



# 微服务云应用平台

## 用户使用指南

天翼云科技有限公司

---

# 目 录

---

<b>1 产品介绍</b> .....	<b>6</b>
1.1 什么是微服务云应用平台（ServiceStage） .....	6
1.2 产品优势 .....	8
1.3 典型应用场景 .....	9
1.3.1 构建微服务应用 .....	9
1.3.2 Web 应用生命周期管理 .....	12
1.4 微服务引擎版本支持机制 .....	12
1.5 名词解释 .....	13
1.6 使用限制 .....	14
1.7 产品规格差异 .....	15
1.8 权限管理 .....	16
<b>2 快速入门</b> .....	<b>25</b>
2.1 快速体验 ServiceStage.....	25
2.2 附录 .....	30
2.2.1 AK/SK 获取方法 .....	30
<b>3 用户指南</b> .....	<b>31</b>
3.1 概述 .....	31
3.2 权限管理 .....	32
3.2.1 创建用户并授权使用 ServiceStage.....	32
3.2.2 创建 ServiceStage 自定义策略.....	33
3.2.3 设置 ServiceStage 依赖服务的权限.....	35
3.3 应用管理 .....	35
3.3.1 创建应用 .....	35
3.3.2 新建应用组件 .....	36
3.3.2.1 应用组件概述 .....	36
3.3.2.2 快速创建组件 .....	39
3.3.2.3 创建微服务组件 .....	40
3.3.2.4 创建 Web 应用组件 .....	43
3.3.2.5 创建通用组件 .....	45
3.3.3 应用组件部署 .....	47

---

3.3.3.1 部署方式说明 .....	47
3.3.3.2 部署组件 .....	47
3.3.4 应用组件管理 .....	52
3.3.5 应用高级设置 .....	56
3.3.5.1 设置应用组件环境变量 .....	56
3.3.5.2 设置应用生命周期 .....	57
3.3.5.3 设置数据存储 .....	58
3.3.5.4 设置分布式会话 .....	65
3.3.6 应用组件构建 .....	66
3.3.7 应用组件流水线 .....	68
3.3.8 应用配置管理 .....	69
3.3.8.1 创建密钥 .....	69
3.3.8.2 创建配置项 .....	72
3.4 环境管理 .....	74
3.4.1 环境概述 .....	74
3.4.2 创建环境 .....	74
3.4.3 纳管资源 .....	75
3.4.4 移除已纳管资源 .....	76
3.4.5 修改环境 .....	76
3.4.6 删除环境 .....	77
3.5 应用运维 .....	77
3.5.1 维护应用组件实例 .....	77
3.5.2 添加应用组件实例标签 .....	80
3.5.3 配置域名映射 .....	81
3.5.4 设置资源监报告警阈值 .....	81
3.5.5 设置应用组件实例伸缩策略 .....	83
3.5.6 设置应用组件实例调度策略 .....	91
3.5.7 设置应用组件实例升级策略 .....	95
3.5.8 设置应用组件自定义指标监控 .....	97
3.5.9 配置应用日志策略 .....	100
3.5.10 设置应用性能管理 .....	101
3.5.11 设置应用健康检查 .....	104
3.6 持续交付 .....	106
3.6.1 持续交付概述 .....	106
3.6.2 创建源码构建任务 .....	107
3.6.3 创建软件包构建任务 .....	110
3.6.4 流水线管理 .....	112
3.6.5 仓库授权 .....	115
3.7 软件中心 .....	116

3.7.1 镜像仓库 .....	116
3.7.1.1 上传镜像 .....	116
3.7.1.2 管理镜像 .....	117
3.7.2 组织管理 .....	119
3.8 微服务引擎 CSE .....	120
3.8.1 微服务引擎概述 .....	120
3.8.2 创建微服务引擎专享版 .....	120
3.8.3 微服务引擎（CSE）管理 .....	123
3.8.3.1 设置微服务引擎专享版备份与恢复 .....	123
3.8.3.2 设置微服务引擎专享版公网访问 .....	124
3.8.3.3 查看微服务引擎服务注册发现地址和配置中心地址 .....	125
3.8.3.4 查看微服务引擎专享版操作日志 .....	126
3.8.3.5 升级微服务引擎专享版版本 .....	126
3.8.3.6 删除微服务引擎专享版 .....	127
3.8.3.7 管理微服务引擎专享版安全认证 .....	127
3.8.4 使用微服务仪表盘 .....	129
3.8.5 服务治理 .....	130
3.8.5.1 服务治理概述 .....	130
3.8.5.2 治理微服务 .....	130
3.8.6 配置微服务 .....	140
3.8.7 维护微服务 .....	141
3.8.8 系统管理 .....	147
3.8.8.1 系统管理概述 .....	147
3.8.8.2 帐号管理 .....	148
3.8.8.3 角色管理 .....	152
3.9 云审计服务支持的关键操作 .....	155
3.9.1 云审计服务支持的 ServiceStage 操作列表 .....	155
3.9.2 查看云审计日志 .....	156
3.10 查看监控指标及告警 .....	156
<b>4 常见问题 .....</b>	<b>160</b>
4.1 应用开发问题咨询指引 .....	错误!未定义书签。
4.2 环境管理 .....	160
4.2.1 微服务和其他平台服务有什么不同? .....	160
4.3 应用管理 .....	160
4.3.1 如何查看应用组件部署失败的原因? .....	160
4.3.2 实例长期处于创建中怎么办? .....	161
4.3.3 如何解决 Docker 运行 node 应用程序时的依赖问题? .....	161
4.3.4 定制 Tomcat Context path .....	161
4.3.5 如何固定应用组件 IP? .....	162

---

4.3.6 如何体验 ServiceStage 的源码部署功能? .....	162
4.4 持续交付 .....	163
4.4.1 ServiceStage 怎么管理 IDEA 上的代码? .....	163
4.4.2 添加构建服务器地址到 GitLab 服务器安全组 .....	163
4.4.3 添加构建服务器地址到 Maven 服务器安全组 .....	164
4.4.4 使用 ServiceStage 构建失败 .....	165
4.5 软件中心 .....	166
4.5.1 如何解决上传软件包失败的问题? .....	166
4.5.2 如何解决 Docker 客户端 push 镜像失败的问题 .....	166
4.5.3 下载 SWR 软件包失败 .....	167
4.6 基础设施 .....	167
4.6.1 退订服务器是否影响现有程序运行? .....	167
4.6.2 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? .....	168
4.6.3 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? .....	<b>错误!未定义书签。</b>
4.6.4 如何将组件构建集群由共享构建集群切换为私有构建集群? .....	168
4.7 应用运维 .....	169
4.7.1 为什么 ServiceStage 中看不到日志? .....	169
4.7.2 替换弹性 IP 后应用访问方式失效 .....	170
4.7.3 启动一个新服务后导致一个节点内存使用率过高 .....	170
4.8 应用开发问题 .....	170
4.8.1 微服务和普通应用有什么不同? .....	170
4.8.2 部署在云上的微服务如何进行排错? .....	171
4.8.3 如何决定使用 SDK 构建微服务还是使用 ServiceMesh? .....	<b>错误!未定义书签。</b>
4.8.4 如何解决获取依赖失败的问题? .....	171
4.8.5 服务名重复校验范围是什么? .....	172
4.8.6 为什么一定要定义服务契约? .....	172

