 Deleting a ServiceComb Engine

You can delete a ServiceComb engine if it is no longer used. You can delete ServiceComb engines in the following states:

- Available
- Unavailable
- Creation failed
- Resizing failed
- Upgrade failed
- Unknown

Restrictions

- Deleted engines cannot be restored. Exercise caution when performing this operation.
- For engine 1.x, if the cse_admin_trust agency is missing, deleting the engine will cause residual DNS, VPC, and security group resources on the tenant side. You need to delete them by yourself.

Deleting a ServiceComb Engine

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click **Delete** in the **Operation** column of the target engine. Alternatively, click the target engine and click **Delete** in the **ServiceComb Engine Information** area.

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

NOTE

If the deletion fails, click **Force Delete**.

----End

3.4 Viewing Microservice Running Metrics Through the Microservice Dashboard

You can view metrics related to microservices through the dashboard in real time. Based on abundant and real-time dashboard data, you can take corresponding governance actions for microservices.

Restrictions

- This function is supported by ServiceComb engine 1.x and 2.4.0 and later versions.
- If a microservice application is deployed on ServiceStage, you need to configure the microservice engine during application deployment. The application automatically obtains the service registry and discovery address, configuration center address, and dashboard address. You do not need to configure the monitor address.
- If the microservice application is locally started and registered with the ServiceComb engine, manually configure the monitor address before using the dashboard.

  For details, see **Using Dashboard**.
- When the Spring Cloud Huawei framework is used for access, the median latency, 90th latency, and 99th latency cannot be viewed on the dashboard.

Viewing Microservice Running Metrics

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Dashboard**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 On the **Dashboard** page, select the target application. Enter a microservice name in the search box to search for the microservice. The running metrics of the microservice are displayed.

Click **View Diagram** to view the description of operating metrics.

Step 7 Select a sorting order to sort the filtered microservices.

----End

3.5 Managing Microservices

3.5.1 Viewing an Application

The **Application List** tab displays all applications of the current ServiceComb engine. You can search for the target application by application name, or filter applications by environment.

Viewing the Application List

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

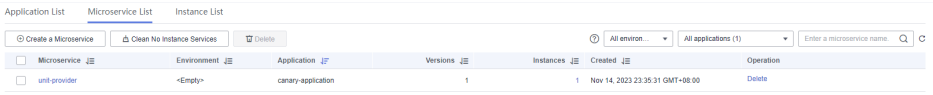
Step 6 Click **Application List** to view details about all applications of the current account under the engine.

You can search for the target application by application name, or filter applications by environment.

----End

3.5.2 Microservice Management

On the **Microservice List** tab, you can create, view, delete, dynamically configure, and dark launch microservices, and clean microservices without instances.



Creating a Microservice

Create a microservice for testing or restoring the microservice that is deleted by mistake. If a microservice that is automatically registered with ServiceComb is deleted by mistake, you can manually create a microservice to restore the deleted microservice. The microservice name, application, version, and environment must be the same as those of the original microservice.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Choose **Microservice List** > **Create a Microservice** and set microservice parameters by referring to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Microservice	Microservice name, for example, **myServiceName**. Enter 1 to 128 characters. Start and end with a digit or letter. Only use digits, letters, and special characters (_-.). The special characters cannot be used consecutively.

Parameter	Description
*Application	Name of the application to which the microservice belongs. Microservices are isolated by applications. Enter 1 to 160 characters. Start and end with a digit or letter. Only use digits, letters, and special characters (_-.). The special characters cannot be used consecutively.
*Version	Microservice version. The default value is **1.0.0**. The microservice version is in the format of X.Y.Z or X.Y.Z.B, where X, Y, Z, and B are digits and range from 0 to 32767. The value contains 3 to 46 characters.
*Environment	Environment where the microservice is located to isolate microservice data, including the version and instance.
Description	Microservice description.

Step 7 Click **OK**.

Once the microservice is created, it will be displayed in **Microservice List**.

----End

Viewing the Microservice List

The microservice list displays all microservices of the current ServiceComb engine. You can search for the target microservice by microservice name, or filter microservices by environment and application.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Microservice List** to view all microservices of the current account under the engine.

You can search for the target microservice by microservice name, or filter microservices by environment and application.

----End

Viewing Microservice Details

On the microservice details page, you can view the instance list, called services, calling services, dynamic configuration, and service contract. You can also perform **Dynamic Configuration** and **Dark Launch** on microservice-level configurations.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the microservice to be viewed in **Microservice List**. On the displayed page, view the instance list, called services, calling services, dynamic configuration, dark launch, and service contract.

----End

Cleaning Versions Without Instances

Delete microservice versions with no running instances to optimize resource utilization, improve governance efficiency, and ensure system stability.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Choose **Microservice List** > **Clean No Instance Services**. Select the microservice version without instances to be cleaned.

You can search for the target microservice by microservice name, or filter microservices by environment and application.

Step 7 Click **OK**.

----End

Dynamic Configuration

Create, edit, disable, and delete microservice-level configurations, and view historical versions of microservice-level configurations.

Configuration items are stored in plaintext. Do not include sensitive data.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Microservice List** and click a microservice.

Step 7 Choose **Dynamic Configuration**. On the **Dynamic Configuration** tab, perform the following operations.

Operation	Procedure
Create a configuration item	See **Creating a Configuration for ServiceComb Engine 2.x**. **Microservice-level** is selected for **Configuration Range** and **Microservices** is set to the current microservice.
View historical versions	Click **View Historical Version** in the **Operation** column of the target configuration item.

Operation	Procedure
Disable a configuration item	1. In the **Operation** column of the target configuration item, choose **More** > **Disable**. 2. Click **OK**.
Modify a configuration item	1. Click **Edit** in the **Operation** column corresponding to the target configuration item. 2. On the configuration details page, click **Edit**. 3. On the **Configuration Details** tab, enter the new configuration. 4. Click **Save**.
Delete a configuration item	1. In the **Operation** column of the target configuration item, choose **More** > **Delete**. 2. Click **OK**.

----End

Dark Launch

In dark launch, new features are tested in a selected group of users. When the features become mature and stable, they are released to all users. This ensures the smooth feature rollout.

- For microservices developed based on the ServiceComb Java Chassis framework, add dependency **darklaunch** or **handler-router** to POM and add **servicecomb.router.type=router** to the **microservice.yaml** configuration file.
- For microservices developed based on the Spring Cloud Huawei framework, add dependency **spring-cloud-starter-huawei-router** to POM.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 In the microservice list, click a microservice. On the displayed page, choose **Dark Launch**.

Step 7 Click **Add Launch Rule**.

- To add a launch rule by **Weight**:
  a. Click **Weight**.
  b. Set the following parameters.

Parameter	Description
Rule Name	Name of the rule. Enter 3 to 24 characters. Start with a letter. Only use digits, letters, and special characters (.@_-).
Scope	▪ Microservice version to which the rule applies. ▪ Select **Do you want to add a customized version?** and add a new version as prompted.
Rule Configuration	Traffic allocation rate for the selected version. Traffic is evenly allocated to the selected service versions based on the configured value.

  c. Click **OK** to complete the weight rule configuration and dark launch.
- To add a launch rule by **Customization**:
  a. Click **Custom**.
  b. Set the following parameters.

Parameter	Description
Rule Name	Name of the rule. Enter 3 to 24 characters. Start with a letter. Only use digits, letters, and special characters (.@_-).
Scope	▪ Microservice version to which the rule applies. ▪ Select **Do you want to add a customized version?** and add a new version as prompted.

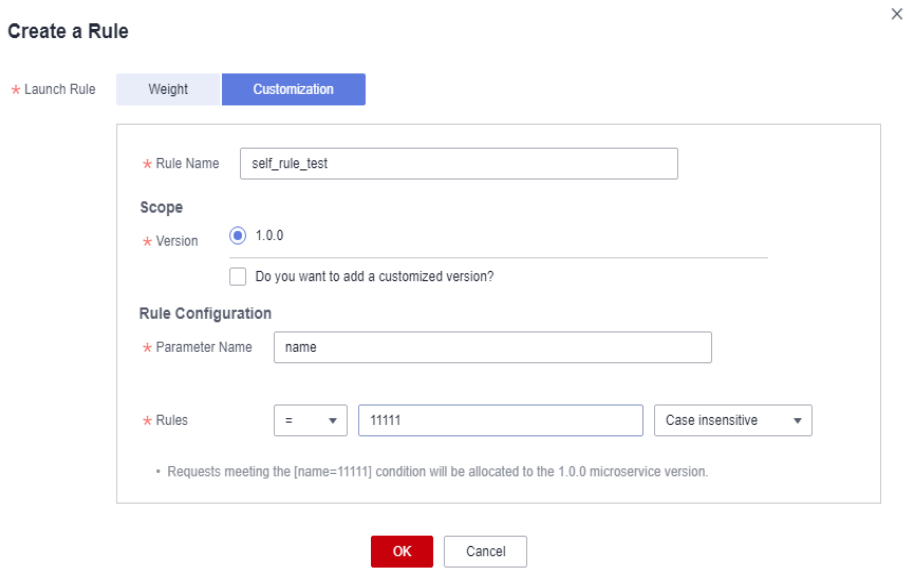
Parameter	Description
Rule Configuration	Configure the matching rule. When **darklaunch** is used to implement dark launch, this parameter configures **policyCondition**. When **handler-router** is used to implement dark launch, this parameter configures **Headers**. ■ Parameter Name Set this parameter based on the parameter name of contract or the customized key of the header. ■ Rules By selecting the matching character and the value corresponding to the key of contract or the key of the header, requests that meet the rules are allocated to the microservice version. **NOTE** ○ If **~** is selected from the drop-down list next to **Rules**, the asterisk (*) and question mark (?) in the **Rules** value can be used for fuzzy matching. The asterisk (*) indicates a character of any length, and the question mark (?) indicates one character. For example, if the rule value of **Name** is set to ***1000**, all **Name** fields ending with 1000 can be matched. ○ If **~** is not selected from the drop-down list next to **Rules**, the asterisk (*) and question mark (?) in the **Rules** value cannot be used for fuzzy matching.

c. Click **OK** to complete the customization rule configuration and dark launch.

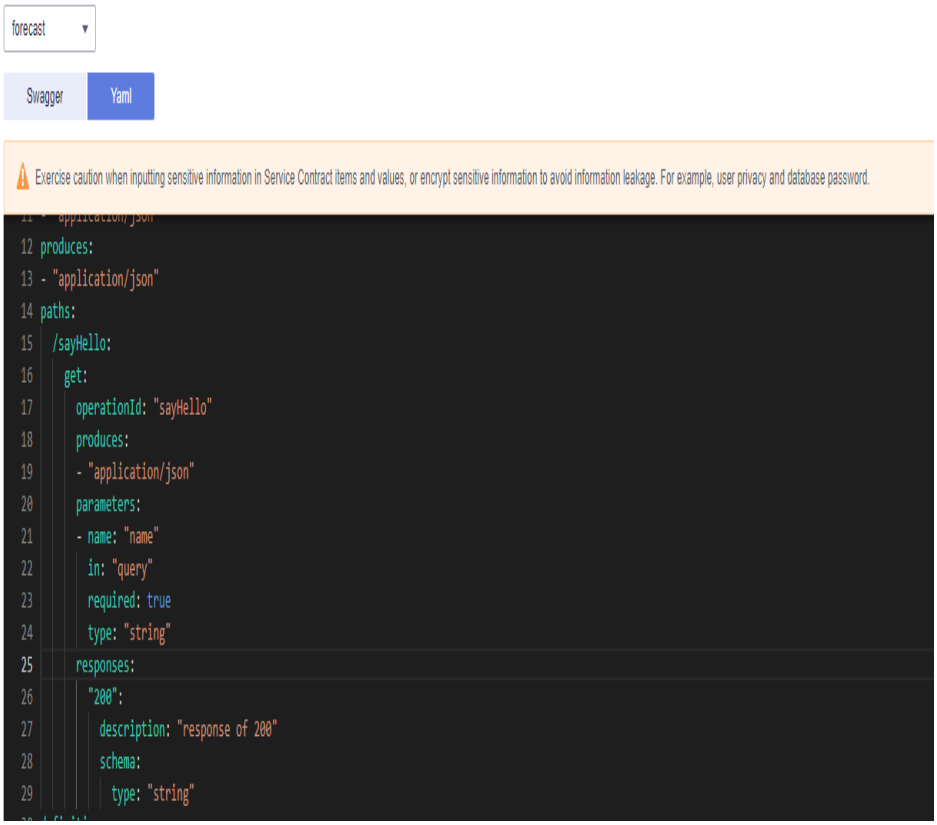
----End

Examples of delivering dark launch rules:

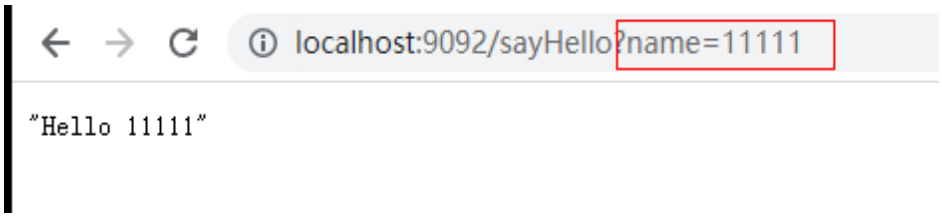
- For microservices developed based on the ServiceComb Java Chassis framework, rules are delivered based on dependency **darklaunch** on the ServiceComb engine page. You can add dark launch rules in customized mode.



This key must exist in the contract. It is possible that the server API is **String paramA**, but **paramB** is actually generated after the annotation is added. Therefore, **paramB** should be set here.

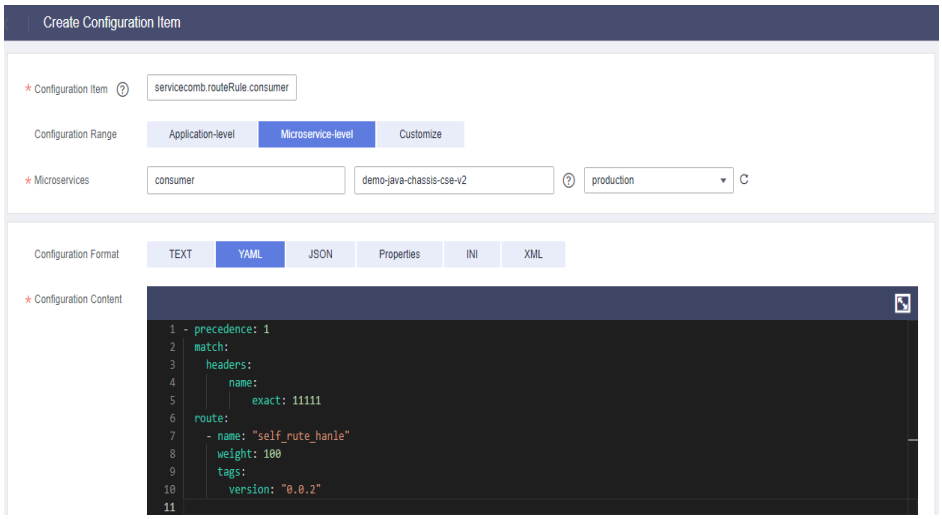


By selecting the matching character and the value corresponding to the key of contract, requests that meet the rules are allocated to the microservice version.

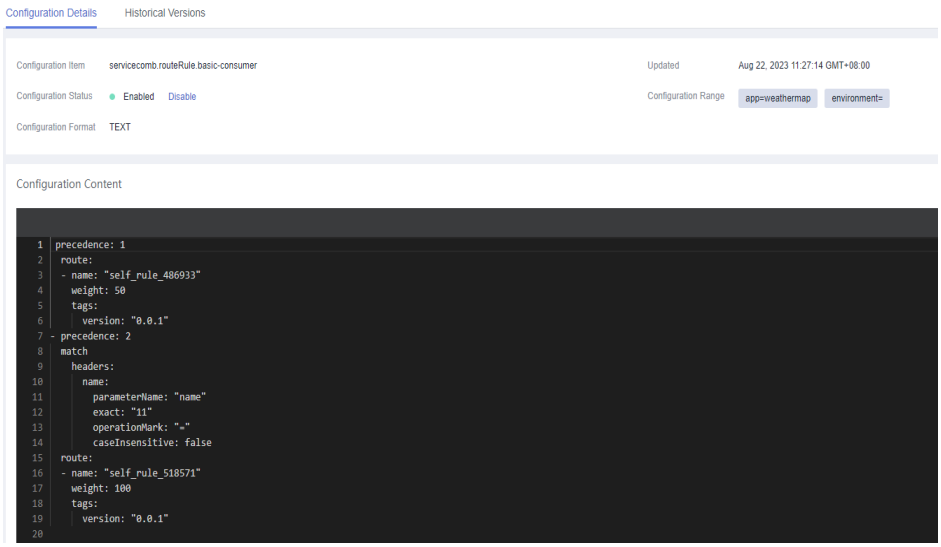


A delivered rule is as follows. The configuration item is **cse.darklaunch.policy.*${serviceName}***.

- For microservices developed based on the ServiceComb Java Chassis framework, dark launch rules that depend on **handler-router** need to be manually delivered in the configuration center. The configuration item is **servicecomb.routeRule.*${serviceName}***. The content is as follows:



- For microservices developed based on the Spring Cloud Huawei framework, dark launch rules delivered on the ServiceComb engine page are as follows:



Deleting a Microservice

Delete a microservice that is no longer used.

- After a microservice is deleted, you can restore it by referring to **Restoring Backup Data**.
- If the service to be deleted has instances, delete the instances first. Otherwise, the service will be registered again.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Microservice List**.

