

ServiceStage

User Guide

Issue 01
Date 2021-04-06



Copyright © Huawei Technologies Co., Ltd. 2021. All rights reserved.

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

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks 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 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.

Contents

1 Overview.....	1
2 Permissions Management.....	4
2.1 Creating a User and Granting Permissions.....	4
2.2 Creating a Custom Policy.....	5
2.3 Assigning Permissions on Services that ServiceStage Depends On.....	6
3 Application Management.....	8
3.1 Creating an Application.....	8
3.2 Creating Application Components.....	9
3.2.1 Application Components.....	9
3.2.2 Quickly Creating a Component.....	14
3.2.3 Creating a Microservice Component.....	17
3.2.4 Creating a Web Component.....	20
3.2.5 Creating a Common Component.....	23
3.3 Deploying Application Components.....	25
3.3.1 Deployment Mode.....	26
3.3.2 Deploying a Component.....	26
3.4 Managing Application Components.....	31
3.5 Performing Advanced Settings for an Application.....	34
3.5.1 Setting Application Environment Variables.....	34
3.5.2 Configuring the Lifecycle of an Application.....	36
3.5.3 Configuring Data Storage.....	37
3.5.4 Configuring Distributed Sessions.....	46
3.5.5 Configuring Relational Databases.....	47
3.6 Building an Application Component.....	48
3.7 Pipelining an Application Component.....	48
3.8 Application Configuration.....	50
3.8.1 Creating a Secret.....	50
3.8.2 Creating a ConfigMap.....	53
4 Environment Management.....	56
5 Application O&M.....	58
5.1 Maintaining Application Component Instances.....	58
5.2 Adding Labels for Application Component Instances.....	61

5.3 Configuring Domain Name Mappings.....	62
5.4 Setting Alarm Thresholds for Resource Monitoring.....	62
5.5 Setting Scaling Policies for Application Component Instances.....	65
5.6 Setting Scheduling Policies for Application Component Instances.....	69
5.7 Setting Upgrade Policies for Application Component Instances.....	74
5.8 Setting Custom Metric Monitoring for Application Components.....	76
5.9 Configuring Application Log Policies.....	79
5.10 Configuring Application Performance Management.....	81
5.11 Configuring Health Check.....	82
6 Microservice Governance.....	85
6.1 Overview.....	85
6.2 Using the Microservice Dashboard.....	85
6.3 Governing Microservices.....	86
6.4 Configuring Microservices.....	93
6.5 Maintaining Microservices.....	95
7 Continuous Delivery.....	100
7.1 Overview.....	100
7.2 Creating a Source Code Build Task.....	101
7.3 Creating a Package Build Task.....	105
7.4 Managing Pipelines.....	107
7.5 Authorizing a Repository.....	111
8 Software Center.....	112
8.1 Software Repository.....	112
8.1.1 Managing Software Packages.....	112
8.1.2 Packaging Specifications of Software Packages.....	116
8.2 Image Repository.....	117
8.2.1 Uploading an Image.....	117
8.2.2 Managing Images.....	119
8.3 Organization Management.....	122
9 Infrastructure Management.....	125
9.1 Cloud Service Engines.....	125
9.1.1 Creating an Exclusive Microservice Engine.....	125
9.1.2 Configuring Backup and Restoration of an Exclusive Microservice Engine.....	127
9.1.3 Configuring Public Network Access for an Exclusive Microservice Engine.....	128
9.1.4 Viewing the Access Address of a Microservice Engine.....	129
9.1.5 Viewing Operation Logs of an Exclusive Microservice Engine.....	130
9.1.6 Upgrading an Exclusive Microservice Engine.....	130
9.1.7 Deleting an Exclusive Microservice Engine.....	131
9.2 VMAgent Manager.....	132

1 Overview

ServiceStage is an application management and O&M platform that lets you deploy, roll out, monitor, and maintain applications all in one place. Java, Go, PHP, Node.js, Docker, and Tomcat are supported. Web applications, microservice applications such as Apache ServiceComb, Spring Cloud, Dubbo, and service mesh, and common applications make it easier to migrate enterprise applications to the cloud.

This document describes how to use ServiceStage to create, deploy, and maintain applications and perform service governance.

Console Description

[Table 1-1](#) describes ServiceStage console.

Table 1-1 ServiceStage console

Module	Description
Overview	Provides ServiceStage overview, including the ServiceStage package selection and purchase entry, tutorials, applications, environments, and components.
Application Management	<ul style="list-style-type: none">• Application List Provides application lifecycle management, such as application creation, component addition, component list, environment view, component deployment, component details, and O&M.• Application Component Displays the components (including static and deployed components) of the application, and component details and O&M.• Application Configuration Supports configuration item and secret management.

Module	Description
Environment Management	<p>Environment is a collection of infrastructures, covering computing, storage, and networks, used for application deployment and running.</p> <p>Provides environment creation, editing, and deletion, and displays resource information in an existing environment.</p>
Continuous Delivery	<p>Supports project build and release.</p> <ul style="list-style-type: none"> • Build The software package or image package can be generated with a few clicks in job building. In this way, the entire process of source code pull, compilation, packaging, and archiving is automatically implemented. • Pipeline One-click deployment can be achieved through pipeline. In this way, the entire process of source code pull, complication, packaging, archiving, and deployment is automatically implemented. This unifies the integration environment and standardizes the delivery process. • Repository Authorization You can create repository authorization so that build projects and application components can use the authorization information to access the software repository.
Software Center	<p>Provides functions such as organization management, software repository, and image repository.</p> <ul style="list-style-type: none"> • Organization management is used to isolate images and assign access permissions (read, write, and manage) to different users. • Image repositories are used to store and manage Docker images. • Software repositories are used to store, manage, and deploy software packages.
Infrastructure	<p>Provides application infrastructure management, such as Cloud Service Engine (CSE) and VM agent management (VMAgent).</p> <p>On the CSE page, go to its console to perform microservice governance.</p>
Operation List	<p>After the Cloud Trace Service (CTS) is enabled, the system automatically traces operations and changes of all cloud resources of the current tenant and saves the information as traces for seven days. Advanced functions, such as trace transfer (long-term storage) and encrypted storage, can be configured in the tracker list.</p>
Help Center	<p>Provides an overview of ServiceStage documentation.</p>

 **NOTE**

The VM agent management function depends on the ECS and AOM services. If these services are not installed, the VM agent management function is unavailable.

Package Description

Log in to the ServiceStage console and select an edition on the **Overview** page. Currently, ServiceStage provides basic edition and professional edition..

Table 1-2 ServiceStage edition description

Edition	Package Description
Basic	20 instances are free to use.
Professional	One exclusive CSE engine and AOM enterprise edition are free to use.

 **NOTE**

For product pricing of each edition, see [Product Pricing Details](#).

2 Permissions Management

[Creating a User and Granting Permissions](#)

[Creating a Custom Policy](#)

[Assigning Permissions on Services that ServiceStage Depends On](#)

2.1 Creating a User and Granting Permissions

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

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

If your HUAWEI CLOUD account does not require individual IAM users, skip this section.

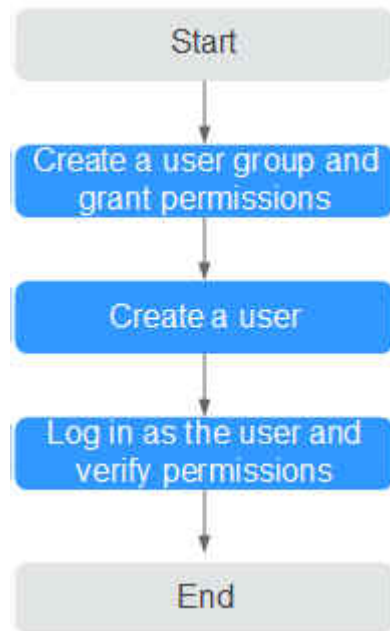
This section describes the procedure for granting permissions (see [Figure 2-1](#)).

Prerequisites

Before assigning permissions to user groups, you should learn about the ServiceStage permissions listed in [Permissions Management](#). For the system policies of other services, see [System Permissions](#).

Process Flow

Figure 2-1 Process for granting ServiceStage permissions



1. Create a user group and grant permissions to it.
Create a user group on the IAM console, and assign the **ServiceStage ReadOnlyAccess** policy to the group.
2. Create a user.
Create a user on the IAM console and add the user to the group created in **1**.
3. Log in and verify permissions.
Log in to the ServiceStage console as the created user, and verify that it only has read permissions for ServiceStage.
 - Select **ServiceStage** from **Service List**. Choose **Application Management > Application List** from the navigation tree. On the page that is displayed, click **Create Application**. If a message appears indicating insufficient permissions to access the service, the **ServiceStage ReadOnlyAccess** policy has already taken effect.
 - Choose any other service in the **Service List**. If a message appears indicating insufficient permissions to access the service, the **ServiceStage ReadOnlyAccess** policy has already taken effect.

2.2 Creating a Custom Policy

Custom policies can be created as a supplement to the system policies of ServiceStage.

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

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.

- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see [Creating a Custom Policy](#). The following section contains examples of common ServiceStage custom policies.

Example Custom Policy

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

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

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

After authorization, users in the group can verify their permissions using the console or REST APIs.

The following are the steps to be performed on the console.

1. Log in to HUAWEI CLOUD as an IAM user.
 - Account name: Name of the account used to create the IAM user
 - Username and password: Username and password specified for the IAM user
2. On the ServiceStage console, choose **Infrastructure > Cloud Service Engines**, and buy a microservice engine. If error 403 is returned, the permissions are correct and have already taken effect.

2.3 Assigning Permissions on Services that ServiceStage Depends On

Granting CCE Namespace Permissions

You can grant only common operation permissions on CCE cluster resources to the ServiceStage user group using IAM, excluding the namespace permissions of the clusters with Kubernetes RBAC authentication enabled. Therefore, you need to separately grant the namespace permissions for the clusters.

For details about how to set CCE namespace permissions, see [Permissions Management](#).

Granting CTS Permissions

Currently, Cloud Trace Service (CTS) does not support fine-grained authorization. After the permissions are set for ServiceStage using IAM, they do not take effect for the CTS service on which ServiceStage depends. You need to set the CTS service permissions separately.

For details about how to set CTS namespace permissions, see [Permissions Management](#).

3 Application Management

[Creating an Application](#)

[Creating Application Components](#)

[Deploying Application Components](#)

[Managing Application Components](#)

[Performing Advanced Settings for an Application](#)

[Building an Application Component](#)

[Pipelining an Application Component](#)

[Application Configuration](#)

3.1 Creating an Application

An application is a service system with functions and consists of one or more application components.

For example, the weather forecast is an application that contains the weather and forecast components. ServiceStage organizes multiple components by application, and supports dark launch and quick cloning of applications in different environments.

Creating an Application

Step 1 Log in to ServiceStage and choose **Application Management > Application List**.

Step 2 Click **Create Application** and set basic application information.

1. **Name:** Enter an application name. This name cannot be changed after the application is created.
2. **Enterprise Project:** Set an enterprise project.

Enterprise projects provide a cloud resource management mode, in which cloud resources and members are centrally managed by project.

It is available after the [enterprise center](#) is enabled.

3. **Description:** (Optional) Enter an application description.

Step 3 Click **OK**.

----End

Adding Environment Variables

An environment is a collection of infrastructures, covering computing, storage, and networks, used for application deployment and running. ServiceStage combines basic resources (such as CCE and ECS) and optional resources (such as ELB, RDS, and DCS) in the same VPC into an environment, such as the development environment, testing environment, pre-production environment, and production environment. Networks in an environment can communicate with each other. You can manage resources and deploy services by environment, simplifying infrastructure O&M.

Environment variables are parameters set in the system or user applications. You can obtain the values of environment variables by calling APIs. During deployment, parameters are specified through environment variables instead of in the code, which makes the deployment flexible.

Step 1 Log in to ServiceStage and choose **Application Management > Application List**.

Step 2 Click the application name. The **Overview** page is displayed.

Step 3 Click **Environment Variables** and select a created environment from the **Environment** drop-down list.

Step 4 Click **Add Environment Variable** and enter the values in **Key** and **Value**.

Key indicates the name of the environment variable, and **Value** indicates the value of the environment variable. Click **Submit**.

For example, set **Key** to **User** and **Value** to **admin**. That is, when the program code reads the **User** environment variable, **admin** is obtained. For example, you can start subprocesses as the admin user and read files as the admin user. The actual execution effect depends on the code.

----End

3.2 Creating Application Components

3.2.1 Application Components

An application component implements a service feature of an application. It is in the form of code or software packages and can be deployed independently.

After creating an application on ServiceStage, you can add components to the application. Currently, microservice, web, and common application components are supported.

You can create a static component by setting the component type, framework, runtime system, and component source, and then deploy this component.

In the process of adding a component, you can configure the component using a template (**Template**) or customize the configuration (**Custom**).

- **Template** provides default configurations of the component type, runtime system, and framework to help you quickly create components.
- **Custom** allows you to select the desired component type, runtime system, and proper framework/service mesh.

Existing Templates

Table 3-1 Existing templates

Type	Runtime System	Framework
ServiceComb MicroService	Java8	Java Chassis
SpringCloud MicroService	Java8	SpringCloud
Web(Tomcat) WebApp	Tomcat8	Web

Microservice Components

Supported Runtime System	Supported Framework/ Service Mesh	Supported Source Code/Software Package
Java8	Java Chassis	Source code repository, template, and JAR package
Tomcat8		Source code repository, template, and WAR package
Docker		This parameter does not need to be set.
Java8	Mesher	Source code repository and JAR package
Tomcat8		Source code repository and WAR package
Node.js8		Source code repository and ZIP package
Php7		Source code repository and ZIP package
Docker		This parameter does not need to be set.
Python3		Source code repository and ZIP package

Supported Runtime System	Supported Framework/Service Mesh	Supported Source Code/Software Package
Docker	Go Chassis	This parameter does not need to be set.
Java8	Spring Cloud	Source code repository and JAR package
Tomcat8		Source code repository and WAR package
Docker		This parameter does not need to be set.
Java8	Dubbo	Source code repository, template, and JAR package
Tomcat8		Source code repository, template, and WAR package
Docker		This parameter does not need to be set.

Web Application Components

Supported Runtime System	Supported Source Code/Software Package
Java8	Source code repository, template, and JAR package
Nodejs8	Source code repository, template, and ZIP package
Php7	Source code repository, template, and ZIP package
Tomcat8	Source code repository, template, and WAR package
Docker	This parameter does not need to be set.
Python3	Source code repository and ZIP package

Common Components

Supported Runtime System	Supported Source Code/Software Package
Java8	Source code repository, template, and JAR package

Supported Runtime System	Supported Source Code/Software Package
Tomcat8	Source code repository, template, and WAR package
Node.js8	Source code repository, template, and ZIP package
Php7	Source code repository, template, and ZIP package
Docker	This parameter does not need to be set.
Python3	Source code repository and ZIP package

Component Source

Component Source	Description
Source code repository	Create authorization by referring to Authorizing a Repository and set the code source.
JAR package	Supports the following uploading modes: <ol style="list-style-type: none">1. Select the corresponding software package from the SWR software repository. You need to upload the software package to the software repository in advance. For details, see Uploading the Software Package.2. Select the corresponding software package from OBS. You need to upload the software package to the OBS bucket in advance. For details, see Uploading a File.

Component Source	Description
WAR package	<p>Supports the following uploading modes:</p> <ol style="list-style-type: none"> 1. Select the corresponding software package from the SWR software repository. You need to upload the software package to the software repository in advance. For details, see Uploading the Software Package. 2. Select the corresponding software package from OBS. You need to upload the software package to the OBS bucket in advance. For details, see Uploading a File.
ZIP package	<p>Supports the following uploading modes:</p> <ol style="list-style-type: none"> 1. Select the corresponding software package from the SWR software repository. You need to upload the software package to the software repository in advance. For details, see Uploading the Software Package. 2. Select the corresponding software package from OBS. You need to upload the software package to the OBS bucket in advance. For details, see Uploading a File.
Image package	<p>If you use a private image to create your containerized application, upload the private image to the image repository. Choose Software Center > Image Repository and upload the image to the image repository by referring to Managing Images.</p>
Template	<p>Create authorization by referring to Authorizing a Repository and set the organization and repository names.</p> <p>ServiceStage provides component templates. You can select a template to quickly create an application and generate a development project in the configured code repository. For details, see Template Framework.</p>

Template Framework

Runtime System	Framework Provided by the Template	Framework Description
Java8	CSE-Java (SpringMVC)	Based on the ServiceComb microservice development framework, supports SpringMVC annotations and uses the SpringMVC style to develop microservices.
	CSE-Java (JAX-RS)	Based on the ServiceComb microservice development framework, supports JAX-RS annotations and uses the JAX-RS mode to develop microservices.
	CSE-Java (POJO)	Based on the ServiceComb microservice development framework, supports APIs and API implementation, and uses transparent RPC to develop microservices.
Tomcat8	SpringBoot-Webapp-Tomcat	Web applications, running on an independent web server.
	SpringBoot-WebService-Tomcat	Web services, running on an independent web server.
Nodejs8	Express	A Node.js web framework that supports high compatibility, and fast and simple deployment.
	Koa	Next-generation web development framework based on the Node.js platform.
Php7	Laravel	A PHP development framework for web developers.
	Slim	A lightweight micro-PHP framework.

3.2.2 Quickly Creating a Component

ServiceStage provides three default templates. For details, see [Existing Templates](#).

A template provides default configurations of the component type, language/runtime system, and framework/service mesh to help you quickly create a component.

Prerequisites

1. An application has been created cause components can only be added to applications. For details, see [Creating an Application](#).
2. If you create a microservice component based on the source code repository or template, create repository authorization first. For details, see [Authorizing a Repository](#).
3. If you create a microservice component based on the software package, upload the software package to the software repository or OBS bucket.
 - Upload the software package to the software repository. For details, see [Uploading the Software Package](#).
 - Upload the software package to OBS. For details, see [Uploading a File](#).

Procedure

- Step 1** Log in to ServiceStage and choose **Application Management > Application List**.
- Step 2** Select the created application and click **Create Component** in the **Operation** column.
- Step 3** Select **Template** for **Configuration Method**, select a template, and click **Next**.
- Step 4** Configure component information according to the following table. Parameters marked with an asterisk (*) are mandatory.

Table 3-2 Basic component information

Parameter	Description
*Name	Component name.

Parameter	Description
*Source Code/ Software Package	<ul style="list-style-type: none">• Select Source code repository.<ul style="list-style-type: none">– Create authorization by referring to Authorizing a Repository and set the code source.– Set Build parameters to build the application component. Set Command, Organization, and CPU Architecture, and select a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management. <p>NOTICE If Custom command is selected for Command: Exercise caution when inputting sensitive information in the echo, cat, or debug command, or encrypt sensitive information to avoid information leakage.</p> <ul style="list-style-type: none">• Select JAR package or WAR package.<p>NOTE Select JAR package if Java8 is selected as the Runtime System. Select WAR package if Tomcat8 is selected as the Runtime System.</p><ol style="list-style-type: none">1. Select Upload Method. Select the corresponding software package from the SWR software repository. Upload the software package to the software repository in advance. For details, see Uploading the Software Package. Upload the software package to OBS. For details, see Uploading a File.2. (Optional) Set Build parameters to build the application component. Set Organization and CPU Architecture, and select a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management.• Set the following parameters if Template is selected.<ol style="list-style-type: none">1. Select the template framework. ServiceStage provides template frameworks. You can select one to quickly create an application component.2. Set Code Archive. See Authorizing a Repository to create authorization and set Username/Organization and Repository.

Step 5 Complete component creation.

- Click **Create Now** to create a static component.

- Click **Create and Deploy**. The deployment page is displayed. For details, see [Deploying a Component](#).

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

----End

3.2.3 Creating a Microservice Component

ServiceStage provides a microservice framework that enables you to develop and deploy applications on the cloud. It provides code framework generation, service registry and discovery, load balancing, and service reliability including fault tolerance, circuit breaker, rate limiting, and service degradation. This section describes how to create a static microservice application component using ServiceStage. For details about how to deploy a component, see [Deploying a Component](#).

Prerequisites

1. An application has been created cause components can only be added to applications. For details, see [Creating an Application](#).
2. If you create a microservice component based on the source code repository or template, create repository authorization first. For details, see [Authorizing a Repository](#).
3. If you create a microservice component based on the software package, upload the software package to the software repository or OBS bucket.
 - Upload the software package to the software repository. For details, see [Uploading the Software Package](#).
 - Upload the software package to OBS. For details, see [Uploading a File](#).

Procedure

Step 1 Log in to ServiceStage and choose **Application Management > Application List**.

Step 2 Select the created application and click **Create Component** in the **Operation** column.

Step 3 Select **Custom** for **Configuration Method** and **Microservice** for **Select Component Type**, and click **Next**.

Step 4 Select **Runtime System** and click **Next**.

Different frameworks support different runtime systems. For details, see [Microservice Components](#).

Step 5 Select **Framework/Service Mesh**.

For details about the framework/service mesh, see [Microservice Components](#).

Step 6 Select whether you want to save the preceding configurations as a template for future use.

- If you select this function, enter a template name. Then, go to [Step 7](#).
- If you do not select this function, go to [Step 7](#).

Step 7 Check whether **Docker** is selected in **Step 4**.

- If yes, click **Next** and go to **Step 8**.
- If no, click **Next** and go to **Step 9**.

Step 8 Create a Docker component.

1. Enter a component name.
2. Create a component.
 - Click **Create Now** to create a static component.
 - Click **Create and Deploy**. The deployment page is displayed. For details, see **Deploying a Component**.
3. No further action is required.

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

Step 9 Configure component information according to the following table. Parameters marked with an asterisk (*) are mandatory.

Table 3-3 Basic component information

Parameter	Description
*Name	Component name.

Parameter	Description
<p>*Source Code/ Software Package</p>	<ul style="list-style-type: none"> ● Select Source code repository. <ol style="list-style-type: none"> 1. Create authorization by referring to Authorizing a Repository and set the code source. 2. Set Build parameters to build the application component. Set Command, Organization, and CPU Architecture, and build a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management. <p>NOTICE If Custom command is selected for Command: Exercise caution when inputting sensitive information in the echo, cat, or debug command, or encrypt sensitive information to avoid information leakage.</p> ● Select JAR package, WAR package, or ZIP package. <p>NOTE Select JAR package if Java8 is selected as the Runtime System. Select WAR package if Tomcat8 is selected as the Runtime System. Select ZIP package if Nodejs8, Php7, or Python3 is selected as the Runtime System.</p> <ol style="list-style-type: none"> 1. Select Upload Method. Select the corresponding software package from the SWR software repository. Upload the software package to the software repository in advance. For details, see Uploading the Software Package. Upload the software package to OBS. For details, see Uploading a File. 2. (Optional) Set Build parameters to build the application component. Set Organization and CPU Architecture, and build a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management. ● Set the following parameters if Template is selected. <ol style="list-style-type: none"> 1. ServiceStage provides template frameworks. You can select one to quickly create an application component. 2. Set Code Archive. See Authorizing a Repository to create authorization and set Username/Organization and Repository. <p>NOTE This parameter is invalid if you select Mesh or Spring Cloud for Framework/Service Mesh in Step 5.</p>

Step 10 Create a component.