# 1 产品介绍

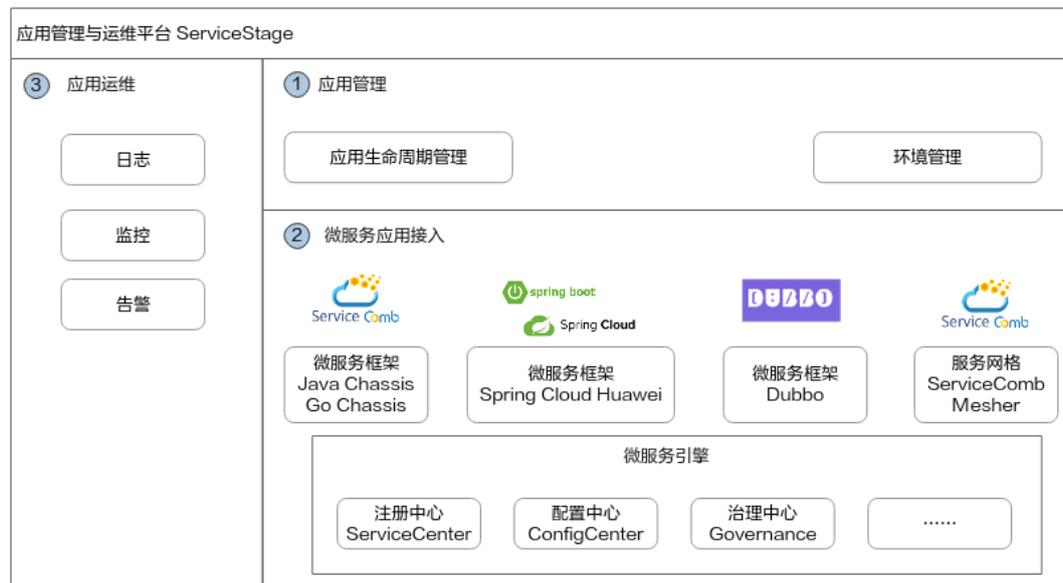
## 1.1 什么是微服务云应用平台（ServiceStage）

微服务云应用平台（ServiceStage）是面向企业的微服务云应用平台，提供应用发布、部署、监控与运维等一站式解决方案。支持 Java、Go、Node.js、Docker、Tomcat 等运行环境。支持 Apache ServiceComb Java Chassis（Java Chassis）、Spring Cloud、Dubbo、Mesher 服务网格等微服务应用，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、Go Chassis、Spring Cloud、Dubbo 微服务框架；支持 Mesher 服务网格。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，请参考“帮助中心 > 微服务云应用平台 > 开发指南 > 微服务开发指南”章节。
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/War/Zip）和容器镜像进行应用组件创建，实现应用部署。
  - 应用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、网络和中间件等基础资源的集合。ServiceStage 把相同 VPC 下的计算资源（如云容器引擎 CCE 等）、网络资源（如弹性负载均衡 ELB、弹性公网 EIP 等）和中间件（如分布式缓存 DCS、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。

微服务引擎采用的注册发现中心 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注册发现中心，以供服务消费者发现并使用。

- 兼容主流微服务开源框架

为 Spring Cloud 与 Dubbo 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中心。您可以参考[服务治理](#)，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"> <li>• 资源获取效率低 (&gt;1 天)</li> <li>• 资源利用率低 (&lt;30%)</li> </ul>	<ul style="list-style-type: none"> <li>• 自助高效获取资源 (分钟级)</li> </ul>
业务开发阶段	<ul style="list-style-type: none"> <li>• 架构耦合，牵一发而动全身</li> <li>• 技术单一，需要想办法用一种技术解决所有问题</li> <li>• 只能按大颗粒系统发布版本，响应周期长</li> </ul>	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 各种技术灵活接入 支持 JAVA、Go、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。</li> <li>• 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。</li> </ul>
安装部署阶段	<ul style="list-style-type: none"> <li>• 烟囱式系统</li> </ul>	开发者只需使用 ServiceStage+任意源码

应用生命周期	传统平台	使用 ServiceStage 平台
	<ul style="list-style-type: none"> <li>全手工部署</li> </ul>	仓库，实现一键自动部署和更新。
应用配置	<ul style="list-style-type: none"> <li>配置项复杂</li> <li>每个环境配一遍</li> <li>易出错</li> </ul>	<ul style="list-style-type: none"> <li>支持配置以文件的形式导入到应用中。</li> <li>实现配置文件与环境解耦，一次维护，多个环境共用。</li> <li>配置文件支持多版本，方便更新和回滚。</li> </ul>
应用升级	<ul style="list-style-type: none"> <li>打补丁方式</li> <li>手动升级</li> <li>中断业务</li> </ul>	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"> <li>应用崩溃，闪退</li> <li>服务端响应慢</li> <li>系统资源不足</li> <li>故障难定位</li> </ul>	<ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>

## 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，微服务应用可以完美解决上述问题。

##### 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、Go、Node.js、Docker、Tomcat 等运行环境，可无侵入托管 Apache ServiceComb Java Chassis、Spring Cloud、

Dubbo、服务网格等微服务应用，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 提供了业内领先的微服务应用解决方案，具有以下优势：

- 支持原生 ServiceComb、Spring Cloud、Dubbo 和 Service Mesh 多种微服务框架，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、Go、Node.js 等。
- 提供服务中心、配置中心、仪表盘、微服务灰度发布等功能。
- 提供容错、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。
- 实现 Spring Cloud、Java Chassis 和 Go 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

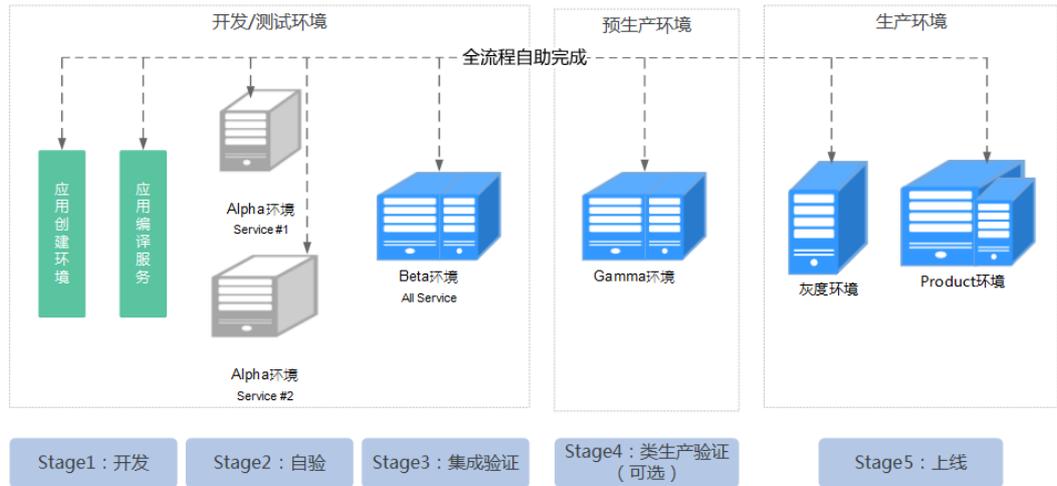
由人工执行变成了自动化执行，减少错误出现，提高工作效率。

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1-2 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。

