

Cloud Application Engine

Getting Started

Issue 01
Date 2024-07-01



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI 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

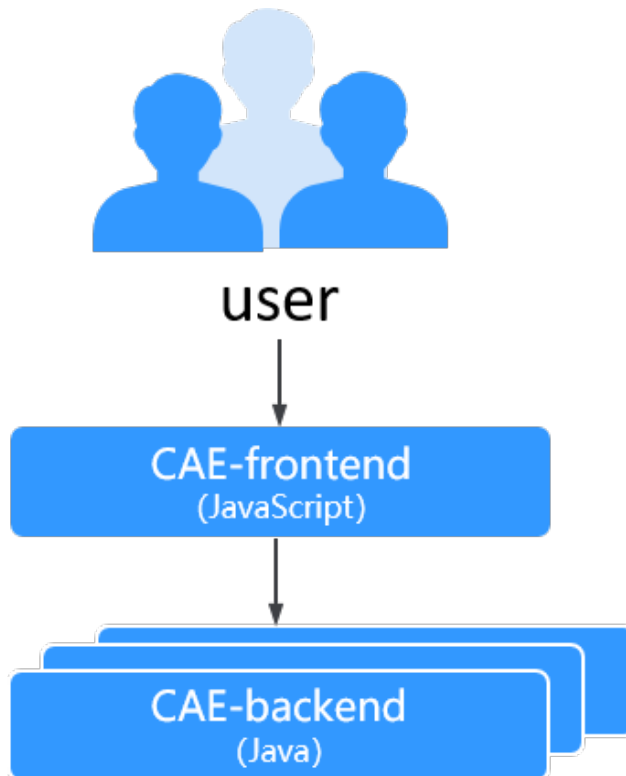
1 Quick Experience..... 1

1 Quick Experience

Cloud Application Engine (CAE) simplifies hosting for serverless applications. With CAE, applications can be deployed within minutes using source code, software packages, or container images. CAE supports mainstream languages such as Java, Go, and Tomcat and multiple runtime systems. It provides seamless hosting of multiple types of applications, including web applications, microservices, and APIs. Also, it supports auto scaling based on resources or custom service metrics to cope with unexpected user access traffic. Related resources are billed on the pay-per-use basis. In addition, CAE enables zero infrastructure O&M, so that users can focus more on application service development.

In this example, CAE is bound to the GitHub source code repository, and a frontend and a backend component are deployed to build and archive source code and create applications.

The following figure shows the demo logical networking and calling relationship.



- Component CAE-frontend: Frontend GUI developed using the Vue framework. It functions as the application entry and can send requests to the backend.
- Component CAE-backend: Backend service developed using Spring Boot. It processes requests sent by the demo-frontend component.

Prerequisites

- You have [signed up for a HUAWEI ID and enabled Huawei Cloud services](#) and logged in.
- You have registered a [GitHub](#) account and created a private token for subsequent authorization.

GitHub addresses:

- demo-frontend: <https://github.com/servicestage-demo/cae-frontend>
- demo-backend: <https://github.com/servicestage-demo/cae-backend>

Creating an Environment

Step 1 Log in to the CAE console. If you use CAE for the first time, a message is displayed indicating that no environment has been created.

Step 2 Click **Create Now**. On the displayed page, set parameters by referring to [Table 1-1](#).

Table 1-1 Creating an environment

Parameter	Description
Environment	Enter an environment name.

Parameter	Description
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.
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.

Step 3 Click **OK**. The creation progress is displayed in the **Creating environment** window. Then the environment is successfully created.

----End

Creating an Application

Step 1 Choose **Components**.

Step 2 Click  on the right of **Application** on the top to create an application.

Step 3 Enter an application name, for example, **cae_test**.

Step 4 Click **OK**.

----End

Creating a Component


Create a frontend component.

Step 1 In the created application, click **Create Component**.

 **NOTE**

You can also click **Try Now** under the **Getting Started** template on the overview page. Select the corresponding environment and application, and click **Deploy** to create components demo-backend and demo-frontend.

Step 2 Configure the component.

- **Component:** Enter **cae-frontend**.
- **Version:** The default value is **1.0.0**.
- **Specifications:** Retain the default value. Change it if necessary.
- **Instances:** Retain the default value. Change it if necessary.
- **Code Source:** Select **GitHub**, and set **Authorization, Username/ Organization, Repository, and Branch**.
 - a. If you use CAE for the first time, create an authorization first. Click **Create Authorization** and set **Authorization** and **Mode**.
 - **Authorization:** Retain the default value. Change it if necessary.
 - **Mode:** Select **Token**.
 - **Token:** Obtain a token from the GitHub website. You can click  to view details.
 - b. Click **OK**. After the authorization is created, enter **Username/ Organization, Repository, and Branch**.
 - **Username/Organization:** Select the username/organization created for GitHub.
 - **Repository:** Select **cae-frontend**.
 - **Branch:** Select **master**.
- **Language/Runtime System:** Select **Nodejs8**.
- **Custom Build:** Select **Default**.