- Click **Create Now** to create a static component.
- Click **Create and Deploy**. The deployment page is displayed. For details, see [Deploying a Component](#).

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

----End

3.2.4 Creating a Web Component

This section describes how to create a static web application component using ServiceStage. For details, see [Deploying a Component](#).

Prerequisites

1. An application has been created cause components can only be added to applications. For details, see [Creating an Application](#).
2. If you create a microservice component based on the source code repository or template, create repository authorization first. For details, see [Authorizing a Repository](#).
3. If you create a microservice component based on the software package, upload the software package to the software repository or OBS bucket.
 - Upload the software package to the software repository. For details, see [Uploading the Software Package](#).
 - Upload the software package to OBS. For details, see [Uploading a File](#).

Procedure

Step 1 Log in to ServiceStage and choose **Application Management > Application List**.

Step 2 Select the created application and click **Create Component** in the **Operation** column.

Step 3 Select **Custom** for **Configuration Method**, select **Web** for **Component Type**, and click **Next**.

Step 4 Select **Runtime System** and click **Next**.

Different frameworks support different runtime systems. For details, see [Microservice Components](#).

Step 5 Select whether you want to save the preceding configurations as a template for future use.

- If you select this function, enter a template name. Then, go to [Step 6](#).
- If you do not select this function, go to [Step 6](#).

Step 6 Check whether **Docker** is selected in [Step 4](#).

- If yes, click **Next** and go to [Step 7](#).
- If no, click **Next** and go to [Step 8](#).

Step 7 Create a Docker component.

1. Enter a component name.
2. Create a component.
 - Click **Create Now** to create a static component.
 - Click **Create and Deploy**. The deployment page is displayed. For details, see [Deploying a Component](#).
3. No further action is required.

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

Step 8 Configure component information according to the following table. Parameters marked with an asterisk (*) are mandatory.**Table 3-4** Basic component information

Parameter	Description
*Name	Component name.

Parameter	Description
<p>*Source Code/ Software Package</p>	<ul style="list-style-type: none"> ● Select Source code repository. <ol style="list-style-type: none"> 1. Create authorization by referring to Authorizing a Repository and set the code source. 2. Set Build parameters to build the application component. Set Command, Organization, and CPU Architecture, and select a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management. <p>NOTICE If Custom command is selected for Command: Exercise caution when inputting sensitive information in the echo, cat, or debug command, or encrypt sensitive information to avoid information leakage.</p> ● Select JAR package, WAR package, or ZIP package. <p>NOTE Select JAR package if Java8 is selected as the Runtime System. Select WAR package if Tomcat8 is selected as the Runtime System. Select ZIP package if Nodejs8, Php7, or Python3 is selected as the Runtime System.</p> <ol style="list-style-type: none"> 1. Select Upload Method. Select the corresponding software package from the SWR software repository. Upload the software package to the software repository in advance. For details, see Uploading the Software Package. Upload the software package to OBS. For details, see Uploading a File. 2. (Optional) Set Build parameters to build the application component. Set Organization and CPU Architecture, and select a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management. ● Set the following parameters if Template is selected. <ol style="list-style-type: none"> 1. ServiceStage provides template frameworks. You can select one to quickly create an application component. 2. Set Code Archive. See Authorizing a Repository to create authorization and set Username/Organization and Repository. <p>NOTE This parameter is invalid if you select Python3 for Runtime System in Step 4.</p>

Step 9 Create a component.

- Click **Create Now** to create a static component.
- Click **Create and Deploy**. The deployment page is displayed. For details, see [Deploying a Component](#).

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

----End

3.2.5 Creating a Common Component

This section describes how to create a static common application component using ServiceStage. For details about how to deploy a component, see [Deploying a Component](#).

Prerequisites

1. An application has been created cause components can only be added to applications. For details, see [Creating an Application](#).
2. If you create a microservice component based on the source code repository or template, create repository authorization first. For details, see [Authorizing a Repository](#).
3. If you create a microservice component based on the software package, upload the software package to the software repository or OBS bucket.
 - Upload the software package to the software repository. For details, see [Uploading the Software Package](#).
 - Upload the software package to OBS. For details, see [Uploading a File](#).

Procedure

Step 1 Log in to ServiceStage and choose **Application Management > Application List**.

Step 2 Select the created application and click **Create Component** in the **Operation** column.

Step 3 Select **Custom** for **Configuration Method**, select **Common** for **Component Type**, and click **Next**.

Step 4 Select **Runtime System** and click **Next**.

Different frameworks support different runtime systems. For details, see [Microservice Components](#).

Step 5 Select whether you want to save the preceding configurations as a template for future use.

- If you select this function, enter a template name. Then, go to [Step 6](#).
- If you do not select this function, go to [Step 6](#).

Step 6 Check whether **Docker** is selected in [Step 4](#).

- If yes, click **Next** and go to [Step 7](#).

- If no, click **Next** and go to **Step 8**.

Step 7 Create a Docker component.

1. Enter a component name.
2. Create a component.
 - Click **Create Now** to create a static component.
 - Click **Create and Deploy**. The deployment page is displayed. For details, see **Deploying a Component**.
3. No further action is required.

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

Step 8 Configure component information according to the following table. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Name	Component name.
*Source Code/ Software Package	<ul style="list-style-type: none"> • Source code repository: Create authorization by referring to Authorizing a Repository and set the code source. • JAR package/WAR package/ZIP package: Set Upload Method. Select the corresponding software package from the SWR software repository. Upload the software package to the software repository in advance. For details, see Uploading the Software Package. Upload the software package to OBS. For details, see Uploading a File. NOTE Select JAR package if Java8 is selected as the Runtime System. Select WAR package if Tomcat8 is selected as the Runtime System. Select ZIP package if Nodejs8, Php7, or Python3 is selected as the Runtime System. • Set the following parameters if Template is selected. <ol style="list-style-type: none"> 1. ServiceStage provides template frameworks. You can select one to quickly create an application component. 2. Set Code Archive. See Authorizing a Repository to create authorization and set Username/Organization and Repository. NOTE This parameter is invalid if you select Python3 for Runtime System in Step 4.

Parameter	Description
*Python framework	<p>This parameter is mandatory if you select Python3 in Step 4. Set Module Name and Variable Name for all Python frameworks except Python3-Django.</p> <ul style="list-style-type: none"> If the entry point file of the Python project is server.py, Module Name is server. If the application function of the server.py entry point file of the Python project is app=get_wsgi_application(), Variable Name is app.
Build	<ul style="list-style-type: none"> If Source code repository is selected for Source Code/Software Package, set Build parameters to build the application component. Set Command, Organization, and CPU Architecture, and select a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management. <p>NOTICE If Custom command is selected for Command: Exercise caution when inputting sensitive information in the echo, cat, or debug command, or encrypt sensitive information to avoid information leakage.</p> <ul style="list-style-type: none"> (Optional) If JAR package, WAR package, or ZIP package is selected for Source Code/Software Package, set Build parameters to build the application component. Set Organization and CPU Architecture, and select a cluster based on service requirements. You can also specify Node Label to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see Node Management.

Step 9 Create a component.

- Click **Create Now** to create a static component.
- Click **Create and Deploy**. The deployment page is displayed. For details, see [Deploying a Component](#).

After the component is created, you can view the component status in the **Component List** on the **Overview** tab.

----End

3.3 Deploying Application Components

3.3.1 Deployment Mode

Deploying a Component Using CCE

Cloud Container Engine (CCE) provides highly scalable, high-performance, enterprise-class Kubernetes clusters and supports Docker containers. With CCE, you can easily deploy, manage, and scale containerized applications on the cloud platform.

If the build function is not enabled for the created component, the component cannot be deployed using a container.

Deploying a Component Using VM

The created component can be deployed using a VM only when **Java8**, **Tomcat8**, or **Nodejs8** is selected for **Select Runtime System**.

3.3.2 Deploying a Component

This section describes how to deploy static components in the corresponding environment.

When creating an application component, you can also select **Create and Deploy**. The deployment procedure is the same as that described in this section.

Prerequisites

1. An application component has been created or is being created, and has been configured. For details, see [Creating Application Components](#).
2. The environment has been created. For details, see [Environment Management](#).
3. If you deploy components based on software packages or image packages, you need to upload the software packages or image packages.
 - Upload the software package to the software repository. For details, see [Uploading the Software Package](#).
 - Upload the software package to OBS. For details, see [Uploading a File](#).
 - Upload the image package to the image repository. For details, see [Uploading an Image](#).
4. Configure the AK/SK. For details, see [Checking and Configuring the AKSK Authentication Mode](#).

Procedure

- Step 1** Log in to ServiceStage and choose **Application Management >Application List**.
- Step 2** Click the name of the created application. The **Overview** page is displayed.
- Step 3** On the **Component List** tab, select a created component and click **Deploy** in the **Operation** column.
- Step 4** Set basic parameters. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Environment	Select an environment you created. NOTE Only the environment of the same enterprise project can be selected.
*Version	Component version number, for example, 1.0.0.
Description	Provides supplementary information about the component.
*Deployment System	Supports Cloud Container Engine, and VM. For details, see Deployment Mode .
*Resource Type	This parameter is valid when VM is selected for Deployment System . AS Groups and Elastic Cloud Servers (ECSs) are supported.
*Basic Resource	The basic resources contained in the selected environment are automatically loaded. Select the resources as required.
*Instances	Number of instances in an application component. An application component can have one or more instances. You can specify the number of instances as required. Configuring multiple instances for an application component ensures high reliability of the application component. For such a component, if an instance is faulty, the component can still run properly. NOTE This parameter is not displayed when you select VM deployment. The number of component instances is determined by the number of Basic Resources .
*Resource Quota	Components cannot be scheduled to nodes whose residual resources are fewer than the requested amount. For details about how to configure the request and limit parameters, see Managing Resources for Containers . You can customize CPU and Memory as required. NOTE This parameter is not displayed during deployment when the component type is Common and the runtime system is Docker .
Component Status	Sets the component status as required. NOTE This parameter is available when the component type is Common , the runtime system is Docker , and Cloud Container Engine is selected for Deployment System .

Step 5 Click **Next** to configure the component.

- When the component type is **Common** and the runtime system is **Docker**, perform the following operations:
 - a. Select an image. Multiple containers are supported. You can click **Add Container** to add an image.

- b. Select an image version.
- c. Enter a container name.
- d. (Optional) Set **Resource Quota**. Components cannot be scheduled to nodes whose residual resources are fewer than the requested amount. For details about how to configure the request and limit parameters, see [Managing Resources for Containers](#). You can customize **CPU** and **Memory** as required.
- e. (Optional) Set advanced parameters.
 - Choose **Advanced Settings > Component Configuration** and set environment variables. For details, see [Setting Application Environment Variables](#).
 - Choose **Advanced Settings > Deployment Configuration**.
 - Set **Startup Command** and **Lifecycle**. For details, see [Configuring the Lifecycle of an Application](#).
 - Set **Data Storage**. For details, see [Configuring Data Storage](#).
 - Choose **Advanced Configuration > O&M Monitoring**.
 - Set **Log Collection**. For details, see [Configuring Application Log Policies](#).
 - Set **Health Check**. For details, see [Configuring Health Check](#).
- f. (Optional) Enable **Public Network Access**.
 - i. Set **Public Network Load Balancer**.

Select the created load balancer.

If no load balancer exists, click **Add Load Balancer** to create one. For details, see [Using Shared Load Balancers — Entry Level](#).
 - ii. (Optional) Set **HTTPS**.

If HTTPS is enabled, click **Use existing** to select an existing certificate.

If no certificate exists, click **Create new** to create a server certificate. For details about how to create a server certificate, see [Creating a Certificate](#).
 - iii. Set **Domain Name**.

Enter a customize domain name if **Bound** is selected. For details, see [Configuring Domain Name Mappings](#).
 - iv. Set **Listening Port**.

Set the listening port of the application process.
- g. (Optional) Set **Database**.

Select **Distributed session**. For details, see [Configuring Distributed Sessions](#).

Select **RDS DB instance**. For details, see [Configuring Relational Databases](#).
- h. (Optional) Set **Local Time**.

Change the time zone of the container node. By default, the time zone is the same as that of the region where the container node is located.

- i. (Optional) Set **Scheduling Policies**. For details, see [Setting Scheduling Policies for Application Component Instances](#).
- j. (Optional) Set **Upgrade Policies**. For details, see [Setting Upgrade Policies for Application Component Instances](#).
- k. (Optional) Set **Performance Management**. For details, see [Configuring Application Performance Management](#).
- For other types of components, perform the following operations:
 - a. Set **Image**.
 - If the application source is a software package, source code, or template, the configured static component information will be loaded.
 - If **Runtime System** is set to **Docker**, select an image package from the SWR image repository.
 - b. (Optional) Enable **Public Network Access**.
 - i. Set **Public Network Load Balancer**.
Select the created load balancer.
If no load balancer exists, click **Add Load Balancer** to create one. For details, see [Using Shared Load Balancers — Entry Level](#).
 - ii. (Optional) Set **HTTPS**.
If HTTPS is enabled, click **Use existing** to select an existing certificate.
If no certificate exists, click **Create new** to create a server certificate. For details about how to create a server certificate, see [Creating a Certificate](#).
 - iii. Set **Domain Name**.
Enter a customize domain name if **Bound** is selected. For details, see [Configuring Domain Name Mappings](#).
 - iv. (Optional) Set **Listening Port**.
Listening port of an application process. If **Tomcat8** is selected as the **Runtime System**, this port is set to **8080** by default. You can customize this port.
 - c. (Optional) Set **JVM**.
This parameter is mandatory when **Runtime System** is set to **Java8** or **Tomcat8**.
Enter the JVM parameter, for example, **-Xms256m -Xmx1024m**. Multiple parameters are separated by spaces. If the parameter is left blank, the default value is used.
 - d. (Optional) Configure Tomcat parameters.
This parameter is mandatory when **Runtime System** is set to **Tomcat8**.
 - i. Select **Parameter settings**. The **Tomcat** dialog box is displayed.
 - ii. Click **Use Sample Code** and edit the template file based on service requirements.
 - iii. Click **OK**.

- e. (Optional) Configure **Cloud Service Engine**.
This parameter is mandatory for microservice components.
By default, the microservice engine added in the environment is selected.
For details about how to create a microservice engine, see [Creating an Exclusive Microservice Engine](#).
- f. (Optional) Set **Database**.
Select **Distributed session**. For details, see [Configuring Distributed Sessions](#).
Select **RDS DB instance**. For details, see [Configuring Relational Databases](#).
- g. (Optional) Set **Local Time**.
Change the time zone of the container. By default, the time zone is the same as that of the region where the container node is located.
- h. (Optional) Set advanced parameters.
If the deployment system is set to **VM**, only environment variables can be set.
 - Choose **Advanced Settings > Component Configuration** and set environment variables. For details, see [Setting Application Environment Variables](#).
 - Choose **Advanced Settings > Deployment Configuration**.
 - Set **Startup Command** and **Lifecycle**. For details, see [Configuring the Lifecycle of an Application](#).
 - Set **Data Storage**. For details, see [Configuring Data Storage](#).
 - Set **Scheduling Policy**. For details, see [Setting Scheduling Policies for Application Component Instances](#).
 - Set **Upgrade Policy**. For details, see [Setting Upgrade Policies for Application Component Instances](#).
 - Choose **Advanced Configuration > O&M Monitoring**.
 - Set **Log Collection**. For details, see [Configuring Application Log Policies](#).
 - Set **Health Check**. For details, see [Configuring Health Check](#).
 - Set **Performance Management**. For details, see [Configuring Application Performance Management](#).
 - Set **O&M Policy**. For details, see [Setting Custom Metric Monitoring for Application Components](#).

Step 6 Click **Next**, confirm the specifications, and click **Deploy**.

After the component is deployed, you can view the component status in the **Environment View** on the **Overview** tab.

----End

3.4 Managing Application Components

After a component is created or deployed, you can perform the following management operations:

- **Viewing Application Components:** View the list of components created under the application.
- **Deploying a Component:** Deploy the created static components.
- **Updating Component Source:** Update the source code/software package, version, and environment configuration of the components. Components whose runtime system is **Docker** do not support this operation.
- **Deleting Components:** Delete the created components.
- **Creating a Pipeline for a Component:** One-click deployment can be achieved through pipeline. In this way, the entire process of source code pull, compilation, packaging, archiving, and deployment is automatically implemented. This unifies the integration environment and standardizes the delivery process.
- **Viewing Application Component Building:** View the status of the application component building project.
- **Maintaining Component Instances:** Maintain the deployed application component instances.
- **Managing Component Instance Access Mode:** Set the access mode of the component instances.

Viewing Application Components

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 Click the **Component List** tab to view the list of components created for the application.

----End

Deploying a Component

For details about how to deploy a component, see [Deploying a Component](#).

Updating Component Source

After a component is created, you can update the source code/software package, version, and environment configuration of the component.

Components whose runtime system is **Docker** do not support this operation.

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 Click the **Component List** tab.

- To update the source of a single component, select the component and click **Update Source** in the **Operation** column.
- To update the component sources in batches, select multiple components and click **Update Component Source**.

Step 4 Set **Source Code Repository/Software Package**.

- **Source code repository:** Create authorization by referring to [Authorizing a Repository](#) and set the code source.
- **Software Package:**
 - Click **Replace Software Package** and select the corresponding software package from the SWR software repository. Upload the software package to the software repository in advance. For details, see [Uploading the Software Package](#).
 - Click **Replace Software Package** and select the corresponding software package from OBS. You need to upload the software package to the OBS bucket in advance. For details, see [Uploading a File](#).

Step 5 Set the target version and choose the environment to be upgraded.

Step 6 Click **Confirm**.

----End

Deleting Components

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 Click the **Component List** tab.

- Delete a single component.
Select the component to be deleted and click **Delete** in the **Operation** column. In the displayed dialog box, click **OK**.
- Delete components in batches.
Select the components to be deleted and click **Delete Component**. In the displayed dialog box, click **OK**.

----End

Creating a Pipeline for a Component

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 On the **Component List** page, click a component name to go to the **Overview** page.

Step 4 Choose **Pipeline > Create Pipeline** to create a pipeline. For details, see [Managing Pipelines](#).

 **NOTE**

- Pipelines cannot be created for component instances deployed on VMs.
- Pipelines cannot be created for components whose runtime system is **Docker**.

----End

Viewing Application Component Building

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 On the **Component List** page, click a component name to go to the **Overview** page.

Step 4 Click the **Build Job** tab to view the status of the application component building project. For details, see [Creating a Source Code Build Task](#).

----End

Maintaining Component Instances

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 On the **Environment View** tab, select an environment.

- You can view the deployment of the application component in each environment.
- (Optional) Select an application component version whose type is **Microservice** and click **Console** to go to the microservice console for service governance. For details, see [Microservice Governance](#).
- Select an application component version and click **Perform O&M**. The **Overview** page is displayed, where you can view the component instance details.
- Select an application component version and click **Operation**. You can perform O&M operations such as component upgrade, scaling, event viewing, start/stop, restart, rollback, and deletion. For details, see [Application O&M](#).
- Select **All** or the corresponding application component, and click **Upgrade Component** to change the version number, software package, or image package of the component.

----End

Managing Component Instance Access Mode

Step 1 Log in to ServiceStage and choose **Application Management >Application List**.

Step 2 Click the name of the created application. The **Overview** page is displayed.

Step 3 On the **Component List** tab, click the name of the created component to go to the **Overview** page.

You can view the component version on the card of the corresponding environment.

Step 4 Select a component whose status is **Running** and click **Set**. On the **Access Mode** page that is displayed, click **Add Service**.

Step 5 Set the following parameters. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Service Name	Sets the name of the service to be accessed.
Access Mode	Sets the service access mode. The options are as follows: <ul style="list-style-type: none">• Intra-cluster access: allows access from other services in the same cluster over TCP/UDP.• Intra-VPC access: allows access from other services in the same VPC over TCP/UDP.• Public network access: allows access from the Internet over TCP/UDP, including EIP.
Intra-VPC load balancing	This parameter is valid when Access Mode is set to Intra-VPC access .
* Access Type	<ul style="list-style-type: none">• This parameter is valid when Access Mode is set to Intra-VPC access and Intra-VPC load balancing is enabled.• This parameter is valid when Access Mode is set to Public network access.
Service Affinity	This parameter is valid when Access Mode is set to Intra-VPC access or Public network access .
Port Mapping	Sets Protocol , Container Port , and Access Port for accessing the service.

Step 6 Click **OK**.

----End

3.5 Performing Advanced Settings for an Application

3.5.1 Setting Application Environment Variables

Environment variables are set in the container running environment and can be modified after application component deployment, ensuring the flexibility of applications.

This section describes how to configure application environment variables for deployment using CCE and a VM.

CCE

If **Cloud Container Engine** is selected for **Deployment System** on the **Configure Basic Settings** page during component deployment, add environment variables by referring to the following steps.

Step 1 On the **Configure Component** page, choose **Advanced Settings > Component Configuration**.

Step 2 Add environment variables by referring to [Table 3-5](#).

Currently, environment variables can be added using any of the following methods:

Table 3-5 Environment variable types

Environment Variable Type	Procedure
Add manually	<ol style="list-style-type: none"> 1. Click Add Environment Variable and select Add manually. 2. Set Name and Variable/Variable Reference to add an environment variable.
Add from secret	<ol style="list-style-type: none"> 1. Create a secret. For details, see Creating a Secret. 2. Click Add Environment Variable and select Add from secret. 3. Set Name. 4. Select a secret from the Variable/Variable Reference drop-down list.
Add from ConfigMap	<ol style="list-style-type: none"> 1. Create a ConfigMap. For details, see Creating a ConfigMap. 2. Click Add Environment Variable and select Add from ConfigMap. 3. Enter Variable Name. 4. Select a ConfigMap from the Variable/Variable Reference drop-down list.
Import	<p>Click Import and select a local configuration file. The imported file must be a key-value pair mapping file in JSON or YAML format. For example:</p> <pre>{"key1":"value1","key2":"value2"}</pre>

----End

VM

If **VM** is selected for **Deployment System** on the **Configure Basic Settings** page during component deployment, add environment variables by referring to the following steps.

