ServiceStage

FAQs

Issue	01
Date	2025-08-25

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

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.serivcecomb**.

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**.

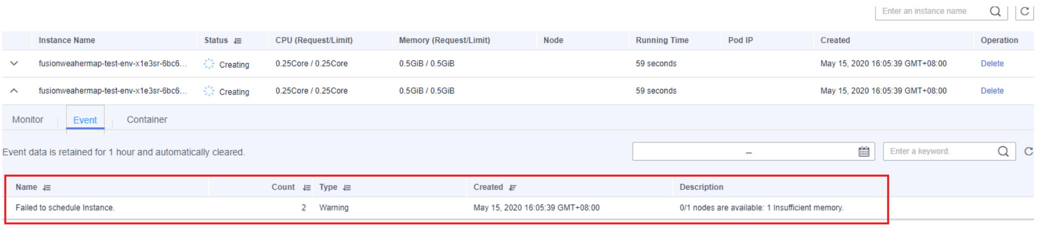
Step 3 In the instance list, click ⌄ next to the target instance.

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

----End

3.2 What 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-x1e3sr-6bc6...	Creating	0.25Core / 0.25Core	0.5GiB / 0.5GiB		59 seconds		May 15, 2020 16:05:39 GMT+08:00	Delete
	fusionweahermap-test-env-x1e3sr-6bc6...	Creating	0.25Core / 0.25Core	0.5GiB / 0.5GiB		59 seconds		May 15, 2020 16:05:39 GMT+08:00	Delete

Event data is retained for 1 hour and automatically cleared.

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 .

3.3 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.

3.4 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"/>
```

3.5 How Do I Use a Fixed Application Component IP?

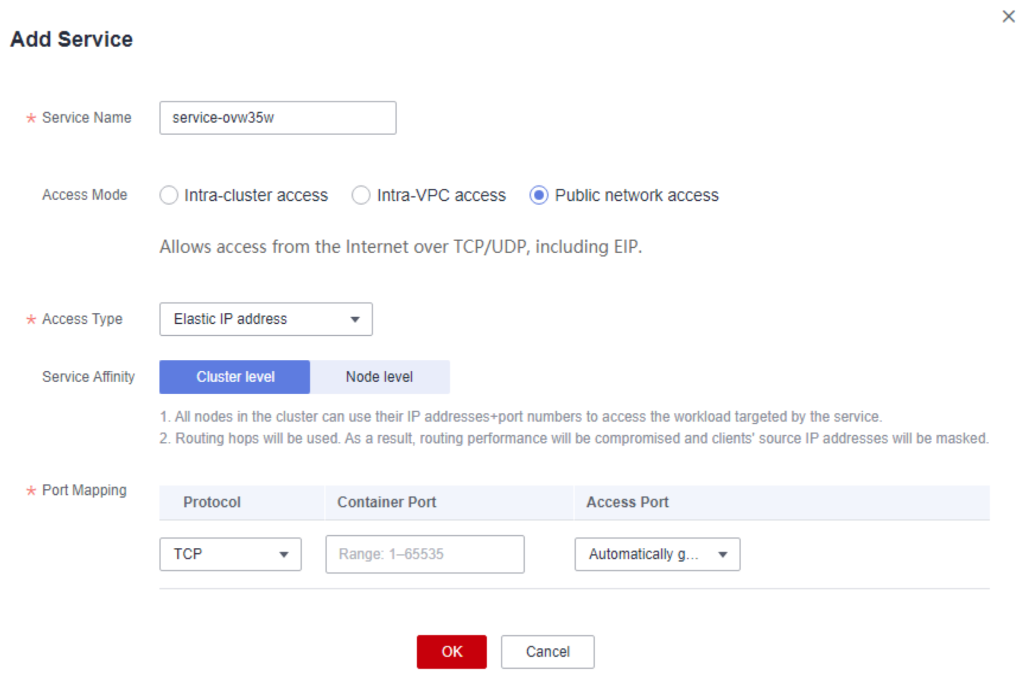
Symptom

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: service-ovw35w

Access Mode: ○ Intra-cluster access ○ Intra-VPC access ● Public network access

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

* Access Type: Elastic IP address

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
TCP	Range: 1–65535	Automatically g...

3.6 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**.

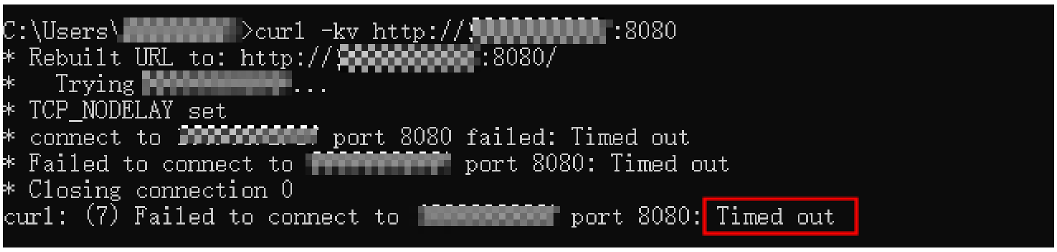
Step 2 Contact technical support.

----End

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

3.8 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.

3.9 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

3.10 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 a Component**.

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**

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

3.11 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

3.12 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

3.13 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

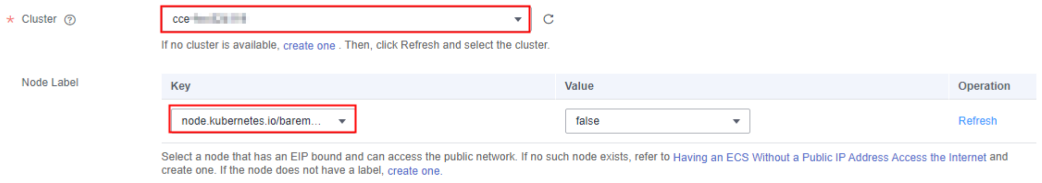
For details about the network segment where ServiceStage is located, contact technical support.

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



* Cluster ⓘ cce[redacted]

If no cluster is available, create one . Then, click Refresh and select the cluster.

Node Label

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

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

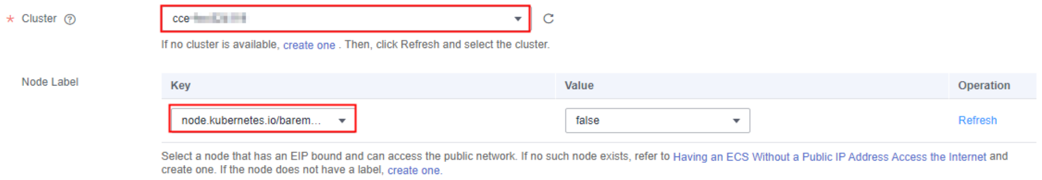
Add the EIP of the build node in the build cluster to the security group of the node where the private Maven service is located to enable the build service to access the Maven server to download the dependency package.

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



* Cluster

cce-[redacted]

If no cluster is available, create one . Then, click Refresh and select the cluster.

Node Label

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

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 Do I Do If ServiceStage Failed to Build a Job?

There are many causes for software engineering build failures. Troubleshoot them using the following methods.

Build Error After Scheduling Build Job to Containerd Container Engine Node in CCE Cluster

Symptom

The build failed and the build log displays one of the following error messages:

- /proc/sys/user/max_user_namespaces needs to be set to non-zero.
- /proc/sys/user/max_user_namespaces=100 may be low. Consider setting to >= 1024.

Cause Analysis

The build job is scheduled to the Containerd container engine node of the CCE cluster; therefore, rootless is required to ensure build security. During this process, namespaces need to be created. Settings of the node VM must meet the build requirements. However, the default **max_user_namespaces** of some VM images is **0** or too small. As a result, the build fails and an error is reported.

Solution

Step 1 Log in to all Containerd container engine nodes in the cluster where the build job is executed as user **root**.

Step 2 Run the following command to set **max_user_namespaces** of the VM image to **1024** by default and confirm the setting:

```
echo 1024 > /proc/sys/user/max_user_namespaces
cat /proc/sys/user/max_user_namespaces
```

Step 3 Restart the build job. For details, see **Starting a Build Job**.

- If the build is successful, no further action is required.
- If the build fails, contact customer service for assistance.

----End

Build Error After Configuring Taint and Tolerance Policy for CCE Cluster Node

Symptom

The build failed and the build log displays this error message:

```
0/1 nodes are available: 1 node(s) had untolerated taint {node.kubernetes.io/route-unschedulable: }.
preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.
```

Cause Analysis

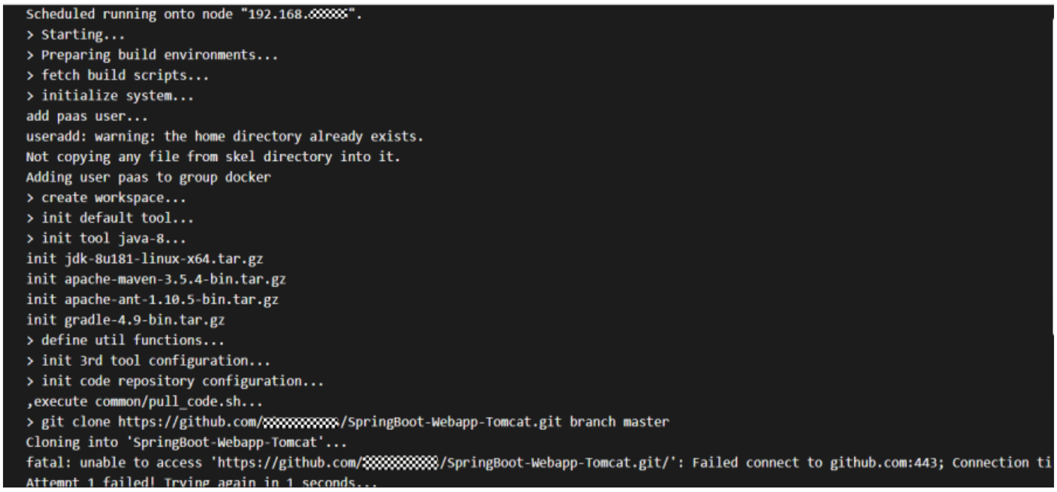
The build job is scheduled to a restricted scheduling node of the CCE cluster, and taint management is configured for the node (as shown by **node.kubernetes.io/route-unschedulable** in the error). The taint makes the node exclude some pods from its scheduling. This also causes the other nodes in the CCE cluster to be unavailable for scheduling.

Solution

Remove the taint on the restricted scheduling node to ensure that at least one cluster node is available for scheduling. For details, see **Managing Node Taints**.

Code Cannot Be Pulled

See the following figure.



```
Scheduled running onto node "192.168.[redacted]".
> 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/[redacted]/SpringBoot-Webapp-Tomcat.git branch master
Cloning into 'SpringBoot-Webapp-Tomcat'...
fatal: unable to access 'https://github.com/[redacted]/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:

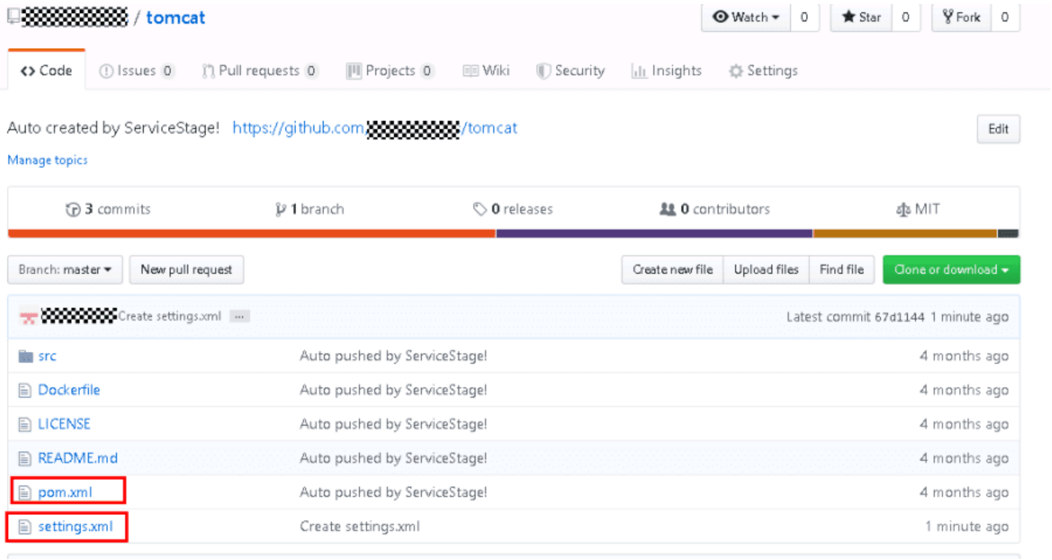
- If an application is built on your own node, the node may not be bound to an EIP. For example, in the preceding figure, node 192.168.x.x is not bound to an EIP. Solution: Bind an EIP to the node. For details, see **Assigning an EIP and Binding It to an ECS** .
- The authorization information has expired, 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 Private Maven Repository

There are two solutions:

- Create the **settings.xml** file in the root directory of your project, and specify the private Maven repository address in it. 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 shows example paths of the **settings.xml** and **pom.xml** files:



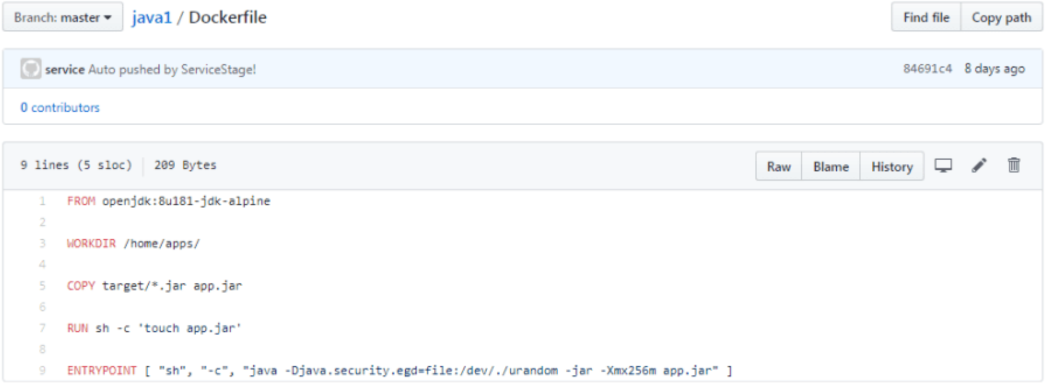
Auto created by ServiceStage! https://github.com[redacted]/tomcat

[redacted] Create settings.xml ... Latest commit 67d1144 1 minute ago

src	Auto pushed by ServiceStage!	4 months ago
Dockerfile	Auto pushed by ServiceStage!	4 months ago
LICENSE	Auto pushed by ServiceStage!	4 months ago
README.md	Auto pushed by ServiceStage!	4 months ago
pom.xml	Auto pushed by ServiceStage!	4 months ago
settings.xml	Create settings.xml	1 minute ago

Invalid Dockerfile for Source Code Job Building

For details about how to write a Dockerfile, see the **official website** or **ServiceStage template demo**.



```
FROM openjdk:8u181-jdk-alpine

WORKDIR /home/apps/

COPY target/*.jar app.jar

RUN sh -c 'touch app.jar'

ENTRYPOINT [ "sh", "-c", "java -Djava.security.egd=file:/dev/./urandom -jar -Xmx256m app.jar" ]
```

Project Code Depends On CSE SDK and CodeArts Private Maven Repository

Perform the following steps:

Step 1 Create the **settings.xml** file in the root directory of your project.

