ServiceStage

Service Overview

Issue	01
Date	2025-07-04

HUAWEI

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

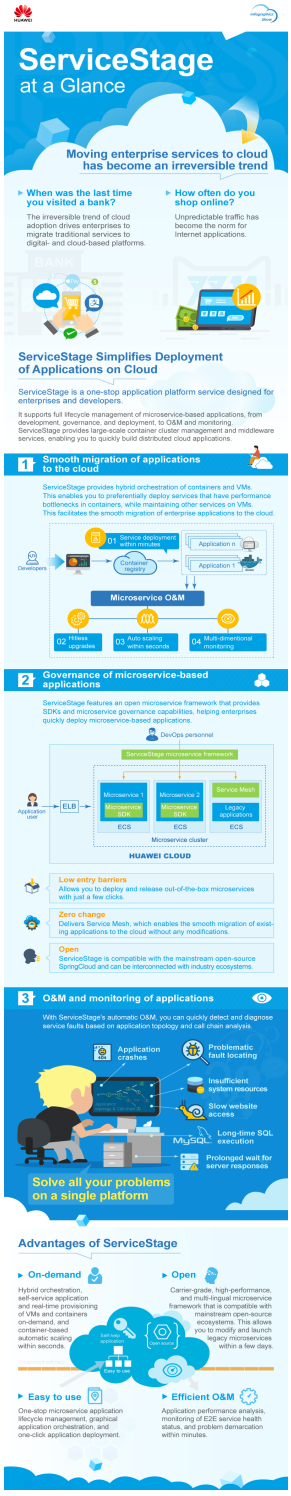
Website: https://www.huaweicloud.com/intl/en-us/

Contents

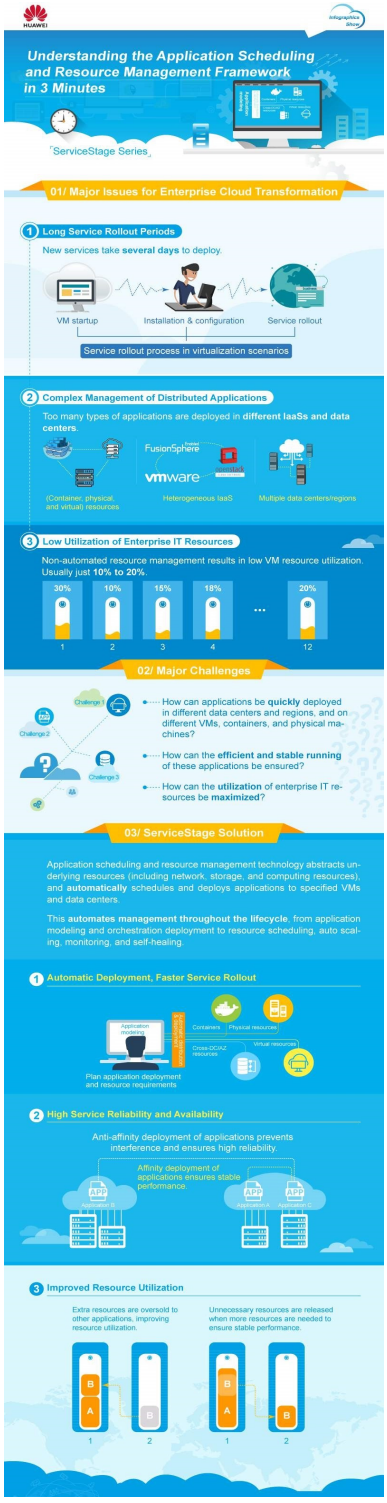
1 ServiceStage Infographics....1
1.1 Introduction to ServiceStage....2
1.2 Application Scheduling and Resource Management Framework....3
1.3 Microservice Operation and Governance Framework....4
1.4 Intelligent Application O&M....5
2 What Is ServiceStage?....6
3 Product Advantages....9
4 Application Scenario....12
4.1 Constructing Microservice Applications....12
4.2 Web Application Lifecycle Management....15
5 Security....16
5.1 Shared Responsibilities....16
5.2 Authentication and Access Control....18
5.3 Data Protection....18
5.4 Audit and Logs....19
5.5 Resilience....20
5.6 Security Risk Monitoring....20
6 Microservice Engine Versions....21
7 Restrictions....23
8 Edition Differences....27
9 Permissions Management....31
10 Relationship with Other Cloud Services....45
11 Glossary....47

1 ServiceStage Infographics

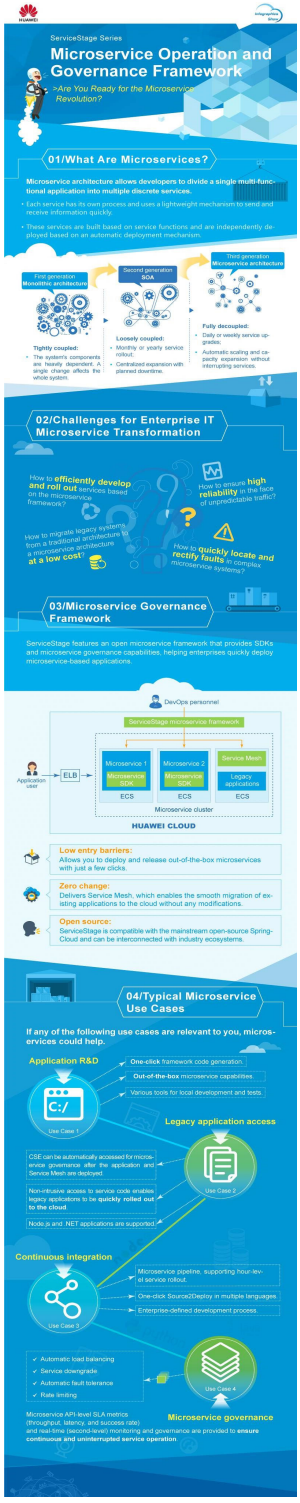
1.1 Introduction to ServiceStage



1.2 Application Scheduling and Resource Management Framework



1.3 Microservice Operation and Governance Framework



1.4 Intelligent Application O&M

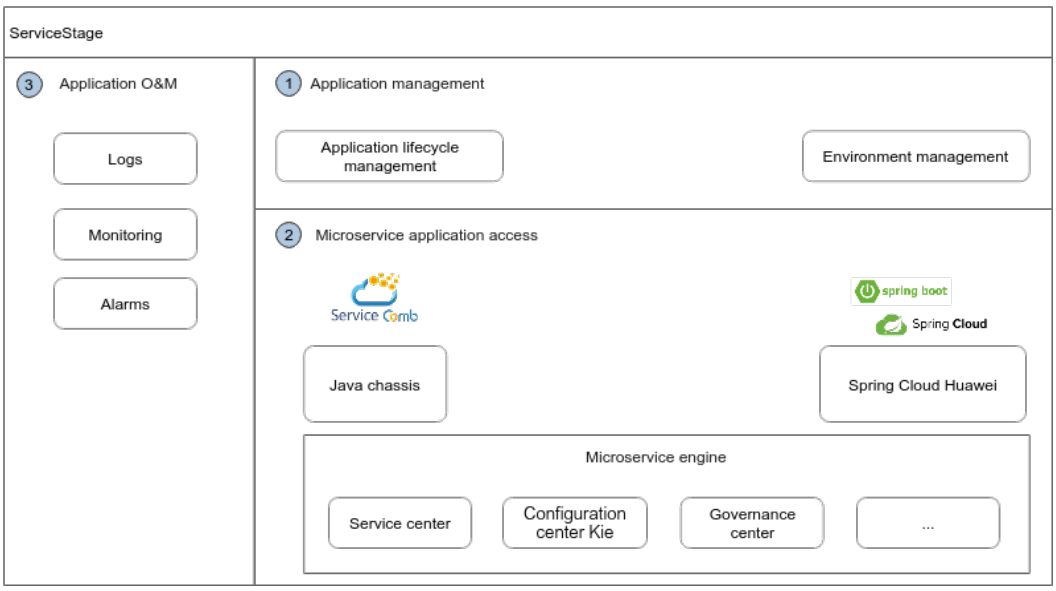
2 What Is ServiceStage?