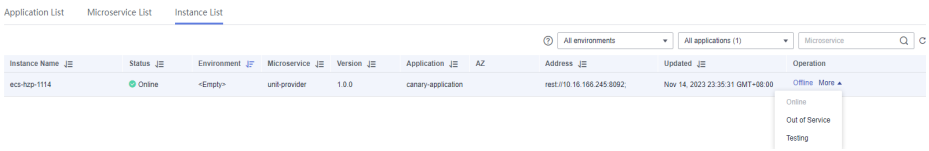
- To delete microservices in batches, select the microservices to be deleted and click **Delete** above the microservices.
- To delete one microservice, locate the row that contains the microservice to be deleted and click **Delete** in the **Operation** column.

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

----End

3.5.3 Instance Management

CSE allows you to view and change the status of all microservice instances registered with the ServiceComb engine.



Viewing the Instance List

The instance list displays all instances of the current ServiceComb engine. You can search for the target instance by microservice name, or filter instances by environment and application.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Instance List** to view all instances of the engine.

You can search for the target instance by microservice name, or filter instances by environment and application.

----End

Changing the Instance Status

Status indicates the status of a microservice instance.

The status of microservice instances synchronized by binding ServiceComb engines cannot be changed during component creation and deployment by referring to **Creating and Deploying a Component**.

The following table describes the microservice instance statuses.

Status	Description
Online	The instance is running and can provide services.
Offline	Before the instance process ends, the instance is marked as not providing services externally.
Out of Service	The instance has been registered with the ServiceComb engine and does not provide services.
Testing	The instance is in the internal joint commissioning state and does not provide services.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Catalog**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Instance List**, select the target instance, and change the instance status.

- Offline

  In the **Operation** column, click **Offline**.
- Online

  In the **Operation** column, choose **More** > **Online**.
- Out of Service

  In the **Operation** column, choose **More** > **Out of Service**.
- Testing

  In the **Operation** column, choose **More** > **Testing**.

----End

3.6 Service Scenario Governance (Applicable to ServiceComb Engine 2.x)

3.6.1 Service Scenario Governance Overview

ServiceComb engines provide unified traffic feature governance based on dynamic configurations for different microservice development frameworks, such as Spring Cloud and Java chassis. You can use the microservice governance function of CSE by introducing related governance components to the development frameworks.

ServiceComb engine governance consists of two steps: creating a service scenario and creating a governance policy. The two steps can be performed before microservice deployment for independent governance planning.

Restrictions

- Service scenario governance is applicable to ServiceComb engine 2.x.
- If you want to delete a governance policy in use, pay attention to the following information:
  - Risks: The governance policy may become invalid, which reduces the capability of the microservice system to resist abnormal traffic. When there is abnormal traffic, problems such as unbalanced call distribution and microservice avalanche may occur.
  - Precautions: Perform the operations during off-peak hours. Before the operations, ensure that there is no abnormal traffic, such as many calls and call timeout.

Governance Policies

You can configure the following policies: rate limiting, circuit breaker, retry, and bulkhead. For details, see the following table.

Policy	Description
Rate limiting	In the case of a traffic storm or predictable traffic spikes, rate limiting is performed on non-key service scenarios to prevent service and data breakdown caused by instantaneous heavy traffic.
Retry	When a service encounters a non-fatal error (such as occasional timeout), retries can be performed to prevent service failures.
Bulkhead	In the case of a large-scale concurrent traffic storm or predictable traffic impact, the concurrent traffic is controlled to prevent service and data breakdown caused by instantaneously large concurrent traffic.

Policy	Description
Circuit breaker	When the error rate of a service scenario exceeds the threshold, all requests in the service scenario will be rejected within one minute to ensure the availability of the entire service system. Then 50% of the service requests will be accepted and statistics on the service error rate will be collected until the error rate in the service scenario is reduced to a value lower than the threshold.

3.6.2 Creating a Service Scenario

You can create service scenarios based on your service process and requirements, and implement governance policies accordingly to ensure stable and efficient service running.

Prerequisites

- You have deployed an application by referring to **Creating and Deploying a Component**.
- You need to understand the API design of the microservice to be governed and create service scenarios based on the API features.

Creating a Service Scenario

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Service Scenario Governance**.

NOTE

If the ServiceComb engine version is 2.0.0 or later but earlier than 2.4.0, choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Choose **Service Scenarios** > **Create Service Scenario** and set parameters by referring to the following table.

Parameter	Description
Scenario	Enter a service scenario name. Enter up to 20 characters.
Environment	Select a microservice environment.
Application	Select the application to which the service scenario to be created belongs.
Matching Rule	You can set **Method**, **Path**, and **Headers** rules to filter requests that meet specific conditions. Only requests that meet these rules are included in the created service scenario, which facilitates centralized governance of specific types of requests. Click **Add Matching Rule** to set the request marker. • **Method**: (Mandatory) Select the method of marking the traffic request feature. The **GET**, **PUT**, **POST**, **DELETE**, and **PATCH** methods are supported. • **Path**: (Mandatory) Set features contained in the traffic request URI. • **Headers**: (Optional) Click **Add Headers Rule** and set the marker of the traffic request header.

Step 7 Click **OK**. When a service scenario is created, the configuration starting with **servicecomb.matchgroup.** is automatically generated.

- Click in the row that contains a service scenario to view the matching rule details.
- Click **Edit** in the **Operation** column of a service scenario to edit the service scenario.
- Click **Delete** in the **Operation** column of a service scenario to delete the service scenario.

----End

3.6.3 Creating a Governance Policy

A governance policy is a method used for microservice governance. Each governance policy can be bound to a service scenario. A policy cannot be bound to multiple service scenarios. Different governance policies can be bound to the same service scenario.

Prerequisites

- You have created a service scenario by referring to **Creating a Service Scenario**.
- You need to enable the dynamic configuration-based traffic feature governance function for the development framework of the microservice to be governed. If the function is not enabled, the microservice governance function can still be used, but the governance has no effects.

Creating a Governance Policy

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Service Scenario Governance**.

NOTE

If the ServiceComb engine version is 2.0.0 or later but earlier than 2.4.0, choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Go to the **Governance Policy** page and click **Create Governance Policy**.

Step 7 Select a governance mode, click **Create Policy**, and set parameters.

- Rate limiting

Parameter	Description
Policy	Enter a governance policy name. Enter up to 32 characters.
Service Scenarios	Set the service scenarios to which the governance policy applies. – Click **Select Service Scenario** and select the created service scenario. – Click **Create Service Scenario**. For details, see **Creating a Service Scenario**.
Requests per Unit	Set the number of requests and time segment. When the number of requests sent by the rate limiting object to the current service instance within the specified period exceeds the specified value, the excess requests are limited and error code 429 is returned.

- Retry

Parameter	Description
Policy	Enter a governance policy name. Enter up to 32 characters.
Service Scenarios	Set the service scenarios to which the governance policy applies. – Click **Select Service Scenario** and select the created service scenario. – Click **Create Service Scenario**. For details, see **Creating a Service Scenario**.
Response Error Code	Enter response error codes to define the error types that trigger retries.
Retry Attempts	Set the number of retries.
Retry Interval	Select a retry policy. – **Fixed**: The retry interval is fixed. For example, if the retry interval is set to 500 ms, the interval between two consecutive retries remains 500 ms regardless of the number of retries. – **Exponential**: Each retry interval is randomly determined based on an algorithm. Generally, the retry interval increases exponentially with the number of retries. For example, the retry interval may be 100 ms for the first retry, 200 ms for the second retry, and 400 ms for the third retry.
Interval Duration	Set the retry interval duration. – If **Retry Interval** is set to **Fixed**, set the fixed retry interval. The value is an integer ranging from 1 to 60,000, in seconds or milliseconds. – If **Retry Interval** is set to **Exponential**, set the retry benchmark time. The value ranges from 1 to 60,000, in seconds or milliseconds.

- Bulkhead

Parameter	Description
Policy	Enter a governance policy name. Enter up to 32 characters.
Service Scenarios	Set the service scenarios to which the governance policy applies. – Click **Select Service Scenario** and select the created service scenario. – Click **Create Service Scenario**. For details, see **Creating a Service Scenario**.

Parameter	Description
Max. Concurrency	Set the maximum number of concurrent requests based on the actual service processing capability of the system.
Block Duration	When the number of concurrent requests exceeds the maximum, the requests are discarded after the blocking duration. The value is an integer ranging from 1 to 300,000, in seconds or milliseconds.

- Circuit breaker

Parameter		Description
Policy		Enter a governance policy name. Enter up to 32 characters.
Service Scenarios		Set the service scenarios to which the governance policy applies. – Click **Select Service Scenario** and select the created service scenario. – Click **Create Service Scenario**. For details, see **Creating a Service Scenario**.
Coverage	Sliding Window Type	Select a sliding window type. Sliding window: range of request call times. The system monitors calls within this range to trigger the circuit breaker when they exceed the baseline. – **Time**: The window range is determined by time. – **Requests**: The window range is determined by the number of requests.
	Sliding Window Size	Set the size of the sliding window. – If **Sliding Window Type** is set to **Time**, the calls in the last n seconds or minutes are recorded and collected. – If **Sliding Window Type** is set to **Requests**, the latest n calls are recorded and collected. n is the size of the sliding window.
	Calls Baseline	Set the baseline of the number of calls, that is, the minimum number of calls required for collecting statistics on the call error rate. For example, if **Calls Baseline** is set to **10**, at least 10 call must be recorded to collect statistics on the error rate.

Parameter		Description
Triggers	Error Threshold	Percentage of call errors. This parameter is valid when **Set circuit breaker to trigger at an error threshold** is selected. When the call error rate is greater than or equal to the error threshold, circuit breaker occurs and response code 429 is returned.
	Slow Request Ratio	This parameter is valid when **Set circuit breaker to trigger at a specific request speed and ratio** is selected. Set the following parameters: – **Slow Speed**: defines the slow request threshold. If the response time of a request exceeds the threshold, the request is a slow request. – **Slow Threshold**: When the specified slow request ratio is reached, circuit breaker occurs and response code 429 is returned.

Step 8 Click **Create** to make the governance policy take effect.

In the governance policy list, click  in the row where the service scenario is located:

- Click **Enable** in the **Operation** column of a governance policy to enable the policy.
- Click **Disable** in the **Operation** column of a governance policy to disable the policy.
- Click **Edit** in the **Operation** column of a governance policy to edit the policy.
- Click **Delete** in the **Operation** column of a governance policy to delete the policy.

----End

3.7 Microservice Governance (Applicable to ServiceComb Engine 1.x and 2.4.0+)

3.7.1 Microservice Governance Overview

If an application is developed using the microservice framework, the microservice is automatically registered with the corresponding ServiceComb engine after the application is managed and started. You can perform service governance on the engine console.

After a microservice is deployed, you can govern the request traffic, troubleshooting, and load balancing of the microservice based on the microservice running status.

Governance Policies

You can configure the following policies: load balancing, rate limiting, fault tolerance, service degradation, circuit breaker, fault injection, and black and white list. For details, see the following table.

Name	Description
Load balancing	• Application scenario Generally, multiple instances are deployed for a microservice. Load balancing controls the policy for a microservice consumer to access multiple instances of a microservice provider to balance traffic. It includes polling, random, response time weigh, and session stickiness. • For details about the configuration example of the governance policy and how to add dependencies to POM, see **Load Balancing**.
Rate limiting	• Application scenario This policy controls the number of requests for accessing microservices to prevent the system from being damaged due to traffic impact. • For details about the configuration example of the governance policy and how to add dependencies to the POM, see **Rate Limiting**.
Service degradation	• Application scenario When a microservice invokes other microservices, the default value is forcibly returned or an exception is thrown instead of sending the request to the target microservice. In this way, the access to the target microservice is shielded and the pressure on the target microservice is reduced. • For details about the configuration example of the governance policy and how to add dependencies to the POM, see **Service Degradation**.
Fault tolerance	• Application scenario If an exception occurs when a microservice consumer accesses a provider, for example, the instance network is disconnected, the request needs to be forwarded to another available instance. Fault tolerance is often referred to as retry. • For details about the configuration example of the governance policy and how to add dependencies to POM, see **Fault Tolerance**.

Name	Description
Circuit breaker	• Application scenario If an exception occurs when a microservice consumer accesses a provider, for example, the instance network is disconnected or the request times out, and the exception accumulates to a certain extent, the consumer needs to stop accessing the provider and return an exception or a default value to prevent the avalanche effect. Automatic circuit breaker is supported, which determines a circuit breaker according to the error rate. • For details about the configuration example and how to add dependencies to the POM, see **Circuit Breaker**.
Fault injection	This policy applies only to microservices accessed through Java chassis. • Application scenario Fault injection can simulate an invoking failure, which is mainly used for function verification and fault scenario demonstration. • Governance of microservices connected to the Java Chassis development framework. For details about the configuration example of the governance policy and how to add dependencies to POM, see **Fault Injection**.
Blacklist/ Whitelist	This policy applies only to microservices accessed through Java chassis. • Application scenario Based on the public key authentication mechanism, the ServiceComb engine provides the blacklist and whitelist functions. The blacklist and whitelist can be used to control which services can be accessed by microservices. • Governance of microservices accessed through Java chassis The blacklist and whitelist take effect only after public key authentication is enabled. For details, see **Configuring Blacklist and Whitelist**.

3.7.2 Configuring a Load Balancing Policy

Generally, multiple instances are deployed for a microservice. Load balancing controls the policy for a microservice consumer to access multiple instances of a microservice provider to balance traffic. It includes polling, random, response time weigh, and session stickiness.

Prerequisites

You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.

Configuring Load Balancing

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Choose **Load Balancing**.

Step 7 Click **New**. Select the microservices to be governed and select a proper load balancing policy. For details, see the following table.

Figure 3-1 Configuring load balancing (for microservices accessed through Spring Cloud)

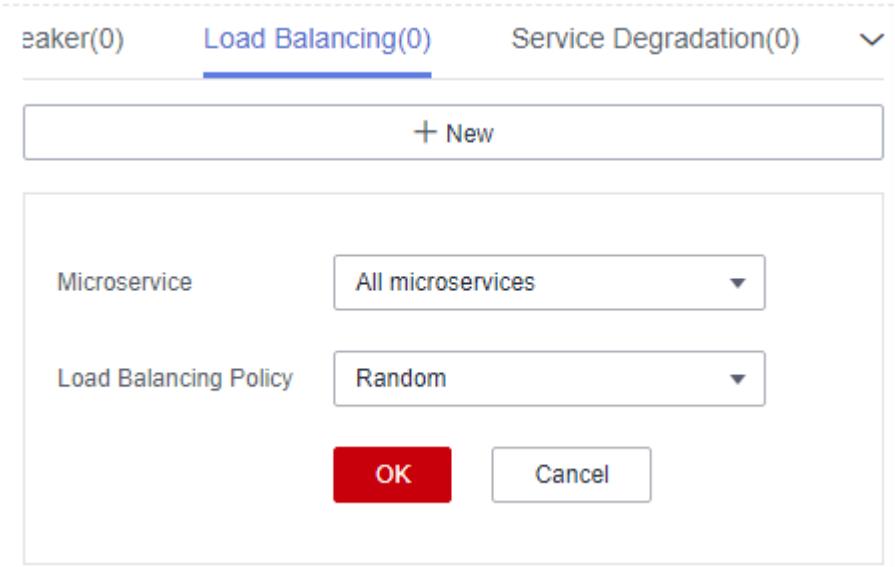
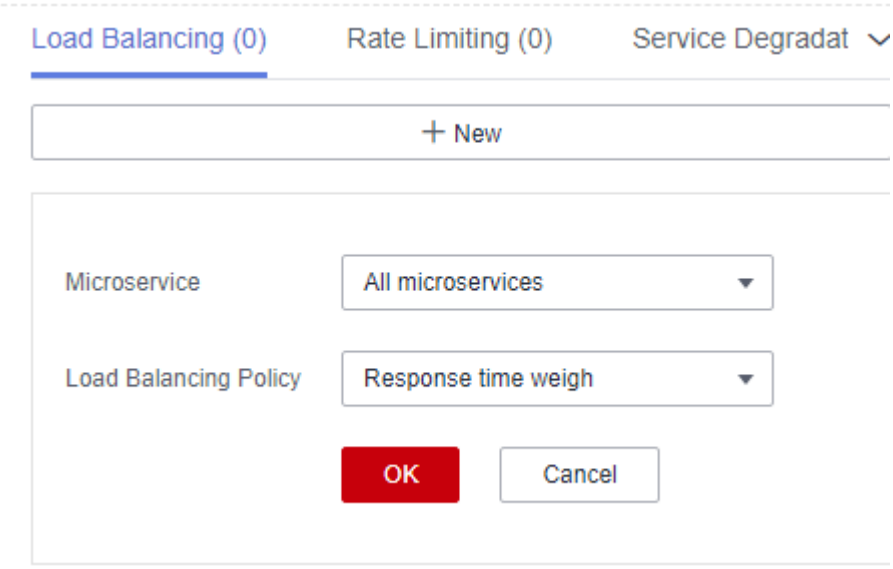


Figure 3-2 Configuring load balancing (for microservices accessed through Java chassis)



Policy	Description
Round robin	Supports routes according to the location information about service instances.
Random	Provides random routes for service instances.
Response time weigh	This configuration applies to microservices accessed through Java chassis. Provides weight routes with the minimum active number (latency) and supports service instances with slow service processing in receiving a small number of requests to prevent the system from stopping response. This load balancing policy is suitable for applications with low and stable service requests.
Session stickiness	This configuration applies to microservices accessed through Java chassis. Provides a mechanism on the load balancer. In the specified session stickiness duration, this mechanism allocates the access requests related to the same user to the same instance. • **Stickiness Duration**: time limit for keeping a session. The value ranges from 0 to 86400, in seconds. • **Failures**: number of access failures. The value ranges from 0 to 10. If the upper limit of failures or the session stickiness duration exceeds the specified values, the microservice stops accessing this instance.

