

ServiceStage

FAQs

Issue 01
Date 2023-12-05



Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2023. 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 Application Development FAQs.....	1
2 Environment Management.....	3
2.1 What Are the Differences Between the Microservice and Platform Service?.....	3
3 Application Management.....	4
3.1 How Do I View the Causes of Application Component Deployment Failures?.....	4
3.2 What Should I Do If I Don't See the VM Agent After Installing It?.....	4
3.3 What Should I Do If the VM Agent Is Offline?.....	6
3.4 What If an Instance Is Being Created for a Long Time?.....	7
3.5 How Do I Solve the Dependency Problem When a Node Program Runs in Docker?.....	8
3.6 How Do I Customize a Tomcat Context Path?.....	8
3.7 How Do I Use a Fixed Application Component IP?.....	8
3.8 What Should I Do If an ECS Error Occurs When I Create and Deploy a Component on a VM?.....	9
3.9 What Should I Do If I Cannot Access the Port When I Create and Deploy a Component on a VM?.....	10
3.10 Which Directories Do I Use to Write Files for VM-Deployed Application Components?.....	11
3.11 What Should I Do If "host status is not active" Is Reported When a VM-Deployed Component Fails to Be Deleted?.....	11
3.12 How Do I Use the ServiceStage Source Code Deployment Function?.....	12
3.13 What Should I Do If Components Fail to Be Upgraded in ServiceStage Dark Launch?.....	13
3.14 How Do I Mount Items to Modify the Configuration File of a Container-based Component?.....	13
3.15 What Should I Do If the Application Name Displayed on Microservice Governance Is Different from That Displayed on ServiceStage Application Management After an Application Component Is Connected to a Microservice Engine?.....	14
4 Continuous Delivery.....	16
4.1 How Does ServiceStage Manage Code on IDEA?.....	16
4.2 How Do I Add the Build Server Address to the GitLab Server Security Group?.....	16
4.3 How Do I Add the Build Server Address to the Maven Server Security Group?.....	17
4.4 What Should I Do If I Failed to Use ServiceStage to Build a Job?.....	18
5 Software Center.....	22
5.1 What If a Software Package Fails to Be Uploaded?.....	22
5.2 What If the Docker Client Fails to Push Images?.....	23
5.3 What Should I Do If I Cannot Download an SWR Software Package?.....	24
6 Infrastructure.....	25

6.1 Are Existing Programs Affected If I Unsubscribe Servers?.....	25
6.2 What Should I Do If the Service Registration Fails After IPv6 Is Enabled for the Exclusive Microservice Engine with Security Authentication Enabled?.....	25
6.3 What Should I Do If a Non-Microservice Engine Error Occurs When I Operate an Exclusive Microservice Engine?.....	26
6.4 How Do I Switch a Cluster Used for Component Building from a Shared Cluster to a Private One?.....	26
6.5 What Should I Do If the Access Address Fails to Be Processed During CSE Creation?.....	27
6.6 What Do I Need to Know Before Upgrading an Exclusive Microservice Engine?.....	28
7 Application O&M.....	31
7.1 Why Can't I View ServiceStage Logs?.....	31
7.2 What Should I Do If Application Access Mode Becomes Invalid When EIP Is Replaced?.....	32
7.3 What Should I Do If the Memory Usage of a Node Becomes Too High After a New Service Is Started?.....	32
7.4 How Do I Uninstall a Service?.....	32
8 Application Development.....	34
8.1 What Are the Differences Between the Microservice and Common Application?.....	34
8.2 How Do I Handle a Microservice Registry Failure (Java Chassis)?.....	35
8.3 How Do I Troubleshoot Microservices Deployed on the Cloud?.....	37
8.4 Should I Use the SDK or ServiceMesh to Build a Microservice?.....	37
8.5 What If I Fail to Obtain a Dependency?.....	38
8.6 What Is Service Name Duplication Check?.....	39
8.7 Why Do I Have to Define Service Contracts?.....	39
8.8 Why Are Microservice Development Framework and Netty Versions Unmatched?.....	40

1 Application Development FAQs

Key Information

To facilitate quick fault locating, provide detailed key information when posting an issue in the community. You are advised to provide a demo that can reproduce the fault.

The following uses ServiceComb Java Chassis as an example:

1. Framework logs: By default, framework logs are printed with service logs, and the **cse.log** file is generated in the root directory. If the log framework such as Log4j2 or Logback is used on the service side, search for the key information based on the customized log policy.
 - a. Key information about service startup:

Table 1-1 Key information about service startup

Keyword	Description
choose org.apache.servicecomb	ServiceComb Java Chassis supports two types of REST communication channels. You need to determine the communication channel to be used based on logs. The choose org.apache.servicecomb.transport.rest.vertx.VertxRestTransport framework uses the REST over Vertx communication channel by default. That is, Vertx is used as the HTTP server. The choose org.apache.servicecomb.transport.rest.servlet.ServletRestTransport framework also supports the REST over Servlet communication channel. That is, other HTTP servers, such as Tomcat, are used.
endpoint to publish	Microservice release address.

Keyword	Description
Register microservice instance success	Flag indicating a successful service instance registration.

- b. Key information about service calling:

Table 1-2 Key information about service calling

Keyword	Description
find instances	Before calling the server (called service), the consumer (calling service) queries the server instance from the service center of the microservice engine.
accesslog	The access.log file records the request sources, such as APIs and status codes for calling the service. By default, this function is disabled.

The **access.log** file printing is affected by the communication channel and log framework. If the REST over Vertx communication channel is used, the **access.log** file is recorded by Vertx. For details, see https://servicecomb.apache.org/references/java-chassis/en_US/build-provider/access-log-configuration/.

The recommended format of the **access.log** file is as follows:

```
servicecomb.accesslog.pattern: "%h - - %t cs-uri %s %B %D %H %SCB-traceId"
```

By default, the **access.log** file is generated in the root directory. If the log framework such as Log4j2 or Logback is used on the service side, you can switch the log framework by referring to https://servicecomb.apache.org/references/java-chassis/en_US/build-provider/access-log-configuration/.

If the REST over Servlet communication channel is used, the **access.log** file is recorded by the HTTP server. To use the **access.log** file, find the related reference.

For example, enable the built-in Tomcat of Spring Boot as follows:

```
server:
  tomcat:
    accesslog:
      enabled: true
      pattern: '%h %l %u %t "%r" %s %b %D'
      directory: accesslogs
      buffered: false
      basedir: ./logs
```

2. Versions of the microservice engine and SDK. You can click the engine name to view the microservice engine version. For the SDK version, search for the dependency whose **groupId** is **org.apache.serviccomb**.

2 Environment Management

2.1 What Are the Differences Between the Microservice and Platform Service?

The microservice is an architecture model used to build an application system. The platform service is the middleware service provided by the cloud.

You must purchase a platform service to use it. To use a microservice, first develop it and release it on the cloud through the service discovery capability provided by the cloud.

3 Application Management

3.1 How Do I View the Causes of Application Component Deployment Failures?

Symptom

After the application component is deployed, the status is displayed as **Not Ready**, indicating that the application component fails to be deployed.

Solution

Step 1 Log in to ServiceStage.

Step 2 Use either of the following methods to go to the **Instance List** page.

- On the **Application Management** page, click the application to which the component belongs, and click the target component in **Component List**. In the left navigation pane, choose **Instance List**.
- On the **Component Management** page, click the target component. In the left navigation pane, choose **Instance List**.

Step 3 In the instance list, click next to the target instance.

Step 4 On the **Events** tab page, view a failure event and determine its cause.

----End

3.2 What Should I Do If I Don't See the VM Agent After Installing It?

Symptom

After the agent is installed on the VM, the message "Install agent success!" is displayed.

However, **Agent Status** indicates that the agent is still missing and to install it first.

Solution

Step 1 Log in to the ECS where the VM agent is offline. For details, see [Logging In to an ECS](#).

Step 2 Run the following commands to check **Authorization Model** selected during the VM agent installation.

```
cd /opt/servicestage-agent
```

```
cat servicestage-agent.conf
```

- If the returned AK and SK values are empty, **Authorization Model** is **Agency**. Then, go to [Step 3](#).
- If the returned AK and SK values are not empty, **Authorization Model** is **AKSK**. Go to [Step 4](#).