ServiceStage is an application management and O&M platform that lets you deploy, roll out, monitor, and maintain applications all in one place. It supports technology stacks such as Java, PHP, Python, Node.js, Docker, and Tomcat, and supports microservice applications such as Apache ServiceComb Java Chassis (Java chassis) and Spring Cloud, making it easier to migrate enterprise applications to the cloud.

ServiceStage provides the following capabilities:

1. Application management: application lifecycle management and environment management.
2. Microservice application access: Java chassis, and Spring Cloud. Furthermore, ServiceStage works with CSE to implement service registration and discovery, configuration management, and service governance. For details, see **Microservice Development Guide**.
3. AOM: supports application O&M through logs, monitoring, and alarms.

Figure 2-1 ServiceStage functions



Application Management

- Application lifecycle management

  After an application is developed, it can be hosted on ServiceStage, which provides you with complete application lifecycle management:

  - Application creation by using the source code, software packages (JAR, WAR, or ZIP), and container images, achieving application deployment.
  - Entire process management from application creation to logout, covering application creation, deployment, start, upgrade, rollback, scaling, stop, and deletion.

- Environment management

  An environment is a collection of compute, network, and middleware resources used for deploying and running an application component. ServiceStage combines the compute resources (such as CCE clusters, CCI instances, and ECSs), network resources (such as ELB instances and EIPs), and middleware (such as DCS instances, RDS instances, and CSE engines) into an environment, such as a development environment, testing environment, pre-production environment, or production environment.

  The resources within an environment can be networked together. Managing resources and deploying services by environment simplifies O&M.

Microservice Application Access

The microservice engine of ServiceStage supports access and governance of mainstream microservice frameworks. You can select a suitable microservice technology to quickly develop cloud applications to achieve complex and ever-changing service requirements.

- Supports the native ServiceComb microservice framework.

  Microservices developed by using the ServiceComb framework can be seamlessly connected to microservice engines.

  The microservice engine uses Apache ServiceComb Service Center, which is a RESTful-style, high-availability, and stateless service registry and discovery center and provides microservice discovery and management. Service providers can register their instance information with the registry and discovery center for users to discover and use. For details about Apache ServiceComb Service Center, see the following:

  - **https://github.com/apache/servicecomb-service-center/**
  - **https://service-center.readthedocs.io/en/latest/user-guides.html**

- Compatible with mainstream microservice open-source frameworks.

  Provides a simple access mode for microservices developed by using Spring Cloud. You only need to modify the dependencies and configurations to access microservice engines and use the unified governance policies.

- Provides microservice governance.

  After an application developed using the microservice framework is managed on ServiceStage, the microservice will be registered with the service registry and discovery center after the application instance starts. You can govern microservices by referring to **Microservice Governance**.

Application O&M

- Multi-dimensional metrics monitoring for application components, helping you understand the running status of online applications.
- GUI-based log query and search, helping you quickly locate faults.

3 Product Advantages

ServiceStage integrates successful experience in cloud transformation and technological innovation. As a one-stop application cloud platform, it has the following advantages over traditional platforms:

Table 3-1 Product advantages

Application Lifecycle	Traditional Platform	ServiceStage
Environment preparation	• Low resource obtaining efficiency (> 1 day) • Low resource utilization (< 30%)	• Self-service and efficient resource obtaining (minute-level) • Pay-per-use payment (auto scaling)

Application Lifecycle	Traditional Platform	ServiceStage
Service development	• Architecture coupled, a small change requires significant reconstruction • Single technology, one technology is required to resolve all problems • System release at a large granularity, requiring long response period	• Architecture decoupled Open API-based development makes the development, test, document, collaboration, and control activities of microservices are standardized and automated. • Flexible access of various technologies Supports Java, PHP, Python, and Node.js. The high-performance REST/RPC microservice development framework provides out-of-the-box tools to reduce the development threshold. Provides the ServiceComb, Spring Cloud, and service mesh commercial editions. • Agile The one-stop microservice governance console provides governance capabilities for microservices, such as load balancing, rate limiting, degradation, circuit breaker, fault tolerance, and fault injection. Supports microservice upgrade dark launch.

Application Lifecycle	Traditional Platform	ServiceStage
Installation and deployment	• Siloed system • Manual deployment	Developers only need to use ServiceStage and source code software repository to implement one-click automatic deployment and update.
Application upgrade	• Patch installation • Manual upgrade • Services interrupted	During rolling upgrades, services are evenly distributed to new and old instances; therefore, services are not interrupted. Dark launch allows you to select users to test your beta versions, reducing launch risks and ensuring smooth rollout of new features.
Application O&M	• Application breakdown or crash • Slow service response • Insufficient system resources • Difficult fault locating	• Real-time graphical display of application monitoring metrics CPU usage, alarms, node exceptions, running logs, and key events can be monitored in real time • Microservice governance Supports microservice API-level SLA metrics (throughput, latency, and success rate) monitoring and governance in real time (in seconds), ensuring continuous service running.

4 Application Scenario

4.1 Constructing Microservice Applications

Application Scenarios

Scenario

Different service modes of traditional projects in a single architecture must adopt a unified technical solution and technical platform. Each service module cannot be reused. If a module in the entire system is faulty, the entire system becomes unavailable. With the increasing complexity of enterprise services, the traditional monolithic architecture becomes more and more cumbersome, and it is difficult to adapt to flexible service requirements. Microservice applications can solve these problems.

Value

Microservice-based applications allow enterprises to divide a cumbersome system into several small service components. Among which, these components communicate with each other through lightweight protocols, decoupling the lifecycle management of each component.

Ever growing services may encounter various unexpected situations, such as instantaneous and large-scale concurrent access, service errors, and intrusion. The microservice architecture can be used to implement fine-grained service management and control to support service requirements.