Step 8 Click **OK**.

----End

3.7.3 Configuring a Rate Limiting Policy

This policy controls the number of requests for accessing microservices to prevent the system from being damaged due to traffic impact.

Prerequisites

You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.

Configuring Rate Limiting

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Click **Rate Limiting**.

Step 7 Click **New**. The following table describes configuration items of rate limiting.

Figure 3-3 Configuring rate limiting (for microservices accessed through Spring Cloud)

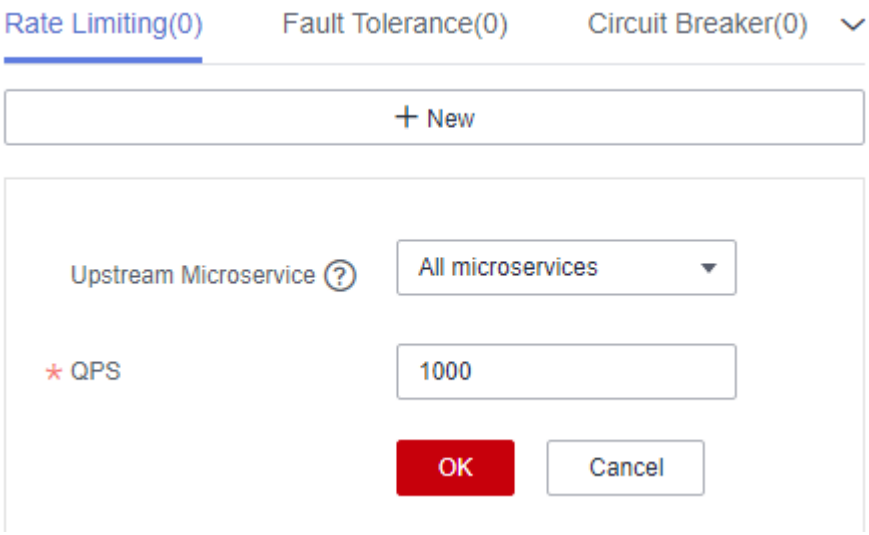
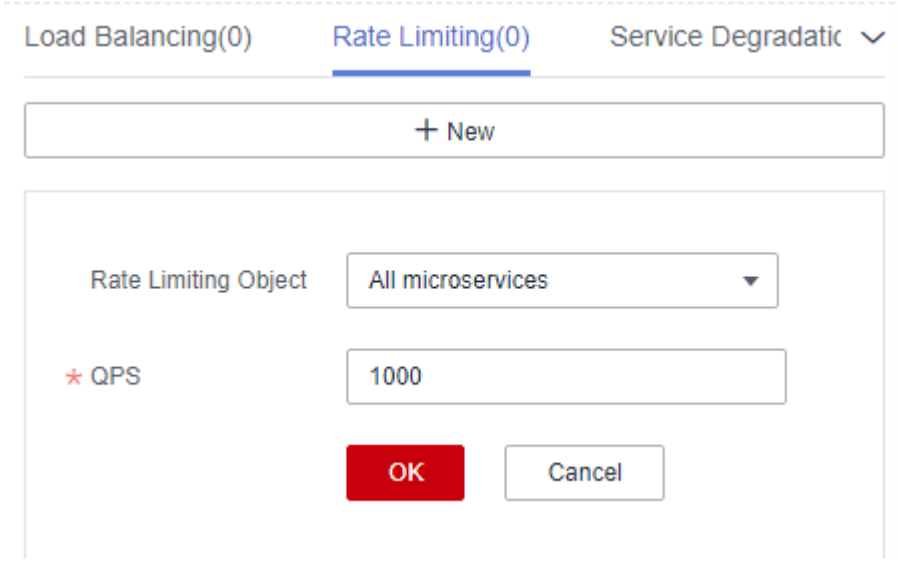


Figure 3-4 Configuring rate limiting (for microservices accessed through Java chassis)



Parameter	Description
Rate Limiting Object	This configuration applies to microservices accessed through Java chassis. Scope for the rate limiting rule to take effect.

Parameter	Description
Upstream Microservice	This configuration applies to microservices accessed through Spring Cloud. In the microservice architecture, each microservice works with other microservices to implement service functions. Select another microservice that invokes the current microservice.
QPS	Requests generated per second. When the number of requests sent by the rate limiting object to the current service instance exceeds the specified value, the current service instance no longer accepts requests from the rate limiting object. The value ranges from 1 to 99999. If a microservice has three instances, the rate limiting of each instance is set to 2700 QPS, then the total QPS is 8100. In this case, rate limiting is triggered only when the QPS exceeds 8100.

Step 8 Click **OK**.

----End

3.7.4 Configuring a Service Degradation Policy

When a microservice invokes other microservices, the default value is forcibly returned or an exception is thrown instead of sending the request to the target microservice. In this way, the access to the target microservice is shielded and the pressure on the target microservice is reduced.

Prerequisites

You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.

Configuring Service Degradation

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Click **Service Degradation**.

Step 7 Click **New** and select a proper policy. The following table describes the configuration items of service degradation.

Figure 3-5 Configuring service degradation (for microservices accessed through Spring Cloud)

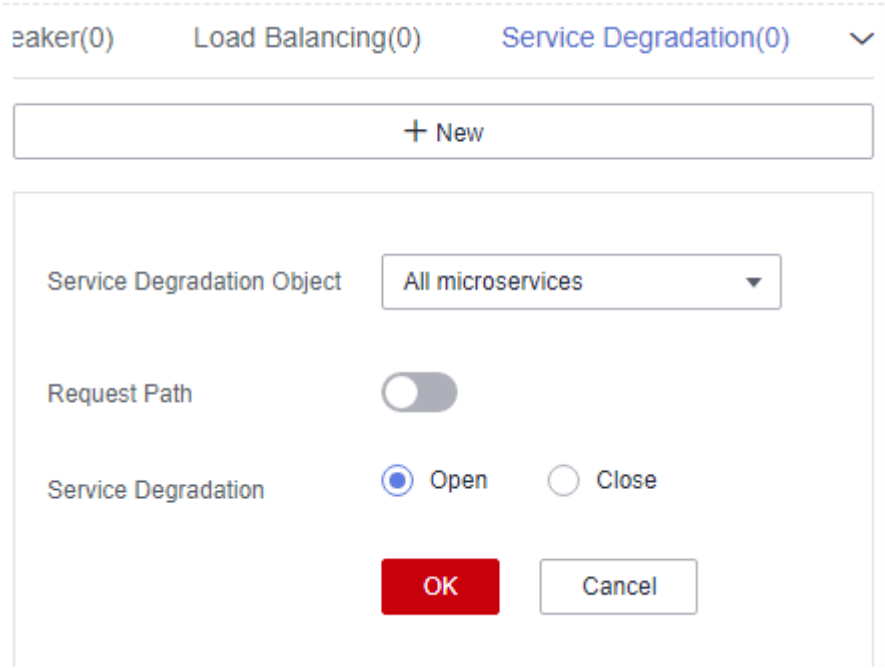
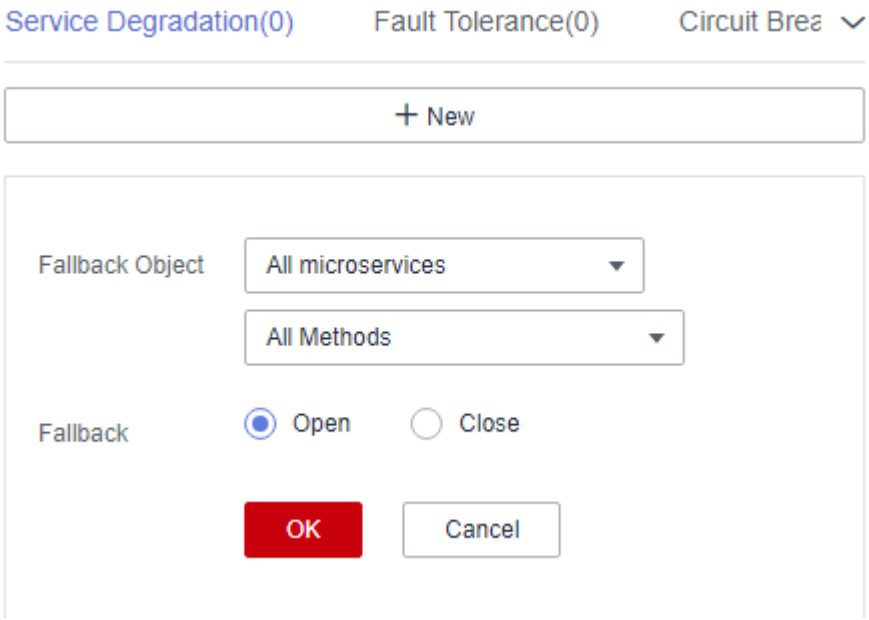


Figure 3-6 Configuring service degradation (for microservices accessed through Java chassis)



Parameter	Description
Fallback Object	Microservice to be degraded. When microservices are accessed through Java chassis, one or more methods in the selected microservice are used as the object.
Request Path	This configuration applies to microservices accessed through Spring Cloud. You can click  and set **Method** (such as GET, POST, and PUT), **Path** (request path), and **Headers** to filter requests.
Fallback	• **Open**: enables service degradation and sets parameters. • **Close**: disables service degradation.

Step 8 Click **OK**.

----End

3.7.5 Configuring a Fault Tolerance Policy

If an exception occurs when a microservice consumer accesses a provider, for example, the instance network is disconnected, the request needs to be forwarded to another available instance. Fault tolerance is often referred to as retry.

Prerequisites

You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding

service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.

Configuring Fault Tolerance

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Click **Fault Tolerance**.

Step 7 Click **New** and select a proper policy. The following table describes the configuration items of fault tolerance.

Figure 3-7 Configuring fault tolerance (for microservices accessed through Spring Cloud)

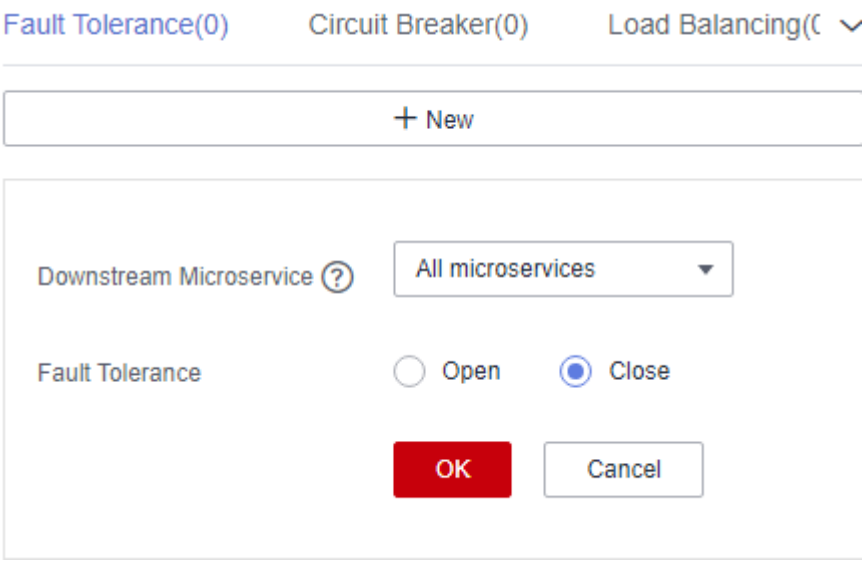
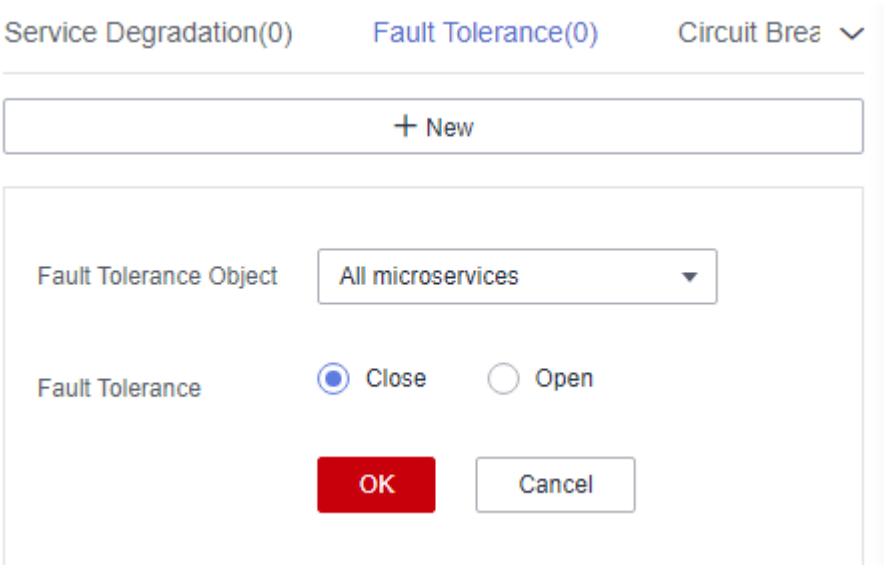


Figure 3-8 Configuring fault tolerance (for microservices accessed through Java chassis)



Parameter	Description
Downstream Microservice	This configuration applies to microservices accessed through Spring Cloud. Configure fault tolerance for the microservice to invoke the downstream microservice. You can select a downstream microservice from the drop-down list.
Fault Tolerance Object	This configuration applies to microservices accessed through Java chassis. Microservice or method that the application relies on.
Fault Tolerance	**Open**: The system processes a request sent to the fault tolerance object based on the selected fault tolerance policy when the request encounters an error. **Close**: The system waits until the timeout interval expires and then returns the failure result even though the service request fails to be implemented.

Parameter	Description
FT Policy	This parameter is mandatory when **Fault Tolerance** is set to **Open**. For microservices accessed through Spring Cloud, set the following parameters: • Number of attempts to the same microservice instance • Number of attempts to the new microservice instance For microservices accessed through Java chassis, set the following parameters: • Failover The system attempts to connect to different servers. • Failfast The system does not attempt to connect to a server. After a request fails, a failure result is returned immediately. • Failback The system attempts to connect to the same server. • custom – **Request Attempts of One Server**: number of attempts to connect to the same server – **Request Attempts of a New Server**: number of attempts to connect to another server

Step 8 Click **OK**.

----End

3.7.6 Configuring a Circuit Breaker Policy

If an exception occurs when a microservice consumer accesses a provider, for example, the instance network is disconnected or the request times out, and the exception accumulates to a certain extent, the consumer needs to stop accessing the provider and return an exception or a default value to prevent the avalanche effect.

Prerequisites

You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.

Configuring Circuit Breaker

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Click **Circuit Breaker**.

Step 7 Click **New** and select a proper policy. The following table describes the configuration items of circuit breaker.

Figure 3-9 Configuring circuit breaker (for microservices accessed through Spring Cloud)

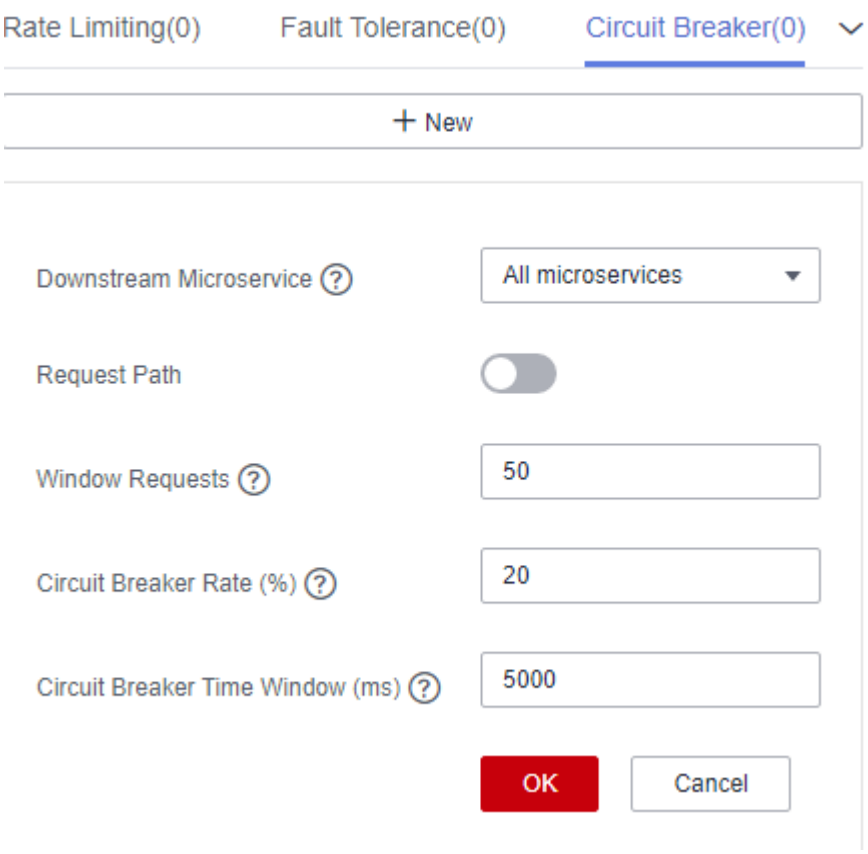
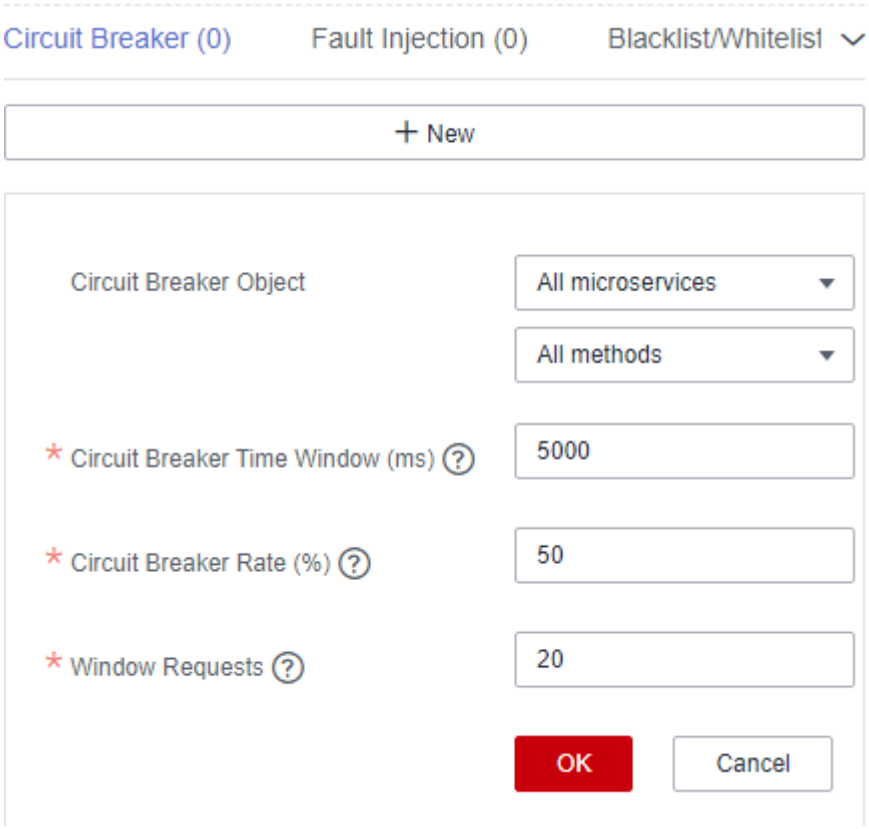


Figure 3-10 Configuring circuit breaker (for microservices accessed through Java chassis)



Parameter	Description
Downstream Microservice	This configuration applies to microservices accessed through Spring Cloud. Configure circuit breaker for the microservice to invoke the downstream microservice. You can select a downstream microservice from the drop-down list.
Circuit Breaker Object	This configuration applies to microservices accessed through Java chassis. Microservice or method called by the application.
Request Path	This configuration applies to microservices accessed through Spring Cloud. You can click  and set **Method** (such as GET, POST, and PUT), **Path** (request path), and **Headers** to filter requests.
Circuit Breaker Time Window	Circuit breaker duration. No response is sent within the time window. Unit: ms.