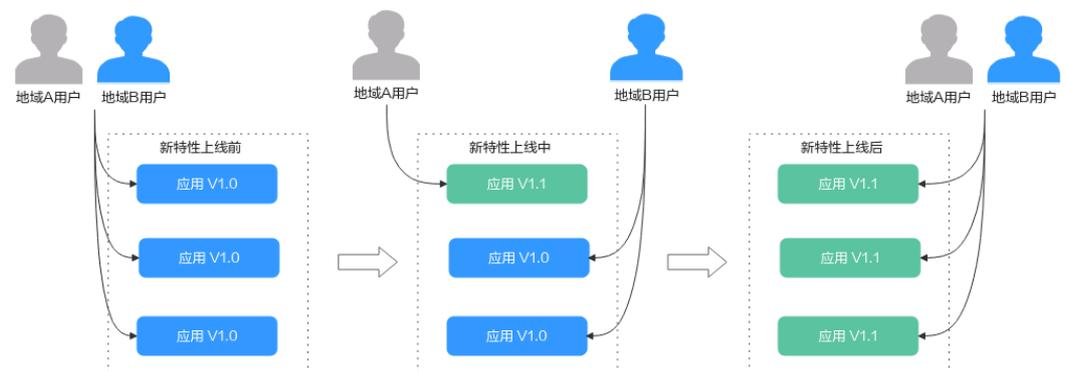
### 价值

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1-3 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、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。

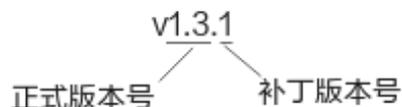
### 版本号说明

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，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 1.3、1.2、1.1，则支持由 1.1、1.2 升级到 1.3。

#### 📖 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、网络和中间件等基础资源的集合。

ServiceStage 把相同 VPC 下的计算资源（如云容器引擎 CCE）、网络资源（如弹性负载均衡 ELB、弹性公网 EIP 等）和中间件（如分布式缓存 DCS、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应用组件组成。

### 应用组件

应用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。

## 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，微服务对应应用组件。

### 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
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	-

项目	限制	备注
	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。

## 1.7 产品规格差异

### 微服务引擎服务数限制说明

- 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。

微服务引擎支持最大服务数说明如下。

表1-5 微服务引擎最大服务限制说明

引擎类型	规格（微服务实例数）
微服务引擎专业版	20
微服务引擎专享版	100
	200

引擎类型	规格（微服务实例数）
	500
	2000

## 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，并使用策略来控制他们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，您希望他们拥有 **ServiceStage** 的使用权限，但是不希望他们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，控制他们对 **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，请参见帮助中心 > 统一身份认证服务 > 用户指南 > 产品简介。

### 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 1-6 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1-6 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	微服务云应用平台所有权限。	系统策略	无
ServiceStage ReadOnlyAccess	微服务云应用平台只读权限。	系统策略	无
ServiceStage Developer	微服务云应用平台开发者权限。 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 Admin	微服务云应用平台管理员，拥有该服务下的所有权限。	系统角色	CCE Administrator、VPC Administrator、SWR Administrator 和 OBS Administrator。
ServiceStage Operator	微服务云应用平台操作员，拥有该服务下的只读权限。	系统角色	CCE Administrator、VPC Administrator、SWR Administrator 和 OBS Administrator。
ServiceStage Developer	微服务云应用平台开发者，拥有该服务下的所有权限。	系统角色	CCE Administrator、VPC Administrator、SWR Administrator 和 OBS Administrator。

如果表 1-6 所列的这些权限不满足实际需求，您可以参考表 1-7 和表 1-8，在这个基础上自定义策略。

表1-7 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	√	√
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	√	√	√

操作	ServiceStage ReadOnlyAccess	ServiceStage Developer	ServiceStage FullAccess
权			
删除仓库授权	x	√	√

表1-8 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	√
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

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 1-9](#)。
- ServiceStage 细粒度权限依赖说明请参见[表 1-10](#)。

表1-9 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"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	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"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	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。
cse:engine:delete	删除引擎	<ul style="list-style-type: none"> <li>cse:engine:list</li> </ul>	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"> <li>cse:engine:get</li> <li>cse:engine:list</li> <li>ecs:cloudServerFlavors:get</li> <li>vpc:vpcs:get</li> <li>vpc:vpcs:list</li> <li>vpc:subnets:get</li> <li>vpc:ports:get</li> <li>vpc:ports:create</li> </ul>	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"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:config:get</li> </ul>	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"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	查看服务的配置。
cse:governance:modify	服务治理中心修改	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:config:get</li> <li>cse:config:modify</li> <li>cse:registry:get</li> <li>cse:registry:modify</li> <li>cse:governance:get</li> </ul>	创建与修改服务治理。

权限名称	权限描述	权限依赖	应用场景
cse:governance:get	服务治理中心查看	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:config:get</li> <li>cse:registry:get</li> </ul>	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"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:registry:get</li> </ul>	服务修改。
cse:dashboard:modify	dashboard 管理修改	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:registry:get</li> <li>cse:dashboard:get</li> <li>cse:registry:modify</li> </ul>	仪表盘修改。
cse:dashboard:get	dashboard 管理查看	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:registry:get</li> </ul>	仪表盘查看。
cse:registry:get	服务注册管理查看	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	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1-10 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"> <li>servicestage:app:get</li> <li>servicestage:app:list</li> <li>servicestage:assembling:get</li> <li>servicestage:assembling:list</li> <li>servicestage:assembling:create</li> </ul>	创建应用。
servicestage:app:modify	更新应用	<ul style="list-style-type: none"> <li>servicestage:app:get</li> <li>servicestage:app:list</li> </ul>	更新应用。

权限名称	权限描述	权限依赖	应用场景
		<ul style="list-style-type: none"> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> <li>• servicestage:assembling:modify</li> </ul>	
servicestage:app:delete	删除应用	<ul style="list-style-type: none"> <li>• servicestage:app:get</li> <li>• servicestage:app:list</li> <li>• servicestage:assembling:delete</li> </ul>	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"> <li>• servicestage:app:get</li> <li>• servicestage:app:list</li> </ul>	创建环境。
servicestage:environment:modify	更新环境	<ul style="list-style-type: none"> <li>• servicestage:app:get</li> <li>• servicestage:app:list</li> </ul>	更新环境。
servicestage:environment:delete	删除环境	<ul style="list-style-type: none"> <li>• servicestage:app:get</li> <li>• servicestage:app:list</li> </ul>	删除环境。
servicestage:pipeline:get	查看流水线信息	<ul style="list-style-type: none"> <li>• servicestage:pipeline:list</li> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> </ul>	查看流水线信息。
servicestage:pipeline:create	创建流水线	<ul style="list-style-type: none"> <li>• servicestage:pipeline:list</li> <li>• servicestage:pipeline:get</li> <li>• servicestage:assembling:create</li> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> </ul>	创建流水线。