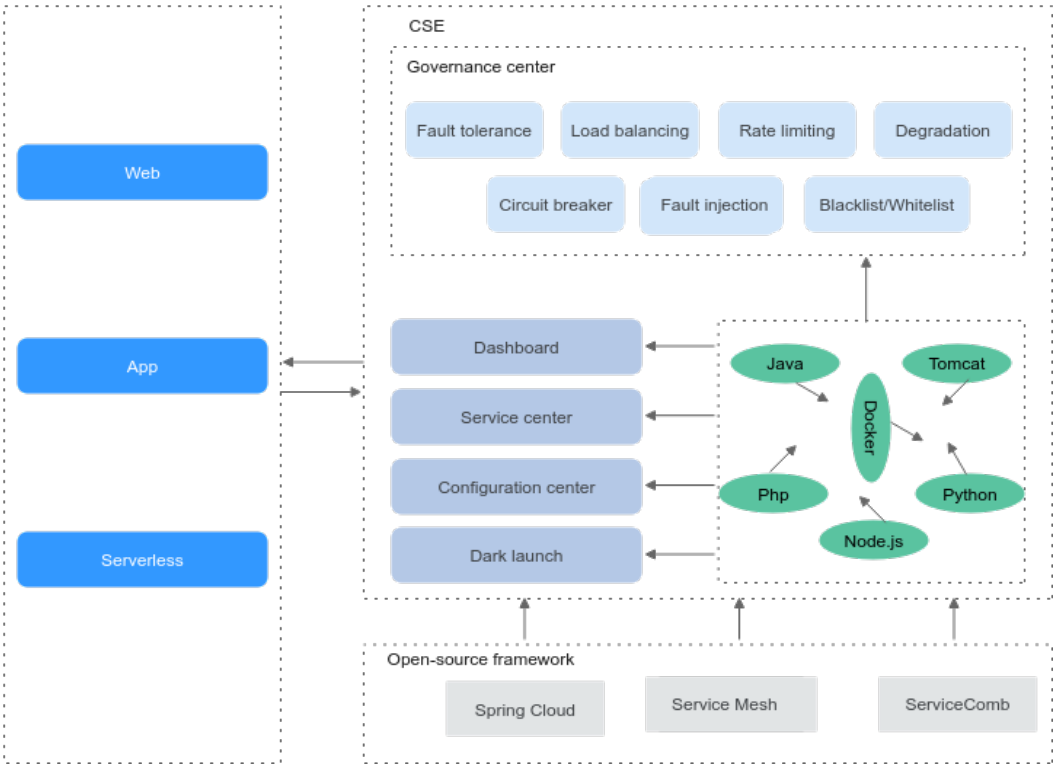
ServiceStage supports full lifecycle management of microservice applications. It supports stacks such as Java, PHP, Python, Node.js, Docker, and Tomcat, and manages microservice applications such as Apache ServiceComb Java chassis and Spring Cloud without intrusion. In addition, it provides functions such as configuration management, monitoring and O&M, and service governance, making it easier to migrate enterprise microservice applications to the cloud.

Advantage

ServiceStage provides microservice application solutions and has the following advantages:

- Supports multiple microservice frameworks, such as native ServiceComb and Spring Cloud, and supports the dual-stack mode (SDK and service mesh interconnection). The service code can be directly managed on the cloud without modification.
- API First, which supports Swagger-based API management.
- Supports multiple languages, such as Java, PHP, Python, and Node.js.
- Provides functions such as service center, configuration center, dashboard, and dark launch.
- Provides complete microservice governance policies, including load balancing, fault tolerance, rate limiting, service degradation, circuit breaker, fault injection, and blacklist and whitelist. GUI-based operations can be performed for different service scenarios, greatly improving the availability of service governance.

Figure 4-1 Microservice application solution



- Implements mutual discovery between Spring Cloud and Java chassis.

Continuous Integration and Delivery

Scenario

It takes a lot of manpower and time in project creation, compilation, build, self-verification, integration verification, production environment-like verification, and rollout for a complex system, which is prone to errors caused by human factors. Continuous integration and delivery can resolve such problems due to its standardization and automation.

Value

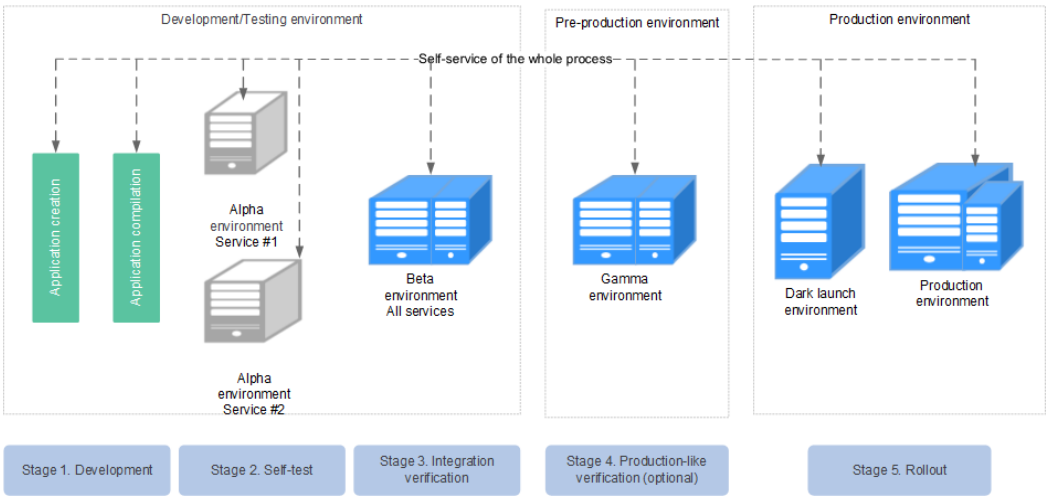
Manual execution is changed to automatic execution, which reduces errors and improves work efficiency.

The environment and process standards are unified, which facilitates service expansion and reduces upgrade and reconstruction costs.

Advantage

Based on the ServiceStage pipeline, the integration environment is unified and the delivery process is standardized. You can implement the self-service development, self-verification, integration verification, and rollout.

Figure 4-2 Continuous integration and delivery



Dark Launch

Scenario

In dark launch, users are selected to test the beta version, ensuring smooth rollout of new features. Once new features become mature and stable, a formal version is released for all users to use.

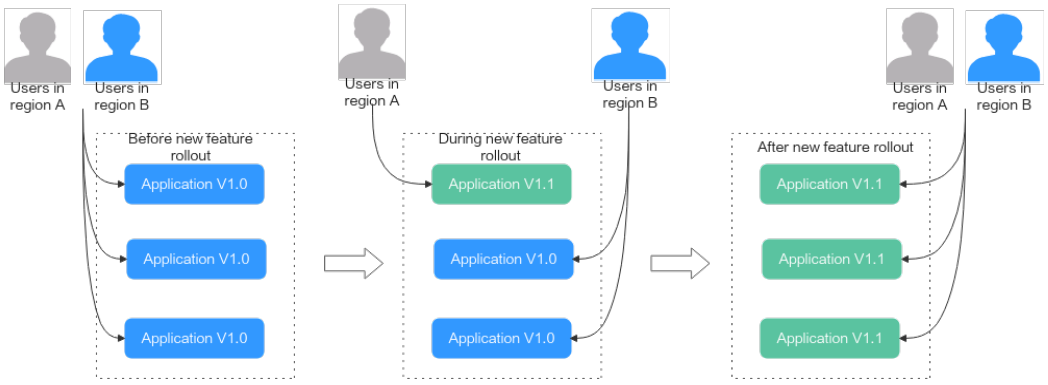
Value

Dark launch ensures stability of the entire system. During initial dark launch, problems can be detected and fixed.

Advantage

ServiceStage supports dark launch.

Figure 4-3 Dark launch



4.2 Web Application Lifecycle Management

Application Scenarios

Scenario

Web applications are widely used. Enterprise service systems, online store systems, forums, blogs, Wiki systems, and online games may be web applications. Managing the lifecycle of web applications with different technical architectures is one of the main tasks of the enterprise IT department.

Value

Using a unified platform to manage various web applications can greatly reduce workload, improve efficiency, and quickly respond to complex and changeable service requirements.

Advantage

ServiceStage greatly improves the efficiency of enterprise-level web application development and O&M, making enterprises focus on service innovation. It has the following advantages:

- Deployment with a few clicks using WAR, JAR, or ZIP packages.
- One-stop O&M provides various O&M capabilities, such as upgrade, rollback, log, monitoring, and auto scaling.
- Seamless integration supports seamless integration with cloud services and applications such as ELB, RDS, and DCS.