Step 1 On the **Configure Component** page, click **Add Environment Variable**.

Step 2 Set **Key** and **Value**, and click **OK**.

----End

3.5.2 Configuring the Lifecycle of an Application

If **Cloud Container Engine** or **Cloud Container Instance** is selected for **Deployment System** on the **Configure Basic Settings** page during component deployment, ServiceStage provides callback functions that can be invoked in specific phases of the application lifecycle. For example, if an operation needs to be performed on an application component before the component is stopped, you can register the corresponding hook function.

ServiceStage provides the following lifecycle callback functions:

- Startup command: used to start a container.
- Post-start processing: triggered after an application is started.
- Pre-stop processing: triggered before an application is stopped.

Procedure

Step 1 When deploying an application component, click **Advanced Settings > Deployment Configuration** on the **Configure Component** page.

Step 2 Click **Startup Command** to set **Command** and **Parameter** for the container .

A Docker image has metadata that stores image information. If no **Lifecycle** command or parameter is set, the container runs the default command and parameter provided during image creation. The Docker defines the default command and parameter as **CMD** and **Entrypoint**. For details about the two fields, see [Entrypoint Description](#) and [CMD Description](#).

If the running command and parameter of the application are set during application component deployment, the default **Entrypoint** and **CMD** will be overwritten during image building. [Table 3-6](#) describes the rules.

Table 3-6 Startup command parameters

Image Entrypoint	Image CMD	Application Running Command	Application Running Parameter	Final Execution
[touch]	[/root/test]	Not set	Not set	[touch /root/test]
[touch]	[/root/test]	[mkdir]	Not set	[mkdir]
[touch]	[/root/test]	Not set	[/opt/test]	[touch /opt/test]
[touch]	[/root/test]	[mkdir]	[/opt/test]	[mkdir /opt/test]

Step 3 Click **Lifecycle** and set **Post-start Processing** and **Pre-stop Processing**. [Table 3-7](#) describes the parameters.

Table 3-7 Container lifecycle parameters

Parameter	Description
CLI Mode	<p>Command to be executed in the component instance. The command format is Command <i>Args[1] Args[2]...</i> Command is a system command or a user-defined executable program. If no path is specified, an executable program in the default path will be selected. If multiple commands need to be executed, write the commands into a script for execution.</p> <p>For example, the following commands need to be executed:</p> <pre>exec: command: - /install.sh - install_agent</pre> <p>Write /install.sh install_agent in the script.</p> <p>This command indicates that the agent will be installed after the component is deployed.</p>
HTTP Request Mode	<p>HTTP call request. The related parameters are described as follows:</p> <ul style="list-style-type: none">● Path: (optional) URL of a request.● Port: (mandatory) request port.● Host Address: (optional) IP address of the request. The default value is the IP address of the node where the application resides.

----End

3.5.3 Configuring Data Storage

Container storage is a component that provides storage for applications. Multiple types of storage are supported. An application component can use any amount of storage.

If **Cloud Container Engine** is selected for **Deployment System** on the **Configure Basic Settings** page during component deployment, you can set data storage.

Scenario

Table 3-8 Storage scenarios

Storage Type	Scenario
EVS disks	<p>EVS supports three specifications: common I/O, high I/O, and ultra-high I/O.</p> <ul style="list-style-type: none">• Common I/O: The backend storage is provided by the Serial Advanced Technology Attachment (SATA) storage media. Common I/O is applicable to scenarios where large capacity is needed but high read/write rate is not required, and the volume of transactions is low. Examples include development testing and enterprise office applications.• High I/O: The backend storage is provided by the Serial Attached SCSI (SAS) storage media. High I/O is applicable to scenarios where relatively high performance, high read/write rate, and real-time data storage are required. Examples include creating file systems and sharing distributed files.• Ultra-high I/O: The backend storage is provided by the Solid-State Drive (SSD) storage media. Ultra-high I/O is applicable to scenarios where high performance, high read/write rate, and data-intensive applications are required. Examples include NoSQL, relational database, and data warehouse (such as Oracle RAC and SAP HANA).
SFS file systems	<p>Scalable File Service (SFS) file systems apply to a wide range of scenarios, including media processing, content management, big data, and analysis applications.</p>
OBS buckets	<ul style="list-style-type: none">• Standard OBS buckets: This type of OBS buckets applies to scenarios where a large number of hotspot files or small-sized files need to be accessed frequently (multiple times per month on average) and data can be quickly obtained. For example, cloud applications, data analysis, content analysis, and hotspot objects.• Infrequent access OBS buckets: This type of OBS buckets applies to scenarios where data is not frequently accessed (less than 12 times per year on average) but fast access response is required. For example, static website hosting, backup/active archiving, storage resource pools or backup storage for cloud services.

Storage Type	Scenario
HostPath	<p>The file directory of the host where the application component is located is mounted to the specified mounting point of the application. If the application component needs to access /etc/hosts, use HostPath to map /etc/hosts.</p> <p>NOTICE Do not mount the file directory to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect the application component instance startup. Otherwise, the file will be replaced, causing application component instance startup exceptions.</p>
EmptyDir	Used for temporary storage. The lifecycle of temporary storage is the same as that of an application component instance. When an application instance disappears, EmptyDir will be deleted and the data is permanently lost.
ConfigMap	Keys in a ConfigMap are mapped to an application so that configuration files can be mounted to the specified application component directory.
Secret	Sensitive information such as application authentication and application keys is stored in a secret, and the secret is mounted to a specified path of the application component.

EVS Disks

- Step 1** When deploying an application component, click **Advanced Settings > Deployment Configuration** on the **Configure Component** page.
- Step 2** Choose **Data Storage > Cloud Storage > Add Cloud Storage** and set parameters by referring to **Table 3-9**.

Table 3-9 EVS disks

Parameter	Description
Storage Type	<p>Select EVS disk.</p> <p>The method of using an EVS disk is the same as that of using a traditional disk. However, EVS disks have higher data reliability and I/O throughput and are easier to use. They apply to file systems, databases, or other system software or workloads that require block storage devices.</p>

Parameter	Description
Storage Allocation Mode	<ul style="list-style-type: none"> • Manual Select a created storage. You need to create a storage in advance. For details, see Using EVS Volumes. • Automatic A storage is created automatically. You need to enter the storage capacity. <ol style="list-style-type: none"> 1. If Storage Class is set to EVS Disk, select an AZ for creating the EVS disk first. 2. Select a storage sub-type. High I/O: EVS disks that have high I/O and use SAS. Common I/O: EVS disks that use SATA. Ultra-high I/O: EVS disks that have ultra-high I/O and use SSD. 3. Enter the storage capacity, in the unit of GB. Ensure that the storage capacity quota is not exceeded; otherwise, creation will fail.
Add Docker Mounting	<ol style="list-style-type: none"> 1. Set Mounting Path: Enter the application path to which the data volume is mounted. NOTICE <ul style="list-style-type: none"> - Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the file will be replaced, causing application startup exceptions. As a result, the application fails to be created. - When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the application; otherwise, high-risk files on the host may be damaged. 2. Set Permission. <ul style="list-style-type: none"> - Read-only: allows you only to read data volumes in the application path. - Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

SFS File Systems

Step 1 When deploying an application component, click **Advanced Settings > Deployment Configuration** on the **Configure Component** page.

Step 2 Choose **Data Storage > Cloud Storage > Add Cloud Storage** and set parameters by referring to [Table 3-10](#).

Table 3-10 SFS file systems

Parameter	Description
Storage Type	Select SFS . Scalable File Service (SFS) applies to a wide range of scenarios, including media processing, content management, big data, and analysis application.
Storage Allocation Mode	<ul style="list-style-type: none"> ● Manual Select a created storage. You need to create a storage in advance. For details, see Using SFS Volumes. ● Automatic A storage is created automatically. You need to enter the storage capacity. <ol style="list-style-type: none"> 1. Select a storage sub-type. Set the sub-type to NFS. 2. Enter the storage capacity, in the unit of GB. Ensure that the storage capacity quota is not exceeded; otherwise, creation will fail.
Add Docker Mounting	<ol style="list-style-type: none"> 1. Set Sub-path and Container Path to the path to which the data volume is mounted. NOTICE <ul style="list-style-type: none"> - Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the files will be replaced, causing application startup exceptions. As a result, the application fails to be created. - When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the container; otherwise, high-risk files on the host may be damaged. 2. Set Permission. <ul style="list-style-type: none"> - Read-only: allows you only to read data volumes in the application path. - Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

OBS Buckets

Step 1 When deploying an application component, click **Advanced Settings > Deployment Configuration** on the **Configure Component** page.

Step 2 Choose **Data Storage > Cloud Storage > Add Cloud Storage** and set parameters by referring to [Table 3-11](#).

Table 3-11 OBS buckets

Parameter	Description
Storage Type	Select OBS . Standard and Infrequent Access OBS classes are supported. OBS buckets apply to scenarios such as big data analytics, cloud native application data, static website hosting, and backup/active archiving.
Storage Allocation Mode	<ul style="list-style-type: none">• Manual Select a created storage. You need to create a storage in advance. For details, see Using OBS Volumes.• Automatic Select a storage sub-type. The sub-type can be set to standard OBS buckets or infrequent access OBS buckets.
Add Docker Mounting	<ol style="list-style-type: none">1. Set Mounting Path: Enter the application path to which the data volume is mounted. NOTICE<ul style="list-style-type: none">- Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the file will be replaced, causing application startup exceptions. As a result, the application fails to be created.- When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the application; otherwise, high-risk files on the host may be damaged.2. Set Permission.<ul style="list-style-type: none">- Read-only: allows you only to read data volumes in the application path.- Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

HostPath

The file or directory of the host is mounted to the application component.

Step 1 When deploying an application component, click **Advanced Settings > Deployment Configuration** on the **Configure Component** page.

Step 2 Choose **Data Storage > Local Disk > Add Local Disk** and set parameters by referring to [Table 3-12](#).

Table 3-12 HostPath

Parameter	Description
Local Disk Type	Select HostPath .
Host Path	Enter the host path, for example, /etc/hosts .
Docker Mounting	<p>1. Set Mounting Path: Enter the application path to which the data volume is mounted.</p> <p>NOTICE</p> <ul style="list-style-type: none"> - Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the file will be replaced, causing application startup exceptions. As a result, the application fails to be created. - When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the application; otherwise, high-risk files on the host may be damaged. <p>2. Set Permission.</p> <ul style="list-style-type: none"> - Read-only: allows you only to read data volumes in the application path. - Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

EmptyDir

EmptyDir applies to temporary data storage, disaster recovery, and shared running. It will be deleted upon deletion or transfer of application component instances.

Step 1 When deploying an application component, click **Advanced Settings > Deployment Configuration** on the **Configure Component** page.

Step 2 Choose **Data Storage > Local Disk > Add Local Disk** and set parameters by referring to [Table 3-13](#).

Table 3-13 EmptyDir

Parameter	Description
Local Disk Type	Select EmptyDir .

Parameter	Description
Disk Media	<ul style="list-style-type: none"> If you select Memory, the running speed is improved, but the storage capacity is limited by the memory size. This mode applies to a small amount of data with high requirements on reading and writing efficiency. When Memory is not selected, data is stored in disks, which is applicable to a large amount of data with low requirements on reading and writing efficiency.
Docker Mounting	<ol style="list-style-type: none"> Set Mounting Path: Enter the application path to which the data volume is mounted. NOTICE <ul style="list-style-type: none"> Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the file will be replaced, causing application startup exceptions. As a result, the application fails to be created. When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the application; otherwise, high-risk files on the host may be damaged. Set Permission. <ul style="list-style-type: none"> Read-only: allows you only to read data volumes in the application path. Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

ConfigMap

ServiceStage separates the application codes from configuration files. **ConfigMap** is used to process application component configuration parameters.

Step 1 When deploying an application component, expand **Advanced Settings** > **Component Configurations** on the **Configure Component** page.

Step 2 Choose **Data Storage** > **Local Disk** > **Add Local Disk** and set parameters by referring to [Table 3-14](#).

Table 3-14 ConfigMap

Parameter	Description
Local Disk Type	Select ConfigMap .

Parameter	Description
Configuration Item	Select the desired ConfigMap name. Create a ConfigMap. For details, see Creating a ConfigMap .
Docker Mounting	<ol style="list-style-type: none"> Set Mounting Path: Enter the application path to which the data volume is mounted. <p>NOTICE</p> <ul style="list-style-type: none"> Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the file will be replaced, causing application startup exceptions. As a result, the application fails to be created. When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the application; otherwise, high-risk files on the host may be damaged. <ol style="list-style-type: none"> Set Permission. <ul style="list-style-type: none"> Read-only: allows you only to read data volumes in the application path. Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

Secret

The data in the secret is mounted to the specified application component. The content of the secret is user-defined.

Step 1 When deploying an application component, expand **Advanced Settings > Component Configurations** on the **Configure Component** page.

Step 2 Choose **Data Storage > Local Disk > Add Local Disk** and set parameters by referring to [Table 3-15](#).

Table 3-15 Secret

Parameter	Description
Local Disk Type	Select Secret .
Secret Item	Select the desired secret name. Create a secret. For details, see Creating a Secret .

Parameter	Description
Docker Mounting	<ol style="list-style-type: none">1. Set Mounting Path: Enter the application path to which the data volume is mounted. NOTICE<ul style="list-style-type: none">- Do not mount a data volume to a system directory such as / or /var/run. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the file will be replaced, causing application startup exceptions. As a result, the application fails to be created.- When the data volume is mounted to a high-risk directory, you are advised to use a low-permission account to start the application; otherwise, high-risk files on the host may be damaged.2. Set Permission.<ul style="list-style-type: none">- Read-only: allows you only to read data volumes in the application path.- Read/Write: allows you to modify the data volumes in the application path. To prevent data loss, newly written data will not be migrated during application migration.

Step 3 Click **OK**.

----End

3.5.4 Configuring Distributed Sessions

Traditional single-instance applications use local session management. Session contexts generated by user requests are stored in the process memory. After the load balancing module is added, multi-instance sessions need to be shared using distributed storage.

ServiceStage provides the out-of-the-box distributed session function. It uses the [Distributed Cache Service](#) as the session persistence layer. Without code modification, ServiceStage supports distributed session management for Tomcat applications, Node.js applications that use express-session, and PHP applications that use session handle.

During component deployment, you can bind distributed sessions in **Database**. The procedure is as follows: After the binding is complete, query environment variables during application running to obtain the distributed cache information. For details about the environment variables, see [Configuration Information About the Middleware Bound to Applications](#).

Prerequisites

Before setting a distributed session, you need to create a distributed session. For details, see [Buying a DCS Redis Instance](#).

Procedure

Step 1 During component deployment, select **Distributed session** on the **Configure Component** page.

- Step 2** Click **Add Distributed Session** and set distributed session parameters.
- Step 3** Select **New Creation** for **Configuration Mode** to create a distributed session.
In some regions, only **Use existing** can be selected.
- Step 4** Set **Distributed Cache Service**, **Subnet**, and **Password**.
- Step 5** Retain the default values for other parameters. For details, see [Buying a DCS Redis Instance](#).
- Step 6** Click **OK**.
- End

3.5.5 Configuring Relational Databases

To store application data permanently, you need to use Relational Database Service (RDS). Based on the cloud computing platform, ServiceStage provides RDS for MySQL which is reliable, scalable, easy to manage, and ready for use. [RDS for MySQL](#) enables you to easily set and scale relational databases on the cloud. Using the RDS service, you can perform nearly all necessary tasks without programming. This service simplifies operation procedures and reduces routine O&M workloads, so that you can focus on application and service development.

When deploying an application component, you can bind a relational database in the database settings.

Prerequisites

Before configuring an RDS database instance, you need to create an RDS database instance. For details, see [Buy a DB Instance](#).

Procedure

- Step 1** During component deployment, select **RDS DB instance** on the **Configure Component** page.
- Step 2** Click **Add RDS DB Instance** and set relational database parameters.
- Step 3** Select the **New creation** configuration method.
Some regions may not support **New Creation**. In this case, you can only select **Use existing**.
- Step 4** Enter a database instance name.
- Step 5** Set the subnet.
- Step 6** Select a connection type.
- **JNDI**: standard Java connection mode.
 - **Spring Cloud Connector**: Spring connection mode.
- Step 7** Enter a database name.
- Step 8** Enter and confirm the new administrator password.
- Step 9** Enter a database account.

Step 10 Retain the default values for other parameters. For details, see [Buy a DB Instance](#).

Step 11 Click **OK**.

----End

3.6 Building an Application Component

ServiceStage needs to build a software package into an image and then deploy it. Therefore, after you set the application source when creating an application component, ServiceStage generates a build job for the application component.

NOTE

Components whose runtime system is **Docker** cannot be built.

Viewing Application Component Building

Step 1 Log in to ServiceStage, choose **Application Management > Application List**, and click the application name to go to the **Overview** page.

Step 2 On the **Component List** page, click a component name to go to the **Overview** page.

Step 3 Click the **Build Job** tab to view the status of the application component build job.

----End

Maintaining a Build Job

Table 3-16 Maintenance

Operation	Operation Description
Build Now	Click Build Now to start a build job.
Query details/build history	<ul style="list-style-type: none">Click View Other Build Records and view the build history.Click Logs to view the build log.Click Code Check to view the code check overview and details. Currently, the following code check plug-ins are supported: Checkstyle, FindBugs, and PMD. <p>NOTE Only the Maven build project supports code check.</p>

3.7 Pipelining an Application Component

One-click deployment can be achieved through pipeline. In this way, the entire process of source code pull, compilation, packaging, archiving, and deployment is automatically implemented. This unifies the integration environment and standardizes the delivery process.

In the new pipeline, the "phase/task" model is optimized to the "build/environment" model. Each pipeline includes a group of build tasks and one or more groups of environment (such as development environment, production-like environment, and production environment) tasks, each group of environment tasks contains one or more subtasks (such as deployment and test tasks) and provides templates.

 **NOTE**

- Pipelines cannot be created for instances deployed on VMs.
- Pipelines cannot be created for components whose runtime system is **Docker**.

Creating a Pipeline

Step 1 Log in to ServiceStage, choose **Application Management > Application List**, and click the application name to go to the **Overview** page.

Step 2 On the **Component List** page, click a component name to go to the **Overview** page.


Step 3 Click the **Pipeline** tab and click **Create Pipeline**.

Step 4 Enter the basic pipeline information.

1. Enter a pipeline name.
2. (Optional) Enter the **Description**.

Step 5 Set pipeline.

1. Add a build task.
The build task of the component is automatically loaded.
2. Add a deploy task.
Click **Add Environment**. The deployed components are automatically loaded.
3. Set pipeline approval.

Click  in the environment area to set the approval mode and approver.

- **Approval Mode: By all** and **By one person** are now supported.
- **Approved By:** You can select multiple accounts as approvers. The system automatically loads all subaccounts of the account.

Step 6 Click **Create and Start** to start the pipeline.

Click **Create** to save the settings and do not execute the pipeline.

----End

Related Operations

After the pipeline is started, you can build and upgrade applications in one-click mode. For details about maintenance after application components are upgraded, see [Application O&M](#).

3.8 Application Configuration

3.8.1 Creating a Secret

Secrets are user-defined resources that store authentication and sensitive information such as application keys. They can be used as files or environment variables in applications.

Prerequisites

- You have created a cluster that requires a secret.
 - For details about how to create a hybrid cluster, see [Buying a Hybrid Cluster](#).
 - For details about how to create a BMS cluster, see [Create a BMS](#).
- You have created a namespace for the secret.

Creating a Secret

Step 1 Log in to ServiceStage and choose **Application Management > Application Configuration > Secret**

Step 2 Click **Create**.

Step 3 Create a secret by **Visualization** or **YAML**.

- Method 1: **Visualization**. On the displayed page, set the parameters listed in the following table. Parameters marked with an asterisk (*) are mandatory.

Table 3-17 Parameters for creating a secret

Parameter	Description
Basic Information	
*Name	Name of the secret, which must be unique in the same namespace.
*Cluster	Cluster where the secret will be used. Click Create Cluster to create a cluster.
*Namespaces	Namespace to which the secret belongs. The default value is default .
Description	Description of the secret. Click Create Namespace to create a namespace.

Parameter	Description
*Secret Type	<p>Select the type of the secret to be created based on service requirements.</p> <ul style="list-style-type: none"> - Opaque: general secret type. If the secret type is not explicitly set in the secret configuration file, the default secret type is Opaque. - kubernetes.io/dockerconfigjson: a secret that stores the authentication information required for pulling images from a private repository. - IngressTLS: a secret that stores the certificate required by ingresses (layer-7 load balancing services). - Other: Enter a secret type that is none of the above.
*Repository Address	<p>This parameter is valid only when Secret Type is set to kubernetes.io/dockerconfigjson. Enter the address of the image repository.</p>
Secret Data	<p>Value of the data field in the application secret file.</p> <ul style="list-style-type: none"> - If the secret type is Opaque, enter the key and value. The value must be encoded using Base64. For more information, see Base64 Encoding. Click Add Data to add secret data. - If the secret type is kubernetes.io/dockerconfigjson, enter the image repository address, username, and password. - If the secret type is IngressTLS, upload the certificate file and private key file. - If the secret type is Other, enter the secret type, key, and value.
Secret Label	<p>Labels are attached to objects, such as applications, nodes, and services, in the form of key-value pairs. Labels define the identifiable attributes of these objects and are used to manage and select the objects.</p> <ol style="list-style-type: none"> 1. Click Add Label. 2. Set keys and values.

- Method 2: **YAML**.

 **NOTE**

To create a secret by uploading a file, ensure that a resource description file has been created. ServiceStage supports resource description files in YAML format. For more information, see [Secret Resource File Configuration](#).

- a. Select a cluster from the **Cluster** drop-down list.
- b. (Optional) Click **Upload File**, select the created secret file, and then click **Open**.
Upload a file whose size is less than 2 MB.

- c. Write or modify the secret file in **Orchestration content**.

Step 4 After the configuration is complete, click **Create**.

The new secret is displayed in the secret list.

----End

Secret Resource File Configuration

This section provides examples of configuring secret resource description files.

For example, you can retrieve the username and password for an application through a secret.

username: my-username

password: my-password

The content in the secret file **secret.yaml** is as follows. The value must be encoded using Base64. For more information, see [Base64 Encoding](#).

