



应用管理与运维平台（ServiceStage）

用户指南

发布日期 2023-03-30

目录

1 产品介绍	1
1.1 什么是应用管理与运维平台 (ServiceStage)	1
1.2 产品优势	2
1.3 典型应用场景	3
1.3.1 构建微服务应用	3
1.3.2 Web 应用生命周期管理	6
1.4 微服务引擎版本支持机制	6
1.5 名词解释	7
1.6 使用限制	8
1.7 产品规格差异	11
1.8 权限管理	13
1.9 与其他云服务之间的关系	21
2 快速入门	22
3 概述	27
4 权限管理	30
4.1 创建用户并授权使用 ServiceStage	30
4.2 创建 ServiceStage 自定义策略	31
4.3 设置 ServiceStage 依赖服务的权限	32
5 环境管理	33
5.1 环境概述	33
5.2 创建环境	33
5.3 CCE 资源管理	35
5.3.1 绑定 CCE 集群	35
5.3.2 解绑 CCE 集群	36
5.3.3 管理命名空间	36
5.3.4 管理配置项	40
5.3.5 管理密钥	43
5.4 纳管资源	47
5.5 移除已纳管资源	48
5.6 修改环境	48
5.7 删除环境	49

6 应用管理	50
6.1 创建应用	50
6.2 查看应用概览	51
6.3 管理应用环境变量	52
6.4 编辑应用	54
6.5 删除应用	55
7 组件管理	57
7.1 组件概述	57
7.2 创建并部署组件	60
7.3 查看组件详情	68
7.4 管理组件标签	68
7.5 管理组件实例	70
7.6 升级单个组件版本配置	71
7.6.1 单批发布方式升级	71
7.6.2 滚动发布方式升级	74
7.6.3 灰度发布 (金丝雀) 方式升级	76
7.7 批量升级组件版本配置	80
7.8 回滚组件版本配置	81
7.9 重新部署组件	82
7.9.1 单批发布方式重新部署	82
7.9.2 滚动发布方式重新部署	84
7.9.3 灰度发布 (金丝雀) 方式重新部署	86
7.10 设置组件访问方式	89
7.11 修改组件访问域名	89
7.12 设置组件实例伸缩策略	90
7.13 组件运维	94
7.13.1 查看组件运行指标	94
7.13.2 自定义组件运行指标	95
7.13.3 管理组件运行日志	95
7.13.4 设置资源监控告警阈值	96
7.13.5 查看组件运行事件	98
7.14 查看组件运行环境	99
7.15 启停组件实例	99
7.16 删除组件	99
7.17 组件高级设置	100
7.17.1 设置组件环境变量	100
7.17.2 设置组件生命周期	101
7.17.3 设置数据存储	102
7.17.4 设置分布式缓存	109
7.17.5 设置云数据库	110
7.17.6 设置组件实例调度策略	111
7.17.7 设置应用日志策略	113

7.17.8 设置组件自定义指标监控.....	116
7.17.9 设置健康检查.....	117
8 部署源管理.....	119
8.1 镜像仓库.....	119
8.1.1 上传镜像.....	119
8.1.2 管理镜像.....	120
8.2 组织管理.....	122
9 持续交付.....	124
9.1 概述.....	124
9.2 查看构建任务.....	125
9.3 创建源码构建任务.....	125
9.4 创建软件包构建任务.....	128
9.5 维护构建任务.....	130
9.6 流水线管理.....	133
9.7 仓库授权.....	136
10 微服务引擎.....	138
10.1 微服务引擎概述.....	138
10.2 创建微服务引擎.....	138
10.3 管理微服务引擎.....	140
10.3.1 查看微服务引擎信息.....	140
10.3.2 获取微服务引擎服务注册发现地址.....	141
10.3.3 获取微服务引擎配置中心地址.....	142
10.3.4 查看微服务引擎实例数配额.....	142
10.3.5 查看微服务引擎配置条目配额.....	143
10.3.6 设置微服务引擎备份与恢复.....	143
10.3.7 管理微服务引擎公网访问.....	145
10.3.7.1 绑定公网 IP.....	145
10.3.7.2 解绑公网 IP.....	146
10.3.8 查看微服务引擎操作日志.....	146
10.3.9 升级微服务引擎版本.....	147
10.3.10 删除微服务引擎.....	147
10.3.11 管理微服务引擎安全认证.....	148
10.4 使用微服务引擎.....	150
10.4.1 使用微服务仪表盘.....	150
10.4.2 管理微服务.....	151
10.4.3 微服务治理.....	160
10.4.3.1 概述.....	160
10.4.3.2 治理微服务.....	160
10.4.4 配置管理 (适用于微服务引擎版本为 2.x).....	176
10.4.5 配置管理 (适用于微服务引擎版本为 1.x).....	184
10.4.6 系统管理.....	187

10.4.6.1 系统管理概述.....	187
10.4.6.2 帐号管理.....	188
10.4.6.3 角色管理.....	191
11 云审计服务支持的关键操作.....	196
11.1 云审计服务支持的 ServiceStage 操作列表.....	196
11.2 查看云审计日志.....	197
12 查看监控指标及告警.....	198
13 常见问题.....	201
13.1 应用开发问题咨询指引.....	201
13.2 环境管理.....	203
13.2.1 微服务和其他平台服务有什么不同?	203
13.3 应用管理.....	203
13.3.1 如何查看应用组件部署失败的原因?	203
13.3.2 实例长期处于创建中怎么办?	203
13.3.3 如何解决 Docker 运行 node 应用程序时的依赖问题?	204
13.3.4 定制 Tomcat Context path.....	204
13.3.5 如何固定应用组件 IP?	204
13.3.6 如何体验 ServiceStage 的源码部署功能?	205
13.4 持续交付.....	206
13.4.1 ServiceStage 怎么管理 IDEA 上的代码?	206
13.4.2 添加构建服务器地址到 GitLab 服务器安全组.....	206
13.4.3 添加构建服务器地址到 Maven 服务器安全组.....	207
13.5 基础设施.....	208
13.5.1 如何处理开启了安全认证的微服务引擎专享版开启 IPv6 后服务注册失败?	208
13.5.2 如何处理操作微服务引擎专享版时遇到非微服务引擎本身错误?	208
13.6 应用运维.....	209
13.6.1 为什么 ServiceStage 中看不到日志?	209
13.6.2 替换弹性 IP 后应用访问方式失效.....	209
13.6.3 启动一个新服务后导致一个节点内存使用率过高.....	210

1 产品介绍

1.1 什么是应用管理与运维平台 (ServiceStage)

应用管理与运维平台 (ServiceStage) 是面向企业的应用管理与运维平台，提供应用发布、部署、监控与运维等一站式解决方案。支持Java、Node.js、Docker、Tomcat等技术栈。支持Apache ServiceComb Java Chassis (Java Chassis)、Spring Cloud等微服务应用，让企业应用上云更简单。

ServiceStage主要包含如下能力：

1. 应用管理：支持应用生命周期管理、环境管理。
2. 微服务应用接入：支持Java Chassis、Spring Cloud微服务框架。配合微服务引擎可实现服务注册发现、配置管理和服务治理。
3. 应用运维：通过日志、监控、告警支持应用运维管理。

图 1-1 ServiceStage 产品功能



应用管理

- 应用生命周期管理
应用完成开发后，可以托管在ServiceStage上，为您提供完整的应用生命周期管理：
 - 使用源码、软件包 (Jar) 进行应用组件创建，实现应用部署。
 - 应用从创建到下线的全流程管理，包括创建、部署、启动、升级、回滚、伸缩、停止和删除应用等功能。
- 环境管理
环境是用于应用组件部署和运行的计算、网络、中间件等基础资源的集合。ServiceStage把计算资源（如云容器引擎CCE）、网络资源（如弹性负载均衡ELB、弹性IP等）和中间件（如分布式缓存DCS、RDS、微服务引擎CSE等）组合为一个环境，如：开发环境，测试环境，预生产环境，生产环境。
环境内网络互通，可以按环境维度来管理资源、部署服务，减少具体基础资源运维管理的复杂性。

微服务应用接入

ServiceStage微服务引擎支持主流微服务框架接入和治理，您可以灵活选择最适合的微服务技术，快速开发云应用，适应复杂多变的业务需求。

- 支持原生ServiceComb微服务框架
使用ServiceComb框架开发的微服务，可以无缝接入微服务引擎。
微服务引擎采用的Apache ServiceComb Service Center，是一个RESTful风格的、高可用无状态的服务注册发现中心，提供微服务发现和微服务管理功能。服务提供者可以将自身的实例信息注册到服务注册发现中心，以供服务消费者发现并使用。关于Apache ServiceComb Service Center的详细内容请参考：
 - <https://github.com/apache/servicecomb-service-center/>
 - <https://service-center.readthedocs.io/en/latest/user-guides.html>
- 兼容主流微服务开源框架
为Spring Cloud开发的微服务提供了非常简单的接入方式，开发者只需要修改依赖关系和少量的配置，就可以接入微服务引擎，使用统一的治理能力。
- 提供微服务治理能力
使用微服务框架开发的应用托管在ServiceStage后，启动应用实例会将微服务注册到服务注册发现中心。您可以参考《用户指南》的“治理微服务”章节，针对微服务进行相关的治理。

应用运维

- 提供应用组件多维度的指标监控，帮助您把握应用上线后的运行状况。
- 提供界面化的日志查看、搜索能力，帮助您快速定位问题。

1.2 产品优势

ServiceStage集合了全面云化转型的成功经验和技术创新成果，作为一站式应用云平台，与使用传统平台相比，具有如表1-1所示的优势。

表 1-1 产品优势

应用生命周期	传统平台	使用ServiceStage平台
环境准备阶段	<ul style="list-style-type: none"> 资源获取效率低 (>1天) 资源利用率低 (<30%) 	<ul style="list-style-type: none"> 自助高效获取资源 (分钟级) 按需付费 (弹性伸缩)
业务开发阶段	<ul style="list-style-type: none"> 架构耦合, 牵一发而动全身 技术单一, 需要想办法用一种技术解决所有问题 只能按大颗粒系统发布版本, 响应周期长 	<ul style="list-style-type: none"> 架构解耦 基于契约(Open API)的开发模式, 让微服务的开发、测试、文档、协作和管控活动标准化、自动化。 各种技术灵活接入 支持Java、PHP、Python、Node.js开发语言。 高性能REST/RPC微服务开发框架, 提供开箱即用的工具, 降低开发门槛。 敏捷高效 一站式微服务治理控制台, 提供微服务负载均衡、限流、降级、熔断、容错、错误注入等治理能力。 支持微服务级升级、灰度发布。
安装部署阶段	<ul style="list-style-type: none"> 烟囱式系统 全手工部署 	开发者只需使用ServiceStage+源码仓库, 实现一键自动部署和更新。
应用升级	<ul style="list-style-type: none"> 打补丁方式 手动升级 中断业务 	支持滚动升级: 升级过程中, 业务会同时均衡分布到新老实例上, 因此业务不会中断。
应用运维	<ul style="list-style-type: none"> 应用崩溃, 闪退 服务端响应慢 系统资源不足 故障难定位 	<ul style="list-style-type: none"> 实时图形化展示应用监控指标 CPU占用、告警、节点异常、运行日志、关键事件实时掌握。 微服务治理 支持微服务接口级SLA指标 (吞吐量、时延、成功率) 实时 (秒级) 监控和治理, 保障应用运行不断服。

1.3 典型应用场景

1.3.1 构建微服务应用

典型业务应用场景

应用场景

对于传统的单体架构项目, 不同的业务模式必须采取统一的技术方案及技术平台, 每个业务模块也不能独立出来复用, 系统中一个模块出现问题会导致整个系统不可用。

随着企业业务的复杂度不断提升，传统单体架构模式越来越臃肿，难以适应灵活多变的业务需求，微服务应用可以很好的解决上述问题。

价值

通过应用微服务化，企业可将一个臃肿的系统拆分成若干小的服务组件，组件之间的通讯采用轻量的协议完成，实现各组件生命周期管理的解耦。

随着业务增长，服务会遇到各种意外情况，如：瞬时大规模并发访问、服务出错、入侵等情况。使用微服务架构可以对服务做细粒度管控，支撑业务需求。

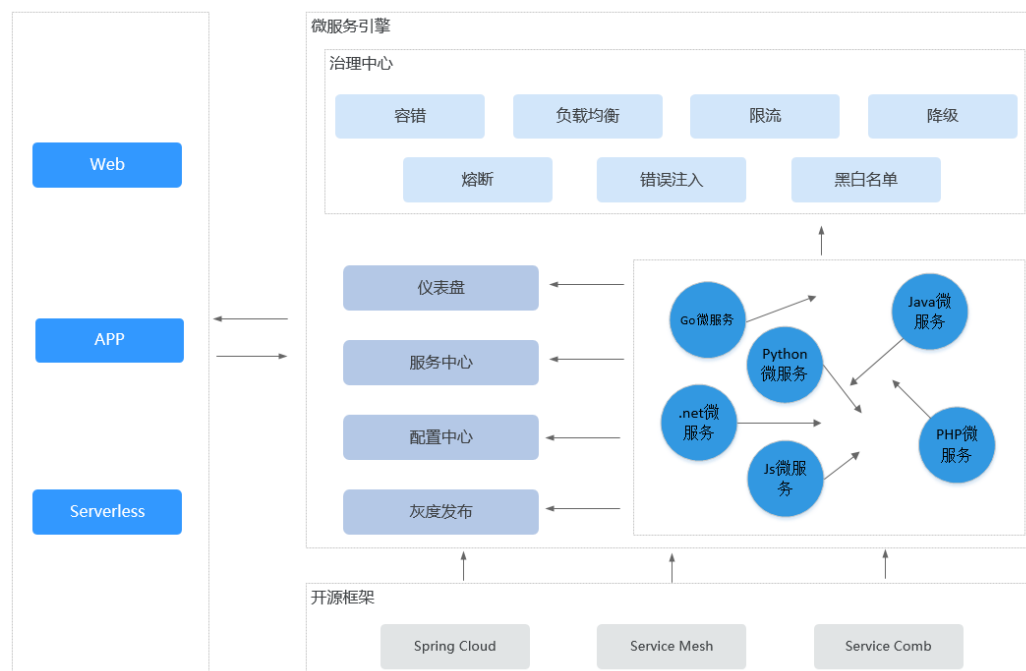
ServiceStage可承载微服务应用的全生命周期管理。支持Java、Node.js、Docker、Tomcat等技术栈，可无侵入托管Apache ServiceComb Java Chassis、Spring Cloud等微服务应用，另外还提供配置管理、监控运维和服务治理等更多功能，让企业微服务应用上云更简单。

优势

ServiceStage提供了微服务应用解决方案，具有以下优势：

- 支持原生ServiceComb、Spring Cloud等多种微服务框架，支持双栈模式（SDK和服务网格互通），无需更改业务代码直接托管上云。
- API First，支持基于Swagger的API管理。
- 支持多语言微服务，如JAVA、Node.js等。
- 提供服务中心、配置中心、仪表盘、灰度发布等功能。
- 提供负载均衡、容错、限流、降级、熔断、错误注入、黑白名单等全套微服务治理策略。可针对业务场景进行界面化操作，极大提高了服务治理的可用性。

图 1-2 微服务应用解决方案



- 实现Spring Cloud、Java Chassis之间的互相发现。

持续集成和持续交付

应用场景

对于复杂的业务系统，从项目创建、编译、构建、自验、集成验证、类生产验证、上线的各个阶段都需要耗费大量的人力和时间，并且容易受到人为因素影响而出错。持续集成和持续交付由于具有标准化和自动化特点，可以很好的解决该问题。

价值

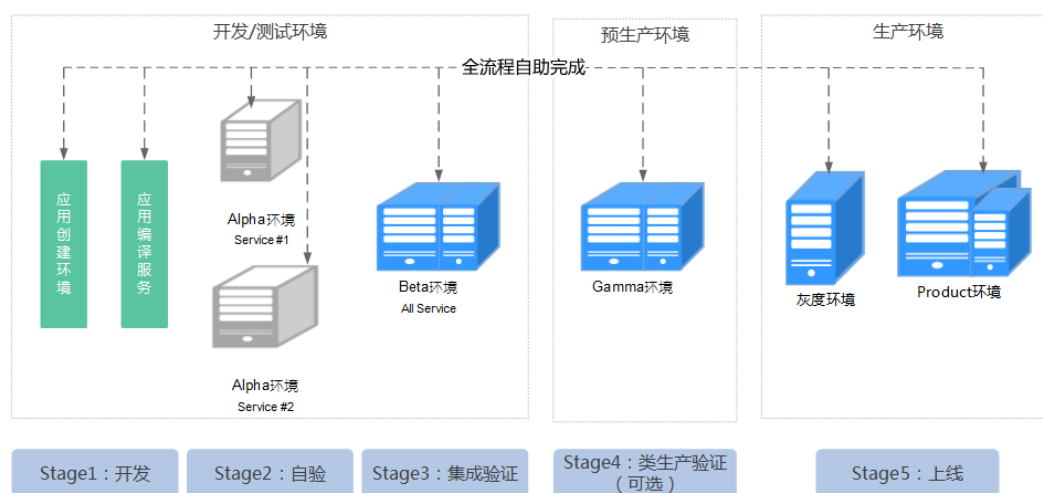
由人工执行变成了自动化执行，减少错误出现，提高工作效率。

环境及流程标准统一，利于业务扩展，降低升级改造成本。

优势

基于ServiceStage流水线，实现了集成环境统一、交付流程标准化，您可以实现全流程“自助式”开发、自验、集成验证与上线。

图 1-3 持续集成和持续交付



灰度发布

应用场景

为保障新特性平稳上线，可以通过灰度发布功能选择少部分用户试用，待新特性成熟以后，再切换版本让所有用户使用。

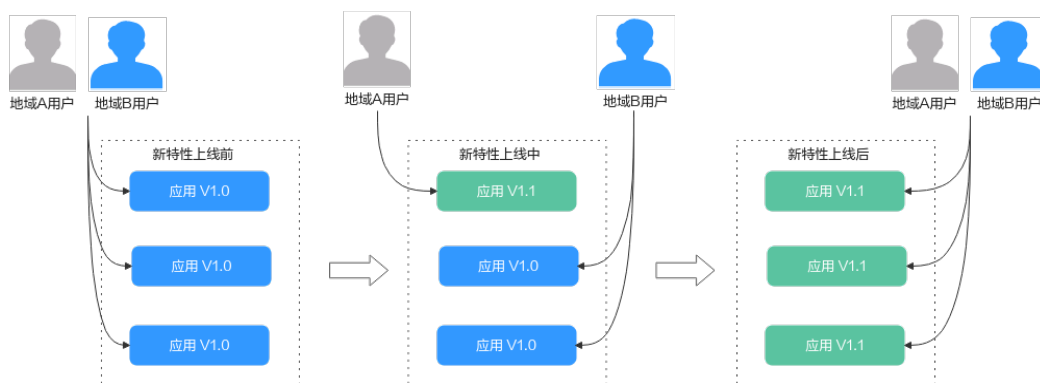
价值

灰度发布可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以减少其影响度。

优势

ServiceStage提供了微服务级灰度发布的能力。

图 1-4 灰度发布



1.3.2 Web 应用生命周期管理

典型业务应用场景

应用场景

Web程序的应用范围非常广泛，日常使用的企业业务系统、网上商城系统、论坛、博客、Wiki知识系统、网络游戏等都可能是Web应用。针对不同技术架构的Web应用进行生命周期的管理，是企业IT部门主要工作内容之一。

价值

使用统一的平台管理各种Web应用，能够大大简化工作量，提高效率，快速响应复杂多变的业务需求。

优势

ServiceStage一站式运维平台，提升了企业级Web应用开发和运维的效率，使企业专注业务创新。具有以下优势：

- 一键部署，支持War、Jar、Zip软件包一键部署。
- 一站式运维，提供升级、回滚、日志、监控、弹性等丰富的运维能力。
- 无缝集成，支持与ELB、RDS、DCS等云服务与应用无缝集成。

1.4 微服务引擎版本支持机制

本章节为您介绍微服务引擎专享版的版本支持机制。

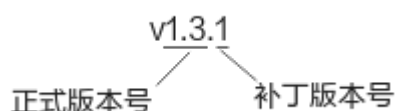
版本号说明

版本号格式为：{major}.{minor}.{patch}。

其中：

- {major}.{minor}为正式版本号。
- {patch}为补丁版本号。

例如，v1.3.1。1.3为正式版本号，1为补丁版本号。



版本支持机制

- 微服务引擎创建
只能创建最新版本的微服务引擎，不支持创建指定版本微服务引擎。
- 微服务引擎维护
支持同时维护最新的3个正式版本。其余的版本，将不再提供技术支持，包括支持新的功能、社区bugfix回合、漏洞修复、升级等。
- 微服务引擎版本升级
 - 正式版本升级：
 - 1.3、1.2版本支持平滑升级到2.4及以后版本，且功能兼容。
 - 支持当前最新的3个正式版本中的2个较低版本升级到最新版本。例如，当前最新的3个正式版本为2.4、1.3、1.2，则支持由1.2、1.3升级到2.4。

📖 说明

当引擎升级已超出可支持升级的版本范围，例如由1.0升级到1.3，可能导致微服务引擎的管理功能不可用，请谨慎操作。

您可以通过联系技术支持工程师，进行升级前风险评估。

- 补丁版本升级，微服务引擎后台提供补丁版本自动升级，例如由1.3.0升级到1.3.1。

版本约束

微服务引擎版本升级后，不支持版本回滚。

1.5 名词解释

环境

环境是用于应用组件部署和运行的计算、网络、中间件等基础资源的集合。

ServiceStage把计算资源（如云容器引擎CCE）、网络资源（如弹性负载均衡ELB、弹性IP等）和中间件（如分布式缓存DCS、云数据库RDS、微服务引擎CSE等）组合为一个环境，如：开发环境，测试环境，预生产环境，生产环境。

环境内网络互通，可以按环境维度来管理资源、部署服务，减少具体基础资源运维管理的复杂性。

基础资源

在ServiceStage中，基础资源指微服务应用托管与运维所依赖或可选依赖的基础类服务，如云容器引擎等。

应用

应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。

组件

组件是组成应用的某个业务特性实现，以代码或者软件包为载体，可独立部署运行。

技术栈

技术栈包含组件运行所依赖的操作系统、框架、运行时，由技术栈名称、类型、状态、版本等属性构成。其中，版本号遵从[语义化版本控制规范](#)。

ServiceStage为您提供并管理技术栈生命周期，您只需聚焦具体业务开发，提升您的应用托管体验。

ServiceComb

ServiceComb是一个Apache微服务开源项目，提供开箱即用、高性能、兼容流行生态、支持多语言的一站式开源微服务解决方案。致力于帮助企业、用户和开发者将企业应用轻松微服务化上云，并实现对微服务应用的高效运维管理。

微服务

微服务是业务概念，某个进程提供某种服务，那它就是个微服务。每一个服务都具有自主运行的业务功能，对外开放不受语言限制的API (最常用的是HTTP)。多个微服务组成应用程序。

说明

在ServiceStage中，微服务对应组件。

微服务实例

一个微服务的最小运行和部署单元，通常对应一个应用进程。

1.6 使用限制

ServiceStage具有如下限制，下边的每条限制都适用于任何一个Region的每个租户。

限制不等于资源配额限制，而是表示ServiceStage无法向租户提供超过限制的能力，最终用户的技术选型和方案设计时需要注意这些限制。

注册发现

微服务引擎专业版 (Cloud Service Engine) 限制请参见[表1-2](#)。

表 1-2 微服务引擎专业版限制

项目	限制
心跳上报	每个微服务实例最频繁每30s一次
服务发现	每个微服务实例最频繁每30s一次

项目	限制
微服务实例注册	每秒10个

微服务引擎专享版限制请参见表1-3。

表 1-3 微服务引擎专享版限制 (最高规格)

项目	限制	备注
心跳上报	每个微服务实例最频繁每20s一次	总限流为2000tps
服务发现	每个微服务实例最频繁每20s一次	-
微服务实例注册	每秒1000个	-

微服务引擎专享版引擎类型说明

1.3.0或以上版本的微服务引擎专享版，不同规格均提供了对如下表所示的引擎类型的支持，您可以根据具体业务需求选择使用对应类型的微服务引擎专享版。

表 1-4 微服务引擎专享版引擎类型说明

引擎类型	适用业务场景	可用区CPU架构
集群	集群模式部署，主机级容灾，高可用。	x86或ARM，不支持混合架构。
单机	单节点部署，不支持容灾，非高可用。 须知 “单机”类型的微服务引擎专享版仅适用于开发测试环境，不适用于生产环境。	x86或ARM。

微服务框架版本要求

微服务开发框架推荐版本如下表所示。

- 如果已经使用低版本的微服务开发框架构建应用，建议升级到推荐版本，以获取最稳定和丰富的功能体验。
- 如果已使用Spring Cloud微服务开发框架开发了应用，推荐使用[Spring Cloud Huawei](#)接入应用。
- 如果基于开源开放和业界生态组件新开发微服务应用，可选择Spring Cloud框架。
- 如果希望使用微服务引擎提供的开箱即用的治理能力和高性能的RPC框架，可选择Java Chassis框架。

框架	推荐版本	说明
Spring Cloud Huawei	1.10.6-2021.0.x及以上	<p>采用Spring Cloud Huawei项目提供接入支持：</p> <ul style="list-style-type: none"> • 适配的Spring Cloud版本为2021.0.3 • 适配的Spring Boot版本为2.6.7 <p>Spring Cloud微服务开发框架的版本说明请参见：https://github.com/HuaweiCloud/spring-cloud-huawei/releases。</p>
Java Chassis	2.7.10及以上	<p>可以直接使用开源项目提供的软件包接入，不需要引用其他第三方软件包。</p> <p>Java Chassis微服务开发框架的版本说明请参见：https://github.com/apache/servicecomb-java-chassis/releases。</p>

微服务引擎专享版配额说明

配额是指您在微服务引擎专享版实例中可创建的资源数量限制，具体的资源配额限制如表1-5所示。

表 1-5 资源配额限制

功能	资源	最大配额	是否支持修改配额	注意事项
微服务管理	微服务版本数量 (个)	10,000	暂不支持	-
	单个实例数据量 (KB)	200	支持	扩大配额后，将增加微服务发现的时延。
	单个微服务契约数量 (个)	500	暂不支持	-
配置管理	单个配置数据量 (KB)	128	暂不支持	-
	单个应用级配置数量 (个)	2,000	暂不支持	-
微服务治理	应用级的治理策略	1,000	暂不支持	所有的应用的治理策略总和不能超过1000条。

📖 说明

- 单个治理策略包含：治理规则和业务场景。治理规则和业务场景实际会等量占用配置中心的配额。
- 微服务版本数：微服务场景中版本用来标记微服务的迭代记录，方便对微服务的不同迭代进行管理。
- 微服务实例数：实例是一个微服务的最小运行和部署单元，通常对应一个应用进程。同一个微服务通过部署在多个容器或虚拟机，可以实现多个实例同时运行。
- 配置条目数：微服务场景中的配置是指对程序代码中某些变量的取值控制。比如，动态配置就是通过微服务运行过程中对某些变量的取值进行动态变更。

1.7 产品规格差异

表 1-6 功能说明

功能		规格说明
管理规模	单个Region下最大支持应用组件实例个数	5000个
	单个组件支持的最大实例个数	200
微服务	微服务引擎	可创建的微服务引擎数量同引擎类型及微服务引擎所在的可用区下的虚拟机资源数量有关系。 每个可用区下最多有50台虚拟机资源，其中： 创建一个“集群”类型的微服务引擎，需消耗5台虚拟机资源。
	JAVA语言微服务开发SDK	支持
	Spring Cloud微服务接入	
	服务注册中心	
	服务配置中心	
	实时仪表盘	
	负载均衡	
	服务限流	
	服务降级	
	服务容错	
	服务熔断	
	错误注入	
	黑白名单	

功能		规格说明
应用生命周期管理	多语言应用管理 (Java/PHP/Python/Node.js/Tomcat/Docker)	支持
	应用生命周期管理 (伸缩、升级、回退、启动、停止、重启、删除)	
	应用基础监控 (运行状态、CPU、内存、磁盘使用率等)	
	容器部署, 当前支持Java、PHP、Python、Node.js、Tomcat、Docker	
	访问权限控制	
	应用域名管理	
	弹性伸缩	
	事件分析	
	日志分析	
	阈值告警	
部署源管理	Docker镜像包管理	支持
	仓库权限管理	
应用基础运维	日志存储与搜索	1TB/月, 存储时长30天
	日志分析	500GB
	查询日志数据API调用	10GB/月
	主机监控	200VM, 指标存储时长1年
	秒级监控指标数量	2000个
	自定义指标	500个
	查询指标数据API调用	500万次/月
	事件和告警总条数	50万条/月, 存储时长30天
	智能阈值规则	50条

微服务引擎实例规格说明

微服务引擎分为微服务引擎专业版和微服务引擎专享版两种产品规格:

- 微服务引擎专业版：专业版引擎Cloud Service Engine是ServiceStage提供的免费体验引擎。专业版引擎可以体验ServiceStage的所有产品能力，比如服务治理、配置管理等。引擎资源为所有租户共享，性能可能会受其他租户影响；专业版引擎不支持升级到专享版。
- 微服务引擎专享版：专享版引擎，是可支持大规模微服务应用管理的商用引擎。您可根据业务需要选择不同规格，不支持规格变更；专享版引擎资源独享，性能不受其他租户影响。

引擎支持的最大实例规格说明如下。

表 1-7 微服务引擎实例规格说明

引擎类型	微服务实例数配额	配置条目数配额
微服务引擎专业版	1,000	-
微服务引擎专享版	100	600
	500	3,000
	2,000	12,000

1.8 权限管理

如果您需要对ServiceStage的资源，给企业中的员工设置不同的访问权限，以达到不同员工之间的权限隔离，您可以使用统一身份认证服务 (Identity and Access Management, 简称IAM) 进行精细的权限管理。该服务提供用户身份认证、权限分配、访问控制等功能，可以帮助您安全的控制云资源的访问。

通过IAM，您可以在云帐号中给员工创建IAM用户，并使用策略来控制员工对云资源的访问范围。例如您的员工中有负责软件开发的人员，您希望员工拥有ServiceStage的使用权限，但是不希望员工拥有删除等高危操作的权限，那么您可以使用IAM为开发人员创建用户，通过授予仅能使用ServiceStage，但是不允许删除的权限策略，控制员工对ServiceStage资源的使用范围。

如果云帐号已经能满足您的要求，不需要创建独立的IAM用户进行权限管理，您可以跳过本章节，不影响您使用ServiceStage服务的其它功能。

IAM是云服务提供权限管理的基础服务，无需付费即可使用，您只需要为您帐号中的资源进行付费。关于IAM的详细介绍，请参见[《产品介绍》](#)。

ServiceStage 权限

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略，才能使得用户组中的用户获得策略定义的权限，这一过程称为授权。授权后，用户就可以基于策略对云服务进行操作。

ServiceStage资源通过物理区域划分，为项目级服务。授权时，“作用范围”需要选择“区域级项目”，然后在指定区域对应的项目中设置相关权限，并且该权限仅对此项目生效；如果在“所有项目”中设置权限，则该权限在所有区域项目中都生效。访问ServiceStage时，需要先切换至授权区域。

根据授权精细程度分为角色和策略：

- 角色：IAM最初提供的一种根据用户的工作职能定义权限的粗粒度授权机制。该机制以服务为粒度，提供有限的服务相关角色用于授权。由于各云服务之间存在业务依赖关系，因此给用户授予角色时，可能需要一并授予依赖的其他角色，才能正确完成业务。角色并不能满足用户对精细化授权的要求，无法完全达到企业对权限最小化的安全管控要求。
- 策略：IAM最新提供的一种细粒度授权的能力，可以精确到具体服务的操作、资源以及请求条件等。基于策略的授权是一种更加灵活的授权方式，能够满足企业对权限最小化的安全管控要求。

如表1-8所示，包括了ServiceStage的所有系统权限。推荐使用系统策略，系统角色仅用于兼容已有的权限配置。

表 1-8 ServiceStage 系统权限说明

系统角色/策略名称	描述	类别	依赖系统权限
ServiceStage FullAccess	应用管理与运维平台所有权限。	系统策略	无
ServiceStage ReadOnlyAccess	应用管理与运维平台只读权限。	系统策略	无
ServiceStage Developer	应用管理与运维平台开发者权限。 拥有应用、组件、环境的操作权限，但无基础设施创建权限。	系统策略	无
CSE Admin	微服务引擎服务管理员权限。	系统策略	无
CSE Viewer	微服务引擎服务查看权限。	系统策略	无
ServiceStage Administrator	应用管理与运维平台管理员，拥有该服务下的所有权限。	系统角色	CCE Administrator、VPC Administrator、SWR Administrator、Server Administrator、OBS Administrator。

系统角色/策略名称	描述	类别	依赖系统权限
ServiceStage Operator	应用管理与运维平台操作员，拥有该服务下的只读权限。	系统角色	CCE Administrator、VPC Administrator、SWR Administrator、Server Administrator和OBS Administrator。
ServiceStage Developer	应用管理与运维平台开发者，拥有该服务下的所有权限。	系统角色	CCE Administrator、VPC Administrator、SWR Administrator、Server Administrator和OBS Administrator。

如果表1-8所列的这些权限不满足实际需求，您可以参考表1-9和表1-10，在这个基础上自定义策略。

表 1-9 ServiceStage 常用操作与系统权限之间的关系

操作	ServiceStage ReadOnlyAccess	ServiceStage Developer	ServiceStage FullAccess
创建应用	x	√	√
修改应用	x	√	√
查询应用	√	√	√
删除应用	x	√	√
创建组件	x	√	√
查询组件	√	√	√
部署组件	x	√	√
维护组件	x	√	√
删除组件	x	√	√
创建构建工程	x	√	√

操作	ServiceStage ReadOnlyAccess	ServiceStage Developer	ServiceStage FullAccess
修改构建工程	x	√	√
查询构建工程	√	√	√
启动构建工程	x	√	√
删除构建工程	x	√	√
创建流水线	x	√	√
修改流水线	x	√	√
查询流水线	√	√	√
启动流水线	x	√	√
克隆流水线	x	√	√
删除流水线	x	√	√
新建仓库授权	x	√	√
修改仓库授权	x	√	√
查询仓库授权	√	√	√
删除仓库授权	x	√	√

表 1-10 CSE 常用操作与系统权限之间的关系

操作	CSE Viewer	CSE Admin
创建微服务引擎	x	√
维护微服务引擎	x	√
查询微服务引擎	√	√
删除微服务引擎	x	√
创建微服务	x	√
查询微服务	√	√
维护微服务	x	√

操作	CSE Viewer	CSE Admin
删除微服务	x	√
创建微服务配置	x	√
查询微服务配置	√	√
编辑微服务配置	x	√
删除微服务配置	x	√
创建微服务治理策略	x	√
查询微服务治理策略	√	√
编辑微服务治理策略	x	√
删除微服务治理策略	x	√

细粒度权限

说明

- SWR未支持细粒度权限，相关权限需要另外授权。

使用自定义细粒度策略，请使用管理员用户登录IAM控制台，按需选择ServiceStage、CSE的细粒度权限进行授权操作。

- CSE细粒度权限依赖说明请参见[表1-11](#)。
- ServiceStage细粒度权限依赖说明请参见[表1-12](#)。

表 1-11 CSE 细粒度权限依赖说明

权限名称	权限描述	权限依赖	应用场景
cse:engine:list	列出所有引擎	无	查看引擎列表。
cse:engine:get	查看引擎信息	cse:engine:list	查看引擎详情，仅微服务引擎专享版支持。
cse:engine:modify	修改引擎	<ul style="list-style-type: none"> cse:engine:list cse:engine:get 	修改引擎的操作包括：开启/关闭公网访问操作、开启/关闭安全认证操作、引擎失败任务重试操作，仅微服务引擎专享版支持。
cse:engine:upgrade	升级引擎	<ul style="list-style-type: none"> cse:engine:list cse:engine:get 	升级引擎的操作包括：引擎版本升级操作，仅微服务引擎专享版支持。

权限名称	权限描述	权限依赖	应用场景
cse:engine:delete	删除引擎	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● vpc:ports:get ● vpc:ports:delete 	删除引擎，仅微服务引擎专享版支持。
cse:engine:create	创建引擎	<ul style="list-style-type: none"> ● cse:engine:get ● cse:engine:list ● ecs:cloudServerFlavors:get ● vpc:vpcs:get ● vpc:vpcs:list ● vpc:subnets:get ● vpc:ports:get ● vpc:ports:create 	创建引擎的操作包括：创建引擎操作，引擎备份/恢复任务创建操作，仅微服务引擎专享版支持。
cse:config:modify	服务配置管理修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:config:get 	全局配置功能与治理功能涉及的配置修改。
cse:config:get	服务配置管理查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get 	查看服务的配置。
cse:governance:modify	服务治理中心修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:config:get ● cse:config:modify ● cse:registry:get ● cse:registry:modify ● cse:governance:get 	创建与修改服务治理。
cse:governance:get	服务治理中心查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:config:get ● cse:registry:get 	查看服务治理功能。
cse:registry:modify	服务注册管理修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get 	服务修改。

权限名称	权限描述	权限依赖	应用场景
cse:dashboard:modify	dashboard管理修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get ● cse:dashboard:get ● cse:registry:modify 	仪表盘修改。
cse:dashboard:get	dashboard管理查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get 	仪表盘查看。
cse:registry:get	服务注册管理查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get 	服务目录查看。

📖 说明

仪表盘不需要授权，但是依赖registry权限，因为仪表盘区分服务需要使用服务目录功能。

表 1-12 ServiceStage 细粒度权限依赖说明

权限名称	权限描述	权限依赖	应用场景
servicestage:app:get	查询应用信息	servicestage:app:list	查询应用信息。
servicestage:app:create	创建应用	<ul style="list-style-type: none"> ● servicestage:app:get ● servicestage:app:list ● servicestage:ass embling:get ● servicestage:ass embling:list ● servicestage:ass embling:create 	创建应用。