5 Security

5.1 Shared Responsibilities

Huawei guarantees that its commitment to cyber security will never be outweighed by the consideration of commercial interests. To cope with emerging cloud security challenges and pervasive cloud security threats and attacks, Huawei Cloud builds a comprehensive cloud service security assurance system for different regions and industries based on Huawei's unique software and hardware advantages, laws, regulations, industry standards, and security ecosystem.

Unlike traditional on-premises data centers, cloud computing separates operators from users. This approach not only enhances flexibility and control for users but also greatly reduces their operational workload. For this reason, cloud security cannot be fully ensured by one party. Cloud security requires joint efforts of Huawei Cloud and you, as shown in **Figure 5-1**.

- **Huawei Cloud**: Huawei Cloud is responsible for infrastructure security, including security and compliance, regardless of cloud service categories. The infrastructure consists of physical data centers, which house compute, storage, and network resources, virtualization platforms, and cloud services Huawei Cloud provides for you. In PaaS and SaaS scenarios, Huawei Cloud is responsible for security settings, vulnerability remediation, security controls, and detecting any intrusions into the network where your services or Huawei Cloud components are deployed.
- **Customer**: As our customer, your ownership of and control over your data assets will not be transferred under any cloud service category. Without your explicit authorization, Huawei Cloud will not use or monetize your data, but you are responsible for protecting your data and managing identities and access. This includes ensuring the legal compliance of your data on the cloud, using secure credentials (such as strong passwords and multi-factor authentication), and properly managing those credentials, as well as monitoring and managing content security, looking out for abnormal account behavior, and responding to it, when discovered, in a timely manner.

Figure 5-1 Huawei Cloud shared security responsibility model

Cloud security responsibilities are determined by control, visibility, and availability. When you migrate services to the cloud, assets, such as devices, hardware, software, media, VMs, OSs, and data, are controlled by both you and Huawei Cloud. This means that your responsibilities depend on the cloud services you select. As shown in **Figure 5-1**, customers can select different cloud service types (such as IaaS, PaaS, and SaaS services) based on their service requirements. As control over components varies across different cloud service categories, the responsibilities are shared differently.

- In on-premises scenarios, customers have full control over assets such as hardware, software, and data, so tenants are responsible for the security of all components.
- In IaaS scenarios, customers have control over all components except the underlying infrastructure. So, customers are responsible for securing these components. This includes ensuring the legal compliance of the applications, maintaining development and design security, and managing vulnerability remediation, configuration security, and security controls for related components such as middleware, databases, and operating systems.
- In PaaS scenarios, customers are responsible for the applications they deploy, as well as the security settings and policies of the middleware, database, and network access under their control.
- In SaaS scenarios, customers have control over their content, accounts, and permissions. They need to protect their content, and properly configure and protect their accounts and permissions in compliance with laws and regulations.

5.2 Authentication and Access Control

Authentication

Users can access ServiceStage through its console or RESTful APIs. In essence, requests are sent through REST APIs provided by ServiceStage. You can use either of the following authentication methods to call APIs:

- Token authentication: Requests are authenticated using tokens. During token-based API authentication, the token is added to requests to get permissions for calling the API.
- AK/SK-based authentication: Requests are encrypted using an AK/SK pair A request must contain a signature value. The signature value is calculated based on the AK/SK of the requester and the specific information carried in the request body. AK/SK authentication is used to authenticate the identity of a request sender. For more information about access keys and how to obtain them, see **Access Keys**.

Access Control

ServiceStage helps you set different access permissions for employees in an enterprise to isolate permissions of different employees. You can use Identity and Access Management (IAM) to implement fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, enabling secure access to your cloud resources.

With IAM, you can use your public cloud account to create IAM users for your employees, and assign permissions to the users to control their access to specific resource types. For example, some software developers in your enterprise need to use ServiceStage resources but must not delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using ServiceStage resources.

- For more information about IAM, see **IAM Service Overview**.
- For details about the system permissions supported by ServiceStage, see **Permissions Management**.

5.3 Data Protection

To ensure that your personal data, such as the username and password, will not be obtained by unauthorized or unauthenticated entities or individuals, ServiceStage encrypts your data during storage and transmission to prevent data leakage.

Personal Data

Table 5-1 lists the personal data generated or collected by ServiceStage.

Table 5-1 Personal data collected

Type	Source	Description	Modifiable	Mandatory
Repository authorization (including the user name, password, OAuth authorization, and private token. Different repositories support different authorization modes.)	When creating repository authorization, you need to enter the repository authorization information on the console.	Used to access the code repository and pull code from the code repository during code building.	The administrator permissions can be modified through the API.	Yes

Data Storage Security

Repository authorization information is encrypted and stored using the common security component and AES algorithm.

5.4 Audit and Logs

Audit

Cloud Trace Service (CTS) records operations performed on cloud resources in your account. The operation logs can be used to perform security analysis, track resource changes, perform compliance audits, and locate faults.

After you enable CTS and create a tracker, CTS starts to record operations for audit. For details, see **CTS Getting Started**.

After CTS is enabled, you can view **Viewing IAM Audit Logs**. CTS stores operation logs of the last seven days. For details about ServiceStage operations that can be tracked by CTS, see **ServiceStage Operations That Can Be Recorded by CTS**.

CTS allows you to **Configuring Key Event Notifications**. You can add high-risk and sensitive operations related to ServiceStage to real-time monitoring of CTS as key operations. When you use ServiceStage, if a key operation in the monitoring list is triggered, CTS records the operation log and sends a notification in real time.

Logs

ServiceStage supports the Application Operations Management (AOM) capability. You can view application run logs or view related run logs on the AOM console.

For details, see **Configuring Application Log Policies**.

5.5 Resilience

- Redundancy: All services are stateless. Requests are distributed to different instances to implement load balancing based on the server load balancing capability provided by the IaaS layer. API gateways are used to provide services for downstream systems. The gateways use methods such as rate limiting, circuit breaker, and service degradation to ensure that services are not interrupted during upgrades.
- Cross-AZ DR: Services are evenly distributed in different AZs and can be rebuilt across AZs. If instances do not exist in an AZ, the IaaS layer schedules new instances to other AZs to ensure that the system is not overloaded. In the single-AZ failure scenario, the integrity of persistent data can still be ensured.
- For the IaaS layer, monitoring metrics such as the CPU, memory, network, and disk are provided. For the application layer, information such as metrics and logs is automatically reported. In addition, alarms are generated when critical problems occur.
- Each service has the rate limiting capability and will not be overwhelmed by heavy traffic.
- Services use the container environment, and the IaaS layer provides lifecycle management and container scheduling. When a crash occurs, instances will be scheduled and new ones will work.

5.6 Security Risk Monitoring

ServiceStage supports the configuration of application log policies. You can view related run logs on the Application Operations Management (AOM) console. For details, see **Configuring Application Log Policies**.

ServiceStage allows you to configure application performance management during or after application component deployment. Application Performance Management (APM) helps you quickly locate application problems and analyze performance bottlenecks. For details, see **Configuring Application Performance Management**.

6 Microservice Engine Versions

This section describes the versions supported by exclusive microservice engines.

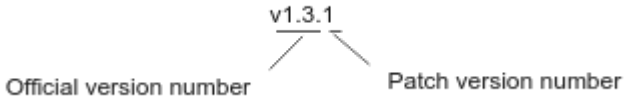
Version Description

The version number format is {major}.{minor}.{patch}.

where,

- {major}.{minor} indicates the official version number.
- {patch} indicates the patch version number.

For example, v1.3.1. 1.3 is the official version number, and 1 is the patch version number.