Step 3 If **Authorization Model** is **Agency**:

1. Log in to the cloud server console.
2. Choose **Elastic Cloud Server** and click the ECS whose VM agent is offline.
3. In the **Management Information** area of the **Basic Information** tab page, view the IAM agency bound to the ECS.
4. Log in to the Identity and Access Management (IAM) console.
5. Choose **Agencies** and click the agency obtained in [Step 3.3](#).
 - a. On the **Basic Information** tab page, check whether **Cloud Service** is ECS.
 - b. On the **Permissions** tab page, check whether the permission is **Tenant Administrator**.

If both are yes, go to [Step 5](#).

If either is no, [modify an agency](#) and go to [Step 3.6](#).

6. Log in to the ECS where the VM agent is offline. For details, see [Logging In to an ECS](#).

7. Run the following commands to restart the agent. Replace *x.x.x* with the version of **servicestage-agent** in the actual environment.

```
cd /opt/servicestage-agent/servicestage-agent-x.x.x
```

```
su agent ./servicestage-agent.sh start
```

8. Check whether the agent is online.
 - If yes, no further operation is required.
 - If no, go to [Step 5](#).

Step 4 If **Authorization Model** is **AKSK**, perform the following operations:

1. Obtain the AK or SK with the correct permissions or create new AK or SK. For details, see [Access Keys](#).
2. Log in to the ECS where the VM agent is offline. For details, see [Logging In to an ECS](#).
3. Run the following commands to change the AK and SK values in the configuration file, and save and exit the file.

```
cd /opt/servicestage-agent
vi servicestage-agent.conf
```

4. Run the following commands to restart the agent. Replace *x.x.x* with the version of **servicestage-agent** in the actual environment.

```
cd /opt/servicestage-agent/servicestage-agent-x.x.x
su agent ./servicestage-agent.sh start
```

5. Check whether the agent is online.
 - If yes, no further operation is required.
 - If no, go to [Step 5](#).

Step 5 If the fault persists, contact customer service.

----End

3.3 What Should I Do If the VM Agent Is Offline?

Symptom

The agent has been installed but is offline and cannot work properly.

Solution

Step 1 Log in to the ECS where the VM agent is offline. For details, see [Logging In to an ECS](#).

Step 2 Run the following commands to check **Authorization Model** selected during the VM agent installation.

```
cd /opt/servicestage-agent
cat servicestage-agent.conf
```

- If the returned AK and SK values are empty, **Authorization Model** is **Agency**. Then, go to [Step 3](#).
- If the returned AK and SK values are not empty, **Authorization Model** is **AKSK**. Go to [Step 4](#).

Step 3 If **Authorization Model** is **Agency**:

1. Log in to the cloud server console.
2. Choose **Elastic Cloud Server** and click the ECS whose VM agent is offline.
3. In the **Management Information** area of the **Basic Information** tab page, view the IAM agency bound to the ECS.
4. Log in to the Identity and Access Management (IAM) console.
5. Choose **Agencies** and click the agency obtained in [Step 3.3](#).
 - a. On the **Basic Information** tab page, check whether **Cloud Service** is ECS.
 - b. On the **Permissions** tab page, check whether the permission is **Tenant Administrator**.

If both are yes, go to [Step 5](#).

If either is no, [modify an agency](#) and go to [Step 3.6](#).

6. Log in to the ECS where the VM agent is offline. For details, see [Logging In to an ECS](#).
7. Run the following commands to restart the agent. Replace *x.x.x* with the version of **servicestage-agent** in the actual environment.

```
cd /opt/servicestage-agent/servicestage-agent-x.x.x
su agent ./servicestage-agent.sh restart
```

Step 4 If **Authorization Model** is **AKSK**, perform the following operations:

1. Obtain the AK or SK with the correct permissions or create new AK or SK. For details, see [Access Keys](#).
2. Log in to the ECS where the VM agent is offline. For details, see [Logging In to an ECS](#).
3. Run the following commands to change the AK and SK values in the configuration file, and save and exit the file.

```
cd /opt/servicestage-agent
vi servicestage-agent.conf
```

4. Run the following commands to restart the agent. Replace *x.x.x* with the version of **servicestage-agent** in the actual environment.

```
cd /opt/servicestage-agent/servicestage-agent-x.x.x
su agent ./servicestage-agent.sh restart
```

Step 5 If the fault persists, contact customer service.

----End

3.4 What If an Instance Is Being Created for a Long Time?

After an application component is created, if the service instance is in the **Not ready** state for a long time, go to the service instance list and check the instance details. On the **Event** tab page, you can see that the memory is insufficient.

Instance Name	Status	CPU (Request/Limit)	Memory (Request/Limit)	Node	Running Time	Pod IP	Created	Operation
fusionweahermap-test-env-v1e3sr-9bdc...	Creating	0.25Core / 0.25Core	0.5GIB / 0.5GIB		59 seconds		May 15, 2020 16:05:39 GMT+08:00	Delete
fusionweahermap-test-env-v1e3sr-9bdc...	Creating	0.25Core / 0.25Core	0.5GIB / 0.5GIB		59 seconds		May 15, 2020 16:05:39 GMT+08:00	Delete

Name	Count	Type	Created	Description
Failed to schedule instance	2	Warning	May 15, 2020 16:05:39 GMT+08:00	0/1 nodes are available: 1 insufficient memory

To solve this problem, add a node. For details, see .

3.5 How Do I Solve the Dependency Problem When a Node Program Runs in Docker?

Symptom

A node program depends on node-gyp when running in the microservice docker. How can I install the dependency before the program runs?

Solution

Customize a Dockerfile and add the node-gyp dependency to the Dockerfile.

3.6 How Do I Customize a Tomcat Context Path?

When creating and deploying a Tomcat application, Tomcat configurations are required. Specifically, the default **server.xml** configuration is used, the context path is `/`, and no application path is specified.

- If **Public Network Access** is enabled, the application access address is **http://*Public domain name of the application*:*Application access port***, for example, **http://example_domain.com:30317**.
- If **Public Network Access** is not enabled, the application access address is **http://*Intranet access address of the VPC*:*Application access port***, for example, **http://192.168.0.168:30317**.

During the component configuration for component deployment, you can customize the application path based on the actual service when configuring Tomcat parameters.

1. Select **Parameter settings**.
2. Click **Use Sample Code** and edit the template file based on service requirements.
3. Modify the value of **Context path** by referring to the following example. For example, after you change the value to **app-path**, the application access address is changed to **http://example_domain.com:30317/app-path** or **http://192.168.0.168:30317/app-path**.

```
<Host name="localhost" appBase="webapps"
  unpackWARs="true" autoDeploy="true" >
  <Context path="app-path" docBase="ROOT.war"/>
```

3.7 How Do I Use a Fixed Application Component IP?

Symptom

If **TCP/UDP Route Configuration** is not set during application component deployment, the access IP address of the application changes when the container restarts. This may create difficulties in your configuration.

Solution

Set **TCP/UDP Route Configuration** when creating or deploying an application component. You can solve the problem using any of the following methods:

- Intra-cluster access: An application can be accessed by other applications in the same cluster using an internal domain name.
- Intra-VPC access: An application can be accessed by other applications in the same VPC using the IP address of a cluster node or the IP address of an ELB service in a private network.
- External access: An EIP is used to access applications from a public network. This access mode is applicable to services that need to be exposed to a public network in the system. In this access mode, EIP must be bound to any node in the cluster and a port mapped to the node must be configured.

Add Service ×

* Service Name

Access Mode Intra-cluster access Intra-VPC access Public network access

Allows access from the Internet over TCP/UDP, including EIP.

* Access Type

Service Affinity Cluster level Node level

1. All nodes in the cluster can use their IP addresses+port numbers to access the workload targeted by the service.
2. Routing hops will be used. As a result, routing performance will be compromised and clients' source IP addresses will be masked.