Step 2 Log in to the CodeArts private dependency repository and select the 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.hauweicloud.com/repository/maven</url>
</mirror>
```

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

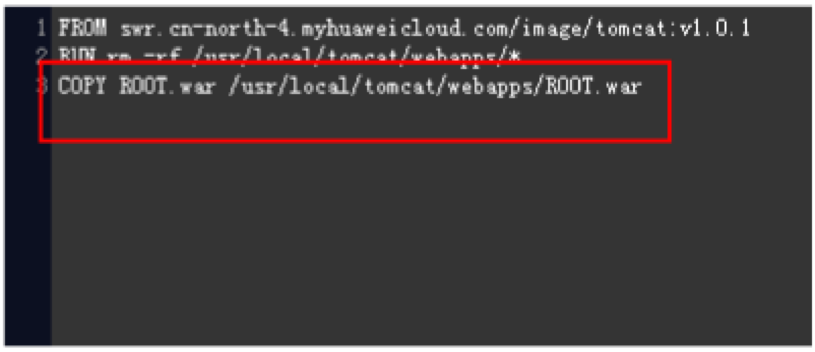
```
<repository>
     <id>HuaweiCloudSDK</id>
     <url>https://repo.huaweicloud.com/repository/maven/huaweicloudsdk/</url>
     <releases>
     <enabled>true</enabled>
     </releases>
     <snapshots>
     <enabled>true</enabled>
     </snapshots>
</repository>
```

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

----End

Customizing Dockerfile When Creating Software Package Building Job

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



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

4.5 How Can I Access Dependent Services Through VPC Endpoints When Building Images?

Background

VPC Endpoint is a cloud service that provides secure and private channels to connect your VPCs to VPC endpoint services. It allows you to plan networks flexibly without having to use EIPs.

When building images for components in the Kubernetes environment, you can create VPC endpoints to communicate with Object Storage Service (OBS) and SoftWare Repository for Container (SWR), and use API Gateway (APIG) to call functional APIs of ServiceStage based on configured private domain names.

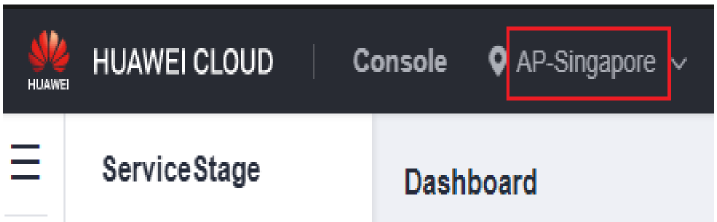
NOTE

VPC endpoints can be used to access dependent services only in CN East 2, CN South-Guangzhou, AP-Singapore, AF-Johannesburg, and CN East-Shanghai1, CN North-Beijing4, and LA-Mexico City2.

Procedure

Step 1 Log in to ServiceStage.

Step 2 Select the region where your service is located, for example, **AP-Singapore**.



Step 3 In the address box of the browser, obtain the value of the **region** field.

The following information in bold is an example of **region**:

```
https://console-intl.huaweicloud.com/servicestage/?agencyId=d6****************************41&region=ap-southeast-3&locale=zh-cn#/overview
```

Step 4 Create VPC endpoints for SWR and APIG. For details, see **Buying a VPC Endpoint for Accessing Interface VPC Endpoint Services**.

1. **Region**: Select the region specified in **Step 2**.
2. **Service Category**: Select **Find a service by name**.
3. **VPC Endpoint Service Name**: Enter the VPC endpoint service name for each cloud service by referring to **Table 4-1**.

   NOTE

   Replace *${region}* in the following table with the value obtained in **Step 3**.

   Table 4-1 Accessing interface VPC endpoint services

Cloud Service	VPC Endpoint Service Name
SWR	**com.myhuaweicloud.*${region}*.swr** **NOTE** If you select the **CN South-Guangzhou** region, the VPC endpoint service name of SWR is **swr.cn-south-1.myhuaweicloud.com**.
APIG	**com.myhuaweicloud.*${region}*.api**

4. Select **Create a Private Domain Name**.
5. **VPC**: Select the same VPC for each VPC endpoint service listed in **Table 4-1**.
6. **Subnet**: Select a subnet for each VPC endpoint service listed in **Table 4-1**.
7. Set other parameters based on site requirements.

Step 5 Create VPC endpoints for OBS. For details, see **Buying a VPC Endpoint for Accessing Gateway VPC Endpoint Services**.

You need to create VPC endpoints for the VPC endpoint services of OBS listed in **Table 4-2**.

1. **Region**: Select the region specified in **Step 2**.
2. **Service Category**: Select **Find a service by name**.
3. **VPC Endpoint Service Name**: Enter the VPC endpoint service name for OBS by referring to **Table 4-2**.

   Table 4-2 Accessing gateway VPC endpoint services

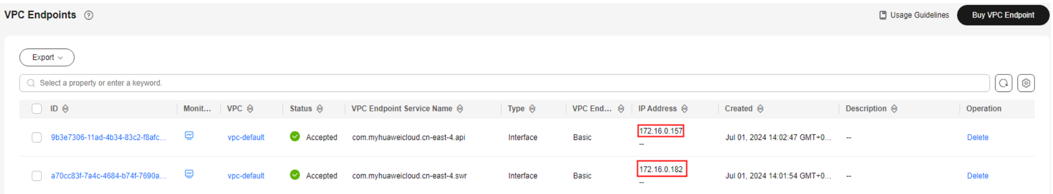
Region	VPC Endpoint Service Name
CN South-Guangzhou	cn-south-1.com.myhuaweicloud.v4.obsv2
	cn-south-1.com.myhuaweicloud.v4.obsv2.lz05
	cn-south-1.com.myhuaweicloud.v4.obsv2.lz08
	cn-south-1.com.myhuaweicloud.v4.obsv2.lz09
CN East2	com.myhuaweicloud.cn-east-4.obslb01.v4.obsv2.vxlan
	com.myhuaweicloud.cn-east-4.obslb01.v6.obsv2.vxlan
AP-Singapore	ap-southeast-3.com.myhuaweicloud.v4.obsv2

Region	VPC Endpoint Service Name
	ap-southeast-3.com.myhuaweicloud.v6.obsv2
AF-Johannesburg	af-south-1.myhuaweicloud.v4.obsv2
	af-south-1.myhuaweicloud.v6.obsv2
CN East-Shanghai1	cn-east-3.com.myhuaweicloud.v4.global.obsv2
	cn-east-3.v4obsv2_new.58c9f146-63f0-4f07-98d3-18fe4874086b
	cn-east-3.myhuaweicloud.v4.obsv2.lz03
	cn-east-3.myhuaweicloud.v4.obsv2.lz04
	cn-east-3.com.myhuaweicloud.v4.obsv2.lz09
	cn-east-3.com.myhuaweicloud.v4.obsv2.lz11
CN North-Beijing4	cn-north-4.com.myhuaweicloud.v4.obsv2.OBSCluster9
	cn-north-4.com.myhuaweicloud.v4.byte.obsv2
	cn-north-4.com.myhuaweicloud.v4.CBG.obsv2
	cn-north-4.com.myhuaweicloud.v4.obsv2.lz11
	cn-north-4.com.myhuaweicloud.v4.storage.lz13
	cn-north-4.com.myhuaweicloud.v4.obsv2.lz25
LA-Mexico City2	la-north-2.com.myhuaweicloud.v4.obsv2

4. **VPC**: Select the VPC specified in **Step 4** for all VPC endpoint services listed in **Table 4-2**.
5. **Subnet**: Select a subnet for each VPC endpoint service listed in **Table 4-2**.
6. Set other parameters based on site requirements.

Step 6 In the endpoint list, obtain the service addresses of the VPC endpoint created for APIG and SWR in **Step 4**.

Replace *${region}* with the value obtained in **Step 3**.



ID	Monit...	VPC	Status	VPC Endpoint Service Name	Type	VPC End...	IP Address	Created	Description	Operation
9b3e7306-11ad-4b34-83c2-f8afc...		vpc-default	Accepted	com.myhuaweicloud.cn-east-4.api	Interface	Basic	172.16.0.157 --	Jul 01, 2024 14:02:47 GMT+0...	--	Delete
a70cc83f-7a4c-4684-b74f-7690a...		vpc-default	Accepted	com.myhuaweicloud.cn-east-4.swr	Interface	Basic	172.16.0.182 --	Jul 01, 2024 14:01:54 GMT+0...	--	Delete

Step 7 Create private domain names. For details, see **Creating a Private Zone**.

1. **Domain Name**: Enter the following private domain names:

NOTE

Replace *${region}* with the value obtained in **Step 3**.

- **servicestage.*${region}*.myhuaweicloud.com**
- **swr-api.*${region}*.myhuaweicloud.com**
- **swr.*${region}*.myhuaweicloud.com**

2. **VPC**: Select the VPC specified in **Step 4**.
3. Set other parameters based on site requirements.

Step 8 Add record sets for all private domain names created in **Step 7**.

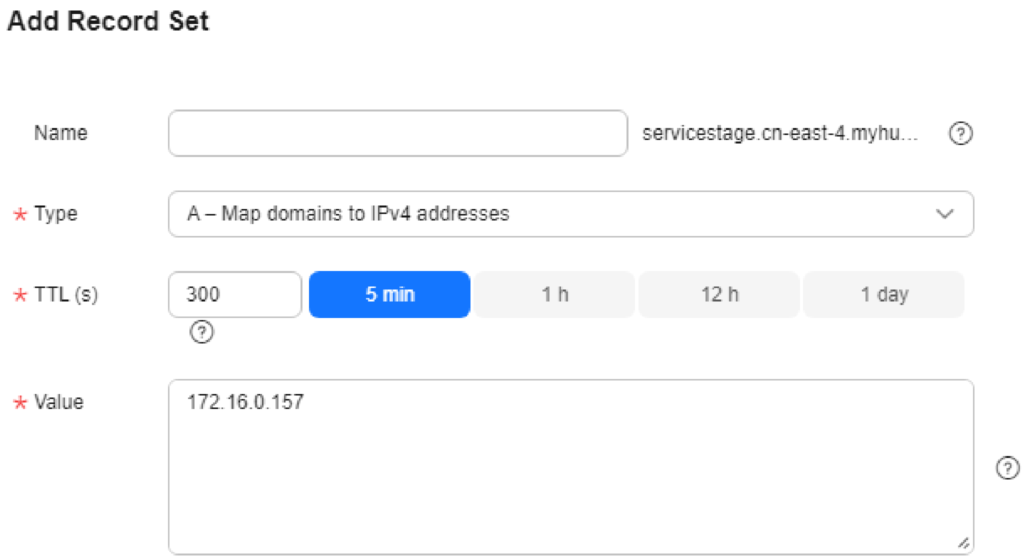
1. **Type**: Select **A – Map domains to IPv4 addresses**.
2. **Value**: Set this parameter by referring to the following table.

NOTE

Replace *${region}* in the following table with the value obtained in **Step 3**.

Private Domain Name	Record Value
servicestage.*${region}*.myhuaweicloud.com	Enter the address of the **com.myhuaweicloud.*${region}*.api** VPC endpoint service obtained in **Step 6**.
swr-api.*${region}*.myhuaweicloud.com	
swr.*${region}*.myhuaweicloud.com	Enter the address of the **com.myhuaweicloud.*${region}*.swr** VPC endpoint service obtained in **Step 6**.

3. Set other parameters based on site requirements.


Add Record Set

Name: servicestage.cn-east-4.myhu...

* Type: A – Map domains to IPv4 addresses

* TTL (s): 300

* Value: 172.16.0.157

----End

5 Software Center

5.1 What If a Software Package Fails to Be Uploaded?

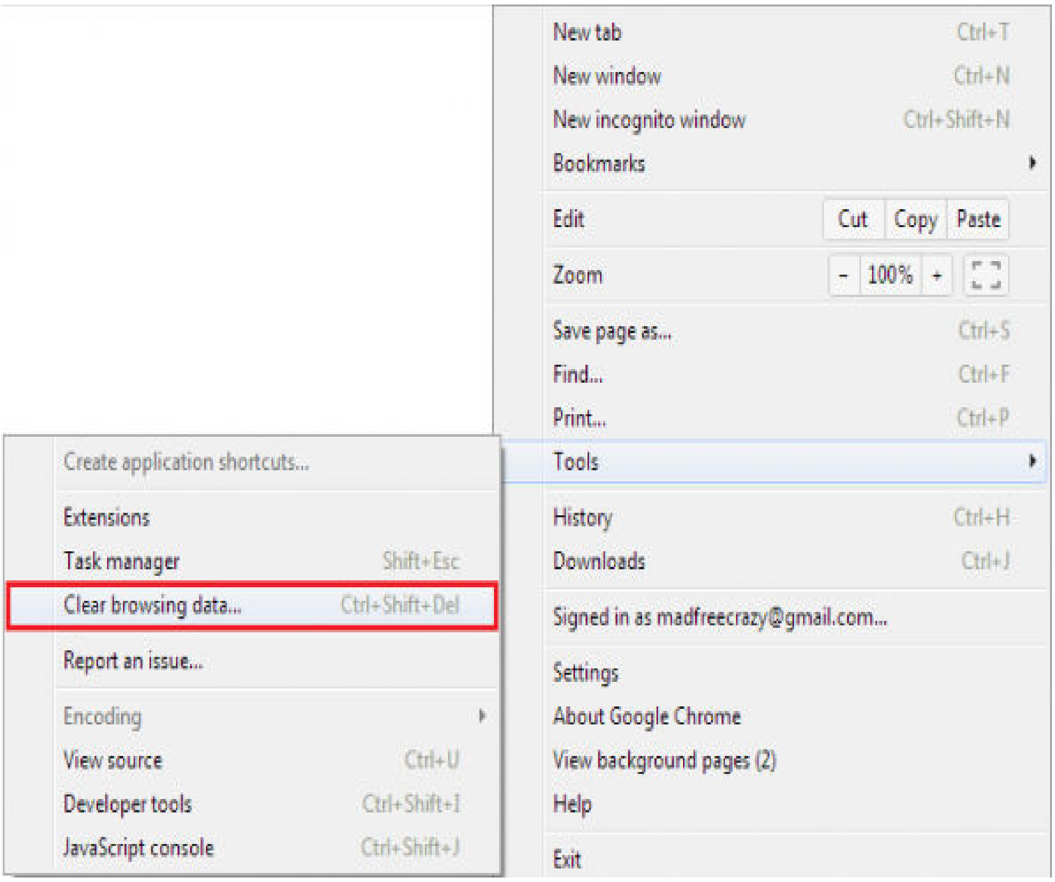
Symptom

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 How Can I Access Dependent Services Through VPC Endpoints When Installing VM Agents?

Background

VPC Endpoint is a cloud service that provides secure and private channels to connect your VPCs to VPC endpoint services. It allows you to plan networks flexibly without having to use EIPs.

When installing VM agents in the VM environment, you can create VPC endpoints to communicate with Log Tank Service (LTS), Application Operations Management (AOM), Object Storage Service (OBS), and SoftWare Repository for Container (SWR), and use API Gateway (APIG) to call functional APIs of ServiceStage, ECS, VPC, and AOM based on configured private domain names.

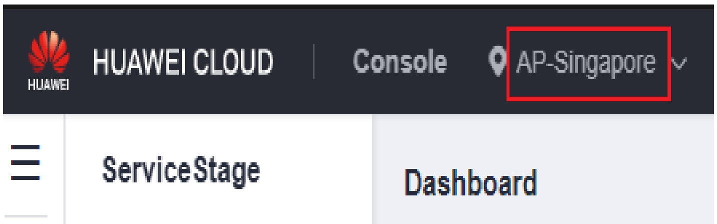
NOTE

VPC endpoints can be used to access dependent services only in CN East 2, CN South-Guangzhou, AP-Singapore, AF-Johannesburg, and CN East-Shanghai1, CN North-Beijing4, and LA-Mexico City2.

Procedure

Step 1 Log in to ServiceStage.

Step 2 Select the region where your service is located, for example, **AP-Singapore**.



Step 3 In the address box of the browser, obtain the value of the **region** field.

The following information in bold is an example of **region**:

```
https://console-intl.huaweicloud.com/servicestage/?agencyId=d6**************************41&region=ap-southeast-3&locale=zh-cn#/overview
```

Step 4 Create VPC endpoints for the services listed in **Table 6-1**. For details, see **Buying a VPC Endpoint for Accessing Interface VPC Endpoint Services**.

1. **Region**: Select the region specified in **Step 2**.
2. **Service Category**: Select **Find a service by name**.
3. **VPC Endpoint Service Name**: Enter the VPC endpoint service name for each cloud service by referring to **Table 6-1** and then click **Verify**.

   NOTE

   Replace *${region}* in the following table with the value obtained in **Step 3**.

Table 6-1 Accessing interface VPC endpoint services

Cloud Service	VPC Endpoint Service Name
LTS	**com.myhuaweicloud.*${region}*.lts-access**
AOM	**com.myhuaweicloud.*${region}*.aom-access**
SWR	**com.myhuaweicloud.*${region}*.swr** **NOTE** If you select the **CN South-Guangzhou** region, the VPC endpoint service name of SWR is **swr.cn-south-1.myhuaweicloud.com**.
APIG	**com.myhuaweicloud.*${region}*.api**

4. Select **Create a Private Domain Name**.
5. **VPC**: Select the same VPC for each VPC endpoint service listed in **Table 6-1**.
6. **Subnet**: Select a subnet for each VPC endpoint service listed in **Table 6-1**.
7. Set other parameters based on site requirements.

Step 5 Create VPC endpoints for OBS. For details, see **Buying a VPC Endpoint for Accessing Gateway VPC Endpoint Services**.

You need to create VPC endpoints for the VPC endpoint services of OBS listed in **Table 6-2**.

1. **Region**: Select the region specified in **Step 2**.
2. **Service Category**: Select **Find a service by name**.
3. **VPC Endpoint Service Name**: Enter the VPC endpoint service name for OBS in different regions by referring to **Table 6-2**.

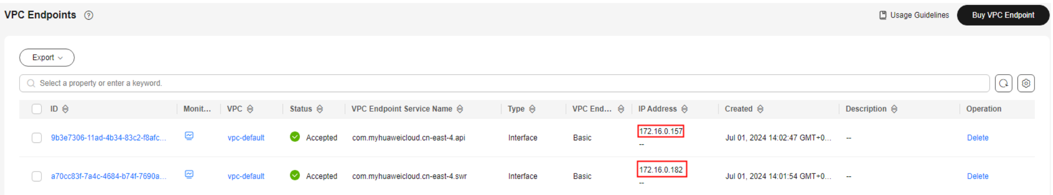
Table 6-2 Accessing gateway VPC endpoint services

Region	VPC Endpoint Service Name
CN South-Guangzhou	cn-south-1.com.myhuaweicloud.v4.obsv2
	cn-south-1.com.myhuaweicloud.v4.obsv2.lz05
	cn-south-1.com.myhuaweicloud.v4.obsv2.lz08
	cn-south-1.com.myhuaweicloud.v4.obsv2.lz09
CN East2	com.myhuaweicloud.cn-east-4.obslb01.v4.obsv2.vxlan
	com.myhuaweicloud.cn-east-4.obslb01.v6.obsv2.vxlan
AP-Singapore	ap-southeast-3.com.myhuaweicloud.v4.obsv2
	ap-southeast-3.com.myhuaweicloud.v6.obsv2
AF-Johannesburg	af-south-1.myhuaweicloud.v4.obsv2
	af-south-1.myhuaweicloud.v6.obsv2
CN East-Shanghai1	cn-east-3.com.myhuaweicloud.v4.global.obsv2
	cn-east-3.v4obsv2_new.58c9f146-63f0-4f07-98d3-18fe4874086b
	cn-east-3.myhuaweicloud.v4.obsv2.lz03
	cn-east-3.myhuaweicloud.v4.obsv2.lz04
	cn-east-3.com.myhuaweicloud.v4.obsv2.lz09
	cn-east-3.com.myhuaweicloud.v4.obsv2.lz11
CN North-Beijing4	cn-north-4.com.myhuaweicloud.v4.obsv2.OBSCluster9
	cn-north-4.com.myhuaweicloud.v4.byte.obsv2
	cn-north-4.com.myhuaweicloud.v4.CBG.obsv2
	cn-north-4.com.myhuaweicloud.v4.obsv2.lz11
	cn-north-4.com.myhuaweicloud.v4.storage.lz13
	cn-north-4.com.myhuaweicloud.v4.obsv2.lz25
LA-Mexico City2	la-north-2.com.myhuaweicloud.v4.obsv2

4. **VPC**: Select the VPC specified in **Step 4** for all VPC endpoint services listed in **Table 6-2**.
5. **Subnet**: Select a subnet for each VPC endpoint service listed in **Table 6-2**.
6. Set other parameters based on site requirements.

Step 6 In the endpoint list, obtain the service addresses of the VPC endpoint created for APIG and SWR in **Step 4**.

Replace *${region}* with the value obtained in **Step 3**.



ID	Monit...	VPC	Status	VPC Endpoint Service Name	Type	VPC End...	IP Address	Created	Description	Operation
9b3e7306-11ad-4b34-83c2-f8afc...		vpc-default	Accepted	com.myhuaweicloud.cn-east-4.api	Interface	Basic	172.16.0.157 --	Jul 01, 2024 14:02:47 GMT+0...	--	Delete
a70cc83f-7a4c-4684-b74f-7690a...		vpc-default	Accepted	com.myhuaweicloud.cn-east-4.swr	Interface	Basic	172.16.0.182 --	Jul 01, 2024 14:01:54 GMT+0...	--	Delete

Step 7 Create private domain names. For details, see **Creating a Private Zone**.

1. **Domain Name**: Enter the following private domain names:

   NOTE

   Replace *${region}* with the value obtained in **Step 3**.

   - **servicestage.*${region}*.myhuaweicloud.com**
   - **ecs.*${region}*.myhuaweicloud.com**
   - **vpc.*${region}*.myhuaweicloud.com**
   - **aom.*${region}*.myhuaweicloud.com**
   - **swr-api.*${region}*.myhuaweicloud.com**
   - **swr.*${region}*.myhuaweicloud.com**
2. **VPC**: Select the VPC specified in **Step 4**.
3. Set other parameters based on site requirements.

Step 8 Add record sets for all private domain names created in **Step 7**.

1. **Type**: Select **A – Map domains to IPv4 addresses**.
2. **Value**: Set this parameter by referring to the following table.

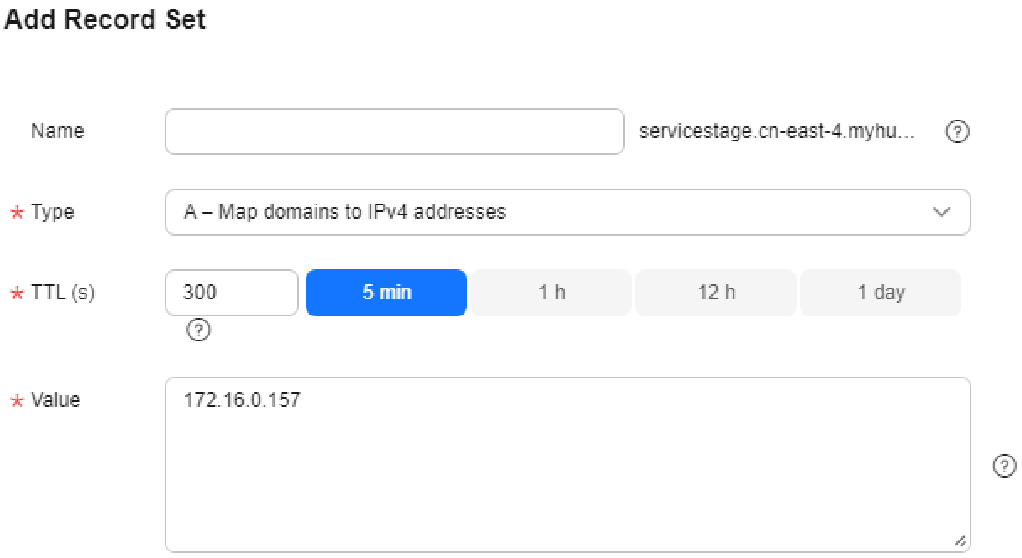
   NOTE

   Replace *${region}* in the following table with the value obtained in **Step 3**.

Private Domain Name	Record Value
servicestage.*${region}*.myhuaweicloud.com	Enter the address of the **com.myhuaweicloud.*${region}*.api** VPC endpoint service obtained in **Step 6**.
ecs.*${region}*.myhuaweicloud.com	
vpc.*${region}*.myhuaweicloud.com	
aom.*${region}*.myhuaweicloud.com	
swr-api.*${region}*.myhuaweicloud.com	

Private Domain Name	Record Value
swr.*${region}*.myhuaweicloud.com	Enter the address of the **com.myhuaweicloud.*${region}*.swr** VPC endpoint service obtained in **Step 6**.

3. Set other parameters based on site requirements.


Add Record Set

Name: servicestage.cn-east-4.myhu...

* Type: A – Map domains to IPv4 addresses

* TTL (s): 300

* Value: 172.16.0.157

----End

6.3 What Do 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 with the offline VM agent. For details, see **Logging In to an ECS**.

Step 2 Run the following commands to find out the **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**. 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 with the offline VM agent.
3. In the **Management Information** area under the **Basic Information** tab, 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 from **Step 3.3**.
   a. On the **Basic Information** tab, check whether **Cloud Service** is ECS.
   b. On the **Permissions** tab, 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 with the offline VM agent. For details, see **Logging In to an ECS**.
7. Run the following commands to restart the agent, where **x.x.x** is your actual **servicestage-agent** version.

   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**:

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 with the offline VM agent. 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, where **x.x.x** is your actual **servicestage-agent** version.

   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

6.4 What Do I Do If the VM Agent Is Offline?

Symptom

The agent has been installed but is offline and does not work.

Solution

Step 1 Log in to the ECS with the offline VM agent. For details, see **Logging In to an ECS**.

Step 2 Run the following commands to find out the **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**. 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 with the offline VM agent.
3. In the **Management Information** area under the **Basic Information** tab, 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 from **Step 3.3**.
   a. On the **Basic Information** tab, check whether **Cloud Service** is ECS.
   b. On the **Permissions** tab, 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 with the offline VM agent. For details, see **Logging In to an ECS**.
7. Run the following commands to restart the agent, where **x.x.x** is your actual **servicestage-agent** version.

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

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

Step 4 If **Authorization Model** is **AKSK**:

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 with the offline VM agent. 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, where **x.x.x** is your actual **servicestage-agent** version.

   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

6.5 What Should I Do If the VM Agent Is Subhealthy?

Symptom

On the environment details page, the agent status of the managed VM is subhealthy.

Possible Cause

VM network was abnormal.

Solution

Step 1 Log in to ServiceStage.

Step 2 Choose **Environment Management**. The **Environment Management** page is displayed.

Step 3 Click the environment where the faulty VM is located. The environment details page is displayed.

Step 4 In the **All Resources** area, choose **Elastic Cloud Server** or **Custom servers** from **Compute**.

Step 5 In the list of managed resources on the right, obtain the name of the VM whose agent status is **Subhealthy**.

Step 6 Log in to this VM.

Step 7 Run the following command to open the VM agent-critical log:

```
vi /var/log/servicestage-agent/servicestage-agent-critical.log
```

Step 8 Search for alarm time as reported in the logs and locate the error information for that time.

Step 9 Rectify the fault using the error information in the logs. If the fault persists, contact technical support.

----End

6.6 What Do I Do If Service Registration Fails After Enabling IPv6 for an Exclusive Microservice Engine?

Symptom

A microservice developed based on Java chassis was registered with an exclusive microservice engine with security authentication enabled. The microservice registry center address was the IPv4 address of the engine registry center. The microservice was successfully registered and started.

However, changing the registry center address to the IPv6 address of the engine registry center caused the registration to fail and returned the error "java.net.SocketException: Protocol family unavailable".

Possible Cause

Selecting a VPC network with IPv6 enabled when creating an exclusive microservice engine gives the engine IPv6 support. The IPv6 dual-stack function must be enabled for the selected CCE cluster to deploy services using containers through an IPv6 network segment.

Thus, IPv6 must be enabled for the selected cluster to avoid network disconnection and error.

Solution

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

For details, see **Editing an Environment**.

Step 2 Redeploy the application. For details, see **Creating and Deploying a Component**.

----End

6.7 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.

6.8 How Do I Switch a Component Building Cluster from Shared to Private?

Symptom

The ServiceStage shared cluster is shared by all users and 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

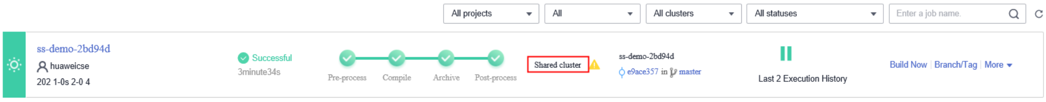
1. A cluster has been created. For details, see **Buying a Cluster**.
2. An EIP has been bound to the CCE cluster node. For details, see **Binding an EIP to an Instance**.

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**.



All projects | All | All clusters | All statuses | Enter a job name.

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

6.9 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 create new instances in CSE, you need to grant permissions to CSE. For details, see **Creating a User and Granting Permissions**.
- Since it depends on VPC, CSE needs permissions. Create a cloud service agency **cse_admin_trust** by referring to **Creating an Agency**"Creating an Agency" in *Identity and Access Management User Guide*.

6.10 What Do I Need to Know Before Upgrading an Exclusive ServiceComb 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)
  ```

  or

  ```
  [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 the 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**: Importing a configuration file in the original 2.x configuration center format failed and the error reported an empty file or invalid file format.

  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 ServiceComb 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
  ```

6.11 Obtain Configurations Failed

Symptom

After a microservice is connected to the corresponding microservice development framework (such as spring-cloud-huawei and java-chassis), it fails to obtain configuration items from the ServiceComb engine by calling the configuration query API through SDK.

Possible Cause

If the connection between a microservice and the registration center jitters due to network and CPU problems, the request may be abnormal.

Solution

The microservice framework has the self-healing capability. If the configuration fails to be obtained, the retry mechanism takes effect for pulling configurations. Service exceptions do not occur. Check whether configuration items can be obtained next time. If not, contact customer service.

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 a Log Policy of an Application**.

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 Do I Do with Excessive Memory Usage from Starting a New Service on a Node?

Symptom

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

Solution

Set affinity by referring to **Configuring a Scheduling Policy of a Component Instance** 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 How Are Microservice and Common Applications Different?

Microservice architecture breaks an application into multiple parts. Therefore, microservice applications are actually distributed applications.

Such applications are more responsive and reliable.

Type	Microservice	Common Application
Development	Light workloads. A two-pizza team can rewrite all code in two weeks. 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. For details about microservice development, see **Developing Microservice Applications**.	Heavy workloads. Complex logic, coupled modules, bloated code, difficult modification, and low version iteration efficiency.
Deployment	Multiple-microservice complexity requires orchestration. For details about microservice application deployment, see **Creating and Deploying a Component**.	Large size requires long build and deploy times. This is not conducive to frequent deployment and hinders continuous delivery. This affects mobile apps the most.

Type	Microservice	Common Application
O&M	Focuses on governance (in addition to metrics monitoring and log collection). The core concept is about maintaining system performance through modifications while the system is running. For details about application O&M, see **Component O&M**.	Focuses on full upgrading to rectify online faults. This concept takes a long time.

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 the AK/SK in the **microservice.yaml** file. To obtain correct AK/SK, see **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 the AK/SK in the **microservice.yaml** file. To obtain correct AK/SK, see **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 the project information in the **microservice.yaml** file. 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 for the instance's region.

- The following error message indicates that the microservice failed to access the service center or configuration center. The microservice did not access the service center or configuration center and did not leave an error code there.

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

  Checking method:

  a. Check the IP addresses of the service center and configuration center in the **microservice.yaml** file.

  ```
  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 either 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
   ```

- The final error is an occupied microservice port number. Check this by running the following command.

  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

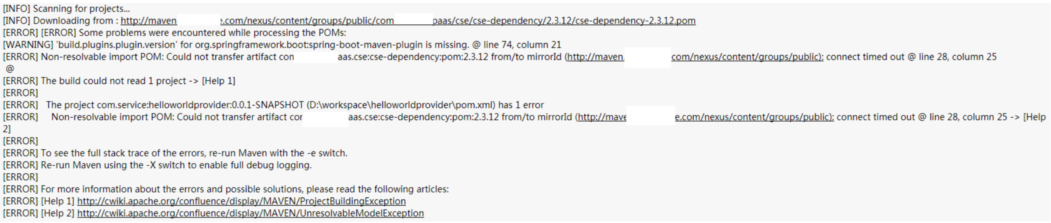
- Using distributed transactions
- Using Java to compile microservices.

- 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[redacted].com/nexus/content/groups/public/com[redacted]paas/cse/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 con[redacted]aas.cse:cse-dependency:pom:2.3.12 from/to mirrorId (http://maven.[redacted]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:helloworldprovider:0.0.1-SNAPSHOT (D:\workspace\helloworldprovider\pom.xml) has 1 error
[ERROR]     Non-resolvable import POM: Could not transfer artifact cor[redacted]aas.cse:cse-dependency:pom:2.3.12 from/to mirrorId (http://mave[redacted]e.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://cwiki.apache.org/confluence/display/MAVEN/ProjectBuildingException
[ERROR] [Help 2] http://cwiki.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. For example, modules in a monolithic application are called using code and any API incompatibility can be resolved economically during early compilation. 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 Mismatched?

