Cloud Application Engine

Best Practices

Issue 01

Date 2024-07-18

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

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 Using CAE to Host Nginx Static File Servers

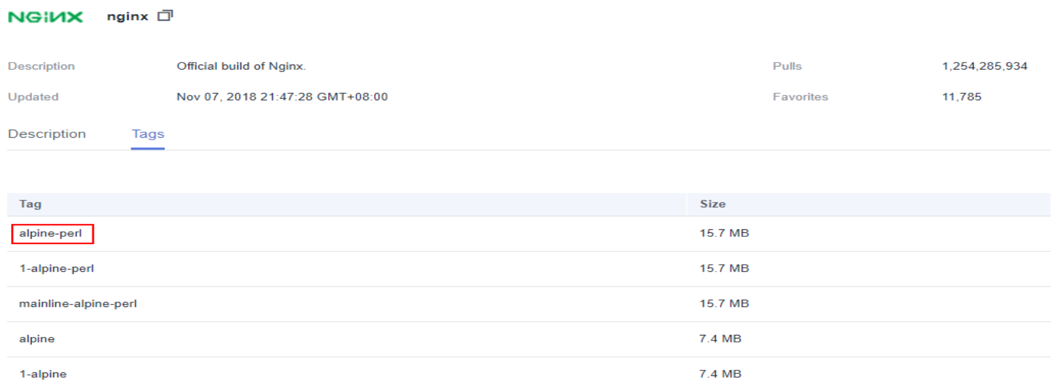
1.1 Overview

Application Scenario

This service is used to host web frontend components of service code and static pages. Create an image for service code and deploy it to CAE, and store the static files in the parallel file system associated with a component. In this way, you can host the frontend components of services and static files. After components are deployed, you can update static page files in the parallel file system to update frontend applications in real time.

This solution uses Nginx **alpine-perl**, which has been provided in the open-source image.

Figure 1-1 Image tag



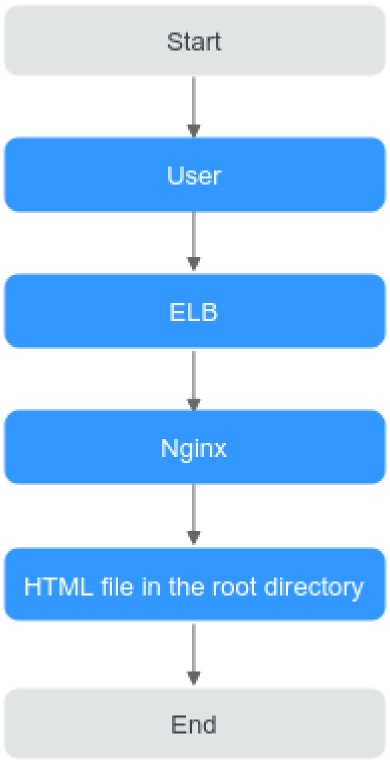
Description Official build of Nginx.
Updated Nov 07, 2018 21:47:28 GMT+08:00
Pulls 1,254,285,934
Favorites 11,785

Tag	Size
alpine-perl	15.7 MB
1-alpine-perl	15.7 MB
mainline-alpine-perl	15.7 MB
alpine	7.4 MB
1-alpine	7.4 MB

Architecture

Nginx is a lightweight web server and an HTTP server for static resources. This practice uses Nginx to describe how to configure a parallel file system in the cloud storage to host static files and update these static files to update the Nginx access page in real time.

Figure 1-2 Nginx access relationship

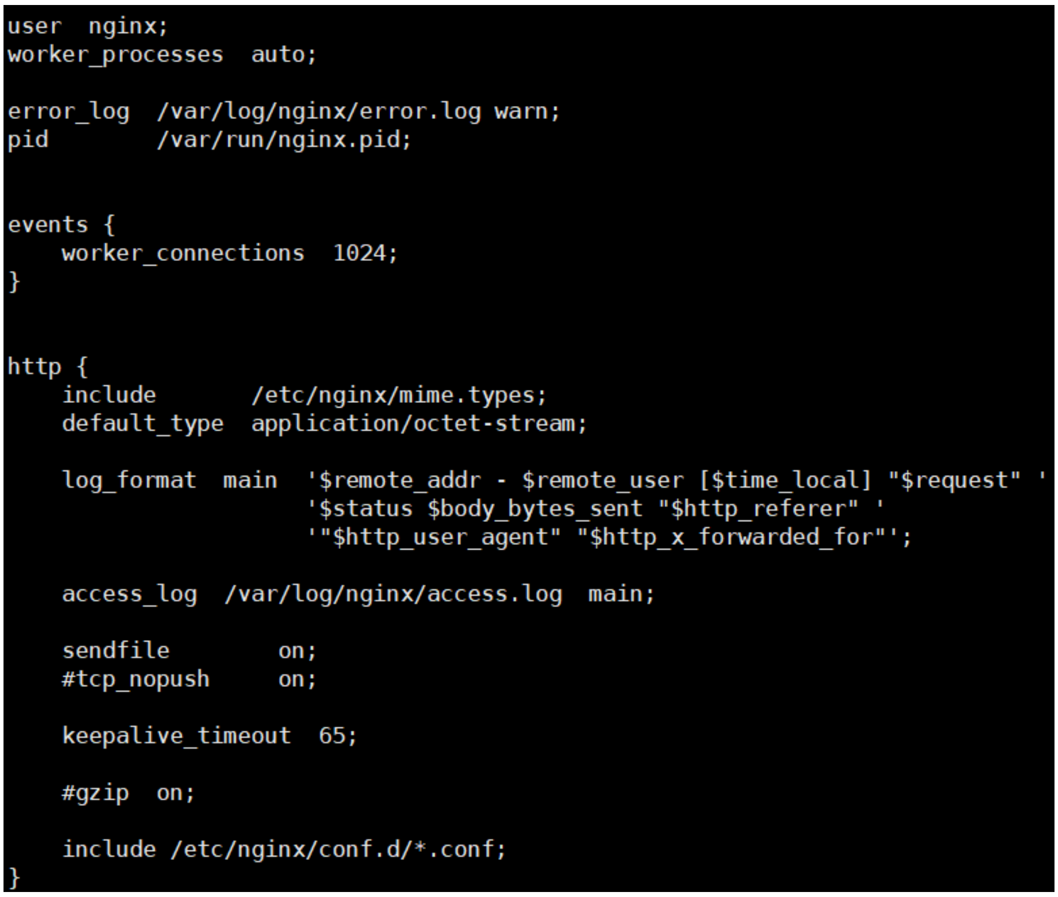


Default Nginx Configurations

Figure 1-3 shows the default Nginx configurations.

The Nginx configuration file (**nginx.conf**) of this tag is **/etc/nginx/nginx.conf**.

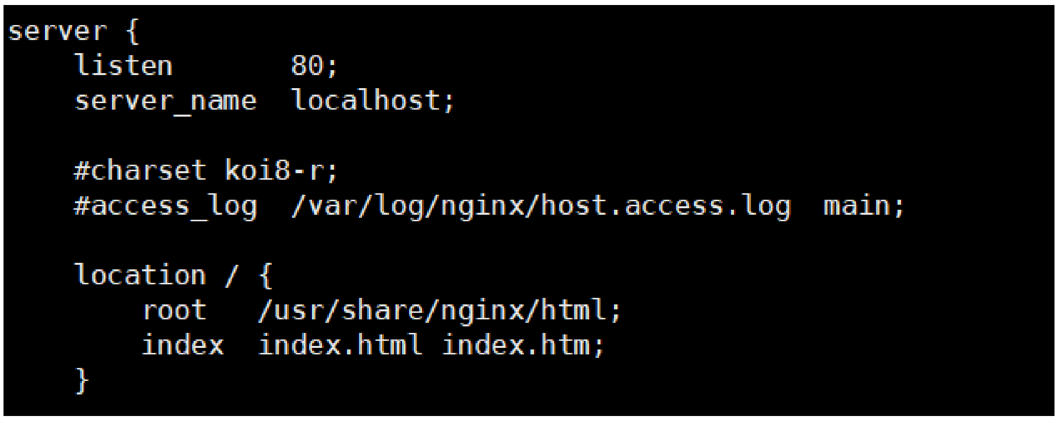
Figure 1-3 nginx.conf



```
user  nginx;
worker_processes  auto;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;


events {
    worker_connections  1024;
}


http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    keepalive_timeout  65;

    #gzip  on;

    include /etc/nginx/conf.d/*.conf;
}
```

Figure 1-4 shows the **default.conf** file of Nginx. **/usr/share/nginx/html** stores static files. There are two default files in this path: **50x.html** and **index.html**.

Figure 1-4 default.conf



```
server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;
    #access_log  /var/log/nginx/host.access.log  main;

    location / {
        root   /usr/share/nginx/html;
        index  index.html index.htm;
    }
```

To replace the page display, place the HTML file to be replaced in the **/usr/share/nginx/html** directory. The HTML file can be downloaded from **Obtaining a Static File**.

1.2 Preparations

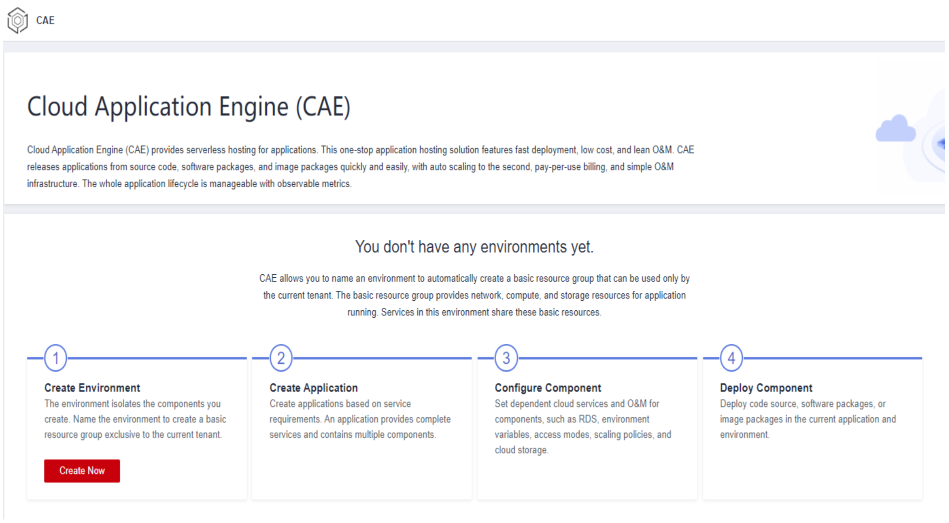
Creating an Environment

Step 1 Log in to CAE.

Step 2 Use either of the following methods to create an environment:

- If you use CAE for the first time, a message is displayed indicating that no environment has been created.
  a. Click **Create Now** in the **Create Environment** card.

Figure 1-5 Creating an environment



CAE

Cloud Application Engine (CAE)

Cloud Application Engine (CAE) provides serverless hosting for applications. This one-stop application hosting solution features fast deployment, low cost, and lean O&M. CAE releases applications from source code, software packages, and image packages quickly and easily, with auto scaling to the second, pay-per-use billing, and simple O&M infrastructure. The whole application lifecycle is manageable with observable metrics.

You don't have any environments yet.

CAE allows you to name an environment to automatically create a basic resource group that can be used only by the current tenant. The basic resource group provides network, compute, and storage resources for application running. Services in this environment share these basic resources.

1 Create Environment
The environment isolates the components you create. Name the environment to create a basic resource group exclusive to the current tenant.
Create Now

2 Create Application
Create applications based on service requirements. An application provides complete services and contains multiple components.

3 Configure Component
Set dependent cloud services and O&M for components, such as RDS, environment variables, access modes, scaling policies, and cloud storage.

4 Deploy Component
Deploy code source, software packages, or image packages in the current application and environment.

  b. In the displayed dialog box, set the parameters by referring to **Table 1-1**.

Table 1-1 Creating an environment

Parameter	Description
Environment	Enter an environment name.
Enterprise Project	Select an enterprise project. An enterprise project facilitates project-level management and grouping of cloud resources and users. The default project is **default**. It is available after the **enterprise project function** is enabled.

Parameter	Description
Virtual Private Cloud	Select the VPC to which the environment resource belongs from the drop-down list. To create a VPC, click **Create VPC**. For details, see **Creating a VPC**. NOTE The VPC cannot be modified after the environment is created.
Subnet	Select a subnet from the drop-down list. If no subnet is available, click **Create Subnet** to access the network console and create a subnet. For details, see **Creating a Subnet for the VPC**. NOTE Keep at least two available network IP addresses for CAE configuration and optimization.
Security Group	Select **Automatically generated**. NOTE This group must allow access from the selected subnet to both the subnet gateway address and the access addresses and ports of middleware such as RDS and CSE instances.
Organization	If you use CAE for the first time, select **Create Organization** from the drop-down list and enter an organization name.

- If this is not your first time using CAE, choose **Components**.
  a. Click ⊕ next to **Environment** in the upper part of the page.
  b. In the displayed **Create Environment** dialog box, enter an environment name.

  NOTE

  You can use an existing environment.

Step 3 Click **OK**.

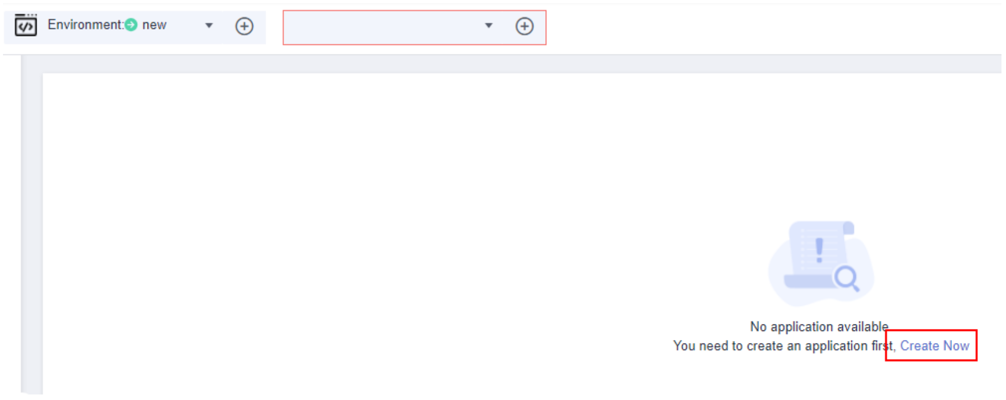
----End

Creating an Application

Step 1 Choose **Components** and use either of the following methods to create an application:

- Click ⊕ next to **Application** in the upper part of the page.
- Click **Create Now** on the **Components** page.

Figure 1-6 Creating an application



Step 2 In the displayed **Create Application** dialog box, enter an application name.

Step 3 Click **OK**.

----End

Obtaining a Static File

You can download the **index.html** and **test.html** static files for update.

Download address: **https://international-cae-demo.obs.ap-southeast-3.myhuaweicloud.com/nginx.zip**

1.3 Procedure

Restrictions

The official Nginx image is executed by the Nginx user. You need to set the file mask (umask) of the mounted file to **0022** to ensure that the Nginx user has the permission to read the mounted file. For details, see **Configuring a Cloud Storage Mount Path**.

Creating and Deploying a Nginx Component

Step 1 Log in to CAE.

Step 2 Choose **Components**.

Step 3 Select the created application and environment from the drop-down lists in the upper part of the page, and click **Create Component**.