* Port Mapping

Protocol	Container Port	Access Port
<input type="text" value="TCP"/>	<input type="text" value="Range: 1-65535"/>	<input type="text" value="Automatically g..."/>

3.8 What Should I Do If an ECS Error Occurs When I Create and Deploy a Component on a VM?

Symptom

The ECS service may be unavailable when you create and deploy a component on a ServiceStage VM.

For example, calling the ECS interface times out during component deployment, and the following error information is displayed in log details:

```
{
  "statusCode": 500,
  "jsonBody": {
    "error_code": "SVCSTG.VMAPP.5001002",
    "error_msg": "read ECS host 471ff77a-c827-41d5-941d-4fea8aaa56ef fail TIMEOUT."
  }
}
```

Solution

Step 1 Redeploy the component and check whether the deployment is successful.

- If yes, no further action is required.
- If no, go to [Step 2](#).

Step 2 Contact technical support.

----End

3.9 What Should I Do If I Cannot Access the Port When I Create and Deploy a Component on a VM?

Symptom

The container port may fail to be accessed when you create and deploy a component on a ServiceStage VM. When the `curl -kv http://{IP address of the ECS node where the application component is deployed}:{Container port}` command is run to access the container port, the system displays a message indicating that the access timed out.

```
C:\Users\>curl -kv http://:8080
* Rebuilt URL to: http://:8080/
* Trying ...
* TCP_NODELAY set
* connect to port 8080 failed: Timed out
* Failed to connect to port 8080: Timed out
* Closing connection 0
curl: (7) Failed to connect to port 8080: Timed out
```

Solution

Step 1 Log in to the ECS console and click **Elastic Cloud Server**.

Step 2 In the ECS list, find the target ECS for deploying the component and click it to open its details page.

Step 3 On the **Security Groups** tab, click **Change Security Group** and check whether the port rule exists in an existing group.

- If yes, select the security group.
- If no, click **Create Security Group** to create one and configure its rules. Next, select the created group.

Step 4 Run the `curl -kv http://{IP address of the ECS node where the application component is deployed}:{Container port}` command again to access the container port and check whether the fault is rectified.

----End

3.10 Which Directories Do I Use to Write Files for VM-Deployed Application Components?

For such components, only their running directory is available for writing files. Such files include log files or zip packages.

This directory is the `/opt/application/${appName}/${appVersion}/${instanceId}` directory on the application ECS, where,

- `${appName}` indicates the component instance name.
- `${appVersion}` indicates the version number of the component instance.
- `${instanceId}` indicates the instance ID.

This rule takes effect only for new and upgraded component instances. Directories of component instances that have been deployed retain the original permissions.

3.11 What Should I Do If "host status is not active" Is Reported When a VM-Deployed Component Fails to Be Deleted?

Symptom

The component deployed on a VM fails to be deleted. Task details show the error information:

```
{
  "statusCode": 400,
  "jsonBody": {
    "error_code": "SVCSTG.VMAPP.4001020",
    "error_msg": "4001020",
    "error_detail": "host status is not active: abb3d0a4-f715-4932-b7ec-6dd917f65778,4f68e35b-6e08-48d0-bd3a-1151be19efa5"
  }
}
```

where

- The error code is **SVCSTG.VMAPP.4001020**.
- The detailed error information is: host status is not active: **abb3d0a4-f715-4932-b7ec-6dd917f65778**, **abb3d0a4-f715-4932-b7ec-6dd917f65778** and **4f68e35b-6e08-48d0-bd3a-1151be19efc6** indicate the IDs of the two ECSs where components are deployed.

Solution

- Step 1** Log in to the ECS console and click **Elastic Cloud Server**.
- Step 2** In the ECS list, search the ECS IDs in the error information to find the ECSs where the components are deployed.
- Step 3** Check whether each of their status is **Running**.

- If yes, go to **Step 2** and search for the next ECS.
- If no, go to **Step 4**.

Step 4 Either restore the status or delete the ECS.

- To restore the ECS status to **Running**: In the **Operation** column, choose **More > Start** or **More > Restart**.
- To delete an ECS you no longer use: In the **Operation** column, choose **More > Delete**.

Step 5 After you have performed **Step 2** to **Step 4** on all ECSs displayed in the error information, delete the components on the ServiceStage page again.

----End

3.12 How Do I Use the ServiceStage Source Code Deployment Function?

ServiceStage provides GitHub demos in different languages, as shown in [Table 3-1](#).

You can fork the demo of a specific language to your GitHub code repository and experience the source code deployment function of ServiceStage by referring to [Creating and Deploying Components](#).

Table 3-1 Demos provided by ServiceStage and GitHub addresses

Demo	Language	GitHub Repository Address
ServiceComb-SpringMVC	Java	https://github.com/servicestage-template/ServiceComb-SpringMVC
ServiceComb-JAX-RS	Java	https://github.com/servicestage-template/ServiceComb-JAX-RS
ServiceComb-POJO	Java	https://github.com/servicestage-template/ServiceComb-POJO
SpringBoot-WebService	Java	https://github.com/servicestage-template/SpringBoot-WebService
SpringBoot-Webapp-Tomcat	Java	https://github.com/servicestage-template/SpringBoot-Webapp-Tomcat
nodejs-express	Node.js	https://github.com/servicestage-template/nodejs-express-4-16
nodejs-koa	Node.js	https://github.com/servicestage-template/nodejs-koa-2-5-2
php-laravel	PHP	https://github.com/servicestage-template/php-laravel-v5-6-28

Demo	Language	GitHub Repository Address
php-slim	PHP	https://github.com/servicestage-template/php-slim-3-10-0

3.13 What Should I Do If Components Fail to Be Upgraded in ServiceStage Dark Launch?

Symptom

Components fail to be upgraded in ServiceStage dark launch. The following error information may be displayed:

- query microservice info failed, microservices should be registered.
- The grayscale service must be a new version.

Solution

Step 1 Determine the failure cause based on the error information.

- If the error message "query microservice info failed, microservices should be registered." is displayed, the component instance may not be a microservice or the component instance is not registered with CSE.
- If the error message "The grayscale service must be a new version." is displayed, the component instance registered with CSE is not of the new version.

Step 2 In **Deployment Record**, select a record that fails in dark launch.

Step 3 Click **Roll Back**.

Step 4 Rectify the fault based on the cause in [Step 1](#).

Step 5 Perform dark launch again and check whether it is successful.

- If yes, no further operation is required.
- If no, contact customer service.

----End

3.14 How Do I Mount Items to Modify the Configuration File of a Container-based Component?

Symptom

For components deployed in a container, the configuration files provided by the ServiceStage technology stack may not meet your service requirements. In this case, modify the files by mounting configuration items.

Solution

- Step 1** Log in to ServiceStage.
- Step 2** Choose **Environment Management** and click the environment of the component.
- Step 3** In the **Resource Settings** area, choose **Cloud Container Engine** from **Compute**.
- Step 4** Click **ConfigMap > Create Configuration Item**.
- Step 5** Select a **Creation Mode**, enter a **Configuration Name**, select the same **Cluster** and **Namespace** as the component, enter the key and value in **Configuration Data**, and click **Create Configuration Item**.
- Step 6** Choose **Component Management** and click the component you are modifying the file for. The component **Overview** page is displayed.
- Step 7** Click **Upgrade**, select an **Upgrade Type**, and click **Next**.
- Step 8** Click **Advanced Settings > Deployment Configuration > Data Storage > Local Disk**.
- Step 9** Click **Add Local Disk**. On the displayed dialog box, select **ConfigMap** for **Local Disk Type**, select the configuration item created in **Step 5**, enter the file path in **Mounting Path**, and click **OK**. (Do not enter an existing path, because overwriting it may cause path pollution.)

NOTE

Generally, the file to mount is read-only. When configuring a non-existing path, copy the file to it so that the file can be modified.

- Step 10** Click the **Startup Command** tab, enter the **Command** and **Parameter**, and click **Upgrade**.

----End

3.15 What Should I Do If the Application Name Displayed on Microservice Governance Is Different from That Displayed on ServiceStage Application Management After an Application Component Is Connected to a Microservice Engine?