权限名称	权限描述	权限依赖	应用场景
servicestage:app:modify	更新应用	<ul style="list-style-type: none"> servicestage:app:get servicestage:app:list servicestage:assembling:get servicestage:assembling:list servicestage:assembling:modify 	更新应用。
servicestage:app:delete	删除应用	<ul style="list-style-type: none"> servicestage:app:get servicestage:app:list servicestage:assembling:delete 	删除应用。
servicestage:app:list	查看环境和应用列表	无	查看环境和应用列表。
servicestage:environment:create	创建环境	<ul style="list-style-type: none"> servicestage:app:get servicestage:app:list 	创建环境。
servicestage:environment:modify	更新环境	<ul style="list-style-type: none"> servicestage:app:get servicestage:app:list 	更新环境。
servicestage:environment:delete	删除环境	<ul style="list-style-type: none"> servicestage:app:get servicestage:app:list 	删除环境。
servicestage:assembling:get	查看构建信息	servicestage:assembling:list	查看构建信息。
servicestage:assembling:create	创建构建	<ul style="list-style-type: none"> servicestage:assembling:get servicestage:assembling:list 	创建构建。
servicestage:assembling:modify	修改构建	<ul style="list-style-type: none"> servicestage:assembling:get servicestage:assembling:list 	修改构建。

权限名称	权限描述	权限依赖	应用场景
servicestage:assembling:delete	删除构建	<ul style="list-style-type: none">servicestage:assembling:getservicestage:assembling:list	删除构建。
servicestage:assembling:list	查看构建列表	无	查看构建列表。

相关链接

- [《产品介绍》](#)
- [4.1 创建用户并授权使用ServiceStage](#)

1.9 与其他云服务之间的关系

ServiceStage吸收了全面云化转型的成功经验和技术创新成果，集成了众多服务的核心功能，对云基础设施服务、存储服务、数据库服务、软件仓库服务、监控运维以及中间件服务进行了深度整合，全力打造功能齐全的一站式应用云平台。

使用ServiceStage可以系统的体验到众多云服务的功能精髓。

- ServiceStage实现了与源码仓库的对接（如Gitee、GitHub、GitLab、Bitbucket），绑定源码仓库后，可以直接从源码仓库拉取源码进行构建。
- ServiceStage集成了部署源管理功能，可以将构建完成的软件包（或者镜像包）归档对应的仓库和组织。
- ServiceStage集成了相关的基础资源（如VPC、CCE、EIP、ELB），在部署应用时可以直接使用已有或者新建所需的基础资源。
- ServiceStage集成了微服务引擎，进入ServiceStage控制台可以进行微服务治理相关的操作。
- ServiceStage集成了应用运维管理服务，可以进行应用运维相关的操作。
- ServiceStage集成了存储、数据库、缓存等服务，通过简单配置即可实现数据持久化存储。

2 快速入门

概述

通过快速入门可以学习到ServiceStage中三个核心概念：环境、应用、组件。

- 环境是把计算资源（如云容器引擎CCE）、网络资源（如弹性负载均衡ELB、弹性IP等）和中间件（如分布式缓存DCS、RDS、微服务引擎CSE等）组合而成，部署应用和组件时需要选择环境进行部署，选择后会自动加载此环境所包含的资源。
- 应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。比如：典型的ERP系统就是一个应用，它一般由会计、财务、生产控制、物流、采购、分销、库存等模块组成，这些模块之间相互有紧密的联系，每个模块就是一个组件。
- 组件是组成应用的某个业务特性实现，如上面的ERP系统举例。在微服务应用场景下，每个组件都有独立的软件包，可独立部署运行，部署的组件称为一个组件实例，一个组件也可以有多个组件实例形成一个集群来保障应用和组件的高可靠性。针对应用的组件实例进行启停、部署（升级组件版本配置）、回退/回滚、伸缩、查看日志、查看事件、设置访问方式、设置阈值告警等运维操作。

本文示例讲解如何搭建环境、创建应用、创建并部署组件、确认部署结果、访问应用、应用运维，供您体验ServiceStage的基本功能。

前提条件

1. 创建一个虚拟私有网络VPC，请参考[创建虚拟私有云和子网](#)，使用默认参数配置进行创建。
2. 创建一个CCE集群，请参考[创建CCE集群](#)。
集群中至少包含1个规格为8vCPUs、16GB内存或者2个规格为4vCPUs、8GB内存的ECS节点，并且绑定弹性IP。
3. 创建用于存储软件包的桶，请参考[创建桶](#)。
4. 创建不开启安全认证的微服务引擎，请参考[10.2 创建微服务引擎](#)。
5. 本示例提供了一个微服务demo，会在本地进行编译构建打包，需要本地机器环境安装了Java JDK和Maven，并且能够访问Maven中央库。

安装后本地打开cmd命令，执行`mvn -v`命令能查到相关的版本信息，说明安装成功。如下所示，本示例使用Maven 3.6.3和JDK 1.8.0版本：

```
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
.....
Java version: 1.8.0_201,.....
```

准备软件包

步骤1 下载[微服务demo](#)源码到本地并解压。

步骤2 在项目根目录下（例如：D:\servicecomb-samples-master\servicecomb-samples-master\ServiceComb-SpringMVC），打开cmd命令，执行**mvn clean package**命令，对Java项目进行编译打包。

```
D:\servicecomb-samples-master\servicecomb-samples-master\ServiceComb-SpringMVC>mvn clean package
.....
[INFO] --- maven-jar-plugin:2.6:jar (default-jar) @ servicecomb ---
[INFO] Building jar: D:\servicecomb-samples-master\servicecomb-samples-master\ServiceComb-SpringMVC\target\servicecomb-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:1.5.9.RELEASE:repackage (default) @ servicecomb ---
[INFO] -----
[INFO] BUILD SUCCESS
.....
```

编译成功后，会在项目根目录下的target子目录（例如：D:\servicecomb-samples-master\servicecomb-samples-master\ServiceComb-SpringMVC\target）下生成servicecomb-0.0.1-SNAPSHOT.jar软件包，并显示“BUILD SUCCESS”。

步骤3 将生成的servicecomb-0.0.1-SNAPSHOT.jar软件包上传到已创建的OBS桶中备用。

上传软件包，请参考《对象存储服务用户指南》的“[上传文件](#)”章节。

----结束

创建组织

步骤1 登录ServiceStage控制台。

步骤2 选择“部署源管理 > 组织管理”。

步骤3 单击“创建组织”，在弹出的页面中填写“组织名称”。

步骤4 单击“确定”。

----结束

创建环境

步骤1 登录ServiceStage控制台。

步骤2 选择“环境管理 > 创建环境”，参照下表设置环境信息。

参数	参数说明
环境名称	输入环境名称（例如：test-env）。
*企业项目	设置企业项目。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 创建企业项目 后可以使用。

参数	参数说明
虚拟私有云(VPC)	选择 前提条件 中已准备好的虚拟私有云VPC。 说明 环境创建完成后，不支持修改VPC。
环境类型	选择Kubernetes。

步骤3 单击“立即创建”，进入环境详情页面。

步骤4 在“资源配置”下左侧列表，选择“计算”资源类型下的“云容器引擎 CCE”，单击“立即绑定”。

步骤5 在弹出的对话框中，选择[前提条件](#)中已创建的CCE集群资源，单击“确定”。

步骤6 在“资源配置”下左侧列表，选择“中间件”资源类型下的“微服务引擎 CSE”，单击“纳管资源”。

步骤7 在弹出的对话框中，勾选[前提条件](#)中已创建的CSE资源，单击“确定”。

----结束

新建应用

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理 > 创建应用”。

步骤3 输入“应用名称”，例如：test-servicestage。选择企业项目。

步骤4 单击“确定”，完成应用创建。

----结束

创建并部署组件

步骤1 登录ServiceStage控制台。

步骤2 单击“应用管理”，进入应用列表。

步骤3 选择[新建应用](#)时创建的应用名称（例如：test-servicestage），单击“操作”列“新增组件”。

步骤4 在“基本信息”区域，参考下表设置必填组件基本信息，其余参数保持默认。

参数	说明
组件名称	输入组件名称（例如：test-cse）。
所属环境	选择 创建环境 时创建的环境（例如test-env）。

步骤5 在“组件包”区域，参考下表设置必填组件包参数，其余参数保持默认。

参数	说明
技术栈	组件技术栈类型选择Java。

参数	说明
上传方式	单击“选择软件包”，选择 准备软件包 时上传的软件包 servicecomb-0.0.1-SNAPSHOT.jar。

步骤6 在“构建”区域，参考下表设置必填构建参数，其余参数保持默认。

参数	说明
组织	组织用于管理组件构建生成的镜像。 选择 创建组织 时创建的组织名称。
构建环境	选择“使用当前环境构建”，使用组件所属的部署环境中的CCE集群进行镜像构建。 当前环境CCE集群的master节点和node节点的CPU架构必须保持一致，否则会导致组件构建失败。

步骤7 单击“下一步”。

步骤8 “资源”区域，参数设置保持默认。

步骤9 “访问方式”区域，参数设置保持默认。

步骤10 “时区”区域，参数设置保持默认。

步骤11 在“高级设置”区域，展开“高级设置 > 微服务引擎”。

1. 单击“绑定微服务引擎”。
2. 选择当前环境下已纳管的微服务引擎。
3. 单击“确定”。

步骤12 单击“创建并部署”，部署组件。

----结束

确认部署结果

步骤1 登录ServiceStage控制台。

步骤2 选择“微服务引擎 > 微服务目录”。

步骤3 在微服务引擎下拉列表选择部署了组件的微服务引擎。

步骤4 在“全部应用”下拉列表中选择springmvc应用。

如果存在已部署的servicecombspringmvc微服务，且微服务实例数为2，则部署成功。



The screenshot shows the '实例列表' (Instance List) tab in the ServiceStage console. It displays a table with the following data:

微服务名称	所属环境	所属应用	版本号	实例数	创建时间	操作
servicecombspringmvc	<空>	springmvc	1	2	2022/08/12 11:10:39 GMT+08:00	删除

----结束

访问应用

- 步骤1** 登录ServiceStage控制台。
- 步骤2** 单击“应用管理”，进入应用列表。
- 步骤3** 单击**新建应用**时创建的应用名称（例如：test-servicestage），进入“应用概览”页。
- 步骤4** 在“组件列表”区域，单击**创建并部署组件**时创建的组件名称（例如：test-cse），进入组件“概览”页。
- 步骤5** 单击“访问方式”。
- 步骤6** 单击“TCP/UDP路由配置”区域的“添加服务”，参考下表设置参数。

参数	参数说明
服务名称	使用默认。
访问方式	选择“公网访问”。
访问类型	选择“弹性IP”。
服务亲和	保持默认。
端口映射	1. 协议：选择“TCP”。 2. 容器端口：输入8080。 3. 访问端口：选择“自动生成”。

- 步骤7** 单击“确定”，生成访问地址。
- 步骤8** 单击“访问地址”列下的访问地址，访问应用。

返回如下结果：

```
{"message":"Not Found"}
```

- 步骤9** 在浏览器地址栏输入<http://步骤7生成的访问地址/rest/helloworld?name=ServiceStage>，再次访问应用。

可以返回如下结果：

```
"ServiceStage"
```

----结束

应用运维

- 步骤1** 登录ServiceStage控制台。
 - 步骤2** 单击“应用管理”。
 - 步骤3** 单击**新建应用**时创建的应用名称（例如：test-servicestage），进入应用“概览”页。
 - 步骤4** 在“组件列表”区域，单击**创建并部署组件**时创建的组件名称（例如：test-cse），进入组件“概览”页，可以进行运维操作。
- 结束

3 概述

应用管理与运维平台 (ServiceStage) 是面向企业的应用管理与运维平台，提供应用发布、部署、监控与运维等一站式解决方案。支持Java、Node.js、Docker、Tomcat等技术栈。支持Apache ServiceComb Java Chassis (Java Chassis)、Spring Cloud等微服务应用，让企业应用上云更简单。


本文档指导您快速了解如何通过ServiceStage创建、部署应用组件以及如何应用运维、服务治理。

使用条件

1. 已注册云帐号。
2. 当前登录帐号拥有使用ServiceStage服务的权限。帐号权限授权与绑定，请参考[4.1 创建用户并授权使用ServiceStage](#)。

登录 ServiceStage 控制台

步骤1 登录云控制台。

步骤2 单击左上角 ，在服务列表单击“应用管理与运维平台 ServiceStage”。

- 如果是首次登录，在弹出的服务授权页面单击“同意授权”，授权ServiceStage服务使用依赖的服务，然后进入“应用管理与运维平台”控制台。
- 如果非首次登录，直接进入“应用管理与运维平台”控制台。

----结束

控制台说明

ServiceStage控制台说明如[表3-1](#)所示。

表 3-1 ServiceStage 控制台说明

类别	说明
总览	“总览”页面提供ServiceStage总体看板，包含文档帮助、应用、环境、组件、告警等信息。

类别	说明
环境管理	环境是用于组件部署和运行的计算、网络、中间件等基础资源的集合。 “环境管理”页面提供环境创建、编辑、删除、资源配置（纳管资源、移除资源）等功能，以列表形式展示已创建的环境。
应用管理	应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。 “应用管理”页面提供应用创建、编辑、删除等功能，以列表形式展示已创建的应用及其下已创建的组件数量，并提供在应用下创建组件的操作入口。
组件管理	组件是组成应用的某个业务特性实现，以代码或者软件包为载体，可独立部署运行。 “组件管理”页面列表展示所有应用下的组件，并提供组件详情页、组件创建及运维入口。
部署源管理	提供组织管理、镜像仓库等功能。 <ul style="list-style-type: none"> 组织管理用于隔离镜像，并为租户下用户指定不同的权限（读取、编辑、管理）。 镜像仓库提供用于存储、管理Docker容器镜像的场所，可以让您轻松存储、管理Docker容器镜像。
持续交付	提供构建工程的查看、发布以及仓库授权等功能。 <ul style="list-style-type: none"> 构建 通过构建任务可以一键式生成软件包或者镜像包，实现“源码拉取->编译->打包->归档”的全流程自动化。 发布 通过流水线工程可以完成一键部署，实现“源码拉取->编译->打包->归档->部署”的全流程自动化。利于集成环境统一、交付流程标准化。 仓库授权 用户可以新建仓库授权，使构建工程、应用组件可以使用授权信息访问软件仓库。
微服务引擎	提供了引擎实例管理、仪表盘使用、微服务目录管理、微服务治理、配置管理、系统管理功能的操作入口。

图 3-1 ServiceStage 控制台



4 权限管理

4.1 创建用户并授权使用 ServiceStage

如果您需要对您所拥有的ServiceStage进行精细的权限管理，您可以使用[统一身份认证服务](#)（Identity and Access Management，简称IAM）。通过IAM，您可以：

- 根据企业的业务组织，在您的云服务帐号中，给企业中不同职能部门的员工创建IAM用户，让员工拥有唯一安全凭证，并使用ServiceStage资源。
- 根据企业用户的职能，设置不同的访问权限，以达到用户之间的权限隔离。
- 将ServiceStage资源委托给更专业、高效的其他帐号或者云服务，这些帐号或者云服务可以根据权限进行代运维。

如果帐号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用ServiceStage服务的其它功能。

本章节为您介绍对用户授权的方法，操作流程如[图4-1](#)所示。

前提条件

给用户组授权之前，请您了解用户组可以添加的ServiceStage权限，并结合实际需求进行选择。ServiceStage支持的系统权限，请参见[1.8 权限管理](#)。

示例流程

图 4-1 给用户授权 ServiceStage 权限流程



1. 在IAM控制台创建用户组，并授予ServiceStage “ServiceStage ReadOnlyAccess” 权限。
2. 在IAM控制台创建用户，并将其加入1中创建的用户组。
3. 并验证权限
新创建的用户登录控制台，验证ServiceStage的只读权限。
 - 在“服务列表”中选择“应用管理与运维平台 ServiceStage”，进入“应用管理”页面，单击“创建应用”，若提示权限不足，表示“ServiceStage ReadOnlyAccess”已生效。
 - 在“服务列表”中选择除ServiceStage外的任一服务，若提示权限不足，表示“ServiceStage ReadOnlyAccess”已生效。

4.2 创建 ServiceStage 自定义策略

如果系统预置的ServiceStage权限，不满足您的授权要求，可以创建自定义策略。

目前支持以下两种方式创建自定义策略：

- 可视化视图创建自定义策略：无需了解策略语法，按可视化视图导航栏选择云服务、操作、资源、条件等策略内容，可自动生成策略。
- JSON视图创建自定义策略：可以在选择策略模板后，根据具体需求编辑策略内容；也可以直接在编辑框内编写JSON格式的策略内容。

具体创建步骤请参见：[创建自定义策略](#)。本章为您介绍常用的ServiceStage自定义策略样例。

自定义策略样例

如下以定制一个IAM用户禁止创建及删除微服务引擎的策略为例。

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

拒绝策略需要同时配合其他策略使用，否则没有实际作用。用户被授予的策略中，一个授权项的作用如果同时存在Allow和Deny，则遵循Deny优先原则。

权限授予成功后，用户可以通过控制台以及REST API等多种方式验证。

此处以上述自定义策略为例，介绍用户如何通过登录ServiceStage控制台验证自定义禁止创建微服务引擎的权限：

1. 使用新创建的用户登录云服务，登录方法选择为“IAM用户”。
 - 租户名为该IAM用户所属云服务帐号的名称。
 - IAM用户名和IAM用户密码为以租户名在IAM创建用户时输入的用户名和密码。
2. 在“微服务引擎”页面，创建微服务引擎，返回403错误，表示权限配置正确并已生效。

4.3 设置 ServiceStage 依赖服务的权限

设置 CCE 命名空间权限

由于IAM只能给ServiceStage用户组授予CCE集群相关资源的普通操作权限，不包括集群（启用Kubernetes RBAC鉴权）的命名空间权限，需要在CCE集群单独设置CCE的命名空间权限。

CCE的命名空间权限设置方法，请参考[权限管理](#)。

设置 CTS 服务权限

通过IAM给ServiceStage设置权限后，对于ServiceStage依赖的CTS服务并不会生效，需要单独设置CTS服务权限。

5 环境管理

5.1 环境概述

环境是用于组件部署和运行的计算、网络、中间件等基础资源的集合。ServiceStage把计算资源（如云容器引擎CCE）、网络资源（如弹性负载均衡ELB、弹性IP等）和中间件（如分布式缓存DCS、云数据库RDS、微服务引擎CSE等）组合为一个环境，如：开发环境、测试环境、预生产环境和生产环境。

环境内网络互通，可以按环境维度来管理资源、部署服务，减少具体基础资源运维管理的复杂性。

同一个项目下，最多支持创建300个环境。

5.2 创建环境

在部署组件之前，您需要先创建环境。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择“环境管理 > 创建环境”，填写环境配置信息，其中带“*”标志的参数为必填参数。

参数	参数说明
*环境名称	环境的名称。
*企业项目	设置企业项目。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 创建企业项目 后可以使用。



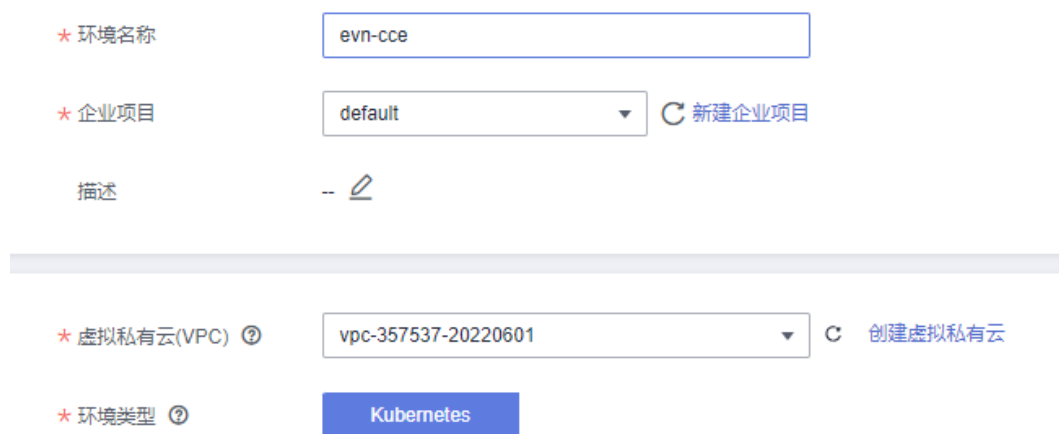
参数	参数说明
描述	环境说明信息。 1. 单击  , 输入环境描述信息。 2. 单击  , 保存描述。
*虚拟私有云 (VPC)	选择环境资源所在VPC。 新建VPC, 请参考 创建虚拟私有云 。 说明 环境创建完成后, 执行 5.6 修改环境 操作时, 不支持修改VPC。
*环境类型	选择环境类型。 Kubernetes: 适用于容器部署方式 (云容器引擎部署), 组件以容器镜像方式通过Kubernetes调度部署。 说明 组件部署方式说明, 请参考 组件部署方式说明 。

图 5-1 设置环境参数



The screenshot shows a configuration form for creating an environment. It includes the following fields and options:

- * 环境名称**: Input field containing "evn-cce".
- * 企业项目**: Dropdown menu showing "default", with a "新建企业项目" (New Enterprise Project) button.
- 描述**: Input field with a "--" placeholder and an edit icon.
- * 虚拟私有云(VPC)**: Dropdown menu showing "vpc-357537-20220601", with a "创建虚拟私有云" (Create Virtual Private Cloud) button.
- * 环境类型**: A blue button labeled "Kubernetes".

步骤3 单击“立即创建”，完成环境创建。

环境创建完成后, 进入环境详情页面, 可以查看环境详情、进行环境资源配置等操作。

说明

如果之前版本的环境纳管了CCE集群和虚拟机资源, 升级到当前版本后, 该环境的类型为“虚拟机+Kubernetes”。

----结束

后续操作

- Kubernetes类型的环境创建完成后, 在使用该环境部署组件前, 需要先为环境绑定CCE集群, 请参考[5.3 CCE资源管理](#)。

- 环境创建完成后，需要把计算资源（不含CCE资源）、网络资源和中间件资源纳管到环境下组合为一个环境，请参考[5.4 纳管资源](#)。

5.3 CCE 资源管理

5.3.1 绑定 CCE 集群

Kubernetes类型的环境创建完成后，在使用该环境部署组件前，需要先为环境绑定1个且只能绑定1个CCE集群。

前提条件

已创建环境待绑定的CCE集群，集群和环境所在的VPC需要一致，且不能被其他环境纳管。

创建CCE集群，请参考[创建CCE集群](#)。

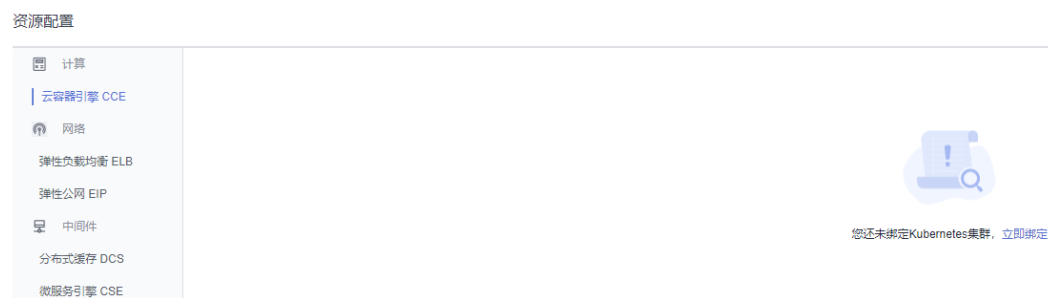
操作步骤

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎 CCE”。

图 5-2 绑定 CCE 集群



步骤4 单击“立即绑定”。

- 已创建CCE集群，在集群下拉列表选择待绑定的CCE集群，单击“确定”。
- 未创建CCE集群，请根据页面提示前往CCE控制台，参考[前提条件](#)创建CCE集群后再次执行绑定操作。

说明

“Kubernetes”类型的环境，如果选择的VPC开启了IPv6且纳管了CCE集群资源时，需选择开启了IPv6开关的CCE集群资源。否则，会导致在该VPC下的开启了安全认证的微服务引擎专享版上注册的Java Chassis微服务在使用IPv6服务注册发现地址时，注册失败。

----结束

后续操作

- 选择“节点列表”页签，您可以查看已绑定的CCE集群各节点的详细信息。
- 单击“查看资源详情”，您可以在CCE控制台查看CCE集群详情。

5.3.2 解绑 CCE 集群

已绑定给Kubernetes类型环境的CCE集群，如果不再使用，可以从环境移除。

前提条件

环境已绑定CCE集群，请参考[5.3.1 绑定CCE集群](#)。

操作步骤

- 步骤1** 登录ServiceStage控制台。
- 步骤2** 在“环境管理”页面，单击待操作环境名称。
- 步骤3** 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎CCE”。
- 步骤4** 单击“解绑”，从环境中移除CCE集群。

图 5-3 解绑 CCE 集群



----结束

5.3.3 管理命名空间

命名空间 (Namespace) 是对一组资源和对象的抽象整合。在同一个CCE集群内可创建不同的命名空间，不同命名空间中的数据彼此隔离。使得它们既可以共享同一个集群的服务，也能够互不干扰。

例如可以将开发环境、测试环境的业务分别放在不同的命名空间。

命名空间的类别说明，如[表5-1](#)所示。

表 5-1 命名空间类别

创建类型	说明
集群默认创建	集群在启动时会默认创建default、kube-public、kube-system、kube-node-lease命名空间。 <ul style="list-style-type: none">• default: 所有未指定Namespace的对象都会被分配在default命名空间。• kube-public: 此命名空间下的资源可以被所有人访问（包括未认证用户），用来部署公共插件、容器模板等。• kube-system: 所有由Kubernetes系统创建的资源都处于这个命名空间。• kube-node-lease: 每个节点在该命名空间中都有一个关联的“Lease”对象，该对象由节点定期更新。
用户创建	用户可以按照需要创建命名空间，例如开发环境、联调环境和测试环境分别创建对应的命名空间。或者按照不同的业务创建对应的命名空间，例如系统若分为登录和游戏服务，可以分别创建对应命名空间。

本章节指导您创建、删除命名空间、管理命名空间资源配额。

前提条件

环境已绑定CCE集群，请参考[5.3.1 绑定CCE集群](#)。

创建命名空间

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎CCE”。

步骤4 选择“命名空间”页签，单击“创建命名空间”。

步骤5 参考下表填写参数，其中带“*”标志的参数为必填参数。

参数	说明
*命名空间	输入命名空间的名称。
命名空间描述	输入命名空间的描述信息。

图 5-4 设置命名空间参数

创建命名空间

⚠ 您最多创建1000个命名空间，您还可以创建996个

* 命名空间

命名空间名称以小写字母开头，由小写字母，数字，中划线(-)组成，63个字符之内，且不能以中划线(-)结尾。

命名空间描述

确定 取消

步骤6 单击“确定”。

创建成功的命名空间会显示在命名空间列表。

----结束

删除命名空间

须知

- 删除命名空间会删除该命名空间下所有的资源（如工作负载、配置项等），请谨慎操作。
- 集群默认创建的命名空间，不支持删除。

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎CCE”。

步骤4 选择“命名空间”页签，删除命名空间。

- 删除单个命名空间：在待删除的用户创建的命名空间所在行“操作”列，单击“删除”。
- 批量删除命名空间：勾选待删除的用户创建的命名空间，单击命名空间列表左上端的“删除”。

步骤5 在弹出的对话框输入“DELETE”后，单击“确定”。

----结束

管理命名空间资源配额

默认情况下，CCE集群运行中的Pod可以无限制的使用Node节点上的CPU和内存，这意味着任意一个Pod都可以无节制地使用集群的计算资源，某个命名空间的Pod可能会耗尽集群的所有资源。

kubernetes在一个物理集群上提供了多个虚拟集群，这些虚拟集群被称为命名空间。命名空间可用于多种工作用途，满足多用户的使用需求，通过为每个命名空间配置资源额度可以有效限制资源滥用，从而保证集群的可靠性。您可为命名空间配置包括CPU、内存、Pod数量等资源的额度，更多信息请参见[Resource Quotas](#)。

用户创建命名空间、集群默认创建的default命名空间支持资源配额管理。

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎CCE”。

步骤4 选择“命名空间”页签，进入命名空间列表。

步骤5 在待操作的命名空间所在行“操作”列，单击“配额管理”。

在弹出的“配额管理”对话框，您可以查看该命名空间下的资源类型及资源配额总量、配额累计使用量。

图 5-5 进入配额管理页面

命名空间	状态	命名空间描述	创建时间	操作
default	可用	--	2022-11-14 19:04:24 GMT+08:00	配额管理 删除
kube-node-lease	可用	--	2022-11-14 19:03:53 GMT+08:00	配额管理 删除
kube-public	可用	--	2022-11-14 19:03:53 GMT+08:00	配额管理 删除
kube-system	可用	--	2022-11-14 19:03:53 GMT+08:00	配额管理 删除
namespace-dev	可用	--	2022-12-14 10:01:59 GMT+08:00	配额管理 删除
namespace-test	可用	--	2022-12-14 09:55:37 GMT+08:00	配额管理 删除

步骤6 单击“编辑配额”，根据您的实际业务需求，设置各资源类型的总配额。

- 如果不限该资源类型的使用量，请输入“0”。
- 如果限制该资源类型的使用量，请输入期望的整型数值。

须知

- 配额累计使用量包含CCE系统默认创建的资源，如default命名空间下系统默认创建的kubernetes服务（该服务可通过后端kubectl工具查看）等，故建议命名空间下的资源配额略大于实际期望值以去除系统默认创建资源的影响。
- 如果限制了命名空间下CPU或内存的总配额，则在[7.2 创建并部署组件](#)、[7.6 升级单个组件版本配置](#)中为部署环境为Kubernetes类型且命名空间为该命名空间的组件设置资源时，必须设置组件运行可以使用的最大、最小CPU核数（Core）和内存数量（GiB）。否则，会导致操作失败。
- 如果限制了命名空间下其他资源类型的总配额，当该资源类型剩余使用量不满足需求时，会导致部署环境为Kubernetes类型且命名空间为该命名空间的组件部署失败。

步骤7 单击“确定”。

----结束

5.3.4 管理配置项

配置项 (ConfigMap) 是一种用于存储应用所需配置信息的资源类型，内容由用户决定。配置项创建完成后，可在应用中作为文件或者环境变量使用。

配置项允许您将配置文件从镜像中解耦，从而增强应用的可移植性。

配置项价值如下：

- 使用配置项功能可以帮您管理不同环境、不同业务的配置。
- 方便您部署相同应用的不同环境，配置文件支持多版本，方便您进行更新和回滚应用。
- 方便您快速将配置以文件的形式导入到应用中。

本章节指导您创建、删除、查看和更新配置项。

前提条件

1. 环境已绑定CCE集群，请参考[5.3.1 绑定CCE集群](#)。
2. 已创建配置项所在命名空间，请参考[创建命名空间](#)。

创建配置项

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎CCE”。

步骤4 选择“配置项”页签，单击“创建配置项”。

ServiceStage支持“可视化”和“YAML”两种创建方式来创建配置项。

- 方式一：可视化。
参照[表5-2](#)设置配置项参数，其中带“*”标志的参数为必填参数。

表 5-2 可视化方式创建配置项参数设置说明

参数	说明
*配置名称	新建的配置项名称，同一个命名空间里命名必须唯一。
*所属集群	使用新建配置项的集群。
*命名空间	新建配置项所在的命名空间，默认为default。
描述	配置项的描述信息。

参数	说明
配置数据	应用配置的数据可以在应用中使用，或被用来存储配置数据。其中，“键”代表文件名；“值”代表文件中的内容。 1. 单击“添加更多配置数据”。 2. 输入键、值。
配置标签	标签以键值对的形式附加到各种对象上（如应用、节点、服务等）。 标签定义了这些对象的可识别属性，用来对它们进行管理和选择。 1. 单击“添加标签”。 2. 输入键、值。

图 5-6 可视化方式设置配置项参数

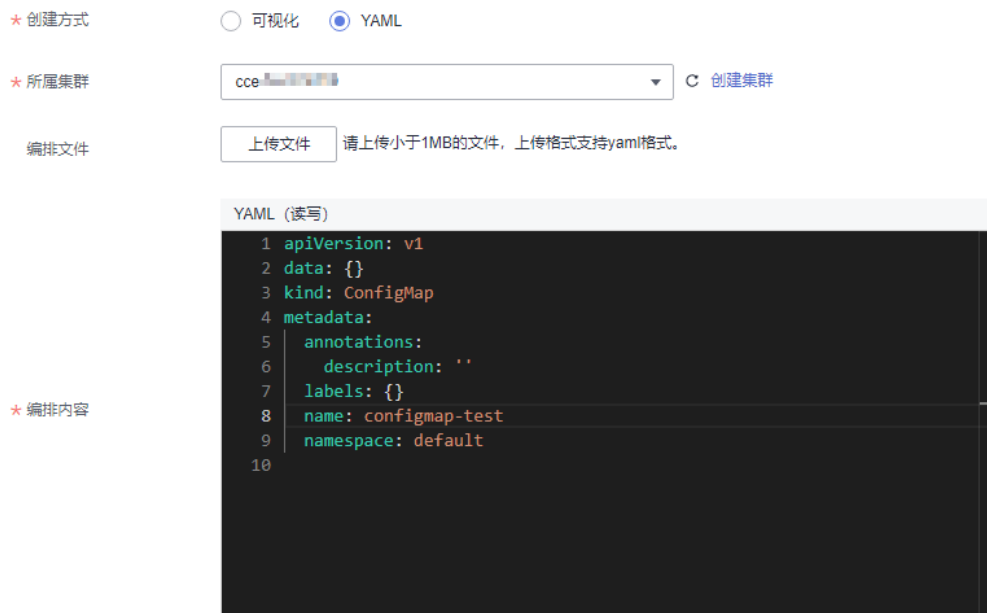
- 方式二：YAML。

📖 说明

若需要通过上传文件的方式创建配置项，请确保配置项描述文件已创建。ServiceStage支持YAML格式，详情请参考[ConfigMap配置项要求](#)。

- 在“所属集群”下拉框中，选择相应的集群。
- （可选）单击“上传文件”，选择本地已创建的ConfigMap类型资源文件后，单击“打开”，等待文件上传成功。
- 在“编排内容”中写作者或者修改ConfigMap资源文件。

图 5-7 YAML 方式设置配置项参数



步骤5 单击“创建配置项”。

配置项创建完成后，应用配置列表中会出现新创建的应用配置。

----结束

后续操作

配置项创建完成后，您可参考[表5-3](#)查找、查看、更新和删除配置项。

📖 说明

- 删除操作无法恢复，请谨慎操作。
- 配置项列表中包含系统配置项资源，系统配置项资源不可更新，也不能删除，只能查看。

表 5-3 配置项管理操作说明

操作项	操作说明
查找配置项	1. 在命名空间下拉列表选择配置项所在命名空间。 2. 在搜索框输入配置项名称。
查看配置项	单击目标配置项“操作”列的“查看YAML”，查看配置项YAML格式文件内容。
更新配置项	1. 单击目标配置项“操作”列的“更新”。 2. 根据 表5-2 更改信息。 3. 单击“更新配置项”。
删除单个配置项	1. 单击目标配置项“操作”列的“删除”。 2. 在弹出的对话框，单击“确定”。

操作项	操作说明
批量删除配置项	1. 勾选需要删除的配置项。 2. 单击“删除配置项”。 3. 在弹出的对话框，单击“确定”。

ConfigMap 配置项要求

ConfigMap资源文件支持yaml文件格式，且文件大小不得超过1MB。

文件配置示例如下：

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

5.3.5 管理密钥

密钥 (Secret) 是一种用于存储应用认证信息、应用密钥等敏感信息的资源，内容由用户决定。密钥创建完成后，可在应用中作为文件或者环境变量使用。

本章节指导您创建、删除、查看和更新密钥。

前提条件

1. 环境已绑定CCE集群，请参考[5.3.1 绑定CCE集群](#)。
2. 已创建密钥所在命名空间，请参考[创建命名空间](#)。

创建密钥

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”资源类型下的资源名称“云容器引擎CCE”。

步骤4 选择“密钥”页签，单击“创建密钥”。

ServiceStage支持“可视化”和“YAML”两种方式来创建密钥。

- 方式一：可视化。参照[表5-4](#)设置参数，其中带“*”标志的参数为必填参数。

表 5-4 可视化方式创建密钥参数设置说明

参数	参数说明
*密钥名称	新建密钥的名称，同一个命名空间内命名必须唯一。
*所属集群	使用新建密钥的集群。 单击“创建集群”，可以新建集群。

参数	参数说明
*命名空间	新建密钥所在的命名空间，默认为default。
描述	密钥的描述信息。
*密钥类型	根据业务需要选择新建的密钥类型。 <ul style="list-style-type: none">- Opaque：一般密钥类型。当密钥配置文件中未作显式设定时，默认的密钥类型是Opaque。- kubernetes.io/dockerconfigjson：存放拉取私有仓库镜像所需的认证信息。- IngressTLS：存放7层负载均衡服务所需的证书。- 其他：若需要创建其他类型的密钥，请手动输入密钥类型。
*镜像仓库地址	当“密钥类型”选择kubernetes.io/dockerconfigjson时有效。输入镜像仓库的地址。
*密钥数据	应用密钥的文件data字段值。 <ul style="list-style-type: none">- 当密钥为Opaque类型时，输入键、值。其中“值”必须使用Base64进行编码，Base64编码方法请参考Base64编码。单击“添加更多配置数据”，可以增加密钥数据。- 当密钥为kubernetes.io/dockerconfigjson类型时，输入“用户名”和“密码”。- 当密钥为IngressTLS类型时，上传“证书文件”和“私钥文件”。- 当密钥为其他类型时，输入对应的“键”、“值”。
密钥标签	标签以Key/value键值对的形式附加到各种对象上（如应用、节点、服务等）。 标签定义了这些对象的可识别属性，用来对它们进行管理和选择。 <ol style="list-style-type: none">1. 单击“添加标签”。2. 输入键、值。

图 5-8 可视化方式设置密钥参数

* 创建方式 可视化 YAML

* 密钥名称 [如何创建密钥?](#)

* 所属集群 [创建集群](#)

* 命名空间 [创建命名空间](#)