Symptom

Development of a microservice application returned the following error message:

```
"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;IZZJ)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 versions.

For ServiceComb 2.0.1 development framework, the matching 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/**.

8.9 How Do I Package a Java or Tomcat Application?

ServiceStage allows you to compress a Java or Tomcat application into a .zip or .tar.gz package for deploying a component on a VM.

Application Package Directory Structure

- Example Java application package: **https://github.com/servicestage-demo/example/tree/master/servicestage-vm-demo/demoJavaExample.zip**
- Example Tomcat application package: **https://github.com/servicestage-demo/example/tree/master/servicestage-vm-demo/demoTomcatExample.zip**

The following uses the Java application package **demoJavaExample.zip** to illustrate the application package directory structure:

```
demoJavaExample/
│
├── scripts/
│   │      ├── pre-install.sh
│   │      ├── pre-stop.sh
│   │      ├── stop.sh
│   │      ├── uninstall.sh
│   │      ├── install.sh
│   │      ├── start.sh
│   │      ├── check.sh
│   │      ├── post-start.sh
├── packages/
│   │      ├── weather-1.0.0.jar
├── config/
│   │      ├── system.cfg
├── appspec.yml
```

The prefix of the package name must be the same as the directory name of the decompressed file. For example, a package named **demoJavaExample.zip** will have a directory after decompression named **demoJavaExample**.

The directories and files in the application package are described as follows:

- **scripts**: (mandatory) stores script files executed in each application lifecycle.
- **packages**: (mandatory) stores application JAR or WAR packages.

- **config**: (mandatory) stores application configuration information. For Java applications, the **system.cfg** file is stored. For Tomcat applications, the **system.cfg**, **logging.properties**, and **server.xml** files are stored.
- **appspec.yml**: (mandatory) records the lifecycle definition and also specifies information such as health check.

appspec.yml File Description

As shown in the following, the **appspec.yml** file defines the entire deployment process, and the environment variables and health check information used during deployment.

```
spec:
  # Custom user for application running.
  deps:
    - name: "@os/linux/user@1.0"
      user: www
      group: www
      home: /home/www

  # Directly import the environment variables of the application lifecycle script.
  env:
  - name: APP_ENV
    value: "{{app.env}}"

  # Import the environment variables of the application lifecycle script stored in /opt/application/${appName}/${appVersion}/${instanceId}/servicestage-vmapp/application.conf as a file.
  value:
  - name: APP_VALUE
    value: "{{app.value}}"

  # Application health check.
  probes:
    # Interface health check.
    # health:
    #   exec:
    #     method: GET
    #     request: http://127.0.0.1:8080/healthcheck
    #     timeout: 5

    # Script command health check.
    liveness:
      exec:
        command:
          - ps -ef | grep ${APP_HOME}/apache-tomcat-*/bin/bootstrap.jar | grep -v grep   # Currently, only APP_HOME can be used to obtain environment variables.
        timeout: 300
        runas: www
    retries: 10
    interval: 6

  # Application lifecycle script.
  lifecycle:
    pre-install:
      - command: scripts/pre-install.sh
        timeout: 300
    install:
      - command: scripts/install.sh
        timeout: 300
    check:
      - command: scripts/check.sh
        timeout: 300
        runas: www
    start:
      - command: scripts/start.sh
        timeout: 300
        runas: www
    post-start:
      - command: scripts/post-start.sh
        timeout: 300
        runas: www
    pre-stop:
      - command: scripts/pre-stop.sh
        timeout: 300
        runas: www
    stop:
      - command: scripts/stop.sh
        timeout: 300
        runas: www
    uninstall:
      - command: scripts/uninstall.sh
        timeout: 300
```

- **deps**: defines the custom user for application running. Set this field by referring to the following example.

  ```
  deps:
      - name: "@os/linux/user@1.0"
        user: www
        group: www
        home: /home/www
  ```

- **env**: records the environment variables during the running of each script. The environment variables can be directly imported during script running.
- **value**: saved in the **${APP_HOME}/servicestage-vmapp/application.conf** file. To use these values in the script, run the following command at the beginning of the script:

  ```
  #!/bin/bash
    . ${APP_HOME}/servicestage-vmapp/application.conf
  # dosomething
  ```

- **probes**: health check mode (**health** or **liveness**). If both modes are used, the component is considered alive only if both modes return normal status. If only one mode is used, the component is considered alive only if this mode returns normal status. If neither mode is used, the component is considered alive by default in all scenarios except when stopping or deleting the component.
- **health**: accesses the health check interface to check the component status. The component is alive when its status code ranges from 200 to 400 (excluding 400).
  - **method**: method used to access the interface (HTTP or HTTPS)
  - **request**: access address.
  - **body**: request body.
  - **timeout**: timeout interval, in seconds.
- **liveness**: runs a command to check the component status. The component is alive when there is no error message and the command output is not empty.
  - **command**: command used to check the component status
  - **runas**: Linux user who runs **command**
  - **timeout**: command execution timeout interval, in seconds
- **retries**: number of times the health check is attempted (default is 10). An application fails startup after this number of retries, counting from when it starts up.
- **interval**: how much time passes before a second health check is attempted (default is 6s). The amount of time increases for each failed attempt, so the time passed is the **Interval** multiplied by the number of previous retries.

- **lifecycle**: script or command executed in each lifecycle.
  - **command**: script file to be executed. The value must be a file path relative to the **${APP_HOME}** directory.
  - **timeout**: timeout interval, in seconds.
  - **runas**: user who performs the operation

Lifecycle execution sequence:

- Deployment sequence: ServiceStage installation technology stack + pre-install.sh -> install.sh -> start.sh -> check.sh -> post-start.sh
- Upgrade/Rollback sequence: ServiceStage installation technology stack + pre-install.sh -> install.sh -> pre-stop.sh -> stop.sh -> uninstall.sh + ServiceStage uninstallation technology stack -> start.sh -> check.sh ->post-start.sh
- Deletion sequence: pre-stop.sh -> stop.sh -> uninstall.sh + ServiceStage

config Description

- **system.cfg** file
  a. Set the **config** parameter.

     When **creating a component based on a VM**, you can set some configuration items by referring to **Adding a Configuration Item**. ServiceStage stores the configured configuration items in the **${APP_HOME}/config/system.cfg** file as key-value pairs. You can also preset some configuration parameters in the **config** file. Example:

     ```
     # Format of the content stored in the system.cfg file
     key1=value1
     key2=value2
     ```

  b. Reference the **config** parameter.

     Example:

     ```
     #!/bin/bash
      . ${APP_HOME}/config/system.cfg
      . ${APP_HOME}/config/user_config.cfg
      echo ${key1}
     ```

- **server.xml** file

  See **server.xml Description of Tomcat**.
- **logging.properties** file

  See **logging.properties Description of Tomcat**.

packages Directory Description

This folder holds the JAR and WAR packages to be executed when running commands in the script.

scripts Directory Description

This folder stores the scripts executed in each lifecycle of an application.

Default System Configurations

The default system configurations are defined in the *${APP_HOME}*/**servicestage-vmapp/application.conf** file and are not directly displayed in environment variables. The environment variables defined in the **application.conf** file can be referenced only after the following command is added to the script:

```
. ${APP_HOME}/servicestage-vmapp/application.conf
```

Script environment variables that can be directly referenced: **APP_HOME**, those specified in **appspec.yml**, and those specified in **Adding a Component Environment Variable**.

Example file content:

```
export LOG_PATH=/var/log/application/zqb-4-vm-wqd-2-7f6fbc/3c719644-f9f5-46b4-a06a-61fcf163e5b5
export APP_HOME=/opt/application/zqb-4-vm-wqd-2-7f6fbc/2023.1207.11314/3c719644-f9f5-46b4-
a06a-61fcf163e5b5
export TOMCAT_STACK_HOME=/opt/application/zqb-4-vm-wqd-2-7f6fbc/2023.1207.11314/3c719644-
f9f5-46b4-a06a-61fcf163e5b5/apache-tomcat-8.5.82
export JRE_STACK_HOME=/opt/application/zqb-4-vm-wqd-2-7f6fbc/2023.1207.11314/3c719644-f9f5-46b4-
a06a-61fcf163e5b5/jre1.8
export APP_VALUE="{{app.value}}"
export APP_USER=www
export APP_GROUP=www
```

- **LOG_PATH**: log recording path. Application logs are recorded in the log file in **LOG_PATH** and named **{lifecycle}_app.log**.
- **APP_HOME**: records the running environment of the current application.
- **TOMCAT_STACK_HOME**: home directory of Tomcat.
- **JRE_STACK_HOME**: home directory of JRE.
- **APP_VALUE**: environment variable specified in **appspec.yml**.
- **APP_USER**: the file's user. You are advised to set this parameter to **www**. Alternatively, you can also custom the user.
- **APP_GROUP**: the file's owner group. You are advised to set this parameter to **www**. Alternatively, you can also custom the user group.

If you customize the user and user group, run the following commands to grant the custom user the permissions to run the script:

```
sudo -- /bin/bash -c "chmod -R 755 ${APP_HOME}/scripts/* ${LOG_PATH};chown -R ${APP_USER}:$
{APP_GROUP} ${APP_HOME}/scripts/start.sh;chown -R ${APP_USER}:${APP_GROUP} ${APP_HOME}/scripts/
stop.sh;chown -R ${APP_USER}:${APP_GROUP} ${LOG_PATH} ${JRE_STACK_HOME}/"
```

Script Writing Description

The directory structure for executing scripts on ServiceStage is:

```
APP_HOME/
│
├── scripts/
│   │      ├── pre-install.sh
│   │      ├── pre-stop.sh
│   │      ├── stop.sh
│   │      ├── uninstall.sh
│   │      ├── install.sh
│   │      ├── start.sh
│   │      ├── check.sh
│   │      ├── post-start.sh
│   ├──packages/
│   │      ├── my-app.jar/my-app.war
├── config/
│   │      ├── system.cfg
├── servicestage-vmapp/
│   │      ├── application.conf
├── jre1.8
├── apache-tomcat-8.5.82
```

- When writing scripts, go to the **${APP_HOME}** directory to obtain the locations of other files.
- To obtain the location of JRE or Tomcat, run the following commands at the beginning of the script. Use **${APP_HOME}/servicestage-vmapp/application.conf** to load environment variables and then use **${JRE_STACK_HOME}** or **${TOMCAT_STACK_HOME}** to obtain the values.
- ServiceStage reports logs in the **${LOG_PATH}** directory. Therefore, record them in the log file in the **${LOG_PATH}** directory and name them **{lifecycle}_app.log** to facilitate subsequent log reporting.
- It is recommended that the script start with the following lines to obtain environment variables and user configurations, and specify the technology stack home directory, log output path, and current execution address.

```
#!/bin/bash

# Obtain user configurations and specify environment variables.
. ${APP_HOME}/config/system.cfg
. ${APP_HOME}/servicestage-vmapp/application.conf

# Specify the technology stack home directory.
JRE_HOME=${JRE_STACK_HOME}
TOMCAT_HOME=${TOMCAT_STACK_HOME}

# Specify the log output path.
installLog="${LOG_PATH}/install_app.log"

# Method of printing logs.
function writeLog()
{
    msg="$1\n"
    printf "[`date '+%Y-%m-%d %H:%M:%S'`] $msg" | sudo tee -a ${installLog};
}

writeLog "-----begin!------"
```

- If you use a custom user to run the script, replace the **www** user name in the following statement in the **stop.sh** script with the custom user name.

  Before:

```
ps -u www -ef|grep -v grep|grep ${JRE_HOME}/bin/java|awk '{print $2}
```

  After:

```
ps -u {Your custom running user} -ef|grep -v grep|grep ${JRE_HOME}/bin/java|awk '{print $2}
```

  Example:

```
ps -u app-user -ef|grep -v grep|grep ${JRE_HOME}/bin/java|awk '{print $2}
```

server.xml Description of Tomcat

- When creating the Tomcat component, you can prepare the **server.xml** file in advance and copy the file to the Tomcat directory. (You are advised to perform this step in the **install** lifecycle.)

```
#!/bin/bash
. ${APP_HOME}/servicestage-vmapp/application.conf
cp ${APP_HOME}/conf/server.xml ${TOMCAT_STACK_HOME}/conf/server.xml
```

- ServiceStage presets a **server.xml** file in **${TOMCAT_STACK_HOME}/conf/server.xml**. The file contains four placeholders. You need to replace them by referring to the following code:

```
#!/bin/bash
. ${APP_HOME}/servicestage-vmapp/application.conf
function replacePara()
{
    sWord=$1
    dWord=$2
    theFile=$3
    if [[ "$sWord" == "" || "$theFile" == "" ]]
    then
        writeLog "[ERROR] ReplacePara has empty param, \$1:$sWord, \$3:$theFile"
        return 1
    fi

    if [[ ! -f ${theFile} ]]
    then
        writeLog "[ERROR] File $theFile does not exist."
        return 1
    fi
    count=`grep -c "$sWord" ${theFile}`
    if [[ "${count}" == "0" ]];then
        return 0
    fi
    sed "s#$sWord#${dWord}#g" ${theFile} > ${theFile}.temp
    if [[ "$?" != "0" ]]
    then
        writeLog "[ERROR] Sed command in replacePara error."
        return 1
    fi
    mv -f ${theFile}.temp ${theFile}
}
replacePara "@{LOG_FILE_PATH_APP}" "${LOG_FILE_PATH_APP}" ${TOMCAT_HOME}/conf/server.xml
replacePara "@{http_port}" "${instance_port_port}" ${TOMCAT_HOME}/conf/server.xml
replacePara "@{server_port}" "${server_port}" ${TOMCAT_HOME}/conf/server.xml
replacePara "@{APP_PACKAGE_NAME}" "examples.war" ${TOMCAT_HOME}/conf/server.xml
```

– **@{server_port}**: port number of the Tomcat server.
– **@{APP_PACKAGE_NAME}**: path of the WAR package relative to **${TOMCAT_STACK_HOME}**.
– **@{http_port}**: port number of the Tomcat connector.
– **@{LOG_FILE_PATH_APP}**: path for printing Tomcat logs.

logging.properties Description of Tomcat

Prepare the **logging.properties** file and copy it to the Tomcat directory. Example of copying the code:

```
#!/bin/bash
. ${APP_HOME}/servicestage-vmapp/application.conf
cp ${APP_HOME}/conf/logging.properties ${TOMCAT_STACK_HOME}/conf/logging.properties
```

The **logging.properties** file contains the following information, where **@{LOG_FILE_PATH_APP}** is the actual log directory.

```
handlers = 1catalina.org.apache.juli.AsyncFileHandler, 2localhost.org.apache.juli.AsyncFileHandler,
3manager.org.apache.juli.AsyncFileHandler, 4host-manager.org.apache.juli.AsyncFileHandler,
java.util.logging.ConsoleHandler

.handlers = 1catalina.org.apache.juli.AsyncFileHandler, java.util.logging.ConsoleHandler

1catalina.org.apache.juli.AsyncFileHandler.level = FINE
1catalina.org.apache.juli.AsyncFileHandler.directory = @{LOG_FILE_PATH_APP}
1catalina.org.apache.juli.AsyncFileHandler.prefix = catalina.

2localhost.org.apache.juli.AsyncFileHandler.level = FINE
2localhost.org.apache.juli.AsyncFileHandler.directory = @{LOG_FILE_PATH_APP}
2localhost.org.apache.juli.AsyncFileHandler.prefix = localhost.

3manager.org.apache.juli.AsyncFileHandler.level = FINE
3manager.org.apache.juli.AsyncFileHandler.directory = @{LOG_FILE_PATH_APP}
3manager.org.apache.juli.AsyncFileHandler.prefix = manager.
```

```
4host-manager.org.apache.juli.AsyncFileHandler.level = FINE
4host-manager.org.apache.juli.AsyncFileHandler.directory = @{LOG_FILE_PATH_APP}
4host-manager.org.apache.juli.AsyncFileHandler.prefix = host-manager.

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].handlers =
3manager.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].handlers = 4host-
manager.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].handlers =
2localhost.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/manager].handlers =
3manager.org.apache.juli.AsyncFileHandler

org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].level = INFO
org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/host-manager].handlers = 4host-
manager.org.apache.juli.AsyncFileHandler
```

FAQ

- How do I run a script as user **root**?
  a. Write the content to be executed by user **root** into the **root-install.sh** script.
  b. Run the following statement in the **install.sh** script to be executed. **${ROOT_PASSWORD}** is the password of user **root**, and the content in the double quotation marks after **-c** is the command to be executed. Environment variables will be lost after you switch user, so add the required environment variables before **sh**.
     ```
     echo "${ROOT_PASSWORD}" | su - root -c "APP_HOME=${APP_HOME} sh root-install.sh"
     ```
  c. If you get the error message "su: Permission denied", find the **/etc/pam.d/su** file and comment out "auth required pam_wheel.so use_uid."