Symptom

After an application component is connected to a microservice engine, the application name displayed on Microservice Governance is different from that displayed on ServiceStage Application Management. For example, after a component created and deployed under **Application Management > canary-application** is connected to the microservice engine, its application name is **canary-application-batch** displayed in **Microservice Catalog > Microservice List**.

Possible Cause

In ServiceStage, an application is a service system with complete functions and consists of one or more feature-related components.

In a microservice, an application can be regarded as a software system that implements a complete service. An application consists of multiple microservices, which can discover and call each other.

- In a Spring Cloud microservice architecture development project, the application name is usually defined in the **bootstrap.yaml** configuration file of a component in the project.
- In a Java Chassis microservice architecture development project, the application name is usually defined in the **microservice.yaml** configuration file of a component in the project.

Configuration files are stored in the **/src/main/resources/** directory of components in the current project.

After a component instance of a ServiceStage application is connected to a microservice engine, its application name is the one defined in the configuration file of the component.

Solution

To ensure application name consistency:

- In a Spring Cloud microservice architecture, change the application name in the configuration file of each component to **`${CAS_APPLICATION_NAME:basic-application}`**.
- In a Java Chassis microservice architecture, change the application name in the configuration file of each component to **`_${CAS_APPLICATION_NAME}`**.

4 Continuous Delivery

4.1 How Does ServiceStage Manage Code on IDEA?

IDEA is a local IDE. You can encode on the IDE, upload the code to a code library, and select the source code repository for deployment.

If applications are developed based on the ServiceComb framework, select the source code repository for deployment and specify an engine to manage the applications.

4.2 How Do I Add the Build Server Address to the GitLab Server Security Group?

Background

If your GitLab service is built on the intranet of a public cloud, and the public network cannot be accessed directly, add the address of the build service to your GitLab server's security group to ensure that the build task can run.

Procedure

- Step 1** Add the network segment where ServiceStage is located to the security group of the node where the GitLab private repository is located. The build service uses this IP address segment to access the GitLab service API.

For details, see [Adding a Security Group Rule](#).

NOTE

For details about the network segment where ServiceStage is located, contact technical support.

- Step 2** Obtain the cluster name and node label for creating an image.
- For application component building, obtain **Cluster** and **Node Label** by referring to [Editing a Source Code Job](#).

- For build job building, obtain **Cluster** and **Node Label** by referring to [Creating a Source Code Job](#).

Figure 4-1 Obtaining the cluster name and node label

The screenshot shows a configuration interface for a cluster. At the top, there is a 'Cluster' dropdown menu with a red box around it, containing the text 'cce-xxxx-xxxx-xxxx'. Below it, a note says 'If no cluster is available, create one. Then, click Refresh and select the cluster.' Below that is a 'Node Label' section with a table:

Key	Value	Operation
node.kubernetes.io/bareme...	false	Refresh

Below the table, a note says 'Select a node that has an EIP bound and can access the public network. If no such node exists, refer to Having an ECS Without a Public IP Address Access the Internet and create one. If the node does not have a label, create one.'

Step 3 Obtain the EIP of the node in the cluster.

- Application component building
 - a. Log in to ServiceStage and choose **Continuous Delivery > Build**.
 - b. Click the name of the target cluster to enter its details page.
 - c. Click **Nodes** to obtain the EIP of the node in the cluster.
- Job building
 - a. Log in to ServiceStage and choose **Continuous Delivery > Build**.
 - b. Select a build job and click a cluster to enter its details page.
 - c. Click **Nodes** to obtain the EIP of the node in the cluster.

Step 4 Add the running node of the build image obtained in [Step 3](#) to the security group of the node where the GitLab private repository is located. During the build, the build service accesses the GitLab service to pull the code.

For details, see [Adding a Security Group Rule](#).

----End

4.3 How Do I Add the Build Server Address to the Maven Server Security Group?

Background

Add the EIP of the build node in the build cluster to the security group of the node where the private Maven service is located to enable the build service to access the Maven server to download the dependency package.

Procedure

Step 1 Obtain the cluster name and node label for creating an image.

- For application component building, obtain **Cluster** and **Node Label** by referring to [Editing a Source Code Job](#).
- For build job building, obtain **Cluster** and **Node Label** by referring to [Creating a Source Code Job](#).

Figure 4-2 Obtaining the cluster name and node label

The screenshot shows the ServiceStage console interface. At the top, there is a 'Cluster' dropdown menu with the value 'cce-xxxx-xxxx' selected. Below it, a note says 'If no cluster is available, create one. Then, click Refresh and select the cluster.' Below that is a 'Node Label' section with a table:

Key	Value	Operation
node.kubernetes.io/bareme...	false	Refresh

Below the table, a note says 'Select a node that has an EIP bound and can access the public network. If no such node exists, refer to [Having an ECS Without a Public IP Address Access the Internet and create one](#). If the node does not have a label, [create one](#).'

Step 2 Obtain the EIP of the node in the cluster.

- Application component building
 - a. Log in to ServiceStage and choose **Continuous Delivery** > **Build**.
 - b. Click the name of the target cluster to enter its details page.
 - c. Click **Nodes** to obtain the EIP of the node in the cluster.
- Job building
 - a. Log in to ServiceStage and choose **Continuous Delivery** > **Build**.
 - b. Select a build job and click a cluster to enter its details page.
 - c. Click **Nodes** to obtain the EIP of the node in the cluster.

Step 3 Add the EIP of the build node in the build cluster to the security group of the node where the private Maven service is located.

For details, see [Adding a Security Group Rule](#).

----End

4.4 What Should I Do If I Failed to Use ServiceStage to Build a Job?

There are many causes for software engineering build failures. You can locate faults using the following methods:

Code Cannot Be Pulled

See the following figure.

```
Scheduled running onto node "192.168.███".
> Starting...
> Preparing build environments...
> fetch build scripts...
> initialize system...
add paas user...
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
Adding user paas to group docker
> create workspace...
> init default tool...
> init tool java-8...
init jdk-8u181-linux-x64.tar.gz
init apache-maven-3.5.4-bin.tar.gz
init apache-ant-1.10.5-bin.tar.gz
init gradle-4.9-bin.tar.gz
> define util functions...
> init 3rd tool configuration...
> init code repository configuration...
,execute common/pull_code.sh...
> git clone https://github.com/██████████/SpringBoot-Webapp-Tomcat.git branch master
Cloning into 'SpringBoot-Webapp-Tomcat'...
fatal: unable to access 'https://github.com/██████████/SpringBoot-Webapp-Tomcat.git/': Failed connect to github.com:443; Connection ti
Attempt 1 failed! Trying again in 1 seconds...
```

The possible causes are as follows:

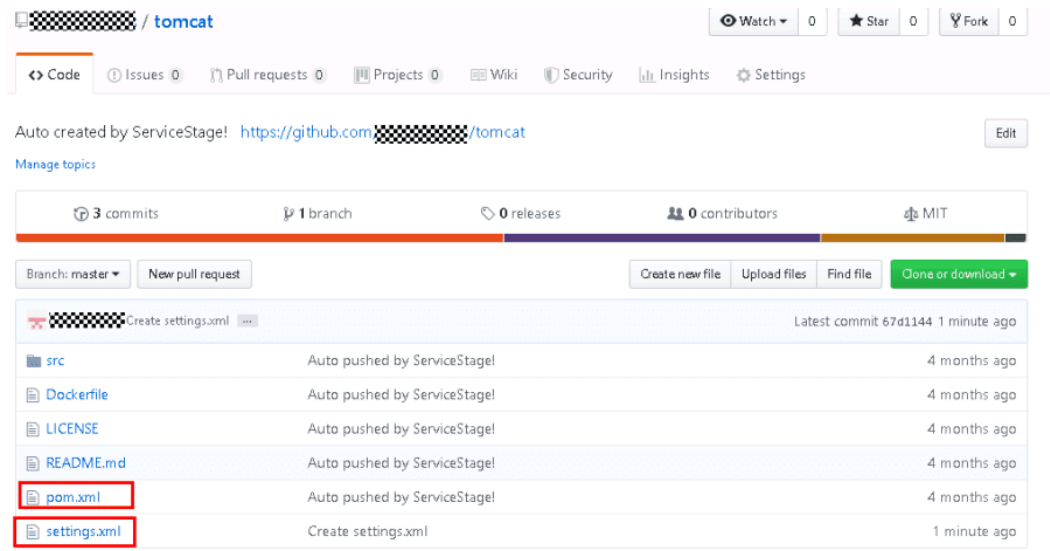
- If an application is built on your own node, the node may not be bound to an EIP. For example, in the preceding figure, node 192.168.x.x is not bound to an EIP. Solution: Bind an EIP to the node. For details, see [Assigning an EIP and Binding It to an ECS](#).
- The authorization information has expired, the private token of the code source has low permission, or the authorization information has been removed. For example, code cannot be obtained from the code source of CodeArts. This may be because the user password used during authorization is incorrect. Solution: Authorize the repository again. For details, see [Authorizing a Repository](#).
- The network between the self-built code source repository and the build node is disconnected. For example, a private Bitbucket is set up on a node in cluster A and the build is performed in cluster B. However, cluster B and cluster A are in different VPCs. As a result, their network is disconnected, and code cannot be pulled up in building. Solution: Connect the network.