描述

* 密钥类型
一般密钥类型

键	值	操作
<input type="text" value="cse_credentials_accessKey"/>	<input type="text" value="....."/>	删除
<input type="text" value="cse_credentials_secretKey"/>	<input type="text" value="....."/>	删除

[添加更多配置数据](#)

键	值	操作
<input type="text" value="lable-secret"/>	<input type="text" value="secret-test"/>	删除

- 方式二：YAML。

📖 说明

若需要通过上传文件的方式创建密钥，请确保密钥描述文件已创建。ServiceStage支持YAML格式，详情请参考[Secret资源文件配置说明](#)。

- 在“所属集群”下拉框中，选择相应的集群。
- (可选)单击“上传文件”，选择已创建的Secret类型资源文件后，单击“打开”，等待文件上传完成。
- 在“编排内容”中写作或者修改上传的Secret资源文件。

图 5-9 YAML 方式设置密钥参数

* 创建方式 可视化 YAML

* 所属集群 [创建集群](#)

编排文件 请上传小于1MB的文件，上传格式支持yaml格式。

* 编排内容

```
YAML (读写)
1  apiVersion: v1
2  data:
3    '': ''
4  kind: Secret
5  metadata:
6    annotations:
7      description: ''
8    labels:
9      lable-secret: secret-test
10   name: secret-g3pqu4
11   namespace: default
12  type: Opaque
13
```

步骤5 单击“创建密钥”。

密钥列表中会出现新创建的密钥。

----结束

后续操作

密钥创建完成后，您可参考[表5-5](#)查找、查看、更新和删除密钥。

📖 说明

- 删除操作无法恢复，请谨慎操作。
- 密钥列表中包含系统密钥资源，系统密钥资源不可更新，也不能删除，只能查看。

表 5-5 密钥管理操作说明

操作项	操作说明
查找密钥	1. 在命名空间下拉列表选择密钥所在命名空间。 2. 在搜索框输入密钥名称。
查看密钥	单击目标密钥“操作”列的“查看YAML”，查看配置项YAML格式文件内容。
更新密钥	1. 单击目标密钥“操作”列的“更新”。 2. 根据 表5-4 更改信息。 3. 单击“更新密钥”。
删除单个密钥	1. 单击目标密钥“操作”列的“删除”。 2. 在弹出的对话框，单击“确定”。
批量删除密钥	1. 勾选需要删除的密钥。 2. 单击“删除密钥”。 3. 在弹出的对话框，单击“确定”。

Secret 资源文件配置说明

本内容主要介绍Secret类型的资源描述文件的配置示例。例如现在有一个应用需要获取帐号密码，可以通过Secret来实现：

```
username: my-username
```

```
password: my-password
```

定义的Secret文件内容如下。其中值需要用Base64进行编码，Base64编码方法请参考[Base64编码](#)。

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecret      # secret的名称
  namespace: default #命名空间，默认为default
data:
  username: ***** #需要用Base64编码
  password: ***** #需要用Base64编码
type: Opaque # type建议不要做修改
```

Base64 编码

对字符串进行Base64加密，可以在本地Linux环境下，直接使用`echo -n '编码内容' | base64`命令即可。示例如下：

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

其中：

- 3306为要编码的内容。
- MzMwNg==为编码后的内容。

5.4 纳管资源

环境创建成功后，需要把计算资源类型（如云容器引擎CCE）、网络资源类型（如弹性负载均衡ELB、弹性IP等）和中间件资源类型（如分布式缓存DCS、云数据库RDS、微服务引擎CSE等）纳管到环境下，组合为一个环境。

如需管理Kubernetes类型的环境下的CCE集群资源，请参考[5.3 CCE资源管理](#)。

前提条件

- 已创建环境待纳管的弹性负载均衡ELB，ELB和环境所在的VPC需要一致。
请参考[创建独享型负载均衡器](#)。
- 已创建环境待纳管的弹性IP。
请参考[申请弹性公网IP](#)。
- 已创建环境待纳管的分布式缓存DCS，DCS和环境所在的VPC需要一致。
请参考[创建实例](#)。
- 已创建环境待纳管的云数据库RDS，RDS和环境所在的VPC需要一致。
请参考[创建实例](#)。
- 已创建环境待纳管的微服务引擎CSE。如果CSE与环境所在VPC不一致，需正确配置VPC连通。
请参考[10.2 创建微服务引擎](#)。

操作步骤

- 步骤1** 登录ServiceStage控制台。
- 步骤2** 在“环境管理”页面，单击待操作环境名称。
- 步骤3** 在“资源配置”下左侧列表，选择“计算”、“网络”或“中间件”资源类型下的资源名称，单击“纳管资源”。
- 步骤4** 在弹出的对话框中，勾选需要纳管的资源，单击“确定”。

说明

- 同一VPC下，已被其他环境纳管的CCE资源，不支持被再次纳管。

----结束

5.5 移除已纳管资源

已纳管到环境下的资源，如果不再使用，可以将其移除。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，单击待操作环境名称。

步骤3 在“资源配置”下左侧列表，选择“计算”、“网络”或“中间件”资源类型下的资源名称。

步骤4 在右侧已纳管的资源列表中：

- 批量移除资源：勾选待移除的资源，单击“移除资源”。
- 移除单个资源：在待移除的资源“操作”列，单击“移除”。

----结束

5.6 修改环境

本章节指导您对已创建的环境进行修改。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，以如下任意方式进入“编辑环境”页面：

- 选择待操作环境，在“操作”列单击“编辑”。
- 单击待操作环境名称，进入环境详情页面，单击“编辑”。

步骤3 参考下表编辑环境信息。



参数	参数说明
环境名称	环境的名称。
*企业项目	企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 创建企业项目 后可以使用。
描述	环境说明信息。 1. 单击  ，输入环境描述信息。 2. 单击  ，保存描述。

图 5-10 编辑环境信息

The screenshot shows a form for editing environment information. It contains the following fields and elements:

- * 环境名称**: A text input field containing "env-ecs".
- * 企业项目**: A dropdown menu showing "default" and a "新建企业项目" (New Enterprise Project) button.
- 描述**: A text area with a "--" icon and an edit icon.
- * 虚拟私有云(VPC) ?**: A dropdown menu that is currently empty.
- * 环境类型 ?**: A button labeled "Kubernetes".

步骤4 单击“保存”，完成环境修改。

----结束

5.7 删除环境

当环境不再使用时，可以将其删除。

📖 说明

- 删除环境前，需确认待删除环境下未部署组件或者部署的组件已经被删除。具体操作请参考 [7.16 删除组件](#)。
- 删除环境，不会删除环境中纳管的资源。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 在“环境管理”页面，选择以下任意方式删除环境。

- 选择待操作环境，在“操作”列单击“删除”。
- 单击待操作环境名称，进入环境详情页面，单击“删除”。

步骤3 单击“确定”，环境删除成功。

----结束

6 应用管理

6.1 创建应用

应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。

例如把天气预报作为一个应用，包含weather和forecast两个组件。以应用维度组织多个组件，可以实现应用在不同环境的快速克隆。

ServiceStage可支持同一个项目下的单个用户最多创建1000个应用。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理 > 创建应用”，参考下表设置应用参数，其中带“*”标志的参数为必填参数。

参数	说明
*应用名称	输入应用名称。 应用名称必须唯一，不能重复。
企业项目	选择企业项目。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 创建企业项目 后可以使用。
描述	输入应用描述信息。

图 6-1 创建应用

创建应用

* 应用名称

企业项目 [新建企业项目](#)

描述

0/128

步骤3 单击“确定”，创建应用。

----结束

6.2 查看应用概览

应用创建完成后，您可以进入“应用概览”页面查看应用概览。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理”。

步骤3 单击待操作的应用名称，进入“应用概览”页面，查看应用详情。

如果应用下已经创建并部署了组件，可以在组件列表查看组件详情。

图 6-2 查看应用概览

组件名称	环境	状态	技术栈	实例数	外部访问地址	更新时间	操作
fusionweather	env-ccc	异常	OpenJDK8	0 / 2 (正常全部)	设置	2022-12-13 11:48:02 GMT+0...	详情 事件 更多
forecast	env-ccc	运行中	OpenJDK8	2 / 2 (正常全部)	设置	2022-12-07 16:41:11 GMT+0...	详情 事件 更多
weather	env-ccc	运行中	OpenJDK8	2 / 2 (正常全部)	设置	2022-12-07 14:48:59 GMT+0...	详情 事件 更多

----结束

6.3 管理应用环境变量

环境变量是指在系统或用户应用程序中设置的一些参数，在代码里调用接口去获取环境变量的值。在部署时通过环境变量去指定参数配置，而不用在代码中指定，使部署具有灵活性。

应用下添加的环境变量，属于全局环境变量，对此应用下的所有组件都起作用。

如果您需要对应用下特定的组件添加环境变量，请参考[7.17.1 设置组件环境变量](#)。

手动添加应用环境变量

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理”。

步骤3 单击待操作的应用名称，进入“应用概览”页面。

步骤4 在左侧导航栏，单击“环境变量”。

步骤5 在下拉列表选择已经创建的环境。

步骤6 单击“添加环境变量”，输入“变量名称”及“变量/变量引用”的值。

其中：

- “变量名称”是应用环境变量的名称，必须唯一。
- “变量/变量引用”是应用环境变量的值。

例如，设置“变量名称”为“User”，“变量/变量引用”为“admin”。即当程序代码读取“User”环境变量时，获取值为“admin”。例如可以使用admin用户启动子进程，可以用admin用户去读取文件等。实际执行效果视代码而定。

步骤7 单击“提交”，完成应用环境变量手动添加。

----结束

导入应用环境变量文件

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理”。

步骤3 单击待操作的应用名称，进入“应用概览”页面。

步骤4 在左侧导航栏，单击“环境变量”。

步骤5 在“环境”下拉列表选择已经创建的环境。

步骤6 单击“导入”，选择本地已创建好的环境变量文件。

导入的文件应为JSON或YAML格式、字符串形式的键值对映射，例如：

```
{"key1": "value1", "key2": "value2"}
```

其中：

- “key1”、“key2”是应用环境变量的名称，必须唯一。

- “value1”、“value2”是对应应用环境变量的值。

步骤7 单击“提交”，完成应用环境变量文件导入。

----结束

编辑应用环境变量

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理”。

步骤3 单击待操作的应用名称，进入应用概览页面。

步骤4 在左侧导航栏，单击“环境变量”。

步骤5 在“环境”下拉列表选择已经创建的环境。

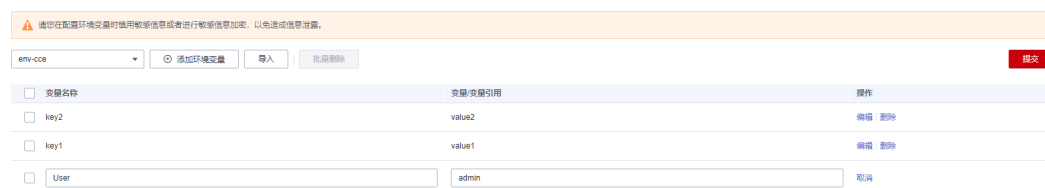
步骤6 选择待编辑的变量名称，在“操作”栏单击“编辑”。

步骤7 重新输入“变量名称”及“变量/变量引用”的值。

- “变量名称”是应用环境变量的名称，必须唯一。
- “变量/变量引用”是应用环境变量的值。

步骤8 单击“提交”，完成应用环境变量编辑。

图 6-3 编辑应用环境变量



----结束

删除应用环境变量

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理”。

步骤3 单击待操作的应用名称，进入应用概览页面。

步骤4 在左侧导航栏，单击“环境变量”。

步骤5 在“环境”下拉列表选择已经创建的环境。

- 删除单个应用环境变量：选择待删除的变量名称，在“操作”栏单击“删除”。

图 6-4 删除单个应用环境变量



- 批量删除应用环境变量：勾选待删除的变量名称，单击“批量删除”。

图 6-5 批量删除应用环境变量



步骤6 在弹出对话框，单击“确定”。

----结束

6.4 编辑应用

已创建的应用，支持修改应用名称和应用描述信息。

操作步骤

- 步骤1 登录ServiceStage控制台。
- 步骤2 选择“应用管理”。
- 步骤3 选择待操作应用，在“操作”列单击“编辑”。
- 步骤4 参考下表重新设置应用信息。

参数	说明
应用名称	输入应用名称。 应用名称必须必须唯一，不能重复。
企业项目	选择企业项目。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 创建企业项目 后可以使用。
描述	输入应用描述信息。

图 6-6 编辑应用

编辑应用

* 应用名称

企业项目 [新建企业项目](#)

描述

0/128

步骤5 单击“确定”。

----结束

6.5 删除应用

当应用不再使用时，可以将其删除。

须知

应用删除后无法恢复，请谨慎操作。

前提条件

删除应用前，需删除该应用下的全部组件，请参考[7.16 删除组件](#)。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择“应用管理”。

步骤3 选择待操作应用，在“操作”列单击“删除”。

图 6-7 删除应用

应用名称	ID	企业项目	创建时间	创建者	操作
app-test	0	default	2022-12-14 17:16:19 GMT+08:00	admin	新建组件 编辑 删除

步骤4 在弹出对话框，单击“确定”。

----结束

7 组件管理

7.1 组件概述

组件是组成应用的某个业务特性实现，以软件包为载体，可独立部署运行。

在ServiceStage上创建应用后，可以在应用中添加组件。同一个应用下，最多可支持创建1000个组件。

您可以根据实际业务需求设置组件技术栈及组件来源，创建并部署组件。

技术栈简介

技术栈包含组件运行所依赖的操作系统、框架和运行时，由技术栈名称、类型、状态、版本等属性构成。其中，版本号遵从[语义化版本控制规范](#)。

ServiceStage为您提供并管理技术栈生命周期，您只需聚焦具体业务开发，提升您的应用托管体验。

技术栈各生命周期阶段定义如下：

- Preview：Beta版本技术栈发布。
- GA (General Availability)：正式版本技术栈发布。
- EOL (End of Life)：技术栈生命周期结束。

技术栈状态定义如下：

- Preview：生命周期处于Preview阶段的技术栈。
- Supported：生命周期处于GA阶段的技术栈。
- Deprecated：生命周期处于GA阶段、但是已经发布了EOL公告的技术栈，或者ServiceStage服务不主推的技术栈。

技术栈的详细信息请参见[表7-1](#)。

表 7-1 技术栈信息说明

技术栈名称	技术栈类型	技术栈状态	发布说明	组件来源和部署方式
OpenJDK8	Java	Supported	<ul style="list-style-type: none"> OpenJDK-8u312 b07: Release Note 镜像OS: EulerOS 2.9.8 	组件来源支持源码或Jar包，支持容器部署，详情请参考 组件部署方式说明 。
Tomcat8/ OpenJDK8	Tomcat	Supported	<ul style="list-style-type: none"> OpenJDK-8u312 b07: Release Note Tomcat-8.5.75: Release Note 镜像OS: EulerOS 2.9.8 	组件来源支持源码或War包，支持容器部署，详情请参考 组件部署方式说明 。
Tomcat9/ OpenJDK8	Tomcat	Supported	<ul style="list-style-type: none"> OpenJDK-8u312 b07: Release Note Tomcat-9.0.58: Release Note 镜像OS: EulerOS 2.9.8 	组件来源支持源码或War包，支持容器部署，详情请参考 组件部署方式说明 。
Node.js8	Node.js	Supported	<ul style="list-style-type: none"> Nodejs-v8.11.3: Release Note 镜像OS: EulerOS 2.9.8 	组件来源支持源码或Zip包，支持容器部署，详情请参考 组件部署方式说明 。
Node.js14	Node.js	Supported	<ul style="list-style-type: none"> Nodejs-v14.18.1: Release Note 镜像OS: EulerOS 2.9.8 	组件来源支持源码或Zip包，支持容器部署，详情请参考 组件部署方式说明 。
Node.js16	Node.js	Supported	<ul style="list-style-type: none"> Nodejs-v16.13.0: Release Note 镜像OS: EulerOS 2.9.8 	组件来源支持源码或Zip包，支持容器部署，详情请参考 组件部署方式说明 。
Docker	Docker	-	CCE提供支持。	组件来源为镜像包，支持容器部署，详情请参考 组件部署方式说明 。

技术栈名称	技术栈类型	技术栈状态	发布说明	组件来源和部署方式
Python3	Python	-	-	组件来源支持源码或Zip包，支持容器部署，详情请参考 组件部署方式说明 。
Php7	Php	-	-	组件来源支持源码或Zip包，支持容器部署，详情请参考 组件部署方式说明 。

组件来源说明

组件来源	说明
源码仓库	参考 9.7 仓库授权 创建授权，然后设置代码来源。
Jar包	支持以下上传方式： 从OBS对象存储选择对应的软件包。需要提前将软件包上传至OBS桶中，相关操作请参考 上传文件 。
War包	支持以下上传方式： 从OBS对象存储选择对应的软件包。需要提前将软件包上传至OBS桶中，相关操作请参考 上传文件 。
Zip包	支持以下上传方式： 从OBS对象存储选择对应的软件包。需要提前将软件包上传至OBS桶中，相关操作请参考 上传文件 。

组件来源	说明
镜像包	<p>容器应用需要基于镜像创建。支持“我的镜像”（私有镜像）和“第三方镜像”。</p> <ul style="list-style-type: none"> 若选择“我的镜像”，您首先需要将镜像上传至镜像仓库，请参考8.1.1 上传镜像。 若选择“第三方镜像”，您需要已获取第三方镜像的地址。镜像地址格式为： {第三方镜像仓库IP地址}:{第三方镜像仓库访问端口号}/{镜像存储路径}/{镜像名称}:{镜像版本号} 或者： {镜像名称}:{镜像版本号} <p>若没标注镜像版本号，则默认为最新版本号latest。</p> <p>目前仅支持获取第三方公开镜像。</p>

组件部署方式说明

组件部署方式	说明
容器部署	<p>云容器引擎部署：云容器引擎CCE提供高度可扩展的、高性能的企业级Kubernetes集群，支持运行Docker容器。借助云容器引擎，您可以在云平台上轻松部署、管理和扩展容器化应用程序。</p>

7.2 创建并部署组件

本节介绍基于ServiceStage创建并部署组件。

ServiceStage支持在同一个应用下创建同名组件。部署组件时，对于同名组件：

- 同一个应用下的同名组件不能部署在同一个环境下。
- 不同应用下的同名组件可以部署在同一个环境下。

前提条件

1. 只能在应用下新增组件，请先创建应用，请参考[6.1 创建应用](#)。
2. 组件需要部署在指定环境下，请先创建环境并纳管资源，请参考[5 环境管理](#)。
3. 需要根据组织管理组件构建生成的镜像，请先创建组织，请参考[创建组织](#)。
4. （可选）如果基于Kubernetes类型的环境创建并部署组件，请先创建命名空间，请参考[创建命名空间](#)。
5. 如果您基于“源码仓库”创建组件，那么您首先需要创建仓库授权，请参考[9.7 仓库授权](#)。

6. 如果您基于软件包创建组件，那么您首先需要将软件包上传至OBS对象存储中。将软件包上传至OBS对象存储中，请参考[上传文件](#)。




操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入“创建组件”页面：

- 选择“组件管理 > 新增组件”。
- 在“应用管理”页面，选择待创建部署组件的应用，单击“操作”列的“新增组件”。
- 在“应用管理”页面，单击待创建部署组件的应用名称，在“应用概览”页面，单击“新增组件”。

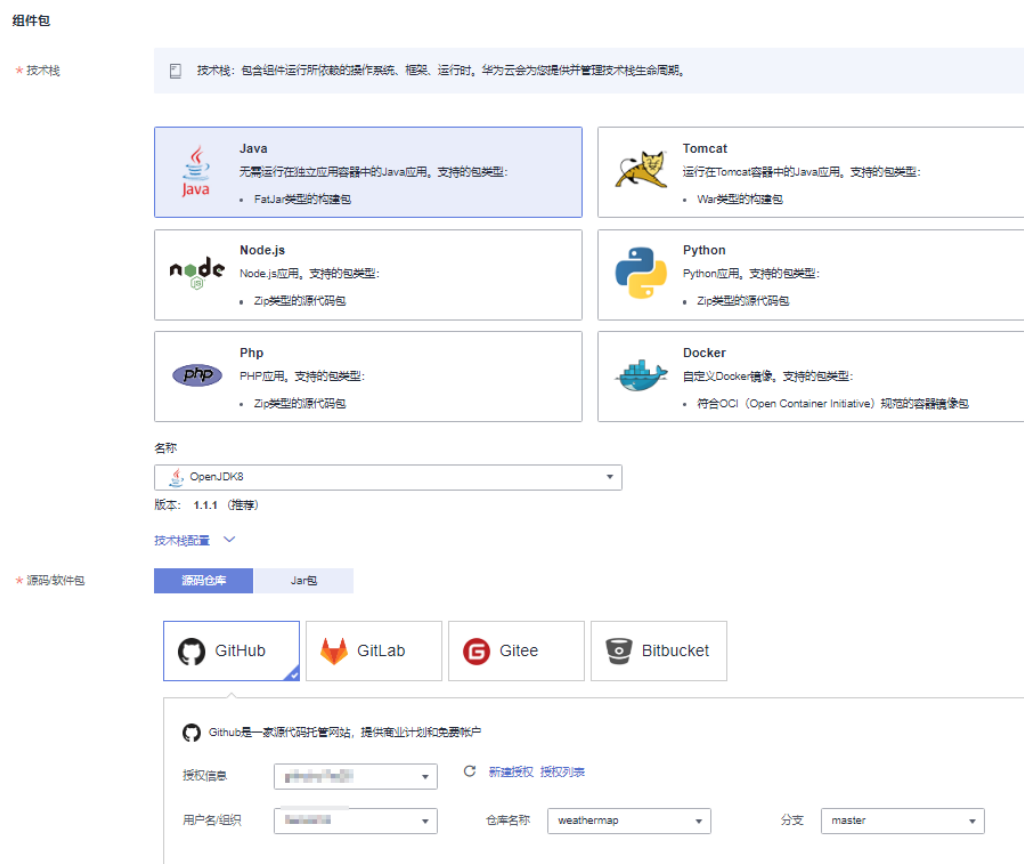
步骤3 在“基本信息”区域，参考下表设置组件基本信息，其中带“*”标志的参数为必填参数。

参数	说明
*组件名称	组件的名称。
*组件版本	<p>组件版本号。</p> <ul style="list-style-type: none"> • 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmms，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 • 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。
*所属环境	组件部署环境。
*部署方式	<p>选择组件的部署方式。</p> <p>当选择的“所属环境”为“虚拟机+Kubernetes”类型时需设置。</p> <p>说明 如果之前版本的环境纳管了CCE集群和虚拟机资源，升级到当前版本后，该环境的类型为“虚拟机+Kubernetes”。</p>
*所属应用	组件所属应用。
描述	<p>组件描述信息。</p> <ol style="list-style-type: none"> 1. 单击 ，在编辑栏输入组件的描述信息。 2. 单击 ，完成组件描述信息设置。 单击 ，取消设置组件描述信息。

步骤4 在“组件包”区域，参考下表设置组件包参数，其中带“*”标志的参数为必填参数。

参数	说明
*技术栈	<ol style="list-style-type: none"> 1. 根据组件部署方式，参考表7-1，选择组件技术栈类型。 2. 在“名称”下拉列表，选择技术栈名称。 3. （可选）设置“JVM”参数，用于配置Java代码运行时的内存参数大小。技术栈类型选择Java、Tomcat时可设置。 单击“技术栈配置”，输入JVM参数，如-Xms256m -Xmx1024m，多个参数以空格间隔，不填则为空。 4. （可选）设置“Tomcat配置”参数，用于配置Tomcat请求路径、端口号等参数。技术栈类型选择Tomcat时可设置。 <ol style="list-style-type: none"> a. 单击“技术栈配置”，勾选“Tomcat配置”，弹出“Tomcat配置”对话框。 b. 单击“使用示例模板”，根据业务要求编辑模板文件。 c. 单击“确定”。
*软件包	<p>选择“源码仓库”，参考9.7 仓库授权创建授权，设置代码来源。选择软件包，则组件来源支持的软件包类型由您选择的技术栈类型决定，具体情况请参考表7-1。</p>
*上传方式	<p>组件来源为软件包或镜像包时，需选择已上传的软件包或镜像包，上传方式请参考组件来源说明。</p>

图 7-1 设置组件包参数



步骤5 在“构建”区域，参考下表设置构建参数，其中带“*”标志的参数为必填参数。

组件部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时需设置。

参数	说明
*编译命令	<p>组件来源是“源码仓库”时，根据业务需要设置“编译命令”。</p> <ul style="list-style-type: none"> 使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。 使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 <p>须知</p> <ul style="list-style-type: none"> 选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。 在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package
*Dockerfile地址	<p>组件来源是“源码仓库”时，根据业务需要设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录（./）所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组织	组织用于管理组件构建生成的镜像。
*构建环境	<p>选择用于构建镜像的环境类型。执行构建的环境，必须是Kubernetes环境，且能够访问Internet。</p> <p>推荐您选择“使用当前环境构建”。如果当前环境的CCE集群不能访问Internet，且您单独规划了构建环境，则可选择“使用独立环境构建”。</p> <ul style="list-style-type: none"> 使用独立环境构建：可使用单独的构建环境进行镜像构建。独立构建环境CCE集群和当前组件部署环境CCE集群的CPU架构必须保持一致，否则会导致组件部署失败。 使用当前环境构建：使用组件所属的部署环境进行镜像构建。当前环境CCE集群的master节点和node节点的CPU架构必须保持一致，否则会导致组件构建失败。
*选择环境	<ul style="list-style-type: none"> “构建环境”选择“使用独立环境构建”时，可选择使用不同于组件所属的独立部署环境。 “构建环境”选择“使用当前环境构建”时，固定选择使用组件所属的部署环境。
*命名空间	选择组件实例所在容器命名空间。

参数	说明
过滤节点标签	通过节点标签可以将构建任务下发到绑定了EIP的固定节点上。新增过滤标签，请参考 管理节点标签 。

图 7-2 设置构建参数

构建

* 组织  

* 构建环境

* 选择环境

温馨提示：执行构建的环境，必须是Kubernetes环境，且能够访问Internet。

* 命名空间  

过滤节点标签

请选择可访问公网（即绑定有EIP）的节点，若无公网IP，请参考[无公网IP访问Internet](#)进行设置，如果没有节点标签，请点击[新增节点标签](#)完成创建。

步骤6 单击“下一步”。

步骤7 在“资源”区域，设置组件所需资源。请参考下表进行设置，其中带“*”标志的参数为必填参数。

参数	说明
*资源需求	组件无法调度到剩余资源小于申请值的节点上，配置方法请参考 资源限制指南 。 可以根据需要自定义“CPU配额”和“内存配额”，设置组件运行可以使用的最大/最小CPU核数（Core）和内存数量（GiB）。 不勾选表示不限制。
*实例数	设置运行在环境下的组件实例数，取值范围为[1, 200]。
*命名空间	选择组件实例所在容器命名空间。

步骤8 （可选）在“访问方式”区域，开启组件公网访问。

单击 ，开启公网访问后，设置以下参数：

- 设置“公网ELB”。
 - 选择所选环境中已经存在的绑定了弹性IP的弹性负载均衡(ELB)资源。
 - 若不存在，请单击“去所选的环境里添加”，在环境编辑页面，单击“新增可选资源”，为该环境添加已创建的弹性负载均衡(ELB)资源。
 - 若需创建弹性负载均衡(ELB)资源，请参考[创建独享型负载均衡器](#)。

说明




- 弹性负载均衡ELB需已绑定弹性IP，且必须与当前组件部署环境下纳管的计算资源处于相同的虚拟私有云VPC及其子网下。
 - 组件在不同部署环境下需绑定不同ELB，避免路由错误。
2. (可选)设置“对外协议”。
 - 选择“HTTP”，会存在安全风险，建议优先选择使用安全的“HTTPS”方式。
 - 选择“HTTPS”，单击“使用已有”选择已经创建的证书。
若证书不存在，请单击“新创建”创建新的服务器证书。创建服务器证书请参考[创建证书](#)。
 3. 设置“域名”。
在输入框中输入自定义域名。
 4. 设置“监听端口”。
输入应用进程的监听端口号，仅容器部署的组件支持设置“监听端口”。

步骤9 (可选)在“时区”区域，设置容器“时区”。

组件部署环境为Kubernetes类型时，可以设置。

容器的时区，默认和容器节点所在区域时区一致。

步骤10 (可选)设置“高级设置”。具体操作请参考下表。

操作类型	操作名称	操作说明
微服务引擎	绑定微服务引擎	<p>绑定微服务引擎，将组件接入微服务引擎。</p> <p>说明</p> <p>以ServiceComb 2.7.8及以后版本或Spring Cloud Huawei 1.10.4-2021.0.x及以后版本微服务框架开发的组件，接入微服务引擎后，在微服务引擎下创建微服务实例时，会在MicroServiceInstance参数的properties参数项中注入以下属性：</p> <ol style="list-style-type: none"> 1. CAS_APPLICATION_ID：组件所属应用ID。 2. CAS_COMPONENT_NAME：组件名称。 3. CAS_ENVIRONMENT_ID：组件部署环境ID。 4. CAS_INSTANCE_ID：组件实例ID。 5. CAS_INSTANCE_VERSION：组件实例版本。 <ol style="list-style-type: none"> 1. 展开“高级设置 > 微服务引擎”。 2. 单击“绑定微服务引擎”。 3. 选择环境下已绑定的微服务引擎实例。 4. 单击“确定”。 <p>说明</p> <p>鼠标移动到已绑定的微服务引擎，可执行以下操作：</p> <ul style="list-style-type: none"> 重新绑定微服务引擎：单击，重新选择待绑定的微服务引擎，单击“确定”。 删除已绑定微服务引擎：单击，删除已绑定微服务引擎。 <ol style="list-style-type: none"> 5. 单击，输入应用进程的监听端口号，可开启多语言接入Mesher服务网格，使用Mesher将非微服务框架开发的组件接入微服务引擎。 <p>说明</p> <ul style="list-style-type: none"> 选择Java、Tomcat或者Docker技术栈开发的非微服务框架组件，您可根据实际业务需要开启Mesher，使用Mesher将组件接入微服务引擎进行微服务注册发现。 选择Python、Php或者Node.js技术栈开发的组件，强制开启Mesher，将组件接入微服务引擎进行微服务注册发现。
分布式缓存	绑定分布式缓存	<p>在分布式系统中，通过分布式缓存服务，提供可扩展和可靠的用户会话管理。</p> <p>展开“高级设置 > 分布式缓存”，绑定分布式缓存，请参考7.17.4 设置分布式缓存。</p>
云数据库	绑定云数据库	<p>应用数据持久化存储，需要用到云数据库。</p> <p>展开“高级设置 > 云数据库”，绑定云数据库，请参考7.17.5 设置云数据库。</p>

操作类型	操作名称	操作说明
组件配置	设置环境变量	<p>环境变量是应用运行环境中设定的一个变量，可以在组件部署后修改，为应用提供极大的灵活性。组件下设置的环境变量，属于局部环境变量，仅对该组件起作用。</p> <p>展开“高级设置 > 组件配置”，设置“环境变量”，请参考7.17.1 设置组件环境变量。</p>
部署配置	设置启动命令	<p>设置“启动命令”后，容器将会以该启动命令启动。</p> <p>展开“高级设置 > 部署配置”，设置“启动命令”，请参考7.17.2 设置组件生命周期。</p>
	设置数据存储	<p>存储是为应用提供存储的组件，支持多种类型的存储，同一个组件可以使用任意数量的存储。</p> <p>展开“高级设置 > 部署配置”，设置“数据存储”，请参考7.17.3 设置数据存储。</p>
	设置生命周期	<p>容器部署的组件，ServiceStage提供了回调函数，在应用的生命周期的特定阶段执行调用，比如组件在停止前希望执行某项操作，就可以注册相应的钩子函数。</p> <p>展开“高级设置 > 部署配置”，设置“生命周期”，请参考7.17.2 设置组件生命周期。</p>
	设置调度策略	<p>容器部署的组件，根据组件的部署特性，ServiceStage把组件拆解成最小的部署实例。应用调度器实时监控应用的实例信息，当发现有新的pod产生需要调度时就会对集群中所有的剩余资源（计算、存储、网络）进行计算，最后得出最适合的调度目标节点。</p> <p>展开“高级设置 > 部署配置”，设置“调度策略”，请参考7.17.6 设置组件实例调度策略。</p>
运维监控	设置日志采集	<p>容器部署的组件，ServiceStage支持配置应用日志策略，你可以在应用运维管理AOM中查看相关运行日志。在部署组件过程中或者组件已经部署，都可以配置日志策略。如果不做任何配置，则系统默认会采集应用标准输出日志。</p> <p>展开“高级设置 > 运维监控”，设置“日志采集”，请参考7.17.7 设置应用日志策略。</p>
	设置健康检查	<p>健康检查是指容器部署的组件在运行过程中，根据需要，定时检查应用健康状况。</p> <p>展开“高级设置 > 运维监控”，设置“健康检查”，请参考7.17.9 设置健康检查。</p>

操作类型	操作名称	操作说明
	设置自定义指标监控	ServiceStage支持获取您的自定义指标，您可使用该方式上报自定义指标。在部署组件过程中或者组件已经部署完成，都可以设置组件自定义指标。云容器引擎部署的组件，支持设置组件自定义指标监控。 展开“高级设置 > 运维监控”，设置“自定义指标监控”，请参考 7.17.8 设置组件自定义指标监控 。

步骤11 单击“创建并部署”。

在组件“部署记录”页面，可查看部署日志，等待组件部署完成。

----结束

7.3 查看组件详情

组件创建并部署完成后，在组件“概览”页面，可查看组件详细信息。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面，查看组件详情。

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

说明

ServiceStage会周期性的从运行组件实例的计算资源（CCE）同步组件信息。如需手动更新同步组件信息，请执行以下操作：

- 选择待操作组件：
 - 在“应用管理”页面，单击组件所属应用名称，进入“应用概览”页面，在“组件列表”中选择待操作的组件。
 - 在“组件管理”页面，选择待操作的组件。
- 在“操作”列选择“操作 > 同步状态”，手动同步更新组件信息。

----结束

7.4 管理组件标签

标签是以key/value键值对的形式附加在工作负载上的。添加标签后，可通过标签对工作负载进行管理和选择，主要用于设置亲和性与反亲和性调度。您可以给多个工作负载打标签，也可以给指定的某个工作负载打标签。

您可以根据业务需求对无状态工作负载、有状态工作负载和守护进程集的标签进行管理，本文以无状态工作负载为例说明如何使用标签管理功能。

如下图，假设为工作负载（例如名称为APP1、APP2、APP3）定义了3个标签：release、env、role。不同工作负载定义了不同的取值，分别为：

- APP 1: [release:alpha;env:development;role:frontend]
- APP 2: [release:beta;env:testing;role:frontend]
- APP 3: [release:alpha;env:production;role:backend]

在使用调度或其他功能时，选择“key/value”值分别为“role/frontend”的工作负载，则会选择到“APP1和APP2”。

图 7-3 标签案例



说明

状态异常的组件，不支持添加标签。

添加组件标签

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面。

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

步骤3 单击“标签管理”。

图 7-4 管理标签



步骤4 单击“添加标签”：

1. 输入“键”、“值”。
输入的“键”名称不能和已有标签的“键”名称重复。
2. 单击“保存”。

图 7-5 添加标签

键	值	操作
<input type="text" value="user"/>	<input type="text" value="admin"/>	删除标签

[+ 添加标签](#)

----结束

删除组件标签

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面。

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

步骤3 单击“标签管理”。

步骤4 选择待删除的标签，单击“操作”列的“删除标签”。

图 7-6 删除标签

键	值	操作
<input type="text" value="user"/>	<input type="text" value="admin"/>	删除标签

[+ 添加标签](#)

步骤5 单击“保存”。

----结束

7.5 管理组件实例

组件创建并部署完成后，在组件“实例列表”页面，可对组件实例进行管理。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“实例列表”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“实例列表”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“实例列表”。

步骤3 在“实例列表”页面，您可执行如下表所示操作。

操作名称	操作说明
重启单个实例	对于部署环境为Kubernetes类型的组件，如果某个实例运行异常，可以通过删除该实例来进行实例重启。 1. 选择待删除实例，单击“操作”列的“删除”。 2. 在弹出对话框，单击“确定”。
查看实例运行监控	通过查看实例运行监控信息，可以了解单个实例运行中的CPU、内存利用率。 1. 单击实例列表待操作实例名称前的▼。 2. 选择“监控”页签，查看该实例运行监控信息。
查看实例运行事件	ServiceStage支持查看指定实例运行过程中发生的事件详情。 1. 单击实例列表待操作实例名称前的▼。 2. 选择“事件”页签，查看该实例运行过程中发生的事件详情。
查看实例运行容器	对于部署环境为Kubernetes类型的组件，ServiceStage支持查看指定实例运行容器信息，包括容器名称、运行状态、挂载镜像等。 1. 单击实例列表待操作实例名称前的▼。 2. 选择“容器”页签，查看该实例运行容器信息。

----结束

7.6 升级单个组件版本配置

7.6.1 单批发布方式升级

组件创建并部署完成后，根据业务需要，可使用“单批发布”方式升级状态为“运行中”、“未就绪”的单个组件版本配置。

“单批发布”方式会一次性升级所有实例，升级过程中组件服务会中断。适用于测试升级场景或者计划停机升级场景，耗时短。

说明

仅部署环境为Kubernetes类型的组件，支持单批发布方式升级。

如需批量升级同一个应用下的多个组件版本配置，请参考[7.7 批量升级组件版本配置](#)。

前提条件

已完成组件创建与部署，请参考[7.2 创建并部署组件](#)。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

步骤3 在页面右上角，单击“升级”。

步骤4 “升级类型”选择“单批发布”。

步骤5 单击“下一步”，参考下表设置组件版本配置信息，其中带“*”标志的参数为必填参数。

参数	说明
技术栈	固定为创建并部署组件时选择的技术栈。
*软件包/镜像	固定为创建并部署组件时选择的组件来源。 选择“源码仓库”，参考 9.7 仓库授权 创建授权，设置代码来源。 选择软件包，则组件来源支持的软件包类型由您选择的技术栈类型决定，具体情况请参考 表7-1 。
*上传方式	选择软件包或者镜像包，重新选择已上传的软件包/镜像包，上传方式请参考 组件来源说明 。
*编译命令	组件来源是源码、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“编译命令”。 <ul style="list-style-type: none">使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 须知 <ul style="list-style-type: none">选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package

参数	说明
*Dockerfile地址	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录(./)所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组件版本	<p>组件的版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmms，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
资源	<p>组件无法调度到剩余资源小于申请值的节点上，配置方法请参考资源限制指南。</p> <p>可以根据需要自定义“CPU配额”和“内存配额”，修改组件运行可以使用的最大/最小CPU核数(Core)和内存数量(GiB)。如需修改，请勾选待修改项后输入新的配置值。</p> <p>不勾选，表示不限制。</p>
JVM参数	<p>技术栈类型选择Java/Tomcat时可设置，用于配置Java代码运行时的内存参数大小。</p> <p>输入JVM参数，如-Xms256m -Xmx1024m，多个参数以空格间隔，不填则为空。</p>
Tomcat配置	<p>技术栈类型选择Tomcat时可设置，用于配置Tomcat请求路径、端口号等参数。</p> <ol style="list-style-type: none"> 勾选“Tomcat配置”，弹出“Tomcat配置”对话框。 单击“使用示例模板”，根据业务要求编辑模板文件。 单击“确定”。
高级设置	<p>请参考步骤10，设置“组件配置”、“部署配置”、“运维监控”参数。</p>

步骤6 单击“升级”。

等待组件状态由“升级/回滚中”转换为“运行中”，表示已成功完成组件版本配置升级。

----结束

后续操作

操作	说明
重新部署组件	根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，重新部署组件，请参考 7.9 重新部署组件 。

7.6.2 滚动发布方式升级

组件创建并部署完成后，根据业务需要，可使用“滚动发布”方式升级状态为“运行中”、“未就绪”的单个组件版本配置。

“滚动发布”方式每次只升级一个或多个实例，升级完成后加入生产环境，不断执行这个过程，直到全部旧版本升级到新版本。升级过程服务不中断。

如需批量升级同一个应用下的多个组件版本配置，请参考[7.7 批量升级组件版本配置](#)。

前提条件

已完成组件创建与部署，请参考[7.2 创建并部署组件](#)。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

步骤3 在页面右上角，单击“升级”。

步骤4 “升级类型”选择“滚动发布”。

步骤5 单击“下一步”，参考下表设置组件版本配置信息，其中带“*”标志的参数为必填参数。

参数	说明
技术栈	固定为创建并部署组件时选择的技术栈。
*软件包/镜像	固定为创建并部署组件时选择的组件来源。 选择“源码仓库”，参考 9.7 仓库授权 创建授权，设置代码来源。 选择软件包，则组件来源支持的软件包类型由您选择的技术栈类型决定，具体情况请参考 表7-1 。
*上传方式	选择软件包或者镜像包，重新选择已上传的软件包/镜像包，上传方式请参考 组件来源说明 。

参数	说明
*编译命令	<p>组件来源是源码、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“编译命令”。</p> <ul style="list-style-type: none"> 使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。 使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 <p>须知</p> <ul style="list-style-type: none"> 选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。 在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package
*Dockerfile地址	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录（./）所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组件版本	<p>组件版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmmss，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
资源	<p>组件部署环境为Kubernetes类型时可设置。</p> <p>组件无法调度到剩余资源小于申请值的节点上，配置方法请参考资源限制指南。</p> <p>可以根据需要自定义“CPU配额”和“内存配额”，修改组件运行可以使用的最大/最小CPU核数（Core）和内存数量（GiB）。如需修改，请勾选待修改项后输入新的配置值。</p> <p>不勾选，表示不限制。</p>
JVM参数	<p>技术栈类型选择Java/Tomcat时可设置，用于配置Java代码运行时的内存参数大小。</p> <p>输入JVM参数，如-Xms256m -Xmx1024m，多个参数以空格间隔，不填则为空。</p>

参数	说明
Tomcat配置	技术栈类型选择Tomcat时可设置，用于配置Tomcat请求路径、端口号等参数。 1. 勾选“Tomcat配置”，弹出“Tomcat配置”对话框。 2. 单击“使用示例模板”，根据业务要求编辑模板文件。 3. 单击“确定”。
高级设置	组件部署环境为Kubernetes类型时可设置。 请参考 步骤10 ，设置“组件配置”、“部署配置”、“运维监控”参数。
*分几批部署	组件部署环境为Kubernetes类型时可设置。 表示分几个批次升级组件实例，取值范围为[1, 总实例数]。总实例数，即组件当前运行的实例数量。 例如，组件总实例数为4，“分几批部署”参数设置为2，则表示会分2批次升级组件版本配置，每批次升级2个组件实例。

步骤6 单击“升级”。

等待组件状态由“升级/回滚中”转换为“运行中”，表示已成功完成组件版本配置升级。

----结束

后续操作

操作	说明
重新部署组件	根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，重新部署组件，请参考 7.9 重新部署组件 。

7.6.3 灰度发布（金丝雀）方式升级

组件创建并部署完成后，根据业务需要，可使用“灰度发布（金丝雀）”方式升级状态为“运行中”、“未就绪”的单个组件版本配置。

“灰度发布（金丝雀）”方式先选择一定比例的实例升级到新版本，将流量引入到新版本，验证新版本功能是否正常，然后通过滚动升级剩余的实例到新版本。可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以减少其影响度。

灰度发布（金丝雀）的类型及具体说明请参见[表7-2](#)。

表 7-2 灰度发布 (金丝雀) 类型及具体说明

类型	说明
微服务灰度发布	适用于微服务组件 (ServiceComb/Spring Cloud)，灰度发布任务作用于微服务，可通过多个微服务协作，共同完成新特性的上线。 1. 要求组件选择了Java、Tomcat或者Docker技术栈。 2. 要求组件绑定了未开启安全认证的微服务引擎且未开启多语言接入Mesher服务网格。 3. ServiceComb要求2.7.8及以后版本；Spring Cloud Huawei要求1.10.4-2021.0.x及以后版本。
ELB灰度发布	适用于通过ELB作为流量入口的任何类型组件；灰度发布任务作用于ELB。 要求组件开放了公网访问并绑定了ELB。

说明

仅部署环境为Kubernetes类型的组件，且组件实例数量在2个及以上时，支持灰度发布 (金丝雀) 升级。

如需批量升级同一个应用下的多个组件版本配置，请参考[7.7 批量升级组件版本配置](#)。

前提条件

已完成组件创建与部署，请参考[7.2 创建并部署组件](#)。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

步骤3 在页面右上角，单击“升级”。

步骤4 “升级类型”选择“灰度发布 (金丝雀)”。

步骤5 单击“下一步”，参考下表设置组件版本配置信息，其中带“*”标志的参数为必填参数。

参数	说明
技术栈	固定为创建并部署组件时选择的技术栈。
*软件包/镜像	固定为创建并部署组件时选择的组件来源。 选择“源码仓库”，参考 9.7 仓库授权 创建授权，设置代码来源。 选择软件包，则组件来源支持的软件包类型由您选择的技术栈类型决定，具体情况请参考 表7-1 。

参数	说明
*上传方式	选择软件包或者镜像包，重新选择已上传的软件包/镜像包，上传方式请参考 组件来源说明 。
*编译命令	<p>组件来源是源码、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“编译命令”。</p> <ul style="list-style-type: none"> 使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。 使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 <p>须知</p> <ul style="list-style-type: none"> 选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。 在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package
*Dockerfile地址	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录(./)所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组件版本	<p>组件版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmmss，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
资源	<p>组件部署环境为Kubernetes类型时可设置。</p> <p>组件无法调度到剩余资源小于申请值的节点上，配置方法请参考资源限制指南。</p> <p>可以根据需要自定义“CPU配额”和“内存配额”，修改组件运行可以使用的最大/最小CPU核数(Core)和内存数量(GiB)。如需修改，请勾选待修改项后输入新的配置值。</p> <p>不勾选，表示不限制。</p>

参数	说明
JVM参数	技术栈类型选择Java/Tomcat时可设置，用于配置Java代码运行时的内存参数大小。 输入JVM参数，如-Xms256m -Xmx1024m，多个参数以空格间隔，不填则为空。
Tomcat配置	技术栈类型选择Tomcat时可设置，用于配置Tomcat请求路径、端口号等参数。 1. 勾选“Tomcat配置”，弹出“Tomcat配置”对话框。 2. 单击“使用示例模板”，根据业务要求编辑模板文件。 3. 单击“确定”。
高级设置	请参考 步骤10 ，设置“组件配置”、“部署配置”、“运维监控”参数。
灰度策略	<ul style="list-style-type: none"> 灰度流量比例：引入到新版本的流量比例。 当前流量比例：引入到当前版本的流量比例。
*首批灰度实例数量	首批灰度发布的实例数量，取值范围为[1, 当前总实例数-1]。当前总实例数，即组件当前运行的实例数量。 例如，当前组件总实例数为6，“首批灰度实例数量”参数设置为1，则表示首批升级组件版本配置的实例数量为1。
剩余实例部署批次	首批灰度发布成功之后，剩余实例滚动升级分多少批次完成。 例如，当前组件总实例数为6，“首批灰度实例数量”参数设置为1，且“剩余实例部署批次”设置为3。则当前剩余实例数为5，那么升级剩余实例会按照2:2:1个实例分批升级。

步骤6 单击“升级”。

等待组件状态由“升级/回滚中”转换为“运行中”，表示已成功完成组件版本配置升级。

----结束

后续操作

操作	说明
查看系统监控	通过灰度发布（金丝雀）升级组件版本配置时，首批灰度发布完成后，选择“系统监控”页签，可以监控灰度版本和当前版本实例的CPU、内存使用情况。
滚动升级剩余实例	通过灰度发布（金丝雀）升级组件版本配置时首批灰度发布成功且验证新版本功能正常后，如需升级剩余组件实例版本到新版本，请执行以下操作： 1. 选择“灰度发布（金丝雀）”类型的“部署记录”。 2. 单击“滚动升级剩余实例”。 3. 在弹出对话框，单击“确定”。 按照 步骤5 设置的升级策略升级剩余的实例到新版本。

操作	说明
回滚组件版本配置	<p>下列情况支持回滚组件版本配置：</p> <ul style="list-style-type: none"> 通过灰度发布（金丝雀）升级组件版本配置时首批灰度发布完成后 组件全部实例的版本配置升级到新版本后 <p>回滚到升级前组件版本配置，请参考7.8 回滚组件版本配置。</p>
重新部署组件	<p>根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，重新部署组件，请参考7.9 重新部署组件。</p>

7.7 批量升级组件版本配置

组件创建并部署完成后，根据业务需要，可对同一应用下的状态为“运行中”、“未就绪”的多个组件通过滚动发布方式重新配置版本并部署。滚动发布方式每次只升级一个或多个实例，升级完成后加入生产环境，不断执行这个过程，直到全部旧版本升级到新版本，升级过程服务不中断。

如需升级单个组件版本配置，请参考[7.6 升级单个组件版本配置](#)。

操作步骤


步骤1 登录ServiceStage控制台。

步骤2 单击“应用管理”。

步骤3 单击待升级组件所在的应用名称，进入“应用概览”页面。

步骤4 勾选“组件列表”中待批量升级的组件，单击“批量升级”。

步骤5 参考下表设置待升级组件版本配置信息。

参数	说明
升级后版本	<p>升级后的组件版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmms，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
软件包/镜像包/源码仓库	<p>单击，重新选择软件包/镜像包/源码仓库，请参考组件来源说明。</p>

参数	说明
部署批次	表示分几个批次升级组件实例，取值范围为[1, 总实例数]。总实例数，即组件当前运行的实例数量。 例如，组件总实例数为4，“部署批次”参数设置为2，则表示会分2批次升级组件版本配置，每批次升级2个组件实例。

单击指定组件“操作”列的，可以取消选择升级该组件。

步骤6 单击“确定”。

等待组件状态由“升级/回滚中”转换为“运行中”，表示已成功完成组件版本配置升级。

----结束

后续操作

操作	说明
回滚组件版本配置	组件全部实例的版本配置升级到新版本后，如需回滚到升级前组件版本，请参考 7.8 回滚组件版本配置 。
重新部署组件	根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，重新部署组件，请参考 7.9 重新部署组件 。

7.8 回滚组件版本配置

根据业务需要，您可以将组件从当前最新版本配置回滚到升级前或重新部署前的版本配置。

已完成回滚的组件，不能再次回滚。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“部署记录”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“部署记录”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“部署记录”。

步骤3 在“部署记录”列表，选择当前最新版本部署记录。

步骤4 单击“回滚”。

步骤5 在弹出的对话框，单击“确定”。

等待回滚完成后，组件版本会回滚到升级前版本。

----结束

7.9 重新部署组件

7.9.1 单批发布方式重新部署

根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，使用“单批发布”方式重新部署组件。

“单批发布”方式会一次性重新部署所有实例，部署过程中组件服务会中断。适用于测试部署场景或者计划停服部署场景，耗时短。

说明

仅部署环境为Kubernetes类型的组件，支持单批发布方式重新部署。

已[7.8 回滚组件版本配置](#)的组件版本配置，不能作为模板重新部署组件。

前提条件

待操作组件已升级过版本配置，请参考[7.6 升级单个组件版本配置](#)或[7.7 批量升级组件版本配置](#)。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“部署记录”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“部署记录”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“部署记录”。

步骤3 在“部署记录”列表，选择待作为配置模板的历史版本部署记录。

步骤4 单击在页面右上方“使用此版本重新部署”，弹出“重新部署”对话框。

步骤5 “部署类型”选择“单批发布”，单击“确定”。

步骤6 参考下表设置组件版本配置信息，其中带“*”标志的参数为必填参数。

参数	说明
技术栈	固定为选择的历史版本配置，不可修改。
*软件包/镜像	固定为创建并部署组件时选择的组件来源。 选择“源码仓库”，参考 9.7 仓库授权 创建授权，设置代码来源。 选择软件包，则组件来源支持的软件包类型由您选择的技术栈类型决定，具体情况请参考 表7-1 。
*上传方式	选择软件包或者镜像包，重新选择已上传的软件包/镜像包，上传方式请参考 组件来源说明 。

参数	说明
*编译命令	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“编译命令”。</p> <ul style="list-style-type: none"> 使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。 使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 <p>须知</p> <ul style="list-style-type: none"> 选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。 在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package
*Dockerfile地址	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录(./)所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组件版本	<p>组件的版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmms，取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
资源	固定为选择的历史版本配置，不可修改。
JVM参数	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件技术栈类型为Java/Tomcat时，可用于配置Java代码运行时的内存参数大小。</p>
Tomcat配置	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件技术栈类型为Tomcat时，可用于配置Tomcat请求路径、端口号等参数。</p>
高级设置	固定为选择的历史版本配置，不可修改。

步骤7 单击“升级”。

在“部署日志”区域，可查看部署进展，等待部署完成。

----结束

7.9.2 滚动发布方式重新部署

根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，使用“滚动发布”方式重新部署组件。

“滚动发布”方式每次只部署一个或多个实例，部署完成后加入生产环境，不断执行这个过程，直到全部旧版本都升级到新版本。重新部署过程服务不中断。

已[7.8 回滚组件版本配置](#)的组件版本配置，不能作为模板重新部署组件。

前提条件

待操作组件已升级过版本配置，请参考[7.6 升级单个组件版本配置](#)或[7.7 批量升级组件版本配置](#)。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“部署记录”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“部署记录”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“部署记录”。

步骤3 在“部署记录”列表，选择待作为配置模板的历史版本部署记录。

步骤4 单击在页面右上方“使用此版本重新部署”，弹出“重新部署”对话框。

步骤5 “部署类型”选择“滚动发布”，单击“确定”。

步骤6 参考下表设置组件版本配置信息，其中带“*”标志的参数为必填参数。

参数	说明
技术栈	固定为选择的历史版本配置，不可修改。
*软件包/镜像	固定为创建并部署组件时选择的组件来源。 选择“源码仓库”，参考 9.7 仓库授权 创建授权，设置代码来源。 选择软件包，则组件来源支持的软件包类型由您选择的技术栈类型决定，具体情况请参考 表7-1 。
*上传方式	选择软件包或者镜像包，重新选择已上传的软件包/镜像包，上传方式请参考 组件来源说明 。

参数	说明
*编译命令	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“编译命令”。</p> <ul style="list-style-type: none"> 使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。 使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 <p>须知</p> <ul style="list-style-type: none"> 选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。 在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package
*Dockerfile地址	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录(./)所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组件版本	<p>组件版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmms，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
资源	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件部署环境为Kubernetes类型时支持此参数。</p>
JVM参数	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件技术栈类型为Java/Tomcat时，可用于配置Java代码运行时的内存参数大小。</p>
Tomcat配置	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件技术栈类型为Tomcat时，可用于配置Tomcat请求路径、端口号等参数。</p>
高级设置	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件部署环境为Kubernetes类型时支持此参数。</p>

参数	说明
*分几批部署	表示分几个批次升级组件实例，取值范围为[1, 总实例数]。总实例数，即组件当前运行的实例数量。 例如，组件总实例数为4，“分几批部署”参数设置为2，则表示会分2批次升级组件版本配置，每批次升级2个组件实例。 组件部署环境为Kubernetes类型时支持此参数。

步骤7 单击“升级”。

在“部署日志”区域，可查看部署进展，等待部署完成。

----结束

7.9.3 灰度发布（金丝雀）方式重新部署

根据业务需要，您可以在“部署记录”列表中选择历史版本配置，并以此版本配置为模板，使用“灰度发布（金丝雀）”方式重新部署组件。

“灰度发布（金丝雀）”方式先选择一定比例的实例升级到新版本，将流量引入到新版本，验证新版本功能是否正常，然后通过滚动升级剩余的实例到新版本。可以保证整体系统的稳定，在初始灰度的时候就可以发现、调整问题，以减少其影响度。

灰度发布（金丝雀）的类型及具体说明请参见[表7-3](#)。

表 7-3 灰度发布（金丝雀）类型及具体说明

类型	说明
微服务灰度发布	适用于微服务组件（ServiceComb/Spring Cloud），灰度发布任务作用于微服务，可通过多个微服务协作，共同完成新特性的上线。 1. 要求组件选择了Java、Tomcat或者Docker技术栈。 2. 要求组件绑定了未开启安全认证的微服务引擎且未开启多语言接入Mesher服务网格。 3. ServiceComb要求2.7.8及以后版本；Spring Cloud Huawei要求1.10.4-2021.0.x及以后版本。
ELB灰度发布	适用于通过ELB作为流量入口的任何类型组件；灰度发布任务作用于ELB。 要求组件开放了公网访问并绑定了ELB。

说明

仅部署环境为Kubernetes类型的组件，且组件实例数量在2个及以上时，支持灰度发布（金丝雀）方式重新部署组件。

已[7.8 回滚组件版本配置](#)的组件版本配置，不能作为模板重新部署组件。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“部署记录”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“部署记录”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“部署记录”。

步骤3 在“部署记录”列表，选择待作为配置模板的历史版本部署记录。

步骤4 单击在页面右上方“使用此版本重新部署”，弹出“重新部署”对话框。

步骤5 “部署类型”选择“灰度发布（金丝雀）”，单击“确定”。

步骤6 参考下表设置组件版本配置信息，其中带“*”标志的参数为必填参数。

须知

通过本操作执行组件微服务灰度发布升级过程中，请勿同时通过CSE执行组件微服务灰度发布，否则会导致本操作失效。

通过CSE执行组件微服务灰度发布，请参考[灰度发布](#)。

参数	说明
技术栈	固定为选择的历史版本配置，不可修改。
*软件包/镜像	固定为创建并部署组件时选择的组件来源。
*上传方式	选择软件包或者镜像包，重新选择已上传的软件包/镜像包，上传方式请参考 组件来源说明 。
*编译命令	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“编译命令”。</p> <ul style="list-style-type: none">使用默认命令或脚本：优先执行代码根目录下的build.sh，不存在则按照所选语言的通用方法编译，如Java语言的mvn clean package。使用自定义命令：根据所选语言自定义编译命令，或修改build.sh文件后选择使用默认命令或脚本。 <p>须知</p> <ul style="list-style-type: none">选择“使用自定义命令”时，请在echo、cat、debug命令中慎用敏感信息或者进行敏感信息加密，以免造成敏感信息泄露。在项目子目录下执行编译命令，需先进入项目子目录，再执行其余脚本命令。例如： cd ./weather/ mvn clean package

参数	说明
*Dockerfile地址	<p>组件来源是“源码仓库”、部署环境为Kubernetes类型且所选择的技术栈类型为Java、Tomcat、Node.js、Python、Php时支持设置“Dockerfile地址”。</p> <p>“Dockerfile地址”即Dockerfile文件相对于项目的根目录(./)所在的目录，Dockerfile文件用于镜像构建。</p> <p>如果未指定“Dockerfile地址”，默认查找项目根目录下的Dockerfile；如果项目根目录下也没有Dockerfile文件，则根据选定的运行环境自动生成Dockerfile。</p>
*组件版本	<p>组件版本号。</p> <ul style="list-style-type: none"> 自动生成版本号：单击“自动生成”，默认以您单击“自动生成”时的时间戳来生成版本号，格式为：yyyy.mmdd.hhmms，s取时间戳中秒数的个位值。例如：时间戳为2022.1214.172318，则版本号为2022.1214.17238。 您也可以自定义版本号，格式为：A.B.C或者A.B.C.D。A、B、C、D为自然数，例如：1.0.0或者1.0.0.0。 <p>须知</p> <ul style="list-style-type: none"> 自定义版本号需唯一，请勿与该组件的历史版本号重复。
资源	固定为选择的历史版本配置，不可修改。
JVM参数	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件技术栈类型为Java/Tomcat时，可用于配置Java代码运行时的内存参数大小。</p>
Tomcat配置	<p>固定为选择的历史版本配置，不可修改。</p> <p>组件技术栈类型为Tomcat时，可用于配置Tomcat请求路径、端口号等参数。</p>
高级设置	固定为选择的历史版本配置，不可修改。
灰度策略	<ul style="list-style-type: none"> 灰度流量比例：引入到新版本的流量比例。 当前流量比例：引入到当前版本的流量比例。
*首批灰度实例数量	<p>首批灰度发布的实例数量，取值范围为[1, 当前总实例数-1]。当前总实例数，即组件当前运行的实例数量。</p> <p>例如，当前组件总实例数为6，“首批灰度实例数量”参数设置为1，则表示首批升级组件版本配置的实例数量为1。</p>
剩余实例部署批次	<p>首批灰度发布成功之后，剩余实例滚动升级分多少批次完成。</p> <p>例如，当前组件总实例数为6，“首批灰度实例数量”参数设置为1，且“剩余实例部署批次”设置为3。则当前剩余实例数为5，那么升级剩余实例会按照2:2:1个实例分批升级。</p>

步骤7 单击“升级”。

在“部署日志”区域，可查看部署进展，等待部署完成。

----结束

7.10 设置组件访问方式

本章节指导您设置组件访问方式。设置成功后，就能够以设置好的方式访问组件提供的服务。

仅部署环境为Kubernetes类型且状态为“运行中”的组件支持设置组件访问方式。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“访问方式”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“访问方式”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“访问方式”。

步骤3 单击“添加服务”，设置如下参数，其中带“*”标志的参数为必填参数。

参数	参数说明
*服务名称	自定义服务名称，可以与组件名称保持一致。
访问方式	设置访问服务的方式，支持： <ul style="list-style-type: none">集群内访问：提供支持TCP/UDP协议的被同一集群内其他服务访问的入口。VPC内网访问：提供支持TCP/UDP协议的可被同一VPC下的其他服务访问的入口。公网访问：提供支持TCP/UDP协议的Internet访问入口，包含弹性IP方式。
VPC内网负载均衡	“访问方式”选择“VPC内网访问”时可开启。
*访问类型	<ul style="list-style-type: none">“访问方式”选择“VPC内网访问”且开启“VPC内网负载均衡”后可设置。“访问方式”选择“公网访问”后可设置。
服务亲和	“访问方式”选择“VPC内网访问”或者“公网访问”后可设置。
*端口映射	设置访问服务的“协议”、“容器端口”、“访问端口”。

步骤4 单击“确定”。

----结束

7.11 修改组件访问域名

对于开启了公网访问并设置了访问域名的组件，部署完成后，可以对域名进行修改。

前提条件

- 对于已创建并部署的组件，状态需为“运行中”才可修改域名。
- 已在域名提供者处获取域名。
- 已获取组件绑定的ELB的弹性公网IP。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“访问方式”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“访问方式”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“访问方式”。

步骤3 单击“设置域名”：

1. 输入已获取的“应用域名”。
2. 输入“监听端口”，仅容器部署的组件支持输入“监听端口”。
3. （可选）开启“HTTPS”。
 - 单击“使用已有”选择已经创建的证书。
 - 单击“新创建”创建新的服务器证书。创建服务器证书请参考[创建证书](#)。

---结束

7.12 设置组件实例伸缩策略

伸缩策略设置后，可基于资源变化、固定时间、固定周期自动触发实例的增减，从而降低人为反复调整资源以应对业务变化和高峰压力的工作量，帮助您节约资源和人力成本。

- **优雅缩容**
仅部署环境为Kubernetes类型的组件，支持设置组件实例优雅缩容策略。
您可以设置一个缩容时间窗，用于在组件实例停止前保存重要数据。取值范围为0-9999秒，默认设置为30秒。例如某应用有两个实例，执行缩容操作，缩容后只剩一个实例，则您仍然可以在设置的时间窗内对即将要停止的实例进行某些操作。
您可以设置最大无效实例数，用于设置每天滚动升级时允许的无效实例数阈值。
- **手动伸缩**
配置完成后立即触发实例数的增减。
- **弹性伸缩-HPA**
仅1.15以上版本CCE集群支持弹性伸缩-HPA。
弹性伸缩-HPA是Kubernetes内置组件，实现Pod水平自动伸缩的功能，即Horizontal Pod Autoscaling。在Kubernetes社区HPA功能的基础上，增加了应用级别的冷却时间窗和扩缩容阈值等功能。



设置优雅缩容策略

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“伸缩”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“伸缩”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“伸缩”。

步骤3 在“伸缩”页面，设置优雅缩容策略。

设置“缩容时间窗”：单击，修改参数，单击生效。

----结束



设置手动伸缩策略

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“伸缩”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“伸缩”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“伸缩”。

步骤3 在“伸缩”页面的“手动伸缩”区域，设置手动伸缩策略。

- 部署环境为Kubernetes类型的组件，请参考以下操作：
 - 单击，修改“实例数”。
 - 单击，实例伸缩操作即可生效。


----结束

设置弹性伸缩-HPA 策略

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“伸缩”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“伸缩”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“伸缩”。

步骤3 在“伸缩”页面，单击“弹性伸缩 - HPA”右侧的，开启伸缩策略配置，进入“伸缩策略配置”页签。

- CCE集群未安装metrics-server插件，请执行**步骤4**。
- CCE集群已安装metrics-server插件，请执行**步骤6**。

步骤4 单击“立即前往配置”，在云容器引擎控制台完成metrics-server插件安装。

为CCE集群安装metrics-server插件，请参考[metrics-server](#)。

步骤5 等待插件安装完成后，返回“伸缩策略配置”页面，刷新页面。

步骤6 设置伸缩策略参数。

1. 策略名称

输入策略名称。伸缩策略设置完成后，策略名称不支持修改。

2. 冷却时间

根据实际业务需要，输入扩容/缩容冷却时间。

策略成功触发后，在扩容/缩容冷却时间内，不会再次触发扩容/缩容。

3. 实例范围

根据实际业务需要，输入最小实例数和最大实例数。

策略成功触发后，工作负载实例将在此实例范围内伸缩。

4. 触发条件

支持通过“界面配置”、“YAML配置”两种方式设置触发条件参数。

- “界面配置”方式

根据实际业务需要，设置“CPU利用率”、“内存利用率”指标的“期望值”和“阈值”（缩容阈值、扩容阈值）。

策略成功触发后，通过向上取整（当前CPU或内存利用率指标值 / 期望值 × 当前运行实例数）来计算需要伸缩的实例数。

- 当前CPU或内存利用率的指标值小于缩容阈值时，触发缩容。

- 当前CPU或内存利用率的指标值大于扩容阈值时，触发扩容。

- “YAML配置”方式