Step 3 Click **Create and Deploy Component**. In the displayed dialog box, click **Deploy Now**. Wait until the component is deployed.

 **NOTE**


- **Configure Component:** Click **Configure Component** to go to the component configuration page. After configuring the component by referring to [Configuring a Component](#), click **Set and Deploy Component**.
- **Create and Deploy Component:** Directly create and deploy a component. Select this mode if components do not need to be configured individually.
- **Cancel:** Cancel the creation. The component information will not be saved.

----End

Create a backend component.

Step 1 In the created application, click **Create Component**.

Step 2 Configure the component.

- **Component:** Enter **cae-backend**.
- **Version:** The default value is **1.0.0**.
- **Specifications:** Retain the default value. Change it if necessary.
- **Instances:** Retain the default value. Change it if necessary.
- **Code Source:** Select **GitHub**, and set **Authorization, Username/Organization, Repository,** and **Branch**.
 - a. If you use CAE for the first time, create an authorization first. Click **Create Authorization** and set **Authorization** and **Mode**.
 - **Authorization:** Retain the default value. Change it if necessary.
 - **Mode:** Select **Token**.
 - **Token:** Obtain a token from the GitHub website. You can click  to view details.
 - b. Click **OK**. After the authorization is created, enter **Username/Organization, Repository,** and **Branch**.
 - **Username/Organization:** Select the username/organization created for GitHub.
 - **Repository:** Select **cae-backend**.
 - **Branch:** Select **master**.
- **Language/Runtime System:** Select **java8**.
- **Custom Build:** Select **Default**.

Step 3 Click **Create and Deploy Component**. In the displayed dialog box, click **Deploy Now**. Wait until the component is deployed.

----End

Configuring a Component

Configure a frontend component.

Step 1 Choose **Component Configurations**.

Step 2 Select **cae-frontend** from the drop-down list in the upper part of the page.

Step 3 Click **Edit** in the **Access Mode** module.

Step 4 In the **Access Component from Another Environment** area, click **Load Balancing > Add Load Balancer**.

Step 5 Configure the parameters based on [Table 1-2](#).

Table 1-2 Configuring external network access for load balancing

Parameter	Description
Load Balancer	Select Built-in load balancer .
Access Control	Select Allow all IP addresses . If you have configured an access control whitelist or blocklist, only IP addresses in the whitelist or not in the blocklist can access the component.
Health Check	Use the default value Start .
Port Settings	<ul style="list-style-type: none"> ● Protocol: Select TCP. ● Listening Port: Enter 8080. ● Access Port: Select a valid port ranging from 0 to 65,535, for example, 20,004. The port number must be unique.

Step 6 Click **OK**.

Step 7 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

Configure a backend component.

Step 1 Select **cae-backend** from the drop-down list in the upper part of the page.

Step 2 Click **Edit** in the **Access Mode** module.

Step 3 Select **Load Balancing**, click **Add Load Balancer**, and set parameters by referring to [Table 1-3](#).

Table 1-3 Configuring external network access for load balancing

Parameter	Description
Load Balancer	Select Built-in load balancer .
Access Control	Select Allow all IP addresses . If you have configured an access control whitelist or blocklist, only IP addresses in the whitelist or not in the blocklist can access the component.
Health Check	Use the default value Start .

Parameter	Description
Port Settings	<ul style="list-style-type: none"> • Protocol: Select TCP. • Listening Port: Enter 9090. • Access Port: Select a valid port ranging from 0 to 65,535, for example, 20,001. The port number must be unique.

Step 4 Click **OK**.

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

Configure environment variables for a frontend component.

Step 1 Select **cae-frontend** from the drop-down list in the upper part of the page.

Step 2 Click **Edit** in the **Environment Variables** module.

Step 3 Click **Add Environment Variable** and configure the environment variable by referring to [Table 1-4](#).

Table 1-4 Environment variable parameters

Parameter	Description
Name	Enter VUE_APP_BASE_URL .
Variable/ Variable Reference	Enter the backend access address, which can be obtained from Components , as shown in Figure 1-1 .