Version Support

- Microservice engine creation

  Only the microservice engine of the latest version can be created. The microservice engine of a specified version cannot be created.

- Microservice engine maintenance

  The latest three official versions can be maintained. For other versions, customer service will not be provided, including new functions, bug fixing, vulnerability fixing, and upgrades.

- Microservice engine upgrade
  - Official version upgrade:
    - 1.3 and 1.2 can be smoothly upgraded to 2.4 or later, and their functions are compatible.
    - Two earlier versions among the latest three official versions can be upgraded to the latest version. For example, if the latest three official versions are 2.4, 1.3, and 1.2, 1.2 and 1.3 are upgraded to 2.4.

If the engine upgrade is not supported, for example, from 1.0 to 1.3, the management function of microservice engines may be unavailable. Exercise caution when performing this operation. You can **submit a service ticket** for risk evaluation before the upgrade.

- Patch version upgrade: The microservice engine backend provides automatic patch version upgrade, for example, from 1.3.0 to 1.3.1.

Version Constraints

Version rollback is not supported after the microservice engine version is upgraded.

7 Restrictions

ServiceStage has the following restrictions, and each of them applies to every tenant in any region.

Restriction is not resource quota limit. It indicates the maximum capabilities that ServiceStage can provide for tenants. Users need to pay attention to these restrictions when selecting technologies and designing solutions.

VM-based Deployment

When application components are deployed on VMs, ServiceStage manages a maximum of 5000 VM agents, and 1500 of them can directly communicate with ServiceStage without using VM environment proxies.

Registry and Discovery

For details about the restrictions on professional microservice engines, see **Table 7-1**.

Table 7-1 Restrictions on professional microservice engines

Item	Restrictions
Heartbeat reporting	Every 30s at most for every microservice instance
Service discovery	Every 30s at most for every microservice instance
Microservice instance registration	10 per second

For details about the restrictions on exclusive microservice engines, see **Table 7-2**.

Table 7-2 Restrictions on exclusive microservice engines (maximum specifications)

Item	Restrictions	Remarks
Heartbeat reporting	Every 20s at most for every microservice instance	Total rate limit: 2000 TPS
Service discovery	Every 20s at most for every microservice instance	-
Microservice instance registration	1000 per second	-

Exclusive Microservice Engine Types

An exclusive microservice engine of version 1.3.0 or later supports engine types listed in the following table. You can select an exclusive microservice engine of the required engine type.

Table 7-3 Exclusive microservice engine types

Engine Type	Application Scenario	AZ CPU Architecture
Cluster-deployed	Cluster-deployed engines support HA and host-level DR.	x86 or ARM. The hybrid architecture is not supported.

Microservice Development Framework Versions

The following table lists the recommended versions of the microservice development framework.

- If you have used the microservice development framework of an earlier version to build applications, you are advised to upgrade it to the recommended version to obtain the stable and rich function experience.
- If an application has been developed using the Spring Cloud microservice development framework, you are advised to use **Spring Cloud Huawei** to access the application.
- If new microservice applications are developed based on open source and industry ecosystem components, you are advised to use the Spring Cloud framework.
- If you want to use the out-of-the-box governance capability and high-performance RPC framework provided by microservice engines, you are advised to use the Java chassis framework.

Framework	Recommended Versions	Description
Spring Cloud Huawei	1.10.9-2021.0.x or later	Uses **Spring Cloud Huawei** for connection. • Spring Cloud version 2021.0.5 • Spring Boot 2.6.13 Version description of the Spring Cloud microservice development framework: **https://github.com/huaweicloud/spring-cloud-huawei/releases**
Java Chassis	2.7.10 or later	Uses the software package provided by the open-source project for connection without introducing third-party software packages. Version description of the Java chassis microservice development framework: **https://github.com/apache/servicecomb-java-chassis/releases**.

NOTICE

During system upgrade and reconstruction, third-party software conflict is the most common issue. Traditional software compatibility management policies do not adapt to software development for fast software iteration. In this case, see **Third-Party Software Version Management Policy** for version compatibility.

Exclusive Microservice Engine Quota

A quota refers to the maximum number of resources that you can create in an exclusive microservice engine instance. **Table 7-4** provides the resource quota.

Table 7-4 Resource quota limits of microservice engines

Function	Resource	Quota	Modifiable	Precaution
Microservice management	Microservice versions	10,000	No	-
	Data volume of a single instance (KB)	200	Yes	Increasing quotas prolongs the microservice discovery latency.
	Number of contracts of a single microservice	500	No	-

Function	Resource	Quota	Modifiable	Precaution
Configuration management	Data volume of a single configuration item (KB)	128	No	-
	Data volume of an application-level configuration	2,000	No	-
Microservice governance	Application-level governance policies	1,000	No	A maximum of 1000 governance policies are supported.

NOTE

- A single governance policy contains governance rules and service scenarios. Governance rules and service scenarios occupy the same quota in the configuration center.
- Microservice version: In the microservice scenario, a version is used to mark the iteration record of a microservice to facilitate management of different iterations of a microservice.
- Microservice instance: An instance is the minimum running and deployment unit of a microservice. Generally, it corresponds to an application process. A microservice can be deployed in multiple containers or VMs to enable multiple instances to run at the same time.
- Configuration item: The configuration in the microservice scenario is to control the values of some variables in the program code. For example, dynamic configuration is to dynamically change the values of some variables during microservice running.

8 Edition Differences

Product Package Description

ServiceStage provides the basic edition and professional edition. You can select your edition as required. For details about functions of each edition, see **Table 8-1**.

Table 8-1 Functions

Function		Basic Edition	Professional Edition
Management scale	Maximum number of application component instances supported by an IAM account	20 instances are free of charge	More than 100 instances are supported
	Maximum number of instances supported by a component	200	
	Maximum number of configuration items	100	300
Application lifecycle management	Multi-language application management (Java, PHP, Python, Node.js, Tomcat, and Docker)	Supported	Supported
	Application lifecycle management (dark launch, auto scaling, upgrade, rollback, start, stop, restart, and deletion)		
	Basic application monitoring (running status, CPU, memory, and disk usage)		
	Component deployment using VM		
	Component deployment using CCE		
	Access control		

Function		Basic Edition	Professional Edition
	Application domain name management		
	Auto scaling		
	Affinity and anti-affinity deployment		
	Event analysis		
	Log analysis		
	Metric management		
	Threshold rules		
Continuous delivery	Build management	Supported	Supported
	Source code repository (GitHub, GitLab, and Bitbucket)		
	Compilation task (Java, PHP, Python, Node.js, Tomcat, and Docker)		
	Cluster build		
	Pipeline management		
Software repository	SWR software package management	Supported	Supported
	Docker image package management		
	Repository permission management		
Container resources	VM cluster	Supported	Supported
	BMS cluster		
	Windows cluster		
	Container node management		
	Container storage management		
Application orchestration	Stack management Template management Charts Service catalog Designer	Supported	Supported

Function		Basic Edition	Professional Edition
Microservice	Java microservice development SDK	Supported	Supported
	Spring Cloud microservice access		
	Registry center		
	Configuration center		
	Real-time dashboard		
	Load balancing		
	Rate limiting		
	Service degradation		
	Fault tolerance		
	Circuit breaker		
	Fault injection		
	Blacklist and whitelist		
	Dark launch		
	Exclusive microservice engine		
Application performance management	Automatic topology discovery	Not supported	Not supported
	Application transaction analysis		
	Application KPIs (service throughput, error rate, latency, and load status)		
	Slow SQL analysis		
	Intelligent alarm		
	SQL performance analysis		
	Tracing		
	Non-intrusive collection		
Support	Service manager	Not supported	Not supported
	Remote technical support engineers		
	Onsite support for direct troubleshooting		

CSE Instance Specifications

Microservice engines have professional and exclusive editions.