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret      #Secret name.
  namespace: default #Namespace. The default value is default.
data:
  username: ***** #The value must be encoded using Base64.
  password: ***** #The value must be encoded using Base64.
type: Opaque #You are advised not to change this parameter value.
```

Base64 Encoding

To encrypt a string using Base64, run the **echo -n 'Content to be encoded' | base64** command in the local Linux environment. Example:

```
root@ubuntu:~# echo -n '3306' | base64
MzMwNg==
```

In the preceding example, **3306** is the content to be encoded.

Managing Secrets

Operation	Description
Modifying a secret	<ol style="list-style-type: none"> 1. Click Modify in the Operation column of the secret to be modified. 2. Modify the secret information according to Table 3-17. 3. Click Modify Secret.
Deleting a secret	Click Delete in the Operation column of the secret to be deleted, and follow the system prompts to delete the secret.
Deleting secrets in batches	<ol style="list-style-type: none"> 1. Select the secrets to be deleted. 2. Click Delete in the upper left of the page, and follow the system prompts to delete the secrets.

Operation	Description
Viewing a secret	Click Show YAML in the Operation column of the target secret to view the content of the YAML file of the secret.

 **NOTE**

The secret list contains system secrets, which can only be viewed and cannot be modified or deleted.

3.8.2 Creating a ConfigMap

ConfigMaps store user-defined application configurations. They can be used as files or environment variables in applications.

Scenarios

ConfigMaps allow you to decouple configuration files from images to enhance the portability of applications.

Benefits of ConfigMaps:

- Manage configurations of different environments and services.
- Deploy applications in different environments. You can maintain configuration files in multiple versions, which makes it easy to update and roll back applications.
- Quickly import configurations in the form of files to containers.

Prerequisites

- You have created a cluster that requires a secret.
 - For details about how to create a hybrid cluster, see [Buying a Hybrid Cluster](#).
 - For details about how to create a BMS cluster, see [Create a BMS](#).
- You have created a namespace for the secret.

Creating a ConfigMap

Step 1 Log in to ServiceStage and choose **Application Management > Application Configuration > ConfigMap**

Step 2 Click **Create**.

Step 3 Create a ConfigMap by **Visualization** or **YAML**.

- Method 1: **Visualization**. On the displayed page, set the parameters listed in the following table. Parameters marked with an asterisk (*) are mandatory.

Table 3-18 Parameters for creating a ConfigMap

Parameter	Description
Basic Information	
* Configuration Name	Name of a ConfigMap, which must be unique in a namespace.
* Cluster	Cluster where the ConfigMap will be used.
* Namespace	Namespace to which the ConfigMap belongs. If you do not specify this parameter, the value default is used by default.
Description	Description of the ConfigMap.
Configuration Data	Used in applications or used to store configuration data. Key is a file name, and Value is the content of the file. 1. Click Add Data . 2. Set keys and values.
Configuration Labels	Labels are attached to objects, such as applications, nodes, and services, in the form of key-value pairs. Labels define the identifiable attributes of these objects and are used to manage and select the objects. 1. Click Add Label . 2. Set keys and values.

- Method 2: **YAML**.

 **NOTE**

To create ConfigMaps by uploading a file, ensure that the resource description file has been created. ServiceStage supports resource description files in YAML format. For details, see [Configuration Item Requirements](#).

- Select a cluster from the **Cluster** drop-down list.
- (Optional) Click **Upload File**, select the created ConfigMap file, and then click **Open**.
Upload a file whose size is less than 2 MB.
- Write or modify the ConfigMap file in **Orchestration content**.

Step 4 After the configuration is complete, click **Create**.

The new configuration item is displayed in the configuration item list.

----End

Configuration Item Requirements

A ConfigMap resource file should be in YAML format, and the file size cannot exceed 2 MB.

The following shows an example of a ConfigMap resource file named **configmap.yaml**:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: test-configmap
data:
  data-1: value-1
  data-2: value-2
```

Managing Configuration Items

Operation	Description
Modifying a configuration item	<ol style="list-style-type: none"> 1. Click Modify in the Operation of the configuration item to be modified. 2. Modify the configuration item information according to Table 3-18. 3. Click Modify.
Deleting a configuration item	Click Delete in the Operation column of the configuration item to be deleted, and follow the system prompts to delete this configuration item.
Deleting configuration items in batches	<ol style="list-style-type: none"> 1. Select the configuration items to be deleted. 2. Click Delete in the upper left of the page, and follow the system prompts to delete the configuration items.
Viewing a configuration item	Click Show YAML in the Operation column of the target configuration item to view the content of the YAML file of the configuration item.

NOTE

The configuration item list contains system configuration items, which can only be viewed and cannot be modified or deleted.

4 Environment Management

Environment is a collection of infrastructures, covering computing, storage, and networks, used for application deployment and running. ServiceStage combines basic resources (such as CCE and ECS) and optional resources (such as ELB, RDS, and DCS) in the same VPC into an environment, such as a development environment, testing environment, pre-production environment, and production environment. Networks in an environment can communicate with each other. You can manage resources and deploy services by environment, simplifying infrastructure O&M.

Creating an Environment

Step 1 Log in to ServiceStage, choose **Environment Management**, and click **Create Environment**.

Step 2 Set basic parameters. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Environment	Environment name, for example, dev-bate .
Description	Environment description.
*VPC	Select the VPC to which the infrastructure belongs. For details about how to create a VPC, see Creating a VPC . NOTE After a VPC is selected, infrastructure resources in the VPC are loaded for selection. Resources that are not in the VPC cannot be selected.
*Basic Resource	Select at least one of the following infrastructures: Cloud Container Engine (CCE), Elastic Cloud Server (ECS), and Auto Scaling (AS). You can select multiple infrastructures.
Optional Resource	You can select Elastic Load Balance (ELB), Elastic IP (EIP), Distributed Cache Service (DCS), Relational Database Service (RDS), or Cloud Service Engine (CSE) as required.

Step 3 Click **Create Now**.

After the environment is created, you can view the environment information on the **Environment Management** page.

----End

Modifying an Environment

Step 1 Log in to ServiceStage, choose **Environment Management**, and click **Edit** on an existing environment.

Step 2 Set basic parameters. Parameters marked with an asterisk (*) are mandatory.

Parameter	Description
*Environment	You can change the environment name.
Description	Environment description.
*VPC	The VPC cannot be modified. You can only add infrastructure resources in the selected VPC. Resources that are not in the VPC cannot be selected.
*Basic Resource	You can add or delete basic resources, including Cloud Container Engine (CCE), Auto Scaling (AS), and Elastic Cloud Server (ECS).
Optional Resource	You can add or delete optional resources, including Elastic Load Balance (ELB), Elastic IP (EIP), Distributed Cache Service (DCS), Relational Database Service (RDS), or Cloud Service Engine (CSE) as required.

Step 3 Click **Save**.

After the environment is modified, you can view the environment information on the **Environment Management** page.

----End

Deleting an Environment

NOTE

- Before deleting an environment, ensure that no application component is deployed in the environment or the deployed application components have been deleted. For details, see [Managing Application Components](#).
- Deleting an environment does not delete resources in the environment.

Step 1 Log in to ServiceStage, choose **Environment Management**, and click **Delete** on an existing environment.

Step 2 In the dialog box that is displayed, click **OK**.

----End

5 Application O&M

- [Maintaining Application Component Instances](#)
- [Adding Labels for Application Component Instances](#)
- [Configuring Domain Name Mappings](#)
- [Setting Alarm Thresholds for Resource Monitoring](#)
- [Setting Scaling Policies for Application Component Instances](#)
- [Setting Scheduling Policies for Application Component Instances](#)
- [Setting Upgrade Policies for Application Component Instances](#)
- [Setting Custom Metric Monitoring for Application Components](#)
- [Configuring Application Log Policies](#)
- [Configuring Application Performance Management](#)
- [Configuring Health Check](#)

5.1 Maintaining Application Component Instances

Scaling Application Components

You can define auto-scaling policies as required, releasing you from the workload of repeatedly adjusting resources in response to service changes and heavy burden during peak hours and saving resource and labor costs. For details, see [Setting Scaling Policies for Application Component Instances](#).

Starting and Stopping an Application Component

After an application component is deployed, you can start or stop it as required.

Step 1 Log in to ServiceStage and choose **Application Management** > **Application List** to view all applications.

Step 2 Click an application name. The **Overview** page is displayed.

- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Start or stop a component.
- Click **Stop** in the **Operation** column to stop an application component in the **Running** or **Not ready** state.
 - Click **Start** in the **Operation** column to start an application component in the **Stopped** state.
 - Click **Restart** in the **Operation** column to restart an application component in the **Running** or **Not ready** state.
- End

Upgrading an Application Component

After an application component is deployed, you can re-deploy software packages and modify component configurations as required.

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Click **Upgrade** in the **Operation** column to upgrade an application component.
- Step 5** Update the configuration and version based on service requirements.

NOTICE

The component can be rolled back to the source version only if the component version has been changed.

- Step 6** Configure advanced settings. For details, see [Performing Advanced Settings for an Application](#).
- Step 7** After the configuration is complete, click **Re-deployment**.
- Step 8** Click **OK** and wait until the component upgrade is complete.
- End

Viewing Instance Details

After an application component is deployed, you can view the overview, instance list, and access mode on the application component details page.

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

- Step 4** Click the name of an application component to access its details page.
----End

Querying Application Logs

After an application component starts, you can query application logs to learn about its running status.

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Click the name of an application component to access its details page.
- Step 5** Click **Logs** to view application component logs.
- Select an instance, log file name, and time granularity to view logs of the specified instance in a specified period.
 - Enter a keyword in the text box to search for logs.
- End

Rolling Back an Application Component

After an application component is upgraded, you can roll it back to its target version.

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Click **Roll Back** in the **Operation** column to roll back an application component.
- Step 5** Locate the target version and click **Roll back to this version** in the **Operation** column.
- End

Deleting an Application Component

NOTICE

Deleted application components cannot be restored. Exercise caution when performing this operation.

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Click **Delete** in the **Operation** column delete an application component.
Click **OK** as prompted.
- End

5.2 Adding Labels for Application Component Instances

Labels are attached to application components using key-value pairs. After creating labels for application components, you can manage and select application components by labels. You can add labels to multiple application components or a specific application component.

As shown in [Figure 5-1](#), three labels (**release**, **env**, and **role**) are defined for the application components APP 1, APP 2, and APP 3. Different label values are defined for different application components.

Figure 5-1 Label example



Procedure

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Click the name of an application component to access its details page.
- Step 5** Click **Manage Label**.
- Step 6** Click **Add Label**, set **Key** and **Value**, and click **Save**.

NOTE

- The key name must be unique.
- Labels cannot be added to application component instances that are abnormal or deployed on VMs.

----End

5.3 Configuring Domain Name Mappings

For application components that have Internet access enabled, you must define a domain name on ServiceStage and configure the domain name mapping in the domain name provider.

Prerequisites

- An automatically generated domain name is valid only for seven days. After the validity period expires, the domain name must be changed to a custom domain name.
- You can change the domain name only when the application is in the **Running** state.
- You have obtained the domain name from the domain name provider.
- You have obtained the elastic public IP address of the ELB bound to the application component.

Procedure

Step 1 Log in to ServiceStage and choose **Application Management > Application List** to view all applications.

Step 2 Click an application name. The **Overview** page is displayed.

Step 3 On the **Environment View** tab, select an environment to view the application components that have been deployed in the environment.

Step 4 Click the name of an application component to access its details page.

Step 5 Set a domain name.

1. Choose **Access Mode > Set domain**, and enter the obtained domain name.
2. Enable **HTTPS**.

If HTTPS is enabled, click **Use existing** to select an existing certificate.

If no certificate exists, click **New creation** to create a server certificate. For details about how to create a server certificate, see [Creating a Certificate](#).

Step 6 Configure domain name mapping in the domain name provider.

----End

5.4 Setting Alarm Thresholds for Resource Monitoring

When you need to monitor some resources and respond to exceptions in a timely manner, you can create threshold rules for metrics of these key resources in routine O&M so that you can find and handle exceptions in time.

- If the metric meets the threshold conditions within a specified period, the system sends a threshold alarm.
- If no metric is reported within a specified period, the system sends a data insufficiency event.

- If you cannot query the change information about the threshold rule status on the ServiceStage console due to non-business hours or business trips, you can enable the notification function to send the change information to related personnel through SMS messages or emails.

Procedure

- Step 1** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 2** Click an application name. The **Overview** page is displayed.
- Step 3** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 4** Click the name of an application component to access its details page.
- Step 5** Choose **Threshold Alarms > Set Threshold Rule** and set threshold rule parameters based on [Table 5-1](#). The parameters marked with * are mandatory.

Table 5-1 Threshold rule parameters

Parameter	Description
* Threshold Name	Name of the threshold rule to be added. The name must be unique and cannot be modified once specified.
Description	Description about the threshold rule.
Statistics Mode	Method used to measure metrics.
Statistical Cycle	Interval at which metric data is collected.
Metric	Select the metrics to be monitored.
* Threshold Condition	Trigger of a threshold alarm. A threshold condition consists of two parts: operators (\geq , \leq , $>$, and $<$) and threshold value. For example, if this parameter is set to ≥ 80 , the system generates a threshold alarm when the metric is greater than or equal to 80.
Continuous Cycle	When the metric meets the threshold condition for a specified number of consecutive periods, a threshold alarm will be generated.
Alarm Severity	Severity of the threshold alarm.
Send Notification	Whether to send notifications. <ul style="list-style-type: none">• If you select Yes (recommended), the system sends an email or SMS message to the user.• If you select No, the system does not send an email or SMS message to the user.


Step 6 Click **OK**.

----End

Managing Threshold Alarms

After creating a threshold rule, you can perform operations described in [Table 5-2](#).

Table 5-2 Related operations

Operation	Description
Modify a threshold alarm.	<p>When you find that the current threshold rule is not properly set, you can perform the following operations to modify the threshold rule to better meet your service requirements.</p> <ol style="list-style-type: none">1. Click Modify in the Operations column of the threshold alarm list.2. On the Modify Threshold Rule page, modify the parameters of the threshold rule as prompted.3. Click Modify.
Delete a threshold alarm.	<p>When you find that the current threshold rule is no longer needed, you can perform the following operations to delete the threshold rule to release more threshold rule resources.</p> <ol style="list-style-type: none">1. Delete one or multiple threshold rules.<ul style="list-style-type: none">• To delete a single threshold, click Delete in the Operations column of the threshold rule list.• To delete one or more threshold rules, select one or more threshold rules and click Delete on the upper part of the page.2. In the dialog box displayed, click OK.
Search for threshold alarms.	<ol style="list-style-type: none">1. Select a time segment from the drop-down list.2. Enter the keyword of the alarm name or description in the search box on the upper right corner of the page.3. Click  or press Enter. You can also click Advanced Search to set the search criteria and click Search to query.
View threshold-crossing alarms.	<p>If the metric meets the threshold conditions within a specified period, the system sends a threshold alarm. View the alarm in the threshold alarm list.</p>
Check the data insufficiency event.	<p>If no metric is reported within a specified period, the system sends a data insufficiency event. You can click the Event tab to view the event in the event list.</p>

5.5 Setting Scaling Policies for Application Component Instances

After scaling policies are set, instances can be automatically added or deleted based on resource changes or a specified schedule. This reduces manual resource adjustment to cope with service changes and service peak, helping you save resources and labor costs.

- Auto scaling: Metric-based, scheduled, and periodic policies are supported. After configuration, instances can be automatically added or deleted based on resource changes or a specified schedule.
To use auto scaling, you must be an APM administrator.
- Manual scaling: The number of instances is increased or decreased immediately after the configuration is complete.

Graceful Scaling-In

You can set a graceful scaling-in time window to save important data before an application component instance stops. The value ranges from 0 to 9999, in units of seconds. The default value is 30s. For example, if an application has two instances and only one instance will be kept after the scale-in operation, you can still perform certain operations on the instance to be stopped in the specified time window.

You can also set the maximum number of unavailable instances allowed during the rolling upgrade every day.




Step 1 Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

Step 2 Click an application name. The **Overview** page is displayed.

Step 3 On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

Step 4 Click the name of an application component to access its details page.

Step 5 Choose **Scaling**.

- Set **Graceful Time Window (s)**. Specifically, click , enter a value, and click .
- Set **Maximum number of unavailable instances**. Specifically, click , enter the maximum number (or select **percentage** and enter the maximum percentage), and click **Save**.

Click **View Component Details**. The instance is displayed in the **Upgrading/Rolling back the component** state. When the status changes to **Running**, the scaling is complete.

----End

Manual Scaling



Step 1 Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

Step 2 Click an application name. The **Overview** page is displayed.

Step 3 On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

Step 4 Click the name of an application component to access its details page.

Step 5 Choose **Scaling**. In the **Manual Scaling** area:

1. Click  and change the number of instances.
2. Click .

Click **View Component Details**. The instance is displayed in the **Upgrading/Rolling back the component** state. When the status changes to **Running**, the scaling is complete.

----End

Auto Scaling

NOTE

- CCE clusters of 1.15 or later do not support auto scaling.
- VM deployment does not support auto scaling.

Step 1 Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

Step 2 Click an application name. The **Overview** page is displayed.

Step 3 On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

Step 4 Click the name of an application component to access its details page. Choose **Scaling**.

Step 5 In the **Auto Scaling** area, click **Edit Scaling Rule**.

1. Set **Cooling Time**, **Maximum Instances**, and **Minimum Instances** based on service requirements.
2. Click **Save**.

Step 6 In the **Auto Scaling** area, click **Add Scaling Policy**.

Currently, ServiceStage supports the following types of auto scaling policies:

- **Alarm Policy**: scaling based on the CPU or memory settings. After an application component is deployed, instances in this application can be automatically scaled in or out when the number of CPU cores or memory amount exceeds or is less than a specified value.

Table 5-3 Parameters for adding a metric-based policy

Parameter	Description
Policy Name	Name of the scaling policy.
Policy Type	Set this parameter to Alarm Policy .
Metric	Select a metric. Metrics reflects the resource performance or status. <ul style="list-style-type: none"> - CPU usage of the measured object. This metric indicates the percentage of the CPU cores actually used by the measured object to the total CPU cores that the measured object has applied for. - Physical memory usage. This metric indicates the percentage of the physical memory size used by the measured object to the physical memory size that the measured object has applied for. - Disk read rate, which indicates the data volume read from the disk per second. - Physical memory size that the measured object has applied for. - Data receiving rate, which indicates the data volume received by the measured object per second. - Disk write rate, which indicates the data volume written into the disk per second. - Size of the physical memory used by the measured object. - Total number of CPU cores that the measured object has applied for. - Data sending rate, which indicates the data volume sent by the measured object per second. - Number of error packets received by the measured object. - Number of CPU cores used by the measured object.
Triggering Condition	Condition based on which the scaling policy is triggered.
Duration	Metric statistics period. For example, if the parameter is set to 20s, metric statistics is collected every 20s.
Continuous Cycle	Number of consecutive times that the threshold is triggered. For example, if the parameter is set to 3, the action is triggered if the threshold is reached for three consecutive measurement periods.
Action	Select Add or Reduce to set the action to be executed after the policy is triggered.

 NOTE

Click **Show/Hide Preview** to set **Triggering Condition**, **Duration**, **Continuous Cycle**, and **Action**.

- **Scheduled Policy:** Instances in an application can be automatically scaled in or out at a specified time. This policy is applicable to high traffic scenarios, such as flash sales and premier shopping events, where a large number of application instances need to be added.

Table 5-4 Parameters for adding a scheduled policy

Parameter	Description
Policy Name	Name of the scaling policy.
Policy Type	Set it to Scheduled Policy .
Trigger Time	Set the time at which the policy is enforced.
Action	Select Add , Reduce , or Set to set the action to be executed after the policy is triggered.

- **Periodic Policy:** Scaling policies can be executed daily, weekly, or monthly. This policy is applicable to scenarios where traffic changes periodically.

Table 5-5 Parameters for adding a periodic policy

Parameter	Description
Policy Name	Name of the scaling policy.
Policy Type	Set it to Periodic Policy .
Trigger Time	Set the time at which the policy is enforced.
Action	Select Add , Reduce , or Set to set the action to be executed after the policy is triggered.

Step 7 Click **OK**.

In the **Auto Scaling** area, check that the policy has been started. When the trigger is met, the auto scaling policy immediately takes effect.

----End

5.6 Setting Scheduling Policies for Application Component Instances

ServiceStage provides a variety of scheduling policies, including static global scheduling policies and dynamic runtime scheduling policies. You can select or combine these policies as required.

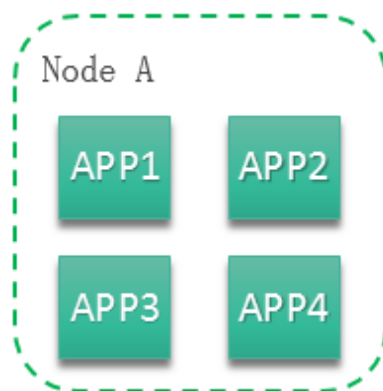
Concepts

- Application-AZ Affinity and Anti-Affinity
 - **Affinity with AZs:** Application components can be deployed in specific AZs.
 - **Non-affinity with AZs:** Application components cannot be deployed in specific AZs.
- Application-Node Affinity and Anti-Affinity
 - **Affinity with Nodes:** Application components can be deployed on specific nodes.
 - **Non-affinity with Nodes:** Application components cannot be deployed on specific nodes.
- Application Affinity

It determines whether application components are deployed on the same node or different nodes.

 - **Affinity with Applications:** Application components are deployed on the same node. You can deploy application components based on service requirements. The nearest route between application components is used to reduce network consumption. For example, [Figure 5-2](#) shows affinity deployment, in which all applications are deployed on the same node.

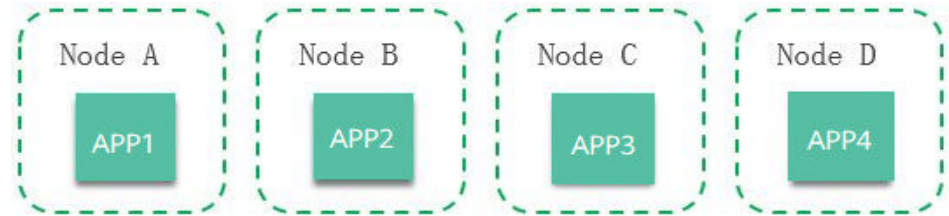
Figure 5-2 Application affinity



- **Anti-affinity with Applications:** Different applications or multiple instances of the same application component are deployed on different nodes. Anti-affinity deployment for multiple instances of the same application reduces the impact of system breakdowns. Anti-affinity deployment for applications can prevent interference between the applications.