Built Code Depends On the Private Maven Repository

There are two solutions:

- Add the **settings.xml** file to the root directory of your project, and specify the private Maven repository address in the **settings.xml** file. If your private Maven repository needs to be authenticated, configure your authentication information, username, and password in the **settings.xml** file.
- Specify a private Maven repository in the **pom.xml** file of your project.

The following is an example of the paths of the **settings.xml** and **pom.xml** files:



Dockerfile for Source Code Job Building Is Incorrectly Set


For details about how to write a Dockerfile, see the [official website](#) or [demo](#) generated based on ServiceStage templates.



```
Branch: master java1 / Dockerfile Find file Copy path
service Auto pushed by ServiceStage! 84691c4 8 days ago
0 contributors
9 lines (5 sloc) | 289 Bytes Raw Blame History
1 FROM openjdk:8u181-jdk-alpine
2
3 WORKDIR /home/apps/
4
5 COPY target/*.jar app.jar
6
7 RUN sh -c 'touch app.jar'
8
9 ENTRYPOINT [ "sh", "-c", "java -Djava.security.egd=file:/dev/./urandom -jar -Xmx256m app.jar" ]
```

Project Code Depends On the CSE SDK and the CodeArts Private Maven Repository

Perform the following steps:

- Step 1** Add the **settings.xml** file to the root directory of your project.
- Step 2** Log in to the CodeArts private dependency repository and select the specified Maven private dependency repository from the repository list on the left.
- Step 3** Click  in the upper right corner and choose **Configuration Guide** from the shortcut menu.
- Step 4** Click **Download Configuration File** to download the **settings.xml** file.
- Step 5** Modify the downloaded **settings.xml** file as follows:

1. Add **!HuaweiCloudSDK** to **<mirrorOf>**.

```
<mirror>
  <id>z_mirrors</id>
  <mirrorOf>*,!releases,!snapshots,!HuaweiCloudSDK</mirrorOf>
  <url>https://repo.huaweicloud.com/repository/maven</url>
</mirror>
```

2. Add a Maven repository under **<repositories>** in **<profiles>**.

```
<repository>
  <id>HuaweiCloudSDK</id>
  <url>https://repo.huaweicloud.com/repository/maven/huaweicloudsdk</url>
  <releases>
    <enabled>true</enabled>
  </releases>
  <snapshots>
    <enabled>true</enabled>
  </snapshots>
</repository>
```

- Step 6** Place the modified **settings.xml** file in the root directory of your project and build the job.

----End

Customizing Dockerfile When Software Package Building Job Is Created

When you select a software package, the system automatically uploads the software package to the current working directory of the image. The following shows a Dockerfile example.


```
1 FROM swr.cn-north-4.myhuaweicloud.com/image/tomcat:v1.0.1
2 RUN rm -rf /usr/local/tomcat/webapps/*
3 COPY ROOT.war /usr/local/tomcat/webapps/ROOT.war
```

5 Software Center

5.1 What If a Software Package Fails to Be Uploaded?

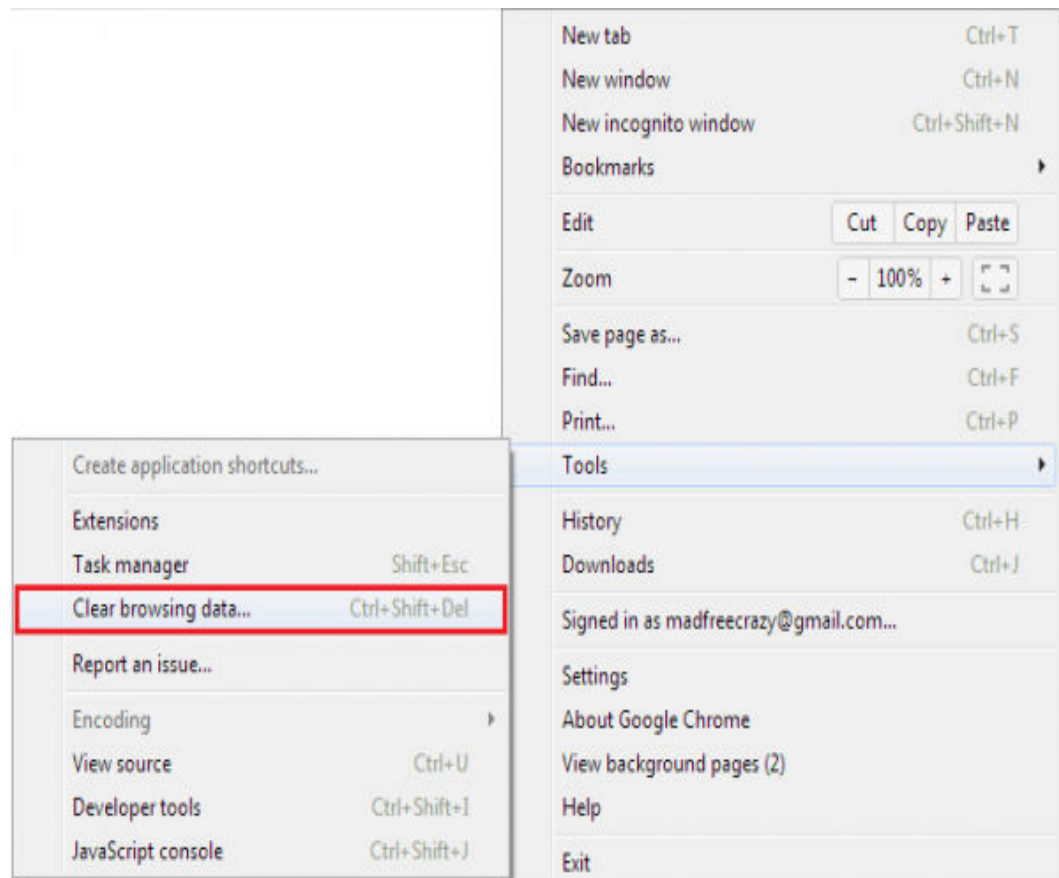
Symptom

When a software package that meets the system requirements is uploaded, the system displays **No access permission. Contact the administrator.**

Solution

Step 1 In Google Chrome, choose **More tools > Clear browsing data.**

Figure 5-1 Clearing browsing data



Step 2 In the displayed **Clear browsing data** dialog box, retain the default settings and click **CLEAR DATA**.

----End

5.2 What If the Docker Client Fails to Push Images?

Symptom

A tenant or user logs in to the Docker client and runs the following command to push an image:

```
docker push 10.125.54.133:20202/test1/busybox:latest
```

NOTE

- *10.125.54.133:20202* indicates the IP address and port number of the repository to which the tenant or user is to push an image.
- *test1* indicates a namespace.

However, the image fails to be pushed, and the following information is displayed on the Docker client:

```
unauthorized: authentication required
```

Solution

Step 1 Use a correct AK/SK to log in to the Docker client.

Step 2 Push an image to a namespace on which the current tenant or user has the operation permissions, or replace the namespace with a new one.