权限名称	权限描述	权限依赖	应用场景
servicestage:pipeline:modify	修改流水线	<ul style="list-style-type: none"> <li>• servicestage:pipeline:get</li> <li>• servicestage:pipeline:list</li> <li>• servicestage:assembling:modify</li> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> </ul>	修改流水线。
servicestage:pipeline:delete	删除流水线	<ul style="list-style-type: none"> <li>• servicestage:pipeline:get</li> <li>• servicestage:pipeline:list</li> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> <li>• servicestage:assembling:delete</li> </ul>	删除流水线。
servicestage:pipeline:list	查看流水线列表	<ul style="list-style-type: none"> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> </ul>	查看流水线列表
servicestage:pipeline:execute	执行流水线	<ul style="list-style-type: none"> <li>• servicestage:pipeline:get</li> <li>• servicestage:pipeline:list</li> <li>• servicestage:assembling:modify</li> <li>• servicestage:assembling:get</li> <li>• servicestage:assembling:list</li> <li>• servicestage:app:get</li> <li>• servicestage:app:list</li> <li>• servicestage:app:modify</li> </ul>	执行流水线。
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"> <li>• servicestage:assembling:get</li> </ul>	创建构建。

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

# 2 快速入门

## 2.1 快速体验 ServiceStage

### 概述

微服务云应用平台（ServiceStage）是面向企业的微服务云应用平台，提供应用发布、部署、监控与运维等一站式解决方案。支持 Java、Go、Node.js、Docker、Tomcat 等运行环境。支持 Apache ServiceComb Java Chassis（Java Chassis）、Spring Cloud、Dubbo、Mesher 服务网格等微服务应用，让企业应用上云更简单。

ServiceStage 新应用模型提供了环境管理功能，把相同 VPC 下的基础资源（如 CCE 集群）加上可选资源（如 ELB、DCS 等）实例组合为一个环境，部署应用时选择环境会自动加载包含的资源。

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。应用组件是组成应用的某个业务特性的实现，以代码或者软件包为载体，可独立部署运行。

针对应用的组件提供启停、升级、回退、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。

本例基于 ServiceComb（SpringMVC）框架，快速创建微服务应用，供您体验 ServiceStage 新应用模型功能。

#### 说明

ServiceStage 基于 GitHub 提供了一些不同语言的 demo。体验特定语言 demo 在 ServiceStage 中的源码部署功能，请参考[如何体验 ServiceStage 的源码部署功能?](#)

### 前提条件

- 注册帐号，并登录成功。
- 获取 AK/SK，请参考 [AK/SK 获取方法](#)。

#### 说明

- 如果使用微服务引擎专业版，需要配置 AK/SK。
- 如果使用微服务引擎专享版，不需要配置 AK/SK。但需先创建未开启安全认证的微服务引擎专享版，请参考[创建微服务引擎专享版](#)。

本操作使用微服务引擎专业版（Cloud Service Engine）。

- 创建一个虚拟私有网络 VPC，请参考“帮助中心 > 虚拟私有云 > 用户指南 > 虚拟私有云和子网 > 虚拟私有云 > 创建虚拟私有云和子网”。
- 创建一个 CCE 集群，集群中至少包含一个 ECS 节点（为方便后续步骤的操作，节点规格最好选择 4vCPUs、8GB 内存）并且绑定弹性 IP。集群绑定弹性 IP，请参考“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 创建 CCE 集群”。
- 本例基于 ServiceStage 绑定 GitHub 源码仓库，实现源码构建、归档、应用创建，需要先到 [GitHub](#) 官网注册帐号。
- 已在 ServiceStage 创建仓库授权，授权访问个人 GitHub 仓库，请参考[仓库授权](#)。

## Fork 源码

登录个人 GitHub 帐号，并 Fork Demo 源码。

Demo 源码地址：<https://github.com/servicestage-template/ServiceComb-SpringMVC>。

## 创建组织

步骤 1 登录 ServiceStage 控制台，选择“软件中心 > 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”，例如：org-ss。

步骤 3 单击“确定”。

----结束

## 创建环境

步骤 1 登录 ServiceStage，选择“环境管理”。

步骤 2 单击“创建环境”，参考下表设置环境信息。

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，默认使用 default（开通企业项目管理后才显示该参数）。
虚拟私有云 (VPC)	在下拉列表选择 <a href="#">前提条件</a> 中已创建的虚拟私有云 VPC。 说明 <ul style="list-style-type: none"><li>• 选定 VPC 后，会加载该 VPC 下的资源供选择，不在该 VPC 下的资源无法选择。</li><li>• 环境创建完成后，不支持修改 VPC。</li></ul>
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“纳管资源”。

步骤 5 在弹出的对话框中，勾选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“Cloud Service Engine”资源，单击“确定”。

----结束

## 新建应用

步骤 1 登录 ServiceStage，选择“应用管理 > 应用列表”，单击“创建应用”。

步骤 2 设置应用基本信息。

1. “应用名称”：输入应用名称，例如：“servicecomb”。
2. “企业项目”：默认使用“default”（开通企业项目管理后才显示该参数）。

步骤 3 单击“确定”，完成应用创建。

----结束

## 新增组件

步骤 1 登录 ServiceStage，选择“应用管理 > 应用列表”。

步骤 2 选择 servicecomb 应用，在“操作”栏单击“新增组件”。

步骤 3 “配置方式”选择“使用模板配置”，选择“ServiceComb”，微服务模板，单击“下一步”。

步骤 4 设置组件信息。

1. “组件名称”：输入“java-test”。
2. “源码/软件包”：选择“源码仓库”，勾选“GitHub”。
  - a. “授权信息”：选择创建仓库授权时创建的授权信息。
  - b. “用户名/组织”：选择 Fork 源码时登录您的 GitHub 使用的用户名。
  - c. “仓库名称”：选择已 Fork 到您的 GitHub 下的天气预报源码仓库的名称，例如：ServiceComb-SpringMVC。
  - d. “分支”：选择“master”。

步骤 5 开启构建。

1. “组织”：选择创建组织时创建的组织名称。
2. “选择集群”：选择创建环境时选择的 CCE 集群。
3. 其余参数，保持默认。

步骤 6 单击“立即创建”，创建静态组件。

----结束

## 部署组件

步骤 1 登录 ServiceStage，选择“应用管理 > 应用列表”。

步骤 2 单击 servicecomb 应用名称，进入应用“概览”页。

步骤 3 在“组件列表”选择已经创建的组件 java-test，在“操作”栏单击“部署”。

步骤 4 设置基本配置。

1. “环境”：选择[创建环境](#)时已经创建的环境“test-env”。
2. “部署版本”：默认设置为“1.0.0”。
3. “部署系统”：选择“云容器引擎”。
4. “基础资源”：使用环境 test-env 中的 CCE 资源（自动加载）。
5. “实例数量”：设置为“1”。
6. “资源配额”：使用默认配置。

步骤 5 单击“下一步 组件配置”，进行组件配置。

1. “镜像”：默认加载已经配置的组件静态信息。
2. “微服务引擎”：默认使用环境 test-env 中的 Cloud Service Engine 微服务引擎专业版。

### 说明

- 应用组件部署以后，微服务会注册到选择的微服务引擎。
  - 所有应用组件需要注册到同一个微服务引擎，才能互相发现。
3. 设置环境变量。  
选择“高级设置 > 组件配置”，进入“环境变量”，单击“添加环境变量”，添加如下环境变量。