Parameter	Description
Circuit Breaker Rate	Percentage of failed requests within the specified number of window requests. For example, if **Window Requests** is set to 20 and **Circuit Breaker Rate** is set to 50%, circuit breaker is triggered when 10 of the 20 (50%) requests fail.
Window Requests	Number of requests received within the time window. Circuit breaker is triggered only when **Circuit Breaker Rate** and **Window Requests** both reach their thresholds.

Step 8 Click **OK**.

----End

3.7.7 Configuring a Fault Injection Policy

Fault injection can simulate an invoking failure, which is mainly used for function verification and fault scenario demonstration. This policy applies only to microservices accessed through Java chassis.

Prerequisites

You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.

Configuring Fault Injection

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Click **Fault Injection**.

Step 7 Click **New** and select a proper policy. The following table describes the configuration items of fault injection.

Figure 3-11 Configuring fault injection (delayed)

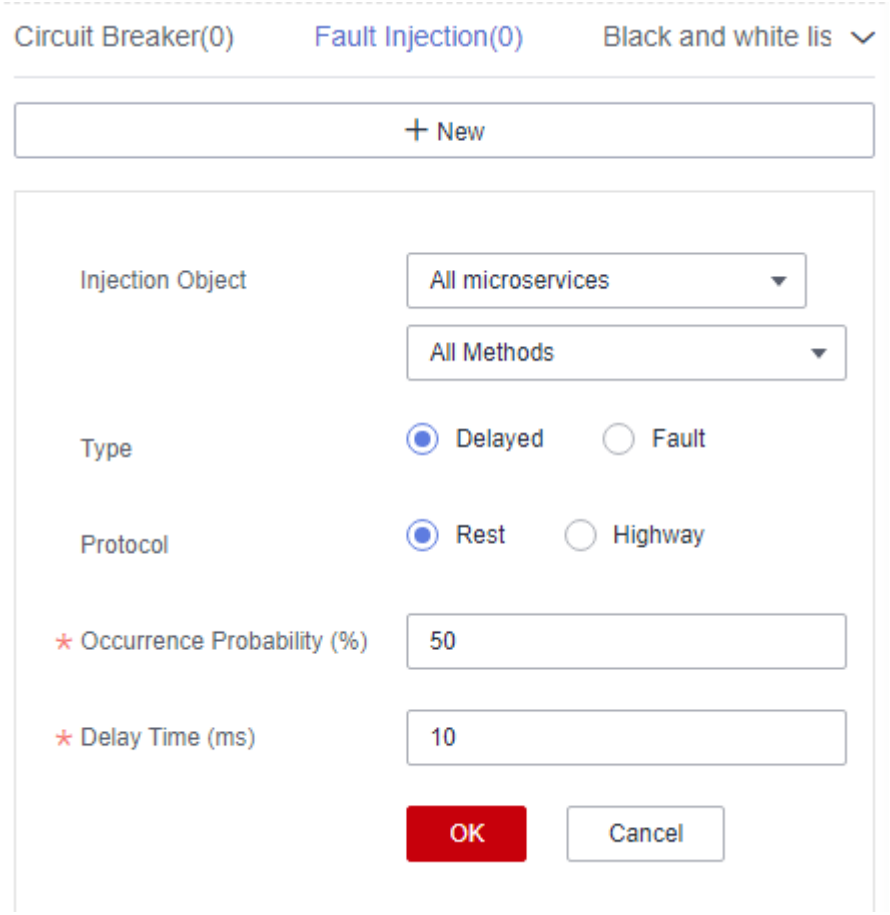
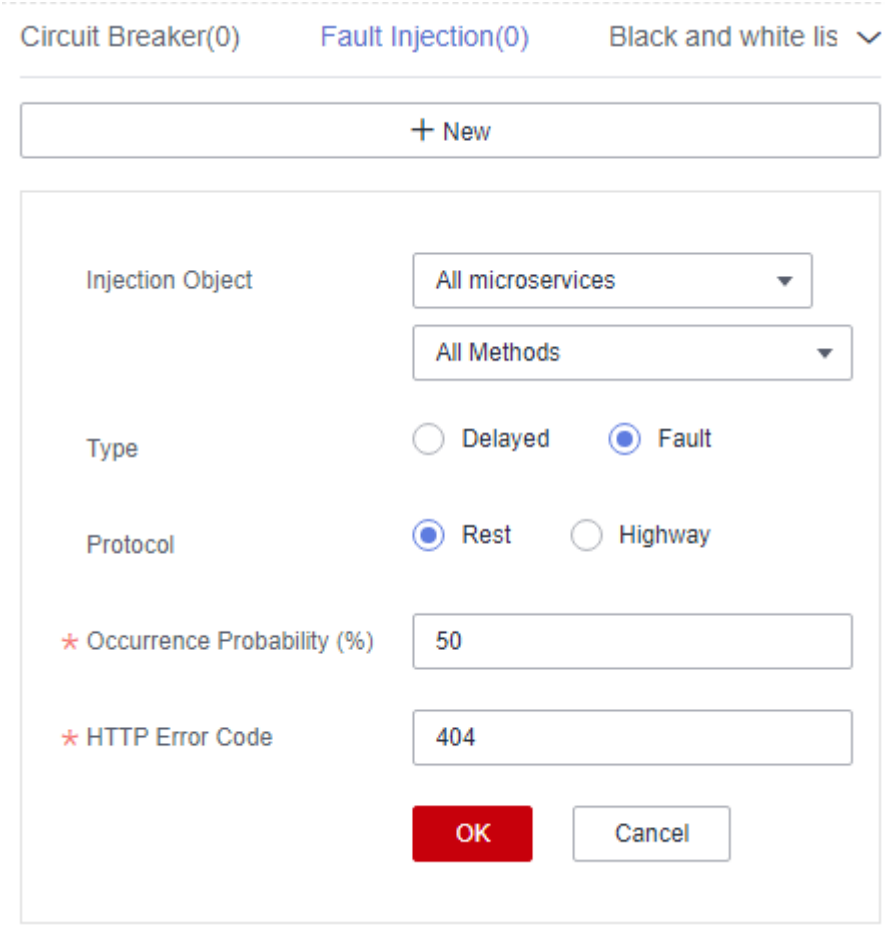


Figure 3-12 Configuring fault injection (fault)



Parameter	Description
Injection Object	Microservices for which fault injection is required. You can specify a method for this configuration item.
Type	Type of the fault injected to the microservice. • Delayed • Fault
Protocol	Protocol for accessing the microservice when latency or fault occurs. • Rest • Highway
Occurrence Probability	Probability of latency or fault occurrence.
Delay Time	Duration of the latency during microservice access. This parameter is required when **Type** is set to **Delayed**.

Parameter	Description
HTTP Error Code	HTTP error code during microservice access. This parameter is required when **Type** is set to **Fault**. This error code is an HTTP error code.

Step 8 Click **OK**.

----End

3.7.8 Configuring Blacklist and Whitelist

Based on the public key authentication mechanism, the ServiceComb engine provides the blacklist and whitelist functions. The blacklist and whitelist can be used to control which services can be accessed by microservices. This policy applies only to microservices accessed through Java chassis.

Prerequisites

- You have created a microservice by referring to **Creating a Microservice**. After the microservice starts, the service instance is registered with the corresponding service based on the configurations in the YAML file. If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.
- The blacklist and whitelist take effect only after public key authentication is enabled. For details, see **Configuring Public Key Authentication**.

Configuring Blacklist and Whitelist

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Microservice Governance**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

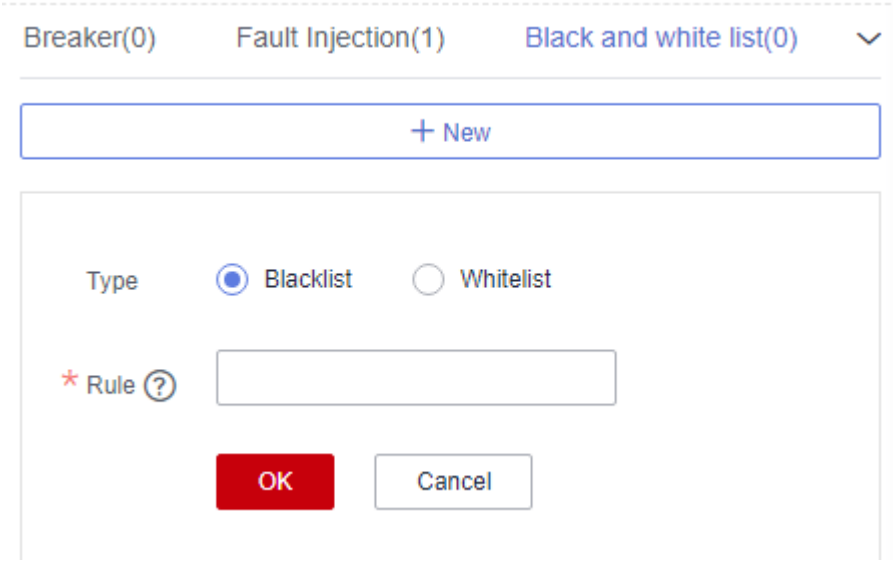
NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target microservice. Click **Black and white list**.

Step 7 Click **New** to add a blacklist or whitelist for the application. The following table describes configuration items of blacklist and whitelist.

Figure 3-13 Configuring blacklist and whitelist



Parameter	Description
Type	• **Blacklist**: Microservices that match the matching rule are not allowed to access the current service. • **Whitelist**: Microservices that match the matching rule are allowed to access the current service.
Rule	Use a regular expression. For example, if **Rule** is set to **data***, services whose names start with **data** in the blacklist are not allowed to access the current service, or services whose names start with **data** in the whitelist are allowed to access the current service.

Step 8 Click **OK**.

----End

3.7.9 Configuring Public Key Authentication

Public key authentication is a simple and efficient authentication mechanism between microservices provided by CSE. Its security is based on the reliable interaction between microservices and the service center. That is, the authentication mechanism must be enabled between microservices and the service center. The procedure is as follows:

1. When a microservice starts, a key pair is generated and the public key is registered with the service center.
2. Before accessing the provider, the consumer uses its own private key to sign a message.
3. The provider obtains the public key of the consumer from the service center and verifies the signed message.

To enable public key authentication, perform the following steps:

1. Enable public key authentication for both the consumer and provider.

```
servicecomb:
  handler:
    chain:
      Consumer:
        default: auth-consumer
      Provider:
        default: auth-provider
```

2. Add the following dependency to the **pom.xml** file:

```
<dependency>
    <groupId>org.apache.servicecomb</groupId>
    <artifactId>handler-publickey-auth</artifactId>
  </dependency>
```

3.8 Configuration Management (Applicable to ServiceComb Engine 2.x)

3.8.1 Creating a Configuration for ServiceComb Engine 2.x

ServiceComb engines define a configuration mechanism that is irrelevant to development frameworks. A configuration item consists of a key, label, and value. The label is used to identify whether a configuration item belongs to global configuration or microservice configuration. The label can also indicate the value type.

Restrictions

- Configuration items are stored in plaintext. Do not include sensitive data.
- When the configuration item quota specified by the engine specifications is about to be used up, the engine allows new configuration items that exceed the remaining quota to be created to ensure capacity availability. Expand the capacity of the engine as soon as possible to prevent configuration creation failures.
- When editing or deleting a configuration item used by a microservice, the microservice may fail to read the configuration or read an incorrect configuration, causing service exceptions. Therefore, back up the configuration before modifying or deleting configuration items.

Creating an Application-Level Configuration

Associates the new configuration with an application, and adds the application name and environment label.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.

- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Create Configuration Item** and set parameters by referring to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Configuration Item	Enter a configuration item. The configuration item is the global ID of the configuration. In the coding phase, the configuration item is used to index and operate the configuration. You are advised to use the Java package naming rule (for example, **cse.service.registry.address**) to ensure the readability and uniqueness of the configuration. Configuration items starting with **servicecomb.matchGroup.** cannot be created during application-level configuration creation. Such configuration items conflict with the configuration generated during service scenario governance creation, so the service scenario cannot be displayed.
Configuration Scope	Select **Application-level**.
*Application	1. Select or enter an application name. 2. Select an environment.
Configuration Format	Select a configuration format. Common configuration formats such as TEXT, YAML, JSON, Properties, INI and XML can be edited online. The default value is **TEXT**.
*Configuration Content	Enter the configuration content.
Enable Configuration	Determine whether to enable the configuration item. • **Enable now**: The configuration item takes effect immediately once being created. • **Not Enabled**: The configuration item does not take effect.

Step 7 Click **Create Now** to enable the configuration item.

----End

Creating a Microservice-Level Configuration

Associates the new configuration with a microservice, and adds the microservice name, application name, and environment.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Create Configuration Item** and set parameters by referring to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Configuration Item	Enter a configuration item. The configuration item is the global ID of the configuration. In the coding phase, the configuration item is used to index and operate the configuration. You are advised to use the Java package naming rule (for example, **cse.service.registry.address**) to ensure the readability and uniqueness of the configuration.
Configuration Scope	Select **Microservice-level**.
*Microservice	1. Select or enter a microservice name. 2. Select or enter an application name. 3. Select an environment.
Configuration Format	Select a configuration format. Common configuration formats such as TEXT, YAML, JSON, Properties, INI and XML can be edited online. The default value is **TEXT**.
*Configuration Content	Enter the configuration content.

Parameter	Description
Enable Configuration	Determine whether to enable the configuration item. • **Enable now**: The configuration item takes effect immediately once being created. • **Not Enabled**: The configuration item does not take effect.

Step 7 Click **Create Now** to enable the configuration item.

----End

Creating a Customized Configuration Item

If application-level and microservice-level configurations cannot meet service requirements, you can customize configuration files.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Create Configuration Item** and set parameters by referring to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Configuration Item	Enter a configuration item. The configuration item is the global ID of the configuration. In the coding phase, the configuration item is used to index and operate the configuration. You are advised to use the Java package naming rule (for example, **cse.service.registry.address**) to ensure the readability and uniqueness of the configuration.
Configuration Scope	Select **Customize**.

Parameter	Description
Tag	If application-level and microservice-level configurations cannot meet service requirements, you can use labels to customize configurations.
Configuration Format	Select a configuration format. Common configuration formats such as TEXT, YAML, JSON, Properties, INI and XML can be edited online. The default value is **TEXT**.
*Configuration Content	Enter the configuration content.
Enable Configuration	Determine whether to enable the configuration item. • **Enable now**: The configuration item takes effect immediately once being created. • **Not Enabled**: The configuration item does not take effect.

Step 7 Click **Create Now** to enable the configuration item.

----End

3.8.2 Managing Configurations for ServiceComb Engine 2.x

Editing a Configuration

You can edit the configuration information of a ServiceComb engine as required. The configuration can be dynamically updated without restarting the service.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Edit** in the **Operation** column of the target configuration item. You can also click the target configuration item and then click **Edit** on the displayed configuration details page.

Step 7 Enter the configuration information in the **Configuration Content** text box and click **Save**.

----End

Importing Configurations

ServiceComb engines support the import of local configuration files. You can import a predefined configuration template (such as a YAML or JSON file) to initialize the configurations of many microservice instances at a time. If data in the configuration center is lost or deleted by mistake, you can import the backup configuration file to quickly rebuild the service configuration environment.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Import** in the upper right corner to Import files in different formats as required, and set parameters by referring to the following table.