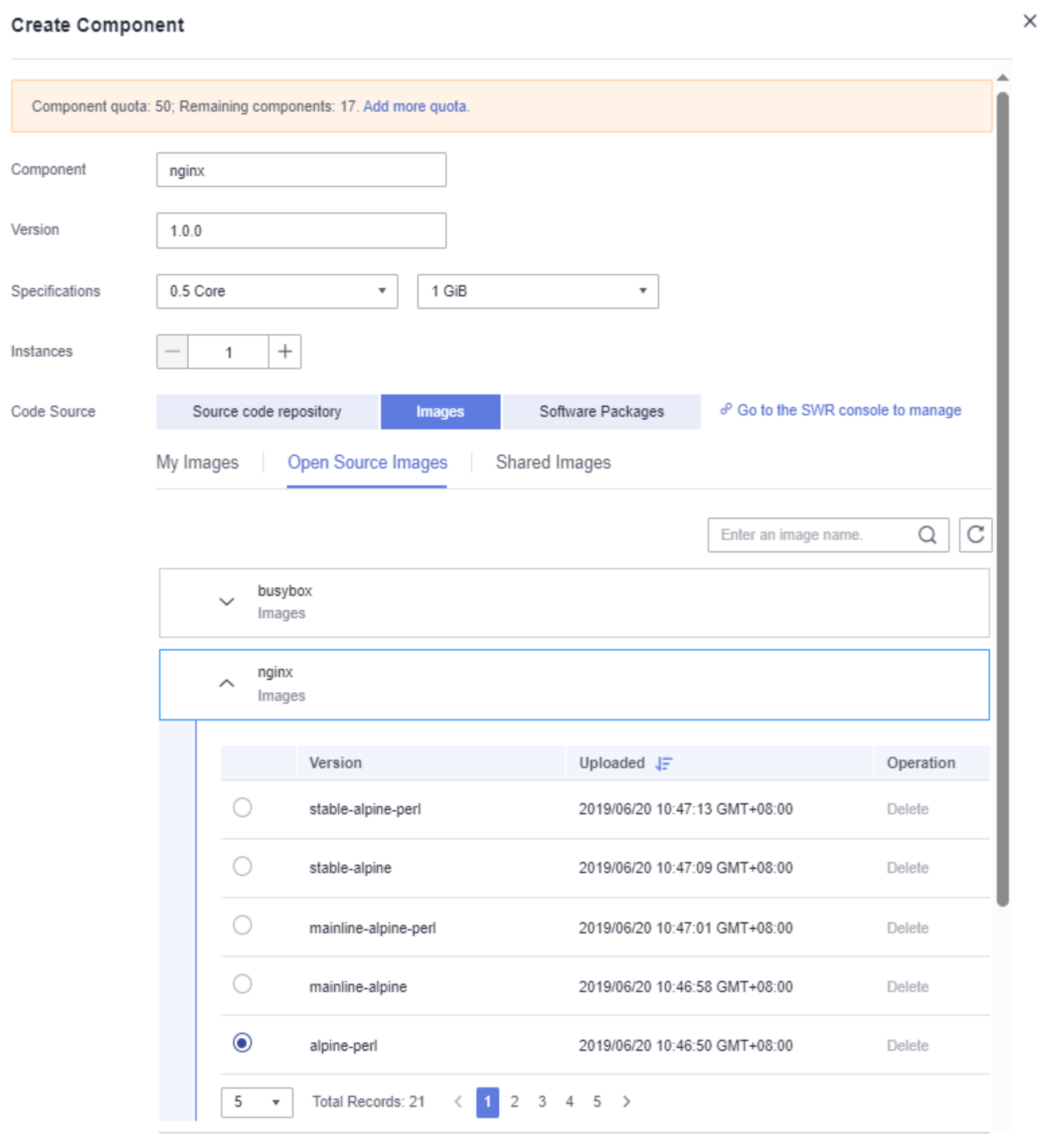
Step 4 Configure the component by referring to **Table 1-2**.

Table 1-2 Basic component information

Parameter	Description
Component	Enter a component name. In this practice, enter **nginx**.

Parameter	Description
Version	Enter a component version. In this practice, enter **1.0.0**.
Specifications	Select instance specifications, for example, 0.5 Core and 1 GiB.
Instances	Set it to **1**.
Code Source	Choose **Images** > **Open Source Images** > **nginx**. In this practice, select Nginx **alpine-perl**.

Figure 1-7 Creating a component



Create Component

Component quota: 50; Remaining components: 17. Add more quota.

Component: nginx
Version: 1.0.0
Specifications: 0.5 Core, 1 GiB
Instances: 1
Code Source: Source code repository | Images | Software Packages — Go to the SWR console to manage

My Images | Open Source Images | Shared Images

Enter an image name.

busybox
Images

nginx
Images

	Version	Uploaded	Operation
○	stable-alpine-perl	2019/06/20 10:47:13 GMT+08:00	Delete
○	stable-alpine	2019/06/20 10:47:09 GMT+08:00	Delete
○	mainline-alpine-perl	2019/06/20 10:47:01 GMT+08:00	Delete
○	mainline-alpine	2019/06/20 10:46:58 GMT+08:00	Delete
●	alpine-perl	2019/06/20 10:46:50 GMT+08:00	Delete

5 Total Records: 21 < 1 2 3 4 5 >

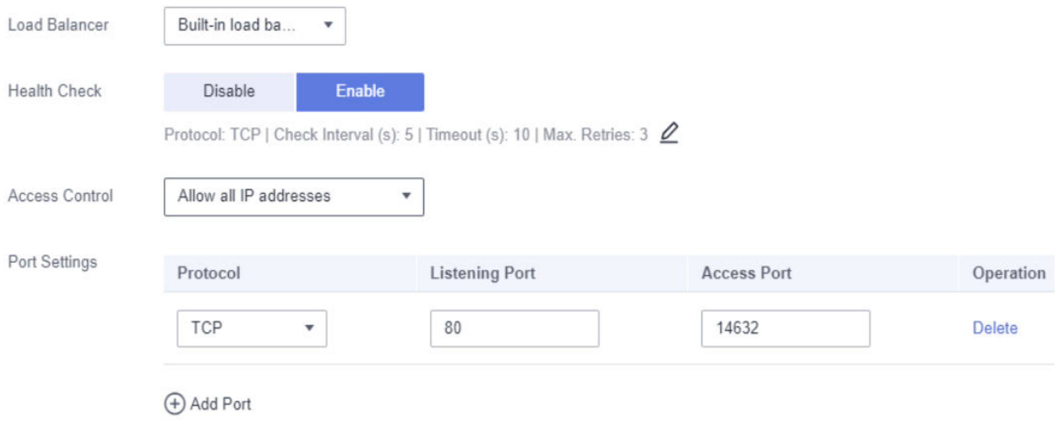
Step 5 Click **Configure Component**.

Step 6 On the **Component Configurations** page, click **Edit** in the **Access Mode** module to configure the Nginx component.

Step 7 On the **Load Balancing** tab, click **Add Load Balancer** and set parameters.

- **Load Balancer**: Select **Built-in load balancer**.
- **Health Check**: Use the default value **Start**.
- **Access Control**: Select **Allow all IP addresses**.
- **Protocol**: Select **TCP**.
- **Listening Port**: Enter **80**.
- **Access Port**: Enter an access port. In this practice, enter **14632**.

Figure 1-8 Adding external network access for load balancing



Load Balancer: Built-in load ba...

Health Check: Disable | Enable

Protocol: TCP | Check Interval (s): 5 | Timeout (s): 10 | Max. Retries: 3

Access Control: Allow all IP addresses

Port Settings

Protocol	Listening Port	Access Port	Operation
TCP	80	14632	Delete

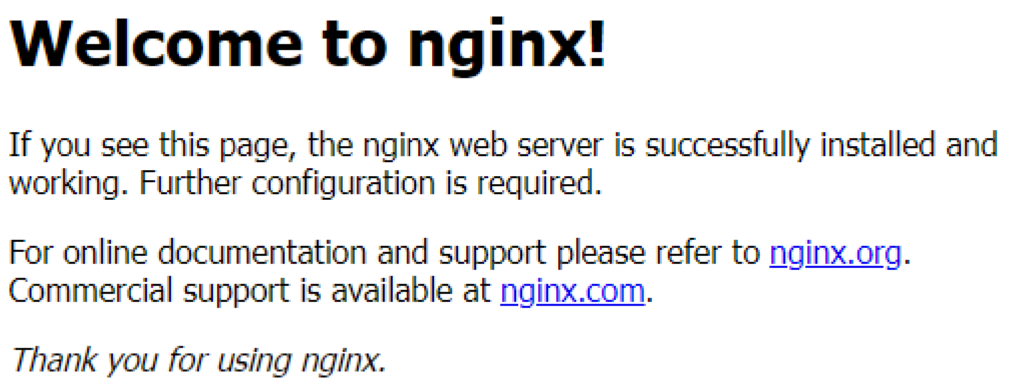
Step 8 Click **OK**.

Step 9 On the **Component Configurations** page, click **Set and Deploy Component** for the Nginx component access mode to take effect.

Step 10 After deployment, choose **Components**.

Click the IP address in the **Access Address** column of the component to view the Nginx static web page.

Figure 1-9 Accessing a static page



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

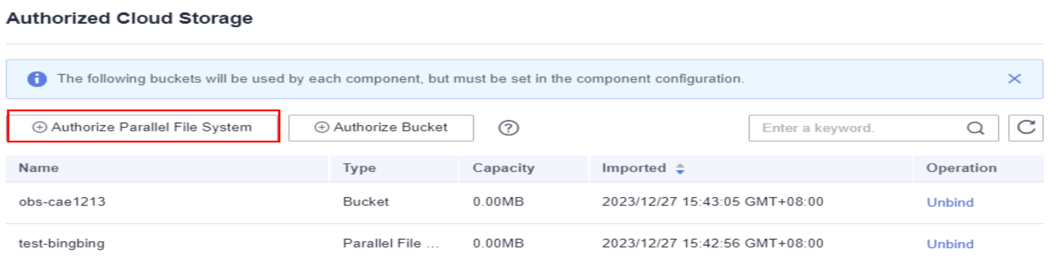
----End

Cloud Storage Authorizations

Step 1 Choose **System Settings** > **Cloud Storage Authorizations** and click **Edit**. The **Authorized Cloud Storage** dialog box is displayed.

Step 2 Click **Authorize Parallel File System**.

Figure 1-10 Authorize Parallel File System



Authorized Cloud Storage

The following buckets will be used by each component, but must be set in the component configuration.

Name	Type	Capacity	Imported	Operation
obs-cae1213	Bucket	0.00MB	2023/12/27 15:43:05 GMT+08:00	Unbind
test-bingbing	Parallel File ...	0.00MB	2023/12/27 15:42:56 GMT+08:00	Unbind

Step 3 Click **Create Parallel File System** and enter a name, for example, **test-nginx**.

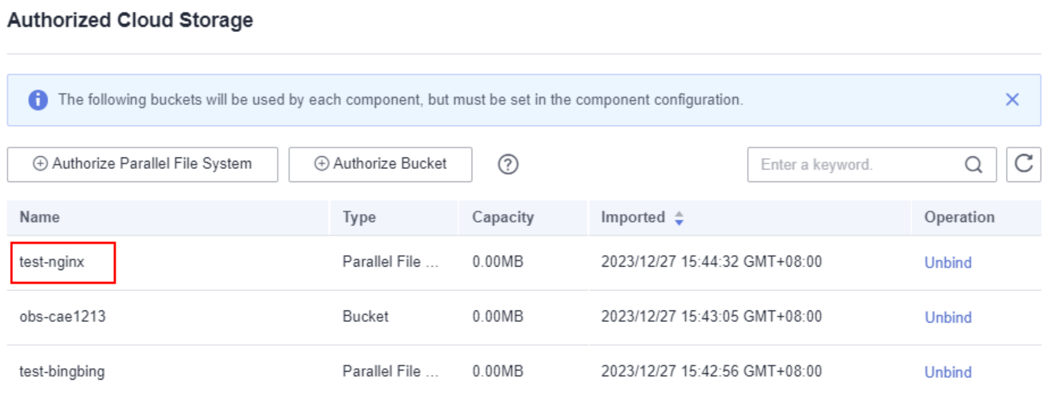
Step 4 Click **OK**.

Step 5 Select the created parallel file system. You can also enter a keyword in the search box above the list to filter data.

Step 6 Click **Authorize**. The parallel file system is authorized.

You can view the authorized parallel file systems on the **Authorized Cloud Storage** page.

Figure 1-11 Authorized parallel file system



Authorized Cloud Storage

The following buckets will be used by each component, but must be set in the component configuration.

Name	Type	Capacity	Imported	Operation
test-nginx	Parallel File ...	0.00MB	2023/12/27 15:44:32 GMT+08:00	Unbind
obs-cae1213	Bucket	0.00MB	2023/12/27 15:43:05 GMT+08:00	Unbind
test-bingbing	Parallel File ...	0.00MB	2023/12/27 15:42:56 GMT+08:00	Unbind

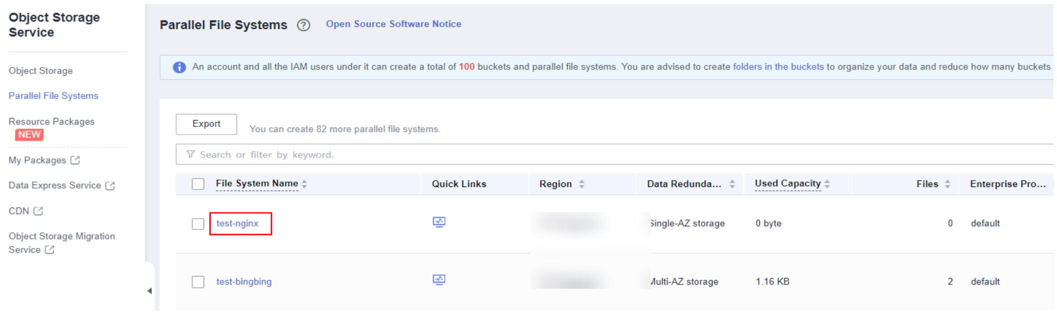
----End

Uploading a File

Step 1 Log in to OBS.

Step 2 Choose **Parallel File Systems** and click file system **test-nginx** to go to the parallel system page.

Figure 1-12 Parallel file systems



An account and all the IAM users under it can create a total of 100 buckets and parallel file systems. You are advised to create folders in the buckets to organize your data and reduce how many buckets

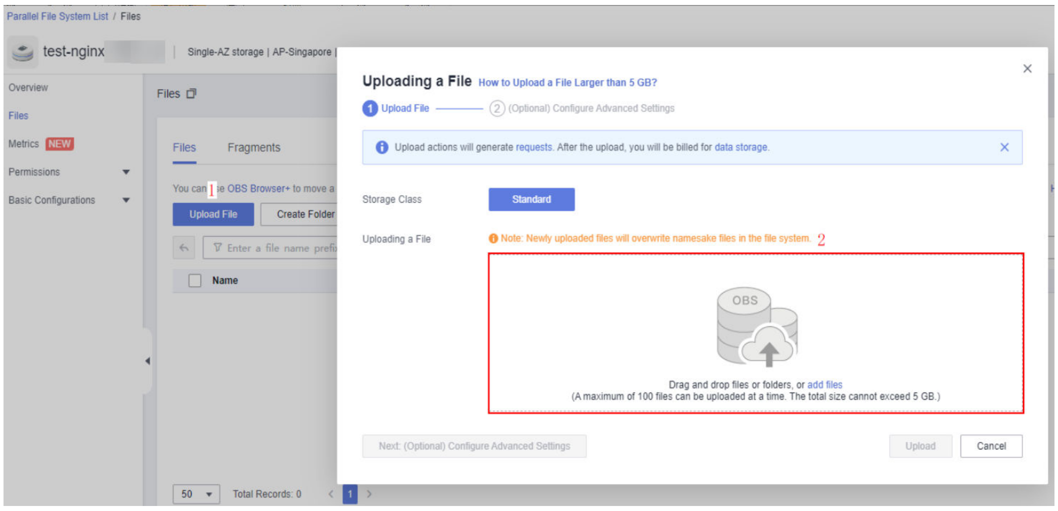
You can create 82 more parallel file systems.