As shown in [Figure 5-3](#), four applications are deployed on four different nodes. The four applications are deployed in anti-affinity mode.

Figure 5-3 Application anti-affinity



Precautions

When setting application component-node affinity and application component-application component affinity, ensure that the affinity relationships are not mutually exclusive; otherwise, application deployment will fail. For example, application deployment will fail when the following conditions are met:

- Anti-affinity is configured for two application components APP 1 and APP 2. For example, APP 1 is deployed on node A and APP 2 is deployed on node B.
- When APP 3 is deployed on node C and goes online, affinity is configured between APP 3 and APP 2. As a result, affinity relationships are mutually exclusive, and APP 3 fails to be deployed.

Procedure

When the component type is **Common** and the runtime system is **Docker**, perform the following operations:

Step 1 Access the page for setting a scheduling policy of an application component instance.

- To set a scheduling policy during **component configuration** in the application component deployment, go to [Step 2](#).
- To set a scheduling policy after an application component is deployed, go to [Step 3](#).

Step 2 On the **Configure Component** page:

1. Set the scheduling policy for the application component instance based on the following table.

Purpose	Procedure
Setting application component-AZ affinity	<ol style="list-style-type: none">1. Click Add an affinity object.2. Set the object type to Availability Zone, and select the desired AZ.3. Click OK.

Purpose	Procedure
Setting application component-AZ anti-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Availability Zone, and select the desired AZ. 3. Click OK.
Setting application component-node affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Node, and select the desired node. 3. Click OK.
Setting application component-node non-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Node, and select the desired node. 3. Click OK.
Setting application component-application component affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Component, and select the desired application components. 3. Click OK. The selected application components are deployed on the same node.
Setting application component-application component anti-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Component, and select the desired application components. 3. Click OK. The selected application components are deployed on different nodes.

2. Click **Next** to complete the component deployment.

Step 3 Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

Step 4 Click an application name. The **Overview** page is displayed.

Step 5 On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

Step 6 Click the name of an application component to access its details page.

Step 7 Choose **Scheduling Policy** and set the following parameters:

Purpose	Procedure
Setting application component-AZ affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Availability Zone, and select the desired AZ. 3. Click OK.

Purpose	Procedure
Setting application component-AZ anti-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Availability Zone, and select the desired AZ. 3. Click OK.
Setting application component-node affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Node, and select the desired node. 3. Click OK.
Setting application component-node non-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Node, and select the desired node. 3. Click OK.
Setting application component-application component affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Component, and select the desired application components. 3. Click OK. The selected application components are deployed on the same node.
Setting application component-application component anti-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Component, and select the desired application components. 3. Click OK. The selected application components are deployed on different nodes.

Step 8 Click **Re-deployment** to complete the setting.

----End

For other types of components, perform the following operations:

Step 1 Access the page for setting a scheduling policy of an application component instance.

- To set a scheduling policy during **component configuration** in the application component deployment, go to **Step 6**.
- To set a scheduling policy after an application component is deployed, go to **Step 2**.

Step 2 Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

Step 3 Click an application name. The **Overview** page is displayed.

Step 4 On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

Step 5 Click the name of an application component to access its details page. Choose **Upgrade**.

Step 6 Choose **Advanced Settings > Deployment Configuration** and set the following parameters on the **Scheduling Policies** tab page.

Purpose	Procedure
Setting application component-AZ affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Availability Zone, and select the desired AZ. 3. Click OK.
Setting application component-AZ anti-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Availability Zone, and select the desired AZ. 3. Click OK.
Setting application component-node affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Node, and select the desired node. 3. Click OK.
Setting application component-node non-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Node, and select the desired node. 3. Click OK.
Setting application component-application component affinity	<ol style="list-style-type: none"> 1. Click Add an affinity object. 2. Set the object type to Component, and select the desired application components. 3. Click OK. The selected application components are deployed on the same node.
Setting application component-application component anti-affinity	<ol style="list-style-type: none"> 1. Click Add anti-affinity objects. 2. Set the object type to Component, and select the desired application components. 3. Click OK. The selected application components are deployed on different nodes.

Step 7 Complete the settings of the scheduling policy.

- If the scheduling policy is set during **component configuration**, click **Next**.
- If the scheduling policy is set after the application component is deployed, click **Re-deployment**.



----End

5.7 Setting Upgrade Policies for Application Component Instances

You can set an upgrade policy when deploying an application component or when an application component has been deployed.

Procedure

When the component type is **Common** and the runtime system is **Docker**, perform the following operations:

- Step 1** Access the page for setting an upgrade policy of an application component instance.
- To set an upgrade policy during **component configuration** in the application component deployment, go to **Step 2**.
 - To set an upgrade policy after an application component is deployed, go to **Step 3**.
- Step 2** On the **Configure Component** page, set the upgrade policy for the application component instance.
1. Select an upgrade mode for the application component instance.
-  **NOTE**
- The default upgrade mode is **Rolling upgrade**.
 - Rolling upgrade
Install a new instance and then remove the old one. In this pattern, services are evenly distributed to new and old instances during the upgrade, so services are not interrupted.
 - In-place upgrade
Delete the old instance and then create a new one. Services are interrupted during the upgrade.
2. Click **Next** to complete the component deployment.
- Step 3** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 4** Click an application name. The **Overview** page is displayed.
- Step 5** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 6** Click the name of an application component to access its details page.
- Step 7** Choose **Upgrade** to set the upgrade policy.
1. Select an upgrade mode for the application component instance.
-  **NOTE**
- The default upgrade mode is **Rolling upgrade**.
 - Rolling upgrade

Install a new instance and then remove the old one. In this pattern, services are evenly distributed to new and old instances during the upgrade, so services are not interrupted.

- In-place upgrade

Delete the old instance and then create a new one. Services are interrupted during the upgrade.

2. Click **Re-deployment** to complete the setting.

----End

For other types of components, perform the following operations:

- Step 1** Access the page for setting an upgrade policy of an application component instance.
 - To set an upgrade policy during **component configuration** in the application component deployment, go to **Step 6**.
 - To set an upgrade policy after an application component is deployed, go to **Step 2**.
- Step 2** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 3** Click an application name. The **Overview** page is displayed.
- Step 4** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 5** Click the name of an application component to access its details page. Choose **Upgrade**.
- Step 6** Choose **Advanced Settings > Deployment Configuration**. On the **Upgrade Policy** tab page, select the upgrade mode.

 **NOTE**

The default upgrade mode is **Rolling upgrade**.

- Rolling upgrade
Install a new instance and then remove the old one. In this pattern, services are evenly distributed to new and old instances during the upgrade, so services are not interrupted.
- In-place upgrade
Delete the old instance and then create a new one. Services are interrupted during the upgrade.

- Step 7** Complete the setting of the upgrade policy.
 - If the upgrade policy is set during **component configuration**, click **Next**.
 - If the upgrade policy is set after the application component is deployed, click **Re-deployment**.

----End

5.8 Setting Custom Metric Monitoring for Application Components

ServiceStage allows you to obtain monitoring data based on custom metrics.

You can set custom metric monitoring when deploying an application component or when an application component has been deployed.

Precautions

- Currently, only **Gauge metrics** of Prometheus can be obtained.
- Before setting custom metric monitoring for an application component, you must understand **Prometheus** and provide the GET API for obtaining custom metric data in your application component so that ServiceStage can obtain custom metric data using this API.

Procedure

When the component type is **Common** and the runtime system is **Docker**, perform the following operations:

- Step 1** Access the page for setting custom metric monitoring for an application component.
- To set custom metric monitoring during **component configuration** in the application component deployment, go to **Step 2**.
 - To set custom metric monitoring after an application component is deployed, go to **Step 3**.

- Step 2** On the **Configure Component** page:

1. Specify the following parameters to set custom metric monitoring for the application component.

Parameter	Description	Mandatory
Report Path	URL provided by the exporter for ServiceStage to obtain custom metric data. Example: /metrics	Yes
Report Port	Port provided by the exporter for ServiceStage to obtain custom metric data. Example: 8080	Yes

Parameter	Description	Mandatory
Monitoring Metrics	Name of the custom metric provided by the exporter. Example: ["cpu_usage","mem_usage"] <ul style="list-style-type: none"> - If this parameter is not set, ServiceStage collects data of all custom metrics. - If you set this parameter, for example, to ["cpu_usage","mem_usage"], ServiceStage collects the data of the specified cpu_usage and mem_usage metrics. 	No

2. Click **Next** to complete the component deployment.

Step 3 Log in to ServiceStage and choose **Application Management >Application List** to view all applications.

Step 4 Click an application name. The **Overview** page is displayed.

Step 5 On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.

Step 6 Click the name of an application component to access its details page.

Step 7 Choose **O&M Configurations** and set the following parameters.

Parameter	Description	Mandatory
Report Path	URL provided by the exporter for ServiceStage to obtain custom metric data. Example: /metrics	Yes
Report Port	Port provided by the exporter for ServiceStage to obtain custom metric data. Example: 8080	Yes
Monitoring Metrics	Name of the custom metric provided by the exporter. Example: ["cpu_usage","mem_usage"] <ul style="list-style-type: none"> • If this parameter is not set, ServiceStage collects data of all custom metrics. • If you set this parameter, for example, to ["cpu_usage","mem_usage"], ServiceStage collects the data of the specified cpu_usage and mem_usage metrics. 	No

Step 8 Click **Re-deployment** to complete the setting.

 **NOTE**

After the configuration and deployment are complete, you can view the monitoring metric data on the AOM page. For details, see [Metric Monitoring](#).

----End

For other types of components, perform the following operations:

- Step 1** Access the page for setting custom metric monitoring for an application component.
- To set custom metric monitoring during **component configuration** in the application component deployment, go to **Step 6**.
 - To set custom metric monitoring after an application component is deployed, go to **Step 2**.
- Step 2** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 3** Click an application name. The **Overview** page is displayed.
- Step 4** On the **Environment View** tab page, select an environment to view the application components that have been deployed in the environment.
- Step 5** Click the name of an application component to access its details page. Choose **Upgrade**.
- Step 6** Choose **Advanced Configuration > O&M Monitoring**. On the **O&M Policy** tab page, set the following parameters.

Parameter	Description	Mandatory
Report Path	URL provided by the exporter for ServiceStage to obtain custom metric data. Example: /metrics	Yes
Report Port	Port provided by the exporter for ServiceStage to obtain custom metric data. Example: 8080	Yes
Monitoring Metrics	Name of the custom metric provided by the exporter. Example: ["cpu_usage","mem_usage"] <ul style="list-style-type: none"> If this parameter is not set, ServiceStage collects data of all custom metrics. If you set this parameter, for example, to ["cpu_usage","mem_usage"], ServiceStage collects the data of the specified cpu_usage and mem_usage metrics. 	No

- Step 7** Complete the setting of the custom metric monitoring.
- If the custom metric monitoring policy is set during **component configuration**, and click **Next**.

- If the custom metric monitoring policy is set after the application component is deployed, and click **Re-deployment**.

 **NOTE**

After the configuration and deployment are complete, you can view the monitoring metric data on the AOM page. For details, see [Metric Monitoring](#).

----End

5.9 Configuring Application Log Policies

ServiceStage supports setting of application log policies. You can view related logs on the AOM console.

Log policies can be configured during or after application component deployment.

If no configuration is performed, the system collects standard application output logs by default.

Procedure

- Step 1** Go to the log policy configuration page.
- To configure the application log policy during **component configuration** in the component deployment, go to **Step 6**.
 - To configure the application log policy after the component is deployed, go to **Step 2**.
- Step 2** Log in to ServiceStage and choose **Application Management > Application List** to view all applications.
- Step 3** Click an application name. The **Overview** page is displayed.
- Step 4** On the **Environment View** tab, select an environment to view the application components that have been deployed in the environment.
- Step 5** Click the name of an application component to access its details page. Choose **Upgrade**.
- Step 6** Choose **Advanced Settings > O&M Monitoring > Log Collection**, click **Add Log Policy**, and set the parameters listed in the following table.

Parameter	Description
Storage Type	Select a storage type. <ul style="list-style-type: none">• HostPath: Mount a host path to a specified container path.• Mounting Path: Logs are exported only to the container path. You do not need to mount the host path.
Host Path	This parameter is mandatory when Storage Type is set to HostPath . Enter the log storage path on the host.

Parameter	Description
<p>Docker Mounting</p>	<p>1. Set Mounting Path: Enter the application path to which the data volume is mounted.</p> <p>NOTICE</p> <ul style="list-style-type: none"> - Do not mount a data volume to a system directory such as <code>/</code> or <code>/var/run</code>. Otherwise, an exception occurs. An empty directory is recommended. If the directory is not empty, ensure that the directory does not contain any files that affect application startup. Otherwise, the files will be replaced, causing application startup exceptions. As a result, the application fails to be created. - If you intend to mount the volume to a high-risk directory, you are advised to use an account with minimum permissions to start the container. Otherwise, high-risk files on the host may be damaged. <p>2. Set Extended Host Path.</p> <ul style="list-style-type: none"> - None: No extended path is configured. - PodUID: ID of a Pod. - PodName: name of a Pod. - PodUID/ContainerName: ID of a Pod or ontainer name. - PodName/ContainerName: Pod name or container name. <p>3. Set Aging Period.</p> <ul style="list-style-type: none"> - Hourly: Log files are scanned every hour. If the size of a log file exceeds 20 MB, the system compresses the log file to a historical file, dumps the historical file to the directory where the log file is stored, and clears the original log file. - Daily: Log files are scanned once a day. If the size of a log file exceeds 20 MB, the system compresses the log file to a historical file, dumps the historical file to the directory where the log file is stored, and clears the original log file. - Weekly: Log files are scanned once a week. If the size of a log file exceeds 20 MB, the system compresses the log file to a historical file, dumps the historical file to the directory where the log file is stored, and clears the original log file.

Step 7 Click **Confirm**.

Step 8 Complete the application log policy configuration.

- If the log policy is set during **component configuration** in the component deployment, click **Next**.
- If the log policy is set after the component is deployed, click **Re-deployment**.

 **NOTE**

After the configuration and deployment are complete, you can view run logs on the AOM console. For details, see [Viewing Log Files](#).

----End

5.10 Configuring Application Performance Management

ServiceStage allows you to configure application performance management during or after application component deployment.

The Application Performance Management (APM) service helps you quickly locate application problems and analyze performance bottlenecks, improving user experience.

Selecting Java probe will start APM and install Java probes on the nodes deployed with APM, which consumes a small amount of resources. Java probes use the bytecode enhancement technology to trace Java application calls and generate topology and call chain data.

Precautions

- JDK 7 and JDK 8 are supported.
- Tomcat 6.x, 7.x, and 8.x are supported. For details, see [Usage Restrictions](#).

Procedure

When the component type is **Common** and the runtime system is **Docker**, perform the following operations:

- Step 1** Go to the performance management configuration page.
 - To configure performance management during [component configuration](#) in the application component deployment, go to [Step 2](#).
 - To configure performance management after the component is deployed, go to [Step 3](#).
- Step 2** Go to the **Configure Component** page and configure performance management.
 1. Select **Java probe** and select a probe version.
 2. Click **Next** to complete the component deployment.
- Step 3** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 4** Click an application name. The **Overview** page is displayed.
- Step 5** On the **Environment View** tab, select an environment to view the application components that have been deployed in the environment.
- Step 6** Click the name of an application component to access its details page.
- Step 7** Click **O&M Configurations**, select **Java probe**, and select a probe version.
- Step 8** Click **Re-deployment** to complete the application performance management configuration.

----End

For other types of components, perform the following operations:

- Step 1** Go to the performance management configuration page.
- To configure performance management during **component configuration** in the application component deployment, go to **Step 6**.
 - To configure performance management after the component is deployed, go to **Step 2**.
- Step 2** Log in to ServiceStage and choose **Application Management >Application List** to view all applications.
- Step 3** Click an application name. The **Overview** page is displayed.
- Step 4** On the **Environment View** tab, select an environment to view the application components that have been deployed in the environment.
- Step 5** Click the name of an application component to access its details page. Choose **Upgrade**.
- Step 6** Choose **Advanced Settings > O&M Monitoring** and click **Performance Management**.
- If the performance management is configured during **component configuration** in the component deployment, select **Java Probe** and select a probe version.
 - If the performance management is configured after the component is deployed, select **Java probe** and select a probe version.
- Step 7** Complete the application performance management configuration.
- If the performance management is configured during **component configuration** in the component deployment, click **Next**.
 - If the performance management is configured after the component is deployed, click **Re-deployment**.
- End

5.11 Configuring Health Check

Health check periodically checks application health status during application component running according to your needs.

ServiceStage provides the following health check methods:

- **Component Liveness Probe:** checks whether an application component exists. It is similar to the **ps** command that checks whether a process exists. If the liveness check of an application component fails, the cluster restarts the application component. If the liveness check is successful, no operation is executed.
- **Component Service Probe:** checks whether an application component is ready to process user requests. It may take a long time for some applications to start before they can provide services. This is because that they need to load disk data or rely on startup of an external module. In this case, the application process exists, but the application cannot provide services. This check method is useful in this scenario. If the application component readiness check fails, the cluster masks all requests sent to the application component. If the application component readiness check is successful, the application component can be accessed.

Health Check Modes

- HTTP request-based check

This health check mode is applicable to application components that provide HTTP/HTTPS services. The cluster periodically sends an HTTP/HTTPS GET request to such application components. If the return code of the HTTP/HTTPS response is within 200–399, the check is successful. Otherwise, the check fails. In this health check mode, you must specify an application listening port and an HTTP/HTTPS request path.

For example, if the application component provides the HTTP service, the port number is 80, the HTTP check path is **/health-check**, and the host address is **containerIP**, the cluster periodically initiates the following request to the application:

```
GET http://containerIP:80/health-check
```

NOTE

If the host address is not set, the instance IP address is used by default.

- TCP port-based check

For applications that provide a TCP communication service, the cluster periodically establishes a TCP connection to the application. If the connection is successful, the probe is successful. Otherwise, the probe fails. In this health check mode, you must specify an application listening port. For example, if you have a Nginx application component with service port 80, after you configure a TCP port-based check for the application component and specify port 80 for the check, the cluster periodically establishes a TCP connection with port 80 of the application component. If the connection is successful, the check is successful. Otherwise, the check fails.

- CLI-based check

In this mode, you must specify an executable command in an application component. The cluster will periodically execute the command in the application component. If the command output is **0**, the health check is successful. Otherwise, the health check fails.

The CLI mode can be used to replace the following modes:

- TCP port-based check: Use a program to connect to an application component port. If the connection is successful, the script returns **0**. Otherwise, the script returns **-1**.
- Http request-based check: Use a script to run the **wget** command for an application component.

wget http://127.0.0.1:80/health-check

Check the return code of the response. If the return code is within 200–399, the script returns **0**. Otherwise, the script returns **-1**.

NOTICE

- Put the program to be executed in the application component image so that the program can be executed.
 - If the command to be executed is a shell script, add a script interpreter instead of specifying the script as the command. For example, if the script is `/data/scripts/health_check.sh`, you must specify `sh/data/scripts/health_check.sh` for command execution. The reason is that the cluster is not in the terminal environment when executing programs in an application component.
-

Procedure

- Step 1** Go to the application health check configuration page.
- To configure health check during **component configuration** in the application component deployment, go to **Step 6**.
 - To configure health check after the component is deployed, go to **Step 2**.
- Step 2** Log in to ServiceStage and choose **Application Management > Application List** to view all applications.
- Step 3** Click an application name. The **Overview** page is displayed.
- Step 4** On the **Environment View** tab, select an environment to view the application components that have been deployed in the environment.
- Step 5** Click the name of an application component to access its details page. Choose **Upgrade**.
- Step 6** Choose **Advanced Settings > O&M Monitoring**, click **Health Check**, and set health check parameters based on service requirements.
- Step 7** Complete the application health check configuration.
- If the health check is configured during **component configuration** in the component deployment, click **Next**.
 - If the health check is configured after the component is deployed, click **Re-deployment**.

----End

6 Microservice Governance

[Overview](#)

[Using the Microservice Dashboard](#)

[Governing Microservices](#)

[Configuring Microservices](#)

[Maintaining Microservices](#)

6.1 Overview

If an application is developed using the microservice framework, the microservice is automatically registered with the corresponding microservice engine after the application is managed and started. You can perform service governance on the microservice engine console. Service governance applies only to Java Chassis and Go Chassis development frameworks.

Currently, ServiceStage provides the Cloud Service Engine (professional edition). You can use it directly. You can also create exclusive microservice engines. For details, see [Creating an Exclusive Microservice Engine](#).

6.2 Using the Microservice Dashboard

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

Background

- If a microservice application is deployed on ServiceStage, you need to configure the microservice engine during application deployment. The application automatically obtains the service registration center address, configuration center address, and dashboard address. You do not need to configure the monitor address. Currently, only Java Chassis and Go Chassis support automatic discovery of a dashboard address.

- If the microservice application is locally started and registered with the microservice engine, you need to manually configure the monitor address before using the dashboard.

Procedure

- Step 1** Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.
- Step 2** On the page that is displayed, select a microservice engine, and click **Console**.
- Step 3** On the **Dashboard** page, select an application from the drop-down list box and enter a microservice name in the search box. The operating metrics of the microservice are displayed.

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

- Step 4** Select a sorting order to sort the filtered microservices.

----End

6.3 Governing Microservices

After a microservice is deployed, you can govern it based on its running statuses.

Prerequisites

- You can create a microservice in **Microservice List** from **Service Catalog** and start the microservice. After the microservice starts, the service instance is registered under the corresponding service based on configurations in the **.yaml** file.
- If the microservice is not created in advance or has been deleted, the microservice is automatically created when the service instance is registered.
- After a microservice is created, you need to register the service instance before performing the corresponding operation.

Governance Policies

You can configure the **Load Balancing**, **Rate Limiting**, **Fault Tolerance**, **Service Degradation**, **Circuit Breaker**, and **Fault Injection** policies. For details, see the following table.