- Professional microservice engine: Cloud Service Engine (CSE) is a free experience engine provided by ServiceStage. You can use a professional engine to experience all product capabilities of ServiceStage, such as service governance and configuration management. Engines are shared by all tenants; however, the performance may be affected by other tenants. A professional engine cannot be upgraded to the exclusive edition.
- Exclusive microservice engine: Exclusive engines are commercial engines that manage large-scale microservice applications. You can select different engine specifications based on service requirements, and specifications cannot be changed. Exclusive engines are exclusively used; therefore, the performance is not affected by tenants.

The following describes the maximum number of instances supported by CSE.

Table 8-2 CSE instance specifications

Type	Microservice Instances	Configuration Items
Professional microservice engine	20	-
Exclusive microservice engine	100	600
	200	600
	500	3,000
	2,000	12,000

9 Permissions Management

If you need to grant your enterprise personnel permission to access your ServiceStage resources, use Identity and Access Management (IAM). IAM provides identity authentication, fine-grained permissions management, and access control. IAM helps you secure access to your cloud resources.

With IAM, you can create IAM users and grant them permission to access only specific resources. For example, if you want some software developers in your enterprise to be able to use ServiceStage resources but do not want them to be able to delete ServiceStage resources or perform any other high-risk operations, you can create IAM users and grant permission to use ServiceStage resources but not permission to delete them.

If your cloud account does not require individual IAM users for permissions management, you can skip this section.

IAM is free of charge. You pay only for the resources in your account. For details, see **IAM Service Overview**

System Permissions

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and then attach policies or roles to these groups. The users then inherit permissions from the groups and can perform specified operations on cloud services based on the permissions they have been assigned.

ServiceStage is a project-level service deployed for specific regions. To assign ServiceStage permissions to a user group, specify the scope as region-specific projects and select projects for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing ServiceStage, the users need to switch to the authorized region.

You can grant permissions by using roles and policies.

- Roles: A coarse-grained authorization strategy that defines permissions by job responsibility. Only a limited number of service-level roles are available for authorization. When you grant permissions using roles, you also need to attach any existing role dependencies. Roles are not ideal for fine-grained authorization and least privilege access.

- Policies: A fine-grained authorization strategy that defines permissions required to perform operations on specific cloud resources under certain conditions. This type of authorization is more flexible and is ideal for least privilege access.

Table 9-1 lists all the system-defined policies for ServiceStage. System policies are recommended. System roles are used only for compatibility with existing permission configurations.

Table 9-1 ServiceStage system permissions

Role/Policy Name	Description	Type	Dependency
ServiceStage FullAccess	Full permissions for ServiceStage.	System-defined policy	None
ServiceStage ReadOnlyAccess	Read-only permissions for ServiceStage.	System-defined policy	None
ServiceStage Development	ServiceStage developer, including permissions for operating applications, components, and environments, but excluding permissions for approving and for creating infrastructure.	System-defined policy	None
CSE FullAccess	Administrator permissions for microservice engines.	System-defined policy	None
CSE ReadOnlyAccess	View permissions for microservice engines.	System-defined policy	None
ServiceStage Administrator	ServiceStage administrator, who has full permissions for this service.	System-defined role	Permissions to create Tenant Guest, Server Administrator, CCE Administrator, and APM Administrator.
ServiceStage Operator	ServiceStage operator, who has the read-only permission for this service.	System-defined role	Tenant Guest

Role/Policy Name	Description	Type	Dependency
ServiceStage Developer	ServiceStage developer, who has full permissions for this service but does not have the permission for creating infrastructure.	System-defined role	Tenant Guest

If these policies do not meet actual requirements, you can customize policies based on **Table 9-2** and **Table 9-3**. For more information, see **Creating a Custom Policy**.

√: supported; x: not supported.

Table 9-2 Common ServiceStage operations supported by each system policy

Operation	ServiceStage ReadOnlyAccess	ServiceStage Development	ServiceStage FullAccess
Create an application	x	√	√
Modify an application	x	√	√
Query an application	√	√	√
Delete an application	x	√	√
Create a component	x	√	√
Search for a component	√	√	√
Deploy a component	x	√	√
Maintain a component	x	√	√

Operation	ServiceStage ReadOnlyAccess	ServiceStage Development	ServiceStage FullAccess
Delete a component	x	√	√
Create a build job	x	√	√
Modify a build job	x	√	√
Query a build job	√	√	√
Start a build job	x	√	√
Delete a build job	x	√	√
Create a pipeline	x	√	√
Modify a pipeline	x	√	√
Query a pipeline	√	√	√
Start a pipeline	x	√	√
Clone a pipeline	x	√	√
Delete a pipeline	x	√	√
Create repository authorization	x	√	√
Modify repository authorization	x	√	√
Query repository authorization	√	√	√

Operation	ServiceStage ReadOnlyAccess	ServiceStage Development	ServiceStage FullAccess
Delete repository authorization	x	√	√

Table 9-3 Common CSE operations supported by each system policy

Operation	CSE ReadOnlyAccess	CSE FullAccess
Create a microservice engine	x	√
Maintain a microservice engine	x	√
Query a microservice engine	√	√
Delete a microservice engine	x	√
Create a microservice	x	√
Query a microservice	√	√
Maintain a microservice	x	√
Delete a microservice	x	√
Create microservice configurations	x	√
Query microservice configurations	√	√
Edit microservice configurations	x	√
Delete microservice configurations	x	√
Create a microservice governance policy	x	√
Query a microservice governance policy	√	√
Edit a microservice governance policy	x	√
Delete a microservice governance policy	x	√

Fine-grained Permissions

NOTE

- SWR does not support fine-grained permissions. Related permissions need to be authorized separately.
- When an exclusive microservice engine is created and its **Billing Mode** is set to **Yearly/monthly**:
  - If you do not pay for orders, you must have the BSS Operator permission (queries cost analysis, budget details, and cost tags in the Cost Center).
  - If you pay for orders, you must have the BSS Administrator permission (performs all operations in the Cost Center).

To use a custom fine-grained policy, log in to the IAM console as an administrator and select the desired fine-grained permissions for ServiceStage and CSE.

- **Table 9-4** describes fine-grained permission dependencies of CSE.
- **Table 9-5** describes fine-grained permission dependencies of ServiceStage.

Table 9-4 Fine-grained permission dependencies of CSE

Permission Name	Description	Dependencies	Scenario
cse:engine:list	List all microservice engines	None	View engine list
cse:engine:get	View engine information	cse:engine:list	View engine details (supported by only exclusive microservice engines)
cse:engine:modify	Modify an engine	• cse:engine:list • cse:engine:get	Modify an engine, including enabling or disabling public access, enabling or disabling security authentication, and retrying failed tasks, supported by only exclusive microservice engines
cse:engine:upgrade	Upgrade an engine	• cse:engine:list • cse:engine:get	Engine upgrade, including upgrading the engine version, supported by only exclusive microservice engines.