Table 3-6 V2.0 file

Parameter	Description
File Format	Select a format of the file to be imported. The default format is **V2.0**.
Import to Specific Environment	• Disabled: The imported configuration does not change the environment label. • Enabled: Importing the configuration to a specific environment will change the environment label. Select an environment from the drop-down list.

Parameter	Description
Same Configuration	• **Terminate**: If a configuration is the same as that in the system, the import terminates. • **Skip**: During import, if a configuration is the same as that in the system, the configuration is skipped and other configurations are imported. • **Overwrite**: During import, if a configuration is the same as that in the system, the value of the configuration will be replaced.
Configuration File	Click **Import** and select the target file. Only JSON files can be imported and the file size cannot exceed 2 MB.

Table 3-7 V1.0 file

Parameter	Description
File Format	Select **V1.0**.
*Import to Specific Environment	Select a microservice environment.
Microservice	Select the microservice to which the configuration is imported from the drop-down list.
Microservice Version	Select a version of the microservice to which the configuration is imported from the drop-down list.
*Configuration File	Click **Import** and select the target file. Only JSON files can be imported and the file size cannot exceed 2 MB.

Step 7 Click **Close**.

----End

Exporting Configurations

You can export the selected configuration file to a local path to prevent data loss (due to such as server faults and manual deletion) and periodically export configurations for restoration.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Select the configuration items to be exported and click **Export**.

- Click **Export** above the configuration items. In the displayed dialog box, click **Export**.
- Click **Export** in the upper right corner. In the displayed dialog box, select the file format (V2.0 by default) and click **OK**.

NOTE

If the format of the exported configuration file is V1.0, you need to select the microservice environment, microservice name, and microservice version from the drop-down list.

----End

Viewing Configuration Details

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target configuration item. The configuration item details page shows the configuration details.

----End

Viewing Configurations of a Historical Version

You can view the configurations of different historical versions.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **View Historical Version** in the **Operation** column of a configuration item. Alternatively, click **Historical Versions** on the configuration details page. On the **Historical Versions** page that is displayed, you can view the historical versions of the configuration item. On this page, you can compare the configuration version with the rollback version.

----End

Comparing Configuration Versions

You can compare different historical versions.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the configuration item to be compared.

Step 7 Click **View Historical Version**.

Step 8 In **Historical Versions** on the left, select the historical version to be viewed. A maximum of 100 historical versions can be displayed.

In the **Configuration file** on the right, you can view the differences between the current and historical versions.

----End

Rolling Back a Version

Roll back to the selected historical version.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click the target configuration item.

Step 7 Click **View Historical Version**.

Step 8 In **Historical Versions** on the left, select the target historical version.

Step 9 In **Configuration file** on the right, click **Roll Back to the Selected Version**.

----End

Disable a configuration item

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.

- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 In the **Operation** column of the target configuration item, click **More** > **Disable**.

Step 7 Click **OK**.

----End

Deleting a Configuration Item

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Delete** in the **Operation** column of the target configuration item. You can also click the target configuration item and then click **Delete** on the displayed configuration details page.

Step 7 Click **OK**.

----End

3.9 Configuration Management (Applicable to ServiceComb Engine 1.x)

3.9.1 Creating a Configuration for ServiceComb Engine 1.x

Configuration management provides common configurations for microservices, such as log levels and running parameters. After being added, the configuration

item is used as the default one if no same configuration items are defined for microservices.

Configuration items are stored in plaintext. Do not include sensitive data.

Creating a Configuration

Configuration management provides common configurations for microservices, such as log levels and running parameters. After being added, the configuration item is used as the default one if no same configuration items are defined for microservices.

Configuration items are stored in plaintext. Do not include sensitive data.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Create Configuration Item**.

Step 7 On the **Create Configuration Item** page, select a microservice environment and enter **Configuration Item** and **Value**.

Step 8 Click **OK**.

----End

3.9.2 Managing Configurations for ServiceComb Engine 1.x

Editing a Configuration

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.

- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Edit** in the **Operation** column of the target configuration item and edit the values of the configuration item.

Step 7 Click **OK**.

----End

Importing Configurations

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Import**.

Step 7 Select a microservice environment, click **Import**, and select the target file. A maximum of 150 configuration items can be imported at a time.

Step 8 Click **Close**.

----End

Exporting Configurations

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Click **Export All**.

----End

Deleting a Configuration

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target engine.

Step 4 Choose **Configuration Management**.

- For engines with security authentication disabled, go to **Step 6**.
- For engines with security authentication enabled, if the login user is the user imported in **Importing an IAM Account**, go to **Step 6**. For other users, go to **Step 5**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 6 Select the target configuration item and click **Delete**. You can also click **Delete** in the **Operation** column of the target configuration item.

Step 7 Click **OK**.

----End

3.10 System Management

3.10.1 Overview

A ServiceComb engine may be used by multiple users. Different users must have different engine access and operation permissions based on their responsibilities and permissions.

The exclusive ServiceComb engine with security authentication enabled provides the system management function using the role-based access control (RBAC) through the microservice console.

The exclusive ServiceComb engine with security authentication enabled supports the access of Spring Cloud and Java chassis microservice frameworks.

NOTE

- The RBAC-based system management function is irrelevant to IAM permission management. It is only an internal permission management mechanism of CSE.
- To operate a ServiceComb engine on CSE, you must have both the IAM and RBAC permissions, and the IAM permission takes precedence over the RBAC permission.
- If you perform operations on a ServiceComb engine through APIs or the microservice framework, you only need to have the RBAC permissions.

1. You can use an account associated with the **admin** role to create an account and associate a proper role with the account based on service requirements. The user who uses this account has the access and operation permissions on the ServiceComb engine.
   - When you create an exclusive ServiceComb engine with security authentication enabled, the system automatically creates the **root** account associated with the **admin** role. The **root** account cannot be edited or deleted.
   - You can create an account using the **root** account of the ServiceComb engine or an account associated with the **admin** role of the engine. For details about how to create and manage an account, see **Accounts**.
2. You can create a custom role using an account associated with the **admin** role and grant proper ServiceComb engine access and operation permissions to the role based on service requirements.
   - The system provides two default roles: administrator (**admin**) and developer (**developer**). Default roles cannot be edited or deleted.
   - You can create a custom role using the **root** account of the ServiceComb engine or an account associated with the **admin** role of the engine. For details about how to create and manage a role, see **Roles**.
   - For details about role permissions, see **Table 3-8**.

Table 3-8 Role permissions

Role	Permission Description
Admin	Full permissions for all microservices, accounts, and roles of the ServiceComb engine.
Developer	Full permissions for all microservices of the ServiceComb engine.

Role	Permission Description
Custom role	You can create roles based on service requirements and grant microservice operation and configuration permissions to the roles.

3.10.2 Accounts

You can use an account associated with the **admin** role to log in to the ServiceComb engine console and create an account or manage a specified account created in the engine based on service requirements.

Table 3-9 Account management operations

Operation	Description
Adding an Account	Creates an account and associates a proper role with the account. Users who use the account have the access and operation permissions on the microservice engine. You can create up to 1000 accounts, including new accounts and the imported IAM account.
Importing an IAM Account	Imports an IAM account and associates roles with it. Users who use this IAM account have the access and operation permissions on the microservice engine. If the imported IAM account needs to connect microservice applications to the engine through programming interface authentication, **reset** its password and then use the new password to configure security authentication parameters. When you use this IAM account to log in to the CSE console with security authentication enabled, you do not need to enter the account and password. CSE can manage up to 1000 accounts, including new accounts and the imported IAM account.
Viewing Role Permissions	Displays the permissions of the role associated with a specified account.
Editing an Account	Adds or deletes roles for an account. The **root** account cannot be edited.

Operation	Description
Changing the Password	Changes the password of an account that has logged in to the ServiceComb engine based on service requirements or security regulations. • If the account and password are used to register a microservice in the SDK, changing the account and password may affect the service running of the microservice (the microservice cannot be registered with the ServiceComb engine). As a result, the service system will be damaged. Exercise caution when performing this operation. • After the password is changed, update the microservice authentication configuration in a timely manner. – Spring Cloud: see **Connecting Spring Cloud Applications to ServiceComb Engines**. – Java chassis: see **Connecting Java Chassis Applications to ServiceComb Engines**. • After the password is changed, the account may be locked due to three consecutive incorrect password attempts. The account will be unlocked after 15 minutes.
Resetting a Password	Based on service requirements or security regulations, you can use the account that has logged in to the ServiceComb engine to reset the passwords of other accounts under the engine. • If the account and password are used to register a microservice in the SDK, resetting the account and password may affect the service running of the microservice (the microservice cannot be registered with the ServiceComb engine). As a result, the service system will be damaged. Exercise caution when performing this operation. • After the password is reset, update the microservice authentication configuration in a timely manner. – Spring Cloud: see **Connecting Spring Cloud Applications to ServiceComb Engines**. – Java chassis: see **Connecting Java Chassis Applications to ServiceComb Engines**. • After the password is reset, the account may be locked due to three consecutive incorrect password attempts. The account will be unlocked after 15 minutes.
Deleting an Account	Deletes an account that is no longer used. The **root** account cannot be deleted. If the account and password are used to register a service in the SDK, deleting the account will affect the service running (the service cannot be registered with the engine). As a result, the service system will be damaged. Exercise caution when performing this operation.

Adding an Account

Before adding an account, you can create a role based on service requirements. For details, see **Creating a Role**.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.

Step 5 Choose **Accounts** > **Create Account** and configure account parameters by referring to the following table:

Parameter	Description
Account	Enter an account name. The account name cannot be changed once the account is created.
Role	Select a role based on service requirements. An account can be associated with up to five roles.
Password	Enter the password.
Confirm Password	Enter the password again.

Step 6 Click **OK**.

----End

Importing an IAM Account

Before importing an IAM account, you can create a role based on service requirements. For details, see **Creating a Role**.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 Choose **Accounts** > **Import IAM User Name**.

Step 6 Select the IAM account to be imported and select account roles. An account can be associated with up to five roles.

Step 7 Click **Confirm**.

The imported account cannot be used for login using a password. If you want to use the imported IAM account to connect microservice applications to the engine through programming interface authentication, **Resetting a Password** first.

----End

Viewing Role Permissions

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 Click the role in the **Role** column of the account to be viewed in the account list. On the displayed page, view the role and permission configuration associated with the account.

----End

Editing an Account

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Accounts** tab page, click **Edit Account** in the **Operation** column of the account to be edited.

Step 6 Select a role based on service requirements. An account can be associated with up to five roles.

Step 7 Click **Save**.

----End

Changing the Password

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**.

Step 5 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- The account for connecting to the ServiceComb engine is not associated with the admin role. You can only change the password of the current login account.
- The account for connecting to the ServiceComb engine is associated with the admin role. You can change the passwords of all accounts of the engine.
- For details about how to create an account, see **Adding an Account**.

Step 6 On the **Accounts** tab, select the account for logging in to the ServiceComb engine and click **Reset Own Password** in the **Operation** column.

1. Enter the old password and a new password, and confirm the password.
2. After confirming that the password needs to be changed, select **I Understand**.

NOTE

You can also click **Reset Own Password** in the upper right corner of the **System Management** page to change the password of the current login account.

Step 7 Click **Save**.

----End

Resetting a Password

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Accounts** tab page, select the account whose password is to be reset, and click **Reset Password** in the **Operation** column.

1. Enter a new password and confirm the password.
2. After confirming that the password needs to be reset, select **I Understand**.

Step 6 Click **Save**.

----End

Deleting an Account

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Accounts** tab page, click **Delete** in the **Operation** column of the account to be deleted.

Step 6 In the displayed dialog box, enter **DELETE** and click **OK**.

----End

3.10.3 Roles

In addition to the default roles **admin** and **developer**, you can use a ServiceComb engine account associated with the **admin** role to log in to the CSE console and perform operations based on service requirements.

Creating a Role

Creates a role and configures permission actions for the role in different service and configuration groups.

A maximum of 100 roles can be created.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Roles** tab page, click **Create Role**.

Step 6 Enter a role name. The role name cannot be changed once the role is created.

Step 7 Configure permissions.

1. Set **Permission Group**.
   a. Set the service permissions.
      - If you select **All Services**:

        You can perform corresponding permission actions on all microservices of the ServiceComb engine.
      - If you select **Custom Service Groups**, set the parameters according to **Table 3-10**.

Table 3-10 Custom service group operations

Operation	Description
Adding a Matching Rule	Click **Add Service Group Matching Rule**. Select **Application**, **Environment**, and **Service** based on service requirements to filter the microservices on which the role can perform permission actions. ○ A maximum of 20 microservice matching rules can be set for a custom service group. ○ If multiple matching rules are set for a custom service group, the role has the operation permission on those microservices which meet any of the matching rules. **Application**, **Environment**, and **Service** are three parameters of a microservice: ○ If only one parameter is set for a single matching rule, the role has the operation permission on the microservice that matches the parameter value. For example, if you add **Environment: production**, the role has the operation permission only on the microservice whose environment name is **production**. ○ If more than one parameter is set for a single matching rule, the role has the operation permission on the microservices that match all parameter values. For example, if you add **Environment: production Application: abc**, the role has the operation permission on the microservice whose environment name is **production** and application name is **abc**. ○ When automatic discovery is enabled, microservices query the instance addresses of services such as the registry center, configuration center, and dashboard through the registry center. When you grant the query permission to a microservice, the permission of the default application must be included. In this case, add the matching rule **Application: default**. After the microservice matching rule is set, click **OK**.
Editing a Matching Rule	Click  next to the matching rule to be edited. You can reconfigure **Service Group** and **Action** of the matching rule based on service requirements. After the service group matching rule is set, click **OK**.
Deleting a Matching Rule	Click  next to the matching rule to be deleted. You can delete the matching rule based on service requirements.

b. Set the configuration permissions.

- If you select **All Configurations**:

  You can perform corresponding permission actions on all configurations of the ServiceComb engine.

- If you select **Custom Configuration Groups**, set the parameters according to **Table 3-11**.

Table 3-11 Custom configuration group operations

Operation	Description
Adding a Matching Rule	Click **Add Configuration Group Matching Rule**. Select **Application**, **Environment**, and **Service** based on service requirements to filter the configurations on which the role can perform permission actions. If application-level and microservice-level configurations cannot meet service requirements, you can customize a matching rule to match the configured custom labels and filter the permission actions that can be performed by the role. ○ A maximum of 20 matching rules can be set for a custom configuration group. ○ If multiple matching rules are set for a configuration service group, the role has the operation permission on those configurations which meet any of the matching rules. **Application**, **Environment**, and **Service** are three parameters of a configuration: ○ If only one parameter is set for a single matching rule, the role has the operation permission on the configuration that matches the parameter value. For example, if you add **Environment: production**, the role has the operation permission only on the configuration whose environment name is **production**. ○ If more than one parameter is set for a single matching rule, the role has the operation permission on the configurations that match all parameter values. For example, if you add **Environment: production Application: abc**, the role has the operation permission on the configuration whose environment name is **production** and application name is **abc**. After the configuration matching rule is set, click **OK**.

Operation	Description
Editing a Matching Rule	Click  next to the matching rule to be edited. You can reconfigure **Configuration Group** and **Action** of the matching rule based on service requirements. After the configuration group matching rule is set, click **OK**.
Deleting a Matching Rule	Click  next to the matching rule to be deleted. You can delete the matching rule based on service requirements.

2. Set **Action**.

   Configure the permission actions that can be performed by the role on the selected service group and configuration group based on service requirements. You can select multiple permission actions.

   - **All**: Add, delete, modify, and query resources in the service group and configuration group.
   - **Add**: Add resources to the service group and configuration group.
   - **Delete**: Delete resources from the service group and configuration group.

     If only **Delete** is selected, you cannot delete resources in the service group and configuration group. You must select **View** at the same time.
   - **Modify**: Modify resources in the service group.

     If only **Modify** is selected, you cannot modify resources in the service group and configuration group. You must select **View** at the same time.
   - **View**: View resources in the service group and configuration group.

Step 8 Click **Create**.

----End

Editing a Role

Modifies the permissions of the created role.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Roles** tab page, click **Edit** in the **Operation** column of the role to be edited.

Step 6 Modify **Service Group**, **Configuration Group**, and **Action** based on service requirements.

Step 7 Click **Save**.

----End

Deleting a Role

Deletes a role that is no longer used.

- Deleted roles cannot be restored. Exercise caution when performing this operation.
- Before deleting a role, ensure that the role is not associated with any account. For details about how to cancel the association between a role and an account, see **Editing an Account**.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Roles** tab, click **Delete** in the **Operation** column of the role to be deleted. In the displayed dialog box, enter **DELETE** and click **OK**.

- Deleted roles cannot be restored. Exercise caution when performing this operation.
- Before deleting a role, ensure that the role is not associated with any account. For details about how to cancel the association between a role and an account, see **Editing an Account**.

----End

Viewing a Role

View the created roles of the ServiceComb engine based on the keyword of the role name.

Step 1 Log in to **CSE**.

Step 2 Choose **Exclusive ServiceComb Engines**.

Step 3 Click the target ServiceComb engine with security authentication enabled.

Step 4 Choose **System Management**. In the displayed **Security Authentication** dialog box, enter the name and password of the account associated with the **admin** role under the ServiceComb engine, and click **OK**.

NOTE

- If you connect to the ServiceComb engine for the first time, enter the account name **root** and the password entered when **Creating a ServiceComb Engine**.
- For details about how to create an account, see **Adding an Account**.

Step 5 On the **Roles** tab page, click ﹀ next to the role to be viewed to expand the role details.

Service Group, **Configuration Group**, and **Action** of the role are displayed.