```
metrics:
- type: Resource
  resource:
    name: cpu
    target:
      type: Utilization
      averageUtilization: 50
- type: Resource
  resource:
    name: memory
    target:
      type: Utilization
      averageUtilization: 50
- type: Pods
  pods:
    metric:
      name: packets-per-second
    target:
      type: AverageValue
      averageValue: 1k
- type: Object
  object:
    metric:
      name: requests-per-second
    describedObject:
      apiVersion: networking.k8s.io/v1beta1
      kind: Ingress
      name: main-route
    target:
      type: Value
      value: 10k
```

如以上示例所示，使用YAML格式除了使用CPU（cpu）、内存（memory）的利用率（Utilization）作为指标外，还可以自定义指标参数配置，并支持更多指标，如Pods、Object、External等。

📖 说明

使用“YAML配置”方式配置自定义指标参数，CCE集群需要已安装prometheus插件。

为CCE集群安装prometheus插件，请参考[prometheus](#)。

步骤7 单击“确定”，完成策略设置。

📖 说明

弹性伸缩-HPA策略设置完成后，您可以根据业务需要，执行以下操作：

- [修改弹性伸缩-HPA策略](#)
- [查看弹性伸缩-HPA策略运行情况](#)
- [删除弹性伸缩-HPA策略](#)

----结束

修改弹性伸缩-HPA 策略

您可以根据实际业务需要，编辑已设置好的弹性伸缩-HPA策略，重新设置策略参数。

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“伸缩”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“伸缩”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“伸缩”。

步骤3 在“伸缩”页面，选择“伸缩策略配置”页签，单击“编辑伸缩策略”，重新设置参数。

1. 冷却时间

根据实际业务需要，修改扩容/缩容冷却时间。

2. 实例范围

根据实际业务需要，修改最小实例数和最大实例数。

3. 触发条件

支持通过“界面配置”、“YAML配置”两种方式修改触发条件参数。

- “界面配置”方式

根据实际业务需要，修改“CPU利用率”、“内存利用率”指标的“期望值”和“阈值”（缩容阈值、扩容阈值）。

- “YAML配置”方式

使用YAML格式可以自定义指标参数配置，并支持更多指标，如pods、Object、External等。

📖 说明

使用“YAML配置”方式配置自定义指标参数，CCE集群需要已安装prometheus插件。

为CCE集群安装prometheus插件，请参考[prometheus](#)。

步骤4 单击“确定”，完成修改。

----结束

查看弹性伸缩-HPA 策略运行情况

ServiceStage支持查看已设置好的弹性伸缩-HPA策略的运行状态和运行事件。

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“伸缩”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“伸缩”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“伸缩”。

步骤3 在“伸缩”页面：

- 选择“状态”页签，查看策略运行状态。
- 选择“事件”页签，查看策略运行中发生的事件。

----结束

删除弹性伸缩-HPA 策略

您可以根据实际业务需要，删除不再使用的弹性伸缩-HPA策略。


须知

删除操作无法恢复，请谨慎操作。

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“伸缩”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“伸缩”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“伸缩”。

步骤3 在“伸缩”页面，单击“弹性伸缩 - HPA”右侧的 。

步骤4 单击“确定”，完成策略删除。

----结束

7.13 组件运维

7.13.1 查看组件运行指标

组件创建并部署完成后，您可以进入组件“指标图表”页面，查看组件运行指标的统计结果。



操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“指标图表”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏选择“运维 > 监控概览”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏选择“运维 > 监控概览”。

步骤3 在“指标图表”页面，查看最近一个小时内组件运行指标的统计结果，统计时间间隔1分钟。

- 单击需要暂停统计的组件运行指标页上的，可暂时停止统计该运行指标。
- 单击需要继续统计的组件运行指标页上的，可开始继续统计该运行指标。

----结束

7.13.2 自定义组件运行指标

部署环境为Kubernetes类型的组件，创建并部署后，您可以进入组件“指标图表”页面，自定义需要查看的组件运行指标。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“指标图表”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏选择“运维 > 监控概览”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏选择“运维 > 监控概览”。

步骤3 在“指标图表”页面，单击“设置指标图表”。

- 勾选需要查看的系统指标，选择“统计方式”。
- 取消勾选不再需要查看的系统指标。

单击“已选对象”后的“清除”，可清除所有已选的系统指标。

步骤4 单击“确定”。

----结束

7.13.3 管理组件运行日志

ServiceStage支持查看、搜索、导出组件运行日志记录，用于排查和解决组件运行中出现的问题。


操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“运行日志”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏选择“运维 > 运行日志”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏选择“运维 > 运行日志”。

步骤3 在“运行日志”页面，参考下表执行管理组件运行日志操作。

操作名称	操作说明
查看日志	<ol style="list-style-type: none">1. 选择待查看日志的实例名称。2. 选择待查看的日志文件名称。3. 选择待查看的日志所在时间范围。 如选择“自定义时间段”，时间范围不能大于30天。4. 在搜索框输入搜索关键字，单击, 可查看指定的日志详细内容。
导出日志	<ol style="list-style-type: none">1. 单击“导出文件”。2. 选择需要导出的日志记录条数。3. 打开已导出到本地的“log.txt”文件，查看导出的日志记录。
查看AOM日志	单击“查看AOM日志”，可在应用运维管理AOM中查看组件运行日志。 具体操作，请参考 查看日志文件 。

----结束

7.13.4 设置资源监控告警阈值

组件部署环境为Kubernetes类型时，当您需要对某些资源进行重点监控并在异常情况下及时响应，可对这些重点资源的指标创建阈值规则，以便您在第一时间发现异常并进行处理。

- 在已设的连续周期内，当指标数据满足阈值条件时，系统会发送阈值告警。
- 在已设的连续周期内，当没有指标数据上报时，系统会发送数据不足事件。
- 当由于非工作时间或出差等原因导致不在现场，无法通过ServiceStage控制台即时查询阈值规则状态的变更信息时，可通过启用通知将该变更信息以短消息或邮件的方式自动发送给相关人员。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“阈值告警”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏选择“运维 > 阈值告警”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏选择“运维 > 阈值告警”。

步骤3 单击“阈值规则设置”，参考[表7-4](#)设置阈值规则参数信息。其中带“*”标志的参数为必填参数。

表 7-4 阈值规则参数说明

参数	说明
*阈值名称	阈值规则的名称。 说明 命名必须唯一，且创建后不能再修改。
描述	描述信息。
统计方式	指标数据按照所设置的统计方式进行聚合。
统计周期	指标数据按照所设置的统计周期进行聚合。
选择指标	选择需要监控的指标。
*阈值条件	阈值告警的触发条件，由判断条件 (\geq 、 \leq 、 $>$ 、 $<$) 和阈值组成。 例如，“阈值条件”设置为“ ≥ 80 ”，表示指标的实际值大于等于已设置的阈值80时，系统将产生阈值告警。
连续周期	连续多少个周期满足阈值条件后，发送阈值告警。
告警级别	阈值告警的级别。
发送通知	选择是否发送通知。 <ul style="list-style-type: none">选择“是”（推荐选择），发送邮件或短信通知用户。选择“否”，不会发送邮件或短信通知用户。
*主题名称	“发送通知”选择“是”时，选择已创建的主题后单击其后的▲。 创建主题，请参见 创建主题 。
*触发场景	“发送通知”选择“是”时，发送通知的触发条件。 <ul style="list-style-type: none">出现告警：当出现阈值告警时，系统会将该变更信息以邮件或短信方式发送给指定用户。恢复正常：当告警取消恢复正常时，系统会将该变更信息以邮件或短信方式发送给指定用户。数据不足：当没有指标数据上报时，系统会将该变更信息以邮件或短信方式发送给指定用户。


步骤4 单击“确定”。

---结束

后续操作

阈值规则创建完成后，您还可以按照[表7-5](#)中的步骤来管理阈值告警。

表 7-5 管理阈值告警相关操作

操作	说明
修改阈值告警	当您发现当前的阈值规则设置不合理时，您可以参考如下操作对阈值规则进行修改，以便更好的满足您的业务需求。 <ol style="list-style-type: none">单击阈值告警列表“操作”列的“修改”。在“修改阈值规则”页面中根据界面提示修改阈值规则的相关参数。单击“修改”。
删除阈值告警	随着业务的变动，当您发现当前的阈值规则已不再需要时，您可以参考如下操作删除阈值规则，以便释放更多的阈值规则资源。 <ol style="list-style-type: none">删除阈值规则。<ul style="list-style-type: none">删除单个阈值：单击阈值规则列表“操作”列的“删除”。批量删除阈值规则：选中单个或多个阈值规则前的复选框，单击页面上方的“删除”。在弹出的对话框中单击“确定”。
搜索阈值告警	<ol style="list-style-type: none">在下拉列表框中选择时间段。在页面右上角的搜索框中输入阈值告警名称或描述的关键字。单击，或在键盘上按“Enter”键。 也可单击“高级搜索”设置搜索条件，单击“搜索”进行查询。
查看阈值告警	在已设的连续周期内，当指标数据满足阈值条件时，系统会发送阈值告警。 在阈值告警列表中查看该告警。
查看历史告警	单击阈值规则列表“操作”列的“历史记录”，查看历史告警。
查看数据不足事件	在已设的连续周期内，当没有指标数据上报时，系统会发送数据不足事件。 可在“事件”页面查看该事件，请参考 7.13.5 查看组件运行事件 。

7.13.5 查看组件运行事件

部署环境为Kubernetes类型的组件，支持查看组件运行过程中发生的事件，用于排查和解决组件运行中出现的问题。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“事件”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏选择“运维 > 事件”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏选择“运维 > 事件”。

步骤3 在组件“事件”页面，查看组件运行事件。

- 选择查询时间，可查看指定时间范围内的组件运行事件。
- 输入事件关键字，可搜索查看特定的组件运行事件。

----结束

7.14 查看组件运行环境

组件部署成功以后，您可以在组件“基础设施”页面查看组件运行所依赖的资源（例如CCE集群、微服务引擎等）以及资源的状态及使用情况。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“基础设施”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“基础设施”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“基础设施”。

步骤3 在组件“基础设施”页面，选择组件运行资源，查看资源状态及使用情况。

----结束

7.15 启停组件实例

组件部署成功以后，可以根据需要重新启动或者停止。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“概览”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称。
- 在“组件管理”页面，单击待操作组件名称。

步骤3 进行启停操作。

- 单击“停止”，停止状态为“运行中”、“未就绪”的应用组件。
- 单击“启动”，启动状态为“停止”的应用组件。
- 单击“重启”，重启状态为“运行中”、“未就绪”的应用组件。

----结束

7.16 删除组件

本章节指导您删除不再使用的组件。

须知

删除操作无法恢复，请谨慎操作。

操作步骤

步骤1 登录ServiceStage控制台。

步骤2 在“应用管理”页面，单击组件所属应用名称。

步骤3 在“组件列表”：

- 删除单个组件：选择待删除组件，在“操作”列选择“删除”。
- 批量删除组件：勾选待删除组件，单击“批量删除”。

步骤4 在弹出对话框单击“确定”，完成组件删除。

----结束

7.17 组件高级设置

7.17.1 设置组件环境变量

环境变量是组件运行环境中设定的一个变量，可以在组件部署后[7.6 升级单个组件版本配置](#)时修改，为应用提供极大的灵活性。

应用组件下设置的环境变量，属于局部环境变量，仅对该应用组件起作用。

如果您在该组件所在的应用下添加了应用环境变量，且应用环境变量的变量名称和该应用下的组件环境变量名称相同，则该应用环境变量会被组件环境变量屏蔽，不再对该组件起作用。添加应用环境变量，请参考[6.3 管理应用环境变量](#)。

本节为您介绍组件部署过程中，不同部署方式下的组件环境变量的配置方法。组件部署方式说明，请参考[组件部署方式说明](#)。

容器部署方式

容器部署方式部署组件时，在设置“高级设置”时，请参考本方法添加环境变量。

步骤1 展开“高级设置 > 组件配置”。

步骤2 参考[表7-6](#)添加环境变量。

当前支持四种类型。

表 7-6 环境变量类型

环境变量类型	操作步骤
手动添加	1. 单击“添加环境变量”，选择“手动添加”。 2. 输入新增环境变量的“变量名称”和“变量/变量引用”。

环境变量类型	操作步骤
密钥导入	<ol style="list-style-type: none"> 1. 创建密钥，请参考创建密钥。 2. 单击“添加环境变量”，选择“密钥导入”。 3. 输入“变量名称”。 4. 在“变量/变量引用”的下拉框中选择对应的密钥名称。
配置项导入	<ol style="list-style-type: none"> 1. 创建配置项，请参考创建配置项。 2. 单击“添加环境变量”，选择“配置项导入”。 3. 输入“变量名称”。 4. 在“变量/变量引用”的下拉框中选择对应的配置项。
从文件导入	<p>单击“从文件导入”，选择本地配置文件。</p> <p>导入的文件应为字符串形式的键值对映射，且仅支持json及yaml格式。例如：</p> <pre>{"key1":"value1","key2":"value2"}</pre>

----结束

7.17.2 设置组件生命周期

容器部署组件时，ServiceStage提供了回调函数，在应用的生命周期的特定阶段执行调用，比如应用组件在停止前希望执行某项操作，就可以注册相应的钩子函数。

目前提供的生命周期回调函数如下所示。

- 启动命令：容器将会以该启动命令启动。
- 启动后处理：应用启动后触发。
- 停止前处理：应用停止前触发。

操作步骤

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 单击“启动命令”，设置容器“运行命令”和“运行参数”。

Docker的镜像拥有存储镜像信息的相关元数据，如果不设置“生命周期”命令和参数，应用运行时将运行镜像制作时提供的默认的命令和参数，Docker将这两个字段定义为“Entrypoint”和“CMD”。关于这两个参数的详细信息，请查看Docker的[Entrypoint说明](#)和[CMD说明](#)。

如果在部署应用组件时填写了应用的“运行命令”和“运行参数”，将会覆盖镜像构建时的默认命令“Entrypoint”、“CMD”，规则如[表7-7](#)所示。

表 7-7 启动命令参数说明

镜像 Entrypoint	镜像CMD	应用运行命令	应用运行参数	最终执行
[touch]	[/root/test]	未设置	未设置	[touch /root/test]
[touch]	[/root/test]	[mkdir]	未设置	[mkdir]
[touch]	[/root/test]	未设置	[/opt/test]	[touch /opt/test]
[touch]	[/root/test]	[mkdir]	[/opt/test]	[mkdir /opt/test]

步骤3 单击“生命周期”，设置“启动后处理”和“停止前处理”参数，参数说明如表7-8所示，选择其中一个参数即可。

表 7-8 生命周期参数说明

参数	说明
命令行方式	<p>在组件实例中执行指定的命令，配置为需要执行的命令。命令的格式为 Command Args[1] Args[2]…（Command为系统命令或者用户自定义可执行程序，如果未指定路径则在默认路径下需找可执行程序），如果需要执行多条命令，建议采用将命令写入脚本执行的方式。</p> <p>需要执行的命令示例如下：</p> <pre>exec: command: - /install.sh - install_agent</pre> <p>请在执行脚本中填写： /install.sh install_agent。 这条命令表示组件部署成功后将执行install_agent安装。</p>
Http请求方式	<p>发起一个HTTP调用请求。配置参数如下：</p> <ul style="list-style-type: none"> ● 路径：请求的URL路径，可选项。 ● 端口：请求的端口，必选项。 ● 主机地址：请求的IP地址，可选项，默认是应用所在的节点IP。

----结束

7.17.3 设置数据存储

存储是为应用提供存储的组件，支持多种类型的存储，同一个应用组件可以使用任意数量的存储。

容器部署的组件，支持设置数据存储。

使用场景

表 7-9 存储项使用场景说明

项目	场景
云硬盘存储	<p>EVS目前支持普通I/O、高I/O、超高I/O三种规格。</p> <ul style="list-style-type: none"> 普通I/O：后端存储由SATA存储介质提供，适用于大容量、读写速率要求不高、事务处理较少的场景，如：开发测试、企业办公应用。 高I/O：后端存储由SAS存储介质提供，适用于性能相对较高、读写速率要求高、有实时数据存储需求的场景，如：创建文件系统、分布式文件共享。 超高I/O：后端存储SSD存储介质提供，适用于高性能、高读写速率要求、数据密集型的场景，如：NoSQL、数据仓库（如Oracle RAC、SAP HANA）。
文件存储	文件存储适用于媒体处理、内容管理、大数据和分析工作负载程序等场景。
对象存储	<ul style="list-style-type: none"> 标准存储： 适用于有大量热点文件或小文件，且需要频繁访问（平均一个月多次）并快速获取数据的业务场景。例如云应用、数据分析、内容分析、热点对象等。 低频访问存储： 适用于不频繁访问（平均一年少于12次），但需要快速获取数据的业务场景。例如静态网站托管、备份/活跃归档、作为云服务的存储资源池或者备份存储等。
主机路径挂载	<p>将应用组件所在宿主机的文件目录挂载到应用指定的挂载点中，如应用组件需要访问/etc/hosts则可以使用HostPath映射/etc/hosts等场景。</p> <p>须知 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用组件实例异常。建议挂载在空目录下。若目录不为空，请确保目录下无影响应用组件实例启动的文件。否则文件会被替换，导致应用组件实例启动异常。</p>
临时路径挂载	用于临时存储，生命周期与应用组件实例相同。应用实例消亡时，EmptyDir会被删除，数据会永久丢失。
配置项挂载	将配置项中的key映射到应用中，可以用于挂载配置文件到指定应用组件目录。
密钥挂载	将应用认证信息、应用密钥等敏感信息存储在密钥中，并将密钥挂载到应用组件的指定路径中。

云硬盘存储

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 云存储 > 添加云存储”，参考表7-10设置参数。

表 7-10 云硬盘存储

参数	说明
云存储类型	<p>选择“云硬盘”。</p> <p>云硬盘的使用方式与传统服务器硬盘完全一致。同时，云硬盘具有更高的数据可靠性，更高的I/O吞吐能力和更加简单易用等特点。适用于文件系统、数据库或者其他需要块存储设备的系统软件或工作负载。</p>
分配方式	<ul style="list-style-type: none"> • 使用已有存储 选择已创建的存储。 • 自动分配存储 自动创建存储，需要输入存储的容量。 <ol style="list-style-type: none"> 1. 存储类型选择云硬盘时，需要先选择创建云硬盘的可用区。 2. 选择存储子类型。 高IO：指由SAS存储介质构成的云硬盘。 普通IO：指由SATA存储介质构成的云硬盘。 超高IO：指由SSD存储介质构成的云硬盘。 3. 输入存储容量，单位为GB。请不要超过存储容量配额，否则会创建失败。
添加容器挂载	<ol style="list-style-type: none"> 1. 设置“子路径”、“挂载路径”：输入数据存储挂载到应用上的路径。 须知 <ul style="list-style-type: none"> - 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 - 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 2. 设置“权限”。 <ul style="list-style-type: none"> - 只读：只能读应用路径中的数据存储。 - 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

----结束

文件存储

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 云存储 > 添加云存储”，参考表7-11设置参数。

表 7-11 文件存储

参数	说明
云存储类型	选择“文件存储”。 文件存储适用于媒体处理、内容管理、大数据和分析应用程序等场景。
分配方式	<ul style="list-style-type: none"> • 使用已有存储 选择已创建的存储。 • 自动分配存储 自动创建存储，需要输入存储的容量。 <ol style="list-style-type: none"> 1. 选择存储子类型。 文件存储子类型为NFS。 2. 输入存储容量，单位为GB。请不要超过存储容量配额，否则会创建失败。
添加容器挂载	<ol style="list-style-type: none"> 1. 设置“子路径”、“挂载路径”：输入数据存储挂载到应用上的路径。 须知 <ul style="list-style-type: none"> - 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 - 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 2. 设置“权限”。 <ul style="list-style-type: none"> - 只读：只能读应用路径中的数据存储。 - 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

----结束

对象存储

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 云存储 > 添加云存储”，参考表7-12设置参数。

表 7-12 对象存储

参数	说明
云存储类型	选择“对象存储”。 对象存储有标准存储和低频访问存储两个存储类型，主要适用于大数据分析、原生云应用程序数据、静态网站托管、备份/活跃归档等场景。

参数	说明
分配方式	<ul style="list-style-type: none"> 使用已有存储 选择已创建的存储。 自动分配存储 <ol style="list-style-type: none"> 设置“密钥”。其中： “命名空间”为创建并部署组件时选择的组件实例所在容器命名空间，不可修改。 单击“使用已有密钥”，可选择组件实例所在容器命名空间下的密钥名称。 也可以创建新密钥：输入新密钥名称后，单击“添加密钥文件”，上传已获取到本地的密钥文件，。 选择存储“子类型”。对象存储子类型为标准存储、低频访问存储。
添加容器挂载	<ol style="list-style-type: none"> 设置“子路径”、“挂载路径”：输入数据存储挂载到应用上的路径。 须知 <ul style="list-style-type: none"> 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 设置“权限”。 <ul style="list-style-type: none"> 只读：只能读应用路径中的数据存储。 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

---结束

主机路径挂载

表示在组件上挂载宿主机上的文件或目录。

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 本地磁盘 > 挂载本地磁盘”，参考[表7-13](#)设置参数。

表 7-13 主机路径挂载

参数	说明
本地存储类型	选择“主机路径挂载”。
主机路径	输入主机路径，如/etc/hosts。

参数	说明
容器挂载	<p>1. 设置“子路径”、“挂载路径”：输入数据存储卷挂载到应用上的路径。</p> <p>须知</p> <ul style="list-style-type: none"> - 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 - 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 <p>2. 设置“权限”。</p> <ul style="list-style-type: none"> - 只读：只能读应用路径中的数据存储。 - 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

----结束

临时路径挂载

适用于临时存储、灾难恢复、共享运行时数据等场景，应用组件实例的删除或者迁移会导致临时路径被删除。

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 本地磁盘 > 挂载本地磁盘”，参考[表7-14](#)设置参数。

表 7-14 临时路径挂载

参数	说明
本地存储类型	选择“临时路径挂载”。
磁盘介质	<ul style="list-style-type: none"> • 若勾选“内存”，可以提高运行速度，但存储容量受内存大小限制。适用于数据量少，读写效率要求高的场景。 • 若不勾选“内存”，即存储在硬盘上，适用于数据量大，读写效率要求低的场景。

参数	说明
容器挂载	<ol style="list-style-type: none"> 设置“子路径”、“挂载路径”：输入数据存储挂载到应用上的路径。 须知 <ul style="list-style-type: none"> 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 设置“权限”。 <ul style="list-style-type: none"> 只读：只能读应用路径中的数据存储。 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

----结束

配置项挂载

平台提供应用代码和配置文件的分离，“配置项挂载”用于处理应用组件配置参数。

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 本地磁盘 > 挂载本地磁盘”，参考表7-15设置参数。

表 7-15 配置项挂载

参数	说明
本地存储类型	选择“配置项挂载”。
配置项	选择对应的配置项名称。 创建配置项请参考 创建配置项 。
容器挂载	<ol style="list-style-type: none"> 设置“子路径”、“挂载路径”：输入数据存储卷挂载到应用上的路径。 须知 <ul style="list-style-type: none"> 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 设置“权限”。 <ul style="list-style-type: none"> 只读：只能读应用路径中的数据存储。 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

----结束

密钥挂载

将密钥中的数据挂载到指定的应用组件中，密钥内容由用户决定。

步骤1 部署组件时，在“高级设置”区域，展开“高级设置 > 部署配置”。

步骤2 选择“数据存储 > 本地磁盘 > 挂载本地磁盘”，参考表7-16设置参数。

表 7-16 密钥挂载

参数	说明
本地存储类型	选择“密钥挂载”。
密钥	选择对应的密钥名称。 创建密钥请参考 创建密钥 。
容器挂载	<ol style="list-style-type: none">设置“子路径”、“挂载路径”：输入数据存储挂载到应用上的路径。 须知<ul style="list-style-type: none">请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。设置“权限”。<ul style="list-style-type: none">只读：只能读应用路径中的数据存储。读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。

步骤3 单击“确定”。

----结束

7.17.4 设置分布式缓存

传统单实例应用使用本地会话管理，用户请求产生的会话上下文都被存于进程内存中。在加入负载均衡模块后，多实例的会话需要使用分布式存储进行共享。

ServiceStage提供开箱即用的分布式会话功能，使用分布式缓存服务作为会话持久化层，无需代码更改，即可赋予Tomcat应用，使用express-session的Node.js应用以及使用session handle的PHP应用分布式会话管理能力。

在部署组件过程中，可以在设置“高级设置”时绑定分布式缓存。

前提条件

已创建分布式缓存，请参考[创建Redis实例](#)。

操作步骤

- 步骤1** 展开“高级设置 > 分布式缓存”。
 - 步骤2** 单击“绑定分布式缓存”。
 - 步骤3** 选择环境下已绑定的分布式缓存实例。
如果环境下未绑定分布式缓存实例，单击“去所选的环境里添加”，在环境编辑页面，单击“新增可选资源”，为该环境添加已创建的分布式缓存(DCS)资源。
 - 步骤4** 如果选择的分布式缓存实例访问方式为密码访问，请在“密码”输入栏输入该分布式缓存实例的密码。
 - 步骤5** 单击“确定”。
- 结束

7.17.5 设置云数据库

应用数据持久化存储，需要用到云数据库，ServiceStage提供了基于云计算平台的即开即用、稳定可靠、弹性伸缩、便捷管理的在线云数据库服务（MySQL）。使用云数据库MySQL能够在云中轻松的进行设置和扩展关系型数据库。通过云数据库服务的管理控制台，用户几乎可以执行所有必需任务而无需编程，简化运营流程，减少日常运维工作量，从而专注于开发应用和业务发展。

在部署组件过程中，可以在数据库设置中绑定云数据库，步骤如下。

前提条件

设置云数据库前，需创建云数据库，请参考[创建实例](#)。

操作步骤

- 步骤1** 展开“高级设置 > 云数据库”。
- 步骤2** 单击“绑定云数据库”。
- 步骤3** 选择环境下已绑定的云数据库实例，单击“确定”。
如果环境下未绑定云数据库实例，单击“去所选的环境里添加”，在环境编辑页面，单击“新增可选资源”，为该环境添加已创建的云数据库(RDS)资源。
- 步骤4** 参考下表设置参数，其中带“*”标志的参数为必填参数。

参数	说明
*连接类型	选择连接类型。 <ul style="list-style-type: none">● JNDI: java标准的连接方式。● Spring Cloud Connector: Spring的连接方式。
*数据库名	输入数据库名称。

参数	说明
*数据库帐号	输入数据库帐号
*数据库密码	输入数据库密码。

步骤5 单击“确定”。

----结束

7.17.6 设置组件实例调度策略

根据容器部署组件的部署特性，ServiceStage把应用组件拆解成最小的部署实例。应用调度器实时监控应用的实例信息，当发现有新的pod产生需要调度时就会对集群中所有的剩余资源（计算、网络、中间件）进行计算，最后得出最适合的调度目标节点。

ServiceStage支持多种调度算法，包括：应用与可用区的亲和性调度、应用和节点之间的亲和性调度、应用间的亲和性调度。

您可以根据需要自由组合使用这些策略来实现业务需求。

亲和性概念阐述

在应用没有容器化之前，原先一个虚机上会装多个组件，进程间会有通信。

但在做容器化拆分的时候，通常直接按进程拆分容器。比如业务进程一个容器，监控日志处理或者本地数据放在另一个容器，并且有独立的生命周期。这时如果进程分布在网络中两个较远的点，请求经过多次转发，性能会很差。

亲和性可以实现就近部署，增强网络能力实现通信上的就近路由，减少网络的损耗。

反亲和性主要是出于高可靠性考虑，尽量分散实例，某个节点故障的时候，对应用的影响只是N分之一或者只是一个实例。

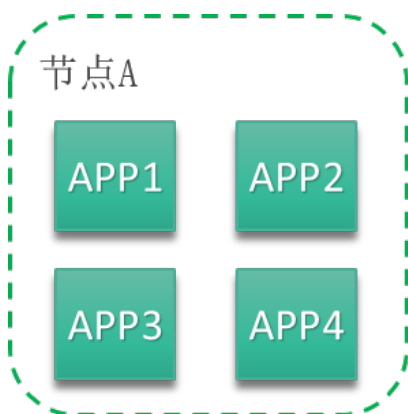
- 应用与可用区的亲和性
 - 亲和：决定应用组件部署在特定的可用区中。
 - 反亲和：决定应用组件不能部署在特定的可用区中。
- 应用与节点间的亲和性
 - 亲和：决定应用组件部署在某些特定的主机中。
 - 反亲和：决定应用组件不能部署在某些特定的主机中。

- 应用间的亲和性

决定应用组件部署在相同或不同节点中。

- 亲和：用户可根据业务需求进行应用组件的就近部署，应用组件间通信就近路由，减少网络消耗。如图7-7所示，APP1、APP2、APP3和APP4部署在相同节点上，为亲和性部署。

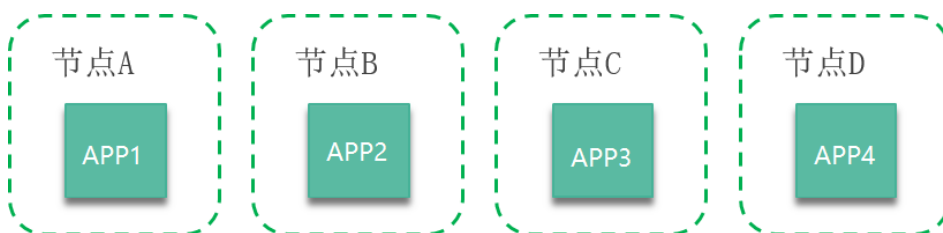
图 7-7 应用间亲和



- 反亲和：同个应用组件的多个实例反亲和部署，减少宕机影响；互相干扰的应用反亲和部署，避免干扰。

如图7-8所示，APP1、APP2、APP3和APP4分别部署在不同节点上，这四个应用为反亲和性部署。

图 7-8 应用间反亲和



注意事项

在设置应用组件间的亲和性和应用组件与节点间的亲和性时，需确保不要出现互斥情况，否则应用会部署失败。例如如下互斥情况，应用将会部署失败：

- APP1、APP2设置了应用组件间的反亲和，例如APP1部署在Node1，APP2部署在Node2。
- APP3部署上线时，既希望与APP2亲和，又希望可以部署在不同节点如Node3上，这就造成了应用组件亲和和节点亲和间的互斥，导致最终应用组件部署失败。

操作步骤

步骤1 展开“高级设置 > 部署配置”。

步骤2 在“调度策略”页签，参考下表设置组件实例调度策略。

目的	步骤
设置应用组件与可用区的亲和	<ol style="list-style-type: none">1. 单击“添加亲和的对象”。2. 对象类型选择“可用区”，选择待亲和的可用区。3. 单击“确定”。

目的	步骤
设置应用组件与可用区的反亲和	<ol style="list-style-type: none"> 1. 单击“添加反亲和的对象”。 2. 对象类型选择“可用区”，选择待亲和的可用区。 3. 单击“确定”。
设置应用组件与节点间的亲和	<ol style="list-style-type: none"> 1. 单击“添加亲和的对象”。 2. 对象类型选择“节点”，选择待亲和的节点。 3. 单击“确定”。
设置应用组件与节点间的反亲和	<ol style="list-style-type: none"> 1. 单击“添加反亲和的对象”。 2. 对象类型选择“节点”，选择待亲和的节点。 3. 单击“确定”。
设置应用组件间亲和	<ol style="list-style-type: none"> 1. 单击“添加亲和的对象”。 2. 对象类型选择“组件”，选择待亲和的应用组件。 3. 单击“确定”。 <p>系统会将您选择的应用组件设置为亲和，部署到相同的节点中。</p>
设置应用组件间反亲和	<ol style="list-style-type: none"> 1. 单击“添加反亲和的对象”。 2. 对象类型选择“组件”，选择待反亲和的应用组件。 3. 设置完成后，单击“确定”。 <p>系统会将您选择的应用组件设置为反亲和，部署到不同的节点中。</p>

----结束

7.17.7 设置应用日志策略

ServiceStage支持为容器部署的应用组件配置应用日志策略，您可以在应用运维管理AOM中查看相关运行日志。

在部署组件过程中，可以配置日志策略。如果不做任何配置，则系统默认会采集应用标准输出日志。


操作步骤

步骤1 展开“高级设置 > 运维监控”。

步骤2 在“日志采集”页签，单击“添加日志策略”并设置如下表所示参数。其中带“*”标志的参数为必填参数。

参数	参数说明
存储类型	选择存储类型。 <ul style="list-style-type: none"> • “主机路径”：将主机上的路径挂载到指定的容器路径。 • “容器路径”：日志仅输出到容器路径，无需挂载主机路径。

参数	参数说明
*主机路径	存储类型为“主机路径”时需要设置。 输入主机上的日志存储路径。

参数	参数说明
*容器挂载	<p>1. 设置“挂载路径”：输入数据存储挂载到应用上的路径。</p> <p>须知</p> <ul style="list-style-type: none"> - 请不要挂载在系统目录下，如“/”、“/var/run”等，会导致应用异常。建议挂载在空目录下，若目录不为空，请确保目录下无影响应用启动的文件，否则文件会被替换，导致应用启动异常，应用创建失败。 - 挂载高危目录的情况下，建议使用低权限帐号启动，否则可能会造成宿主机高危文件被破坏。 <p>2. 设置“主机扩展路径”。存储类型为“主机路径”时需要设置。</p> <ul style="list-style-type: none"> - None：不配置拓展路径。 - PodUID：Pod的ID。 - PodName：Pod的名称。 - PodUID/ContainerName：Pod的ID/容器名称。 - PodName/ContainerName：Pod名称/容器名称。 <p>3. 设置“采集路径”。</p> <p>设置采集路径后，可以精确指定采集内容。采集器ICAgent为5.12.22或以上版本时，支持设置采集路径。当前支持以下设置方式：</p> <ul style="list-style-type: none"> - 不设置，默认采集当前路径下.log.trace.out文件。 - 设置**，表示递归采集5层目录下的.log.trace.out文件。 - 设置*，表示模糊匹配。 <p>例如，采集路径设置为/tmp/**/test*.log，表示采集/tmp目录及其1-5层子目录下的全部以test开头的.log文件。</p> <p>4. 设置“老化周期”。</p> <ul style="list-style-type: none"> - Hourly：每小时扫描一次日志文件。如文件超过20MB，会将其压缩到一个历史文件中并转储到日志所在目录下，并清空原文件。 - Daily：每天扫描一次日志文件。如文件超过20MB，会将其压缩到一个历史文件中并转储到日志所在目录下，并清空原文件。 - Weekly：每周扫描一次日志文件。如文件超过20MB，会将其压缩到一个历史文件中并转储到日志所在目录下，并清空原文件。 <p>5. (可选) 单击，设置“日志格式”。</p> <p>系统默认按行采集显示程序打印的日志。在程序打印日志存在一条完整日志跨占多行的情况下，如果需要采集显示整条日志，您可以设置“日志格式”，开启多行日志。</p> <ul style="list-style-type: none"> - 单行日志：系统默认按行采集日志。 - 多行日志：系统按设置的匹配规则采集日志。 <p>分行模式：“日志格式”选择“多行日志”时需设置，用于匹配每条日志的开头。选择“日志时间”表示采用时间匹配方式；选择“正则模式”表示采用正则表达式匹配。</p> <p>时间通配符：“分行模式”选择“日志时间”时，请输入时间通配符。例如日志中每条日志发生的时间格式是YYYY-MM-</p>

参数	参数说明
	<p>DD hh:mm:ss, 则时间通配符按照规则应该填写YYYY-MM-DD hh:mm:ss。</p> <p>正则: “分行模式”选择“正则模式”时, 请根据日志中每条日志开头的格式输入相应的正则表达式。</p>

说明

组件配置并部署完成后, 可到应用运维管理AOM查看运行日志。具体操作请参考[查看日志文件](#)。

----结束

7.17.8 设置组件自定义指标监控

ServiceStage支持获取容器部署组件时自定义指标, 您可使用该方式上报自定义的组件运行指标。

注意事项

- 当前仅支持获取prometheus的**Gauge指标类型**。
- 在设置应用组件自定义监控前, 您需先了解**prometheus**, 并在您的应用组件中提供获取自定义指标数据的GET接口, 以便ServiceStage能够通过该接口获取您的自定义指标数据。

操作步骤

步骤1 展开“高级设置 > 运维监控”。

步骤2 在“自定义指标监控”页签, 参考下表设置组件自定义指标监控。

参数	说明	是否必须设置
上报路径	exporter提供的供ServiceStage获取自定义指标数据的URL。 例如, /metrics。	是
上报端口	exporter提供的供ServiceStage获取自定义指标数据的端口。 例如, 8080。	是
监控维度	<p>exporter提供的自定义指标名称。 例如, ["cpu_usage","mem_usage"]。</p> <ul style="list-style-type: none"> • 如果不设置, 则ServiceStage会获取所有的自定义指标数据。 • 如果设置, 例如设置为["cpu_usage","mem_usage"], 则ServiceStage会对自定义指标进行过滤, 只获取cpu_usage、mem_usage指标数据。 	否

📖 说明

配置并部署完成后，可到应用运维管理AOM查看监控指标数据。具体操作请参考[指标浏览](#)。

----结束

7.17.9 设置健康检查

健康检查是指容器部署的组件在运行过程中，根据需要定时检查健康状况。

平台提供了两种健康检查的方式：

- **组件存活探针：**该检查方式用于检测应用组件是否存活，类似于执行ps命令检查进程是否存在。如果应用组件的存活检查失败，集群会对该应用组件执行重启操作；若应用组件的存活检查成功则不执行任何操作。
- **组件业务探针：**该检查方式用于检测应用组件是否准备好开始处理用户请求。一些程序的启动时间可能很长，比如要加载磁盘数据或者要依赖外部的某个模块启动完成才能提供服务。这时候程序进程存在，但是并不能对外提供服务。这种场景下该检查方式就非常有用。如果应用组件的就绪检查失败，集群会屏蔽请求访问该组件；若检查成功，则会开放对该应用组件的访问。

健康检查方式

- **HTTP请求检查**

HTTP请求方式针对的是提供HTTP/HTTPS服务的应用组件，集群周期性地对该应用发起HTTP/HTTPS GET请求，如果HTTP/HTTPS response返回码属于200~399范围，则证明探测成功，否则探测失败。使用HTTP请求探测必须指定应用监听的端口和HTTP/HTTPS的请求路径。

例如：提供HTTP服务的应用组件，端口为80，HTTP检查路径为/health-check，主机地址为containerIP，那么集群会周期性地对应用发起如下请求：

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

📖 说明

HTTP请求检查中的主机地址，如果不填写，默认为实例IP。

- **TCP端口检查**

对于提供TCP通信服务的应用，集群周期性地对该应用建立TCP连接。如果连接成功，则证明探测成功，否则探测失败。选择TCP端口探测方式，必须指定应用监听的端口。比如有一个nginx应用组件，它的服务端口是80，对该应用组件配置了TCP端口探测，指定探测端口为80，那么集群会周期性地对该应用组件的80端口发起TCP连接，如果连接成功则证明检查成功，否则检查失败。

- **执行命令检查**

命令检查方式要求用户指定一个应用组件内的可执行命令，集群会周期性地在应用组件内执行该命令，如果命令的返回结果是0则检查成功，否则检查失败。

对于上面提到的TCP端口检查和HTTP请求检查，都可以通过执行命令检查的方式来替代：

- 对于TCP端口探测，可以使用程序来对应用组件的端口进行connect，如果connect成功，脚本返回0，否则返回-1。
- 对于HTTP请求探测，可以使用脚本来对应用组件进行wget：

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

并检查response的返回码，如果返回码在200~399的范围，脚本返回0，否则返回-1。

须知

- 必须把要执行的程序放在应用组件的镜像里面，否则会因找不到程序而执行失败。
- 如果执行的命令是一个shell脚本，由于集群在执行应用组件里的程序时，不在终端环境下，因此不能直接指定脚本为执行命令，需要加上脚本解释器。比如脚本是/data/scripts/health_check.sh，那么使用执行命令检查时，指定的程序应该是sh /data/scripts/health_check.sh。究其原因是集群在执行应用组件里的程序时，不在终端环境下。

公共参数说明

表 7-17 公共参数说明

参数	参数说明
延迟时间/秒	延迟检查时间，单位为秒，此设置与业务程序正常启动时间相关。 例如，设置为30，表明应用启动后30秒才开始健康检查，该时间是预留给业务程序启动的时间。
超时时间/秒	超时时间，单位为秒。如果超过这个时间，本次健康检查就被视为失败。 例如，设置为10，表明执行健康检查的超时等待时间为10秒。若设置为0或不设置，默认超时等待时间为1秒。

操作步骤

步骤1 展开“高级设置 > 运维监控”。

步骤2 在“健康检查”页签，根据业务实际需求设置健康检查参数。

相关公共参数说明请参考[表7-17](#)。

----结束

8 部署源管理

8.1 镜像仓库

8.1.1 上传镜像

创建完组织后，可以上传镜像到组织中，目前支持“客户端上传”和“页面上传”两种方式：

- **页面上传**：直接通过页面将镜像上传到容器镜像服务。
- **客户端上传**：使用命令将镜像上传到容器镜像服务的镜像仓库。

镜像仓库是用于存储、管理docker容器镜像的场所，可以让您便捷的存储、管理、部署docker容器镜像。

前提条件

- 已创建组织，请参见[创建组织](#)。
- 镜像已存为tar或tar.gz文件，请参见[制作镜像压缩包](#)。
- 仅支持上传1.11.2及以上Docker客户端版本制作的镜像压缩包。
- 客户端上传，则上传镜像的容器引擎客户端版本必须为1.11.2及以上。

页面上传镜像

📖 说明

- 每次最多上传10个文件，单个文件大小（含解压后）不得超过2G。
- 文件名称应以大小写字母或数字开头，后面跟大小写字母、数字、下划线、横线或点，长度为1~255个字符。

步骤1 登录ServiceStage控制台。

步骤2 选择“部署源管理 > 镜像仓库”。

步骤3 在“镜像仓库”页面右上角，单击“页面上传”。

步骤4 在弹出的窗口中选择镜像要上传的“组织”，单击“添加文件”，选择要上传的镜像文件。

说明

多个镜像同时上传时，镜像文件会按照顺序逐个上传，不支持并发上传。

步骤5 在弹出的窗口中单击“开始上传”。

待任务进度显示“上传完成”，表示镜像上传成功。

说明

如果镜像上传失败，可能是以下原因：

- 网络异常，请检查网络状况。
- HTTPS证书问题，可按F12键，将请求失败的URL复制到浏览器地址栏，重新打开并同意继续访问，然后返回上传页面尝试重新上传。

----结束

客户端上传镜像

说明

使用客户端上传镜像，镜像的每个layer大小不能超过10G。

步骤1 登录ServiceStage控制台。

步骤2 选择“部署源管理 > 镜像仓库”。

步骤3 在“镜像仓库”页面右上角，单击“客户端上传”。

步骤4 根据界面提示，完成镜像上传。

----结束

8.1.2 管理镜像

获取镜像下载地址


步骤1 登录ServiceStage控制台。

步骤2 选择“部署源管理 > 镜像仓库 > 我的镜像”。

步骤3 在“仓库组织管理”右侧下拉列表框，选择组织。

步骤4 在镜像仓库列表，单击镜像仓库名称，进入镜像仓库详情页。

步骤5 在“镜像版本”页签，可以获取镜像版本下载指令。

单击待下载镜像版本的下载指令右侧的，可以复制下载指令。

----结束

编辑镜像仓库属性

步骤1 登录ServiceStage控制台。

步骤2 选择“部署源管理 > 镜像仓库 > 我的镜像”。

步骤3 在“仓库组织管理”右侧下拉列表框，选择组织。

步骤4 在镜像仓库列表，单击镜像仓库名称，进入镜像仓库详情页。

步骤5 单击右上角“编辑”，根据需要在弹出的窗口中：

- “共享类型”：重新设置“共享类型”为“公共”或“私有”。

说明

公共镜像所有用户都可以下载使用。

- 如果您的节点与镜像仓库在同一区域，访问仓库是通过内网访问。
- 如果您的节点与镜像仓库在不同区域，通过公网才能访问仓库，下载跨区域仓库的镜像需要节点可以访问公网。

- “仓库类型”：重新设置“仓库类型”。
- 更新“描述”：更新镜像仓库描述信息。

步骤6 单击“确认”，完成编辑。

----结束

自动同步镜像

镜像上传后，您可以添加镜像自动同步功能，帮助您把最新推送的镜像自动同步到其他区域镜像仓库内。仅帐号及具有管理员权限的用户才能使用镜像自动同步功能。

说明

镜像自动同步帮助您把最新推送的镜像自动同步到其他区域镜像仓库内，后期镜像有更新时，目标仓库的镜像也会自动更新，但已有的镜像不会自动同步。

已有镜像的同步方法请参见。

步骤1 登录ServiceStage控制台，选择“部署源管理 > 镜像仓库 > 我的镜像”。

步骤2 在“仓库组织管理”右侧下拉列表框，选择组织。

步骤3 在镜像仓库列表，单击镜像仓库名称，进入镜像仓库详情页。

步骤4 单击右上角“自动同步镜像”。

步骤5 在弹出对话框单击“添加”，设置如下参数后，在“操作”栏单击“确定”。

- 目标区域：选择同步的目标区域。
- 目标组织：选择同步的目标组织。
- 是否覆盖：勾选则表示覆盖，同步相同名称的镜像时，同步后会替换已有的镜像。不勾选则表示不覆盖，同步相同名称的镜像时，会取消同步并提示已存在相同名称镜像。

步骤6 添加完成后，单击“确认”。

在镜像详情页面的“镜像同步记录”页签下，可查看镜像同步启动时间、镜像版本、状态、同步类型、同步耗时、目标区域、目标组织、同步者等信息。

----结束

添加镜像授权

在镜像详情中为IAM用户添加授权，授权完成后，该帐号下IAM用户享有读取/编辑/管理该镜像的权限。

- 步骤1** 登录ServiceStage控制台，选择“部署源管理 > 镜像仓库 > 我的镜像”。
- 步骤2** 在“仓库组织管理”右侧下拉列表框，选择组织。
- 步骤3** 在镜像仓库列表，单击镜像仓库名称，进入镜像仓库详情页。