Permission Name	Description	Dependencies	Scenario
cse:engine:delete	Delete an engine	• cse:engine:list • cse:engine:get • vpc:ports:get • vpc:ports:delete	Delete an engine (supported by only exclusive microservice engines)
cse:engine:create	Create an engine	• cse:engine:get • cse:engine:list • ecs:cloudServerFlavors:get • vpc:vpcs:get • vpc:vpcs:list • vpc:subnets:get • vpc:ports:get • vpc:ports:create	Create an engine, including creating an engine and creating a backup or restoration task, supported by only exclusive microservice engines
cse:config:modify	Modify configuration management	• cse:engine:list • cse:engine:get • cse:config:get	Modify global and governance configurations
cse:config:get	View configuration and management functions	• cse:engine:list • cse:engine:get	View service configurations
cse:governance:modify	Modify the governance center	• cse:engine:list • cse:engine:get • cse:config:get • cse:config:modify • cse:registry:get • cse:registry:modify • cse:governance:get	Create and modify a governance policy
cse:governance:get	View the governance center	• cse:engine:list • cse:engine:get • cse:config:get • cse:registry:get	View service governance
cse:registry:modify	Modify service registry and management	• cse:engine:list • cse:engine:get • cse:registry:get	Modify a service

Permission Name	Description	Dependencies	Scenario
cse:dashboard:modify	Modify dashboard management	• cse:engine:list • cse:engine:get • cse:registry:get • cse:dashboard:get • cse:registry:modify	Modify dashboard
cse:dashboard:get	View dashboard management	• cse:engine:list • cse:engine:get • cse:registry:get	View dashboard
cse:registry:get	View service registry and management	• cse:engine:list • cse:engine:get	View service catalog

NOTE

The dashboard does not need to be authenticated but requires registry permissions, because it uses the service catalog function to distinguish services.

Table 9-5 Fine-grained permission dependencies of ServiceStage

Permission Name	Description	Dependencies	Scenario
servicestage:app:get	View application information	servicestage:app:list	View application information
servicestage:app:create	Create an application	• servicestage:app:get • servicestage:app:list • servicestage:assembling:get • servicestage:assembling:list • servicestage:assembling:create	Create an application

Permission Name	Description	Dependencies	Scenario
servicestage:app:modify	Update an application	• servicestage:app:get • servicestage:app:list • servicestage:assembling:get • servicestage:assembling:list • servicestage:assembling:modify	Update an application
servicestage:app:delete	Delete an application	• servicestage:app:get • servicestage:app:list • servicestage:assembling:delete	Delete an application
servicestage:app:list	View the environment and application list	None	View the environment and application list
servicestage:environment:create	Create an environment	• servicestage:app:get • servicestage:app:list	Create an environment
servicestage:environment:modify	Update an environment	• servicestage:app:get • servicestage:app:list	Update an environment
servicestage:environment:delete	Delete an environment	• servicestage:app:get • servicestage:app:list	Delete an environment
servicestage:pipeline:get	View pipeline information	• servicestage:pipeline:list • servicestage:assembling:get • servicestage:assembling:list	View pipeline information

Permission Name	Description	Dependencies	Scenario
servicestage:pipeline:create	Create a pipeline	• servicestage:pipeline:list • servicestage:pipeline:get • servicestage:assembling:create • servicestage:assembling:get • servicestage:assembling:list	Create a pipeline
servicestage:pipeline:modify	Modify a pipeline	• servicestage:pipeline:get • servicestage:pipeline:list • servicestage:assembling:modify • servicestage:assembling:get • servicestage:assembling:list	Modify a pipeline
servicestage:pipeline:delete	Delete a pipeline	• servicestage:pipeline:get • servicestage:pipeline:list • servicestage:assembling:get • servicestage:assembling:list • servicestage:assembling:delete	Delete a pipeline
servicestage:pipeline:list	View the pipeline list	• servicestage:assembling:get • servicestage:assembling:list	View the pipeline list

Permission Name	Description	Dependencies	Scenario
servicestage:pipeline:execute	Execute a pipeline	• servicestage:pipeline:get • servicestage:pipeline:list • servicestage:assembling:modify • servicestage:assembling:get • servicestage:assembling:list • servicestage:app:get • servicestage:app:list • servicestage:app:modify	Execute a pipeline
servicestage:assembling:get	View the build information	servicestage:assembling:list	View the build information
servicestage:assembling:create	Create a build job	• servicestage:assembling:get • servicestage:assembling:list	Create a build job
servicestage:assembling:modify	Modify a build job	• servicestage:assembling:get • servicestage:assembling:list	Modify a build job
servicestage:assembling:delete	Delete a build job	• servicestage:assembling:get • servicestage:assembling:list	Delete a build job
servicestage:assembling:list	View the build list	None	View the build list

Roles/Policies Dependencies of ServiceStage Console

To grant an IAM user the permissions to view or use resources of other cloud services on the ServiceStage console, you must first grant the ServiceStage Administrator, ServiceStage FullAccess, or ServiceStage ReadOnlyAccess policy to the user group to which the user belongs and then grant the dependency policies listed in **Table 9-6** to the user. These dependency policies will allow the IAM user to access resources of other cloud services.

Table 9-6 Roles/Policies dependencies of ServiceStage console

Console Function	Dependent Services	Policy/Role Required
• Dashboard • Alarms • O&M and monitoring	Application Operations Management (AOM)	• An IAM user with the ServiceStage Administrator permission assigned can use this function only after the AOM FullAccess permission is assigned. • IAM users with IAM ReadOnlyAccess, ServiceStage FullAccess, or ServiceStage ReadOnlyAccess assigned can directly use this function.
Performance management	Application Performance Management (APM)	To use a Java probe, you must have the AOM FullAccess and APM FullAccess permissions assigned.
Component management	Auto Scaling (AS)	To use AS resources to deploy components in the VM environment, you must have the AutoScaling FullAccess permissions assigned.
	Cloud Container Engine (CCE)	To use CCE resources to deploy components in the container environment, you must have the CCE FullAccess permissions assigned.
	Elastic Cloud Server (ECS)	To use ECS resources to deploy components in the VM environment, you must have the ECS ReadOnlyAccess permissions assigned.
	Object Storage Service (OBS)	If the component to be deployed comes from the software package stored in OBS, you must have the OBS ReadOnlyAccess permissions assigned.
Microservice engine	Cloud Service Engine (CSE)	To bind CSE to microservice components for service registration, service governance, and configuration management, you must have the CSE FullAccess permissions assigned.
Distributed cache	Distributed Cache Service (DCS)	To bind DCS to a component deployed in a container environment to read environment variables to obtain distributed cache information during application running, you must have the DCS ReadOnlyAccess permissions assigned.
Data storage	EVS	If the components deployed in the container environment need to use EVS disks to store data, you must have the EVS ReadOnlyAccess permissions assigned.

Console Function	Dependent Services	Policy/Role Required
	SFS	If components deployed in a container environment need to use SFS to store data, you must have the SFS Turbo ReadOnlyAccess permissions assigned.
	OBS	If components deployed in a container environment need to store data in object storage mode, you must have the OBS ReadOnlyAccess permissions assigned.
Cloud database	Relational Database Service (RDS)	To bind RDS to components deployed in a container environment for persistent storage of application data, you must have the RDS ReadOnlyAccess permissions assigned.
• Intra-VPC access of components • Domain name access of components	Elastic Load Balance (ELB)	To set intra-VPC access or domain name access for a component to use its services, you must have the ELB ReadOnlyAccess permissions assigned.
Public network access of components	NAT Gateway	To set NAT public network access for a component to use its services, you must have the NAT ReadOnlyAccess permissions assigned.
	Elastic IP (EIP)	To set EIP public network access for a component to use its services, you must have the EIP ReadOnlyAccess permissions assigned.
	Elastic Load Balance (ELB)	To set ELB public network access for a component to use its services, you must have the ELB ReadOnlyAccess permissions assigned.
Component logs	Log Tank Service (LTS)	To interconnect with LTS to view, search for, and export LTS logs for troubleshooting and resolving problems that occur during component running, you must have the LTS FullAccess permissions assigned.
Threshold rules	Simple Message Notification (SMN)	To enable SMN to send threshold alarm messages generated by components deployed in a container environment to users, you must have the SMN ReadOnlyAccess permissions assigned.