File System Name	Quick Links	Region	Data Redunda...	Used Capacity	Files	Enterprise Pro...
test-nginx			Single-AZ storage	0 byte	0	default
test-bingbing			Multi-AZ storage	1.16 KB	2	default

Step 3 Choose **Files** and click **Upload File**.

Step 4 Drag the obtained static files **index.html** and **test.html** to the file upload area. You can also click **add files** to upload the files. The default storage class is **Standard**.

Figure 1-13 Uploading a file


Parallel File System List / Files
test-nginx
Single-AZ storage | AP-Singapore
Overview
Files
Metrics NEW
Permissions
Basic Configurations
Files
Files
Fragments
You can use OBS Browser+ to move a
Upload File
Create Folder
Enter a file name prefix
Name
50
Total Records: 0
1
Uploading a File
How to Upload a File Larger than 5 GB?
1 Upload File
2 (Optional) Configure Advanced Settings
1
2
OBS
Next: (Optional) Configure Advanced Settings
Upload
Cancel

Upload actions will generate requests. After the upload, you will be billed for data storage.

Storage Class: Standard

Uploading a File: Note: Newly uploaded files will overwrite namesake files in the file system.

Drag and drop files or folders, or add files
(A maximum of 100 files can be uploaded at a time. The total size cannot exceed 5 GB.)

Step 5 Click **Upload**.

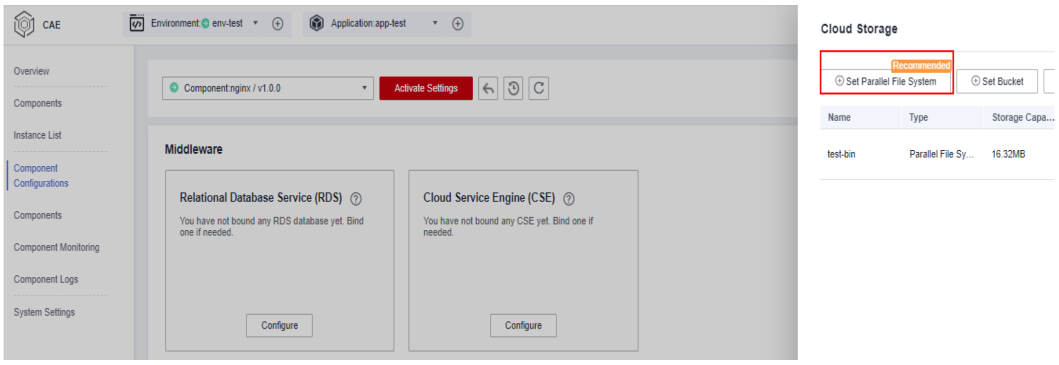
----End

Configuring a Cloud Storage Mount Path

Step 1 Return to CAE and choose **Component Configurations**.

Step 2 Select the Nginx component from the drop-down list in the upper part of the page.

Step 3 Click **Edit** in the **Cloud Storage** module.



You have not bound any RDS database yet. Bind one if needed.

You have not bound any CSE yet. Bind one if needed.

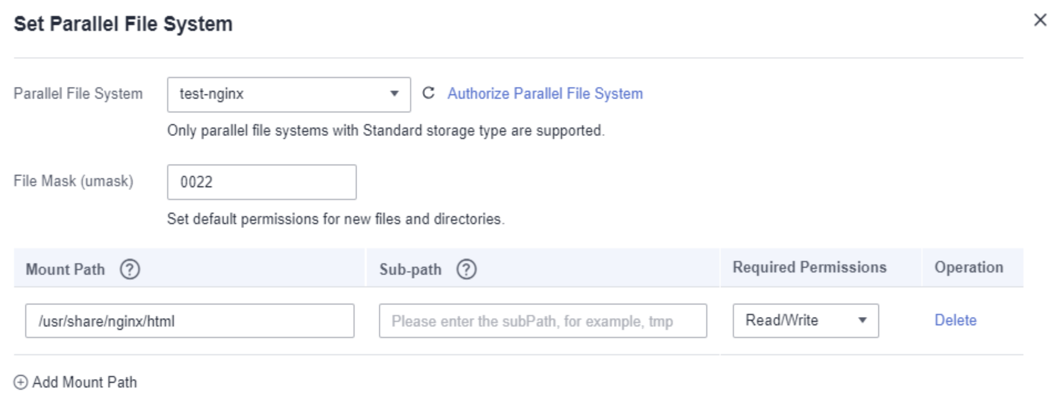
Name	Type	Storage Capa...
test-bin	Parallel File Sy...	16.32MB

Figure 1-14 Configuring cloud storage

Step 4 Click **Set Parallel File System**, enter the mount path, and set permissions.

- **Parallel File System**: Select **test-nginx** authorized in **Cloud Storage Authorizations**.
- **File Mask (umask)**: Enter **0022**.
- **Mount Path**: path to which the data store is mounted. In this practice, use the default path **/usr/share/nginx/html** of **nginx**.
- **Required Permissions**: permissions on the mount path and files in the mount path. The value can be **Read/Write** or **Read only**. In this example, select **Read/Write**.

Figure 1-15 Configuring a cloud storage mount path



Parallel File System: test-nginx

Only parallel file systems with Standard storage type are supported.

File Mask (umask): 0022

Set default permissions for new files and directories.

Mount Path	Sub-path	Required Permissions	Operation
/usr/share/nginx/html	Please enter the subPath, for example, tmp	Read/Write	Delete

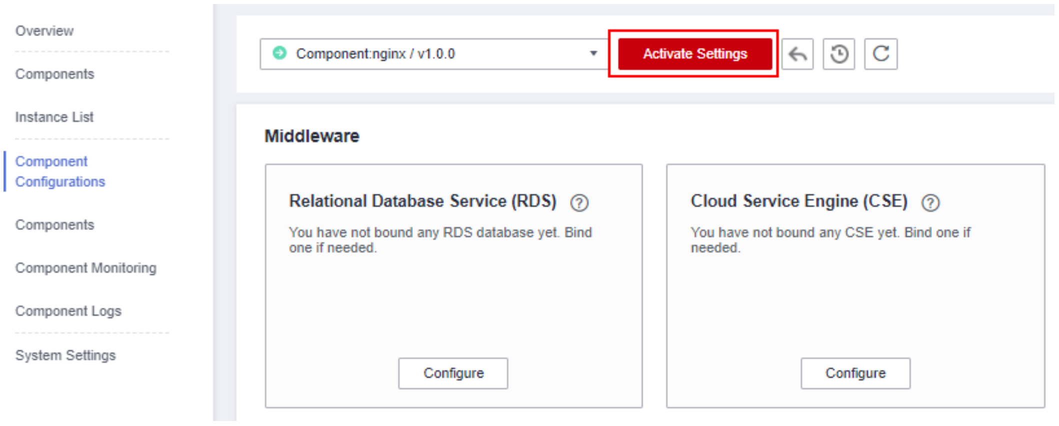
NOTE

- Do not mount a static file path to a directory that contains system files, such as **/** and **/var/run**. Otherwise, the components may be abnormal.
- The **Read/Write** permission indicates that the component has the read and write permissions on the mount path and all files in the path. The **Read only** permission indicates that the component has only the read permission.

Step 5 Click OK. On the **Cloud Storage** page, click **OK** again.

Step 6 On the **Component Configurations** page, click **Activate Settings**.

Figure 1-16 Activate Settings



You have not bound any RDS database yet. Bind one if needed.

You have not bound any CSE yet. Bind one if needed.

----End

Viewing an Updated Page

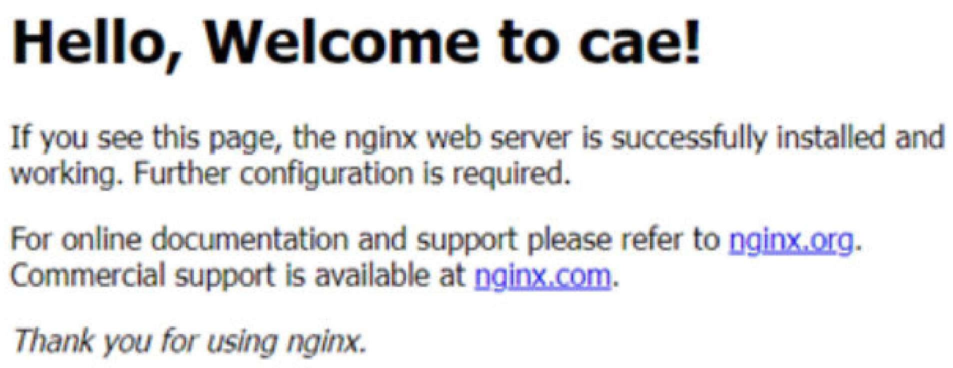
Step 1 Choose **Components**.

Step 2 Click the IP address in the **Access Address** column of the Nginx component to access the Nginx again. The page can be updated in real time (Press **F5**).

NOTE

The new **index.html** file can be customized and uploaded as required.

Figure 1-17 Updated static page



Hello, Welcome to cae!

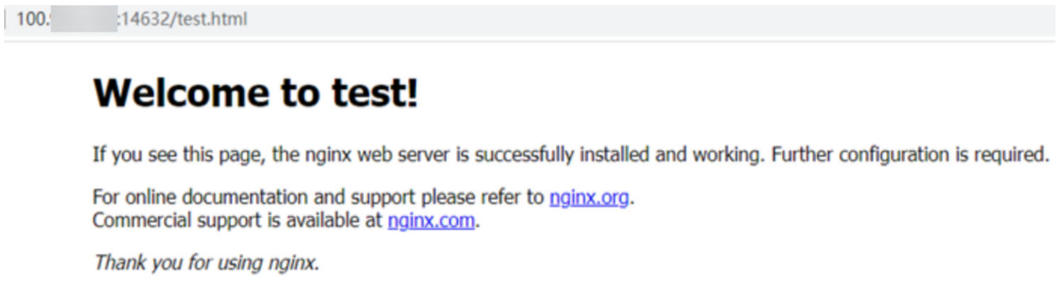
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

You can also access the new static page file using **/test.html**.

Figure 1-18 Accessing the test page


100.[redacted]:14632/test.html

Welcome to test!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

----End

2 Graceful Startup of a Spring Cloud Application

Overview

This section describes how to configure graceful startup for Spring Cloud application to perform operations such as upgrade, restart, and scale-out during component O&M.

Prerequisites

- You have **created an environment**.
- You have **created an application**.
- You have **created and deployed a Spring Cloud component**.

Procedure

Step 1 Enable spring-boot-starter-actuator for Spring Cloud.

1. Add the **spring-boot-starter-actuator** dependency to the **pom.xml** file of the source code corresponding to the component. (Version 2.3.0 or later is recommended.)

```
<dependency>
   <groupId>org.springframework.boot</groupId>
   <artifactId>spring-boot-starter-actuator</artifactId>
   <version>{Your Spring boot version}</version>
</dependency>
```

2. Add the following configurations to the **application.properties** file:

Table 2-1 application.properties

Spring Boot Version	Configuration
2.3.0/2.3.1	management.health.probes.enabled=true

>= 2.3.2	management.endpoint.health.probes.enabled=true
	management.health.livenessState.enabled=true
	management.health.readinessState.enabled=true

3. Update code.
   - If your component is deployed using source code, update the modified source code to the source code repository.
   - If your component is deployed using a software package, pack the new code into a software package and upload the new package to the software package repository.
4. Use the new source code or software package to **upgrade the component**.

Step 2 Configure component health check.

1. Log in to CAE. Choose **Component Configurations**.
2. Select the target component from the drop-down list in the upper part of the page.
3. Configure readiness probe by referring to **Configuring Health Check**. The parameters are as follows:

Table 2-2 Readiness probe parameters

Parameter	Description
Check Method	Select **HTTP**.
Port	Enter the listening port of your component.
Path	– For Spring Boot 2.3 and later, enter **/actuator/health/readiness**. – For Spring Boot versions earlier than 2.3, enter **/actuator/health**.
Protocol	Select the protocol of your component.
Check Interval	Enter **10**.
Latency	Enter **0**.
Timeout Interval	Enter **1**.
Success Threshold	Enter **1**.
Failure Threshold	Enter **3**.

4. Upgrade the component. For details, see **Upgrading a Component**.

Step 3 Verify the configuration.

1. Choose **Components**. If the status of the target component is **Running**, go to the next step. Otherwise, the configuration fails.
2. Choose **Component Events** and select the target component from the drop-down list in the upper part of the page. If the **Component instance healthy** event exists, the configuration is successful. Otherwise, the configuration fails.

----End

3 Health Check

3.1 Overview

Health check is used to check whether your application instances are working properly. It is a mechanism used to ensure normal service running. CAE provides three health check mechanisms: liveliness probe, readiness probe, and startup probe.

- Liveliness probe is used to check whether applications are alive. If an instance is abnormal, Kubernetes deletes the instance and performs the detection again until the instances are normal.

  NOTICE

  - When only the liveliness probe is used, if the network fluctuates or the program starts slowly, the instance will keep restarting and remains in the **Not ready** state.

    The following solutions are available:

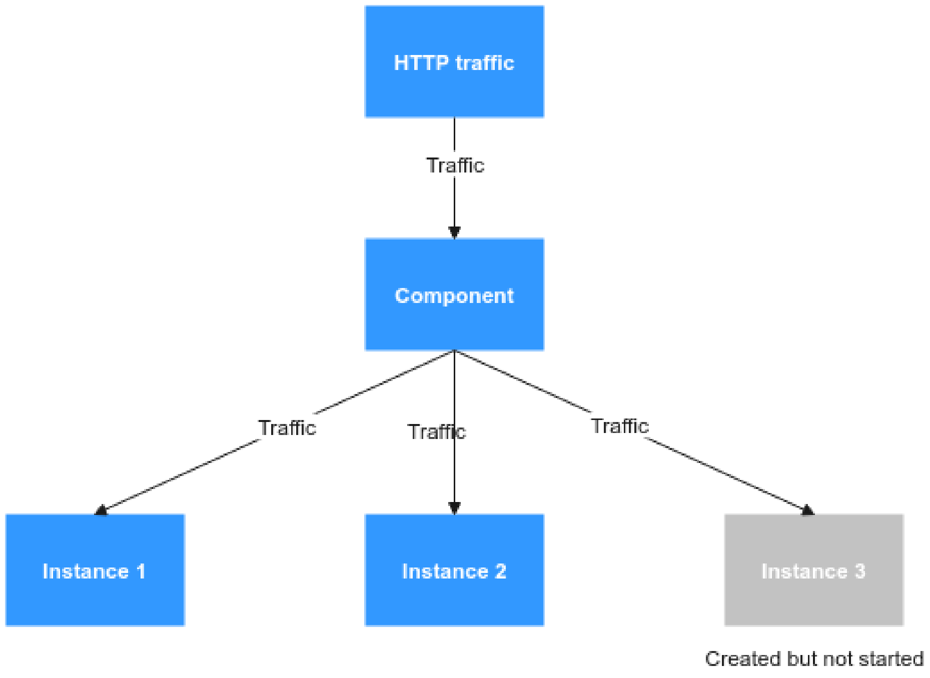
    - Use startup probe together. For details, see **Cooperation Between Startup and Liveliness Probes**.
    - Increase **Failure Threshold** to increase the fault tolerance rate and increase **Latency** to ensure that the program accepts the liveliness probe detection after startup.
  - If status code 200 is returned, the check is successful.
  - If a status code other than 200 is returned and the number of consecutive failures reaches **Failure Threshold**, the check fails.

- Readiness probe is used to check whether an application has been started and is ready to receive requests. If the instance is healthy, traffic will be switched.