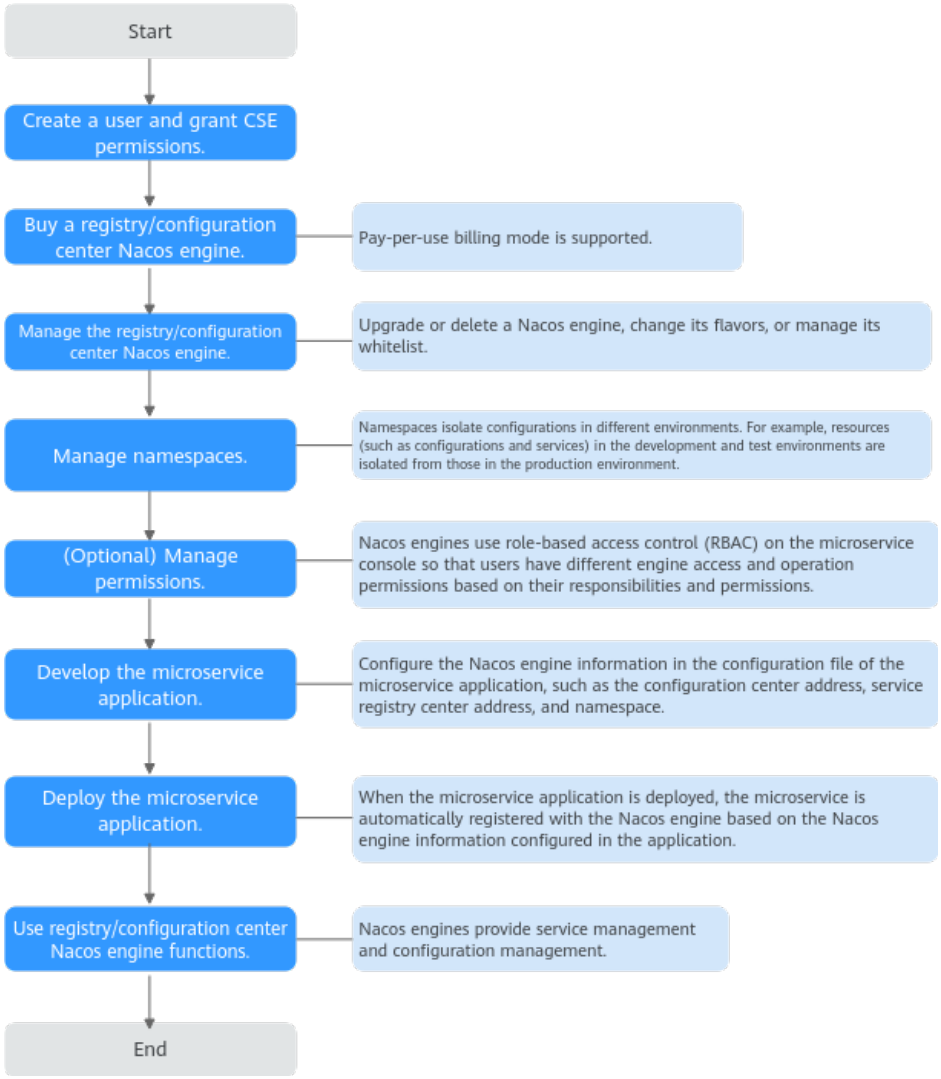
----End

4 Nacos Engines

4.1 Nacos Engine Overview

CSE Nacos is compatible with open-source Nacos and Eureka clients. It provides registry and discovery, dynamic configuration management, access permission control, and observability. It can be used to build highly available and easy-to-manage microservice middleware. Nacos is the infrastructure for service governance and configuration management in the microservice architecture. It is especially suitable for complex distributed systems that require dynamic adjustment, high availability, and multi-environment adaptation.

The following figure shows how to use Nacos as the registry/configuration center.



1. **Create a user and grant CSE permissions**.
2. **Create a registry/configuration center**.
3. Upgrade, back up, restore, or delete a Nacos engine or change its flavors by referring to **Managing Nacos Engines**.
4. **Manage namespaces** to isolate configurations in different environments.
5. (Optional) **Manage permissions** so that different users have different engine access and operation permissions based on their responsibilities and permissions.
6. Develop the microservice application. Configure the Nacos engine information in the configuration file of the microservice application, such as the configuration center address, service registry center address, and namespace.
7. Complete microservice deployment. When the microservice starts, it is automatically registered with the Nacos engine.
8. Use service management and configuration management provided by the Nacos engine by referring to **Managing Nacos Engine Services** and **Managing Nacos Engine Configurations**.

4.2 Creating a Nacos Engine

This section describes how to create an engine whose registry/configuration center is Nacos.

Restrictions

- Cluster nodes in the registry/configuration center are evenly distributed to different AZs. A failure of a single node does not affect external services. The registry/configuration center does not support AZ-level DR but provides host-level DR.
- You cannot use a shared VPC to create a Nacos engine. Otherwise, the engine fails to be created.
- The VPC cannot be changed once the Nacos engine is created.
- You may fail in buying an engine for insufficient underlying resources. To prevent this, delete engines in time so that you will not be charged if your engines restore upon sufficient underlying resources.

Prerequisites

- A Nacos engine runs on a VPC. Before creating a Nacos engine, ensure that VPCs and subnets are available. For details, see **Creating a VPC with a Subnet**.
- The user has the CSE FullAccess and DNS Full Access permissions.

Creating a Registry/Configuration Center

Step 1 Go to the **Buy Registry/Configuration Center Instance** page.

Step 2 Set parameters according to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Billing Mode	Billing mode. Currently, **Pay-per-use** is supported.
*Enterprise Project	Select the project where the Nacos engine is located. You can search for and select the required enterprise project from the drop-down list. Enterprise projects let you manage cloud resources and users by project. An enterprise project can be used after it is created and enabled. For details, see **Enabling the Enterprise Project Function**. By default, **default** is selected. After a registry/configuration center Nacos engine is created, you can remove Nacos engine resources out of the current enterprise project and into a new enterprise project. For details, see **Removing Resources from an Enterprise Project** and **Adding Resources to an Enterprise Project**.

Parameter	Description
*Name	Enter a Nacos engine name. The name contains 3 to 24 characters, including letters, digits, and hyphens (-), and starts with a letter but cannot end with a hyphen (-). The name cannot be **default**.
*Registry/ Configuration Center Instance	Select **Nacos**.
*Instances	Select the required capacity specifications. **NOTE** To create a Nacos engine with more than 2,000 microservice instances, submit a **service ticket**.
Version	Only the latest version can be created.
*Network	Select a VPC and subnet to provision logically isolated, configurable, and manageable virtual networks for your engine. • To use a created VPC, search for and select a VPC created under the current account from the drop-down list. • To use a new VPC, create one on the VPC console. For details, see **Creating a VPC with a Subnet**.
Tag	Tags are used to identify cloud resources. When you have multiple cloud resources of the same type, you can use tags to classify them based on usage, owner, or environment. Click ⊕ **Add Tag**. In the **Add Tag** dialog box, enter a tag key and value. For details about tag naming rules, see **Managing Tags**. In the **Add Tag** dialog box, you can click ⊕ **Add Tag** to add multiple tags at a time, or click ⊖ next to a tag to delete the tag.

Step 3 Click **Buy**. The page for confirming the engine information is displayed.

Step 4 Click **Submit**. When the status becomes **Available**, the engine is created.

----End

4.3 Managing Nacos Engines

4.3.1 Viewing Nacos Engine Information

This section describes how to view details about a Nacos instance on the CSE console.

Viewing Nacos Engine Information

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance. You can click an **Available** instance to go to the **Basic Information** page.

Step 4 View the Nacos engine information shown in **Table 4-1**.

Table 4-1 Engine details

Type	Parameter	Description
Basic Information	Name	Engine name entered when **Creating a Nacos Engine**. Click to copy it. An engine name can be changed. The name contains 3 to 24 characters, including letters, digits, and hyphens (-), and starts with a letter but cannot end with a hyphen.
	ID	Engine ID. Click to copy it.
	Running Status	Engine status.
	Registry/ Configuration Center Instance	Select **Nacos**.
	Version	Engine version.
	Capacity Specifications	Engine specifications selected when **Creating a Nacos Engine**. If the engine is a small-scale engine, click **Change Specifications** on the right to expand the capacity. For details, see **Increasing Nacos Engines**.
	Enterprise Project	Enterprise project selected when **Creating a Nacos Engine**.
	Billing Mode	Billing mode selected in **Creating a Nacos Engine**. Only pay-per-use billing is supported.
	Created	Time when **Creating a Nacos Engine**.
Connection Information	Private IP	Private address of a Nacos engine.
	Private Port	Internal port of a Nacos engine.
	Virtual Private Cloud	VPC selected when you **create a registry/configuration center**.

Type	Parameter	Description
	Subnet	Subnet selected when you **create a registry/configuration center**.
	Whitelist Access	Nacos supports whitelist control. Multiple IP addresses or IP address segments can be configured. IP addresses that are not in the IP address segments cannot be accessed. For details about whitelist access, see **Managing the Nacos Engine Whitelist**.
More Settings	Tag	Tags added to the Nacos engine. You can also click **Tag Management** and perform operations on tags as required. For details, see **Managing Nacos Engine Tags**.

----End

4.3.2 Managing Nacos Engine Tags

Tags facilitate Nacos engine identification and management.

If your organization has configured tag policies for Nacos engines, add tags to engines based on the policies. If a tag does not comply with the tag policies, engine creation may fail. Contact your administrator to learn more about tag policies.

You can add tags to a Nacos engine when creating the engine or add tags on the details page of the created engine. Up to 20 tags can be added to an engine. Tags can be modified and deleted.

A tag consists of a tag key and a tag value. **Table 4-2** lists the tag key and value requirements.

Table 4-2 Tag naming rules

Tag	Rule
Key	• Cannot be left blank. • Must be unique for the same instance. • Contain a maximum of 128 characters. • Contain only letters, digits, spaces, and special characters (_ . : = + - @). • Cannot start with a space or **_sys_** or end with a space.

Tag	Rule
Value	• Contain a maximum of 255 characters. • Contain only letters, digits, spaces, and special characters (_ . : = + - @).

Managing Tags

Adding or modifying tags will affect Nacos engine services for about 10 seconds. Add or modify tags during off-peak hours.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target engine. The details page is displayed.

Step 4 In the **More Settings** area, perform the following operations in the **Tags** field as required:

- Add a tag
  a. Click **Tag Management**. The **Edit Tag** dialog box is displayed.
  b. Click **Add Tag** and enter a tag key and value in the text boxes.
  c. Click **OK**.
- Modify a tag
  a. Click **Tag Management**. The **Edit Tag** dialog box is displayed.
  b. You can modify the tag key and value in the original text boxes.
  c. Click **OK**.
- Delete a tag
  Click in the row that contains the tag to be deleted. In the dialog box that is displayed, click **OK** to delete the tag.

----End

4.3.3 Managing the Nacos Engine Whitelist

The following describes how to manage whitelists of a Nacos engine to allow access only from whitelisted IP addresses.

If no whitelists are added to the engine whitelist or the whitelist function is disabled, all IP addresses that can communicate with the VPC can access the engine.

Setting a Whitelist

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target engine. The details page is displayed.

NOTE

You can click an **Available** engine to go to the **Basic Information** page.

Step 4 In the **Connection Information** area, click . In the **Set Access Whitelist** dialog box, enter **IP Address/Address Segment**. Use commas (,) to separate multiple whitelists.

NOTE

A maximum of 20 IP addresses/address segments can be added for each engine. IPv4 and IPv6 addresses are supported only in CN East 2. In other regions, only IPv4 addresses are supported.

- To modify or delete an IP address/range, modify or delete it in the displayed **Set Access Whitelist** dialog box.
- To add an IP address/range, add it in the displayed **Set Access Whitelist** dialog box.

Step 5 Click **OK**. When the engine status changes from **Configuring** to **Available**, the whitelist takes effect.

----End

4.3.4 Increasing Nacos Engines

You can increase the number of Nacos engines online. Only low-capacity engines support this operation. During increase, the interface request may fail for a short period of time. The Nacos framework provides the retry function. If the request fails, the interface will be called again. You are advised to perform the operation in your own change maintenance time window.

Increasing Nacos Engines

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click **More** > **Change Specifications** in the **Operation** column of the target Nacos engine instance. Alternatively, click the target engine and click **Change Specifications** next to **Capacity Specifications** in the **Basic Information** area on the **Basic Information** page.

Step 4 On the **Change Nacos Engine Specifications** page, select the target capacity.

Step 5 Click **Change Now**, confirm the information, and click **Submit**. When the instance status changes from **Changing** to **Available**, the operation is successful.

----End

4.3.5 Upgrading a Nacos Engine

Nacos engines are created using the latest engine version. When a later version is released, you can upgrade your engine.

Restrictions

- During the Nacos engine upgrade, the microservice and engine are intermittently disconnected, but services of running microservices are not affected. You are advised not to upgrade, restart, or change microservices when upgrading a Nacos engine.
- Version rollback is not supported after the upgrade.
- You can only upgrade to the latest version.

Upgrading a Nacos Engine

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click ⓘ in the **Version** column of the target Nacos engine.

NOTE

If the engine version is the latest, ⓘ does not exist in the **Version** column.

Step 4 In the displayed dialog box, confirm the current and target versions, and click **OK**.

If the upgrade fails, click **Retry** to perform the upgrade again.

----End

4.3.6 Deleting a Nacos Engine

You can delete a Nacos engine if it is no longer used. Deleted engines cannot be restored. Exercise caution when performing this operation. You can delete Nacos engines in the following states:

- Available
- Unavailable
- Creation failed
- Resizing failed
- Upgrade failed
- Unknown

Deleting a Nacos Engine

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click **More** > **Delete** in the **Operation** column of the target Nacos engine instance. Alternatively, click the target Nacos engine, click **Delete** in the upper right corner on the **Basic Information** page, and enter **DELETE** and click **OK** in the displayed dialog box.

NOTE

If the deletion fails, click **Force Delete**.

----End

4.4 Managing Namespaces

Namespaces isolate configurations in different environments. For example, resources (such as configurations and services) in the development and test environments are isolated from those in the production environment. Different namespaces can have the same group or data ID.

Restrictions

- The namespace ID is entered in the SDK connected to a Nacos engine. The namespace name is only the identifier used for viewing on the console.
- If your service SDK uses a namespace ID that is not created on the Nacos server for service registration and discovery, the service can be registered and discovered, but cannot be viewed on the service management page of the registry/configuration center. You need to create the corresponding namespace before viewing the service. For details, see **Creating a Namespace**.

Prerequisites

You have created a Nacos engine instance. For details, see **Creating a Nacos Engine**.

Creating a Namespace

When an instance is created, a default namespace **public** (reserved space) is automatically generated. This namespace cannot be edited or deleted. You can use this namespace to isolate resources and services. Up to 50 namespaces can be created.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 Choose **Namespaces** and click **Create Namespace**.

Step 5 In the displayed dialog box, set the parameters as follows. Configuration items marked with an asterisk (*) are mandatory.

Table 4-3 Namespace parameters

Parameter	Description
*Namespace	The namespace name can be customized and can contain a maximum of 128 characters except @ # $% ^ & *.

Parameter	Description
Namespace ID	Enter a maximum of 128 characters. Use only uppercase letters, lowercase letters, digits, hyphens (-), and underscores (_). The namespace ID must be unique. If no ID is entered during creation, the system randomly generates an ID.

Step 6 Click **OK**.

----End

Editing a Namespace

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 Choose **Namespaces**.

Step 5 Click **Edit** in the **Operation** column of the namespace to be edited to edit the namespace name.

NOTE

The automatically generated namespace **public** cannot be edited.

Step 6 Click **OK**.

----End

Deleting a Namespace

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 Choose **Namespaces**.

Step 5 Click **Delete** in the **Operation** column of the namespace to be deleted.

Step 6 In the displayed dialog box, click **OK**.

----End

4.5 Permission Control

4.5.1 Permission Control Overview

A Nacos engine may be used by many users. Exclusive Nacos engines with security authentication enabled provide permissions management using role-based access control (RBAC) on the microservice console, so that users have different engine access and operation permissions based on their responsibilities and permissions.

The Nacos engine with security authentication enabled supports microservice access.

NOTE

- Only engine 2.1.0.1 and later support this function. For engine earlier than 2.1.0.1, upgrade it to the latest version. For details, see **Upgrading a Nacos Engine**.
- If the Nacos engine version is upgraded from 2.1.0 to 2.1.0.1 or later, you need to enable security authentication to initialize the key information before using the permission control function.
- Eureka-compatible instances do not support security authentication.

4.5.2 Enabling and Disabling Security Authentication

Enabling Security Authentication

By default, security authentication is disabled for Nacos engines. You can enable security authentication on the console.

After security authentication is enabled, only accessible namespaces are displayed on the console. Clients without usernames and passwords cannot access Nacos instances. Exercise caution when performing this operation.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 Click **Set Authentication** and enable **Authenticate Programming Interface**.

Step 6 Click **OK**. After the Nacos engine is updated and the engine status changes from **Configuring** to **Available**, security authentication is enabled successfully.

----End

Disabling Security Authentication

After security authentication is disabled, permissions of each user cannot be controlled. Clients can access Nacos instances without passwords, and all namespaces are displayed on the console. Exercise caution when performing this operation.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 Click **Set Authentication**. On the **Security Settings** page, disable **Authenticate Programming Interface**.

Step 6 In the displayed dialog box, click **OK**. When the status of the engine changes to **Available**, security authentication is disabled.

----End

4.5.3 Accounts

You can log in to the engine console and create an account or manage a specified account created on the engine based on service requirements.

NOTE

If the Nacos engine is upgraded from an earlier version to 2.1.0.1, the system has the built-in account **nacos** by default. The **ROLE_ADMIN** role is associated with the account and cannot be deleted. The default password of the built-in account is **nacos**. You are advised to reset the password.