- 步骤4** 在“权限管理”页签，单击“添加授权”，选择IAM用户名称，添加“读取/编辑/管理”的权限，然后单击“确认”。

添加后，该IAM用户享有对应权限。

----结束

删除镜像

须知

删除后无法恢复，请谨慎操作。

- 步骤1** 登录ServiceStage控制台，选择“部署源管理 > 镜像仓库 > 我的镜像”。
- 步骤2** 在“仓库组织管理”右侧下拉列表框，选择组织。
- 步骤3** 在镜像仓库列表，单击镜像仓库名称，进入镜像仓库详情页。
 - 删除镜像仓库
单击页面右上侧的“删除”，可根据提示删除镜像仓库。
 - 删除镜像版本
在目标镜像版本“操作”列，单击“删除”，可根据提示删除镜像版本。
 - 批量删除镜像版本
勾选目标镜像版本，在镜像版本列表上方，单击“删除”，可根据提示删除镜像版本。

----结束

8.2 组织管理

组织概述

组织用于隔离镜像仓库，一个公司或部门可对应一个组织，将其拥有的软件集中在该组织下。在不同的组织下，可以有同名的软件。同一IAM用户可属于不同的组织。组织支持为帐户下IAM用户分配相应的访问权限（读取、编辑、管理）。

创建组织

- 步骤1** 登录ServiceStage控制台，选择“部署源管理 > 组织管理”。
- 步骤2** 单击“创建组织”，在弹出的页面中填写“组织名称”，单击“确定”。

----结束

添加权限

在组织中为用户添加授权，使用户对组织内所有镜像享有读取/编辑/管理的权限。

只有具备“管理”权限的用户才能添加授权。

用户权限分为三类：

- 读取：只能下载软件，不能上传。
- 编辑：下载软件、上传软件、编辑软件属性。
- 管理：下载软件、上传软件、删除软件或版本、编辑软件属性、添加授权以及共享镜像。

步骤1 登录ServiceStage控制台，选择“部署源管理 > 组织管理”。

步骤2 单击组织名称右侧的“添加授权”。

步骤3 在弹出的窗口中为用户选择“权限”，单击“确认”。

----结束

删除组织

步骤1 登录ServiceStage控制台，选择“部署源管理 > 组织管理”。

步骤2 单击组织名称右侧的“删除”。

删除组织前，请先删除组织下的镜像仓库。

删除镜像仓库，请参考[删除镜像](#)。

步骤3 单击“确定”。

----结束

9 持续交付

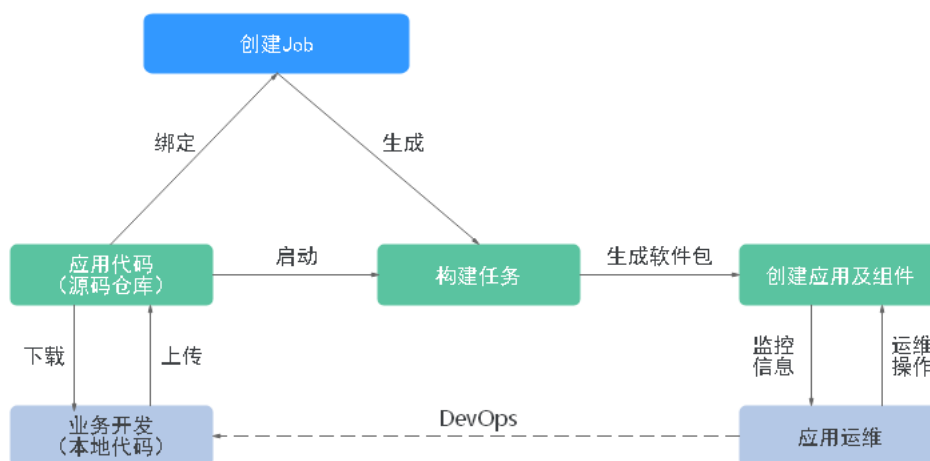
9.1 概述

持续交付用于提供工程构建、发布等功能。

创建构建任务

基于已有业务代码，可以创建构建工程，然后启动构建任务，将业务代码构建打包，归档到部署源，后续部署应用组件时可以直接使用对应的软件包。

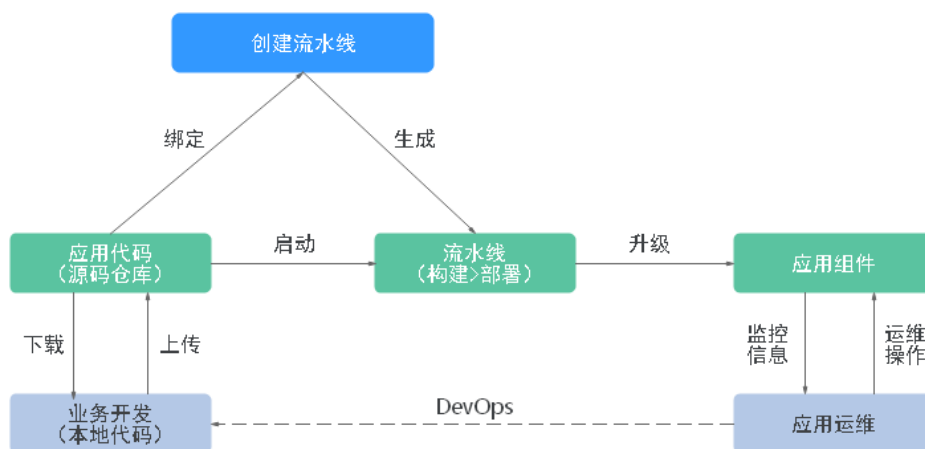
图 9-1 创建构建任务



创建流水线

基于已有业务代码，可以创建流水线，然后启动流水线后完成业务代码构建、升级部署，后续可以在ServiceStage平台上完成应用运维。

图 9-2 创建流水线



9.2 查看构建任务

部署环境为Kubernetes类型的组件，在构建任务列表，您可以查看指定构建任务的构建记录、构建日志，用于指导排查和解决组件部署过程中出现的问题。

操作步骤

- 步骤1** 登录ServiceStage控制台。
- 步骤2** 在左侧导航栏选择“持续交付 > 构建”。
- 步骤3** 在“构建”页面，通过以下任意方式查找指定构建任务：
 - 选择组件部署的CCE集群和构建任务状态，在构建列表选择指定构建任务。
 - 在搜索框输入指定构建任务名称，搜索指定构建任务。
- 步骤4** 单击构建任务名称，进入构建任务详情页面：
 - 查看构建任务“基本信息”和“构建记录”。
 - 单击指定构建记录后的“查看日志”，可以查看“构建详情”、“日志”和“代码检查”详情。

说明

仅Maven构建工程支持代码检查。目前支持的代码检查插件有：checkstyle、findbugs、pmd。

----结束

9.3 创建源码构建任务

通过构建任务可以一键式生成软件包或者镜像包，实现“源码拉取->编译->打包->归档”的全流程自动化。

- 在x86系统构建，构建出来的镜像是x86系统的镜像。
- 在ARM系统构建，构建出来的镜像是ARM系统的镜像。

前提条件

1. 已经创建集群，请参考[创建CCE集群](#)操作。

须知

- 构建任务依赖预置在构建容器中的JDK、Golang、Maven、Gradle、Ant或Node.js编译工具。
- 支持同一个帐号下的不同IAM用户操作同一个构建集群。如需对特定IAM用户取消构建权限，请参考[4.2 创建ServiceStage自定义策略](#)为该用户设置自定义策略，将“servicestage:assembling:create”、“servicestage:assembling:modify”和“servicestage:assembling:delete”权限设置为拒绝策略。

2. 已为构建节点绑定弹性IP，请参考[为弹性云服务器申请和绑定弹性公网IP](#)。

操作步骤

步骤1 登录ServiceStage控制台，选择“持续交付 > 构建”，单击“基于源码构建”。

步骤2 设置基本信息：

1. 输入“名称”。
2. 设置“企业项目”。
企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。
[创建企业项目](#)后可以使用。
3. （可选）输入工程“描述”。
4. 设置“代码源来源”。
 - 参考[9.7 仓库授权](#)创建授权，然后设置代码源来源。
 - 选择“样例库”，选择一个所需的样例。
5. 设置“构建集群”，选择用于构建的集群。
6. （可选）选择“过滤节点标签”，可以通过节点标签将构建任务下发到固定节点上。新增过滤标签请参考[添加节点标签](#)。
7. 单击“下一步”。

步骤3 选择构建模板。


- 选择“Maven”、“Ant”、“Gradle”、“Go”或“Docker”模板构建，可同时编译并归档二进制包或Docker镜像，请执行[步骤4](#)。
- 选择“自定义”构建，可自由定义构建方式，请执行[步骤6](#)。


步骤4 选择归档方式。

- 不归档：不添加Docker构建，也不归档。
- 归档二进制包：不添加Docker构建，归档二进制包。

- 归档镜像编译：添加Docker构建，归档Docker镜像。

步骤5 设置必填项参数。


单击参数配置编辑页的  图标可删除参数设置。

- 构建参数
不同编译参数设置不同，单击编辑框或者编辑框后的 ，可查看参数说明。

- 镜像参数

在  页中输入“任务名称”、“Dockerfile路径”、“镜像名称”和“镜像版本”。

- 镜像归档参数

在  页中设置对应镜像参数的“任务名称”、“归档镜像”、“仓库组织”和“共享类型”，对设置的镜像进行归档。

- 二进制参数

在  页中设置如下参数。

参数名称	参数说明
任务名称	任务的名称。
共享类型	仓库类型分为公有与私有两种。 <ul style="list-style-type: none">- 公有仓库为系统隔离，同一系统中的租户可共享资源。- 私有仓库为租户隔离，当前租户下的用户可共享资源，其他租户不可访问当前租户下的资源。
仓库组织	是指仓库的命名空间。
软件仓库	软件仓库的名称。
软件包名称	构建完成后，归档的软件包名称。
软件包版本	归档的软件包版本。
构建包路径	编译构建完成后，生成的二进制软件包地址。例如，Java项目下的./target/xxx.jar。

步骤6 (可选) 单击“高级配置”进行环境设置。

若要添加多个任务，可在高级配置中自定义添加。

1. 单击左侧栏目中的“添加插件”进入“选择任务类型”界面。
2. 单击目标任务类型的“选择”，添加任务类型。并在“构建环境配置”界面右侧设置任务参数。

须知

在“编译”过程添加了通用构建命令行插件 (Build Common Cmd) 时，需要注意以下注意事项：

- 输入“编译命令”，请在echo、cat、debug等命令中慎用敏感信息或者进行敏感信息加密，以免造成信息泄露。
- “语言”参数选择Python且“Python框架类型”选择符合WSGI标准的Python工程时，需要设置主模块和主模块函数。Python工程主模块和主模块函数名称示例如下：

Python主模块：如果Python工程入口文件名为server.py，主模块名称则为server。

Python主模块函数：如果Python工程入口文件server.py的应用函数名称为app=get_wsgi_application()，则主模块函数名称为app。

步骤7 单击“构建”保存设置并构建工程。

单击“保存”仅保存设置，不进行构建。

----结束

后续操作

应用组件构建成功后，可以把应用组件托管到ServiceStage上，详细操作请参考[组件部署方式说明](#)。

9.4 创建软件包构建任务

通过构建任务可以用软件包一键式生成镜像包，实现“软件包获取->镜像编译->镜像归档”的全流程自动化。

前提条件

1. 已经创建集群，请参考[创建CCE集群](#)操作。

须知

- 构建任务会在CCE集群节点上启动一个构建容器，执行构建相关的操作。为了确保构建的安全性，建议您对CCE集群节点进行安全加固。
- 构建任务依赖预置在构建容器中的JDK、Golang、Maven、Gradle、Ant或Node.js编译工具。
- 支持同一个帐号下的不同IAM用户操作同一个构建集群。如需对特定IAM用户取消构建权限，请参考[4.2 创建ServiceStage自定义策略](#)为该用户设置自定义策略，将“servicestage:assembling:create”、“servicestage:assembling:modify”和“servicestage:assembling:delete”权限设置为拒绝策略。

2. 已为构建节点绑定弹性IP，请参考[为弹性云服务器申请和绑定弹性公网IP](#)。

操作步骤

步骤1 登录ServiceStage控制台，选择“持续交付 > 构建”，单击“基于软件包构建”。

步骤2 输入“Job名称”。

步骤3 设置企业项目。

企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。

[创建企业项目](#)后可以使用。

步骤4 （可选）输入Job“描述”。

步骤5 设置“包来源”。

支持以下上传方式：

- 从OBS对象存储选择对应的软件包，需要提前将软件包上传至OBS桶中，相关操作请参考[上传文件](#)。

单击“选择软件包”，选择对应的软件包。

步骤6 选择构建方式。

- 系统默认
 - a. 选择基础镜像语言，需与[步骤5](#)中选择的软件包编译语言一致。
 - b. 选择“基础镜像版本”。
- 自定义Dockerfile
在编译框中输入自定义命令。

须知

请在echo、cat、debug等命令中慎用敏感信息或者进行敏感信息加密，以免造成信息泄露。

- 内置镜像
选择“基础镜像”，镜像语言需与[步骤5](#)中选择的软件包编译语言一致。

步骤7 设置“镜像类型”。

- 公有：包含常见的标准操作系统镜像，所有用户可见，包括操作系统以及预装的公共应用。请根据您的实际情况自助配置应用环境或相关软件。
- 私有：包含操作系统或业务数据、预装的公共应用以及用户的私有应用的镜像，仅用户个人可见。

步骤8 选择“镜像归档地址”。

步骤9 选择“构建集群”。

使用选择的集群构建任务，可以通过节点标签将构建任务下发到固定节点上，新增过滤标签请参考[添加节点标签](#)。

步骤10 单击“立即构建”启动构建。

单击“保存”仅保存设置，不进行构建。

----结束

后续操作

应用组件构建成功后，可以把应用组件托管到ServiceStage上，详细操作请参考[组件部署方式说明](#)。

9.5 维护构建任务

部署环境为Kubernetes类型的组件，在构建任务列表，您可以对构建任务进行维护操作。

构建任务维护操作说明

表 9-1 维护操作

操作	操作说明
编辑构建任务	请参考 编辑软件包构建工程 或 编辑源码构建工程 。
启动构建任务	请参考 启动构建任务 。
查看详情/构建历史	请参考 9.2 查看构建任务 。
分支/tag构建	请参考 分支/tag构建 。 源码构建任务支持本操作。
删除构建任务	请参考 删除构建任务 。 用户创建的构建任务支持本操作。

编辑软件包构建工程

步骤1 登录ServiceStage控制台。

步骤2 在左侧导航栏选择“持续交付 > 构建”。

步骤3 在“构建”页面，通过以下任意方式查找指定构建任务：

- 选择组件部署的CCE集群和构建任务状态，在构建列表选择指定构建任务。
- 在搜索框输入指定的用户创建的构建任务名称，搜索指定构建任务。

步骤4 选择“更多 > 编辑”，进入构建工程配置界面。

步骤5 输入“Job名称”。

步骤6 （可选）输入“描述”信息。

步骤7 设置“包来源”。

支持以下上传方式：

从OBS对象存储选择对应的软件包。需要提前将软件包上传至OBS桶中，相关操作请参考[上传文件](#)。

步骤8 选择“构建方式”。

- 系统默认
 - a. 选择“基础镜像”语言，需与软件包编译语言一致。
 - b. 选择“基础镜像版本”。

构建节点需要能够访问公网，才能下载基础镜像。
- 自定义Dockerfile
在编译框中输入自定义命令。
- 内置镜像
选择“基础镜像”。

步骤9 设置“镜像类型”。

- 公有：包含常见的标准操作系统镜像，所有用户可见，包括操作系统以及预装的公共应用。请根据您的实际情况自助配置应用环境或相关软件。
- 私有：包含操作系统或业务数据、预装的公共应用以及用户的私有应用的镜像，仅用户个人可见。

步骤10 选择“镜像归档地址”。

步骤11 选择“构建集群”，使用选择的集群进行构建任务。

步骤12 （可选）选择“过滤节点标签”，可以通过节点标签将构建任务下发到固定节点上。
新增过滤标签，请参考[管理节点标签](#)。

步骤13 单击“立即构建”启动构建。

单击“保存”仅保存设置，不进行构建。

----结束

编辑源码构建工程

步骤1 登录ServiceStage控制台。

步骤2 在左侧导航栏选择“持续交付 > 构建”。

步骤3 在“构建”页面，通过以下任意方式查找指定构建任务：

- 创建类别选择“用户创建”后，选择组件部署的CCE集群和构建任务状态，在构建列表选择指定构建任务。
- 在搜索框输入指定的用户创建的构建任务名称，搜索指定构建任务。

步骤4 选择“更多 > 编辑”，进入构建工程配置界面。

步骤5 输入“Job名称”。

步骤6 （可选）输入“描述”信息。

步骤7 单击“修改”，设置“代码源来源”。

您首先需要创建仓库授权，请参考[9.7 仓库授权](#)。

步骤8 选择“构建集群”，使用选择的集群进行构建任务。

1. (可选) 选择“过滤节点标签”，可以通过节点标签将构建任务下发到固定节点上。
2. 单击“下一步”。

步骤9 设置环境。

1. 编辑构建模板
选择“Maven”、“Ant”、“Gradle”、“Go”、“Docker”或“Build Common Cmd”模板构建，可同时编译并归档二进制包或Docker镜像。
2. 选择归档方式
 - Publish Build Artifact: 二进制包归档插件，归档至软件仓库。
 - Publish Build Image: 镜像归档插件，归档至镜像仓库。

步骤10 单击“构建”保存设置并构建工程。

单击“保存”仅保存设置，不进行构建。

----结束

启动构建任务

步骤1 登录ServiceStage控制台。

步骤2 在左侧导航栏选择“持续交付 > 构建”。

步骤3 在“构建”页面，通过以下任意方式查找指定构建任务：

- 选择组件部署的CCE集群和构建任务状态，在构建列表选择指定构建任务。
- 在搜索框输入指定的构建任务名称，搜索指定构建任务。

步骤4 单击“立即构建”，启动构建任务。

----结束

分支/tag 构建

步骤1 登录ServiceStage控制台。

步骤2 在左侧导航栏选择“持续交付 > 构建”。

步骤3 在“构建”页面，通过以下任意方式查找指定构建任务：

- 选择组件部署的CCE集群和构建任务状态，在构建列表选择指定构建任务。
- 在搜索框输入指定的构建任务名称，搜索指定构建任务。

步骤4 单击“更多 > 分支/Tag构建”，设置构建参数。

1. 选择“分支/Tag”。
2. 从下拉框中选择相应的“分支”或“Tag”。
3. 输入对应分支/Tag的CommitId。

步骤5 单击“确定”。

----结束

删除构建任务

步骤1 登录ServiceStage控制台。

步骤2 在左侧导航栏选择“持续交付 > 构建”。

步骤3 在“构建”页面，通过以下任意方式查找指定构建任务：

- 创建类别选择“用户创建”后，选择组件部署的CCE集群和构建任务状态，在构建列表选择指定构建任务。
- 在搜索框输入指定的用户创建的构建任务名称，搜索指定构建任务。

步骤4 选择“更多 > 删除”。

步骤5 单击“确定”。

----结束

9.6 流水线管理

通过流水线工程可以完成一键部署，实现“源码拉取->编译->打包->归档->部署”的全流程自动化。利于集成环境统一、交付流程标准化。

新版流水线中，将原有流水线的“阶段/任务”泛化模型优化为“构建/环境”模型，每条流水线包含一组构建任务和一组或多组环境（如开发环境、类生产环境、生产环境等）任务，每组环境任务又包含一个或多个子任务（如部署、测试等任务），并提供了模板功能。

ServiceStage可支持同一个下的单个用户最多创建100+N个流水线，N为该用户所创建的组件总数。

创建流水线

步骤1 登录ServiceStage控制台，选择“持续交付 > 发布”，单击“创建流水线”。

步骤2 填写流水线基本信息。

1. 输入“流水线名称”。
2. 设置“企业项目”。

企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。

[创建企业项目](#)后可以使用。

3. （可选）输入流水线“描述”。

步骤3 选择流水线模板。

ServiceStage内置了典型场景的流水线模板，选择之后自动生成“构建/环境”模型，用户可以直接使用。

表 9-2 模板说明

模板名称	说明	操作说明
空模板	需要用户自行添加“构建/环境”模型	根据需要设置，操作步骤请参考 步骤3.1 ~ 步骤3.3 。

模板名称	说明	操作说明
简易模板	自动添加“构建”模型，对代码库的源码进行编译构建。	操作步骤请参考 步骤3.1 。
常用模板	自动添加“构建/环境”模型，对代码库的源码进行编译构建，将产生的软件包或镜像，持续发布到应用的生产环境。	操作步骤请参考 步骤3.1~步骤3.3 。

1. 添加构建任务

单击“选择构建”，选择已创建的构建工程，单击“确定”。

若不存在构建任务，选择“选择构建 > 新建构建任务”，新创建源码构建任务或软件包构建任务，具体操作请参考[9.3 创建源码构建任务](#)或[9.4 创建软件包构建任务](#)。

重复此步骤，可以添加多个构建任务。

2. 添加部署任务

单击“添加环境”，输入环境名称。选择已部署的应用组件。

若在该环境下不存在对应应用组件，需创建并部署应用组件，具体步骤请参考[7.2 创建并部署组件](#)。

“选择构建”选择[步骤3.1](#)中添加的构建任务。

选择构建输出。

重复此步骤，可以添加多个环境。

3. 设置流水线审核

单击环境区域的 ，设置“审核模式”及“审核者”。

- 审核模式：支持全部通过和单人通过两种方式。
- 审核者：可以选择多个，系统自动加载该帐号下所有子帐号。

步骤4 流水线编辑完成后，单击“创建并执行”启动流水线。

单击“创建”仅保存设置，不执行流水线。

----结束

设置流水线触发策略

选择“持续交付 > 发布”，进入流水线发布列表界面，可以设置流水线触发策略，说明如下。

表 9-3 触发策略说明

触发策略	触发模式	说明
手动触发	-	选择待触发的流水线任务，单击“启动”，手动启动流水线。
自动触发	-	<p>通过设置代码来源及对应的命名空间、仓库名、分支，当有代码提交至源码仓库对应分支后，自动触发流水线。</p> <p>最多可以设置8个触发来源。</p> <p>操作方法如下：</p> <ol style="list-style-type: none"> 1. 选择流水线，选择“更多 > 触发策略”。 2. “触发类型”选择“自动触发”。 3. 勾选“源码仓库”，推送代码到选择的源码仓库。 4. 单击“确定”。
定时触发	单次触发	<p>设置触发时间，单次定时触发流水线。</p> <p>操作方法如下：</p> <ol style="list-style-type: none"> 1. 选择流水线，选择“更多 > 触发策略”。 2. “触发类型”选择“定时触发”。 3. 设置“触发时间”。 4. 单击“确定”。

触发策略	触发模式	说明
	周期触发	设置触发时间段、间隔、触发周期，可以实现周期触发流水线。 操作方法如下： 1. 选择流水线，选择“更多 > 触发策略”。 2. “触发类型”选择“定时触发”。 3. 开启“周期重复触发”。 4. 设置“重复周期”、“触发时间”、“生效时间”、“触发周期”。 5. 单击“确定”。

克隆流水线

用户可选择克隆流水线，在既有流水线配置基础上，生成新的流水线。

步骤1 登录ServiceStage控制台，选择“持续交付 > 发布”。

步骤2 选择流水线，选择“更多 > 克隆”。

步骤3 ServiceStage自动加载克隆流水线的配置信息，根据需要，参考[创建流水线](#)修改配置参数。

步骤4 流水线编辑完成后，单击“创建并执行”启动流水线。

单击“创建”仅保存设置，不执行流水线。

----结束

后续操作

流水线启动后，可以一键完成应用构建和部署，应用组件部署后的维护操作请参考[7.13 组件运维](#)。

9.7 仓库授权

用户可以新建仓库授权，使构建工程、应用组件等可以使用授权信息访问源码仓库。

步骤1 登录ServiceStage控制台，选择“持续交付 > 仓库授权”，单击“新建授权”。

步骤2 参考下表配置授权信息。参数前面带*号的是必须设置的参数。

表 9-4 授权参数说明

参数	说明
*授权名称	授权名称，创建之后不可更改
*仓库类型	<p>支持以下官方仓库类型：</p> <ul style="list-style-type: none">● GitHub (https://github.com) 授权方式支持：OAuth/私人令牌。● Gitee (https://gitee.com) 授权方式支持：OAuth/私人令牌。● Bitbucket (https://bitbucket.org) 授权方式支持：OAuth/私有Bitbucket。● GitLab (https://gitlab.com) 授权方式支持：OAuth/私人令牌。 <p>说明</p> <p>ServiceStage支持通过私人令牌授权方式访问官方和私有GitLab源码仓库：</p> <ul style="list-style-type: none">- 访问官方GitLab源码仓库，请根据页面提示获取并输入私人令牌，勾选“验证令牌正确性（要求仓库地址可从Internet网络访问）”。- 访问私有GitLab源码仓库，请根据页面提示输入正确的私有GitLab源码仓库地址和私人令牌，无需勾选“验证令牌正确性（要求仓库地址可从Internet网络访问）”。

步骤3 单击“创建”，完成仓库授权创建。

----结束

10 微服务引擎

10.1 微服务引擎概述

微服务引擎 (Cloud Service Engine, CSE) 提供服务注册、服务治理、配置管理等全场景能力; 帮助用户实现微服务应用的快速开发和高可用运维。支持多语言、多运行时; 支持Spring Cloud、Apache ServiceComb Java Chassis (Java Chassis) 框架。

您可以直接使用名称为“Cloud Service Engine”的微服务引擎专业版, 也可以创建微服务引擎专享版。

- 微服务引擎专享版采用物理隔离的方式部署, 租户独占微服务引擎。
- 微服务引擎专业版不支持多可用区 (AZ)。
- 微服务引擎专享版在创建时可以设置多可用区 (AZ)。
- 创建微服务引擎后, 可用区不支持修改, 请根据需要设置。
- 不支持创建跨CPU架构的微服务引擎专享版实例。

10.2 创建微服务引擎

本章节指导您根据实际业务需求创建微服务引擎。

前提条件

- 微服务引擎运行于虚拟私有云, 创建微服务引擎前, 需保证有可用的虚拟私有云和子网。
创建虚拟私有云和子网, 请参考[创建虚拟私有云和子网](#)。
添加安全组规则, 请参考[添加安全组规则](#)。

表 10-1 默认安全组 cse-engine-default-sg 规则说明

方向	优先级	策略	协议端口	类型	源地址
入方向	1	允许	ICMP: 全部	IPv6	::/0

方向	优先级	策略	协议端口	类型	源地址
	1	允许	TCP : 30100-30130	IPv6	::/0
	1	允许	全部	IPv6	cse-engine- default-sg
	1	允许	TCP : 30100-30130	IPv4	0.0.0.0/0
	1	允许	ICMP : 全部	IPv4	0.0.0.0/0
出方向	100	允许	全部	IPv4	0.0.0.0/0
	100	允许	全部	IPv6	::/0


操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 单击页面上方的“创建微服务引擎”。

步骤3 参考下表设置参数，参数前面带*号的是必须设置的参数。

参数	说明
*计费模式	选择计费方式，目前支持“按需计费”。
*企业项目	选择微服务引擎所在的企业项目。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 创建企业项目 后可以使用。 说明 微服务引擎创建成功后，不支持修改企业项目。
*规格	选择引擎规格。
*引擎类型	选择微服务引擎的类型。 引擎类型为集群，其为集群模式部署，主机级容灾。
*引擎名称	输入微服务引擎的名称，引擎创建后不能修改名称。
*可用区	选择可用区。 根据环境可用区数量，为引擎选择1个或者3个可用区。 <ul style="list-style-type: none"> 选择1个可用区，可提供主机级别容灾能力。 选择3个可用区，可提供可用区级别容灾能力。 说明 <ul style="list-style-type: none"> 已创建的微服务引擎不支持更换可用区。 同一个区域内的可用区内网互通。 多可用区可加强容灾能力。

参数	说明
*网络	为引擎选择已创建的虚拟私有云及其子网，可以为您的引擎构建隔离的、自主配置和管理的虚拟网络环境。
描述	单击  ，输入引擎描述信息。
安全认证	<p>开启了“安全认证”的微服务引擎专享版，通过微服务控制台提供了基于RBAC (Role-Based Access Control, 基于角色的访问控制) 的系统管理功能。</p> <ul style="list-style-type: none"> 选择“开启安全认证”： <ol style="list-style-type: none"> 根据业务需要确认是否需要开启“编程接口安全认证”。开启编程接口安全认证后，需要在微服务的配置文件中添加对应用户的帐号密码，否则服务无法注册到引擎。 关闭编程接口安全认证，微服务的配置文件中无需配置帐号密码即可将服务注册到引擎，效率性能更高，建议用于VPC内访问时使用。 输入root帐号的“密码”，并在“再次输入密码”输入框输入密码进行确认。 密码请妥善保管，以免遗失。 选择“关闭安全认证”： <p>关闭安全认证功能，可以在实例创建完成后再设置开启。</p>

步骤4 单击“立即创建”，进入引擎信息确认界面。

步骤5 单击“提交”，等待引擎创建完毕。

说明

- 微服务引擎创建完成，大约需要31分钟。
- 微服务引擎创建成功后，“状态”为“可用”。查看微服务引擎状态，请参考[10.3.1 查看微服务引擎信息](#)。
- 如果微服务引擎创建失败，可在操作日志页面上查看失败原因并处理后可进行以下操作：
 - 可在“微服务引擎信息”区域，单击“重试”重新创建。
 - 如果重试失败，可删除创建失败的微服务引擎，删除微服务引擎，请参考[10.3.10 删除微服务引擎](#)。

----结束

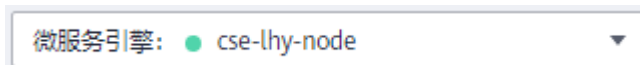
10.3 管理微服务引擎

10.3.1 查看微服务引擎信息

在“微服务引擎信息”区域，可以查看如[表10-2](#)所示的引擎信息。

操作步骤

步骤1 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤2 在“微服务引擎信息”区域，查看如表10-2所示的微服务引擎信息。

表 10-2 引擎详细信息说明

引擎信息名称	信息说明
引擎名称	创建微服务引擎时输入的引擎名称。
引擎ID	引擎的ID，单击可复制。
状态	引擎的状态，包括： <ul style="list-style-type: none">● 创建中● 可用● 不可用● 删除中● 升级中● 变更中● 创建失败● 删除失败● 升级失败● 变更失败
引擎版本	引擎的版本号。
规格	创建微服务引擎时选择的引擎规格。
企业项目	创建微服务引擎时选择的企业项目。 创建企业项目后可以使用。
可用区	创建微服务引擎时选择的可用区。
描述	创建微服务引擎时输入的引擎描述信息。

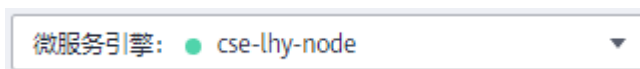
----结束

10.3.2 获取微服务引擎服务注册发现地址

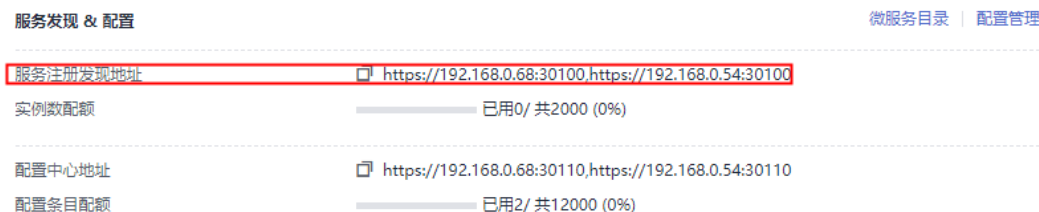
本章节指导您获取微服务引擎的服务注册发现地址。

操作步骤

步骤1 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤2 在“服务发现 & 配置”区域，查看获取引擎服务注册发现地址。



----结束

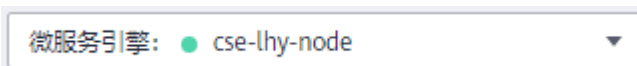
10.3.3 获取微服务引擎配置中心地址

本章节指导您获取微服务引擎的配置中心地址。

操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“服务发现 & 配置”区域，查看获取引擎的配置中心地址。



说明

- 当微服务引擎为1.x版本时，其配置中心地址端口号为30103。
- 当微服务引擎为2.x版本时，其配置中心地址端口号为30110。

----结束

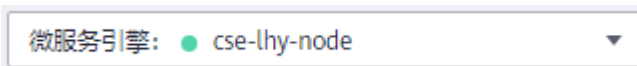
10.3.4 查看微服务引擎实例数配额

本章节指导您查看微服务引擎的实例数配额及配额使用情况。

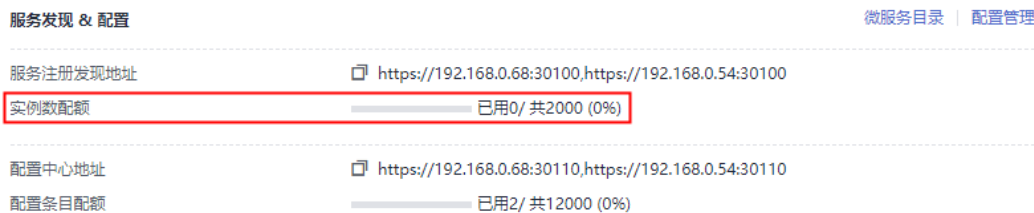
操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“服务发现 & 配置”区域，查看引擎的实例数配额及配额使用情况。



----结束

10.3.5 查看微服务引擎配置条目配额

本章节指导您查看微服务引擎的配置条目配额及配额使用情况。

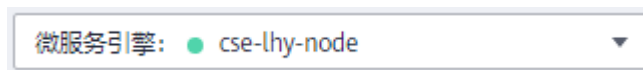
📖 说明

该章节内容只适用于微服务引擎版本为2.x。

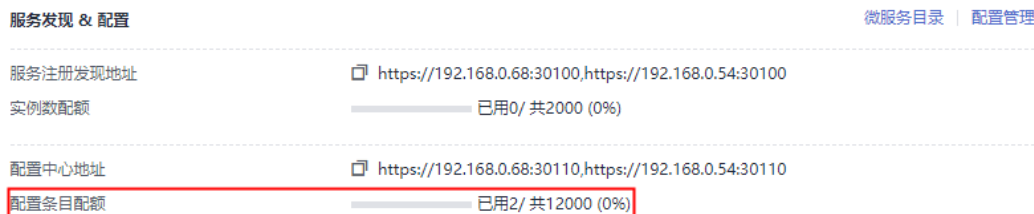
操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“服务发现 & 配置”区域，查看引擎的配置条目配额及配额使用情况。



----结束

10.3.6 设置微服务引擎备份与恢复

ServiceStage控制台提供备份与恢复功能，可对微服务引擎的数据，包括微服务、契约、配置、帐号角色信息进行备份与恢复。

您可自定义备份策略，对微服务引擎进行定期备份，也支持手动对微服务引擎进行备份。

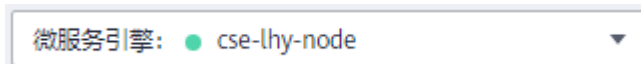
背景介绍

- 每个微服务引擎专享版最多有15个成功的备份数据，其中手动备份最多10个，自动备份最多5个。
- 备份数据会保留10天，超过之后会删除过期的备份数据。

自动备份

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“备份与恢复”区域，单击“自动备份设置”，配置备份参数。

表 10-3 自动备份参数说明

参数	说明
自动备份	关闭自动备份后将会删除之前设置的备份策略，需要重新设置。
备份周期	设置备份周期。 开启“自动备份”后生效。
触发时间	设置备份任务开启的时间，仅支持整点。 开启“自动备份”后生效。

步骤4 单击“确定”，完成备份策略的配置。

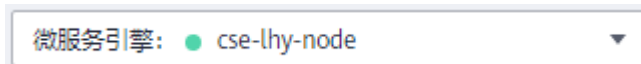
设置备份策略后，将在预设时间之后1小时内触发备份任务。

----结束

手动备份

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“备份与恢复”区域，单击“新增手动备份”，配置备份参数。

表 10-4 手动备份参数说明

参数	说明
名称	输入备份任务的名称，创建备份任务后不能修改名称。
备注	(可选) 添加描述信息。

步骤4 单击“确定”，立即执行备份任务。

----结束

恢复备份数据

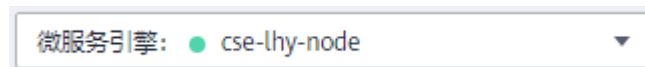
须知

备份数据恢复会使用备份的历史数据覆盖微服务引擎当前数据，可能导致微服务、服务实例错乱，动态配置丢失等问题，请谨慎操作！

若微服务引擎开启了安全认证，则备份数据中包含其帐号信息，建议关闭安全认证后再进行恢复备份数据，否则恢复后可能存在访问微服务引擎认证失败的问题。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“备份与恢复”区域，单击指定备份数据后面“操作”列下的“恢复备份”。

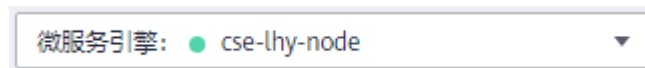
1. 勾选“我已经阅读并充分了解该风险”。
2. 单击“确定”，恢复备份数据。可在“备份与恢复”区域单击“恢复记录”，查看恢复状态。

----结束

删除备份数据

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“备份与恢复”区域，单击指定备份数据右侧“操作”列下的“删除”，在弹出的对话框中输入“DELETE”，单击“确定”。

----结束

10.3.7 管理微服务引擎公网访问

10.3.7.1 绑定公网 IP

未绑定公网IP的微服务引擎专享版，绑定公网IP后可提供公网访问引擎的能力。

须知

未开启安全认证的微服务引擎无认证鉴权能力，开放到公网面临安全风险，增加系统的脆弱性。如：配置、服务信息等数据资产可能会被窃取。

请不要在生产环境和安全要求较高的网络环境中使用该功能。

前提条件

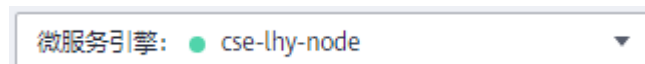
在绑定公网IP前，需已创建可用的弹性公网IP。

创建弹性公网IP，请参考[为弹性云服务器申请和绑定弹性公网IP](#)。

操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“网络配置 & 安全”区域，单击“绑定公网IP”。

步骤4 阅读弹出的对话框中的安全风险提示，勾选“我已知晓安全风险”。

步骤5 在“选择公网IP”下拉列表，选择待绑定的公网IP。

步骤6 单击“确定”，完成绑定。

----结束

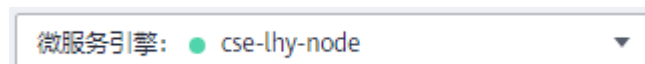
10.3.7.2 解绑公网 IP

已绑定公网IP的微服务引擎专享版，可通过解绑公网IP取消公网访问引擎的能力。

操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“网络配置 & 安全”区域，单击“解绑公网IP”。

步骤4 单击弹出的对话框中的“确定”，完成解绑。

----结束

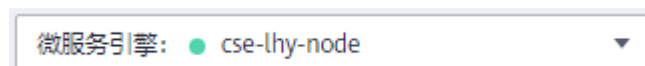
10.3.8 查看微服务引擎操作日志

在“操作日志”区域，可以查看微服务引擎的操作日志信息。

操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。


步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 在“操作日志”区域，查看微服务引擎的操作日志信息。



序号	任务类型	操作结果	用户名	启动时间	结束时间	详细信息
1	创建	成功	admin_user_login11@163.com	2021/04/16 20:47:55 GMT+0800	2021/04/16 21:00:32 GMT+0800	创建引擎 更多

- 单击右上角的 ，可以选择查看某时间区间的操作日志。
- 单击日志列表中指定操作日志“详细信息”列的“更多”，可查看该操作日志的详细信息。

----结束

10.3.9 升级微服务引擎版本

创建微服务引擎时，使用的是当前最新的引擎版本。当后续再发布新版本的引擎时，用户可以选择升级引擎版本。

须知

仅微服务引擎专享版支持升级引擎版本。升级后，不支持版本回退。

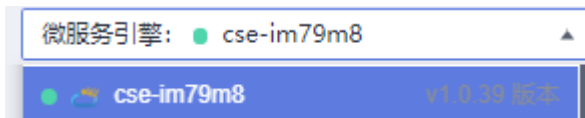
背景介绍

升级过程中，支持两个实例滚动升级，服务不中断。但是可能存在两个接入地址中一个不可用的情况，从而导致该实例不可用，需要支持快速切换到另外一个可用实例。当前，ServiceComb SDK和Mesher都支持快速切换实例。如果是调用服务注册中心、配置中心接口来进行注册发现的，需要实现实例切换。

操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待升级的微服务引擎。



步骤3 在“微服务引擎信息”区域，单击“升级”。

步骤4 选择“升级后版本”，查看版本说明，根据需要决定是否升级。

步骤5 单击“确定”，进行升级。

如果升级失败，可单击“重试”，尝试再次进行升级。

----结束

10.3.10 删除微服务引擎

如不再使用微服务引擎专享版，可执行删除操作。

须知

删除引擎后数据无法恢复，请谨慎操作。

背景介绍

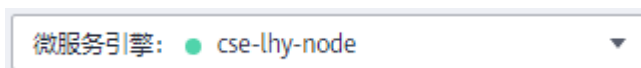
支持删除处于如下状态的微服务引擎专享版：

- 可用
- 不可用
- 创建失败
- 变更失败
- 升级失败

操作步骤

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待删除的微服务引擎。



步骤3 在“微服务引擎信息”区域，单击“删除”，在弹出的对话框中输入“DELETE”，单击“确定”。

----结束

10.3.11 管理微服务引擎安全认证

同一个微服务引擎可能会有多个用户共同使用，而不同的用户根据其责任和权限，需要具备不同的微服务引擎访问和操作权限。开启了“安全认证”的微服务引擎专享版，根据用户接入引擎使用的帐号所关联的角色，赋予该用户不同的微服务引擎访问和操作权限。

安全认证具体内容，请参考[10.4.6 系统管理](#)。

目前支持微服务使用安全认证功能的框架有Java Chassis和Spring Cloud。Java Chassis需要2.3.5及以上版本，Spring Cloud需要集成Spring Cloud Huawei 1.6.1及以上版本。

您可根据实际业务需要，对微服务引擎专享版执行开启或关闭安全认证操作：

- **开启安全认证**

未开启安全认证且状态可用的微服务引擎专享版，根据实际业务需要，可开启安全认证。

开启安全认证并同时开启了编程接口安全认证后，接入该引擎的微服务组件如果没有配置安全认证参数，或者微服务组件配置的安全认证帐号和密码不正确，会导致该微服务组件心跳失败，服务被迫下线。

- **关闭安全认证**

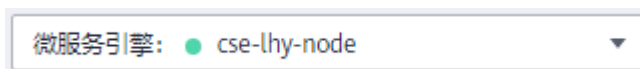
已开启安全认证且状态可用的微服务引擎专享版，根据实际业务需要，可关闭安全认证。

关闭安全认证后，无论接入该引擎的微服务组件是否配置了安全认证参数，微服务组件的正常业务功能不受影响。

开启安全认证

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“网络配置 & 安全”区域，单击“开启安全认证”。

- 如果引擎版本低于1.2.0，执行**步骤4**。
- 如果引擎版本为1.2.0及以上版本，执行**步骤5**。

步骤4 升级引擎至1.2.0或以上版本。

1. 单击“升级”。
2. 选择“升级后版本”，查看版本说明，根据需要决定是否升级到该版本后，单击“确定”。
3. 等待升级成功后，单击“返回微服务引擎”。
4. 选择刚升级成功的微服务引擎，在“网络配置 & 安全”区域，单击“开启安全认证”。

步骤5 在“微服务引擎 > 系统管理”页面开启安全认证。

- 首次开启安全认证，单击“去开启安全认证”。
需先创建root帐号。输入root帐号的“密码”和“确认密码”，单击“立即创建”。
- 再次开启安全认证，输入引擎中已关联了admin角色的帐号名称及其密码。

步骤6 （可选）参考**10.4.6.3 角色管理**，根据业务需要，创建角色。

步骤7 （可选）参考**10.4.6.2 帐号管理**，根据业务需要，创建帐号。

步骤8 单击“设置安全认证”，根据实际业务需要设置安全配置。

- 选择开启“控制台安全认证”，请执行**步骤10**。
开启控制台安全认证后，进入微服务引擎控制台界面，需要使用帐号、密码登录。登录帐号用户只能查看、配置有权限的服务。
- 选择开启“编程接口安全认证”，请执行**步骤9**。
开启编程接口安全认证，会自动同步开启“控制台安全认证”。
开启编程接口安全认证后，需要在微服务的配置文件中添加对应用户的帐号密码，否则服务无法注册到引擎。
关闭编程接口安全认证，微服务的配置文件中无需配置帐号密码即可将服务注册到引擎，效率性能更高，建议用于VPC内访问时使用。

步骤9 配置SDK，对于已部署但未配置安全认证参数的微服务组件，先为组件配置微服务安全认证的帐号名和密码，再升级组件。

步骤10 单击“确定”。

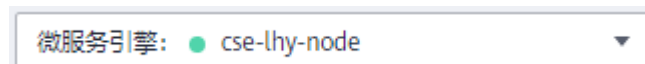
等待微服务引擎更新完成，引擎状态由“配置中”变为“可用”，开启安全认证成功。

----结束

关闭安全认证

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 在“网络配置 & 安全”区域，单击“设置安全认证”。

步骤4 在微服务引擎控制台的“系统管理”页面，单击“设置安全认证”。

步骤5 在“安全配置”页面，关闭“控制台安全认证”。

步骤6 单击“确定”，等待微服务引擎更新完成，引擎状态由“配置中”变为“可用”，关闭安全认证成功。

📖 说明

关闭安全认证后，开启安全认证时该引擎下创建的帐号不会被删除。

----结束

10.4 使用微服务引擎

10.4.1 使用微服务仪表盘

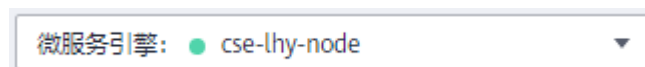
您可以通过仪表盘实时查看微服务运行相关的指标，根据丰富实时的仪表盘数据，对微服务做相应的治理动作。

背景说明

- 如果微服务应用部署在ServiceStage上，部署应用时需要设置微服务引擎，应用会自动获取服务注册发现地址、配置中心地址和仪表盘地址，不需要配置monitor地址，就可以使用仪表盘功能。
- 如果是本地启动微服务应用注册到微服务引擎，需要手工配置monitor地址，才可以使用仪表盘功能。

操作步骤

步骤1 在页面上方“微服务引擎”下拉列表，选择微服务引擎。



步骤2 单击“仪表盘”。

- 未开启安全认证的微服务引擎，请执行**步骤4**。

- 开启安全认证的微服务引擎，请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤4 在“仪表盘”页面，在下拉列表框选择需要查看的应用，在搜索框输入微服务名称，查询微服务，页面将展示筛选出的微服务的运行指标。

单击“查看示例图”，可以查看运行指标参数含义。

步骤5 选择排序方式，筛选出的微服务会按照指定方式进行排序。

----结束

10.4.2 管理微服务

您可通过微服务目录查看微服务详细信息、搜索目标微服务等，达到维护微服务的目的。“微服务目录”页面展示如下信息：

- 应用列表：显示当前微服务引擎的所有应用，支持使用应用名称搜索目标应用，也可以通过选择环境筛选应用。具体操作，请参考[查看应用列表](#)。
- 微服务列表：微服务列表下支持的具体操作请见下表。

操作名称	操作说明
查看微服务列表	显示当前微服务引擎的所有微服务，支持使用微服务名称搜索目标服务，也可以选择环境、应用筛选微服务。
查看微服务详情	在微服务详情页可以分别查看实例列表、被调用服务、调用服务、动态配置、服务契约等信息。
创建微服务	创建新的微服务。
清理无实例版本	清理实例数为0的微服务版本。
删除微服务	根据实际业务需求，删除不再使用的微服务。
动态配置	新建微服务级配置。
灰度发布	通过灰度发布功能选择少部分用户试用，待新特性成熟以后，再切换版本让所有用户使用

- 实例列表：实例列表下支持的具体操作请见下表。

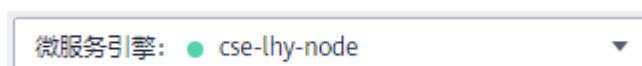
操作名称	操作说明
查看实例列表	显示当前微服务引擎的所有实例，支持使用微服务名称搜索目标实例，也可以选择环境、应用筛选实例。

操作名称	操作说明
修改实例状态	“实例状态”是微服务实例信息的一部分，用于反映微服务实例的状态。

查看应用列表

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“应用列表”，查看该引擎下当前帐号下的所有应用的详细信息。

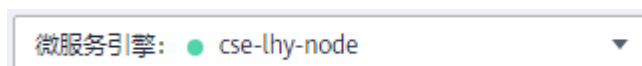
支持使用应用名称搜索目标应用，也可以选择环境筛选应用。

----结束

查看微服务列表

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“微服务列表”，查看该引擎下当前帐号下的所有微服务。

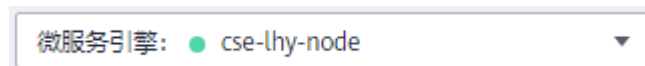
支持使用微服务名称搜索目标服务，也可以选择环境、应用筛选服务。

----结束

查看微服务详情

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

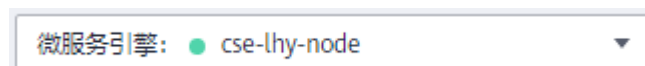
步骤5 单击“微服务列表”下待查看的微服务名称，进入微服务详情页查看实例列表、被调用服务、调用服务、配置、服务契约等信息。

----结束

创建微服务

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 选择“微服务列表 > 创建微服务”，参考下表配置微服务参数。参数前面带*号的是必须设置的参数。

参数	说明
*微服务名称	输入微服务名称，例如：myServiceName。

参数	说明
*微服务应用	输入微服务所属应用的名称，微服务之间是通过应用隔离的。
*版本	输入微服务的版本号，默认为1.0.0。 说明 微服务版本号格式为“X.Y.Z”，“X.Y.Z.B”型，其中X、Y、Z、B为数字且范围在0-32767，长度为3-46个字符。
*微服务环境	选择微服务所在环境，用于隔离微服务数据，包括：版本、实例。
描述	输入微服务描述信息。

步骤6 单击“确定”。

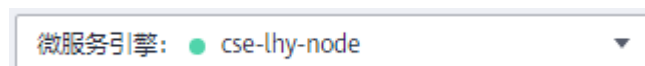
创建完成后，微服务会显示在微服务列表。

----结束

清理无实例版本

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 选择“微服务列表 > 清理无实例版本”，勾选待清理的无实例微服务版本。

支持使用微服务名称搜索目标服务，也可以选择环境、应用筛选微服务。

步骤6 单击“确定”。

----结束

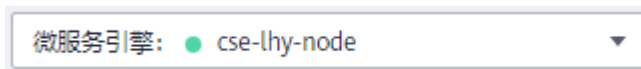
删除微服务

须知

- 微服务删除后，可通过**恢复备份数据**进行恢复。
- 如果删除的服务有实例，需要先将对应实例删除，否则服务会再次注册。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“微服务列表”，进入微服务列表：

- 批量删除微服务：勾选待删除的微服务，单击微服务列表上面的“删除”。
- 删除单个微服务：单击待删除微服务所在行“操作”列的“删除”。

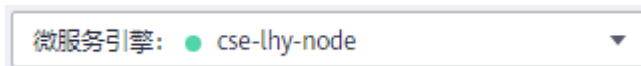
步骤6 在输入框输入“DELETE”确认删除，单击“确定”。

----结束

动态配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“微服务列表”。

步骤6 单击微服务名称。

步骤7 单击“动态配置”，进入动态配置页签。在“动态配置”页签可进行如下操作：

须知

配置项使用明文存储，请勿附带敏感数据。

操作	步骤
新建配置	操作步骤可参考 创建微服务级配置 ，其中，“配置范围”已选择“微服务级配置”，“微服务”为当前微服务。
查看历史版本	单击目标配置项对应“操作”列的“查看历史版本”，可查看历史版本配置。
禁用配置项	1. 单击目标配置项对应“操作”列的“禁用”。 2. 在弹出确认框，单击“确认”禁用该配置。
修改配置项	1. 单击目标配置项对应“操作”列的“更多 > 编辑”。 2. 在配置详情页面单击“编辑”。 3. 在“配置内容”区域输入新的配置信息。 4. 单击“保存”，保存配置修改。
删除配置项	1. 单击目标配置项对应“操作”列的“更多 > 删除”。 2. 单击“确定”，删除配置。

----结束

灰度发布

为保障新特性平稳上线，可以通过灰度发布功能选择少部分用户试用，待新特性成熟以后，再切换版本让所有用户使用。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行[步骤4](#)。
- 开启安全认证的微服务引擎，请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤4 在“微服务列表”页签，单击微服务名称，进入详情页面，单击“灰度发布”。

步骤5 单击“添加发布规则”。

- 按照权重方式添加灰度规则。
 - a. 选择“权重”，显示权重规则设置页签。

图 10-1 权重规则设置页签

创建新规则

* 发布规则

权重

自定义

* 规则名称

作用域

* 版本 0.0.1

是否添加自定义版本

规则配置

* 权重百分比

0% 100%

50

• ["0.0.1"]微服务版本将平均分配50%的流量
• 剩余50%的流量按照原有的路由策略分配到其他实例上

确定

取消

b. 对权重规则进行设置，权重规则配置项如下表所示。

配置项	说明
规则名称	自定义规则名称。
作用域	<ul style="list-style-type: none">选择规则作用的版本。勾选“是否添加自定义版本”，根据界面提示添加新版本。
规则配置	配置所选版本的引流权重，流量会按照配置的百分比分配给所选择的服务版本。

c. 单击“确定”，完成权重规则配置和灰度发布。

● 按照自定义方式添加灰度规则。

a. 选择“自定义”，显示自定义规则设置页签。

图 10-2 自定义规则设置页签

创建新规则

* 发布规则

权重
自定义

×

* 规则名称

作用域

* 版本

0.0.1

 是否添加自定义版本

规则配置

* 参数名

* 规则

=

区分大小写

确定

取消

b. 对自定义规则进行设置，自定义规则配置项如下表所示。

配置项	说明
规则名称	自定义规则名称。
作用域	<ul style="list-style-type: none"> ▪ 选择规则作用的版本。 ▪ 勾选“是否添加自定义版本”，根据界面提示添加新版本。
规则配置	<ul style="list-style-type: none"> ▪ 参数名 根据契约的Key，自定义参数名。 这个参数名key必须是契约里面存在的。有可能服务端接口写的是String paramA，加了注解实际生成的是paramB，所以这里应该是写paramB。 ▪ 规则 契约的Key对应的Value。 说明 <ul style="list-style-type: none"> ○ 当匹配符选~时，可使用*或者?字符的模糊匹配规则自定义规则值，*表示任意长度字符，?表示一个字符。比如，字段Name的规则值配置为*1000时，表示Name后4位为1000的都能匹配上。 ○ 当匹配符没有选~时，*或者?字符只是普通字符，没有模糊匹配的作用。

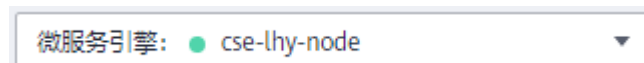
- c. 单击“确定”，完成自定义规则配置和灰度发布。

----结束

查看实例列表

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“实例列表”，进入实例列表查看引擎的所有实例。

支持使用微服务名称搜索目标实例，也可以选择环境、应用筛选实例。

----结束

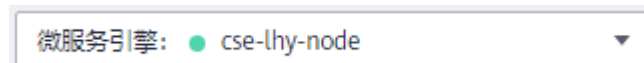
修改实例状态

“实例状态”是微服务实例信息的一部分，用于反映微服务实例的状态。微服务实例状态说明请见下表。

状态	含义
在线	实例运行中，可提供服务。
下线	实例进程结束前，标记该实例不再对外提供服务。
摘机	实例已注册到微服务引擎，暂时不对外提供服务。
拨测	实例处于内部联调状态，不对外提供服务。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的微服务引擎。



步骤3 单击“微服务目录”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击“实例列表”，选择目标实例，修改微服务实例状态。

- 下线
单击“操作”列的“下线”，修改微服务实例状态为“下线”。
- 在线