  For example, if the number of component instances is increased from 2 to 3, the comparison before and after the readiness probe is configured is as follows:

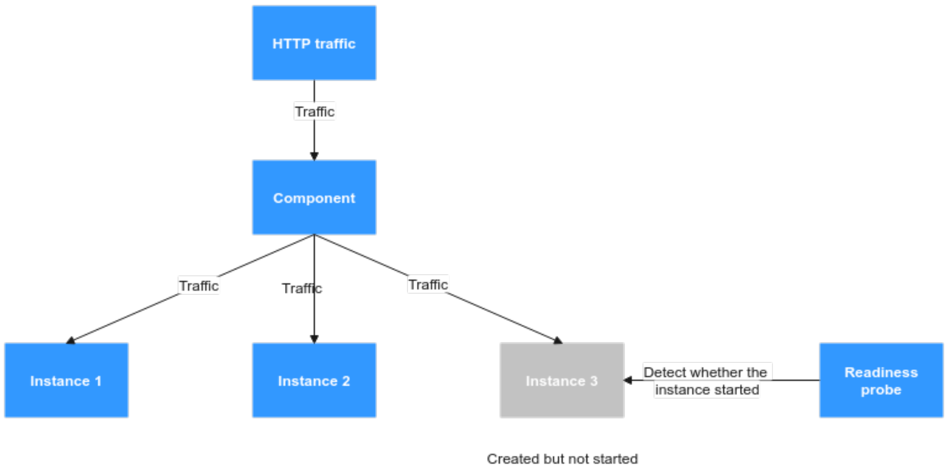
a. When the readiness probe is not configured, the instance has been created, but are not ready to receive traffic due to program reasons. In this case, some traffic still enters instance 3, as shown in **Figure 3-1**.

Figure 3-1 Readiness probe not configured



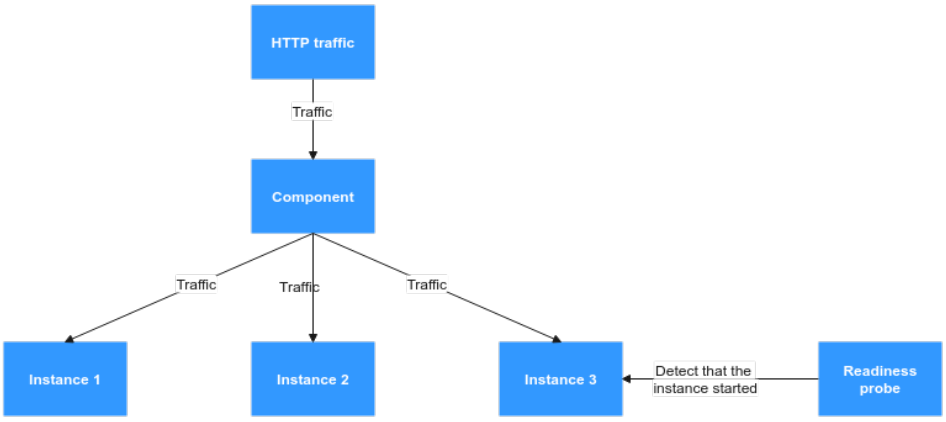
b. After the readiness probe is configured, it detects that instance 3 is not started or ready. So instance 3 does not receive traffic, ensuring that all traffic flows to healthy instances 1 and 2.

Figure 3-2 Before health check



c. When detecting that instance 3 is healthy, the readiness probe allows the traffic to pass through and stops the detection.

Figure 3-3 After health check



- When the startup probe is running, readiness and liveliness probes are disabled. If the startup probe health check fails, the instance will be restarted.

  You are advised to use the startup probe together with the liveliness probe.

3.2 Cooperation Between Startup and Liveliness Probes

Prerequisites

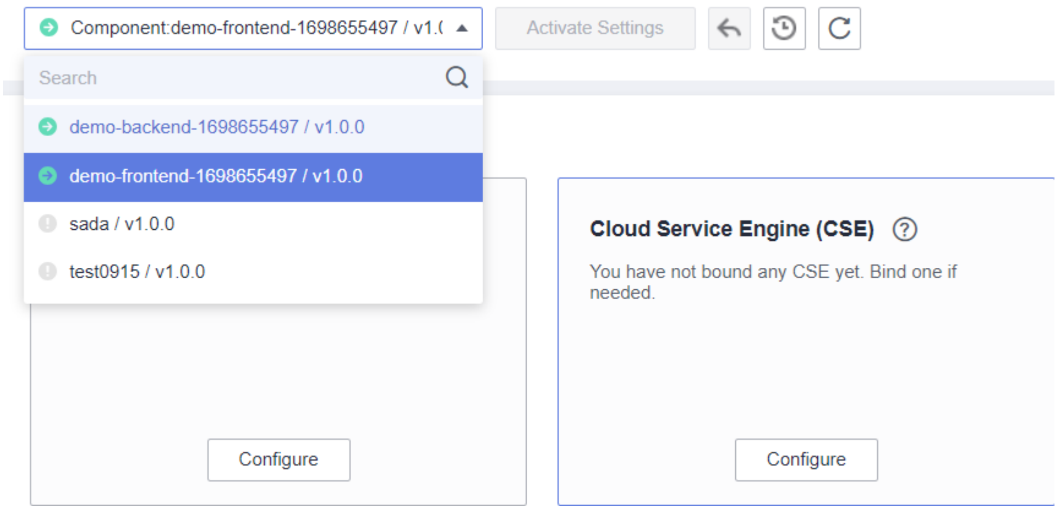
- You have **created an environment**.
- You have **created an application**.
- You have **created and deployed a component**. In this example, use the demo-frontend component.

Procedure

Step 1 Log in to CAE. Choose **Component Configurations**.

Step 2 Select the target component from the drop-down list in the upper part of the page.

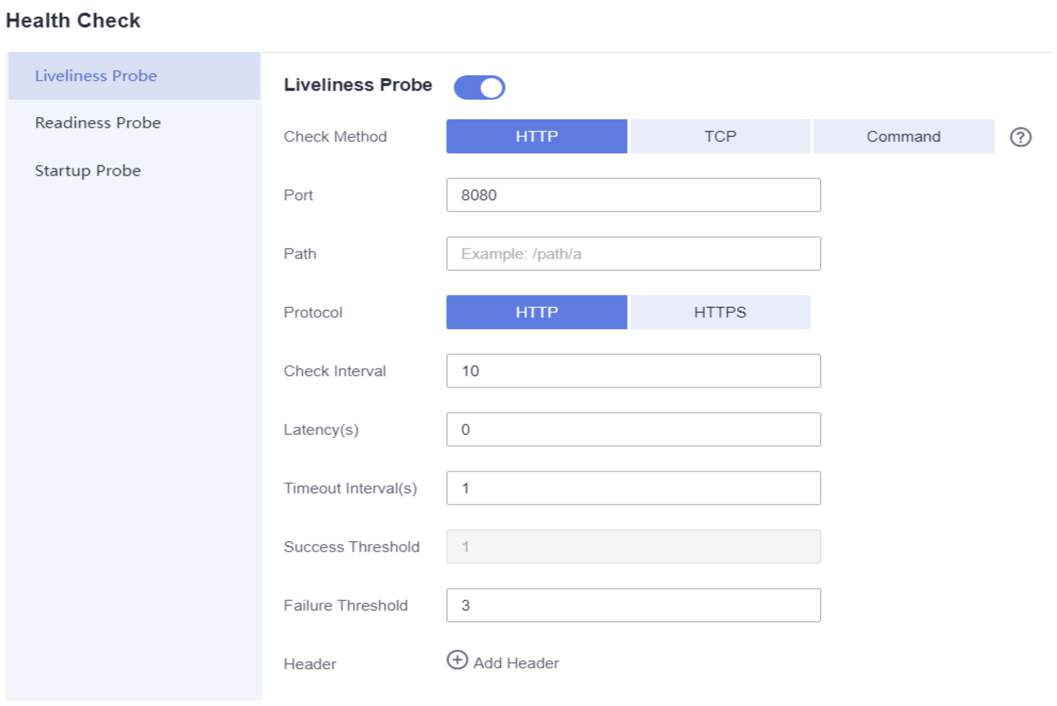
Figure 3-4 Selecting a component


Component:demo-frontend-1698655497 / v1.(
Activate Settings
Search
demo-backend-1698655497 / v1.0.0
demo-frontend-1698655497 / v1.0.0
sada / v1.0.0
test0915 / v1.0.0
Cloud Service Engine (CSE)
Configure
Configure

You have not bound any CSE yet. Bind one if needed.

Step 3 Configure the startup and liveliness probes by referring to **Figure 3-5** and **Figure 3-6**, and make the configurations take effect. For details, see **Configuring Health Check**.

Figure 3-5 Configuring liveliness probe

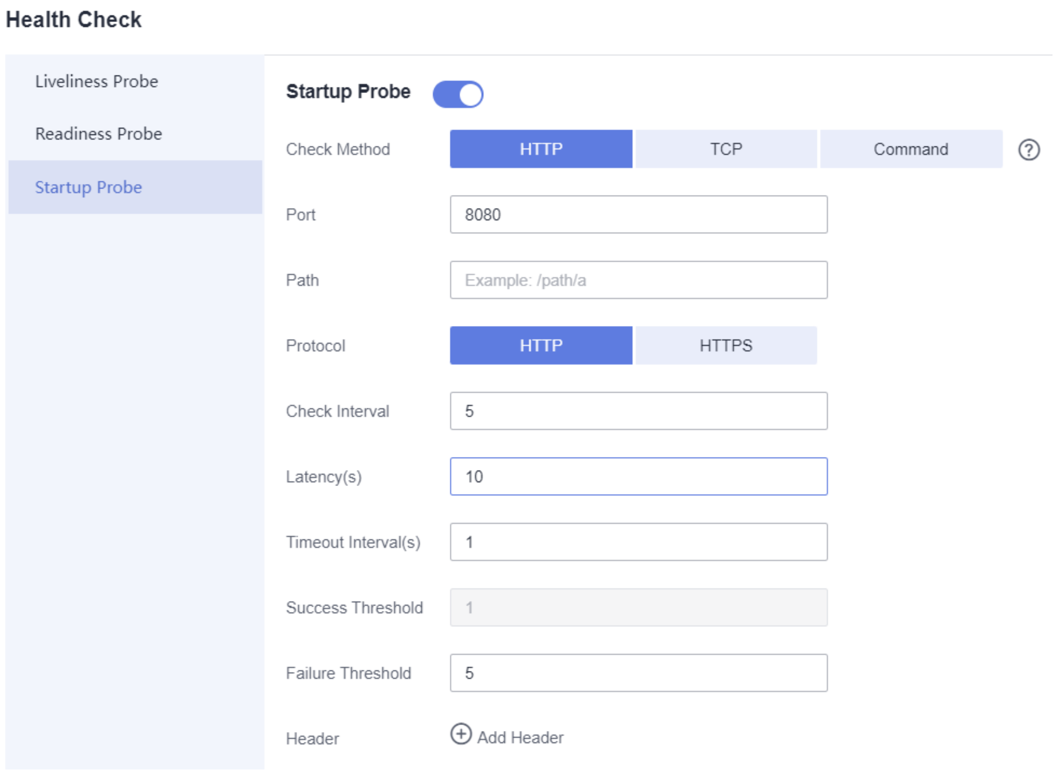

Health Check

Liveliness Probe | Readiness Probe | Startup Probe

Liveliness Probe (on)

Check Method	[✓ HTTP] TCP Command
Port	8080
Path	Example: /path/a
Protocol	[✓ HTTP] HTTPS
Check Interval	10
Latency(s)	0
Timeout Interval(s)	1
Success Threshold	1
Failure Threshold	3
Header	Add Header

Figure 3-6 Configuring startup probe


Health Check

Liveliness Probe | Readiness Probe | Startup Probe

Startup Probe (on)

Check Method	[✓ HTTP] TCP Command
Port	8080
Path	Example: /path/a
Protocol	[✓ HTTP] HTTPS
Check Interval	5
Latency(s)	10
Timeout Interval(s)	1
Success Threshold	1
Failure Threshold	5
Header	Add Header

The startup probe will check an instance every 5 seconds, starting 10 seconds after instance creation. If the check fails for five consecutive times, the container will be restarted. After the startup probe detects that the instance is healthy, the liveliness probe starts detection to prevent the instance from keeping restarting due to slow program startup.

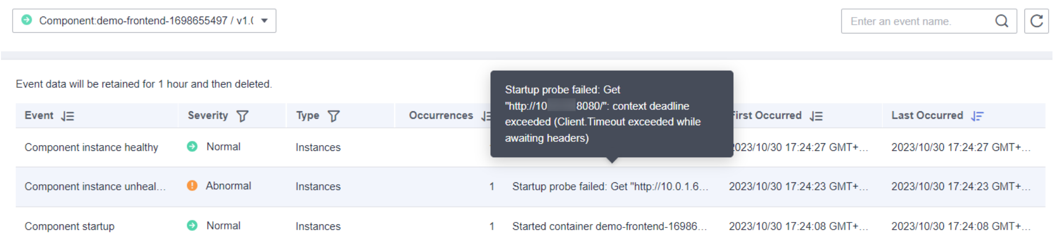
NOTICE

Ensure that the program can start within the following period: **Latency** + **Check Interval** x **Failure Threshold**. Otherwise, the startup probe will keep restarting the instance. If you are not sure about the program startup time, increase **Failure Threshold** and **Latency**.

As shown in **Figure 3-6**, the program needs to start within 35 seconds (10 + 5 x 5 = 35).

Step 4 Click **Component Events**. As shown on the page, the program starts after one startup probe detection failure. Then the liveliness probe is switched to and detects that the instance is healthy.

Figure 3-7 Viewing events

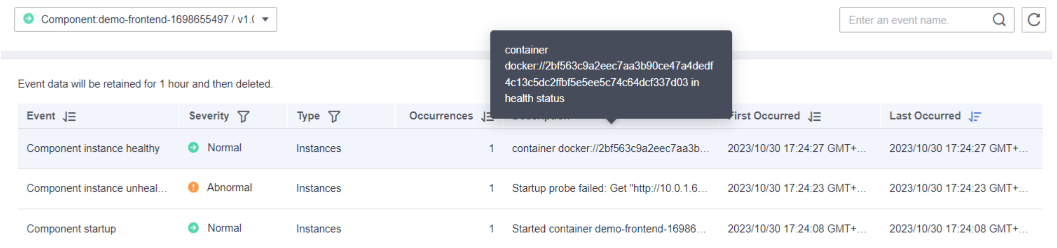


Event data will be retained for 1 hour and then deleted.

Startup probe failed: Get "http://10[redacted]8080/": context deadline exceeded (Client.Timeout exceeded while awaiting headers)

Event	Severity	Type	Occurrences		irst Occurred	Last Occurred
Component instance healthy	Normal	Instances			023/10/30 17:24:27 GMT+...	2023/10/30 17:24:27 GMT+...
Component instance unheal...	Abnormal	Instances	1	Startup probe failed: Get "http://10.0.1.6...	2023/10/30 17:24:23 GMT+...	2023/10/30 17:24:23 GMT+...
Component startup	Normal	Instances	1	Started container demo-frontend-16986...	2023/10/30 17:24:08 GMT+...	2023/10/30 17:24:08 GMT+...