Name	Description
Load Balancing	<p>When the access traffic and traffic volume are large and one server cannot handle the load, you can configure load balancing to distribute traffic to multiple servers for balancing. In this way, the response duration is reduced and server overload can be prevented.</p> <p>You can configure load balancing policies by adding a rule. The rule parameters include Polling, Random, Response Time Weight, and Session Stickiness.</p>

Name	Description
Rate Limiting	<p>Rate limiting is used to solve the problem of traffic distribution across microservices. This ensures that microservices run in their own resource pools without affecting each other.</p> <ul style="list-style-type: none">• When the number of requests sent by the rate limiting object to the current service instance exceeds the specified value, the current service instance no longer accepts requests from the rate limiting object.• Common detection methods include request timeout and excessive traffic.• The parameters include Flow Control Object and QPS.
Service Degradation	<p>Service degradation is a special form of fault tolerance. When the service throughput is large and resources are insufficient, you can use service degradation to disable some services that are not important and have poor performance to avoid occupying resources and ensure that the main services are normal.</p>
Fault Tolerance	<p>Fault tolerance is used when an exception occurs in a service instance after you access that instance. After the exception occurs, you can retry to access the instance, or access another instance based on the configured policy.</p>
Circuit Breaker	<p>If the service is overloaded, you can use circuit breaker to protect the system from breaking down.</p> <p>Circuit breaker is triggered when a service request is handled abnormally. After circuit breaker is triggered, Hystrix considers that the requested service cannot process requests, so it immediately rejects requests and returns an error message to the caller.</p> <p>Hystrix attempts to access backend services at a specified interval. If the services are restored, they will exit the circuit breaker state and resume to accept requests.</p>
Fault Injection	<p>Fault injection is used to test the fault tolerance capability of microservices. This helps the user determine whether the system can run properly when latency or fault occurs.</p> <p>Fault injection allows you to test fault tolerance of microservices with latency or faults.</p>
Routing Policy	<p>Based on the public key authentication mechanism, CSE provides the blacklist and whitelist functions to control the services that can access microservices.</p> <p>The blacklist and whitelist take effect only after public key authentication is enabled. For details, see Configuring Public Key Authentication.</p>

Configuring the Load Balancing Policy

- Step 1** Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.
- Step 2** Click **Console** of a microservice engine and choose **Service Governance**.
- Step 3** Click the microservice to be governed.
- Step 4** Click **Load Balancing**.
- Step 5** Click **Add**. Select the microservices to be governed and select a proper load balancing policy. For details, see the following table.

Policy	Description
RoundRobin	Supports routes according to the location information about service instances.
Random	Provides random routes for service instances.
Response Time Weight	Provides weight routes with the minimum active number (latency) and supports service instances with slow service processing in receiving a small number of requests to prevent the system from stopping response. This load balancing policy is suitable for applications with low and stable service requests.
Session Stickiness	<p>Provides a mechanism on the load balancer. In the specified session stickiness duration, this mechanism allocates the access requests related to the same user to the same instance.</p> <ul style="list-style-type: none"> • Stickiness Duration: Session hold time. Range: 0-86400. Unit: s. • Failures Threshold: Number of access failures. Range: 0-10. If the upper limit of failures or the session stickiness duration exceeds the specified values, the microservice stops accessing this instance.

- Step 6** Click **OK** to save the settings.

----End

Configuring the Rate Limiting Policy

- Step 1** Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.
- Step 2** Click **Console** of a microservice engine and choose **Service Governance**.
- Step 3** Click the microservice to be governed.
- Step 4** Click **Rate Limiting**.
- Step 5** Click **Add**. The following table describes configuration items of rate limiting.

Configuration Item	Description	Value Range
Flow Control Object	Application that access the microservice.	You can select the microservice that the application relies on from the drop-down list.
QPS	Requests generated per second. When the number of requests sent by the rate limiting object to the current service instance exceeds the specified value, the current service instance no longer accepts requests from the rate limiting object.	An integer ranging from 0 to 99999.

 **NOTE**

If a microservice has three instances, the rate limiting of each instance is set to 2700 QPS, then the total QPS is 8100, and rate limiting is triggered only when the QPS exceeds 8100.

Step 6 Click **OK** to save the settings.

----End

Configuring the Service Degradation Policy

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Governance**.

Step 3 Click the microservice to be governed.

Step 4 Click **Service Degradation**.

Step 5 Click **Add**. Select a proper policy. The following table describes the configuration items of service degradation.

Configuration Item	Description
Fallback Object	Microservice to be degraded and the corresponding degradation method.
Fallback Policy	<ul style="list-style-type: none"> ● Open: enables degradation. ● Close: disables degradation.

Step 6 Click **OK** to save the settings.

----End

Configuring the Fault Tolerance Policy

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Governance**.

Step 3 Click the microservice to be governed.

Step 4 Click **Fault Tolerance**.

Step 5 Click **Add**. Select a proper policy. The following table describes the configuration items of fault tolerance.

Configuration Item	Description
Fault Tolerance Object	Microservice or method that the application relies on. You can select it from the drop-down list.
Fault Tolerance	<p>Open: The system processes the service request based on the selected fault tolerance policy when the request sent to the fault tolerance object encounters an error.</p> <p>Close: The system waits until the timeout interval expires and then returns the failure result even though the service request fails to be implemented.</p>
<p>FT Policy</p> <p>NOTE Set this parameter when Fault Tolerance is set to Open.</p>	<ul style="list-style-type: none"> ● Failover The system attempts to reestablish connections on different servers. ● Failfast The system does not attempt to reestablish a connection. After a request fails, a failure result is returned immediately. ● Failback The system attempts to reestablish connections on the same server. ● custom <ul style="list-style-type: none"> – Number of attempts to reestablish connections on the same server – Number of attempts to reestablish connections on new servers

Step 6 Click **OK** to save the settings.

----End

Configuring the Circuit Breaker Policy

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Governance**.

Step 3 Click the microservice to be governed.

Step 4 Click **Circuit Breaker**.

Step 5 Click **Add**. Select a proper policy. The following table describes the configuration items of circuit breaker.

Configuration Item	Description
Fallbreak Object	You can select the microservice or method that the application relies on from the drop-down list.
Triggering Condition	<ul style="list-style-type: none"> ● Manual Fallbreak Circuit breaker is triggered immediately and microservice instances are not called. ● Cancel Fallbreak Circuit breaker taking effect on the microservice instance is canceled and the microservice instance can be called. ● Auto Fallbreak <ul style="list-style-type: none"> - Fallbreak Time Window: circuit breaker duration. No response is sent within the time window. - Failure Rate: trigger condition, that is, request failure rate of the window. - Window Requests: trigger condition, that is, number of requests received by the window. Circuit breaker is triggered only when Failure Rate and Window Requests both reach their thresholds.

Step 6 Click **OK** to save the settings.

----End

Configuring the Fault Injection Policy

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Governance**.

Step 3 Click the microservice to be governed.

Step 4 Click **Fault Injection**.

Step 5 Click **Add**. Select a proper policy. The following table describes the configuration items of fault injection.

Configuration Item	Description
Injection Object	Microservices for which fault injection is required. You can specify a method for this configuration item.
Type	Type of the fault injected to the microservice. <ul style="list-style-type: none"> ● Delayed ● Error

Configuration Item	Description
Protocol	Protocol for accessing the microservice when latency or fault occurs. <ul style="list-style-type: none">• Rest• Highway
Delay Time	Period of latency when accessing a microservice. This parameter is required when Type is set to Delayed .
HTTP Error Code	HTTP error code during microservice access. This parameter is required when Type is set to Error . This error code is an HTTP error code.
Occurrence Probability	Probability of latency or fault occurrence.

Step 6 Click **OK** to save the settings.

----End

Configuring Blacklist and Whitelist

Based on the public key authentication mechanism, CSE provides the blacklist and whitelist functions. The blacklist and whitelist can be used to control which services can be accessed by microservices.

The blacklist and whitelist take effect only after public key authentication is enabled. For details, see [Configuring Public Key Authentication](#).

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Governance**.

Step 3 Click the microservice to be governed.

Step 4 Click **Routing Policy**.

Step 5 Click **Add** to add a blacklist or whitelist for the application. The following table describes configuration items of blacklist and whitelist.

Configuration Item	Description
Type	<ul style="list-style-type: none">• Blacklist: microservices matching the matching rule are not allowed to access the current service.• Whitelist: microservices matching the matching rule are allowed to access the current service.

Configuration Item	Description
Rule	Expressed by a regular expression. For example, if Rule is set to data* , it indicates that a service whose name starts with data in the blacklist cannot access the current service, or a service whose name starts with data in the whitelist can access the current service.

Step 6 Click **OK** to save the settings.

----End

Configuring Public Key Authentication

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

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

To enable public key authentication, perform the following steps:

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

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

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

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

6.4 Configuring Microservices

You can use the global configuration function provided by ServiceStage to configure microservices.

After the global configuration is added, the configuration takes effect immediately if it is used by all microservices registered with the engine.

If dynamic configuration is set for a single microservice, the dynamic configuration overwrites the global configuration. For details about how to set dynamic configuration, see [Dynamic Configuration](#).

Global Configuration

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

NOTICE

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

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Global Configuration**. Perform the following operations.

Operation	Procedure
Export configurations.	Click Export All to export all global configuration items.
Import configurations.	<ol style="list-style-type: none">1. Click Import.2. Click ... to select a configuration item file.3. Click Upload Files to import configuration items in batches.
Add configurations.	<ol style="list-style-type: none">1. Click Create Configuration. The Create Configuration dialog box is displayed.2. Select a microservice environment and enter Configuration Item and Value.3. Click OK to save the settings.
Modify configurations.	<ol style="list-style-type: none">1. Click Edit in the Operations column corresponding to the target configuration item.2. Enter Value.3. Click OK to save the settings.
Delete configurations.	<ol style="list-style-type: none">1. Select the configuration item to be deleted.2. Click Delete in the Operations column.3. In the Delete Configuration dialog box, click OK to delete the global configuration.
Delete configuration items in batches	<ol style="list-style-type: none">1. Select the configuration items to be deleted.2. Click Delete above the configuration item list to delete global configuration items in batches.

----End

6.5 Maintaining Microservices

You can use service catalogs to view microservice details and search for target microservices to maintain microservices.

The following information is displayed on the **Service Catalog** page:

- **Application List:** displays all applications of the current user. You can search for the target application by application name, or filter applications by environment.
- **Microservice List:** displays all microservices of the current user. You can search for the target microservice by microservice name, or filter microservices by environment and application.
- **Instance List:** displays all instances of the current user. You can search for the target instance by microservice name, or filter instances by environment, application, or status.

 **NOTE**

Microservice diagnosis depends on the development framework used by microservice applications. Currently, only microservices developed using the SDK can be diagnosed. Applications developed using the Mesher framework cannot be diagnosed.

Dynamic Configuration

- Step 1** Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.
- Step 2** Click **Console** of a microservice engine and choose **Service Catalog**.
- Step 3** Click a microservice.
- Step 4** Choose **Dynamic Configuration**. The **Dynamic Configuration** page is displayed. On the **Dynamic Configuration** page, perform the following operations.

NOTICE

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

Operation	Procedure
Export configuration S.	<p>Select a scope from the All drop-down list and click Export to export the JSON configuration file of the current scope.</p> <p>The scope format is as follows:</p> <ul style="list-style-type: none"> • Microservice name@application to which the microservice belongs • Microservice name@application to which the microservice belongs#version number

Operation	Procedure
Import configuration S.	<ol style="list-style-type: none"> Click Import and select a scope. The scope format is as follows: <ul style="list-style-type: none"> Microservice name@application to which the microservice belongs Microservice name@application to which the microservice belongs#version number Click ... to select a configuration item file. Click Upload Files to import configuration items in batches. Click Close.
Create configuration S.	<ol style="list-style-type: none"> Click Create Configuration and select a scope. Enter Configuration Item. Enter Value. Click OK to save the settings.
Modify configuration S.	<ol style="list-style-type: none"> Click Edit in the Operations column corresponding to the target configuration item. Enter a new value in the Value text box. Click OK to save the settings.
Delete configuration S.	<ol style="list-style-type: none"> Click Delete in the Operations column corresponding to the target configuration item. Click OK to delete the configuration.

----End

Dark Launch

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Catalog**.

Step 3 On the displayed page, click a microservice. On the page that is displayed, choose **Dark Launch**.

Step 4 Click **Add a Launch Rule**.

- To add a launch rule by **Weight**:
 - a. Click **Weight**.
 - b. On the displayed dialog box, set the following parameters.

Configuration Item	Description
Rule Name	Name of a customized rule.

Configuration Item	Description
Scope	<ul style="list-style-type: none"> ▪ Version of the microservice to which the rule applies. ▪ Add Custom Version: adds a new version as prompted.
Rule Configuration	Traffic allocation rate for the selected version. Traffic is evenly allocated to the selected service versions based on the configured value.

- c. Click **OK** to complete the weight rule configuration and dark launch.
- To add a launch rule by **Customization**:
 - a. Click **Customization**.
 - b. On the displayed dialog box, set the following parameters.

Configuration Item	Description
Rule Name	Name of a customized rule.
Scope	<ul style="list-style-type: none"> ▪ Version of the microservice to which the rule applies. ▪ Add Custom Version: adds a new version as prompted.
Rule Configuration	<ul style="list-style-type: none"> ▪ Parameter Name This name is customized according to the key field provided by the service contract. This key must exist in the contract. It is possible that the server API is String paramA, but paramB is actually generated after the annotation is added. Therefore, paramB should be set here. ▪ Rules Value corresponding to the key of a contract. <p>NOTE</p> <ul style="list-style-type: none"> ○ If ~ is selected from the drop-down list next to Rules, the asterisk (*) and question mark (?) can be used for fuzzy matching when you specify the value of Rules. The asterisk (*) represents an unlimited number of characters, and the question mark (?) represents only one character. For example, if the rule value of Name is set to *1000, all Name fields ending with 1000 can be matched. ○ If ~ is not selected from the drop-down list next to Rules, the asterisk (*) and question mark (?) cannot be used for fuzzy matching.

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

----End

Delete a microservice.

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Catalog**.

Step 3 On the displayed page, select a microservice to be deleted, click **Delete**, and delete the microservice as prompted.

NOTE

- If the number of microservice instances is 0, you can directly delete the microservice.
- If the number of microservice instances is not 0, the microservice will be re-registered with the service center after being deleted for a period of time.
- When the microservice has dependencies, it cannot be deleted.

----End

Viewing Microservice Details

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Catalog**.

Step 3 Click a microservice. The microservice details page is displayed.

On the microservice details page, you can view the instance list, called services, calling services, dynamic configuration, dark launch, and service contract.

----End

Viewing a Service Contract

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Catalog**.

Step 3 Click a microservice. The microservice details page is displayed.

Step 4 Click **Service Contract** to view the service contract.

----End

Adding a Label

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Click **Console** of a microservice engine and choose **Service Catalog**.

Step 3 On the displayed page, click a microservice. On the page that is displayed, and click **Label Management** next to **Label**.

Step 4 Click **Add Label**, and enter **Key** and **Value**.

Step 5 Click **OK** to save the settings.

----End

Modifying Microservice Instance Status

Status indicates the status of a microservice instance. The following table describes the microservice instance statuses.

Status	Description
Online	The instance is running and can provide services.
Offline	The instance process ends.
Out of Service	The instance is running but cannot provide services.
Test	The instance is being tested.

Step 1 Log in to ServiceStage and choose **Infrastructure > Cloud Service Engines**.

Step 2 Select a microservice engine and click **Console**. The microservice engine console is displayed.

Step 3 Choose **Service Catalog > Instance List**.

Step 4 Select the target instance and change the microservice instance status.

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

----End

7 Continuous Delivery

[Overview](#)

[Creating a Source Code Build Task](#)

[Creating a Package Build Task](#)

[Managing Pipelines](#)

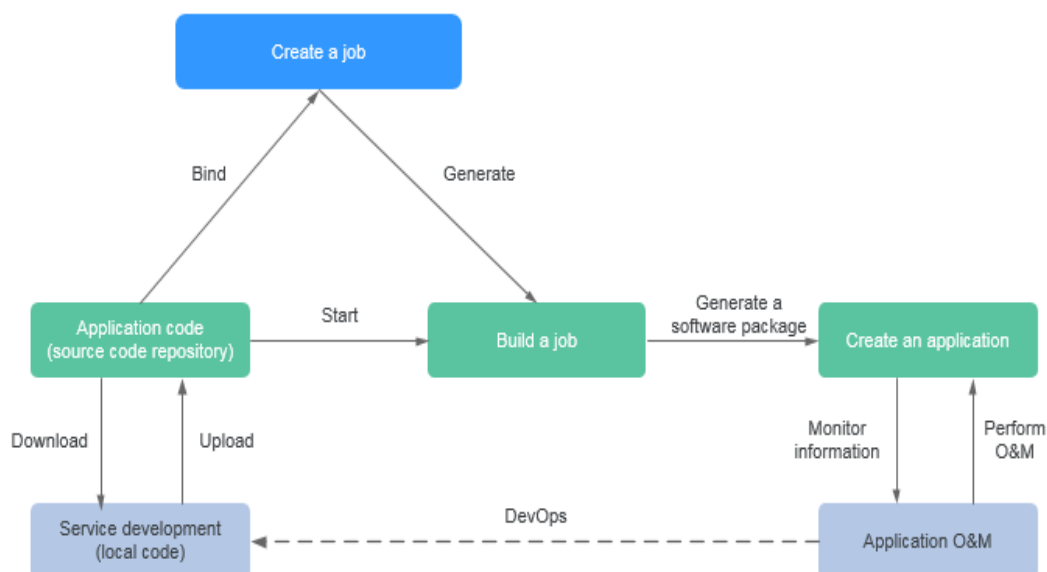
[Authorizing a Repository](#)

7.1 Overview

Creating a Build Job

Based on the existing service code, you can create a build job, start the build task, package the service code, and archive the package to the software center. Then, you can use the package when deploying an application component.

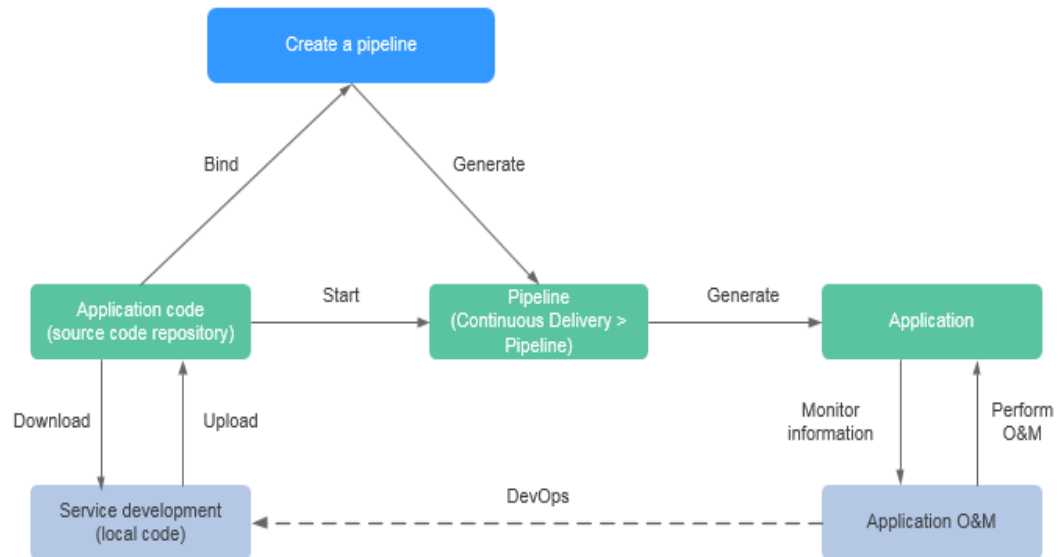
Figure 7-1 Creating a build job



Creating a Pipeline

Based on the existing service code, you can create a pipeline and then start the pipeline to complete service code building and deployment. Application O&M can also be completed on ServiceStage.

Figure 7-2 Creating a pipeline



7.2 Creating a Source Code Build Task

- The software package or image package can be generated with a few clicks in a build job. In this way, the entire process of source code pull, compilation, packaging, and archiving is automatically implemented.
- Images built in the x86-system jobs are ones of the x86 system.
- Images built in the Arm-system jobs are ones of the Arm system.

Prerequisites

1. A cluster has been created. For details, see [Cluster Management](#).
2. You have bound an elastic IP address to the build node by referring to [Assigning an EIP and Binding It to an ECS](#).

Procedure

Step 1 Log in to ServiceStage, choose **Continuous Delivery > Build**, and click **Create Source Code Job**.

Step 2 Perform the following operations to set basic information:

1. Enter **Name**.
2. Specify **Enterprise Project**.

An enterprise project provides a cloud resource management mode, in which cloud resources and members are centrally managed by project.

It is available after the [enterprise center](#) is enabled.

3. (Optional) Specify **Description**.
4. Specify **Code Source**.
 - Create authorization by referring to [Authorizing a Repository](#) and set the code source.
 - Click **Samples** and select a required sample.
5. Select a cluster from the **Cluster** drop-down list box.
6. (Optional) Specify **Node Label** to deliver the build task to a fixed node based on the node label. For details about how to add a node label, see [Node Management](#).
7. Click **Next**.

Step 3 Select a build template.


- If you select **Maven, Ant, Gradle, Go, or Docker**, you can compile and archive binary packages or Docker images at the same time. Go to [Step 4](#).
- If you select **Custom**, you can customize the build mode. Go to [Step 6](#).


Step 4 Select an archive mode.

- **Not archived:** No Docker build job is added or archived.
- **Archive binary package:** No Docker build job is added and binary packages are archived.
- **Archive image compilation:** Docker build job is added and Docker images are archived.


Step 5 Set mandatory parameters.

To delete a parameter setting, click  on the parameter setting page.


- **Compilation parameters**
Compilation parameters are set with different values. For details about parameter description, click a text box or  next to it.
- **Image parameters**

On the  page, enter **Task Name**, **Dockerfile address**, **Image**, and **Image Version**.

- **Image archiving parameters**

On the  page, set **Task Name**, **Archive image**, **Repository Organization**, and **Sharing Type** of the corresponding image to archive the image.

- **Binary parameters**

On the  page, set the following parameters.

Parameter	Description
Task Name	Task name.

Parameter	Description
Sharing Type	Repositories are classified into public repositories and private repositories. <ul style="list-style-type: none">– Public repositories are isolated from each other. Tenants in the same system can resources.– Private repositories are isolated by tenants. Users under the current tenant share resources. Other tenants cannot access resources of the current tenant.
Repository Organization	Namespace of a repository.
Software Repository	Name of a software repository.
Name	Name of the archived software package after the build completes.
Software Package Version	Version of the archived software package.
Build Package Path	Path of the software package archived to the software repository after the build completes.

Step 6 (Optional) Click **Advanced Configuration** to set the environment.

To add multiple tasks, you can customize them in **Advanced Configuration**.

1. Click **Add plugin** in the corresponding stage on the left. The **Select task type** page is displayed.
2. Click **select** of the target task type to add a task type. Then, set task parameters in the right pane of the **Build environment configuration** page.