- Run the following command to push an image to a namespace on which the current tenant or user has the operation permissions:

```
docker push 10.125.54.133:20202/test2/busybox:latest
```

 **NOTE**

- *10.125.54.133:20202* indicates the IP address and port number of the repository to which the tenant or user is to push an image.
- *test2* indicates the namespace on which the tenant or user has the operation permissions.
- Run the following command to replace the namespace with a new one:

```
docker push 10.125.54.133:20202/test3/busybox:latest
```

 **NOTE**

- *10.125.54.133:20202* indicates the IP address and port number of the repository to which the tenant or user is to push an image.
- *test3* indicates a new namespace.

Step 3 After the image is pushed, the following information is displayed:

```
The push refers to a repository [10.125.54.133:20202/test2/busybox]
6a749002dd6a: Pushed
latest: digest: sha256:ecb3f3e96e003af6e02f0f47ac4d25a3b0585db54de0a82bb070f8cb78a79bc7 size: 527
```

If an exception occurs, contact technical support.

----End

5.3 What Should I Do If I Cannot Download an SWR Software Package?

Symptom

When a Tomcat application is created, a message is displayed, indicating that the creation fails. Tomcat logs show that the authentication fails when the SWR software package is downloaded. Error 401 is displayed when the SWR software package is downloaded manually.

Solution

Set the image to a public one. Private packages cannot be obtained, due to insufficient permissions.

6 Infrastructure

6.1 Are Existing Programs Affected If I Unsubscribe Servers?

Question

Are existing programs affected if I unsubscribe servers?

Solution

- Container-based deployment: When a server is unsubscribed, the service instances deployed on it are rescheduled in the CCE cluster.
- VM-based deployment: When a server is unsubscribed, the service instances deployed on it become unavailable and are not rescheduled.

6.2 What Should I Do If the Service Registration Fails After IPv6 Is Enabled for the Exclusive Microservice Engine with Security Authentication Enabled?

Symptom

A microservice developed based on Java Chassis is registered with the exclusive microservice engine with security authentication enabled. The microservice registry center address is the IPv4 address of the microservice engine registry center. The microservice can be successfully registered and started.

If the microservice registry center address is changed to the IPv6 address of the microservice engine registry center, the registration fails and the error "java.net.SocketException: Protocol family unavailable" is reported.

Possible Cause

If you select the VPC network with IPv6 enabled when creating an exclusive microservice engine, the engine supports the IPv6 network. If the service is

deployed using a container through an IPv6 network segment, the IPv6 dual-stack function must be enabled for the selected CCE cluster.

If IPv6 is not enabled for the selected cluster, the service network is disconnected, and the error "java.net.SocketException: Protocol family unavailable" is reported.

Solution

Step 1 Modify the environment where the microservice application is deployed by adding a CCE cluster with the IPv6 dual-stack function enabled.

Modify the environment. For details, see [Modifying an Environment](#).

Step 2 Redeploy the application. For details, see [Creating and Deploying Components](#).

----End

6.3 What Should I Do If a Non-Microservice Engine Error Occurs When I Operate an Exclusive Microservice Engine?

Symptom

When you create, delete, or upgrade an exclusive microservice engine, a non-microservice engine error may occur.

For example, when you create an exclusive microservice engine, the cluster fails to be deployed and the following error message is displayed:

```
{"error_code":"SVCSTG.00500400","error_message":{"kind":"Status","apiVersion":"v1","metadata":{},"status":{"Failure"},"code":400,"errorCode":"CCE.01400013","errorMessage":"Insufficient volume quota.","error_code":"CCE_CM.0307","error_msg":"Volume quota is not enough","message":"volume quota checking failed as [60/240] insufficient volume size quota","reason":"QuotaInsufficient"}}
```

Solution

The displayed error information contains the error code of the corresponding service. Contact the corresponding technical support.

6.4 How Do I Switch a Cluster Used for Component Building from a Shared Cluster to a Private One?

Symptom

The ServiceStage shared cluster is shared by all users. It may have security and resource isolation vulnerabilities.

To maintain security and reliability, configure applications on the shared cluster to a private cluster as soon as you bring the shared cluster offline.

Prerequisites

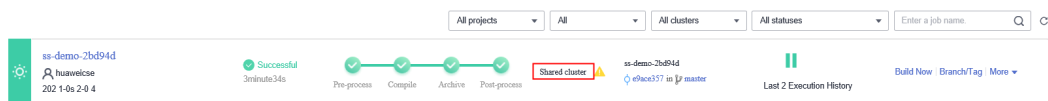
1. A cluster has been created. For details, see [Buying a Cluster](#).
2. An EIP has been bound to the CCE cluster node. For details, see [Assigning an EIP and Binding It to an ECS](#).

Solution

Step 1 Log in to ServiceStage and choose **Continuous Delivery > Build**.

Step 2 Select **All projects**, **All**, **Shared clusters**, and **All statuses**. The filtered results are the clusters you need to switch.

- If there are no results, no further action is required.
- If there are filter results, go to [Step 3](#).



Step 3 Go to the page for editing the build job, depending on creation type.

- If the creation type is **System created**, choose **Application Management > Application Component**, click the name of the component, and choose **Build Job > Edit**.
- If creation type is **User created**, locate the row that contains the target build name, click **More** in the **Operation** column, and select **Edit**.

Step 4 Edit the build job, depending on how the components are deployed.

- For components deployed using source code, see [Editing a Source Code Job](#).
- For components deployed using software packages, see [Editing a Package Job](#).

Specifically:

1. Select **Build with your own cluster** for **Clusters**.
2. In the **Clusters** drop-down list, select a private build job.

----End

6.5 What Should I Do If the Access Address Fails to Be Processed During CSE Creation?

Symptom

During engine creation, the access address fails to be processed, and the following error message is displayed:

```
{"error_code":"SVCSTG.00500404","error_message":{"code":"VPC.0202","message":"Query resource by id xxx fail.the subnet could not be found."}}
```

Possible Cause

CSE is not authorized in the user's project.

Solution

- When you use CSE instances provisioned from ServiceStage and want to provision new instances in CSE, you need to grant permissions to CSE. For details, see [Creating a User and Granting Permissions](#).
- CSE depends on VPC. If you do not grant any permission, create a cloud service agency `cse_admin_trust` by referring to [Creating an Agency \(by a Delegating Party\)](#).

6.6 What Do I Need to Know Before Upgrading an Exclusive Microservice Engine?

There are several problems that may occur during and after the upgrade from 1.x to 2.x. The following are their symptoms and solutions.

- **Symptom 1:** During the upgrade, the configuration could not be obtained or updated using the API, showing the error "Connection refused" or "Connection was closed". Error examples:
[ERROR] Config update from xxx.xxx.xxx.xx failed. Error message is [Connection refused: xxx.xxx.xxx.xx]. org.apache.servicecomb.config.client.ConfigCenterClient\$ConfigRefresh.lambda\$null\$13(ConfigCenterClient.java:428)
[ERROR]Config update from xxx.xxx.xxx.xx failed. Error message is [Connection was closed]. org.apache.servicecomb.config.client.ConfigCenterClient\$ConfigRefresh.lambda\$null\$13(ConfigCenterClient.java:428)
Solution: The configuration center restarts for a short period of time when upgrading. During this restart, the configuration is obtained or updated, causing a disconnection error. To avoid this error, do not update the configuration during the upgrade.
- **Symptom 2:** Service scenario governance is unavailable when using an engine 1.x to access the configuration center.
Solution: The configuration center of engine 2.x has changed to kie, so switch the access mode of the configuration center to kie. For details, see [Using the Configuration Center in Spring Cloud](#).
- **Symptom 3:** When you import a configuration file in the original 2.x configuration center format, the import fails and a message is displayed indicating that the file is empty or the format is incorrect.
Solution: Change the format of the configuration items in the configuration file as required by the engine 2.x. The new configuration file is a JSON file with the following format:

```
{
  "data":[
    {
      "key":"xxx",
      "labels":{"
        "environment":"xxx",
        "service": "xxx",
        "app": "xxx",
        "version": "xxx"
      },
      "value":"xxx",
      "value_type":"text",
      "status":"enabled"
    },
  ],
}
```



```
{
  "key": "xxx",
  "labels": {
    "environment": "xxx"
  },
  "value": "xxx",
  "value_type": "text",
  "status": "enabled"
},
{
  "key": "xxx",
  "labels": {
    "environment": "xxx",
    "service": "xxx"
  },
  "value": "xxx",
  "value_type": "text",
  "status": "enabled"
},
{
  "key": "xxx",
  "labels": {
    "environment": "xxx",
    "service": "xxx",
    "app": "xxx"
  },
  "value": "xxx",
  "value_type": "text",
  "status": "enabled"
}
]
```