Figure 3-8 Startup probe detection



Event data will be retained for 1 hour and then deleted.

container docker://2bf563c9a2eec7aa3b90ce47a4dedf4c13c5dc2ffbf5e5ee5c74c64dcf337d03 in health status

Event	Severity	Type	Occurrences		irst Occurred	Last Occurred
Component instance healthy	Normal	Instances	1	container docker://2bf563c9a2eec7aa3b...	2023/10/30 17:24:27 GMT+...	2023/10/30 17:24:27 GMT+...
Component instance unheal...	Abnormal	Instances	1	Startup probe failed: Get "http://10.0.1.6...	2023/10/30 17:24:23 GMT+...	2023/10/30 17:24:23 GMT+...
Component startup	Normal	Instances	1	Started container demo-frontend-16986...	2023/10/30 17:24:08 GMT+...	2023/10/30 17:24:08 GMT+...

----End

3.3 Using Readiness Probe to Ensure Normal Traffic During Upgrade

Prerequisites

- You have **created an environment**.
- You have **created an application**.

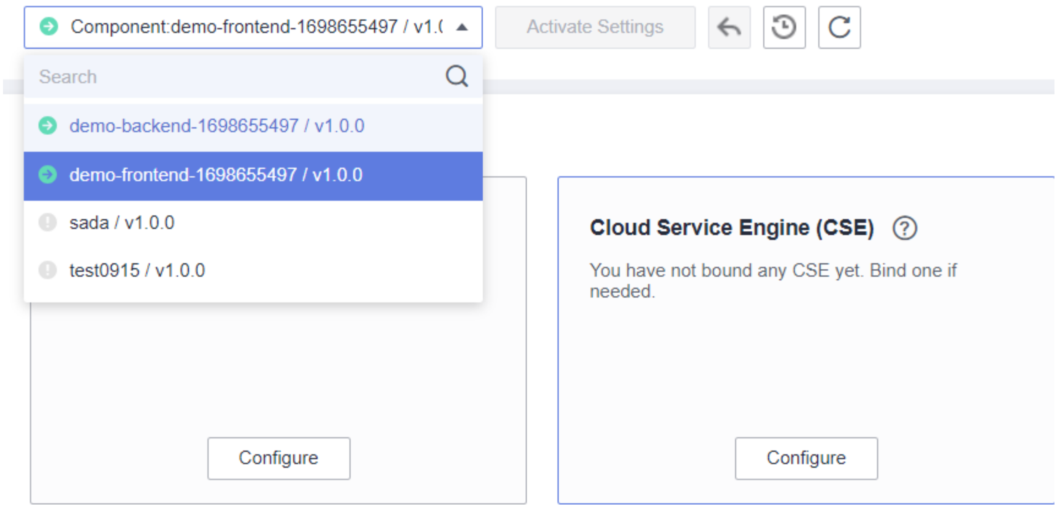
- You have **created and deployed a component**. In this example, use the demo-frontend component.
- You have configured the **access mode** and the configuration has taken effect.

Procedure

Step 1 Log in to CAE. Choose **Component Configurations**.

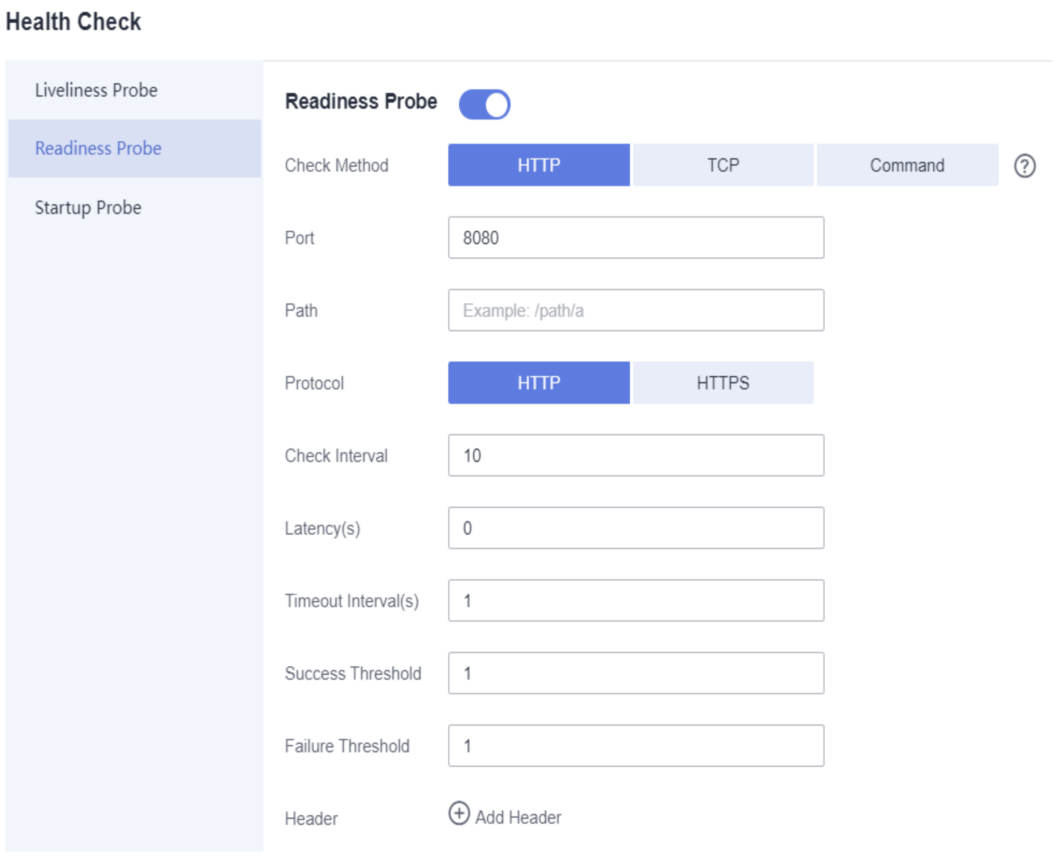
Step 2 Select the target component from the drop-down list in the upper part of the page.

Figure 3-9 Selecting a component


Component:demo-frontend-1698655497 / v1.(
Activate Settings
Search
demo-backend-1698655497 / v1.0.0
demo-frontend-1698655497 / v1.0.0
sada / v1.0.0
test0915 / v1.0.0
Cloud Service Engine (CSE)
You have not bound any CSE yet. Bind one if needed.
Configure
Configure

Step 3 Configure the readiness probes by referring to **Figure 3-10**, and make the configurations take effect. For details, see **Configuring Health Check**.

Figure 3-10 Configuring readiness probe



Health Check

Liveliness Probe
Readiness Probe
Startup Probe

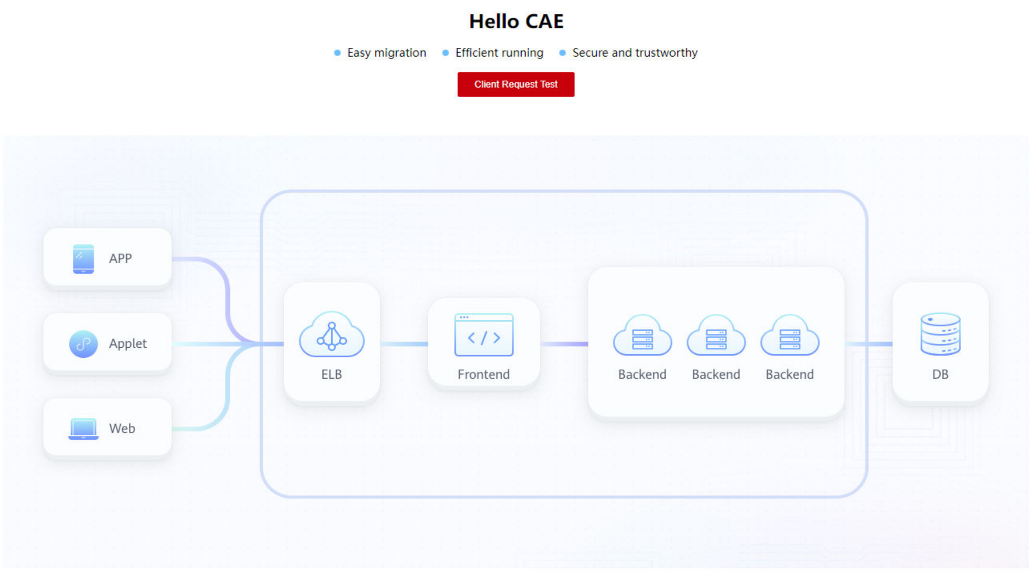
Readiness Probe

Check Method	HTTP / TCP / Command (HTTP selected)
Port	8080
Path	Example: /path/a
Protocol	HTTP / HTTPS (HTTP selected)
Check Interval	10
Latency(s)	0
Timeout Interval(s)	1
Success Threshold	1
Failure Threshold	1
Header	Add Header

Step 4 Upgrade the component. For details, see **Upgrading a Component**.

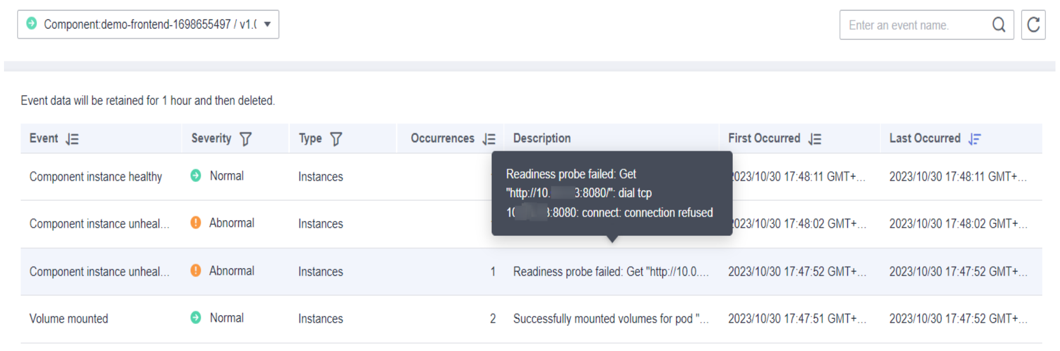
Step 5 During upgrade, choose **Components**, click the IP address in the **Access Address** column of the component to view the application page. As shown in **Figure 3-11**, services are not interrupted.

Figure 3-11 Application page



Step 6 As shown on the **Component Events** page, the instance has unhealthy requests. The probe detects that the new instance is not ready for traffic switchover and continues to use the old instance to provide services.

Figure 3-12 Viewing component events



Event data will be retained for 1 hour and then deleted.

Event	Severity	Type	Occurrences	Description	First Occurred	Last Occurred
Component instance healthy	Normal	Instances			023/10/30 17:48:11 GMT+...	2023/10/30 17:48:11 GMT+...
Component instance unheal...	Abnormal	Instances			023/10/30 17:48:02 GMT+...	2023/10/30 17:48:02 GMT+...
Component instance unheal...	Abnormal	Instances	1	Readiness probe failed: Get "http://10.0....	2023/10/30 17:47:52 GMT+...	2023/10/30 17:47:52 GMT+...
Volume mounted	Normal	Instances	2	Successfully mounted volumes for pod "...	2023/10/30 17:47:51 GMT+...	2023/10/30 17:47:52 GMT+...

Readiness probe failed: Get "http://10.[redacted]3:8080/": dial tcp 1[redacted]:8080: connect: connection refused

----End

4 Lifecycle Management

4.1 Overview

Lifecycle management is a method used to execute calls at a specific stage. CAE provides two lifecycle management modes: post-start and pre-stop processing.

- Post-start processing: After a component instance starts, a post-start event is triggered immediately. However, it is not ensured that the corresponding handler can be executed before EntryPoint of the container.

  NOTE

  The status of the component instance changes to **Running** only after the post-start function is executed. Therefore, when the command is set to an infinite loop, the CAE component status cannot change to **Running**.

- Pre-stop processing: Before a component instance stops, a pre-stop event is sent.

  NOTE

  A pre-stop event is sent immediately before a component instance stops. Unless the grace period of the instance expires, the component instance is blocked until the function is executed.

4.2 Writing Files Using Post-start Processing

Prerequisites

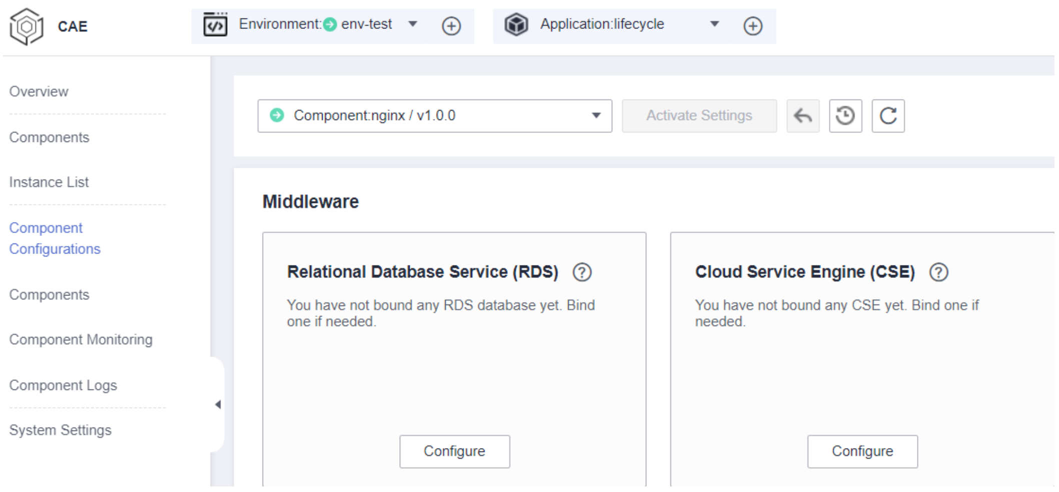
- You have **created an environment**.
- You have **created an application**.
- You have **created and deployed a component**. In this example, use the nginx component.

Procedure

Step 1 Log in to CAE. Choose **Component Configurations**.

Step 2 Select the target component from the drop-down list in the upper part of the page.

Figure 4-1 Selecting a component

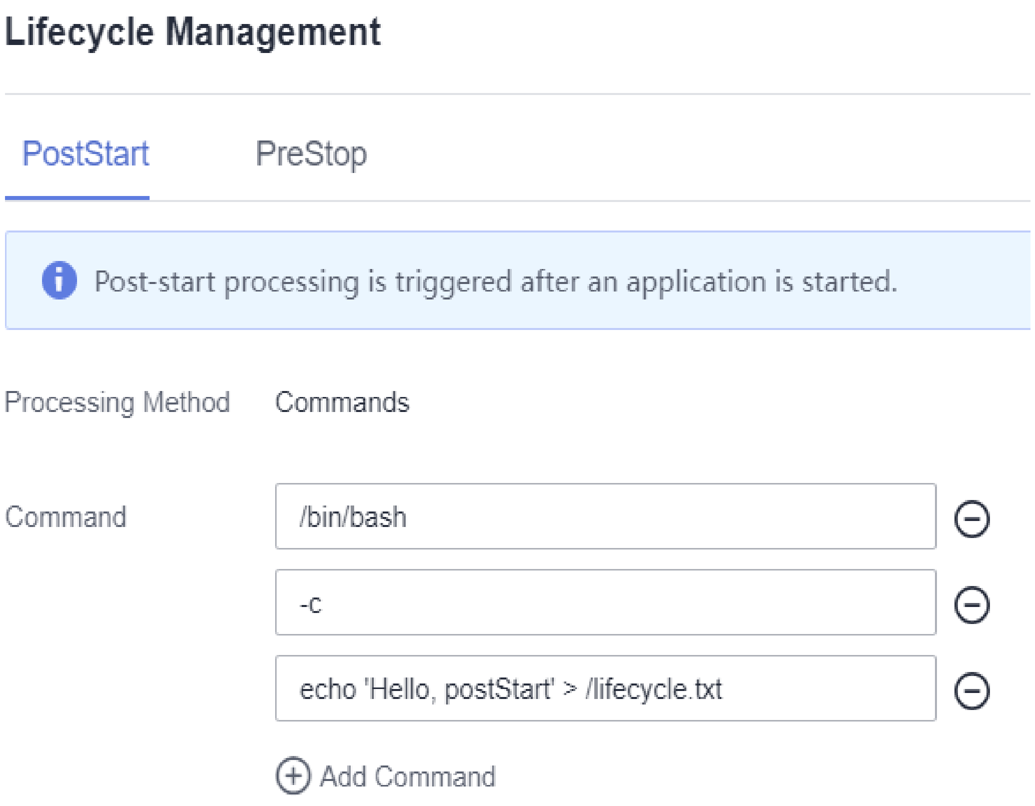


Step 3 Configure post-start processing by referring to **Figure 4-2**. For details, see **Configuring the Lifecycle**.