在“操作”列，选择“更多 > 在线”，修改微服务实例状态为“在线”。
- 摘机
在“操作”列，选择“更多 > 摘机”，修改微服务实例状态为“摘机”。
- 拨测
在“操作”列，选择“更多 > 拨测”，修改实例状态为“拨测”。

----结束

10.4.3 微服务治理

10.4.3.1 概述

如果使用微服务框架开发应用，在应用托管后启动应用，微服务会自动注册到对应的微服务引擎，您可以到微服务引擎控制台，参考[10.4.3.2 治理微服务](#)进行服务治理的相关操作。

说明

此功能只在微服务引擎为1.x和2.4.0及以上版本支持。

10.4.3.2 治理微服务

微服务部署完后，您可以根据微服务的运行情况进行微服务的治理。

前提条件

- 您可以先在“微服务目录 > 微服务列表”中创建微服务，启动微服务后，根据yaml文件的配置，会在对应的服务下注册服务实例。
- 如果没有事先创建微服务或者微服务已删除，在注册服务实例时会自动创建微服务。
- 微服务在创建以后，需要注册服务实例后才能进行对应操作。

治理策略说明

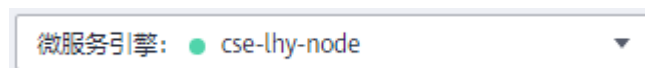
支持负载均衡、限流、容错、降级、熔断、错误注入和黑白名单等策略的配置，具体说明见下表。

名称	说明
负载均衡	<ul style="list-style-type: none"> 应用场景 微服务一般会部署多个实例，负载均衡控制微服务消费者访问微服务提供者的多个实例的策略，以达到流量均衡的目的。策略包括轮询、随机、响应时间权值、会话粘滞等。
限流	<ul style="list-style-type: none"> 应用场景 用于控制访问微服务的请求量大小，避免由于流量冲击对系统造成破坏。
降级	<ul style="list-style-type: none"> 应用场景 用于控制微服务调用其他微服务的时候，强制返回缺省值或者抛出异常，而不将请求发送到目标微服务，以达到屏蔽对目标微服务的访问和降低其压力的目的。
容错	<ul style="list-style-type: none"> 应用场景 当微服务消费者访问提供者出现异常，比如实例网络不通等，需要将请求转发到其他可用的实例。这里的容错，常被称为重试。
熔断	<ul style="list-style-type: none"> 应用场景 当微服务消费者访问提供者出现异常，比如实例网络不通、请求超时等，并且异常积累到一定的程度，需要停止访问提供者，返回一个异常或者缺省值，防止雪崩效应。 熔断提供了自动熔断策略。自动熔断需要结合错误率等判断是否熔断。
错误注入	<ul style="list-style-type: none"> 应用场景 错误注入可以模拟一个调用失败，主要用于功能验证、故障场景演示等场景。 Java Chassis开发框架接入的微服务治理。 <p>说明 该服务治理策略只适用于Java Chassis开发框架接入的微服务。</p>
黑白名单	<ul style="list-style-type: none"> 应用场景 基于公钥认证机制，微服务引擎提供了黑白名单功能。通过黑白名单，可以控制微服务允许其他哪些服务访问。 Java Chassis开发框架接入的微服务治理 只有启用了公钥认证，设置的黑白名单才能生效，请参考公钥认证。 <p>说明 该服务治理策略只适用于Java Chassis开发框架接入的微服务。</p>

设置负载均衡

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击需要治理的微服务。

步骤6 单击“负载均衡”。

步骤7 单击“新增”。先选择需要治理的微服务，再选择合理的负载均衡策略，请参考下表。

图 10-3 设置负载均衡 (Spring Cloud 开发框架接入的微服务)

限流(0) 容错(0) 熔断(0) **负载均衡(0)** 降级(0) ▾

+ 新增

选择微服务 所有微服务 ▾

负载均衡策略 随机 ▾

确定 取消

图 10-4 设置负载均衡 (Java Chassis 开发框架接入的微服务)

负载均衡(0) 限流(0) 降级(0) 容错(0) 熔断(0) ▾

+ 新增

选择微服务 所有微服务 ▾

负载均衡策略 响应时间权值 ▾

确定 取消

策略名	策略说明
轮询	支持按照服务实例的位置信息顺序路由。
随机	提供服务实例随机路由。
响应时间权重 说明 此策略适用于Java Chassis开发框架接入的微服务。	提供最小活跃数（时延）的权重路由，支持业务处理慢的服务实例接收较少的请求，防止系统停止响应。这种负载均衡策略适合请求量少且稳定的应用。
会话粘滞 说明 此策略适用于Java Chassis开发框架接入的微服务。	会话粘滞是负载均衡器上的一种机制，在设定的会话保持时间内，会保证同一用户相关联的访问请求会被分配到同一实例上。 <ul style="list-style-type: none">• 会话保持时间：会话保持的限制时间，0-86400，单位为秒。• 失效次数阈值：访问失败次数，0-10。当微服务访问下实例的失败次数或会话保持时间超过设定的值时，微服务不再访问该实例。

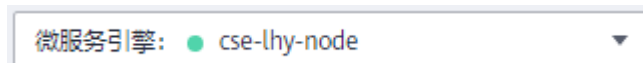
步骤8 单击“确定”保存配置。

---结束

设置限流

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击需要治理的微服务。

步骤6 单击“限流”。

步骤7 单击“新增”，限流配置项下表所示。

图 10-5 设置限流 (Spring Cloud 开发框架接入的微服务)

图 10-6 设置限流 (Java Chassis 开发框架接入的微服务)

配置项	说明	范围
限流对象 说明 此配置适用于Java Chassis开发框架接入的微服务。	访问该微服务的其他微服务。	在“限流对象”后的下拉列表可直接选择。

配置项	说明	范围
上游微服务 说明 此配置适用于 Spring Cloud 开发框架接入的微服务。	设置上游微服务调用本服务的流控制规则。	在“上游微服务”后的下拉列表可直接选择。
QPS	每秒的请求数。当限流对象对当前服务实例的每秒请求数量超过设定的值，当前服务实例不再接受该对象的请求。	1-99999的整数。

📖 说明

对于微服务拥有多个实例的情况，如果设置单个实例的流控是2700QPS，该微服务有3个实例，总的QPS就是最大8100，QPS超过8100才会触发限流。

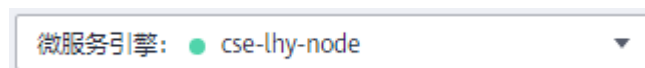
步骤8 单击“确定”保存配置。

----结束

设置降级

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击需要治理的微服务。

步骤6 单击“降级”。

步骤7 单击“新增”，选择合理的策略，降级策略配置项如下表所示。

图 10-7 设置降级 (Spring Cloud 开发框架接入的微服务)

(0) 容错(0) 熔断(0) 负载均衡(0) 降级(0) ▾

+ 新增

降级对象 所有微服务 ▾

指定请求路径

降级策略 开启 关闭

确定 取消

图 10-8 设置降级 (Java Chassis 开发框架接入的微服务)

负载均衡(0) 限流(1) 降级(0) 容错(0) 熔断(1) ▾

+ 新增


降级对象 所有微服务 ▾

所有方法 ▾

降级策略 开启 关闭

确定 取消

配置项	配置项说明
降级对象	选择需要降级的微服务。

配置项	配置项说明
指定请求路径 说明 此配置适用于Spring Cloud开发框架接入的微服务。	可单击  ，通过设置Method、Path和Headers来指定请求路径。
降级策略	<ul style="list-style-type: none">• 开启：开启降级。• 关闭：关闭降级。

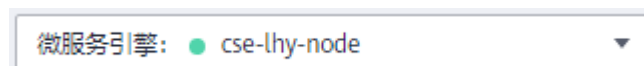
步骤8 单击“确定”，保存配置。

---结束

设置容错

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击需要治理的微服务。

步骤6 单击“容错”。

步骤7 单击“新增”，选择合理的策略，容错策略配置项如下表所示。

图 10-9 设置容错 (Spring Cloud 开发框架接入的微服务)

限流(0) 容错(0) 熔断(0) 负载均衡(0) 降级(0) ▾

+ 新增

下游微服务 ? 所有微服务 ▾

是否开启容错 开启 关闭

确定 取消

图 10-10 设置容错 (Java Chassis 开发框架接入的微服务)

负载均衡(0) 限流(0) 降级(0) 容错(0) 熔断(0) ▾

+ 新增

容错对象 所有微服务 ▾

是否开启容错 关闭 开启

确定 取消

配置项	配置项说明
下游微服务 说明 此配置适用于 Spring Cloud 开发框架接入的微服务。	设置本微服务调用下游微服务的容错规则，下拉菜单可直接选择。
容错对象 说明 此配置适用于 Java Chassis 开发框架接入的微服务。	该应用依赖的应用或方法，下拉菜单可直接选择。

配置项	配置项说明
是否开启容错	开启：向容错对象发起请求时发生错误的处理策略，开启后，会根据选择的处理策略处理请求。 关闭：关闭容错策略，即使请求失败也会等到超时时，再返回失败结果。
容错策略	当“是否开启容错”配置项设置为“开启”时需配置。 Spring Cloud开发框架接入的微服务，需设置： <ul style="list-style-type: none">● 尝试同一个微服务实例次数。● 尝试新的微服务实例次数。 Java Chassis开发框架接入的微服务，需设置： <ul style="list-style-type: none">● Failover 在不同服务器上重新尝试建立连接。● Failfast 不再重新尝试建立连接，即请求失败时会立即返回失败结果。● Failback 在同一个服务器上重新尝试建立连接。● custom<ul style="list-style-type: none">- 尝试同一个服务器次数：尝试与同一个服务器重新建立连接的次数。- 尝试新的服务器次数：尝试与新的服务器建立连接的次数。

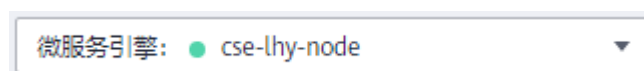
步骤8 单击“确定”，保存配置。

----结束

设置熔断

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击需要治理的微服务。

步骤6 单击“熔断”。

步骤7 单击“新增”，选择合理的策略，熔断策略配置项如下表所示。

图 10-11 设置熔断 (Spring Cloud 开发框架接入的微服务)

The screenshot shows the configuration interface for a circuit breaker in the ServiceStage console. At the top, there are tabs for different治理策略 (governance strategies): 容错(0), 熔断(0), 负载均衡(0), and 降级(0). The '熔断(0)' tab is currently selected. Below the tabs is a '+ 新增' (Add) button. The main configuration area contains the following items:

- 下游微服务 (Downstream Microservice): A dropdown menu set to '所有微服务' (All Microservices).
- 指定请求路径 (Specify Request Path): A toggle switch that is currently turned off.
- 窗口请求数 (Window Request Count): A text input field containing the value '50'.
- 失败率 (%) (Failure Rate (%)): A text input field containing the value '20'.
- 熔断时间窗(ms) (Circuit Breaker Time Window (ms)): A text input field containing the value '5000'.

At the bottom of the configuration area, there are two buttons: a red '确定' (Confirm) button and a white '取消' (Cancel) button.

图 10-12 设置熔断 (Java Chassis 开发框架接入的微服务)

容错 (0) 熔断 (0) 错误注入 (0) 黑白名单 (0) ▼

+ 新增

熔断对象 ▼
所有微服务

所有方法 ▼

* 熔断时间窗(ms): ?

* 失败率(%): ?

* 窗口请求数: ?

确定
取消

配置项	配置项说明
下游微服务 说明 此配置适用于Spring Cloud开发框架接入的微服务。	设置本微服务调用下游微服务的熔断规则。
熔断对象 说明 此配置适用于Java Chassis开发框架接入的微服务。	该应用调用的服务或方法，下拉菜单可直接选择。
指定请求路径 说明 此配置适用于Spring Cloud开发框架接入的微服务。	可单击 ，通过设置Method、Path和Headers来指定请求路径。

配置项	配置项说明
触发条件	<ul style="list-style-type: none">熔断时间窗：熔断的持续时间，该时间窗内不再响应请求。失败率：触发条件，窗口请求的失败率。窗口请求数：触发条件，窗口收到的请求数。“失败率”和“窗口请求数”的条件需同时满足才会触发熔断。

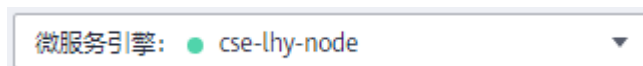
步骤8 单击“确定”，保存配置。

----结束

设置错误注入

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击需要治理的微服务。

步骤6 单击“错误注入”。

步骤7 单击“新增”，选择合理的策略，错误注入策略配置项如下表所示。

图 10-13 设置错误注入 (类型为延迟)

容错(0) 熔断(0) 错误注入(0) 黑白名单(0) ▾

+ 新增

注入对象 所有微服务 ▾

 所有方法 ▾

类型 延迟 错误

协议 Rest Highway

* 触发概率(%) 50

* 延迟时间(ms) 10

确定 取消

图 10-14 设置错误注入 (类型为错误)

容错(0) 熔断(0) 错误注入(0) 黑白名单(0) ▾

+ 新增

注入对象

类型 延迟 错误

协议 Rest Highway

* 触发概率(%)

* http错误码

配置项	配置项说明
注入对象	需要测试容错能力的微服务，可以具体到微服务的方法。
类型	测试微服务的容错能力类型： <ul style="list-style-type: none">● 延迟● 错误
协议	访问微服务出现延时或错误的协议： <ul style="list-style-type: none">● Rest● Highway
触发概率	访问微服务出现延时或错误的概率。
延迟时间	访问微服务出现延时的时长，“类型”选择为“延迟”时才需要配置。
http错误码	访问微服务出现错误的HTTP错误码，“类型”选择为“错误”时才需要配置。该错误码为HTTP标准的错误码。

步骤8 单击“确定”，保存配置。

---结束

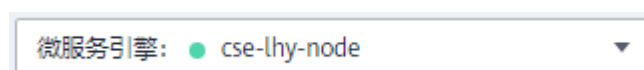
设置黑白名单

基于公钥认证机制，微服务引擎提供了黑白名单功能。通过黑白名单，可以控制微服务允许其他哪些服务访问。

只有启用了公钥认证，设置的黑白名单才能生效，请参考[公钥认证](#)。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待查看的微服务引擎。



步骤3 选择“微服务治理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击需要治理的微服务。

步骤6 单击“黑白名单”。

步骤7 单击“新增”，为应用添加黑白名单，黑白名单配置项如下表所示。

图 10-15 设置黑白名单



配置项	配置项说明
类型	<ul style="list-style-type: none">黑名单：表示根据匹配规则匹配到的微服务都不允许访问当前服务。白名单：表示根据匹配规则匹配到的微服务允许访问当前服务。
匹配规则	使用正则表达式表示。 例如“匹配规则”设置为data*，指在黑名单下匹配到的名称以data开头的服务不允许访问当前服务，或者在白名单下匹配到的名称以data开头的服务允许访问当前服务。

步骤8 单击“确定”，保存配置。

---结束

公钥认证

公钥认证是微服务引擎提供的一种简单高效的微服务之间认证机制，它的安全性建立在微服务与服务中心之间的交互是可信的基础之上，即微服务和服务中心之间必须先启用认证机制。它的基本流程如下：

1. 微服务启动的时候，生成密钥对，并将公钥注册到服务中心。
2. 消费者访问提供者之前，使用自己的私钥对消息进行签名。
3. 提供者从服务中心获取消费者公钥，对签名的消息进行校验。

启用公钥认证步骤如下：

1. 公钥认证需要在消费者、提供者都启用。

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

2. 在pom.xml中增加依赖：

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

10.4.4 配置管理（适用于微服务引擎版本为 2.x）

微服务引擎定义了一套与开发框架无关的配置机制。配置项由key、label和value组成，通过label区分配置项属于全局配置、微服务配置等作用范围，label也可以表示value的类型。

您可以根据需要参考下表选择需要执行的操作。

操作名称	操作说明
创建应用级配置	将新建的配置关联到某一应用，并添加应用名称和所在环境标签。

操作名称	操作说明
创建微服务级配置	将新建的配置关联到某一微服务，并添加微服务名称、应用名称和环境。
创建自定义配置	当应用级配置、微服务级配置不能满足使用需求时，可自定义配置文件。
导入配置	导入本地配置文件。
导出配置	导出选定的配置文件到本地。
对比配置版本	提供了不同历史版本之间差异比对。
回滚版本	从当前版本回滚到选择的历史版本。
查看历史版本	可查看不同历史版本的配置。
编辑配置项	可通过需要编辑配置项配置内容。
禁用配置项	禁用某配置项。
删除配置项	删除某一配置项

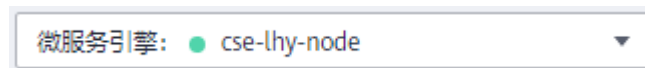
📖 说明

当引擎规格限定的配置条目配额即将用完时，为了保证更大程度可用，引擎允许超出剩余配额的新建配置项操作同时成功。请尽快扩容引擎，避免后续新建配置失败。

创建应用级配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置应用的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击“新建配置”，参考下表设置配置参数。参数前面带*号的是必须设置的参数。

参数名称	参数说明
*配置项	输入配置项。 配置项为该配置的全局ID，在编码阶段通过配置项对配置进行索引及操作。建议采用类Java包命名方式（如cse.service.registry.address）的命名规则来保证配置项的可读性和唯一性。
配置范围	选择“应用级配置”。
*应用	1. 选择或输入应用名称。 2. 选择环境。
配置格式	选择配置格式。
*配置内容	输入配置内容。
是否启用	选择是否启用配置： <ul style="list-style-type: none">● 现在启用：创建成功后，配置立即生效。● 暂不启用：创建成功后，配置暂不生效。

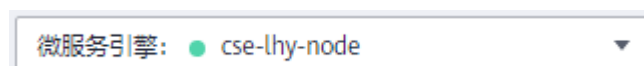
步骤6 单击“立即创建”，启用该配置项。

----结束

创建微服务级配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置微服务的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“新建配置”，参考下表设置配置参数。参数前面带*号的是必须设置的参数。

参数名称	参数说明
*配置项	输入配置项。 配置项为该配置的全局ID，在编码阶段通过配置项对配置进行索引及操作。建议采用类Java包命名方式（如cse.service.registry.address）的命名规则来保证配置项的可读性和唯一性。
配置范围	选择“微服务级配置”。
*微服务	1. 选择或者输入微服务名称。 2. 选择或输入应用名称。 3. 选择环境。
配置格式	选择配置格式。
*配置内容	输入配置内容。
是否启用	选择是否启用配置： <ul style="list-style-type: none">● 现在启用：创建成功后，配置立即生效。● 暂不启用：创建成功后，配置暂不生效。

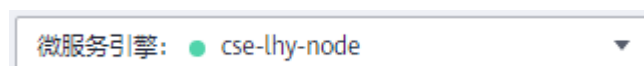
步骤6 单击“立即创建”，启用该配置项。

---结束

创建自定义配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待创建自定义配置的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击“新建配置”，参考下表设置配置参数。参数前面带*号的是必须设置的参数。

参数名称	参数说明
*配置项	输入配置项。 配置项为该配置的全局ID，在编码阶段通过配置项对配置进行索引及操作。建议采用类Java包命名方式（如cse.service.registry.address）的命名规则来保证配置项的可读性和唯一性。
配置范围	选择“自定义配置”。
标签	当应用级配置、微服务级配置不能满足使用需求，可通过标签创建自定义配置。
配置格式	选择配置格式。
*配置内容	输入配置内容。
是否启用	选择是否启用配置： <ul style="list-style-type: none">● 现在启用：创建成功后，配置立即生效。● 暂不启用：创建成功后，配置暂不生效。

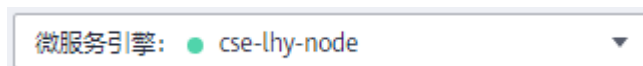
步骤6 单击“立即创建”，启用该配置项。

----结束

导入配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待导入配置的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击右上角的“导入”，参考下表设置导入参数。

参数名称	参数说明
导入至特定环境	<ul style="list-style-type: none">● 关闭：导入的配置不改变环境标签。● 开启：把配置导入至特定环境，将会改变环境标签。

参数名称	参数说明
相同配置	<ul style="list-style-type: none">● 终止导入：导入过程中，如果遇到和系统中相同的配置，导入终止。● 跳过：导入过程中，如果遇到和系统中相同的配置，该配置会被跳过，继续导入其余配置。● 覆盖：导入过程中，如果遇到和系统中相同的配置，该配置的值会被替换。
配置文件	单击“导入文件”，选择待导入配置文件。 说明 待导入的配置文件大小不超过2MB。

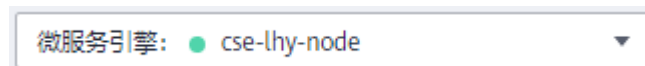
步骤6 单击“关闭”，完成导入。

----结束

导出配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待导出配置的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

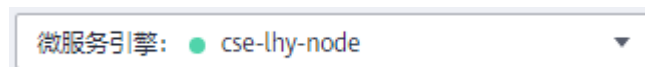
步骤5 勾选待导出的配置项，单击“导出”。

----结束

对比配置版本

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待对比配置版本的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。

- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击待对比的配置项名称。

步骤6 单击“历史版本”。

步骤7 在左侧“历史版本”列表选择待查看的历史版本。

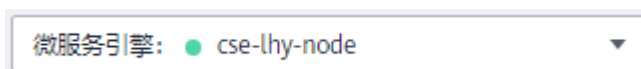
在右侧“配置文件”列，可查看历史版本和当前版本的差异。

----结束

回滚版本

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待回滚版本的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击待操作的配置项名称。

步骤6 单击“历史版本”。

步骤7 在左侧“历史版本”列表，选择待回滚到的历史版本。

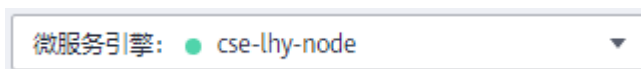
步骤8 在右侧“配置文件”列，单击“回滚到此版本”。

----结束

查看历史版本

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待回滚版本的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

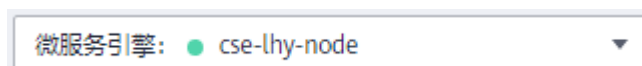
步骤5 单击某配置项右侧“操作”列的“查看历史版本”，进入“历史版本”页面可查看该配置项的历史版本，在该页面也可对比配置版本和回滚版本操作。

----结束

编辑配置项

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待回滚版本的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击待编辑的配置项右侧“操作”列的“编辑”，也可单击待编辑的配置项名称，在配置详情页面，单击“编辑”。

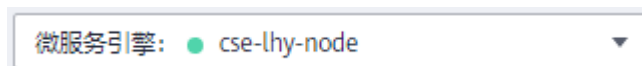
步骤6 在“配置内容”输入框输入配置信息，单击“保存”编辑配置项完成。

----结束

禁用配置项

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待回滚版本的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在待禁用的配置项右侧“操作”列选择“更多 > 禁用”。

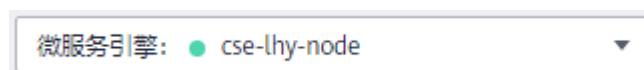
步骤6 在弹出框中单击“确认”禁用该配置项。

----结束

删除配置项

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待回滚版本的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在待删除的配置项右侧“操作”列选择“更多 > 删除”，也可单击待删除的配置项名称，在配置详情页面，单击“删除”。

步骤6 在弹出框中单击“确认”删除该配置项。

----结束

10.4.5 配置管理（适用于微服务引擎版本为 1.x）

在此处添加配置为全局配置，当添加配置后，所有注册到这个引擎的微服务如果使用了该配置项，会立刻生效。

如果针对单个微服务设置了动态配置，则动态配置会覆盖掉全局配置，动态配置的设置请参考[动态配置](#)。

创建配置

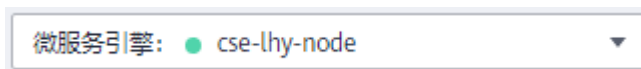
配置管理提供微服务间的公共配置，如日志级别、运行参数等。配置添加以后，如果微服务没有自定义相同的配置项，会将该配置作为默认配置使用。

须知

配置项使用明文存储，请勿附带敏感数据。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置应用的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击“创建配置”。

步骤6 在“创建配置”页面，选择微服务环境，并输入“配置项”和“值”。

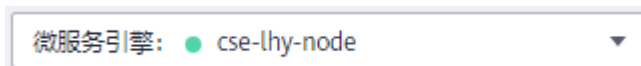
步骤7 单击“确定”完成配置添加。

---结束

导入配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置应用的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行[步骤5](#)。
- 开启安全认证的微服务引擎，请执行[步骤4](#)。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 单击“导入”。

步骤6 选择微服务环境，单击“导入文件”，选择待导入配置文件。

📖 说明

每次导入的配置项条目数量最多为150条。

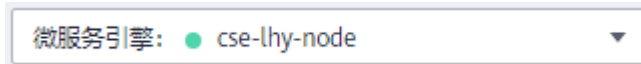
步骤7 单击“关闭”完成配置文件导入。

---结束

导出配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置应用的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

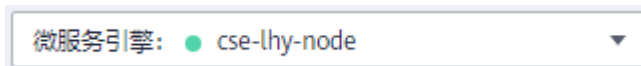
步骤5 单击“全部导出”，配置文件导出成功。

----结束

删除配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置应用的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 选中待操作的配置项，单击“删除”，也可单击待操作配置项“操作”列的“删除”。

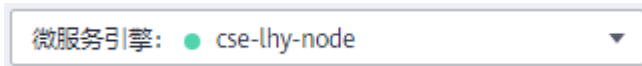
步骤6 在弹出框单击“确定”，删除配置项。

----结束

编辑配置

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择已部署了待配置应用的微服务引擎。



步骤3 单击“配置管理”。

- 未开启安全认证的微服务引擎，请执行**步骤5**。
- 开启安全认证的微服务引擎，请执行**步骤4**。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及**创建微服务引擎**时输入的密码。
- 创建帐号请参考**新增帐号**。

步骤5 单击待操作配置项“操作”列的“编辑”，编辑配置项的值。

步骤6 单击“确定”，配置项修改。

----结束

10.4.6 系统管理

10.4.6.1 系统管理概述

同一个微服务引擎可能会有多个用户共同使用，而不同的用户根据其责任和权限，需要具备不同的微服务引擎访问和操作权限。

开启了“安全认证”的微服务引擎专享版，通过微服务控制台提供了基于RBAC（Role-Based Access Control，基于角色的访问控制）的系统管理功能。

开启了“安全认证”的微服务引擎专享版，支持Spring Cloud、Java Chassis微服务框架正常接入。

说明

- 基于RBAC的系统管理功能与IAM权限管理无关，仅是CSE内部的权限管理机制。
 - 如果您通过微服务引擎控制台操作微服务引擎，必须同时具备IAM和RBAC的操作权限，且IAM权限优先级要高于RBAC权限。
 - 如果您通过API接口或者微服务框架操作微服务引擎，则只需具备RBAC相关权限。
1. 您可以使用关联了admin角色权限的帐号创建新帐号，根据实际业务需求把合适的角色同帐号关联。使用该帐号的用户则具有对该微服务引擎的相应的访问和操作权限。
 - 创建开启了“安全认证”的微服务引擎专享版时，系统自动创建1个关联了admin角色权限的root帐号。不能编辑、删除root帐号。
 - 您可以使用创建该微服务引擎的root帐号或者该微服务引擎下关联了admin角色权限的帐号创建新帐号。创建和管理帐号，请参考**10.4.6.2 帐号管理**。
 2. 您可以使用关联了admin角色权限的帐号创建自定义角色，根据业务需求把合适的微服务引擎访问和操作权限赋予该角色。
 - 系统默认内置两种角色：管理员（admin）、开发者（developer）。不能编辑、删除内置角色。

- 您可以使用创建该微服务引擎的root帐号或者该微服务引擎下关联了admin角色权限的帐号创建自定义角色。创建和管理角色，请参考[10.4.6.3 角色管理](#)。
- 角色权限说明，请参见[表10-5](#)。

表 10-5 角色权限说明

角色	权限说明
admin	具有该微服务引擎下所有微服务、帐号和角色的所有操作权限。
developer	具有该微服务引擎下所有微服务的所有操作权限。
自定义角色	根据实际业务需求创建角色，给角色分配相应微服务的操作权限。

10.4.6.2 帐号管理

您可以使用微服务引擎下关联了admin角色权限的帐号登录该微服务引擎控制台，根据实际业务需求创建新帐号或对该引擎下已创建的指定帐号进行管理操作。

表 10-6 帐号管理操作说明

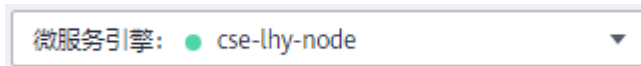
操作	说明
新增帐号	根据实际业务需求创建新帐号，把合适的角色同帐号关联，使用该帐号的用户则具有对该微服务引擎的相应的访问和操作权限。 最多可创建1000个帐号。
查看帐号角色权限	查看指定帐号关联的角色权限配置。
编辑帐号	根据实际业务需求，对帐号进行增加、删除角色操作。不能编辑root帐号。
修改密码	根据业务需求或者安全规定，可修改已登录微服务引擎的帐号密码。 须知 <ul style="list-style-type: none"> • 若在SDK中使用了该帐号密码注册微服务，修改帐号密码可能会影响使用该微服务业务运行（无法注册到微服务引擎），将导致业务系统受损，请谨慎操作。 • 密码修改后，需及时更新微服务认证配置。 • 密码修改后，可能会因密码错误超过3次导致帐号锁定，需要等待15分钟锁定状态才会释放。
删除帐号	根据实际业务需求，删除不再使用的帐号。不能删除root帐号。 须知 若在SDK中使用了该帐号密码注册服务，删除帐号会影响该服务业务运行（无法注册到引擎），将导致业务系统受损，请谨慎操作。

新增帐号

创建新帐号前，可先根据实际业务需要[创建角色](#)。

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“帐号管理”页签，单击“新增帐号”，参考下表设置帐号参数。

参数名称	参数说明
帐号名称	输入新帐号名称。 说明 帐号创建后，帐号名称不可修改。
帐号角色	根据实际业务需求，选择帐号角色。 说明 一个帐号下最多可关联5个角色。
密码	输入帐号密码。
确认密码	再次输入帐号密码。

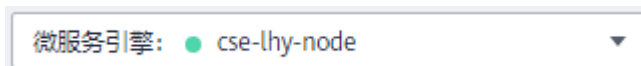
步骤6 单击“确定”，完成新帐号创建。

----结束

查看帐号角色权限

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

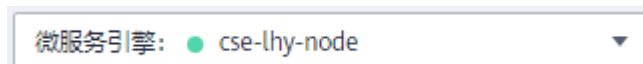
步骤5 单击帐号列表中待查看帐号“角色”列的角色名称，在弹出页面可以查看帐号关联的角色名称及权限配置情况。

----结束

编辑帐号

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“帐号管理”页签，单击待编辑帐号“操作”列的“编辑帐号”。

步骤6 根据实际业务需求，选择“帐号角色”。

说明

一个帐号下最多可关联5个角色。

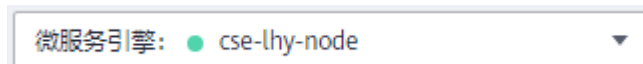
步骤7 单击“保存”，完成帐号编辑。

----结束

修改密码

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入帐号名及其密码，单击“确定”。

说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 连接微服务引擎的帐号未关联admin角色权限，仅可修改当前登录帐号的密码。
- 连接微服务引擎的帐号关联了admin角色权限，可修改该微服务引擎下所有帐号的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“帐号管理”页签，选择登录该微服务引擎的帐号名，单击“操作”列的“修改密码”。

1. 输入“原密码”、“新密码”和“确认密码”。
2. 查看提示信息确认需要修改密码后，勾选“我已确认知晓”。

📖 说明

也可单击“系统管理”页面右上角的“修改密码”，修改当前登录帐号的密码。

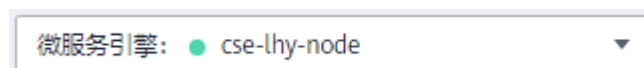
步骤6 单击“保存”，完成密码修改。

----结束

删除帐号

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“帐号管理”页签，单击待删除帐号“操作”列的“删除”。

步骤6 单击“确认”。

----结束

10.4.6.3 角色管理

除了系统中默认提供的两种角色：管理员（admin）和开发者（developer）无法进行操作外，您可以使用该微服务引擎下关联了admin角色权限的“帐号”登录微服务引擎控制台，根据实际的业务需求对角色进行如[表10-7](#)所示的操作。

表 10-7 角色管理操作说明

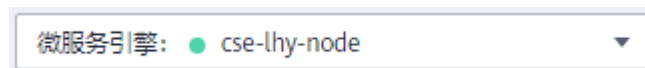
操作	说明
创建角色	根据实际业务需求创建新角色，设置该角色在不同服务组对应的权限动作。 最多可创建100个角色。
编辑角色	根据实际业务需求，修改已创建角色的权限配置。

操作	说明
删除角色	根据实际业务需求，删除不再使用的角色。 说明 <ul style="list-style-type: none">角色删除后无法恢复，请谨慎操作。删除角色前要先确认该角色没有被帐号关联。取消角色同帐号之间的关联，请参考编辑帐号。
查看角色	可按照角色名称的关键字查看该微服务引擎下已创建的角色。

创建角色

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“角色管理”页签，单击“创建角色”。

步骤6 输入新角色名称。



📖 说明

角色创建后，角色名称不可修改。

步骤7 对权限进行配置。

- 设置“服务组”。
 - 选择“全部服务”。
可以对该微服务引擎的所有微服务资源实施相应的权限动作。
 - 选择“自定义服务组”，可按照[表10-8](#)进行设置。

表 10-8 自定义服务组操作

操作名称	操作说明
新增匹配规则	<p>单击“添加服务组匹配规则”，根据实际业务需要，选择“应用”、“环境”、“服务”三个参数值来匹配规则过滤该角色可以实施权限动作的微服务。</p> <p>说明 应用名称、环境名称和服务名称是微服务的三个参数：</p> <ul style="list-style-type: none"> 如果单条匹配规则只设置了一个参数，则角色对与该参数值相匹配的微服务有操作权限。 例如添加匹配规则“环境:production”，那么该角色只对环境名称是“production”的微服务有操作权限。 如果单条匹配规则设置了多于一个参数，则角色对与所有参数值都相匹配的微服务有操作权限。 例如添加匹配规则“环境:production”“应用:abc”，那么该角色对环境名称是“production”且应用名称是“abc”的微服务有操作权限。 在启用自动发现的情况下，微服务会通过注册中心查询注册中心、配置中心、仪表盘等服务的实例地址。对微服务授予查询权限时，需要包含应用default的权限，添加匹配规则“应用:default”。 <p>设置微服务匹配规则后，单击“确定”。</p>
编辑匹配规则	<p>单击待编辑的匹配规则后的“”，即可根据实际业务需要，重新设置该条匹配规则的“服务组”和“权限动作”。重新设置服务组匹配规则后，单击“确定”。</p>
删除匹配规则	<p>单击待删除的匹配规则后的“”，即可根据实际业务需要，删除该条服务组匹配规则。</p>

📖 说明

自定义服务组最多可以同时设置20条微服务匹配规则。

当自定义服务组设置了多条匹配规则时，只要微服务满足其中任意一条匹配规则，角色就对该微服务有操作权限。

2. 设置“权限动作”。

按照实际的业务需求，设置角色对选择的服务组可实施的权限动作，可以勾选多个。

- 全选：对该服务组资源可实施增加、删除、修改和查询四种动作。
- 增加：对该服务组资源可实施增加的动作。
- 删除：对该服务组资源可实施删除的动作。

📖 说明

如果只勾选了删除，将无法通过界面对服务组资源进行删除，必须同时勾选查询。

- 修改：对该服务组资源可实施修改的动作。

📖 说明

如果只勾选了修改，将无法通过界面对服务组资源进行修改操作，必须同时勾选查询。

- 查询：对该服务组资源可实施查询的动作。

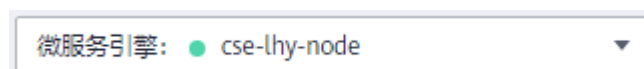
步骤8 单击“创建”，创建角色。

----结束

编辑角色

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“角色管理”页签，单击待编辑角色“操作”列的“编辑”。

步骤6 根据实际业务需求，修改“服务组”和“权限动作”。

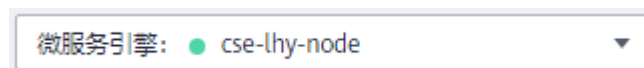
步骤7 单击“保存”，完成角色编辑。

----结束

删除角色

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“角色管理”页签，单击待删除角色“操作”列的“删除”。在弹出的对话框中输入“DELETE”，单击“确定”。

📖 说明

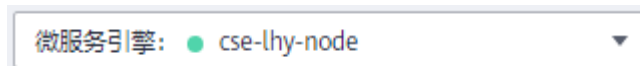
- 角色删除后无法恢复，请谨慎操作。
- 删除角色前要先确认该角色没有被帐号关联。取消角色同帐号之间的关联，请参考[编辑帐号](#)。

----结束

查看角色

步骤1 登录ServiceStage控制台，选择“微服务引擎 > 引擎实例”。

步骤2 在页面上方“微服务引擎”下拉列表，选择待操作的开启了安全认证的微服务引擎。



步骤3 单击“系统管理”。

步骤4 在弹出的“安全认证”对话框输入该微服务引擎下关联了admin角色权限的帐号名及其密码，单击“确定”。

📖 说明

- 首次连接微服务引擎，请输入root帐号名及[创建微服务引擎](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤5 在“角色管理”页签，单击待查看角色名称前的▼，展开待查看角色的详细信息。展开后可以查看角色的“服务组”和“权限动作”。

----结束

11 云审计服务支持的关键操作

11.1 云审计服务支持的 ServiceStage 操作列表

ServiceStage通过云审计服务 (Cloud Trace Service, CTS) 为您提供ServiceStage应用管理操作的操作记录, 供您查询、审计和回溯使用。

[开通云审计服务](#)后, 系统开始记录ServiceStage资源的操作, CTS控制台可以保存最近7天的操作记录。

表 11-1 云审计服务支持的 ServiceStage 操作

操作名称	资源类型	事件名称
创建组件	component	createComponent
删除组件	component	deleteComponent
升级组件	component	updateComponent
启动组件	component	startComponent
停止组件	component	stopComponent
重启组件	component	restartComponent
伸缩组件	component	scaleComponent
回滚组件	component	rollbackComponent
部署组件	component	provisionComponent
卸载组件	component	deprovisionComponent
创建应用	application	createApplication
删除应用	application	deleteApplication
更新应用	application	updateApplication
创建环境	environment	createEnvironment

操作名称	资源类型	事件名称
删除环境	environment	deleteEnvironment

11.2 查看云审计日志

查看ServiceStage云审计日志，请参考[查看审计事件](#)。

12 查看监控指标及告警

概述

应用运维管理服务 (AOM) 可以监控和查看ServiceStage服务的运行状态、各个指标的使用情况，并对监控项创建告警规则。

当您使用ServiceStage服务部署组件后，AOM服务能关联通过在ServiceStage部署组件的监控指标，帮助您实时掌握组件的各项性能指标，精确掌握组件运行情况。

设置监控及告警

CCE会配合AOM对集群进行全方位的监控，在创建节点时会默认安装AOM的ICAgent（在集群kube-system命名空间下名为icagent的DaemonSet），ICAgent默认采集集群底层资源以及运行在集群上负载的监控数据并上传到AOM。另外，[自定义组件运行指标](#)后，ICAgent还能采集负载的自定义指标监控数据并上传到AOM。

参考[7.13.4 设置资源监控告警阈值](#)，完成阈值告警规则设置后，组件运行过程中产生的各种告警会上传到AOM。

支持的监控指标

指标是对资源性能的数据描述或状态描述。

容器部署组件的资源基础监控包含CPU、内存、磁盘等，具体请参考[表12-1](#)。

- **表 12-1 资源监控指标**

监控指标	指标含义	取值范围	单位
CPU内核总量 (cpuCoreLimit)	该指标用于统计测量对象申请的CPU核总量。	≥ 1	核 (Core)
CPU内核占用 (cpuCoreUsed)	该指标用于统计测量对象已经使用的CPU核个数。	≥ 0	核 (Core)
CPU使用率 (cpuUsage)	该指标用于统计测量对象的CPU使用率。服务实际使用的与申请的CPU核数量比率。	0 ~ 100%	百分比 (Percent)

监控指标	指标含义	取值范围	单位
物理内存总量 (memCapacity)	该指标用于统计测量对象申请的物理内存总量。	≥0	兆字节 (Megabytes)
物理内存使用率 (memUsage)	该指标用于统计测量对象已使用内存占申请物理内存总量的百分比。	0~100%	百分比 (Percent)
物理内存使用量 (memUsed)	该指标用于统计测量对象实际已经使用的物理内存 (Resident Set Size)。	≥0	兆字节 (Megabytes)
磁盘读取速率 (diskReadRate)	该指标用于统计每秒从磁盘读出的数据量。	≥0	千字节/秒 (Kilobytes/Second)
磁盘写入速率 (diskWriteRate)	该指标用于统计每秒写入磁盘的数据量。	≥0	千字节/秒 (Kilobytes/Second)
下行 Pps(recvPackRate)	每秒网卡接收的数据包个数。	≥0	个/秒 (Packets/Second)
文件系统容量 (filesystemCapacity)	该指标用于统计测量对象文件系统的容量。仅支持1.11及其更高版本的kubernetes集群中驱动模式为devicemapper的容器。	≥0	兆字节 (Megabytes)
下行 Bps(recvBytesRate)	该指标用于统计测试对象的入方向网络流速。	≥0	字节/秒 (Bytes/Second)
下行包错率 (recvErrPackRate)	每秒网卡接收的错误包个数。	≥0	个/秒 (Packets/Second)
上行 Pps(sendPackRate)	该指标用于统计测试对象的出方向网络流速。	≥0	字节/秒 (Bytes/Second)
上行包错率 (sendErrPackRate)	每秒网卡发送的错误包个数。	≥0	个/秒 (Packets/Second)
上行 Bps(sendBytesRate)	该指标用于统计测试对象的出方向网络流速。	≥0	字节/秒 (Bytes/Second)
容器错包个数 (rxPackErrors)	该指标用于统计测量对象收到错误包的数量。	≥0	个 (Packets)

监控指标	指标含义	取值范围	单位
线程数 (threadsCount)	该指标用于统计主机中当前创建的线程数量。	≥0	无
文件系统可用 (filesystemAvailable)	该指标用于统计测量对象文件系统的可用大小。仅支持1.11及其更高版本的Kubernetes集群中驱动模式为devicemapper的容器。	≥0	兆字节 (Megabytes)
文件系统使用率 (filesystemUsage)	该指标用于统计测量对象文件系统使用率。实际使用量与文件系统容量的百分比。仅支持1.11及其更高版本的Kubernetes集群中驱动模式为devicemapper的容器。	≥0	百分比 (Percent)
句柄数 (handleCount)	该指标用于统计测量对象使用的句柄数。	≥0	无
组件状态 (status)	该指标用于统计应用组状态是否正常。	<ul style="list-style-type: none"> ● 0: 表示正常 ● 1: 表示异常 	无
虚拟内存总量 (virMemCapacity)	该指标用于统计测量对象申请的虚拟内存总量。	≥0	兆字节 (Megabytes)

13 常见问题

13.1 应用开发问题咨询指引

开源社区链接

servicecomb-java-chassis、spring-cloud-huawei等SDK作为开源框架，版本的发布和维护都由开源团队负责，目前已经在开源社区积累了诸多问题的解决经验，若在开发使用过程中遇到问题，请通过如下链接检索历史问题、提交新问题和进行问题咨询：

- servicecomb-java-chassis开源框架
开源社区：<https://github.com/apache/servicecomb-java-chassis/issues>
开发者指导：https://servicecomb.apache.org/references/java-chassis/zh_CN/
- spring-cloud-huawei开源框架
开源社区：<https://github.com/huaweicloud/spring-cloud-huawei/issues>
参考文档：

📖 说明

- 搜索小技巧：
使用站内搜索，提升准确性。比如服务报错：Load balancer does not have available server for client: default，可以使用如下搜索方式：
Load balancer site:bbs.huaweicloud.com或者Load balancer site:github.com/apache/servicecomb-java-chassis/issues
- servicecomb-java-chassis, spring-cloud-huawei均为开源软件，ServiceStage团队对其进行插件扩展以接入云服务，如果遇到框架使用问题而非插件问题，请到对应开源社区提问。

关键信息

为了方便问题的快速定位，提issue时务必提供详细的关键信息，并提供可以复现问题的Demo。

以servicecomb-java-chassis为例，请提供如下关键信息：

1. 框架相关日志：默认框架日志会与业务日志一起打印，并且会在根目录下生成cse.log。若业务侧使用了log4j2或者logback等日志框架，需根据自定义的日志策略查找关键信息。

a. 服务启动类问题关键信息：

表 13-1 服务启动类问题关键信息

关键词	描述
choose org.apache.servicecomb.omb	servicecomb-java-chassis支持两种rest通信通道，需要根据日志判断使用的通信通道。 choose org.apache.servicecomb.transport.rest.vertx.VertxRestTransport框架默认使用Rest over Vertx通信通道，即使用vertx作为http服务器。 choose org.apache.servicecomb.transport.rest.servlet.ServletRestTransport同时支持Rest over Servlet通信通道，即使用其他HTTP服务器，比如tomcat。
endpoint to publish	服务发布地址。
Register microservice instance success	服务实例注册成功的标志。

b. 服务调用类问题关键信息：

表 13-2 服务调用类问题关键信息

关键词	描述
find instances	消费端（发起调用的服务）在调用服务端（被调用的服务）之前，会先从微服务引擎的服务中心查询服务端的实例
accesslog	accesslog会记录调用该服务的请求源，API，状态码等，默认情况下不会开启该功能

accesslog的打印受通信通道和日志框架影响。若使用Rest over Vertx通信通道，accesslog是由Vertx记录。

推荐accesslog日志格式：

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

默认情况下会在根目录生成access.log，若业务侧使用了log4j2或者logback等日志框架。

若使用Rest over Servlet通信通道，accesslog是由使用的HTTP服务器记录，需自行搜索资料开启。

例如springboot内置的tomcat，可以通过如下配置开启。