类型	变量名	变量值
手动添加	servicecomb_credentials_accessKey	获取 <a href="#">AK/SK</a> 中获取的 AK 值。
	servicecomb_credentials_secretKey	获取 <a href="#">AK/SK</a> 中获取的 SK 值。

4. 其他参数使用默认配置。

步骤 6 单击“下一步 规格确认”，确认规格无误后，单击“部署”。

等待组件部署完成。

----结束

## 确认部署结果

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 选择 Cloud Service Engine 微服务引擎，单击“查看控制台”，进入微服务控制台。

步骤 3 选择“服务目录 > 微服务列表”，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，且微服务实例数不为 0，则部署成功。

----结束

## 访问应用

步骤 1 登录 ServiceStage，选择“应用管理 > 应用列表”。

步骤 2 单击 servicecomb 应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，“环境”选择 test-env，查看已经部署的应用组件。

步骤 4 单击 java-test 应用组件名称，进入组件“概览”页面。

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	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	选择默认设置。
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。

图2-1 访问地址



步骤 8 单击图 2-1 所示的“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://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，选择“应用管理 > 应用列表”。

步骤 2 单击 servicecomb 应用名称，进入应用“概览”页。

步骤 3 在“组件列表”单击已经创建的 java-test 组件名称，进入组件“概览”页。

步骤 4 在对应“环境”卡片可以查看 java-test 组件版本。

步骤 5 单击 java-test 组件名称，进入组件实例“概览”页。

可以进行运维操作（启停、升级、回退、伸缩、查看日志、查看事件、设置访问方式、设置阈值告警等）。

----结束

## 2.2 附录

### 2.2.1 AK/SK 获取方法

#### 说明

请以应用所需权限用户登录 ServiceStage 控制台，用户权限说明请参见[权限管理](#)。

步骤 1 登录 ServiceStage 控制台。

步骤 2 单击登录用户名，在下拉菜单选择“我的凭证”。

步骤 3 在“管理访问密钥”页签，单击“新增访问密钥”，通过身份认证后成功创建 AK/SK。

步骤 4 单击“立即下载”。

下载成功后，在 credentials 文件中获取 AK 和 SK 信息：

- Access Key Id 的值即为 AK。
- Secret Access Key 的值即为 SK。

#### 须知

- 每个用户仅允许保留 2 个有效的访问密钥。
- 为保证访问密钥的安全，访问密钥仅在初次生成时自动下载，后续不可再次通过管理控制台界面获取。请妥善保管访问密钥。

----结束

# 3 用户指南

## 3.1 概述

微服务云应用平台（ServiceStage）是面向企业的微服务云应用平台，提供应用发布、部署、监控与运维等一站式解决方案。支持 Java、Go、Node.js、Docker、Tomcat 等运行环境。支持 Apache ServiceComb Java Chassis（Java Chassis）、Spring Cloud、Dubbo、Mesher 服务网格等微服务应用，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、部署应用以及如何应用运维、服务治理。

### 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	提供 ServiceStage 产品总体看板，包含使用教程、应用、环境、组件等信息。
应用管理	<ul style="list-style-type: none"><li>应用列表 提供应用生命周期管理的功能，如应用创建、新增组件、组件列表、环境视图、组件部署、组件详情页及运维入口等。</li><li>应用配置 提供配置项管理和密钥管理功能。</li></ul>
环境管理	环境是用于应用部署和运行的计算、网络、中间件等基础资源的集合。 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。
微服务引擎 CSE	在微服务引擎 CSE 界面，进入微服务引擎控制台，可以进行微服务治理相关的操作。
持续交付	提供工程构建、发布等功能。

类别	说明
	<ul style="list-style-type: none"> <li>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-&gt;编译-&gt;打包-&gt;归档”的全流程自动化。</li> <li>发布 通过流水线工程可以完成一键部署，实现“源码拉取-&gt;编译-&gt;打包-&gt;归档-&gt;部署”的全流程自动化。利于集成环境统一、交付流程标准化。</li> <li>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。</li> </ul>
软件中心	提供组织管理、镜像仓库等功能。 <ul style="list-style-type: none"> <li>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。</li> <li>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）。</li> </ul>

## 3.2 权限管理

### 3.2.1 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 3-1 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[权限管理](#)。

## 示例流程

图3-1 给用户授权 ServiceStage 权限流程



1. 创建用户组并授权  
在 IAM 控制台创建用户组，并授予 ServiceStage “ServiceStage ReadOnlyAccess” 权限。
2. 创建用户  
在 IAM 控制台创建用户，并将其加入 1 中创建的用户组。
3. 用户登录并验证权限  
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“微服务云应用平台 ServiceStage”，进入“应用管理 > 应用列表”，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。

### 3.2.2 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：“帮助中心 > 统一身份认证服务 > 用户指南 > 用户指南 > 权限管理 > 自定义策略 > 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 CSE”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。

### 3.2.3 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“帮助中心 > 云容器引擎 > 用户指南 > 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。

CTS 服务权限设置方法，请参考“帮助中心 > 云审计服务 > 用户指南 > 权限管理”。

## 3.3 应用管理

### 3.3.1 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应用组件组成。

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 可支持单个用户最多创建 1000 个应用。

#### 创建应用

**步骤 1** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

**步骤 2** 单击“创建应用”，设置应用基本信息。

1. “应用名称”：输入应用名称，应用名称在创建完成后可以修改。
2. “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。

开通企业项目后可以使用。

3. “描述”：（可选）输入应用描述。

**步骤 3** 单击“确定”，创建应用。

----结束

## 添加应用环境变量

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，对此应用下的所有应用组件都起作用。

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

**步骤 1** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

**步骤 2** 单击应用名称，进入应用“概览”页。

**步骤 3** 单击“环境变量”，在“环境”下拉列表选择已经创建的环境。

**步骤 4** 单击“添加环境变量”，输入“Key”及“Value”值。

其中“Key”是环境变量的名称，“Value”是环境变量的值。提交后完成环境变量设置。

例如，设置“Key”为“User”，“Value”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 5** 单击“提交”，完成环境变量添加。

----结束

## 3.3.2 新建应用组件

### 3.3.2.1 应用组件概述

应用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。

在 ServiceStage 上创建应用后，可以在应用中添加组件，目前支持的组件类型有微服务、Web、通用。同一个应用下，最多可支持创建 1000 个应用组件。

您可以通过设置组件类型、框架、运行时及组件来源，先创建静态组件，然后再部署。

在新增组件的操作流程中，支持“使用模板配置”和“自定义配置”两种配置方式：

- “使用模板配置”为您提供组件类型、运行时、框架的默认配置，可以帮助您快速创建组件。
- “自定义配置”支持灵活选择组件类型、运行时与合适的框架/服务网格。

## 已有模板说明

表3-2 已有模板信息

组件类型	运行时	框架
------	-----	----

组件类型	运行时	框架
ServiceComb MicroService	Java8	Java Chassis
SpringCloud MicroService	Java8	Spring Cloud
Web(Tomcat) WebApp	Tomcat8	Web