NOTICE

When the Build Common Cmd plug-in is added to the compilation process, pay attention to the following:

- Exercise caution when inputting sensitive information in the **echo**, **cat**, or **debug** command, or encrypt sensitive information to avoid information leakage.
- When **Language** is set to **Python** and **Python Framework Type** is set to a Python project that complies with the **WSGI** standard, you need to set **Main Python Module** and **Function of the Main Python Module**. The following is an example of the main Python module and main function:

Main Python Module: If the entry point file of the Python project is **server.py**, the main module name is **server**.

Function of the Main Python Module: If the application function name of the Python project entry point file **server.py** is **app=get_wsgi_application()**, the function name of the main module is **app**.

Step 7 Click **Build**.

Click **Save** to save the settings (not to start the build).

----End

Related Operations

After an application component is successfully built, you can manage it on ServiceStage. For details, see [Deployment Mode](#).

Maintenance

Table 7-1 Maintenance

Operation	Description
Query details/build history	<ol style="list-style-type: none">1. Click the name of the target build project and view the build history under Build Record.2. Click a record to view the record.3. Click Code Check to view the code check overview and details. Currently, the following code check plug-ins are supported: checkstyle, findbugs, and pmd. <p>NOTE Only the Maven build project supports code check.</p>
Build Now	Select the target build project and click Build Now in the Operations column.

Operation	Description
Branch/Tag	Select the target build project and click Branch/Tag in the Operations column. <ol style="list-style-type: none">1. Select Branch/Tag.2. Select Branch or Tag.3. Specify the CommitId for the branch or tag.4. Click OK.
Edit	Select the target build project and choose More > Edit in the Operations column to edit the build project.
Delete	<ol style="list-style-type: none">1. Select the target build project and choose More > Delete in the Operations column.2. Click OK.

7.3 Creating a Package Build Task

The image package can be generated with a few clicks in a build job. In this way, the entire process of package obtainment, and image compilation and archiving is automatically implemented.

Prerequisites

1. A cluster has been created. For details, see [Cluster Management](#).
2. You have bound an elastic IP address to the build node by referring to [Assigning an EIP and Binding It to an ECS](#).

Procedure

Step 1 Log in to ServiceStage, choose **Continuous Delivery > Build**, and click **Create Package Job**.

Step 2 Enter **Job Name**.

Step 3 Specify **Enterprise Project**.

An enterprise project provides a cloud resource management mode, in which cloud resources and members are centrally managed by project.

It is available after the [enterprise center](#) is enabled.

Step 4 (Optional) Specify **Description**.

Step 5 Specify **Package Source**.

The following upload modes are supported:

- Select the corresponding software package from the SWR software repository. You need to upload the software package to the software repository in advance. For details, see [Uploading the Software Package](#).

- Select the corresponding software package from OBS. You need to upload the software package to the OBS bucket in advance. For details, see [Uploading a File](#).

Click **Select Package** and select the corresponding software package.

Step 6 Select a build type.

- System default
 - a. Select a basic image, which must be the same as the software package compilation language selected in [Step 5](#).
 - b. Specify **Basic Image**.
- Custom Dockerfile
Enter a custom command in the compilation box.

NOTICE

Exercise caution when inputting sensitive information in the **echo**, **cat**, or **debug** command, or encrypt sensitive information to avoid information leakage.

- Image
Specify **Basic Image**, which must be the same as the software package compilation language selected in [Step 5](#).

Step 7 Specify **Image Class**.

- Public: This is a widely used standard image that contains an OS and pre-installed public applications and is visible to all users. You can configure the applications or software in the public image as needed.
- Private: A private image contains an OS or service data, pre-installed public applications, and private applications. It is available only to the user who created it.

Step 8 Specify **Archived Image Address**.

Step 9 Select a cluster.

- Build with shared cluster
Use the shared cluster allocated by the system.
- Build with your own cluster
If you use your own cluster to perform build task, you can deliver build task to fixed nodes through node labels. For details about how to add a node label, see [Node Management](#).

Step 10 Click **Build Now** to start the build.

Click **Save** to save the settings (not to start the build).

----End

Related Operations

After an application component is successfully built, you can manage it on ServiceStage. For details, see [Deployment Mode](#).

Maintenance

Table 7-2 Maintenance

Operation	Description
Query details/build history	<ol style="list-style-type: none">1. Click the name of the target build project and view the build history under Build Record.2. Click a record to view the record.
Build Now	Select the target build project and click Build Now in the Operations column.
Edit	Select the target build project and choose More > Edit in the Operations column to edit the build project.
Delete	<ol style="list-style-type: none">1. Select the target build project and choose More > Delete in the Operations column.2. Click OK.

7.4 Managing Pipelines

One-click deployment can be achieved through pipeline. In this way, the entire process of source code pull, compilation, packaging, archiving, and deployment is automatically implemented. This unifies the integration environment and standardizes the delivery process.

In the new pipeline, the "phase/task" model is optimized to the "build/environment" model. Each pipeline includes a group of build tasks and one or more groups of environment (such as development environment, production-like environment, and production environment) tasks, each group of environment tasks contains one or more subtasks (such as deployment and test tasks) and provides templates.

Creating a Pipeline

Step 1 Log in to ServiceStage, choose **Continuous Delivery > Pipeline**, and click **Create Pipeline**.

Step 2 Enter the basic pipeline information.

1. Specify **Pipeline**.
2. Specify **Enterprise Project**.

An enterprise project provides a cloud resource management mode, in which cloud resources and members are centrally managed by project.

It is available after the [enterprise center](#) is enabled.

3. (Optional) Enter the **Description**.

Step 3 Select a pipeline template.

ServiceStage provides built-in pipeline templates in typical scenarios. After you select a pipeline template, the Build/Environment model is automatically generated. You can directly use the model.

Table 7-3 Template description

Template	Description	Operation Description
Empty template	You need to add the build/environment model.	Set this parameter as required. For details, see Step 3.1 to Step 3.3 .
Simple template	The "build" model is automatically added to compile and build the source code of the code library.	For details, see Step 3.1 .
Common template	The "build/environment" model is automatically added to compile and build the source code in the code library, and the generated software package or image is continuously released to the production environment.	For details, see Step 3.1 to Step 3.3 .

1. Add a build task.

Click **Select Build Job**, select a created build job, and click **OK**.

If no build task is available, choose **Select Build Job > New build task** to create a source code build task or package build task. For details, see [Creating a Source Code Build Task](#) or [Creating a Package Build Task](#).

Repeat this step to add more build tasks.

2. Add a deploy task.

Click **Add Environment** and enter an environment name. Select a deployed application component.


If no application component is available, create and deploy an application component. For details, see [Deploying a Component](#).

Select the build task added in [Step 3.1](#) from the **Select Build Job** drop-down list box.

Select build output.

Repeat this step to add more environments.

3. Set pipeline approval.

Click  in the environment area to set the approval mode and approvers.

- **Approval Mode: By all** and **By one person** are now supported.
- **Approved By:** You can select multiple accounts as approvers. The system automatically loads all subaccounts of the account.

Step 4 Click **Create and Start** to start the pipeline.

Click **Create** to save the settings and do not execute the pipeline.

----End

Configuring the Pipeline Triggering Policy

Choose **Continuous Delivery > Pipeline**. On the **Pipeline** page that is displayed, set the pipeline triggering policy as follows.

Table 7-4 Triggering policies

Policy	Mode	Description
Manual	-	Select the pipeline task to be triggered and click Start to manually start the pipeline.
Automatic	-	Set the code source, corresponding namespace, repository name, and branch. When code is submitted to the corresponding branch of the source code repository, the pipeline is automatically triggered. You can set a maximum of eight trigger sources. The procedure is as follows: <ol style="list-style-type: none"> 1. Select a pipeline and choose More > Triggering Policy. 2. Set Type to Automatic. 3. Specify Source Code Repository to push the code to the selected source code repository. 4. Click OK.

Policy	Mode	Description
Scheduled	Single-time	<p>Set the triggering time to trigger a single-time pipeline.</p> <p>The procedure is as follows:</p> <ol style="list-style-type: none">1. Select a pipeline and choose More > Triggering Policy.2. Set Type to Scheduled.3. Specify Triggered.4. Click OK.
	Periodic	<p>Set the triggering time segment, interval, and period to implement periodic pipeline triggering.</p> <p>The procedure is as follows:</p> <ol style="list-style-type: none">1. Select a pipeline and choose More > Triggering Policy.2. Set Type to Scheduled.3. Enable Periodic Triggering.4. Specify Period, Triggered, Effective Time, and Period.5. Click OK.

Cloning a Pipeline

You can clone a pipeline to generate a new pipeline based on the existing pipeline configuration.

- Step 1** Log in to ServiceStage and choose **Continuous Delivery > Pipeline**.
- Step 2** Select a pipeline and choose **More > Clone**.
- Step 3** ServiceStage automatically loads the configuration information. Modify the configuration parameters as required by referring to [Creating a Pipeline](#).
- Step 4** Click **Create and Start** to start the pipeline.

Click **Create** to save the settings and do not execute the pipeline.

----End

Related Operations

After the pipeline is started, you can build and deploy applications in one-click mode. For details about maintenance operations after application components are deployed, see [Application O&M](#).

7.5 Authorizing a Repository

You can create repository authorization so that build projects and application components can use the authorization information to access the software repository.

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

Step 2 Click **Create Authorization** and configure the authorization information by referring to the following table. Parameters marked with an asterisk (*) are mandatory.

Table 7-5 Authorization information

Parameter	Description
*Name	Authorization name, which cannot be changed after being created.
*Repository Type	The following repositories are supported: <ul style="list-style-type: none">● GitHub Authorization mode: OAuth, private token, or password.● Bitbucket Authorization mode: OAuth, password, or private Bitbucket.● GitLab Authorization mode: OAuth or private token.

Step 3 Click **Create**.

----End

8 Software Center

- [Software Repository](#)
- [Image Repository](#)
- [Organization Management](#)

8.1 Software Repository

8.1.1 Managing Software Packages

To upload a software package to a new software repository, you can create a software repository after selecting an organization during software package creation.


NOTICE

- The software repository does not scan or verify the security of the uploaded software packages. To avoid privacy leakage, do not include privacy information such as unencrypted passwords in uploaded software packages. When downloading public software packages, ensure that they are from trusted repositories and prevent malicious software from being downloaded.
 - If a disk is full, software packages cannot be uploaded to the repository and error information is displayed, but services are not affected. To prevent services such as logs from occupying the entire disk, you are advised to attach an independent disk to the repository.
-

Creating a Software Package

- Step 1** Log in to ServiceStage and choose **Software Center > Software Repository**.
- Step 2** Click **Create Package**. Configure the software package by referring to the following table. Parameters marked with an asterisk (*) are mandatory.

Table 8-1 Parameter description

Parameter	Description
*Software Repository	Select an organization and a software repository. To create a software repository: <ol style="list-style-type: none">1. Click Create Repository and enter a new software repository name.2. Click .
Sharing Type	Type of the software repository. The default value is Private . <ul style="list-style-type: none">● Private: only for the current tenant and users under the current tenant.● Public: for all tenants and users.
Package Name	Software package name, which must be unique in an organization.
Version Number	Software package version. Multiple software package versions can be uploaded.
Package Description	Description of the software package.
Version Description	Description of the software package version.
Upload Software package	<ul style="list-style-type: none">● Upload now: Upload the software package by referring to Step 3 in Uploading the Software Package.● Upload later: After the creation is complete, upload the software package by referring to Uploading the Software Package.

Step 3 Click **OK**.

----End

Uploading the Software Package

NOTICE

A maximum of 10 files can be uploaded at a time. The size of a single file (including the decompressed files) cannot exceed 2 GB.

Step 1 Log in to ServiceStage and choose **Software Center > Software Repository**.

Step 2 Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 Click **Upload Software package** next to the target software package.

1. Click **Select File**, select the target software package, and click **Open**. Alternatively, drag the target software package to the page.
2. Set the parameters in the following table. All the parameters are optional.

Table 8-2 Parameter description

Parameter	Description
Cover	If you select this option, the software package with the same name in the same path will be overwritten.
Package Path	Enter a path to store the software package. The path is the virtual path of the software repository. By default, the root directory is used. By setting the path, you can easily view and manage the software package.

Repeat the preceding steps to upload other software packages.

3. After the software package is selected:
 - Select a software file from the list of software files to be uploaded and click **Upload** in the **Operation** column to upload the specified software file.
 - In the upper part of the list of software to be uploaded, click **Upload** to upload software files in batches.

----End

Editing a Software Package


- Step 1** Log in to ServiceStage and choose **Software Center > Software Repository**.
- Step 2** Select an organization from the drop-down list box on the right of **Organization Management**.
- Step 3** Click the target software package to enter the details page.
- Step 4** Click **Edit** in the upper-right corner and set the following parameters:
 - **Sharing Type**: Set the type of the software repository. **Private**: only for the current tenant and users under the current tenant. **Public**: for all tenants and users.
 - **Package Description**: Enter the description of the software package.
- Step 5** Click **OK**.

----End

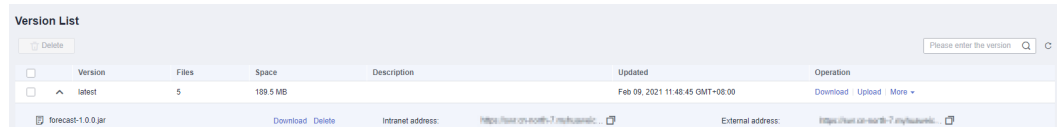
Querying the Address of a Software Package

- Step 1** Log in to ServiceStage and choose **Software Center > Software Repository**.
- Step 2** Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 Click the target software package to enter the details page.

Step 4 In the version list, click  before the target version to view the software package address.

Click  to copy **Internal address** or **External address**.



Version	Files	Space	Description	Updated	Operation
latest	5	188.5 MB		Feb 09, 2021 11:48:45 GMT+08:00	Download Upload More
forecast-1.0.0.jar					Download Delete

NOTE

In the row where a version file is located:

- Click **Download** to download the file.
- Click **Delete** to delete the file.

----End

Deleting a Software Package

Step 1 Log in to ServiceStage and choose **Software Center > Software Repository**.

Step 2 Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 Click **Delete** on the right side of the target software package, and delete the software package as prompted.

NOTE

Before deleting a software package, ensure that all versions in the software package are deleted. For details, see [Deleting a Software Package Version](#).

----End

Deleting a Software Package Version

Step 1 Log in to ServiceStage and choose **Software Center > Software Repository**.

Step 2 Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 Click the target software package to enter the details page. In the version list:

- Deleting a single software package version

Choose **More > Delete** in the **Operations** column of the target version, and delete the software package version as prompted.

- Deleting software package versions in batches

Select the target versions, click **Delete** above the version list, and delete the software package versions as prompted.

----End

8.1.2 Packaging Specifications of Software Packages

JAR and WAR packages can be directly uploaded.

For other types of software packages, such as ZIP packages,

the software package name must be in the following format: software name +suffix. The suffix must be .tar.gz, .tar, or .zip.

NOTE

The suffix must be consistent with the package compression mode. Otherwise, software packages cannot be decompressed.

Directory Structure

For decompressed software packages, ensure that lifecycle command scripts can be normally executed.

The following software package directory structure is recommended:

```
|- bin
  |- xxx.tar.gz
  |- xxx.bin
|- scripts
  |- install.sh
  |- start.sh
...
```

NOTE

Currently, you are advised not to store decompressed software packages in the top-level directory. Otherwise, when you need to modify lifecycle execution commands, you have to use the top-level directory name to find the corresponding scripts.

Table 8-3 Description of the software package directory

Directory	Description
bin	Stores execution information about software packages, such as executable bin files and dependent compressed packages.

Directory	Description
scripts	<p>Stores lifecycle scripts.</p> <p>When creating an application, you can specify execution commands based on the location of lifecycle scripts. For example, specify bash scripts/install.sh in the install phase to run the installation script.</p> <p>Lifecycle commands supported by software package applications are as follows:</p> <ul style="list-style-type: none">● Install: Command for installing software.● PostStart: Operation performed after software is started.● Start: Command for starting software.● Restart: Command for restarting software, which is used to recover the applications failing in health check.● PreStop: Operation which is performed before software is stopped.● Stop: Command for stopping software.● Update: Command for upgrading software.● Uninstall: Command for uninstalling software.

8.2 Image Repository

8.2.1 Uploading an Image

After an organization is created, you can upload an image to it through the page or client.

- **Uploading an Image Through the Page:** Upload an image to SWR through the page.
- **Uploading an Image Through the Client:** Upload an image to an image repository of SWR by running commands on the client.

Uploading an Image Through the Page

NOTE

A maximum of 10 files can be uploaded at a time. The size of a single file (including the decompressed files) cannot exceed 2 GB.

Prerequisites

- An organization has been created. For details, see [Creating an Organization](#).
- The image has been saved as a .tar or .tar.gz file. For details, see [Creating an Image Package](#).
- The image package is created using Docker 1.11.2 or later.

Procedure

Step 1 Log in to ServiceStage and choose **Software Center > Image Repository**.

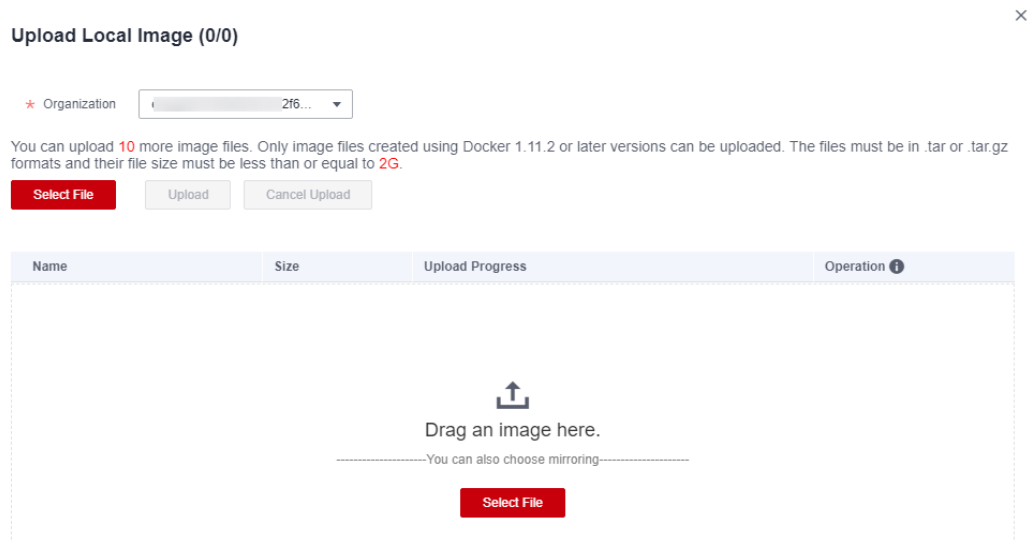
Step 2 On the **My Images** tab, click **Upload Through SWR**.

Step 3 In the dialog box that is displayed, specify **Organization** to which the image is to be uploaded, click **Select File**, and select the image file to be uploaded as shown in [Figure 8-1](#).

 **NOTE**

If you select multiple images to upload, the system uploads them one by one. Concurrent upload is not supported.

Figure 8-1 Uploading an image



Step 4 In the dialog box that is displayed, click **Start Upload**.

If **Upload completed** is displayed, the image is successfully uploaded.

 **NOTE**

If the image fails to be uploaded, the possible causes are as follows:

- The network is abnormal. In this case, check network connectivity.
- The HTTPS certificate has errors. Press **F12** to copy the URL that fails to be requested to the address bar of the browser, open the URL again, agree to continue the access, and return to the upload page to upload the certificate again.

----End

Uploading an Image Through the Client

 **NOTE**

If you use the client to upload an image, each image layer cannot exceed 10 GB.

Prerequisites

- An organization has been created. For details, see [Creating an Organization](#).
- Your container engine client version must be 1.11.2 or later.

Procedure

- Step 1** Log in to ServiceStage and choose **Software Center > Image Repository**.
- Step 2** In the upper-right corner of the **My Images** page, click **Upload Through Docker Client**.
- Step 3** Upload the image as prompted.

----End

8.2.2 Managing Images

Obtaining an Image Pull Address

- Step 1** Log in to ServiceStage, and choose **Software Center > Image Repository > My Images**.
- Step 2** Select an organization from the drop-down list box on the right of **Organization Management**.
- Step 3** In the image repository list, click an image repository name to go to the details page.
- Step 4** Click the **Image Tags** tab and obtain the command for pulling an image.

Click  on the right of the command to copy the command.



----End

Setting Image Repository Attributes

- Step 1** Log in to ServiceStage, and choose **Software Center > Image Repository > My Images**.
- Step 2** Select an organization from the drop-down list box on the right of **Organization Management**.
- Step 3** In the image repository list, click an image repository name to go to the details page.
- Step 4** Click **Edit** in the upper-right corner. In the dialog box that is displayed, perform the following operations:

- Set **Sharing Type** to **Public** or **Private**.

NOTE

- Public images can be downloaded and used by all users.
 - If your node and the image repository are in the same region, you can access the image repository over private networks.
 - If your node and the image repository are in different regions, the node must have access to public networks to pull images from the image repository.
- Specify **Category** to set the repository category.
- Specify **Description** to update the description of the image repository.

Step 5 Click **OK**.

----End

Sharing a Private Image

After pushing a private image, you can share it with other users and grant access permission to them.

Only administrator and Identity and Access Management (IAM) users authorized to manage the private image can share the image. The users with whom you share the image only have the read permission. That is, they can only pull the image.

Step 1 Log in to ServiceStage, and choose **Software Center > Image Repository > My Images**.

Step 2 Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 In the image repository list, click an image repository name to go to the details page.

Step 4 Click the **Sharing** tab, click **Share Image**, and set the following parameters:

1. **Share With:** Enter an account name.
2. **Valid Until:** Set the expiration date. If you want the image to be permanently accessible to the account, select **Permanently valid**.
3. **Description:** Enter the description.
4. **Permission:** Select the permission. Currently, only the **Download** permission is supported.

Step 5 Click **OK**.

- You can view all shared images in the shared image list.
- Select an account name and click **Edit** in the **Operation** column to edit the parameters of the shared image.
- Select an account name and click **Delete** in the **Operation** column to cancel sharing.

----End

Setting Automatic Image Synchronization

If image synchronization is enabled, the latest images are automatically synchronized to image repositories in other regions. Only accounts and users with administrator permissions can configure automatic image synchronization.

NOTE

After you configure automatic image synchronization, image updates will also be synchronized to target repositories. However, images that were pushed to repositories before automatic image synchronization was enabled will not be automatically synchronized.