```
server:
  tomcat:
    accesslog:
      enabled: true
      pattern: '%h %l %u %t "%r" %s %b %D'
      directory: accesslogs
```

```
buffered: false  
basedir: ./logs
```

2. 微服务引擎的版本和SDK的版本。微服务引擎版本可单击引擎名称查看。SDK版本可搜索groupId为org.apache.servicetomb的依赖。

13.2 环境管理

13.2.1 微服务和其他平台服务有什么不同？

微服务是构建应用系统的架构模式，平台服务是云平台提供给平台用户使用的中间件服务。

在使用上，平台服务需要进行订购流程进行服务开通。对于微服务，需自己开发并使用平台提供的服务发现能力进行服务发现。

13.3 应用管理

13.3.1 如何查看应用组件部署失败的原因？

问题描述

应用组件部署完成后，状态显示为“未就绪”，表示应用组件部署失败。

解决方法

步骤1 登录ServiceStage控制台。

步骤2 选择以下任意方式进入组件“实例列表”页面：

- 在“应用管理”页面，单击组件所属应用名称，在“组件列表”单击待操作组件名称，在左侧导航栏单击“实例列表”。
- 在“组件管理”页面，单击待操作组件名称，在左侧导航栏单击“实例列表”。

步骤3 单击实例列表待操作实例名称前的 ∇ 。

步骤4 选择“事件”页签，在事件列表中，查看事件描述信息，判断应用组件部署失败的原因。

----结束

13.3.2 实例长期处于创建中怎么办？

应用组件部署以后，如果服务实例状态长期处于“未就绪”状态时，可以进入服务实例列表，展开实例详细信息，在“事件”页签查看详细信息，显示内存不足，如下图所示。

实例名称	状态	CPU(申请限制)	内存(申请限制)	所在节点	运行时长	地址	创建时间	操作
controller-14fb06-c9b57b546-kkxsp	创建中	0.8Core / 2Core	0.78GiB / 4GiB		11分钟		2019/08/14 17:50:31 GMT+08:00	删除
controller-14fb06-c9b57b546-5vj5m	创建中	0.8Core / 2Core	0.78GiB / 4GiB		11分钟		2019/08/14 17:50:31 GMT+08:00	删除

监控 事件 容器

请输入事件关键词

事件名称	发生次数	事件类型	首次产生时间	描述
实例创建失败	2	告警	2019/08/14 17:50:31 GMT+08:00	0/1 nodes are available: 1 Insufficient cpu, 1 Insufficient memory.

可以通过新增节点解决该问题，操作方法请参考[创建节点](#)。

13.3.3 如何解决 Docker 运行 node 应用程序时的依赖问题？

问题描述

在微服务docker里面运行一个node程序，这个程序依赖一个node-gyp，怎么在程序运行之前安装好这些依赖。

解决办法

可以定制自己的Dockerfile，在Dockerfile里面添加node-gyp依赖。

13.3.4 定制 Tomcat Context path

在创建和部署Tomcat应用时需要设置Tomcat配置，使用默认server.xml配置，上下文路径是"/"，没有指定应用路径。

- 选择开启“公网访问”，应用访问地址为：`http://{应用公网域名}:{应用访问端口号}`。例如，`http://example_domain.com:30317`。
- 未开启“公网访问”，应用访问地址为：`http://{VPC内网访问地址}:{应用访问端口号}`。例如，`http://192.168.0.168:30317`。

在部署组件的组件配置过程中，可在设置“Tomcat 配置”时根据具体业务实际自定义应用路径：

1. 勾选“配置参数”。
2. 单击“使用示例模板”，根据业务要求编辑模板文件。
3. 参考如下示例修改Context path的内容，例如修改为“app-path”，自定义应用路径。则应用访问地址被修改为`http://example_domain.com:30317/app-path`或者`http://192.168.0.168:30317/app-path`。

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

13.3.5 如何固定应用组件 IP？

问题描述

在部署应用组件的过程中，如果不设置“TCP/UDP路由配置”，那么当容器重启时，应用的访问IP会发生变化。这种情况会为您的某些配置造成困扰。

解决方法

创建部署应用组件时或者部署应用组件后设置“TCP/UDP路由配置”即可。以下三种方式均可解决该问题：

- 集群内访问：应用暴露给同一集群内其他应用访问的方式，可以通过集群内部域名访问。
- VPC内网访问：应用可以让同一VPC内其他应用程序访问，通过集群节点的IP或者私网弹性负载均衡ELB的服务地址访问。

- 公网访问：通过弹性IP从公网访问应用，一般用于系统中需要暴露到公网的服务。该访问方式需要给集群内任一节点绑定弹性IP，并设置一个映射在节点上的端口。

添加服务

* 服务名称

访问方式 集群内访问 VPC内网访问 公网访问

提供支持TCP/UDP协议的Internet访问入口，包含弹性IP方式。

* 访问类型

服务亲和 集群级别 节点级别

1、集群下所有节点的IP+访问端口均可以访问到此服务关联的负载。
2、服务访问会因路由跳转导致一定性能损失，且无法获取到客户端源IP。

* 端口映射

协议	容器端口	访问端口
<input type="text" value="TCP"/>	<input type="text" value="范围: 1-65535"/>	<input type="text" value="自动生成"/>

13.3.6 如何体验 ServiceStage 的源码部署功能？

如表13-3所示，ServiceStage基于GitHub提供了一些不同语言的demo。

您可以Fork特定语言的demo源码到自己的GitHub代码仓库中，参考7.2 创建并部署组件去体验ServiceStage的源码部署功能。

表 13-3 ServiceStage 提供的 demo 源码及 GitHub 地址说明

demo名称	语言类型	GitHub代码仓库地址
ServiceComb-SpringMVC	Java	https://github.com/servicestage-template/ServiceComb-SpringMVC
ServiceComb-JAX-RS	Java	https://github.com/servicestage-template/ServiceComb-JAX-RS
ServiceComb-POJO	Java	https://github.com/servicestage-template/ServiceComb-POJO
SpringBoot-WebService	Java	https://github.com/servicestage-template/SpringBoot-WebService
SpringBoot-Webapp-Tomcat	Java	https://github.com/servicestage-template/SpringBoot-Webapp-Tomcat
nodejs-express	Node.js	https://github.com/servicestage-template/nodejs-express-4-16

demo名称	语言类型	GitHub代码仓库地址
nodejs-koa	Node.js	https://github.com/servicestage-template/nodejs-koa-2-5-2
php-laravel	PHP	https://github.com/servicestage-template/php-laravel-v5-6-28
php-slim	PHP	https://github.com/servicestage-template/php-slim-3-10-0

13.4 持续交付

13.4.1 ServiceStage 怎么管理 IDEA 上的代码？

IDEA是本地的IDE，在IDE上编码后上传代码库，在ServiceStage上选择源码部署。

如果是基于ServiceComb框架开发的话，创建ServiceComb应用，选择源码部署，指定引擎，就可以治理了。

13.4.2 添加构建服务器地址到 GitLab 服务器安全组

背景介绍

若您的GitLab服务搭建在云内部网络，且公网无法直接访问，则需将构建服务的相关地址添加到您的GitLab服务器安全组中，以保证构建任务顺利执行。

操作步骤

步骤1 将ServiceStage所在网段加到GitLab私有仓库所在节点的安全组中，构建服务使用该IP访问GitLab服务的接口。

操作方法，请参考[添加安全组规则](#)。

📖 说明

ServiceStage所在网段，请联系技术支持工程师获取。

步骤2 获取构建镜像的集群名称和过滤节点标签信息。

- 应用组件构建，请参考[编辑源码构建工程](#)，获取“构建集群”和“过滤节点标签”。
- 构建任务构建，请参考[9.3 创建源码构建任务](#)，获取“构建集群”和“过滤节点标签”。

图 13-1 获取集群名称和标签信息



步骤3 获取该集群下有该标签的节点弹性IP。

- 应用组件构建
 - a. 在ServiceStage控制台，选择“持续交付 > 构建”，进入构建页面。
 - b. 单击构建集群的名称，进入集群详情页面。
 - c. 单击“节点管理”，获取该集群下有该标签的节点弹性IP。
- 构建任务构建
 - a. 在ServiceStage控制台，选择“持续交付 > 构建”，进入构建页面。
 - b. 选择构建任务，单击该构建任务的构建集群名称，进入集群详情页面。
 - c. 单击“节点管理”，获取该集群下有该标签的节点弹性IP。

步骤4 将**步骤3**中获取的构建镜像的运行节点添加到GitLab私有仓库所在节点的安全组中，构建时会访问GitLab服务获取代码。

操作方法，请参考[添加安全组规则](#)。

----结束

13.4.3 添加构建服务器地址到 Maven 服务器安全组

背景信息

将构建集群的构建节点弹性IP添加到Maven私有服务所在节点的安全组中，以便构建服务访问Maven私有服务器下载依赖包。

操作步骤

步骤1 获取构建镜像的集群名称和过滤节点标签信息。

- 应用组件构建，请参考[编辑源码构建工程](#)，获取“构建集群”和“过滤节点标签”。
- 构建任务构建，请参考[9.3 创建源码构建任务](#)，获取“构建集群”和“过滤节点标签”。

图 13-2 获取集群名称和标签信息



步骤2 获取该集群下有该标签的节点弹性IP。

- 应用组件构建
 - a. 在ServiceStage控制台，选择“持续交付 > 构建”，进入构建页面。
 - b. 单击构建集群的名称，进入集群详情页面。
 - c. 单击“节点管理”，获取该集群下有该标签的节点弹性IP。
- 构建任务构建

- a. 在ServiceStage控制台，选择“持续交付 > 构建”，进入构建页面。
- b. 选择构建任务，单击该构建任务的构建集群名称，进入集群详情页面。
- c. 单击“节点管理”，获取该集群下有该标签的节点弹性IP。

步骤3 将构建集群的构建节点弹性IP添加到Maven私有服务所在节点的安全组中。

操作方法，请参考[添加安全组规则](#)。

----结束

13.5 基础设施

13.5.1 如何处理开启了安全认证的微服务引擎专享版开启 IPv6 后服务注册失败？

问题描述

基于Java Chassis开发的微服务注册到开启了安全认证的微服务引擎专享版，微服务的注册发现地址使用微服务引擎服务注册发现的IPv4地址，可以注册成功并正常启动。

如果修改微服务的注册发现地址为微服务引擎注册发现的IPv6地址后，注册失败并报错“java.net.SocketException: Protocol family unavailable”。

可能原因

创建微服务引擎专享版时，当选择开启了IPv6的VPC网络时，创建引擎支持IPv6网络。当部署服务使用IPv6网段且选择容器部署时，选择的CCE集群需要开启IPv6双栈开关。

如果选择的CCE集群资源没有开启IPv6开关，就会导致服务网络不通，报错“java.net.SocketException: Protocol family unavailable”。

解决方法

步骤1 修改部署了微服务应用的环境，添加开启了“IPv6双栈”开关的CCE集群。

修改环境，请参考[5.6 修改环境](#)。

步骤2 重新部署应用，请参考[7.2 创建并部署组件](#)。

----结束

13.5.2 如何处理操作微服务引擎专享版时遇到非微服务引擎本身错误？

问题描述

在对微服务引擎专享版执行创建、删除、升级等操作时，可能会遇到非微服务引擎本身的错误。

例如，在创建微服务引擎专享版时，集群部署失败，报错如下：