## 微服务组件说明

支持的运行时	支持的框架/服务网格	支持的源码/软件包
Java8	Java Chassis	源码仓库、Jar 包
Tomcat8		源码仓库、War 包
Docker		不需要设置该参数
Java8	Mesher	源码仓库、Jar 包
Tomcat8		源码仓库、War 包
Nodejs8		源码仓库、Zip 包
Php7		源码仓库、Zip 包
Docker		不需要设置该参数
Python3		源码仓库、Zip 包
Docker	Go Chassis	不需要设置该参数
Java8	Spring Cloud	源码仓库、Jar 包
Tomcat8		源码仓库、War 包
Docker		不需要设置该参数
Java8	Dubbo	源码仓库、Jar 包
Tomcat8		源码仓库、War 包
Docker		不需要设置该参数

## Web 应用组件说明

支持的运行时	支持的源码/软件包
Java8	源码仓库、Jar 包
Nodejs8	源码仓库、Zip 包
Php7	源码仓库、Zip 包

支持的运行时	支持的源码/软件包
Tomcat8	源码仓库、War 包
Docker	不需要设置该参数
Python3	源码仓库、Zip 包

### 通用组件说明

支持的运行时	支持的源码/软件包
Java8	源码仓库、Jar 包
Tomcat8	源码仓库、War 包
Nodejs8	源码仓库、Zip 包
Php7	源码仓库、Zip 包
Docker	不需要设置该参数
Python3	源码仓库、Zip 包

### 组件来源说明

组件来源	说明
源码仓库	参考 <a href="#">仓库授权</a> 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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 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“帮助中心 > 对象存储服务

组件来源	说明
	务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。
镜像包	容器应用需要基于镜像创建，若选择私有镜像，用户首先需要将镜像上传至镜像仓库。选择“软件中心 > 镜像仓库”，参考 <a href="#">管理镜像</a> 将镜像上传至镜像仓库。

### 3.3.2.2 快速创建组件

ServiceStage 系统默认提供了 3 个模板，请参见[已有模板说明](#)。

模板提供了组件类型、语言/运行时、框架/服务网格的默认配置，可以帮助您快速创建组件。

#### 前提条件

1. 只能在应用下新增组件，请先创建应用，请参考[创建应用](#)。
2. 如果您基于“源码仓库”创建微服务组件，那么您首先需要创建仓库授权，请参考[仓库授权](#)。
3. 如果您基于软件包创建微服务组件，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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。

#### 操作步骤

- 步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。
- 步骤 2 选择已经创建的应用，单击“操作”栏的“新增组件”。
- 步骤 3 “配置方式”选择“使用模板配置”，选择模板，单击“下一步”。
- 步骤 4 参照下表设置组件信息，参数前面带\*号的是必须设置的参数。

表3-3 组件基本信息

参数	参数说明
*组件名称	组件对应的名称。
*软件包	<ul style="list-style-type: none"> <li>• 选择“源码仓库”                             <ul style="list-style-type: none"> <li>- 参考<a href="#">仓库授权</a>创建授权，设置代码来源。</li> <li>- 设置“开启构建”参数，用于应用组件构建： 根据业务需要选择“编译命令”、“组织”和“选择集群”参数。</li> </ul> </li> </ul>

参数	参数说明
	<p>也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</p> <p><b>须知</b></p> <p>“编译命令”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。</p> <ul style="list-style-type: none"> <li>选择“Jar 包”、“War 包”</li> </ul> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>运行时为“Java8”时，选择“Jar 包”。</li> <li>运行时为“Tomcat8”时，选择“War 包”。</li> </ul> <ol style="list-style-type: none"> <li>选择“上传方式”： 将软件包上传至 OBS 对象存储中，请参考“帮助中心 &gt; 对象存储服务 &gt; 用户指南 &gt; 控制台指南 &gt; 入门 &gt; 上传文件”。</li> <li>（可选）设置“开启构建”参数，用于应用组件构建。 根据业务需要选择“组织”和“选择集群”参数。 也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</li> </ol>

步骤 5 完成组件创建：

- 单击“立即创建”，创建静态组件。
- 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。

组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

----结束

### 3.3.2.3 创建微服务组件

ServiceStage 提供了一套包含代码框架生成，服务注册发现，负载均衡，服务可靠性（容错熔断，限流降级）等功能的微服务框架。本节介绍基于 ServiceStage 创建微服务应用静态组件，部署组件的操作请参考[部署组件](#)。

#### 前提条件

- 只能在应用下新增组件，请先创建应用，请参考[创建应用](#)。
- 如果您基于“源码仓库”创建微服务组件，那么您首先需要创建仓库授权，请参考[仓库授权](#)。

- 如果您基于软件包创建微服务组件，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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。

## 操作步骤

**步骤 1** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

**步骤 2** 选择已经创建的应用，单击“操作”栏的“新增组件”。

**步骤 3** “配置方式”选择“自定义配置”，“选择组件类型”选择“微服务”，单击“下一步”。

**步骤 4** 选择运行时，单击“下一步”。

不同框架支持运行时有所不同，请参考[微服务组件说明](#)。

**步骤 5** 选择框架/服务网格。

框架/服务网格详情，请参考[微服务组件说明](#)。

**步骤 6** 是否将以上配置保存为模板？

- 是，勾选“将以上 3 步的配置保存为模版，以便下次使用相同的配置”，输入模板名称。执行[步骤 7](#)。
- 否，执行[步骤 7](#)。

**步骤 7** [步骤 4](#) 选择的运行时是否为“Docker”？

- 是，单击“下一步”，执行[步骤 8](#)。
- 否，单击“下一步”，执行[步骤 9](#)。

**步骤 8** 创建 Docker 组件：

- 输入“组件名称”。
- 创建组件：
  - 单击“立即创建”，创建静态组件。
  - 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。
- 操作结束。

组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

**步骤 9** 参照下表设置组件信息，参数前面带\*号的是必须设置的参数。

表3-4 组件基本信息

参数	参数说明
*组件名称	组件对应的名称。
*软件包	<ul style="list-style-type: none"> <li>选择“源码仓库”，参考<a href="#">仓库授权</a>创建授权，设置代码来源。</li> </ul>

参数	参数说明
	<ul style="list-style-type: none"> <li>选择“Jar 包”、“War 包”或“Zip 包”，需要设置“上传方式”。</li> </ul> <p>将软件包上传至 OBS 对象存储中，请参考“帮助中心 &gt; 对象存储服务 &gt; 用户指南 &gt; 控制台指南 &gt; 入门 &gt; 上传文件”。</p> <p>说明</p> <ul style="list-style-type: none"> <li>运行时为“Java8”时，选择“Jar 包”。</li> <li>运行时为“Tomcat8”时，选择“War 包”。</li> <li>运行时为“Nodejs8”、“Php7”或者“Python3”时，选择“Zip 包”。</li> </ul>
*Python 框架	<p>当步骤 4 选择的运行时是“Python3”时，需设置本参数。除了“Python3-Django”，选择其他框架均需设置“Module Name”和“Variable Name”：</p> <ul style="list-style-type: none"> <li>如果 Python 工程入口文件名为 server.py，则“Module Name”为“server”。</li> <li>如果 Python 工程入口文件 server.py 的应用函数名称为 app=get_wsgi_application()，则“Variable Name”为“app”。</li> </ul>
开启构建	<ul style="list-style-type: none"> <li>“软件包”选择“源码仓库”时，设置“开启构建”参数，用于应用组件构建。</li> </ul> <p>根据业务需要选择“编译命令”、“组织”和“选择集群”参数。</p> <p>也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</p> <p>须知</p> <p>“编译命令”选择“使用自定义命令”时：</p> <p>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。</p> <ul style="list-style-type: none"> <li>（可选）“软件包”选择“Jar 包”、“War 包”或“Zip 包”时，可以设置“开启构建”参数，用于应用组件构建。</li> </ul> <p>根据业务需要选择“组织”和“选择集群”参数。</p> <p>也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</p>