For details on how to synchronize images pushed before you set the automatic synchronization, see [Can Existing Images be Automatically Synchronized](#).

- Step 1** Log in to ServiceStage, and choose **Software Center > Image Repository > My Images**.
- Step 2** Select an organization from the drop-down list box on the right of **Organization Management**.
- Step 3** In the image repository list, click an image repository name to go to the details page.
- Step 4** Click **Set Image Synchronization** in the upper-right corner.
- Step 5** In the displayed dialog box, click **Add**, set the following parameters, and click **OK** in the **Operation** column.
- **Target Region:** The target region for image synchronization, for example, CN South-Guangzhou.
 - **Target Organization:** The target organization to which the image will be synchronized.
 - **Overwrite Existing Image:** Select this option if you want to overwrite any nonidentical images that have the same name in the target organization. Deselect this option if you do not want any nonidentical images having the same name in the target organization to be overwritten and you want to receive a notification of the existence of such images.

✕

Set Image Synchronization

i After synchronization is set, any images that are pushed are automatically synchronized to the target region and target organization.

Target Region	Target Organization	Overwrit...	Operation
--Select-- ▾	--Select-- ▾	<input type="checkbox"/>	OK Cancel

⊕ Add

OK
Cancel

- Step 6** Click **OK**.
- On the **Synchronization Records** tab of image details page, you can view the details of each synchronization task, including the start time, image tag, task status, type, duration, target region and organization, and task operator.

----End

Adding Image Permissions

To allow IAM users of your account to read, write, and manage a specific image, add the required permissions to the IAM users on the details page of this image.

- Step 1** Log in to ServiceStage, and choose **Software Center > Image Repository > My Images**.

Step 2 Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 In the image repository list, click an image repository name to go to the details page.

Step 4 Click the **Permission Management** tab, click **Add Permission**, select an IAM user, add the **Read**, **Write**, or **Manage** permission, and click **OK**.

Then, this IAM user has the corresponding permission.

----End

Deleting an Image

Step 1 Log in to ServiceStage and choose **Software Center > Image Repository > My Images**.

Step 2 Select an organization from the drop-down list box on the right of **Organization Management**.

Step 3 In the image repository list, click an image repository name to go to the details page.

- Deleting an image repository
Click **Delete** in the upper-right corner of the page and delete the image repository as prompted.
- Deleting an image tag
In the **Operations** column of the target image tag, click **Delete** to delete the image tag as prompted.
- Deleting image tags in batches
Select the target image tags, click **Delete** above the tag list, and delete the image tags as prompted.

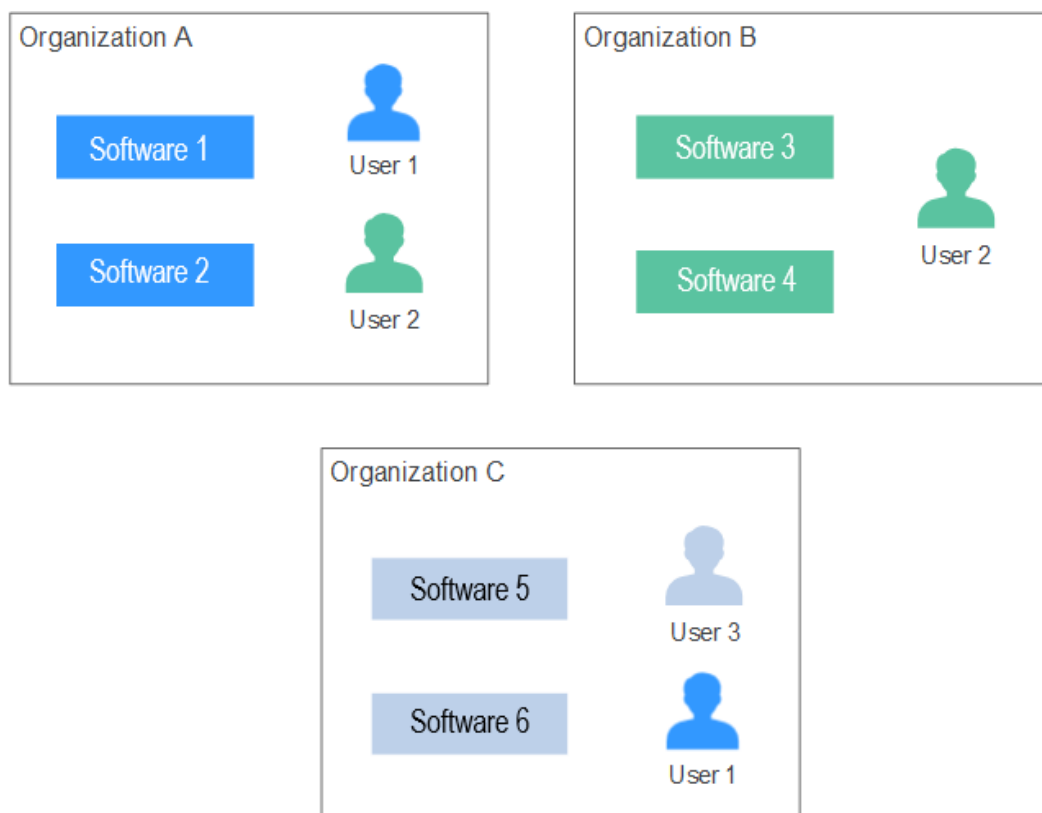
----End

8.3 Organization Management

Overview

Organizations are used to isolate software and image repositories. With each organization being limited to one company or department, software can be managed in a centralized manner. A software name only needs to be unique within an organization. An IAM user can join different organizations. Different permissions, namely read, write, and manage, can be assigned to different IAM users in the same account.

Figure 8-2 Organization



Creating an Organization

- Step 1** Log in to ServiceStage and choose **Software Center > Organization**.
- Step 2** Click **Create Organization**. On the page that is displayed, enter **Organization Name** and click **OK**.

Figure 8-3 Creating an organization

The screenshot shows a dialog box titled "Create Organization" with a close button (X) in the top right corner. Below the title, there is a list of instructions:

- Each organization name must be globally unique.
- Each account can create 200 organizations.
- For centralized management of images, limit each organization to one company, department, or individual.

Example:
Company or department: cloud-hangzhou or cloud-develop
Person: john

Below the instructions is a text input field labeled with a red asterisk and "Organization Name". At the bottom of the dialog are two buttons: "OK" and "Cancel".

----End

Adding Permissions

Grant permissions to users in an organization so that they can read, edit, and manage all images in the organization.

Only users with the **Management** permission can grant permissions.

User permissions include:

- **Read-only:** Users can only download software but cannot upload software.
- **Read/write:** Users can download software, upload software, and edit software attributes.
- **Management:** Users can download and upload software, delete software or versions, edit software attributes, grant permission, and share images.

Step 1 Log in to ServiceStage and choose **Software Center > Org Management**.

Step 2 Click **Add Permission** on the right of an organization.

Step 3 In the dialog box that is displayed, specify **Permission** and click **OK**.

----End

Deleting an Organization

Step 1 Log in to ServiceStage and choose **Software Center > Org Management**.

Step 2 Click **Delete** on the right of an organization.

Before deleting an organization, delete the image and software repositories of the organization.

For details about how to delete an image repository, see [Managing Images](#).

For details about how to delete a software repository, see [Deleting a Software Package](#).

Step 3 Click **OK**.

----End

9 Infrastructure Management

[Cloud Service Engines](#)

[VMAgent Manager](#)

9.1 Cloud Service Engines

9.1.1 Creating an Exclusive Microservice Engine

Cloud Service Engine (CSE) provides service registry, service governance, and configuration management. It allows you to quickly develop microservice applications and implement high-availability O&M. Furthermore, it supports multiple languages and runtime systems, and unified access and governance of intrusive frameworks such as Spring Cloud, Apache ServiceComb (Java chassis/Go chassis), and Dubbo, and non-intrusive Service Mesh.

You can use the professional microservice engine named "Cloud Service Engine" or create an exclusive microservice engine.

- An exclusive microservice engine is physically isolated. Tenants exclusively use the microservice engine. You can customize specifications and features, and you can create a microservice engine with specific instance quantity.
- The professional microservice engine does not support multiple AZs.
- You can configure multiple AZs when creating an exclusive engine.
- After a microservice engine is created, the AZ cannot be modified. Select a suitable AZ when creating a microservice engine.
- Exclusive microservice engines cannot run across CPU architectures.

Prerequisites

An exclusive microservice engine runs on a VPC. Before creating a microservice engine, ensure that VPCs and subnets are available.

Procedure

Step 1 Log in to ServiceStage and choose **Infrastructure > CSE**.

Step 2 Click **Buy Exclusive Microservice Engine** and select **Exclusive Microservice engine**.

 **NOTE**

A maximum of five exclusive microservice engines can be created for each project by default. If you want to create more exclusive microservice engines, [submit a service ticket](#) to increase quotas.


Step 3 Set the parameters. For details, see [Table 9-1](#).

Table 9-1 Description

Parameter	Description
Microservice Engine Name	Name of the microservice engine. The name cannot be changed after the engine is created.
(Optional) Description	Description of the microservice engine.
Enterprise Project	Project where the microservice engine locates.
Specifications	Specifications of the microservice engine. <ul style="list-style-type: none">● HA: The service is more reliable. You can select up to three AZs for the production environment.● Non-HA: You can select only one AZ. You are advised to use this specification only in the service development and test phases.
AZ	Availability zone. <ul style="list-style-type: none">● The AZ of a created microservice engine cannot be changed.● The AZs in one region can communicate with each other over an intranet.
Instances	Maximum number of CSE instances.
Virtual Private Cloud	VPC in which the microservice engine locates. A VPC enables you to provision logically isolated, configurable, and manageable virtual networks for your engine.
Subnet	Select a subnet.

Step 4 Click **Next**. Confirm the specifications and select **I have read and agree to HUAWEI CLOUD User Agreement and Disclaimer**. Click **Submit** to create the microservice engine.

- It takes 10 to 30 minutes to create an exclusive microservice engine.

- After the microservice engine is created, the microservice engine status changes to **Available**.
- If the microservice engine fails to be created, click  and choose **Retry**.

----End

9.1.2 Configuring Backup and Restoration of an Exclusive Microservice Engine

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

Context

- This function applies only to exclusive microservice engines.
- Each exclusive microservice engine supports a maximum of 15 successful backups, including a maximum of 10 manual backups and a maximum of 5 automatic backups.
- The backup data will be stored for 10 days. Expired backup data will be deleted.

Automatic Backup

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
- Step 2** Click an exclusive microservice engine. The **Basic Information** page is displayed.
- Step 3** Click the **Backup and Restore** tab, click **Backup Policy**, and set backup parameters.

Table 9-2 Backup parameter description

Parameter	Description
Automatic Backup	After automatic backup is disabled, the previously set backup policy will be deleted. In this case, you need to set the backup policy again.
Backup Cycle	Backup period.
Start Time	Time at which a backup task starts. Only the hour is supported.

- Step 4** Click **OK** to complete the configuration of the backup policy.

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

----End

Manual Backup

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
- Step 2** Click an exclusive microservice engine. The **Basic Information** page is displayed.
- Step 3** Click the **Backup and Restore** tab, click **Manual Backup**, and set backup parameters.

Table 9-3 Backup parameter description

Parameter	Description
Name	Name of a backup task. The name cannot be changed after the backup task is created.
Remarks	Description about the backup task. This field is optional.

- Step 4** Click **OK** to execute the backup task immediately.
- End

Restoring Backup Data

NOTICE

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

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
- Step 2** Click an exclusive microservice engine. Then, the **Basic Information** page is displayed.
- Step 3** Click the **Backup and Restore** tab, and click **Restore** in the **Operations** column next to the specified backup data.
- Step 4** Select **I have read and fully understood the risk** and click **OK** to restore the backup data.

To view the restoration status, click **Restoration History**.

----End

9.1.3 Configuring Public Network Access for an Exclusive Microservice Engine

Enabling Public Network Access

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.

- Step 2** Click an exclusive microservice engine. Then, the **Basic Information** page is displayed.
- Step 3** Click the switch next to **Public Network Access** and read the warning message.
- Step 4** Confirm that you want to proceed with the operation and click **OK**.

----End

Binding an EIP

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
- Step 2** Click an exclusive microservice engine. Then, the **Basic Information** page is displayed.
- Step 3** If public network access is not enabled for the microservice engine, enable it by referring to [Enabling Public Network Access](#).
- Step 4** Click the drop-down list next to **EIP**. Select an EIP from the drop-down list and click ✓.

If no EIP is available, click **Create EIP** to create one.

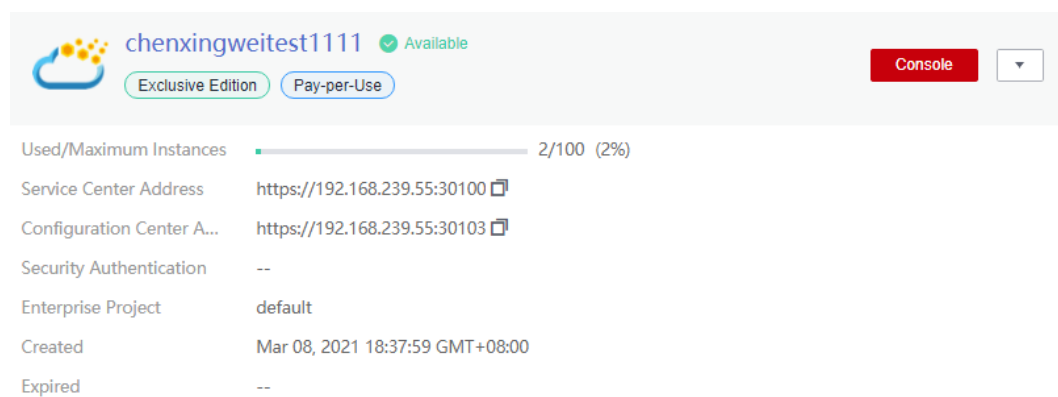
----End

9.1.4 Viewing the Access Address of a Microservice Engine

After you create an exclusive microservice engine, perform the following steps to view the access addresses of all components of the microservice engine.

Procedure

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
- Step 2** Locate the target exclusive microservice engine, and view or copy the connection address.



The screenshot displays the console interface for a microservice engine. At the top, the engine name 'chenxingweitest1111' is shown with a green checkmark and the word 'Available'. Below the name are two buttons: 'Exclusive Edition' and 'Pay-per-Use'. To the right, there is a red 'Console' button and a dropdown menu. The main area contains a table of engine details:

Used/Maximum Instances	2/100 (2%)
Service Center Address	https://192.168.239.55:30100
Configuration Center A...	https://192.168.239.55:30103
Security Authentication	--
Enterprise Project	default
Created	Mar 08, 2021 18:37:59 GMT+08:00
Expired	--

----End

9.1.5 Viewing Operation Logs of an Exclusive Microservice Engine

Operations such as creation, upgrade, deletion, and change will be performed in the backend. You can view the task execution status in the list.

Procedure

- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
- Step 2** Click an exclusive microservice engine. Then, the **Basic Information** page is displayed.
- Step 3** Click the **Tasks** tab, click a task type in the **Task Type** column, and view the operation log details.

No.	Task Type	Username	Status	Started	Ended	Details
1	Create	paas_cse_dev410940_01	Successful	Mar 08, 2021 18:37:59 GMT+08:00	Mar 08, 2021 18:48:13 GMT+08:00	Create a new Engine.

----End

9.1.6 Upgrading an Exclusive Microservice Engine

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

NOTICE

- You cannot roll back exclusive microservice engines after upgrading them.
- Only exclusive microservice engines can be upgraded.


Context

During the upgrade, the performance of HA engines is different from that of non-HA engines.

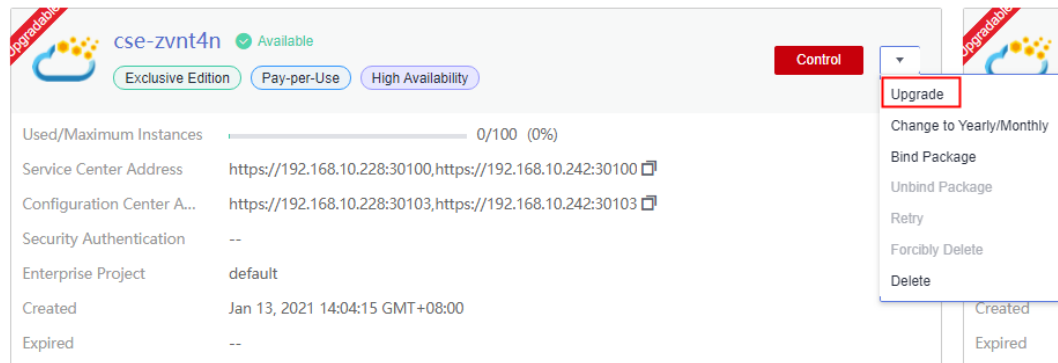
- For HA engines, two instances are upgraded in rolling mode without service interruptions. However, one of the two access addresses may be unavailable. In this case, you need to quickly switch to the other instance. Currently, ServiceComb SDK, Go chassis, and Mesher support instance switching. If you call the APIs of the service center and configuration center for service registration and discovery, instance switching is required.
- A non-HA engine has only one service instance. During the upgrade, services are interrupted and cannot be registered, discovered, or modified. Therefore, you need to evaluate whether the service is affected for an upgrade. The reliability of ServiceComb SDK, Go chassis, and Mesher is enhanced. During service interruption, the client caches data. If an empty instance is found, the client uses the local cache. If you call the APIs of the service center and configuration center for service registration and discovery, you need to use cache.

Procedure

Step 1 Log in to ServiceStage and choose **Infrastructure > CSE**.

Step 2 Select an available microservice engine that can be upgraded, click , and choose **Upgrade**.

You can also click the target microservice engine and click **Upgrade** in the upper right corner of the details page.



Step 3 Select the target version and view the version description.

Step 4 Click **OK** to perform the upgrade.

If the upgrade fails, click  and choose **Retry**.

----End

9.1.7 Deleting an Exclusive Microservice Engine

You can delete an exclusive microservice engine if it is no longer used.

NOTICE

Deleted engines cannot be recovered. Exercise caution when performing this operation.

Context

- You can delete exclusive microservice engines in the following states:
 - Available
 - Unavailable
 - Creation failed
 - Resizing failed
 - Upgrade failed
- The professional microservice engine does not have underlying resources and cannot be deleted.

Procedure

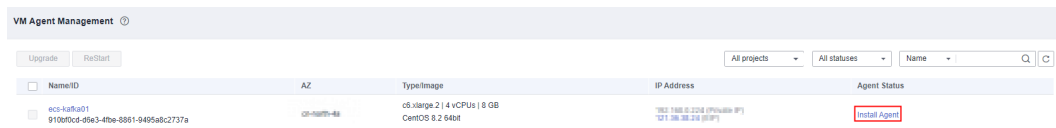
- Step 1** Log in to ServiceStage and choose **Infrastructure > CSE**.
 - Step 2** Click the target exclusive microservice engine to go to the **Basic Information** page.
 - Step 3** In the upper right corner of the page, click **Delete**.
 - Step 4** Enter **DELETE** and click **OK**.
- End

9.2 VM Agent Manager

To deploy an application to a virtual machine (VM), you need to install the agent. After the host is managed, the backend can communicate with the host.

Procedure

- Step 1** Log in to ServiceStage, and choose **Infrastructure > VM Agent Manager**.
- Step 2** Locate the VM where the agent is to be installed and click **Install Agent**.





Name/ID	AZ	Type/Image	IP Address	Agent Status
ecs-k8s-01 9100f0cd-d9e3-4fbc-8881-9495a8c2737a	cn-north-4a	c6.vlarge.2 4 vCPUs 8 GB CentOS 8.2 64bit	192.168.0.123 (Private IP) 121.388.38.218 (Public IP)	Install Agent

- Step 3** On the page that is displayed, select an authorization model.

Authorize the agent to use your authentication information to obtain the deployment, upgrade, start, and stop tasks of an application and execute the task.

You can use agency or AK/SK to perform authorization. You are advised to use agency.

- Select **Agency Authorization** for **Authorization Mode**.

Click , select an agency, and click .

For details about how to create an agency, see [Creating an Agency](#).

NOTE

When creating an agency, you need to delegate the `op_svc_ecs` account to manage resources or ECS cloud service to access cloud resources of another account, and select the Tenant Administrator policy in the corresponding region.

- Select **AK/SK** for **Authorization Mode**.

Enter the AK and SK.

For details about how to obtain the AK/SK, see [Access Keys](#).

- Step 4** Select **Add application access port automatically** based on service requirements.
- Step 5** Copy the command automatically generated in the lower part of the window, that is, the agent installation command.

Example command for the **Agency Authorization** model:


```
export AGENT_INSTALL_URL=https://${Region_Name}-servicestage-vmapp.obs.$
${Region_Name}.myhuaweicloud.com/vmapp/agent/agent-install.sh;if [ -f `which curl` ];then curl -# -O -k $
{AGENT_INSTALL_URL};else wget --no-check-certificate ${AGENT_INSTALL_URL};fi;bash agent-install.sh $
{Project_ID} ${Version} ${Region_Name} ${Flag}
```

Example command for the AK/SK model:

```
export AGENT_INSTALL_URL=https://${Region_Name}-servicestage-vmapp.obs.$
${Region_Name}.myhuaweicloud.com/vmapp/agent/agent-install.sh;if [ -f `which curl` ];then curl -# -O -k $
{AGENT_INSTALL_URL};else wget --no-check-certificate ${AGENT_INSTALL_URL};fi;bash agent-install.sh $
{AK}${SK} ${Project_ID} ${Version} ${Region_Name} ${Flag}
```

- In the preceding command, **AGENT_INSTALL_URL** indicates the installation address of the Agent.

 **NOTE**

- If another region is used, **AGENT_INSTALL_URL** is **https://\${region_name}-servicestage-vmapp.obs.\${region_name}.myhwclouds.com/vmapp/agent/agent-install.sh**.
- If the **Agency Authorization** model is used, the ECS node has the permission to obtain the temporary AK/SK of the user. In this case, you do not need to enter AK/SK in the command.
- **\${AK}/\${SK}** indicates an access key.
- **\${Region_Name}** indicates a region name.
- **\${Project_ID}** indicates a project ID. For details about how to obtain a project ID, see [API Credentials](#).
- **\${Version}** is the version number. Use **latest** to automatically download the latest version.
- **\${Flag}** is a Boolean value, indicating whether to automatically add the application access port. **true** indicates yes and **false** indicates no.

Step 6 Log in to the VM and run the installation command.

----End