Run the following commands:

```
/bin/bash
-c
echo 'Hello, postStart' > /lifecycle.txt
```

Figure 4-2 Configuring the lifecycle



```
/bin/bash
-c
echo 'Hello, postStart' > /lifecycle.txt
```

Step 4 Click **Save**.

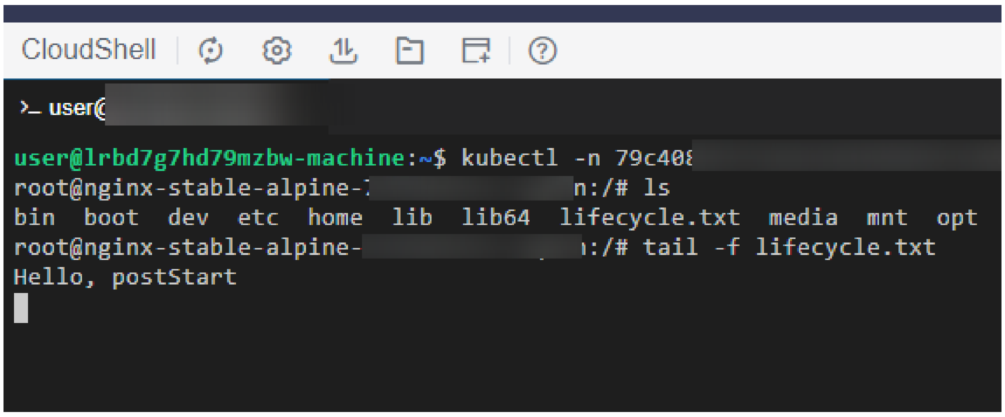
Step 5 Click **Activate Settings** in the upper part of the page. In the dialog box displayed on the right, confirm the configurations and click **OK** for the configurations to take effect.

Step 6 Use CloudShell to log in to the container and check whether the file content takes effect.

1. Choose **Instance List**.
2. Select the target environment and application from the drop-down lists in the upper part of the page, and click the target component.
3. Select a running instance and click **Remote Login** in the **Operation** column. Go to the container and check whether the file content takes effect.
4. View the file content and check whether the file is successfully written.

```
tail -f lifecycle.txt
```

Figure 4-3 Logging in to the container and viewing the file content



```
>_ user@[redacted]
user@lrbd7g7hd79mzbw-machine:~$ kubectl -n 79c408[redacted]
root@nginx-stable-alpine-7[redacted]n:/# ls
bin  boot  dev  etc  home  lib  lib64  lifecycle.txt  media  mnt  opt
root@nginx-stable-alpine-[redacted]:/# tail -f lifecycle.txt
Hello, postStart
```

----End

4.3 Gracefully Stopping Nginx Using Pre-stop Processing

If the container will be stopped by the system, pre-stop processing helps your main program perform necessary cleanup tasks before the stop.

Prerequisites

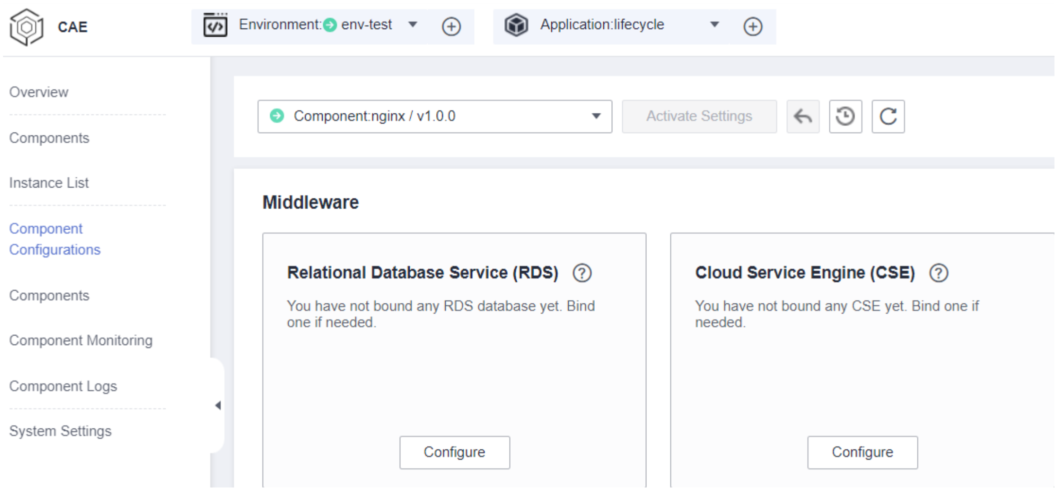
- You have **created an environment**.
- You have **created an application**.
- You have **created and deployed a component**. In this example, use the nginx component.

Procedure

Step 1 Log in to CAE. Choose **Component Configurations**.

Step 2 Select the target component from the drop-down list in the upper part of the page.

Figure 4-4 Selecting a component

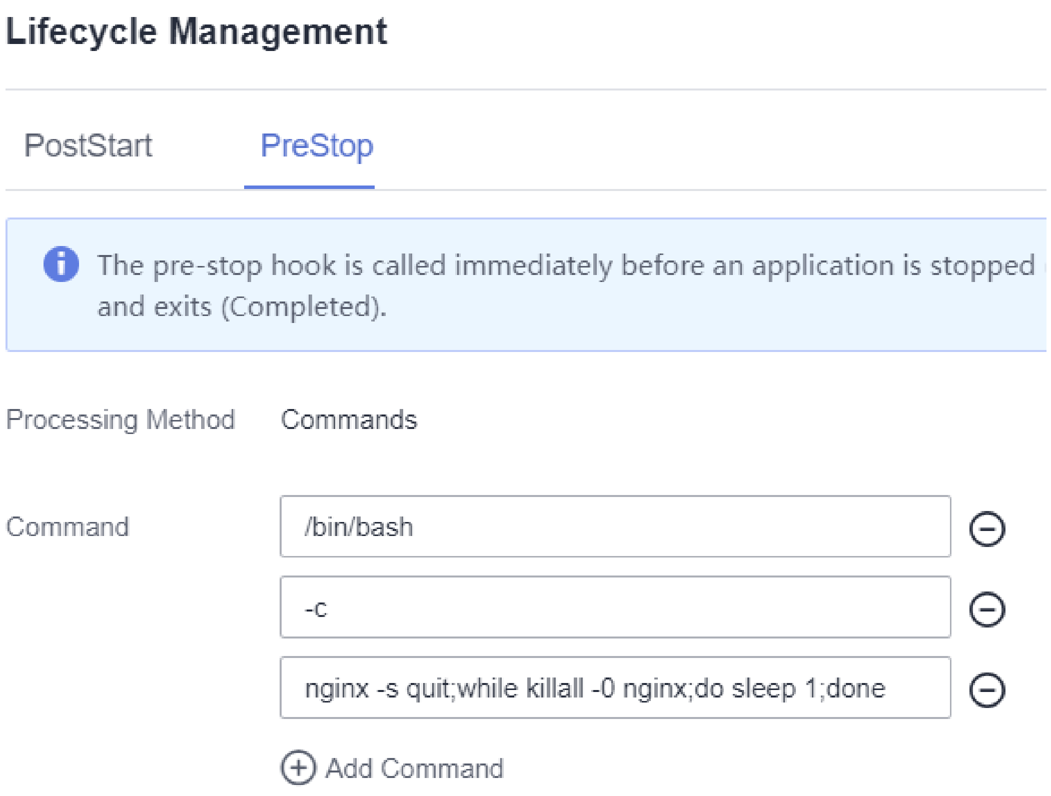


Step 3 Configure pre-stop processing by referring to **Figure 4-5**. For details, see **Configuring the Lifecycle**.

Run the following commands:

```
/bin/bash
-c
nginx -s quit;while killall -0 nginx;do sleep 1;done
```

Figure 4-5 Configuring the lifecycle


Lifecycle Management
PostStart
PreStop
The pre-stop hook is called immediately before an application is stopped and exits (Completed).
Processing Method
Commands
Command
/bin/bash
-c
nginx -s quit;while killall -0 nginx;do sleep 1;done
Add Command

Step 4 Click **Save**.

Step 5 Click **Activate Settings** in the upper part of the page. In the dialog box displayed on the right, confirm the configurations and click **OK** for the configurations to take effect.

----End

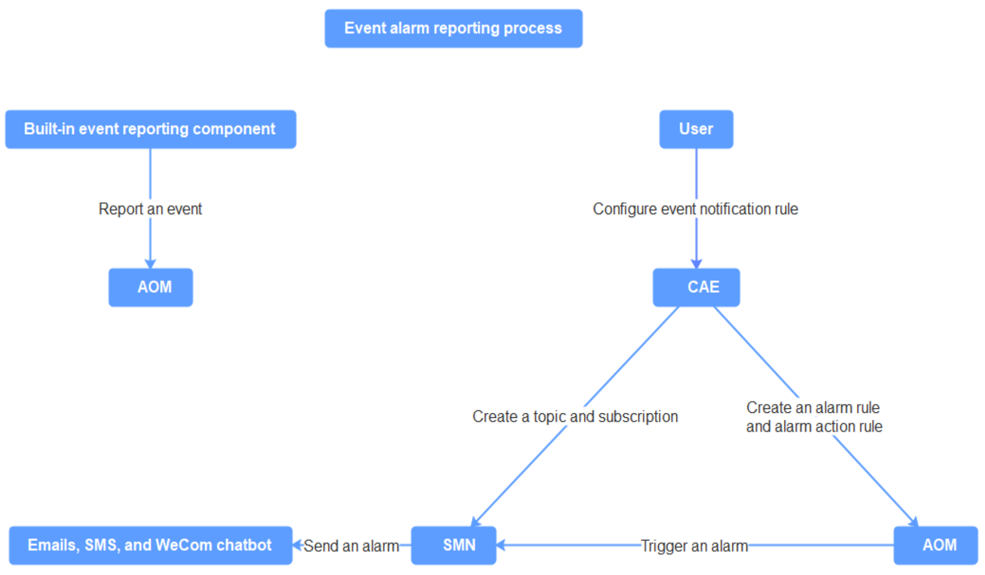
5 Sending Event Alarms to WeCom

Overview

CAE can send notifications when instance scheduling, health check, image pull, volume mounting, or container startup succeeds or fails. By configuring event notification rules, you can learn about the component running status in a timely manner and quickly locate faults. The configuration of event notification rules depends on the AOM and SMN services. Component instance events are reported to AOM. You can choose **Application O&M Management** > **Alarm Management** > **Alarm List** > **Events** to view the reported events. SMN is the sender of alarm messages.

By default, CAE reports component instance events to AOM. After configuring event notification rules, CAE creates topics and subscribers in SMN, and creates alarm rules and alarm action rules in AOM.

Figure 5-1 Event alarm reporting process


Event alarm reporting process
Built-in event reporting component
Report an event
AOM
User
Configure event notification rule
CAE
Create a topic and subscription
Create an alarm rule and alarm action rule
Emails, SMS, and WeCom chatbot
Send an alarm
SMN
Trigger an alarm
AOM

Prerequisites

SMN allows you to subscribe to WeCom, DingTalk, and Lark group messages. Currently, these functions are in the open beta test (OBT). You need to apply for the OBT qualification first.

- You have enabled WeCom and created a group chatbot.
- You have **submitted a service ticket** to apply for the SMN OBT qualification on Huawei Cloud and enabled the OBT.

Procedure

Step 1 **Log in to CAE** and choose **System Settings**.

Step 2 Click **Edit** in the **Event Notification Rules** module.

Step 3 Click **Create Event Notification Rule** and configure basic information by referring to **Table 5-1**.

Table 5-1 Configuring basic information

Parameter	Description
Name	Enter an event notification rule name. For example, **container-Initiate**. The value starts and ends with a letter and contains 1 to 64 characters, including letters, digits, hyphens (-), and underscores (_).
Trigger Event	Select an event that triggers notification from the drop-down list. For example, **Container started up**.
Effected Components	Select **All in the environment**.
Alarm Policy	**Trigger Mode**: Select **Immediate**.

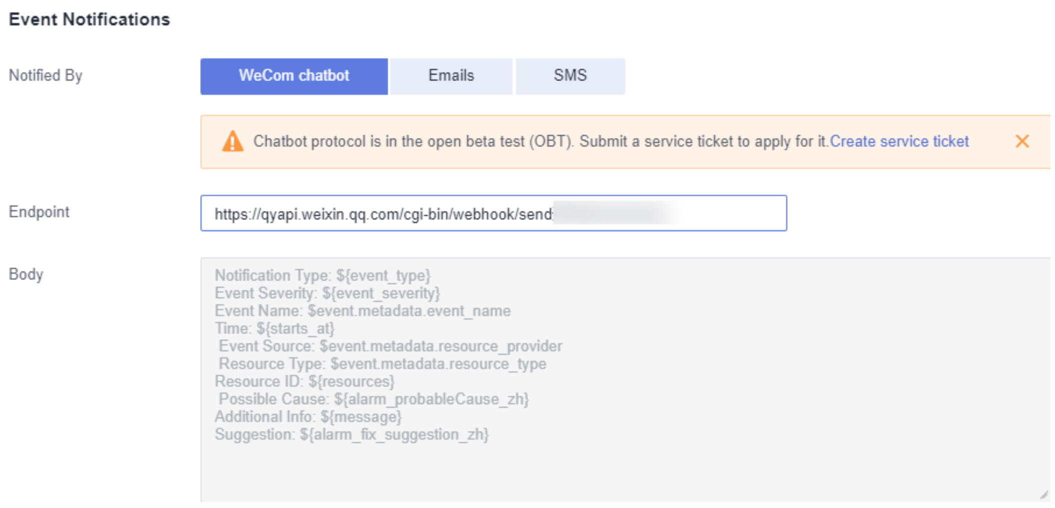
Step 4 Select **WeCom chatbot** for **Notified By**.

Enter the WeCom chatbot endpoint address, which is a webhook address starting with **https://qyapi.weixin.qq.com/cgi-bin/webhook/send**.

NOTE

For details about how to obtain a WeCom subscription endpoint, see **How Does DingTalk, Lark, or WeCom Chatbot Obtain Subscription Endpoints?**

Figure 5-2 Configuring event notification



Chatbot protocol is in the open beta test (OBT). Submit a service ticket to apply for it.Create service ticket

https://qyapi.weixin.qq.com/cgi-bin/webhook/send

```
Notification Type: ${event_type}
Event Severity: ${event_severity}
Event Name: $event.metadata.event_name
Time: ${starts_at}
 Event Source: $event.metadata.resource_provider
 Resource Type: $event.metadata.resource_type
Resource ID: ${resources}
 Possible Cause: ${alarm_probableCause_zh}
Additional Info: ${message}
Suggestion: ${alarm_fix_suggestion_zh}
```

Step 5 Click **OK**.

Step 6 In the component list, select the target component and click **More** > **Restart** in the **Operation** column.

After the component is restarted, log in to AOM to view the event list or receive alarms in your WeCom group.

- Log in to **AOM** to view the list of events reported by CAE.
  a. Choose **Alarm Management** > **Alarm List** > **Events**.
  b. Select **Select all** for **Event Severity**.
  c. Set the filter criterion to **Event Source: CAE** and click 🔍.
- Receive alarms in your WeCom groups.