#### 步骤 10 创建组件：

- 单击“立即创建”，创建静态组件。

- 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。

组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

----结束

### 3.3.2.4 创建 Web 应用组件

本节介绍基于 ServiceStage 创建 Web 静态应用组件，部署组件的操作请参考[部署组件](#)。

#### 前提条件

1. 只能在应用下新增组件，请先创建应用，请参考[创建应用](#)。
2. 如果您基于“源码仓库”创建微服务组件，那么您首先需要创建仓库授权，请参考[仓库授权](#)。
3. 如果您基于软件包创建微服务组件，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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。

#### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

步骤 2 选择已经创建的应用，单击“操作”栏的“新增组件”。

步骤 3 “配置方式”选择“自定义配置”，组件类型选择“Web”，单击“下一步”。

步骤 4 选择运行时，单击“下一步”。

不同框架支持运行时有所不同，请参考[Web 应用组件说明](#)。

步骤 5 选择“框架/服务网格”。

步骤 6 是否将以上配置保存为模板？

- 是，勾选“将以上 3 步的配置保存为模版，以便下次使用相同的配置”，输入模板名称。执行[步骤 7](#)。
- 否，执行[步骤 7](#)。

步骤 7 [步骤 4](#)选择的运行时是否为“Docker”？

- 是，单击“下一步”，执行[步骤 8](#)。
- 否，单击“下一步”，执行[步骤 9](#)。

步骤 8 创建 Docker 组件：

1. 输入“组件名称”。
2. 创建组件：
  - 单击“立即创建”，创建静态组件。
  - 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。

## 3. 操作结束。

组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

步骤 9 参照下表设置组件信息，参数前面带\*号的是必须设置的参数。

表3-5 组件基本信息

参数	参数说明
*组件名称	组件对应的名称。
*软件包	<ul style="list-style-type: none"> <li>选择“源码仓库”，参考<a href="#">仓库授权</a>创建授权，设置代码来源。</li> <li>选择“Jar 包”、“War 包”或“Zip 包”，需要设置“上传方式”。</li> </ul> <p>将软件包上传至 OBS 对象存储中，请参考“帮助中心 &gt; 对象存储服务 &gt; 用户指南 &gt; 控制台指南 &gt; 入门 &gt; 上传文件”。</p> <p>说明</p> <ul style="list-style-type: none"> <li>运行时为“Java8”时，选择“Jar 包”。</li> <li>运行时为“Tomcat8”时，选择“War 包”。</li> <li>运行时为“Nodejs8”、“Php7”或者“Python3”时，选择“Zip 包”。</li> </ul>
*Python 框架	<p>当<a href="#">步骤 4</a>选择的运行时是“Python3”时，需设置本参数。除了“Python3-Django”，选择其他框架均需设置“Module Name”和“Variable Name”：</p> <ul style="list-style-type: none"> <li>如果 Python 工程入口文件名为 server.py，则“Module Name”为“server”。</li> <li>如果 Python 工程入口文件 server.py 的应用函数名称为 app=get_wsgi_application()，则“Variable Name”为“app”。</li> </ul>
开启构建	<ul style="list-style-type: none"> <li>“软件包”选择“源码仓库”时，设置“开启构建”参数，用于应用组件构建。</li> </ul> <p>根据业务需要选择“编译命令”、“组织”和“选择集群”参数。</p> <p>也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</p> <p>须知</p> <p>“编译命令”选择“使用自定义命令”时：</p> <p>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。</p> <ul style="list-style-type: none"> <li>（可选）“软件包”选择“Jar 包”、“War 包”或“Zip 包”</li> </ul>

参数	参数说明
	时，可以设置“开启构建”参数，用于应用组件构建。 根据业务需要选择“组织”和“选择集群”参数。 也可选择“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。新增节点标签请参考“帮助中心 > 云容器引擎 > 用户指南 > 节点管理 > 管理节点标签”。

#### 步骤 10 创建组件：

- 单击“立即创建”，创建静态组件。
- 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。

组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

----结束

### 3.3.2.5 创建通用组件

本节介绍基于 ServiceStage 创建通用静态应用组件，部署组件的操作请参考[部署组件](#)。

#### 前提条件

1. 只能在应用下新增组件，请先创建应用，请参考[创建应用](#)。
2. 如果您基于“源码仓库”创建微服务组件，那么您首先需要创建仓库授权，请参考[仓库授权](#)。
3. 如果您基于软件包创建微服务组件，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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。

#### 操作步骤

**步骤 1** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

**步骤 2** 选择已经创建的应用，单击“操作”栏的“新增组件”。

**步骤 3** “配置方式”选择“自定义配置”，组件类型选择“通用”，单击“下一步”。

**步骤 4** 选择运行时，单击“下一步”。

不同框架支持运行时有所不同，请参考[通用组件说明](#)。

**步骤 5** 是否将以上配置保存为模板？

- 是，勾选“将以上 2 步的配置保存为模版，以便下次使用相同的配置”，输入模板名称。执行[步骤 6](#)。
- 否，执行[步骤 6](#)。

**步骤 6** [步骤 4](#) 选择的运行时是否为“Docker”？

- 是，单击“下一步”，执行步骤 7。
- 否，单击“下一步”，执行步骤 8。

**步骤 7 创建 Docker 组件：**

1. 输入“组件名称”。
2. 创建组件：
  - 单击“立即创建”，创建静态组件。
  - 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。
3. 操作结束。  
组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

**步骤 8 参照下表设置组件信息，参数前面带\*号的是必须设置的参数。**

参数	参数说明
*组件名称	组件对应的名称。
*软件包	<ul style="list-style-type: none"> <li>• 选择“源码仓库”：参考<a href="#">仓库授权</a>创建授权，设置代码来源。</li> <li>• 选择“Jar 包”、“War 包”或“Zip 包”时，需设置“上传方式”： 将软件包上传至 OBS 对象存储中，请参考“帮助中心 &gt; 对象存储服务 &gt; 用户指南 &gt; 控制台指南 &gt; 入门 &gt; 上传文件”。</li> </ul> <p>说明</p> <ul style="list-style-type: none"> <li>• 运行时为“Java8”时，选择“Jar 包”。</li> <li>• 运行时为“Tomcat8”时，选择“War 包”。</li> <li>• 运行时为“Nodejs8”、“Php7”或者“Python3”时，选择“Zip 包”。</li> </ul>
*Python 框架	<p>当步骤 4 选择的运行时是“Python3”时，需设置本参数。</p> <p>除了“Python3-Django”，选择其他框架均需设置“Module Name”和“Variable Name”：</p> <ul style="list-style-type: none"> <li>• 如果 Python 工程入口文件名为 server.py，则“Module Name”为“server”。</li> <li>• 如果 Python 工程入口文件 server.py 的应用函数名称为 app=get_wsgi_application()，则“Variable Name”为“app”。</li> </ul>
开启构建	<ul style="list-style-type: none"> <li>• “软件包”选择“源码仓库”时，设置“开启构建”参数，用于应用组件构建。</li> </ul> <p>根据业务需要选择“编译命令”、“组织”和“选择集群”参数。</p> <p>也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容</p>