where,

- **key** and **value** are mandatory: key and value of the configuration item
- **labels** is mandatory: configuration range, determined by setting the **environment**, **service**, **app**, and **version** fields
- **value_type** is mandatory: configuration item type. Value: **ini**, **json**, **text** (default), **yaml**, **properties**, or **xml**
- **status** is optional: whether to enable the configuration. Value: **enabled** or **disabled** (default)
- **Symptom 4:** If the global configuration is set in the configuration center of microservice engine version 1.x, after the version is upgraded to 2.x, the global configuration automatically adjusts the application scope **environment=\${environmentName}**. The value of *environmentName* can be **empty**, **development**, **testing**, **acceptance**, or **production**. If the SDK uses Kie as the configuration center, you need to add a custom tag to the project configuration file to obtain the configuration. The following uses **environment=production** as an example:

spring-cloud-huawei framework:

```
spring:
  cloud:
    servicecomb:
      config:
        serverType: kie
      kie:
        customLabel: environment
        customLabelValue: production
```

servicecomb-java-chassis framework:

```
servicecomb:  
  kie:  
    customLabel: environment  
    customLabelValue: production
```

7 Application O&M

7.1 Why Can't I View ServiceStage Logs?

Possible reasons why logs cannot be viewed on ServiceStage are:

- ICAgent was not installed on the host you are trying to view logs on.
- User service logs were output to a non-standard location.

Solution

- If ICAgent is not installed on your host:
ServiceStage log viewing is provided by the Application Operations Management (AOM) service and requires ICAgent because it is the AOM collector running on each host to collect metrics, logs, and application performance data in real time.
For details, see [Installing an ICAgent](#).
- If logs are output to a non-standard location:
Since log policies are user-defined, the service logs of the user program are not output to the standard output location. Perform the following steps:
 - VM-based deployment
Check whether the configured log policy writes the user application service logs to the default VM log directory specified by ServiceStage: `(/var/log/application/${Component name}-${Environment name}-${Random character string}/${Version number}/${Instance ID}/start_app.log)`.
Query the service code and adjust the log policy.
 - Container-based deployment
Check the configured log policy to determine where service logs are output to. For details, see [Configuring Application Log Policies](#).

7.2 What Should I Do If Application Access Mode Becomes Invalid When EIP Is Replaced?

Symptom

When I bind a load balancer to an application and replace its EIP, the application access mode cannot be automatically updated.

Solution

Manually delete the original EIP, add a new EIP, and use the new EIP for ELB access.

7.3 What Should I Do If the Memory Usage of a Node Becomes Too High After a New Service Is Started?

Symptom

What should I do if the memory usage of a node is too high after a new service is started?

Solution

Set affinity by referring to [Configuring Application Scheduling Policies](#) so that service instances to be deployed based on affinity nodes.

7.4 How Do I Uninstall a Service?

Step 1 Log in to ServiceStage.

Step 2 Choose **Application Management**.

Step 3 Click a target application. The **Overview** page is displayed.

Step 4 Select all components and click **Bulk Delete**.

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

Step 6 Choose **Microservice Engines > Engines**.

Step 7 Select the target microservice engine from the **Microservice Engine** drop-down list in the upper part of the page.

Step 8 Choose **Microservice Catalog > Microservice List**.

- For microservice engines with security authentication disabled, go to [Step 10](#).
- For microservice engines with security authentication enabled, go to [Step 9](#).

Step 9 In the displayed **Security Authentication** dialog box, enter the account name and password, and click **OK**.

 **NOTE**

- If you connect to the microservice engine for the first time, enter the root account and the password using in [Creating a Microservice Engine](#).
- For details about how to create an account, see [Adding an Account](#).

Step 10 Delete the microservice of the service to completely uninstall the service.

 **NOTE**

To delete a resource, obtain the corresponding permissions. If a message is displayed indicating that you do not have the permissions, apply for the permissions by referring to [ServiceStage Permissions](#).

----End

8 Application Development

8.1 What Are the Differences Between the Microservice and Common Application?

The microservice is an architectural model. Its general idea is to break an application into multiple parts. Therefore, an application using the microservice architecture is actually a distributed application.

This enables services to change faster, and the system to be more reliable.

Type	Microservice	Common Application
Development	<p>The workload of a microservice is light. A two-pizza team can rewrite all the code of a microservice in two weeks. This can be used as a symbol of microservices. When developing a microservice, its APIs need to be available for interconnection with other microservices. Therefore, the API definition-based development mode is highly recommended.</p> <p>For details about microservice development, see Developing Microservice Applications.</p>	<p>Complex logic, coupled modules, bloated code, difficult modification, and low version iteration efficiency.</p>

Type	Microservice	Common Application
Deployment	<p>An application consisting of multiple microservices is complex. Orchestration is required when the application is deployed.</p> <p>For details about microservice application deployment, see Creating and Deploying Components.</p>	<p>Applications are probably large and require much time to build and deploy. This is not conducive to frequent deployment and hinders continuous delivery. This problem is especially serious in mobile application development.</p>
O&M	<p>Microservice O&M focuses on governance, in addition to metrics monitoring and log collection. The core concept of microservice governance is to maintain system performance through modifications while the system is running.</p> <p>For details about application O&M, see Component O&M.</p>	<p>It takes a long time to rectify common online problems. To rectify any online problem, the entire application system must be upgraded.</p>

8.2 How Do I Handle a Microservice Registry Failure (Java Chassis)?

When a microservice is successfully deployed, register it with the service center and the configuration center to enable the microservice discovery and governance capabilities. The microservice registry fails when any of the following conditions are met:

- The AK/SK is not configured or is incorrect.
- The address of the service center or configuration center is incorrect.
- The network connection is faulty.
- The domain name resolution fails.
- The monitoring port is already occupied.

Fault Locating

- If the following error message is displayed, the AK/SK information is incorrectly configured or carried in the request header.

```
{"errorCode":"401002","errorMessage":"Request unauthorized","detail":"Invalid request, header is invalid, ak sk or project is empty."}
```

Checking method:

- a. Check whether the project depends on the following authentication modules (indirect dependency is permitted, for example, the project depends on the cse-solution-service-engine):

```
<groupId>com.huawei.paas.cse</groupId>  
<artifactId>foundation-auth</artifactId>
```

- b. Check whether AK/SK in the **microservice.yaml** file is correct. To obtain correct AK/SK, see [Managing Access Keys](#).

```
cse:
  credentials:
    accessKey: your access key
    secretKey: your secret key
    akskCustomCipher: default
```

- If the following error message is displayed, the AK/SK information is incorrectly configured.

```
{"errorCode":"401002","errorMessage":"Request unauthorized","detail":"Get service token from iam proxy failed,{\"error\":\":validate ak sk error\"}"}
```

Checking method:

Check whether AK/SK in the **microservice.yaml** file is correct. To obtain correct AK/SK, see [Managing Access Keys](#).

```
cse:
  credentials:
    accessKey: your access key
    secretKey: your secret key
    akskCustomCipher: default
```

- If the following error message is displayed, the project information is incorrectly configured.

```
{"errorCode":"401002","errorMessage":"Request unauthorized","detail":"Get service token from iam proxy failed,{\"error\":\":get project token from iam failed. error:http post failed, statuscode: 400\"}"}
```

Checking method:

Check whether the project information in the **microservice.yaml** file is correct. For details, see [Viewing the Project Name](#).

```
cse:
  credentials:
    accessKey: your access key
    secretKey: your secret key
    akskCustomCipher: default
  project: cn-north-1
```

- If the following error message is displayed, there is insufficient quota to add a service instance.

```
{"errorCode":"400100","errorMessage":"Not enough quota","detail":"no quota to create instance, ..."}
```

Checking method:

Log in to the public cloud and view the instance quota on the microservice engine page. If the quota is sufficient, check the service center address and region information configured in the code. Note that you need to check the instance quota of the region where the instance is located.