Creating an Account

Create an account and associate a proper role with the account. Users who use the account have the access and operation permissions on the Nacos engine.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 Choose **Accounts** > **Create Account** and configure account parameters by referring to the following table:

Parameter	Description
Account	Enter an account name. The account name cannot be changed once the account is created.
Password	Enter a password.
Confirm Password	Enter the password again.

Step 6 Click **OK**.

----End

Resetting a Password

For security purposes, you can reset your password on the console.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 On the **Accounts** tab page, click **Reset Password** in the **Operation** column of the target account.

Step 6 Enter and confirm a new password, select **I Understand**, and click **Save**.

----End

Deleting an Account

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 On the **Accounts** tab page, click **Delete** in the **Operation** column of the target account. In the displayed dialog box, enter **DELETE** and click **OK**.

----End

4.5.4 Roles

You can log in to the engine console and create, edit, delete, and view roles of a Nacos engine based on service requirements. Permission control by namespace or finer granularity is supported.

NOTE

If the Nacos engine is upgraded from an earlier version to 2.1.0.1, the system has the built-in role **ROLE_ADMIN** by default. The role cannot be deleted.

Creating a Role

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 On the **Roles** tab page, click **Create Role**.

Step 6 Enter a role name. The role name cannot be changed once the role is created.

Step 7 Set **Associated user**. Select the user created in **Creating an Account** from the drop-down list.

Step 8 Add permission configurations. The role also has the permissions configured here.

Click **Add Permission Configuration** and select a namespace and action (**Read only**, **Write only**, or **Read/write**). You can add multiple permission configurations at a time or click **Delete** in the **Operation** column of a permission configuration to delete it.

Step 9 Click **Create**.

----End

Editing a Role

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine with security authentication enabled.

Step 4 Choose **Permission Control**.

Step 5 On the **Roles** tab page, click **Edit** in the **Operation** column of the role to be edited.

Step 6 Modify **Namespace** and **Action** based on service requirements.

Step 7 Click **Edit**.

----End

Deleting a Role

Deleted roles cannot be restored. Exercise caution when performing this operation.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 On the **Roles** tab, click **Delete** in the **Operation** column of the role to be deleted. In the displayed dialog box, enter **DELETE** and click **OK**.

----End

4.5.5 Console Resource Management

Nacos engines support the association between namespaces and enterprise projects. The relationship is N:1, that is, N namespaces can be associated with one enterprise project.

By default, the namespace created in **Creating a Namespace** is not associated with any enterprise project. You can associate the namespace with an enterprise project by editing the enterprise project.

When editing an enterprise project, you can only change the enterprise project and cannot leave the enterprise project empty.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 Choose **Permission Control**.

Step 5 On the **Console Resource Management** tab, click in the **Enterprise Project** column of the target namespace. In the **Edit Enterprise Project** dialog box, select an enterprise project from the drop-down list and click **OK**.

----End

4.6 Managing Nacos Engine Services

You can use the CSE console to manage services registered with Nacos.

Prerequisites

A Nacos engine instance has been created.

Creating a Service

You can create a service on the console. The newly created service is an empty service (that is, the number of providers is 0). By default, the empty service is displayed in the service list. If you do not want to display the empty service, click next to **Hide Empty Service**.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 Choose **Service Management**.

Step 5 Select a namespace from the Namespace drop-down list. The ID is automatically filled in the Namespace ID box.

NOTE

If the selected namespace is **public**, the namespace ID is empty by default.

Step 6 Click **Create Service**. In the displayed dialog box, set configuration items as follows. Configuration items marked with an asterisk (*) are mandatory.

Table 4-4 Parameters

Parameter	Description
*Service Name	Enter a service name. The value can contain a maximum of 236 characters, including digits, letters, and special characters "_-.:".

Parameter	Description
Group	Set the group to which the service belongs. The value can contain a maximum of 128 characters, including digits, letters, and special characters "_-.:".
*Protection Threshold	If the ratio of healthy instances to the total instances is less than the threshold, a protection threshold is triggered. The value ranges from 0 to 1. The default value is **0**.

Step 7 Click **OK**.

----End

Viewing the Service List

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 Choose **Service Management**. Select a namespace from the **Namespace** drop-down list. The ID is automatically filled in the **Namespace ID** box.

NOTE

If the selected namespace is **public**, the namespace ID is empty by default.

Step 5 View all services in the namespace of the engine.

You can search for the target service by service name or group name.

NOTE

Target service fuzzy search supports characters: ,$*+.|?

----End

Viewing Service Details

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 Choose **Service Management**.

Step 5 Click the target service to view its details.

- View basic service information, including the service name, namespace name, service group, namespace ID, protection threshold, and number of clusters.

- The **Instances** tab displays the instance information, including the IP address, port number, cluster, health status, online/offline status, weight, and metadata. You can also perform **Instance Operations**, such as searching for instances based on metadata, bringing instances online/offline, and modifying weights.
- The **Subscribers** tab displays the list of all client instances that subscribe to the current service. Versions of subscribers and clients are displayed in the list.

----End

Instance Operations

- Search by metadata: On the **Instances** tab, select a cluster from **Clusters**, enter the metadata key and value in **Search Metadata**, and click **Filter** to display the instances that meet the search criteria. Click **Clear** to clear the search data.
- Bring an instance online or offline: On the **Instances** tab, click **Online** or **Offline** in the **Operation** column of the target instance. The instance status will be updated accordingly.
- Modify instance weight: On the **Instances** tab, move the cursor to the **Weight** column of the target instance, click  to modify the weight (ranging from 1 to 99), and click **OK**.

  NOTE

  To use the Nacos weight function for traffic load balancing, register **NacosRule** provided by Nacos as a bean on the client.

  ```
  @Bean
  NacosRule nacosRule() {
     return new NacosRule();
  }
  ```

  Add the following configuration item to the **application.properties** configuration file:

  ```
  xxx-service.ribbon.NFLoadBalancerRuleClassName=com.alibaba.cloud.nacos.ribbon.NacosRule
  ```

  xxx-service indicates the service name of the client, that is,

  ```
  spring.application.name=xxx-service
  ```

Deleting a Service

- Only empty services can be deleted. If the number of instances is not 0, the services cannot be deleted.
- If a service remains empty for more than 1 minute, the Nacos automatically deletes the service.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Service Management** and click **Delete** in the **Operation** column of the target service.

Step 5 In the displayed dialog box, click **OK**.

----End

4.7 Managing Nacos Engine Configurations

4.7.1 Nacos Engine Configuration Overview

In Nacos engines, configuration management is one of the core functions. It is used to solve problems such as centralized configuration management, dynamic update, and environment isolation in the microservice architecture. It uses a unified configuration center to adjust application behavior in real time without modifying code or restarting services, improving system flexibility and maintainability.

Nacos configuration management functions:

- Centralized configuration management
  - Application configurations are stored on the Nacos server, which prevents configurations from being scattered in code or local files and facilitates centralized maintenance and management.
  - Configurations can be managed by application, environment (development, test, and production), and cluster, improving configuration isolation.
- Dynamic configuration update

  Configurations can be pushed to applications in real time after being modified, without restarting services. (This feature requires the support of the application SDK.)
- Configuration version management

  Configuration changes are automatically recorded, and configurations can be rolled back to a previous version, preventing configuration faults caused by misoperations.
- Configuration listening and pushing

  Applications listen to Nacos configuration changes through SDKs, and Nacos proactively pushes updates to ensure that configurations take effect in real time.
- Multi-environment support

  Configurations can be isolated in different environments (development, test, and production). The same application can load different configurations in different environments.

4.7.2 Creating a Nacos Engine Configuration

Creating configurations in Nacos is the basis for centralized management, dynamic update, and multi-environment isolation of microservice configurations. This prevents scattered and disordered configurations, improving O&M efficiency and system stability. This section describes how to create a configuration file for the Nacos engine.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Select a namespace from the Namespace drop-down list. The ID is automatically filled in the Namespace ID box.

NOTE

If the selected namespace is **public**, the namespace ID is empty by default.

Step 6 Click **Create Configuration**. In the displayed dialog box, set the following parameters. Parameters marked with an asterisk (*) are mandatory.

Table 4-5 Parameters

Parameter	Description
*Data ID	The data ID is one of the dimensions for identifying configurations, and usually identifies configuration sets of a system. A system or application can contain multiple configuration sets, and each one can be identified by a name. A unique data ID is generally named like a Java package. This naming rule is optional. The value can contain a maximum of 255 characters, including digits, letters, and special characters "_-.:".
Group	It is a set of configurations in Nacos and one of the dimensions for identifying configurations. The value can contain a maximum of 128 characters, including digits, letters, and special characters "_-.:".
Namespace	Namespace to which the configuration belongs.
Configuration Format	Nacos supports online editing of common configuration formats such as YAML, Properties, TEXT, JSON, XML and HTML. The default value is **TEXT**.

Parameter	Description
*Configuration Content	Enter the configuration content. The configuration content cannot exceed 100 KB. If the configuration content is too large, split the configuration. Adjusting the configuration content size may affect Nacos stability. Exercise caution when performing this operation. To adjust the configuration content size, **submit a service ticket**.
Description	Enter the description. Enter up to 128 characters.
Application	Enter the application to which the configuration belongs. The value can contain a maximum of 128 characters, including digits, letters, and special characters "_-.:".
Label	Enter a label. The value can contain a maximum of 64 characters, including digits, letters, and special characters "_-.:".

Step 7 Click **Release**.

----End

4.7.3 Managing Nacos Engine Configurations

Querying Nacos Engine Configurations

CSE Nacos allows you to query configurations by data ID, group, application, and label.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 In the filter box above the configuration list, filter configurations by data ID, group, application, and label, and click  to display the configurations that meet the filter criteria.

----End

Viewing Nacos Engine Configuration Details

You can view configuration details about a Nacos engine on the CSE console.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Click the target data ID. On the **Configuration Details** page displayed, view the configuration details. In the **Configuration Content** area, click **search** to query the configurations.

----End

Editing Nacos Engine Configurations

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Edit a configuration in either of the following methods:

- Click **Edit** in the **Operation** column of the target data ID.
- Click the target data ID. On the **Configuration Details** page displayed, click **Edit**.

Step 6 On the **Edit Configuration** page, modify the configuration content, format, description, application, and label. Click **Release**. The **Configuration Content Comparison** dialog box is displayed. You can view the differences between the historical and current versions.

Step 7 Click **Release**. The **Edit Configuration** page also provides dark launch. For details, see **Managing Dark Launch of Nacos Engine Configurations**.

----End

Importing Nacos Engine Configurations

Importing configuration files to the Nacos engine can implement centralized and standardized configuration management. Multiple configuration files can be imported in batches to reduce the workload of manually creating configurations. If data in the configuration center is lost or deleted by mistake, you can import the backup configuration file to quickly rebuild the service configuration environment.

Step 1 Log in to **CSE**.

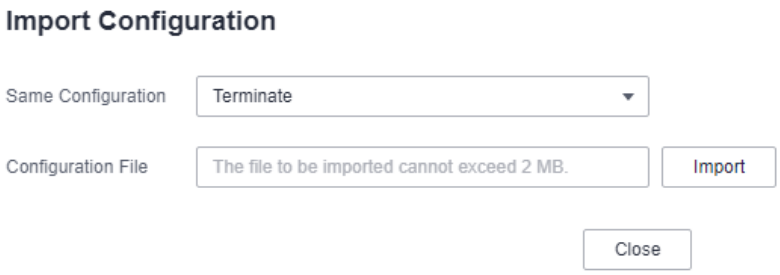
Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Click **Import Configuration** and set parameters by referring to the following table.

Figure 4-1 Importing Configurations



Parameter	Description
Same Configuration	• **Terminate**: If a configuration is the same as that in the system, the import terminates. • **Skip**: During import, if a configuration is the same as that in the system, the configuration is skipped and other configurations are imported. • **Overwrite**: During import, if a configuration is the same as that in the system, the value of the configuration will be replaced.
Configuration File	Click **Import** and select the target file. The file size cannot exceed 2 MB. If the file is too large, divide it into smaller files and import them individually.

Step 6 Click **Close**.

NOTE

- If **Same Configuration** is **Terminate**, the **Terminate** dialog box will be displayed if a configuration is the same as that in the system during the import. Click **OK** to terminate the import.
- If **Same Configuration** is **Skip**, the configuration that is the same as that in the system will be skipped during the import, and other configurations are imported. Then, a dialog box is displayed showing the imported configurations. Click **OK**.

----End

Exporting Nacos Engine Configurations

Export Nacos configurations to a file (such as properties, YAML, or JSON) as offline backup storage to prevent data loss caused by Nacos service faults, misoperations, or configuration deletion by mistake. During engine migration, export the original engine configurations and import them to the new engine in batches, avoiding the trouble of manually rebuilding configurations.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

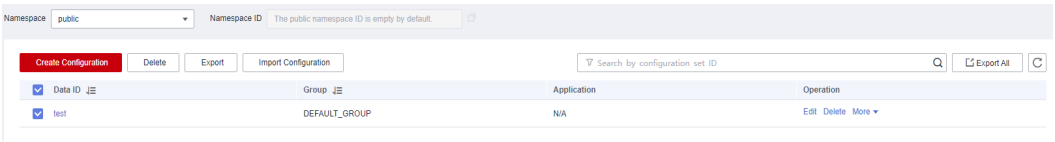
Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Select the target configuration and click **Export**.

NOTE

- Click **Export All** to export all configurations.
- You are advised to export the configurations separately to ensure that the size of an exported configuration file does not exceed 2 MB.

Figure 4-2 Exporting Configurations



Step 6 In the displayed dialog box, click **Export**.

----End

Deleting Nacos Engine Configurations

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Delete a configuration in either of the following methods:

- Click **Delete** in the **Operation** column of the target data ID.
- Select the target data ID and click **Delete** above.

Step 6 Click **OK**.

----End

4.7.4 Managing Dark Launch of Nacos Engine Configurations

The CSE Nacos configuration center supports dark launch. That is, configurations can be partially verified before official release. After verification, configurations will be officially released to reduce the risk of configuration push.

Configuring Dark Launch

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Click **Edit** in the **Operation** column of the target configuration item.

Step 6 On the **Edit Configuration** page, click  to enable dark launch.

Step 7 Select the IP address of the instance to be pushed in dark launch in the text box or manually enter the IP address of the instance for dark launch. Click **Enter**.

NOTE

Multiple instance IP addresses can be configured at the same time.

Step 8 Click **Release**. In the displayed **Configuration Content Comparison** dialog box, compare the configurations between the historical and current versions.

Step 9 Click **Release**.

----End

Viewing Dark Launch Version Configurations

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Click **Edit** in the **Operation** column of the target configuration item that is being dark launched.

Step 6 On the **Dark Launch Version** tab of the **Edit Configuration** page, you can view the dark launch version configuration, and roll back and release dark launch. For details, see **Related Operations**.

----End

Related Operations

- Roll back dark launch: On the **Dark Launch Version** tab of the **Edit Configuration** page, click **Roll Back** to cancel dark launch and roll back to the historical version.
- Release dark launch: On the **Dark Launch Version** tab of the **Edit Configuration** page, click **Release**. In the **Configuration Content Comparison** dialog box, confirm the configuration and click **Release**. The dark launch version becomes an official version.

4.7.5 Managing Historical Nacos Engine Versions

CSE Nacos allows you to view details about historical versions and roll back historical versions.

Viewing Historical Versions

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Go to the **Historical Versions** page in either of the following methods. Click the data ID of a historical version in a time segment to view the historical version information of the configuration item. Historical versions can be retained for a maximum of 30 days.

- In the **Operation** column of the target data ID, choose **More** > **Historical Versions**.
- Click the target data ID. On the **Configuration Details** page displayed, click the **Historical Versions** tab.

----End

Rolling Back a Historical Version

CSE Nacos allows you to roll back a historical version to help you quickly restore incorrect configurations, reducing potential risks in configuration management of the microservice system.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

Step 4 In the left navigation pane, choose **Configuration Management** > **Configurations**.

Step 5 Go to the **Historical Versions** page in either of the following methods.

- In the **Operation** column of the target data ID, choose **More** > **Historical Versions**.
- Click the target data ID. On the **Configuration Details** page displayed, click the **Historical Versions** tab.

Step 6 Click **Roll Back** in the **Operation** column of the target historical version. The **Historical Version Details** page is displayed.

NOTE

Only the configuration whose **Operation** is **Update** can be rolled back.

Step 7 In the **Configuration Content** area, click **Roll Back to the Selected Version**. In the displayed dialog box, click **OK**.

----End

4.7.6 Using the Listening and Query Function of the Nacos Engine

CSE Nacos provides listening query. That is, after modifying a configuration, you need to check whether the modified configuration information has been pushed to the host that listens to the configuration. This helps you better check whether the configuration change has been pushed to the client.

Step 1 Log in to **CSE**.

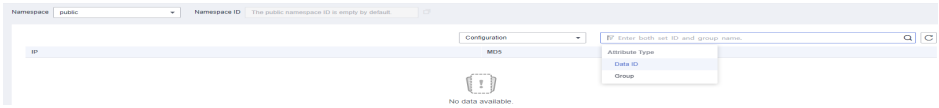
Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos instance.

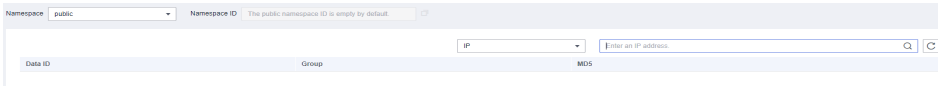
Step 4 In the left navigation pane, choose **Configuration Management** > **Listening Queries**.

Step 5 Select a namespace from the **Namespace** drop-down list, select search criteria, and click to query listening information.

- If you select **Configuration** from the drop-down list, enter the data ID and group name in the text box to query the hosts to which the configuration is pushed and the push status.



- If you select **IP** from the drop-down list, enter the IP address of the listened host is configured in the text box to query all listened configurations of the host.