Console Function	Dependent Services	Policy/Role Required
Image repositories	SoftWare Repository for Container (SWR)	If the components deployed in the container environment come from the image package stored in SWR, you must have the SWR FullAccess permissions assigned.
Tag management	Tag Management Service (TMS)	To use TMS to set tags for managed objects such as components for management and selection, you must have the TMS ReadOnlyAccess permissions assigned.
Environment management	Virtual Private Cloud (VPC)	A VPC is used to isolate basic resources, such as computing, network, and middleware resources, used for component deployment and running in the same virtual network environment during environment creation. The VPC ReadOnlyAccess permission needs to be set.

Helpful Links

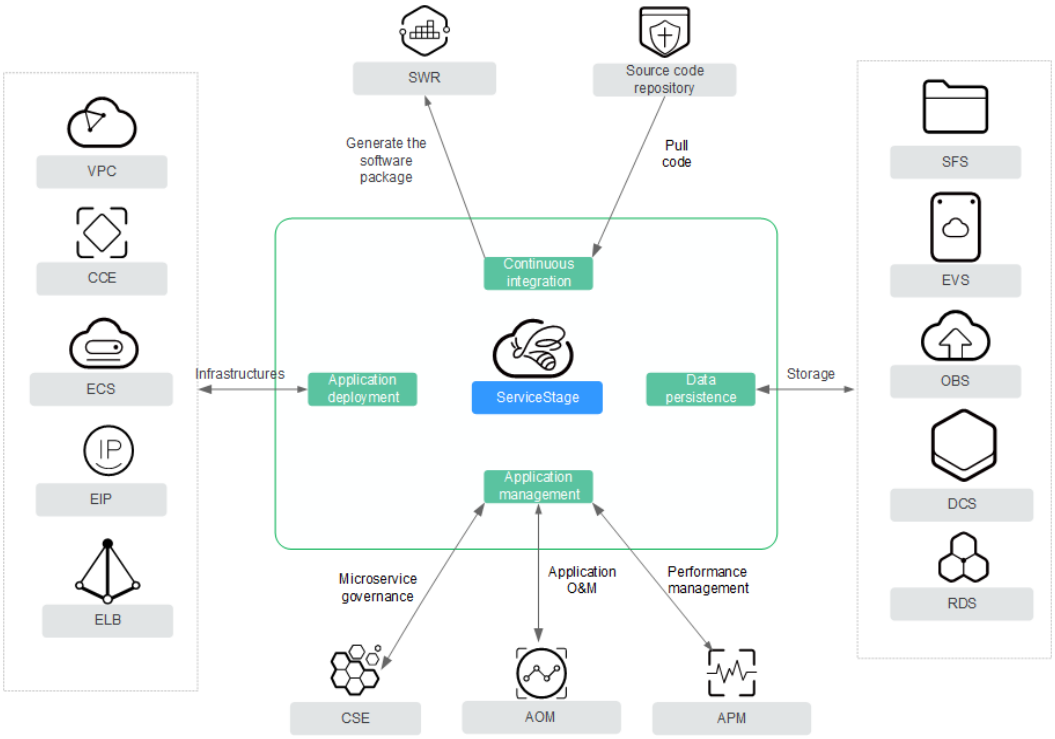
- **IAM Service Overview**
- **Creating a User and Granting Permissions**

10 Relationship with Other Cloud Services

ServiceStage is a one-stop cloud application platform integrating knowledge and experience in public cloud transformation and technology innovation. It integrates core functions of services, covering infrastructure, storage, database, software repository, monitoring and O&M, and middleware services.

ServiceStage can be used to fully experience functions of various services, as shown in **Figure 10-1**.

Figure 10-1 Cloud services integrated on ServiceStage



- ServiceStage implements interconnection with source code repositories, such as GitHub, GitLab, and Bitbucket. After being bound, you can directly pull up the source code from source code repositories for building.

- ServiceStage integrates deployment source management and archives the built software packages (or image packages) to the corresponding repositories and organizations.
- ServiceStage integrates related basic resources, such as VPC, CCE, ECS, EIP, and ELB. When deploying applications, you can directly use existing or new basic resources.
- ServiceStage integrates the Cloud Service Engine (CSE). You can perform operations related to microservice governance on ServiceStage console.
- ServiceStage integrates Application Operations Management (AOM) and Application Performance Management (APM) services. You can perform operations related to application O&M and performance monitoring.
- ServiceStage integrates storage, database, and cache services and implements persistent data storage through simple configuration.

11 Glossary

Environment

An environment is a collection of basic compute (such as CCE and ECS), network (such as ELB and EIP), and middleware (such as DCS and RDS) resources, used for component deployment and running. ServiceStage combines multiple basic resources into an environment, including development, test, pre-production, and production environments. Managing resources, and creating and deploying components by environment simplify O&M management.

Environment Variables of an Environment

Environment variables of an environment describe environment information and environment resource information, such as middleware credentials, connection information, and environment functions. Customers can use these variables to shield environment or resource differences, and switch components to different environments or run them using different cloud service resources without modification.

Priority of environment variables of the component, application, and environment levels: component > application > environment.

Basic Resources

In ServiceStage, basic resources refer to the basic services that are required or optional to microservice application hosting and O&M, such as Cloud Container Engine (CCE).

Application

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

Environment Variables of an Application

Environment variables of an application 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. Environment

variables added to an application are global environment variables and take effect for all application components.

Priority of environment variables of the component, application, and environment levels: component > application > environment.

Component

A component is a service feature implementation of an application. It is carried by code or software packages and can be independently deployed and run in an environment.

Environment Variables of a Component

Environment variables of a component are set in the component running environment and provide great flexibility for applications. Environment variables set for a component are local environment variables and take effect only for this component.

Priority of environment variables of the component, application, and environment levels: component > application > environment.

Stack

A technology stack includes the operating system, framework, and runtime system on which component running depends. It consists of attributes such as the technology stack name, type, status, and version. The version number complies with the **semantic versioning specifications**. ServiceStage provides and manages the stack lifecycle. You only need to focus on service development to improve application hosting experience.

ServiceComb

Apache ServiceComb is an open-source microservice project. It is compatible with popular ecosystems and provides a one-stop open-source microservice solution, featuring out-of-the-box readiness, high performance, and multi-language programming. It aims to help enterprises, customers, and developers easily deploy enterprise applications on the cloud in the form of microservices and implement efficient O&M.

Microservice

Microservice is defined by service. If a process provides a service, it is a microservice. Each service has its own service functions. APIs that are not restricted by languages are open to other systems (HTTP frequently used). Multiple microservices form an application.

NOTE

In ServiceStage, a microservice is relative to a component.

Microservice instance

An instance is the minimum running and deployment unit of a microservice. Generally, it corresponds to an application process.

Release Task

Release task provides functions for releasing applications: single-component release, batch operations (release, upgrade, and clone), and dependency-based orchestration.

Configuration

In ServiceStage, a configuration is a file. By creating a unified configuration file, you can fill the system variables (such as the IP address, port number, database address, and application name associated with the environment) of the environment and application in the configuration to generate a configuration file. When a component is associated with a configuration file for deployment, the system variables are automatically replaced with the actual value. This implements multi-environment use with one-time configuration through file mounting.