- If the microservice fails to access the service center or configuration center, the following information is displayed. The microservice does not access the service center or configuration center. Therefore, no error code can be found in the service center or configuration center.

```
Connection refused: no further information
```

Checking method:

- a. Check whether IP addresses of the service center and configuration center in the **microservice.yaml** file are correct. If not, correct them.

```
cse:
  service: //Information about the service center. The address field
            indicates the service center address.
  registry:
    address: https://cse.cn-north-1.myhuaweicloud.com
  instance:
    watch: false
```



```
config: //Information about the configuration center. The address field
indicates the configuration center address.
client:
  serverUri: https://cse.cn-north-1.myhuaweicloud.com
  refreshMode: 1
  refresh_interval: 5000
```

- b. If IP addresses of the service center and configuration center are correct, run the following commands to check whether the network is normal.

```
ping <servicecenter ip>
```

```
ping <configurationcenter ip>
```

If **ping** command execution is successful, the network connection is normal.

NOTE

If the address of the service center or configuration center is a domain name, change the address configured in the **microservice.yaml** file to the domain name. Then, run the **ping** command.

- c. If the network is normal, run the following command to obtain the IP address of the service center or configuration center.

```
ping <domain name>
```

If the following information is displayed, configure the obtained IP address and domain name in the local **/etc/hosts** file.

```
10.153.78.18 cse.cn-north-1.myhuaweicloud.com
```

- In addition, if a microservice port number is occupied, the microservice may fail to start. In this case, you can run the following command to check whether the service monitoring port is occupied.

```
netstat -ano | findstr 8080
```

If the port is occupied by another application, modify the **microservice.yaml** file and change the monitoring port to an unoccupied port.

```
rest:
  address: 0.0.0.0:8087 //Microservice port. Ensure that the port number is unique.
```

8.3 How Do I Troubleshoot Microservices Deployed on the Cloud?

You can use Dashboard to locate a fault. Dashboard allows you to check the real-time running status of all microservices and instances.

After locating a faulty node, you can use Application Performance Management (APM) to check running logs of the node for further analysis.

8.4 Should I Use the SDK or ServiceMesh to Build a Microservice?

- The SDK applies to self-governed microservices, which enables offline debugging. These microservices need to be developed based on the SDK.

- The microservice built using ServiceMesh requires a ServiceMesh environment during deployment. ServiceMesh enables microservices to be easily developed, and SDKs are not required.

ServiceMesh Scenarios

- Reconstructing service code written in non-Java language into microservices
- Reconstructing old Java services to microservices
- Interconnecting a service that is not compiled in Java SDK with a service compiled in Java SDK

JAVA SDK Scenarios

- Using distributed transactions
- Using Java to compile microservices.
- Using protocols except HTTP 1.1 (HTTP 1.1 is the only protocol supported by ServiceMesh)

8.5 What If I Fail to Obtain a Dependency?

When the Maven image source is configured, the dependency fails to be obtained, as shown in [Figure 8-1](#).

Figure 8-1 Failed to obtain the dependency

```
[INFO] Scanning for projects...
[INFO] Downloading from <http://maven.com/nexus/content/groups/public/com...>: aaas/cse-dependency/2.3.12/cse-dependency-2.3.12.pom
[ERROR] [ERROR] Some problems were encountered while processing the POMs:
[WARNING] 'build.plugins.plugin.version' for org.springframework.boot:spring-boot-maven-plugin is missing. @ line 74, column 21
[ERROR] Non-resolvable import POM: Could not transfer artifact com...:aaas:cse-dependency:pom:2.3.12 from/to mirrorId (<http://maven.com/nexus/content/groups/public>); connect timed out @ line 28, column 25
@
[ERROR] The build could not read 1 project -> [Help 1]
[ERROR]
[ERROR] The project com.service:hellowordprovider:0.0.1-SNAPSHOT (D:\workspace\hellowordprovider\pom.xml) has 1 error
[ERROR] Non-resolvable import POM: Could not transfer artifact cor...:aaas:cse-dependency:pom:2.3.12 from/to mirrorId (<http://maven.com/nexus/content/groups/public>); connect timed out @ line 28, column 25 -> [Help 2]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] <http://wiki.apache.org/confluence/display/MAVEN/ProjectBuildingException>
[ERROR] [Help 2] <http://wiki.apache.org/confluence/display/MAVEN/UnresolvableModelException>
```

If you need to use a proxy to access the external network, configure the Maven proxy. Specifically, configure a proxy in the **setting.xml** file (user configuration) in the **m2** directory in the user directory (for example, **C:\Users\yang***** in the Windows OS) or the **setting.xml** file (system global configuration) in the **conf** directory in the Maven installation directory.

Find the tags in the **setting.xml** file and configure the proxy information, as shown in the following example.

```
<proxies>
  <!-- proxy
  | Specification for one proxy, to be used in connecting to the network.
  |
  -->
  <proxy>
    <id>self-defined proxy ID. The proxy configuration must be unique. </id>
    <active>true</active>
    <protocol>http</protocol>
    <username>proxy authentication account</username>
    <password>proxy authentication password</password>
    <host>enterprise's proxy address</host>
    <port>port number of the proxy address</port>
```

```
</proxy>

<proxy>
  <id>self-defined proxy ID. The proxy configuration must be unique. </id>
  <active>true</active>
  <protocol>http</protocol>
  <username>proxy authentication account</username>
  <password>proxy authentication password</password>
  <host>enterprise's proxy address</host>
  <port>port number of the proxy address</port>
</proxy>

</proxies>
```

8.6 What Is Service Name Duplication Check?

Question

What is service name duplication check?

Solution

Microservice names, applications, versions, and environments are checked.

A primary key uniquely identifies a microservice.

Ensure that each primary key is unique.

8.7 Why Do I Have to Define Service Contracts?

The enterprise-level systems are in large scale use and involve many microservice components. Therefore, unified API management is a key requirement of enterprises. CSE uses contract management to meet this requirement.

For management: Through contract management, API definition files that comply with API description standards for microservices are defined. In this way, API development of multiple development teams can be standardized and coordinated. This reduces communication costs and facilitating management.

For development: During microservice development, different teams or even different independent software vendors (ISVs) can develop the same application or system based on the unified API definition file. This facilitates consistency maintenance for the overall system. Specifically, modules in a monolithic application are called using code. Therefore, any API incompatibility can be resolved during the early compilation with low bug fixing costs. When microservices are decoupled, services are remotely called. Therefore, API inconsistency cannot be found during early compilation, resulting in high bug fixing costs. Using service contracts, contract design can be assured, changes can be strictly reviewed, and code can be reversely generated. This ensures API compatibility.

In addition, for small-scale systems that do not have high requirements on unified management, API definition files can be automatically generated through APIs.

8.8 Why Are Microservice Development Framework and Netty Versions Unmatched?

Symptom

During development of a microservice application, the following error is displayed:

```
"Caused by: java.lang.NoSuchMethodError:  
io.netty.handler.codec.http.websocketx.WebSocketClientHandshakerFactory.newHandshaker(Ljava/net/  
URI;Lio/netty/handler/codec/http/websocketx/WebSocketVersion;Ljava/lang/String;ZLio/netty/handler/codec/  
http/HttpHeaders;IJJ)Lio/netty/handler/codec/http/websocketx/WebSocketClientHandshaker;"
```

Possible Cause

Third-party software introduced a mismatched version dependency.

Solution

Run the **mvn dependency:tree** command in the development environment to view the dependency tree and check whether the microservice development framework and Netty versions are matched.

For ServiceComb 2.0.1 development framework, the matched Netty version is 4.1.45.Final.

For details about how to use Maven to manage complex dependencies, see https://servicecomb.apache.org/cn/docs/maven_dependency_management/.