----End

6 Configuring PromQL to Implement Custom Auto Scaling

Assume that there is a component named **my_component**, its environment is **my_environment**, and the application is **my_application**.

Assume that the component provides the custom metric **http_requests_total**, which indicates the total number of HTTP requests. This section uses this metric as an example to describe how to use PromQL.

Querying Metrics

Run the following PromQL statement to query the latest metric:

```
http_requests_total{environment_name="my_environment",application_name="my_application",component_name="my_component"}
```

Run the following command to query all metrics in the last 5 minutes:

```
http_requests_total{environment_name="my_environment",application_name="my_application",component_name="my_component"}[5m]
```

Processing Queried Metrics

Querying Metrics contains multiple metrics queried. For example, if a component has multiple instances, there are multiple pieces of metrics; if metrics have been queried for a period of time, there are multiple pieces of metrics.

PromQL in an AS policy must return a single value. Therefore, you need to process the queried metrics to obtain a single value. Example:

Query the latest metric and calculate the average value to obtain the average value of the total number of HTTP requests of all instances.

```
avg(http_requests_total{environment_name="my_environment",application_name="my_application",component_name="my_component"})
```

Query all metrics in the last 5 minutes, obtain the change (increased value), and calculate the average value to obtain the average number of increased HTTP requests of each instance in the last 5 minutes.

```
avg(delta(http_requests_total{environment_name="my_environment",application_name="my_application",component_name="my_component"}[5m]))
```

For more information, see **PromQL document** and **PromQL example**.

7 Configuring the Interconnection Between CAE and DEW to Help Applications Obtain Encrypted Secrets from DEW

7.1 Overview

Application Scenario

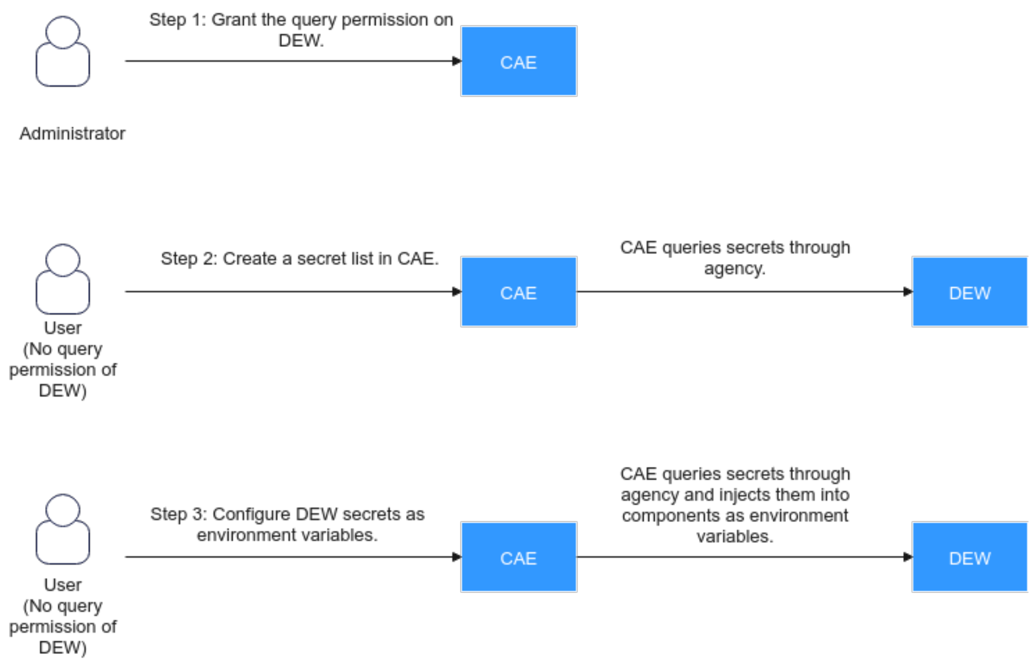
You can connect CAE to DEW to help applications obtain secrets from DEW, which simplifies the connection between applications and DEW and improves security.

If secrets such as access keys and SSH keys need to be injected into your code as environment variables, you can configure DEW secrets to ensure key security.

For details about how to create and manage secrets, see **Cloud Secret Management Service**.

Architecture

Figure 7-1 Process of injecting secrets as environment variables


Step 1: Grant the query permission on DEW.
CAE
Administrator
Step 2: Create a secret list in CAE.
CAE
CAE queries secrets through agency.
DEW
User (No query permission of DEW)
Step 3: Configure DEW secrets as environment variables.
CAE
CAE queries secrets through agency and injects them into components as environment variables.
DEW
User (No query permission of DEW)

Restrictions

You must authorize KMS CMKFullAccess and CSMS ReadOnlyAccess to agency cae_trust. For details, see **Assigning Permissions to a User Group**.

Before enabling this function, you need to perform DEW authorization as an administrator.

7.2 Configuring a Secret and Injecting It as an Environment Variable

In this practice, you can add DEW secrets and import them into components as environment variables to protect data.

Creating a DEW Credential

Step 1 Log in to DEW.

Step 2 Choose **Cloud Secret Management Service**.

Step 3 Click **Create Secret** and set parameters by referring to **Table 7-1**.

Table 7-1 Secret parameters

Parameter	Description
Type	Select **Shared secret**.

Parameter	Description
Secret Name	Enter a secret name. In this practice, enter **db**.
Enterprise Project	ID of the enterprise project to which a secret is bound during creation. In this practice, select **default**.
Secret Value	Select **Plaintext** and enter **123456**.
Description	In this practice, leave it blank.
KMS Encryption Key	Select **csms/default**.
Associated Event	In this practice, select **None**.

Step 4 Click **Next**. The rotation period cannot be set for shared secrets. Click **Next** again to confirm the secret information.

Step 5 Click **OK**.

You can view the created secret in the secret list. The default status of a secret is **Enabled**.

----End

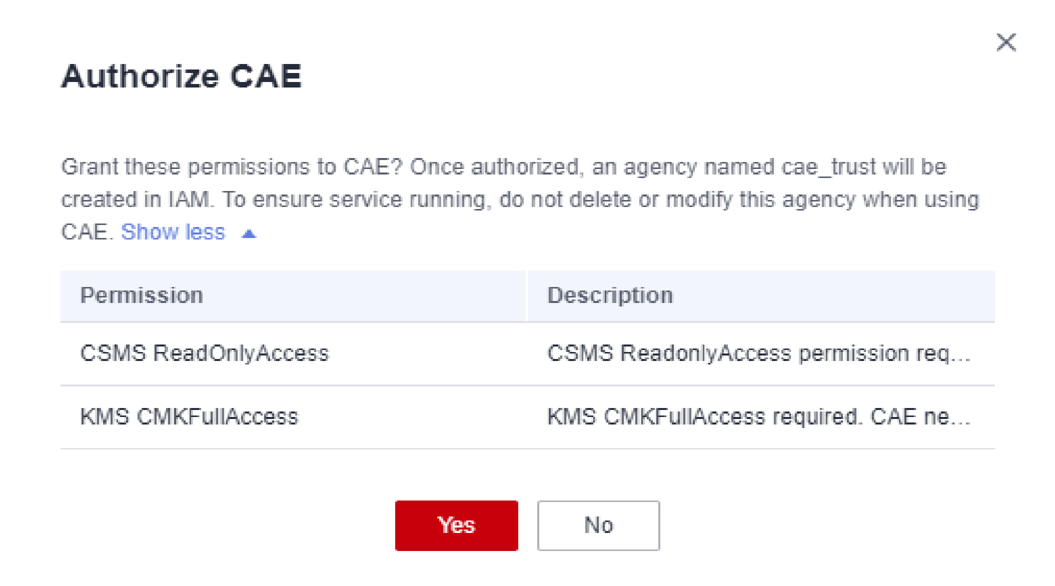
Adding a Secret

Step 1 Log in to CAE.

Step 2 Choose **System Settings**.

If KMS CMKFullAccess and CSMS ReadonlyAccess are not granted, grant them as the administrator.

Figure 7-2 Authorization



Authorize CAE

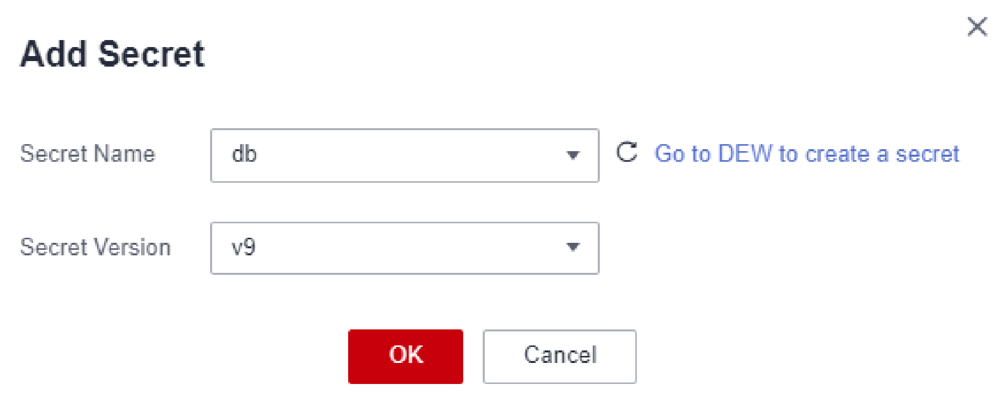
Grant these permissions to CAE? Once authorized, an agency named cae_trust will be created in IAM. To ensure service running, do not delete or modify this agency when using CAE. Show less

Permission	Description
CSMS ReadOnlyAccess	CSMS ReadonlyAccess permission req...
KMS CMKFullAccess	KMS CMKFullAccess required. CAE ne...

Step 3 Click **Edit** in the **Secret Configuration** module.

Step 4 Click **Create Secret**. In the displayed dialog box, select the secret created in **Creating a DEW Secret** and the required version.

Figure 7-3 Adding a secret



Step 5 Click **OK**.

----End

Configuring an Environment Variable

Step 1 Choose **Component Configurations**.

Step 2 Select the target component.

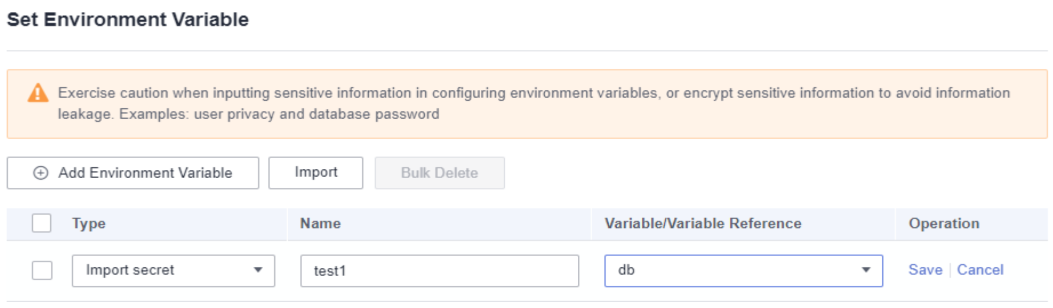
Step 3 Click **Edit** in the **Environment Variables** module.

Step 4 Click **Add Environment Variable** and configure the environment variable by referring to **Table 7-2**.

Table 7-2 Configuring an environment variable

Parameter	Description
Type	Select **Import secret**.
Name	Name of an environment variable, for example, **test1**. The name must be unique.
Variable/Variable Reference	Select the secret created in **Adding a Secret** from the drop-down list.

Figure 7-4 Configuring an environment variable


Set Environment Variable

Exercise caution when inputting sensitive information in configuring environment variables, or encrypt sensitive information to avoid information leakage. Examples: user privacy and database password

Type	Name	Variable/Variable Reference	Operation
Import secret	test1	db	Save | Cancel

Step 5 Click **Save** in the **Operation** column. On the **Set Environment Variable** page, click **OK**.

Step 6 Click **Activate Settings** in the upper part of the page.

In the dialog box displayed on the right, confirm the configurations and click **OK** for the configurations to take effect.

----End

Verifying the Configurations

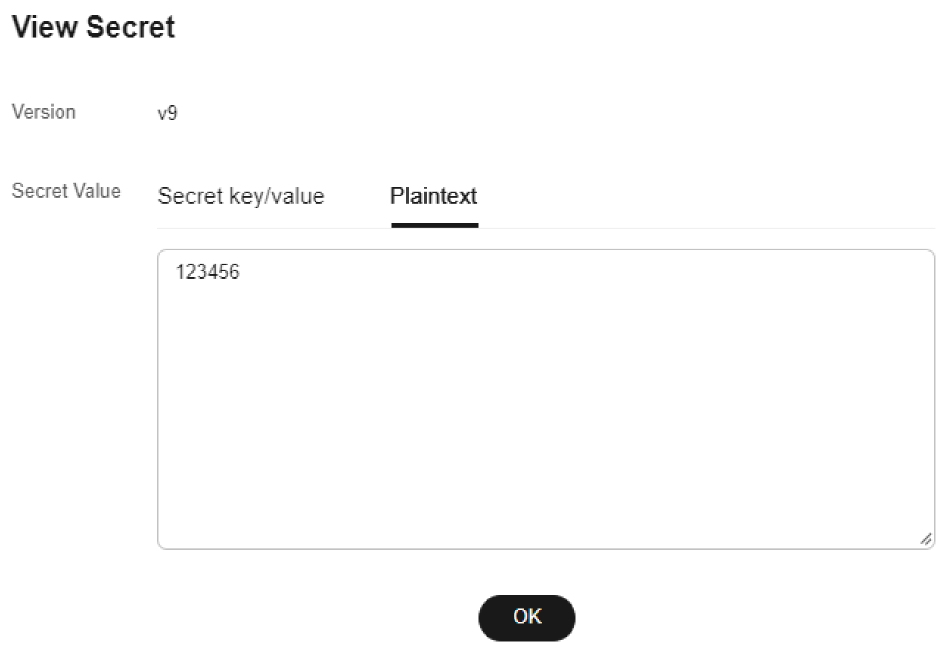
Step 1 Choose **Instance List**.

Step 2 Select the target environment and application from the drop-down lists in the upper part of the page, and click the target component.

Step 3 Select the target instance and click **Remote Login** in the **Operation** column.

Step 4 Check the environment variable, which is the same as the secret value in the DEW secret.

Figure 7-5 Secret value in DEW

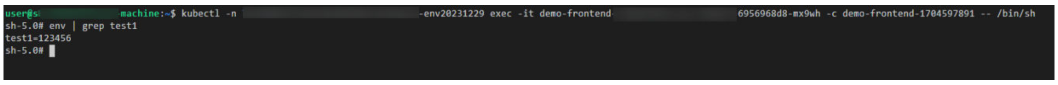

View Secret

Version v9

Secret Value

123456

Figure 7-6 Environment variable in remote login



```
user@s[redacted]machine:~$ kubectl -n [redacted]-env20231229 exec -it demo-frontend[redacted]6956968d8-mx9wh -c demo-frontend-1704597891 -- /bin/sh
sh-5.0# env | grep test1
test1=123456
sh-5.0#
```

----End

7.3 Disabling the Sub-account Permission to Read Keys for Key Protection

When you use many accounts to manage the CAE environment, you can create sub-accounts under your cloud service account for employees in different departments based on your organization structure and set different access permissions for these sub-accounts to isolate user permissions.

For example, you can create a development account and a test account for development and test personnel respectively. Because test personnel do not need to be aware of sensitive information, you can disable the permission of the test account to obtain sensitive information.

This practice describes how to disable all DEW permissions and CAE remote login permission of a sub-account so that the sub-user cannot read keys.