----End

4.8 Viewing Monitoring of a Nacos Engine

When using the Nacos engine, you can view common metrics related to the Nacos engine in the configuration center and registry center on the running monitoring page provided by the CSE console. This section describes how to view monitoring metrics of Nacos engines.

To view the running Nacos monitoring data, ensure that you have the AOM FullAccess permission.

Step 1 Log in to **CSE**.

Step 2 In the left navigation pane, choose **Registry/Configuration Center**.

Step 3 Click the target Nacos engine.

Step 4 In the left navigation pane, choose **Monitoring**. You can view Nacos monitoring metrics.

- On the **Dashboard** tab page, you can view **Microservice Instances** and **Configurations**. In addition, **Microservice Instance Use** and **Configuration Use** are displayed in graphics.
  - Microservice Instance Use: Ratio of the number of connected microservice instances to the recommended max. number of microservice instances. Increase instances for a high ratio.
  - Configuration Use: Ratio of created configurations to the max. configurations allowed.
- On the **Configuration Center Monitoring** tab page, you can select a time from the drop-down list in the upper right corner to view the monitoring data of the configuration center in a specified period, including Configurations, Long Connections, Write Request Frequency, Read Request Frequency, Average Response Time for Write Request, Average Response Time for Read Request, and Configuration Push Time Required. The value can be Last 30 minutes, Last 1 hour, Last 6 hours, Last 1 day, or Last week. By default, the monitoring data of the last 30 minutes is displayed.
- On the **Registry Center Monitoring** tab page, you can select a time from the drop-down list in the upper right corner to view the monitoring data of the registry center in a specified period, including Microservices, Microservice Instances, Write Request Frequency, Read Request Frequency, Average Response Time for Write Request, Average Response Time for Read Request, and Service Push Time Required. The value can be Last 30 minutes, Last 1 hour, Last 6 hours, Last 1 day, or Last week. By default, the monitoring data of the last 30 minutes is displayed.

----End

5 Key Operations Recorded by CTS

5.1 CSE Operations That Can Be Recorded by CTS

With Cloud Trace Service (CTS), you can query, audit, and review operations performed on cloud resources. Traces include the operation requests sent using the management console or APIs as well as the results of these requests.

To collect, record, or query operation logs of Nacos and ServiceComb engines, **enable CTS** first. With CTS, you can view operation records of Nacos and ServiceComb engines in the last seven days. **Table 5-1** and **Table 5-2** list the supported operation logs.

Table 5-1 Nacos engine operations that can be recorded by CTS

Operation	Resource Type	Event Name
Creating an engine	engine	CreateEngineJob
Deleting an engine	engine	DeleteEngineJob
Creating a service	service	createService
Modifying a service	service	modifyService
Deleting a service	service	deleteService
Releasing a configuration	config	publishConfig
Deleting a Configuration	config	deleteConfig
Creating a namespace	namespace	createNamespace
Modifying a namespace	namespace	modifyNamespace
Deleting a namespace	namespace	deleteNamespace

Table 5-2 ServiceComb engine operations that can be recorded by CTS

Operation	Resource Type	Event Name
Creating an engine	engine	createEngine
Deleting an engine	engine	deleteEngine
Upgrading or modifying an engine	engine	upgradeOrModifyEngine
Creating an engine backup task	engine	createEngine_backup
Deleting an engine backup task	engine	deleteEngine_backup
Creating an engine restoration task	engine	createEngine_recovery
Creating an engine backup policy	engine	createEngine_backup_strategy
Deleting an engine backup policy	engine	deleteEngine_backup_strategy
Updating an engine backup policy	engine	updateEngine_backup_strategy
Updating a dark launch rule	engine	ModifyDarklaunch
Deleting a dark launch rule	engine	DeleteDarklaunch
Modifying a configuration item	engine	ModifyConfig
Creating a configuration item	engine	CreateConfig
Deleting a configuration item	engine	DeleteConfig
Updating a governance rule	engine	ModifyGovern_policy
Updating a microservice	engine	modifyMicroservice
Creating a microservice	engine	createMicroservice
Deleting a microservice	engine	deleteMicroservice
Creating a microservice tag	engine	createMicroserviceTag
Updating a microservice tag	engine	updateMicroserviceTag

Operation	Resource Type	Event Name
Deleting a microservice tag	engine	deleteMicroserviceTag
Creating a microservice rule	engine	createMicroserviceRule
Updating a microservice rule	engine	updateMicroserviceRule
Deleting a microservice rule	engine	deleteMicroserviceRule
Creating a microservice schema	engine	createMicroserviceSche-ma
Updating a microservice schema	engine	updateMicroserviceSche-ma
Deleting a microservice schema	engine	deleteMicroserviceSche-ma
Updating microservice dependencies	engine	updateMicroserviceDe-pendency
Updating microservice attributes	engine	updateMicroserviceProp-erty
Updating a microservice	engine	updateMicroservice
Updating monitoring thresholds	engine	updateThreshold
Updating a custom rule	engine	updateItem_meta
Deleting a custom rule	engine	DeleteItem_meta
Clearing configuration items	engine	executeConfig_cleanup
Updating status of a microservice instance	engine	updateInstanceStatus
Updating attributes of a microservice instance	engine	updateInstanceProperty
Creating a microservice instance	engine	createInstance
Deleting a microservice instance	engine	deleteInstance

5.2 Viewing CTS Traces in the Trace List

Scenarios

Cloud Trace Service (CTS) records operations performed on cloud service resources. A record contains information such as the user who performed the operation, IP address, operation content, and returned response message. These records facilitate security auditing, issue tracking, and resource locating. They also help you plan and use resources, and identify high-risk or non-compliant operations.

What Is a Trace?

A trace is an operation log for a cloud service resource, tracked and stored by CTS. Traces record operations such as adding, modifying, or deleting cloud service resources. You can view them to identify who performed operations and when for detailed tracking.

What Is a Management Tracker and Data Tracker?

A management tracker identifies and associates with all your cloud services, recording all user operations. It records management traces, which are operations performed by users on cloud service resources, such as their creation, modification, and deletion.

A data tracker records details of user operations on data in OBS buckets. It records data traces reported by OBS, detailing user operations on data in OBS buckets, including uploads and downloads.

Constraints

- Before the organization function is enabled, you can query the traces of a single account on the CTS console. After the organization function is enabled, you can only view multi-account traces on the **Trace List** page of each account, or in the OBS bucket or the **CTS/system** log stream configured for the management tracker with the organization function enabled. For details about organization trackers, see **Organization Trackers**.
- You can only query operation records of the last seven days on the CTS console. They are automatically deleted upon expiration and cannot be manually deleted. To store them for longer than seven days, configure transfer to Object Storage Service (OBS) or Log Tank Service (LTS) so that you can view them in OBS buckets or LTS log groups.
- After creating, modifying, or deleting a cloud service resource, you can query management traces on the CTS console 1 minute later and query data traces 5 minutes later.
- Data traces are not displayed in the trace list of the new version. To view them, you need to go to the old version.

Prerequisites

1. **Register with Huawei Cloud and complete real-name authentication.**

If you already have a Huawei Cloud account, skip this step. If you do not have one, do as follows:

a. Log in to the **Huawei Cloud official website**, and click **Sign Up** in the upper right corner.

b. Complete the registration as prompted. For details, see **Registering with Huawei Cloud**.

   Your personal information page is displayed after the registration completes.

c. Complete individual or enterprise real-name authentication by referring to **Real-Name Authentication**.

2. **Grant permissions for users.**

   If you log in to the console using a Huawei Cloud account, skip this step.

   If you log in to the console as an IAM user, first contact your CTS administrator (account owner or a user in the **admin** user group) to obtain the **CTS FullAccess** permissions. For details, see **Assigning Permissions to an IAM User**.

Viewing Traces

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, CTS starts recording user operations on data in OBS buckets. CTS retains operation records of the latest seven days.

This section describes how to query and export operation records of the last seven days on the CTS console.

Viewing Real-Time Traces in the Trace List of the New Edition

Step 1 Log in to the **CTS console**.

Step 2 Log in to the management console, click ☰ in the upper left corner, and choose **Management & Deployment** > **Cloud Trace Service**.

Step 3 In the navigation pane, choose **Trace List**.

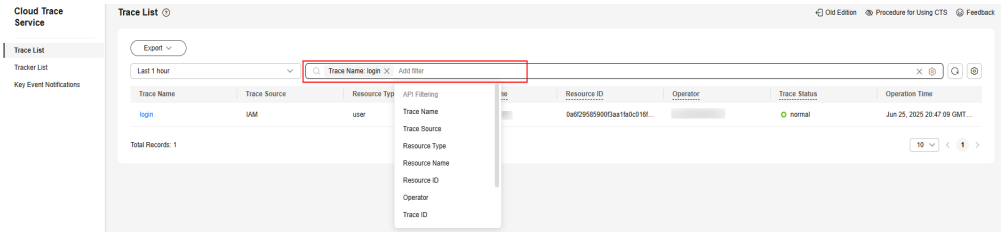
Step 4 In the time range drop-down list above the trace list, select a desired query time range: **Last 1 hour**, **Last 1 day**, or **Last 1 week**. You can also select **Custom** to specify a custom time range within the last seven days.

Step 5 The search box above the trace list supports advanced queries. Combine one or more filters to refine your search.

Table 5-3 Trace filtering parameters

Parameter	Description
Trace Name	Name of a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. For details about the operations that can be audited for each cloud service, see **Supported Services and Operations**section "Supported Services and Operations" in the *Cloud Trace Service User Guide*. Example: **updateAlarm**
Trace Source	Cloud service name abbreviation. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. Example: **IAM**
Resource Name	Name of a cloud resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty. Example: **ecs-name**
Resource ID	ID of a cloud resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. Leave this field empty if the resource has no resource ID or if resource creation failed. Example: *{VM ID}*
Trace ID	Value of the **trace_id** parameter for a trace reported to CTS. The entered value requires an exact match. Fuzzy matching is not supported. Example: **01d18a1b-56ee-11f0-ac81-******1e229**
Resource Type	Type of a resource involved in a trace. The entered value is case-sensitive and requires an exact match. Fuzzy matching is not supported. For details about the resource types of each cloud service, see **Supported Services and Operations**section "Supported Services and Operations" in the *Cloud Trace Service User Guide*. Example: **user**

Parameter	Description
Operator	User who triggers a trace. Select one or more operators from the drop-down list. If the value of **trace_type** in a trace is **SystemAction**, the operation is triggered by the service and the trace's operator may be empty. For details about the relationship between IAM identities and operators and the operator username format, see **Relationship Between IAM Identities and Operators**.
Trace Status	Select one of the following options from the drop-down list: • **normal**: The operation succeeded. • **warning**: The operation failed. • **incident**: The operation caused a fault that is more serious than a normal failure, for example, causing other faults.
Enterprise Project ID	ID of the enterprise project to which a resource belongs. To check enterprise project IDs, go to the Enterprise Project Management Service (EPS) console and choose **Project Management** in the navigation pane. Example: **b305ea24-c930-4922-b4b9-******1eb2**
Access Key	Temporary or permanent access key ID. To check access key IDs, hover over your username in the upper right corner of the console and select **My Credentials** from the pop-up list. On the displayed page, choose **Access Keys** in the navigation pane. Example: **HSTAB47V9V*******TLN9**



Step 6 On the **Trace List** page, you can also export and refresh the trace list, and customize columns to display.

- Enter any keyword in the search box and press **Enter** to filter desired traces.
- Click **Export** to export all traces in the query result as an .xlsx file. The file can contain up to 5,000 records.
- Click to view the latest information about traces.
- Click to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled ( ), excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.

Step 7 (Optional) On the **Trace List** page of the new edition, click **Old Edition** in the upper right corner to switch to the **Trace List** page of the old edition.

----End

Viewing Traces in the Trace List of the Old Edition

Step 1 Log in to the **CTS console**.

Step 2 Log in to the management console, click  in the upper left corner, and choose **Management & Deployment** > **Cloud Trace Service**.

Step 3 In the navigation pane, choose **Trace List**.

Step 4 Each time you log in to the CTS console, the new edition is displayed by default. Click **Old Edition** in the upper right corner to switch to the trace list of the old edition.

Step 5 In the upper right corner of the page, set a desired query time range: **Last 1 hour**, **Last 1 day**, or **Last 1 week**. You can also click **Customize** to specify a custom time range within the last seven days.

Step 6 Set filters to search for your desired traces.

Table 5-4 Trace filtering parameters

Parameter	Description
Trace Type	Select **Management** or **Data**. • Management traces record operations performed by users on cloud service resources, including creation, modification, and deletion. • Data traces are reported by OBS and record operations performed on data in OBS buckets, including uploads and downloads.
Trace Source	Select the name of the cloud service that triggers a trace from the drop-down list.
Resource type	Select the type of the resource involved in a trace from the drop-down list. For details about the resource types of each cloud service, see **Supported Services and Operations**section "Supported Services and Operations" in the *Cloud Trace Service User Guide*.

Parameter	Description
Search By	Select one of the following options: • **Resource ID**: ID of the cloud resource involved in a trace. Leave this field empty if the resource has no resource ID or if resource creation failed. • **Trace name**: name of a trace. For details about the operations that can be audited for each cloud service, see **Supported Services and Operations**section "Supported Services and Operations" in the *Cloud Trace Service User Guide*. • **Resource name**: name of the cloud resource involved in a trace. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty.
Operator	User who triggers a trace. Select one or more operators from the drop-down list. If the value of **trace_type** in a trace is **SystemAction**, the operation is triggered by the service and the trace's operator may be empty. For details about the relationship between IAM identities and operators and the operator username format, see **Relationship Between IAM Identities and Operators**.
Trace Status	Select one of the following options: • **Normal**: The operation succeeded. • **Warning**: The operation failed. • **Incident**: The operation caused a fault that is more serious than a normal failure, for example, causing other faults.

Step 7 Click **Query**.

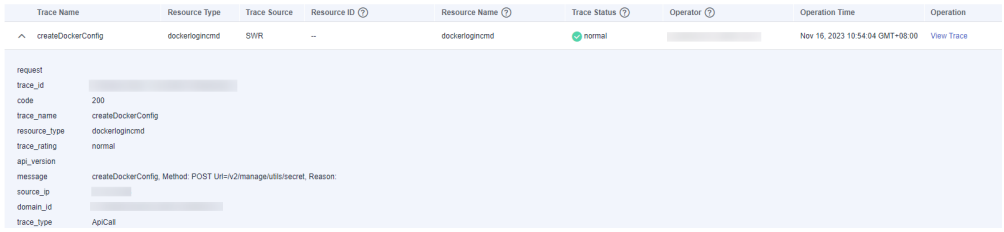
Step 8 On the **Trace List** page, you can also export and refresh the trace list.

- Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5,000 records.
- Click  to view the latest information about traces.

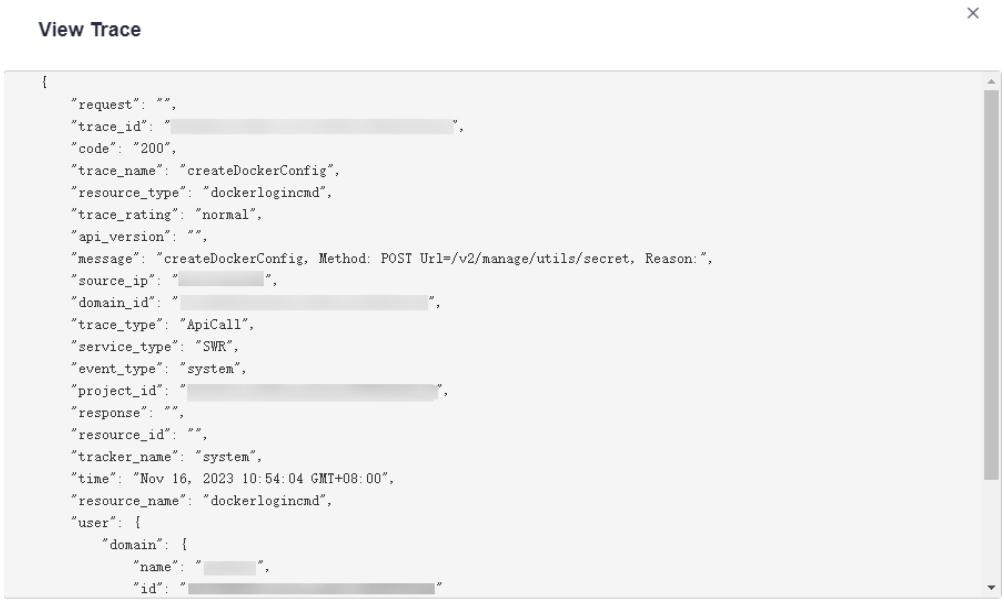
Step 9 In the **Tampered or Not** column of a trace, check whether the trace is tampered with.

- If no, **No** is displayed.
- If yes, **Yes** is displayed.

Step 10 Click on the left of a trace to expand its details.



Step 11 Click **View Trace** in the **Operation** column. The trace details are displayed.



Step 12 (Optional) On the **Trace List** page of the old edition, click **New Edition** in the upper right corner to switch to the **Trace List** page of the new edition.

----End

Helpful Links

- For details about the key fields in the trace structure, see **Trace Structure**section "Trace References" > "Trace Structure" in the *Cloud Trace Service User Guide* and **Example Traces**section "Trace References" > "Example Traces" in the *Cloud Trace Service User Guide*.
- You can use the following examples to learn how to query a specific trace:
  - Use CTS to audit Elastic Volume Service (EVS) creation and deletion operations from the last two weeks. For details, see **Security Auditing**.
  - Use CTS to locate a fault or creation failure for an Elastic Cloud Server (ECS). For details, see **Fault Locating**.
  - Use CTS to check all operation records for an ECS. For details, see **Resource Tracking**.