Figure 1-1 Obtaining the backend access address

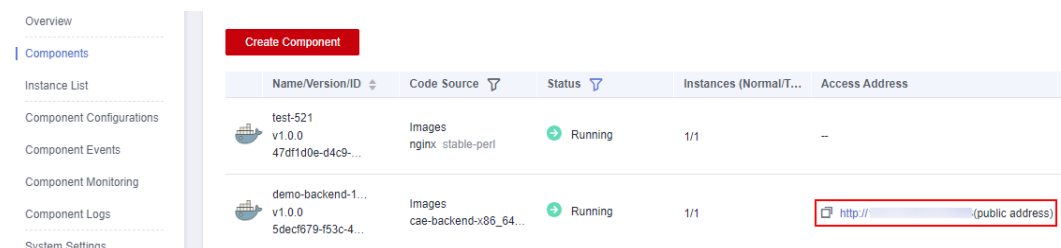
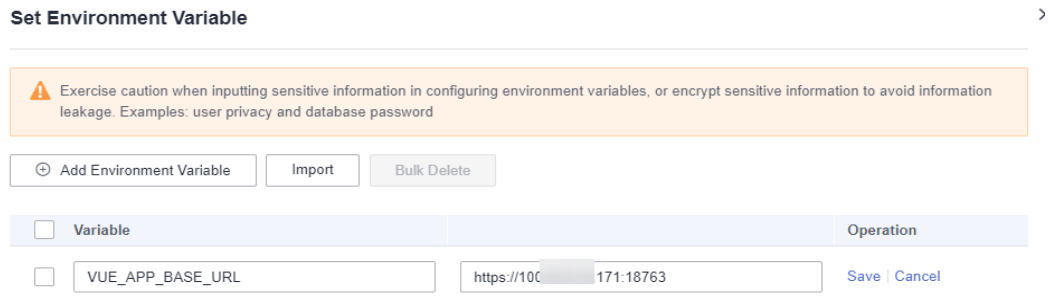


Figure 1-2 Configuring frontend environment variables



Step 4 Click **Save** and then click **OK** to complete the configuration.

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.

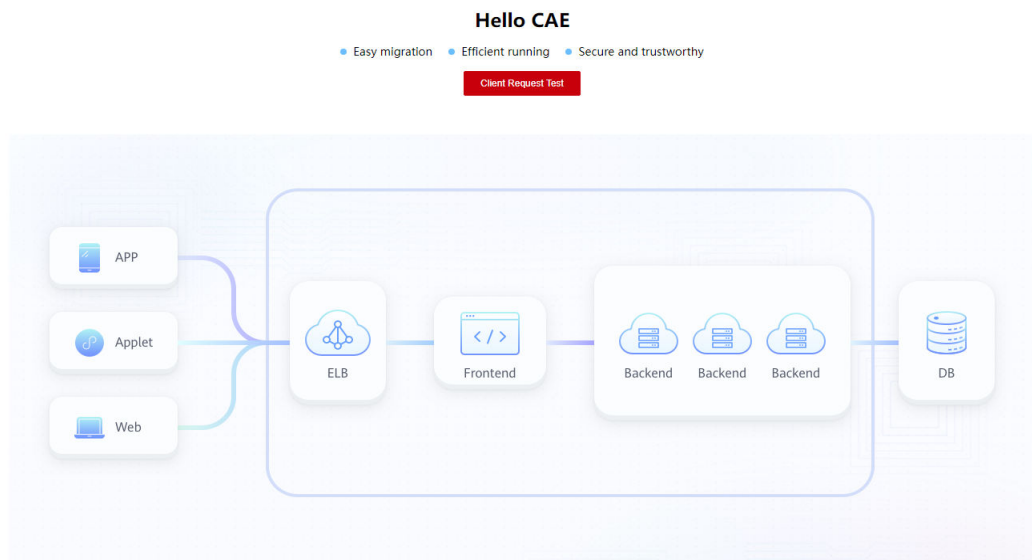
----End

Accessing an Application

Step 1 Choose **Components**.

Step 2 Click the public network address in the **Access Address** column of the **cae-frontend** component to view the application page.

Figure 1-3 Application page



NOTE

If you select the private network access mode, log in to the cluster node and run the **curl** command. For details, see [Private Network Access](#).

----End

Application O&M

Step 1 On the **Overview** page, you can view the component health status and resource usage.

Step 2 Select the environment, application, and component to be viewed.

- Click **Component Events** to display the entire running process of the component.
- Click **Component Monitoring**. The number of running instances of the current component and the CPU and memory usage of each instance are displayed.
- Click **Component Logs**. The logs of each instance are displayed.

----End