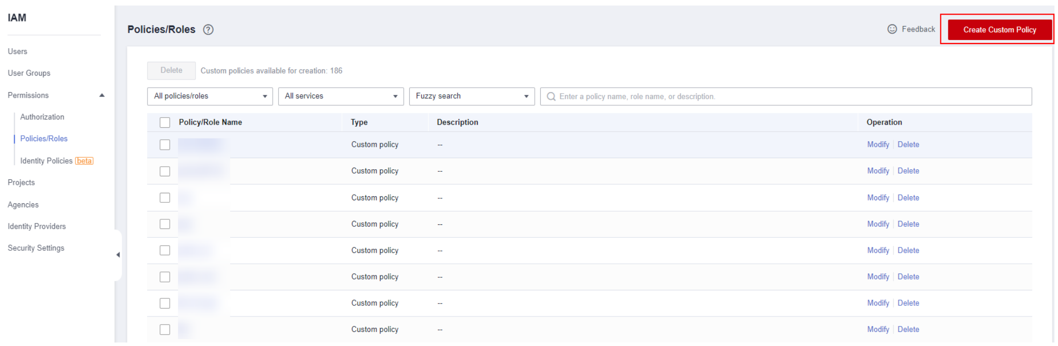
Creating a Custom Policy

Step 1 Log in to IAM.

Step 2 Choose **Permissions** > **Policies/Roles**.

Step 3 Click **Create Custom Policy** in the upper right corner.

Figure 7-7 IAM console



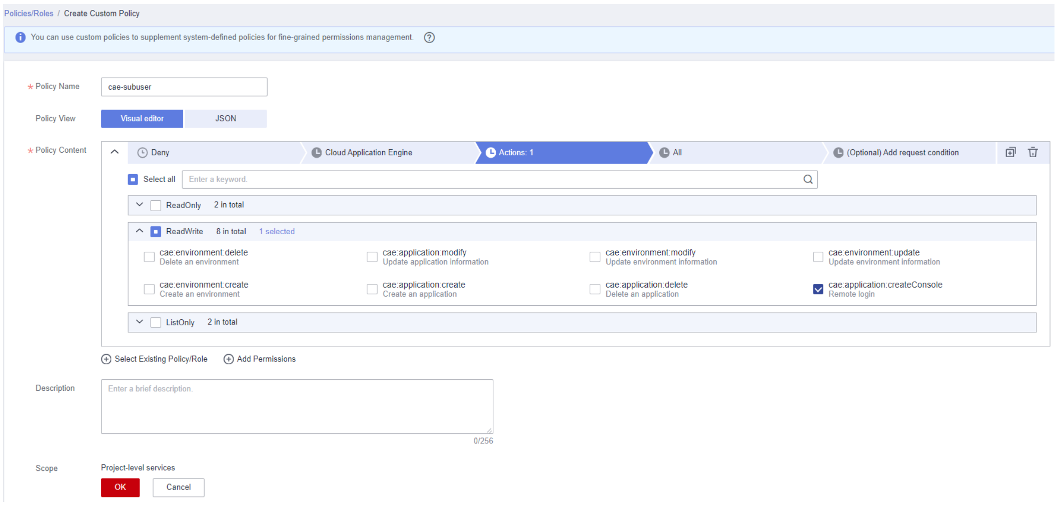
Custom policies available for creation: 186

Policy/Role Name	Type	Description	Operation
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete
[redacted]	Custom policy	--	Modify | Delete

Step 4 Configure a custom policy.

- **Policy Name**: Enter **cae-subuser**.
- **Policy Content**: Select **Deny**.
- **Select service**: Select **Cloud Application Engine**.
- **Select Action**: Select **cae:application:createConsole**.

Figure 7-8 Creating a custom policy



Policies/Roles / Create Custom Policy

You can use custom policies to supplement system-defined policies for fine-grained permissions management.

* Policy Name: cae-subuser
Policy View: Visual editor | JSON
* Policy Content: Deny | Cloud Application Engine | Actions: 1 | All | (Optional) Add request condition

Select all — Enter a keyword.

ReadOnly 2 in total

ReadWrite 8 in total 1 selected

☐ cae:environment:delete Delete an environment	☐ cae:application:modify Update application information	☐ cae:environment:modify Update environment information	☐ cae:environment:update Update environment information
☐ cae:environment:create Create an environment	☐ cae:application:create Create an application	☐ cae:application:delete Delete an application	☑ cae:application:createConsole Remote login

ListOnly 2 in total

Select Existing Policy/Role | Add Permissions

Description: Enter a brief description. 0/256

Scope: Project-level services

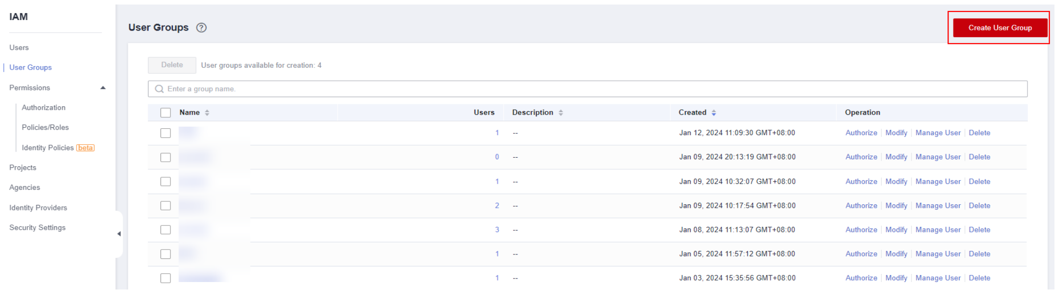
Step 5 Click **OK**.

----End

Creating a User Group and Assigning Permissions

Step 1 Choose **User Groups** and click **Create User Group**.

Figure 7-9 Creating a user group



User Groups

User groups available for creation: 4

Enter a group name.

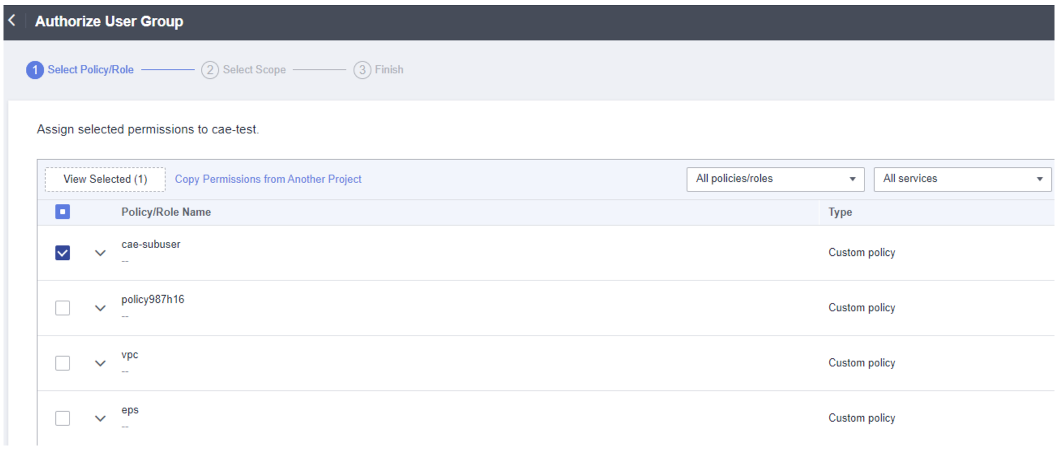
Name	Users	Description	Created	Operation
[redacted]	1	--	Jan 12, 2024 11:09:30 GMT+08:00	Authorize | Modify | Manage User | Delete
[redacted]	0	--	Jan 09, 2024 20:13:19 GMT+08:00	Authorize | Modify | Manage User | Delete
[redacted]	1	--	Jan 09, 2024 10:32:07 GMT+08:00	Authorize | Modify | Manage User | Delete
[redacted]	2	--	Jan 09, 2024 10:17:54 GMT+08:00	Authorize | Modify | Manage User | Delete
[redacted]	3	--	Jan 08, 2024 11:13:07 GMT+08:00	Authorize | Modify | Manage User | Delete
[redacted]	1	--	Jan 05, 2024 11:57:12 GMT+08:00	Authorize | Modify | Manage User | Delete
[redacted]	1	--	Jan 03, 2024 15:35:56 GMT+08:00	Authorize | Modify | Manage User | Delete

Step 2 Enter a user group name, for example, **cae-test**, and click **OK**.

Step 3 In the user group list, click **cae-test**.

Step 4 On the **Permissions** tab, click **Authorization** and select the custom policy created in **Creating a Custom Policy**.

Figure 7-10 Authorizing a custom policy



Authorize User Group

1 Select Policy/Role — 2 Select Scope — 3 Finish

Assign selected permissions to cae-test.

View Selected (1) | Copy Permissions from Another Project | All policies/roles | All services

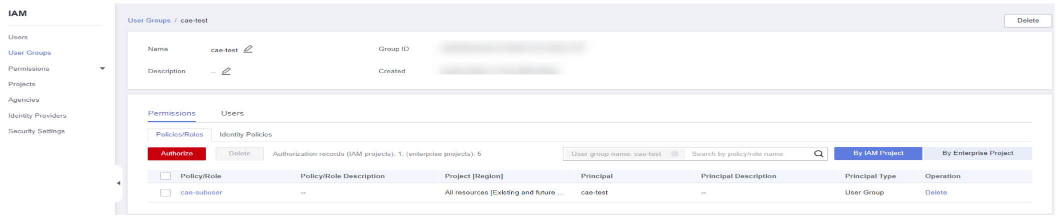
	Policy/Role Name	Type
✓	cae-subuser --	Custom policy
	policy987h16 --	Custom policy
	vpc --	Custom policy
	eps --	Custom policy

Step 5 Click **Next** and set **Scope** to **All resources**.

Step 6 Click **OK**.

Click **Finish** to go back to the **User Groups** page.

Figure 7-11 User group authorized



Name: cae-test
Description: --
Group ID: [redacted]
Created: [redacted]

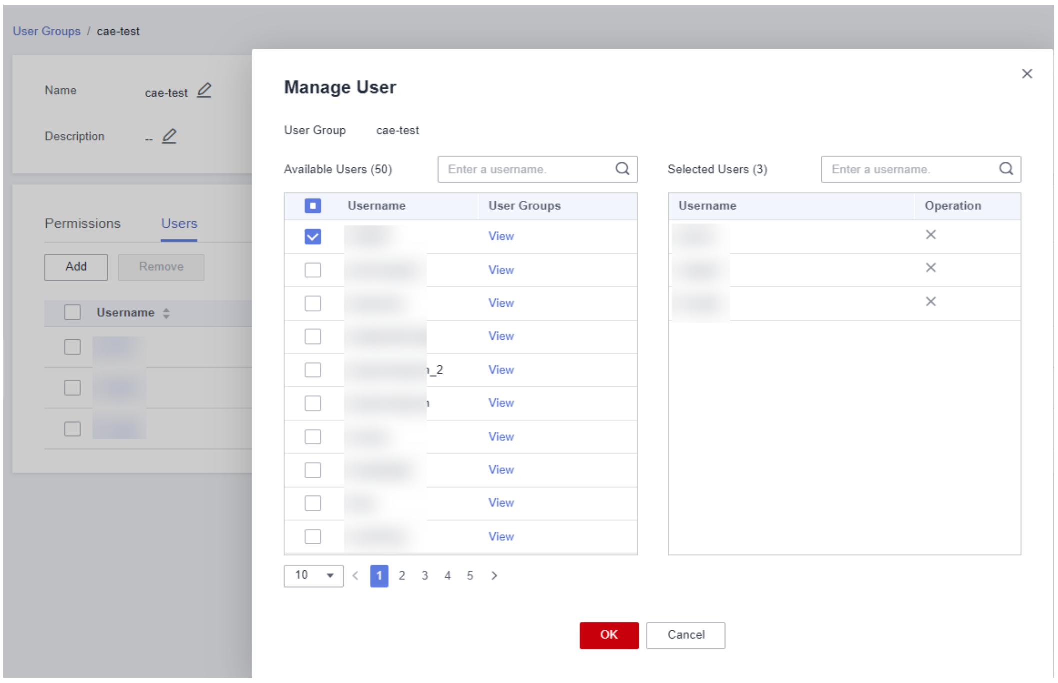
Authorization records (IAM projects): 1; (enterprise projects): 5

Policy/Role	Policy/Role Description	Project [Region]	Principal	Principal Description	Principal Type	Operation
cae-subuser	--	All resources [Existing and future ...	cae-test	--	User Group	Delete

Step 7 Click the **Users** tab.

Step 8 In the user list, select the sub-user whose permissions need to be configured and click **OK**.

Figure 7-12 Configuring user management



User Groups / cae-test

Name cae-test
Description --
Permissions Users
Add Remove
Username

Manage User

User Group cae-test

Available Users (50) — Enter a username.

☐	Username	User Groups
✓		View
☐		View
☐		View
☐		View
☐	_2	View
☐		View
☐		View
☐		View
☐		View
☐		View

10 < 1 2 3 4 5 >

Selected Users (3) — Enter a username.

Username	Operation
	×
	×
	×

OK Cancel

----End

Verifying Sub-user Permissions

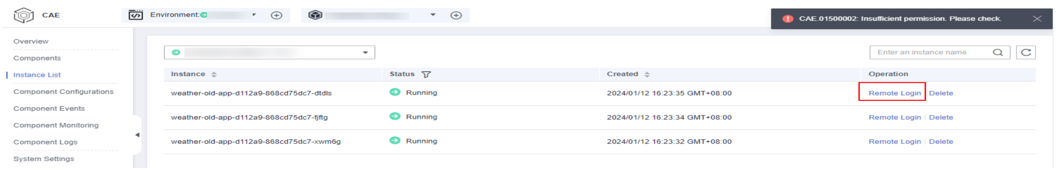
Step 1 Log in to CAE as the sub-user in **Step 8**.

Step 2 Perform operations such as **Adding a Secret** and **Configuring an Environment Variable**. The operations are normal.

Step 3 Choose **Instance List**.

Step 4 Select the target instance and click **Remote Login** in the **Operation** column. The secret details cannot be viewed.

Figure 7-13 No remote login permission



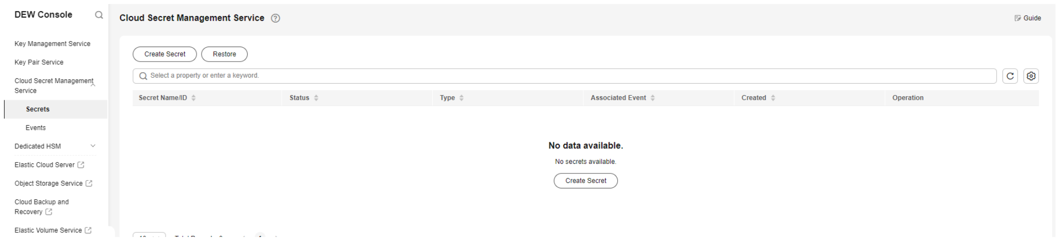
CAE.01500002: Insufficient permission. Please check.

Instance	Status	Created	Operation
weather-old-app-d112a9-868cd75dc7-dtdls	Running	2024/01/12 16:23:35 GMT+08:00	Remote Login | Delete
weather-old-app-d112a9-868cd75dc7-fjftg	Running	2024/01/12 16:23:34 GMT+08:00	Remote Login | Delete
weather-old-app-d112a9-868cd75dc7-xwm6g	Running	2024/01/12 16:23:32 GMT+08:00	Remote Login | Delete

Step 5 Click ☰ in the upper left corner. In the service list, choose **Data Encryption Workshop** to go to the DEW console.

Step 6 Choose **Cloud Secret Management Service** > **Secrets**. The secret details cannot be viewed.

Figure 7-14 No DEW agency permission



No data available.

No secrets available.

----End