```
{"error_code":"SVCSTG.00500400","error_message":{"kind":"Status","apiVersion":"v1","metadata":{},"status":{"Failure"},"code":400,"errorCode":"CCE.01400013","errorMessage":"Insufficient volume"}}
```

```
quota.\","error_code\":"CCE_CM.0307\","error_msg\":"Volume quota is not enough\","message  
\":"volume quota checking failed as [60/240] insufficient volume size quota\","reason\":"QuotaInsufficient  
\}"}}
```

解决方法

页面上展示的错误信息中已经包含相应服务的错误码，根据错误码和错误信息联系相应服务的技术支持工程师提供支持。

13.6 应用运维

13.6.1 为什么 ServiceStage 中看不到日志？

ServiceStage中看不到日志，可能是由于待查看日志的主机未安装ICAgent或者用户业务日志输出位置为非标准位置导致的。

解决方法

- 待查看日志的主机未安装ICAgent
ServiceStage的日志查看能力是由AOM服务提供的。主机是否安装ICAgent是使用AOM的日志能力的前提，否则将无法查看ServiceStage的日志。ICAgent是AOM的采集器，分别运行在每台主机上用于实时采集指标、日志和应用性能数据。
如何为待查看日志的主机安装ICAgent，请参考[安装ICAgent](#)。
- 用户业务日志输出位置为非标准位置
由于用户配置了日志策略，导致用户程序业务日志未输出到标准的输出位置。需参考如下方法进行排查处理：
 - 虚拟机部署
排查配置的日志策略，是否把用户程序业务日志输出位置写到ServiceStage默认指定的虚拟机日志目录（/var/log/application/\${组件名}-\${环境名}-\${随机字符串}/\${版本号}/\${实例ID}/start_app.log）外的其他目录。
请查询业务代码，对日志策略进行调整。
 - 容器部署
排查配置的日志策略，是否把业务日志输出到出标准输出外的其他地方。请参考[7.17.7 设置应用日志策略](#)进行相关配置。

13.6.2 替换弹性 IP 后应用访问方式失效

问题描述

当外网负载均衡绑定到应用时，如果把负载均衡的弹性IP替换掉，则应用访问方式上无法自动更新。

解决方法

需要手动删除之前的记录，重新添加新更换的IP，以新IP为访问地址的ELB访问方式。

13.6.3 启动一个新服务后导致一个节点内存使用率过高

问题描述

启动一个新服务导致一个节点内存使用率过高，怎么处理？

解决方法

设置[调度策略亲和性](#)，可以使服务实例按亲和性节点部署即可。