参数	参数说明
	<p>器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</p> <p>须知</p> <p>“编译命令”选择“使用自定义命令”时：</p> <p>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。</p> <ul style="list-style-type: none"> <li>• （可选）“软件包”选择“Jar 包”、“War 包”或“Zip 包”时，可以设置“开启构建”参数，用于应用组件构建。</li> </ul> <p>根据业务需要选择“组织”和“选择集群”参数。</p> <p>也可选择“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。新增节点标签请参考“帮助中心 &gt; 云容器引擎 &gt; 用户指南 &gt; 节点管理 &gt; 管理节点标签”。</p>

步骤 9 创建组件：

- 单击“立即创建”，创建静态组件。
- 单击“创建并部署”，进入到部署界面，详细操作请参考[部署组件](#)。

组件创建完成后，在应用“概览”页的“组件列表”可查看组件状态。

----结束

## 3.3.3 应用组件部署

### 3.3.3.1 部署方式说明

#### 云容器引擎部署

云容器引擎（Cloud Container Engine，简称 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。

如果创建的组件未开启构建，则不支持容器部署。

### 3.3.3.2 部署组件

本节介绍如何将静态组件部署到对应的环境上。

新建应用组件时，也可以选择“创建并部署”，部署操作与本节介绍步骤相同。

#### 前提条件

1. 已经创建应用组件或者正在创建应用组件并完成了静态组件配置，请参考[新建应用组件](#)。
2. 已经完成环境创建，请参考[环境管理](#)。
3. 如果您基于软件包或者镜像包部署组件，需要将软件包或者镜像包上传：

- 将软件包上传至 OBS 对象存储中，请参考“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。
  - 将镜像包上传至镜像仓库，请参考[上传镜像](#)。
4. 配置 AK/SK，请参考“帮助中心 > 微服务云应用平台 > 开发指南 > 微服开发指南 > 附录 > AK/SK 认证方式排查与切换指导”。

## 操作步骤

- 步骤 1** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。
- 步骤 2** 选择已经创建的应用，单击应用名称，进入应用“概览”页。
- 步骤 3** 在“组件列表”选择已经创建的组件，单击“操作”栏“部署”。
- 步骤 4** 设置基本配置，其中带“\*”标志的参数为必填参数。

参数	参数说明
*环境	选择已创建的环境。 说明 只能选择相同企业项目下的环境部署。
*部署版本	组件版本号，例如：1.0.0。
描述	组件的描述信息。
*部署系统	支持云容器引擎。 详情请参见 <a href="#">部署方式说明</a> 。
*基础资源	会自动加载所选环境包含的基础资源，根据实际业务需要进行选择。
*实例数量	组件可以有一个或多个实例，用户可以设置具体实例个数。 设置多个实例主要用于实现高可靠性，当某个实例故障时，应用组件还能正常运行。 说明 <ul style="list-style-type: none"> <li>• 组件实例个数由您所选择的“基础资源”的数目决定。</li> <li>• 当引擎规格限定的微服务实例数配额即将用完时，为了保证最大程度可用，引擎允许超出剩余配额的部署组件操作同时成功。请尽快扩容引擎，避免后续部署组件失败。</li> </ul>
*资源配额	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 <a href="#">资源限制指南</a> 。 可以根据需要自定义“CPU 配额”和“内存配额”。 说明 <ul style="list-style-type: none"> <li>• “组件类型”为“通用”且运行时为“Docker”的组件，部署时无此参数。</li> </ul>
组件状态	根据需要设置组件状态。

参数	参数说明
	说明 “组件类型”为“通用”且运行时为“Docker”的组件，选择“云容器引擎”部署时可设置此参数。

步骤 5 单击“下一步 组件配置”，配置组件。

- “组件类型”为“通用”且运行时为“Docker”的组件，执行以下操作：
  - a. 选择镜像。支持多容器，可以单击“添加容器”增加镜像。
  - b. 设置“镜像版本”。
  - c. 输入“容器名称”。
  - d. （可选）设置“资源配额”。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”。
  - e. （可选）设置“高级设置”
    - 展开“高级设置 > 组件配置”，可以设置“环境变量”，请参考[设置应用组件环境变量](#)。
    - 展开“高级设置 > 部署配置”：
      - 设置“启动命令”、“生命周期”，请参考[设置应用生命周期](#)。
      - 设置“数据存储”，请参考[设置数据存储](#)。
    - 展开“高级设置 > 运维监控”：
      - 设置“日志采集”，请参考[配置应用日志策略](#)。
      - 设置“健康检查”，请参考[设置应用健康检查](#)。
  - f. （可选）开启“公网访问”
    - i. 设置“公网 ELB”
 

选择环境中已存在的弹性负载均衡(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，详情请参考“帮助中心 > 弹性负载均衡 > 用户指南 > 负载均衡器 > 创建负载均衡器”。

#### 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。
  - ii. （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“帮助中心 > 弹性负载均衡 > 证书管理 > 创建证书”。

iii. 设置“域名”

在输入框中输入自定义域名。详情请参考[配置域名映射](#)。

iv. 设置“监听端口”

设置应用进程的监听端口。

g. （可选）设置“数据库”

选择“分布式会话”，请参考[设置分布式会话](#)。

h. （可选）设置“时区”

修改容器节点的时区，默认和容器节点所在 Region 的时区一致。

i. （可选）设置“调度策略”，请参考[设置应用组件实例调度策略](#)。

j. （可选）设置“升级策略”，请参考[设置应用组件实例升级策略](#)。

k. （可选）设置“性能管理”，请参考[设置应用性能管理](#)。

“部署系统”选择“云容器引擎”且环境已部署应用性能管理 APM 服务时，支持设置。

l. （可选）设置“自定义指标监控”，请参考[设置应用组件自定义指标监控](#)。

“部署系统”选择“云容器引擎”且环境已部署应用性能管理 APM 服务时，支持设置。

• 其他类型的组件，且[步骤 4](#)中的“部署系统”为“云容器引擎”执行以下操作：

a. 设置“镜像”

- 应用来源为软件包、源码仓库，会加载已经配置的组件静态信息。
- 组件运行时为 Docker，需要从 SWR 镜像仓库选择镜像包。

b. （可选）开启“公网访问”

i. 设置“公网 ELB”

选择所选环境中已经存在的弹性负载均衡(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“帮助中心 > 弹性负载均衡 > 用户指南 > 负载均衡器 > 创建负载均衡器”。

## 📖 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。

ii. （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“帮助中心 > 弹性负载均衡 > 证书管理 > 创建证书”。

iii. 设置“域名”

在输入框中输入自定义域名。详情请参考[配置域名映射](#)。

iv. （可选）设置“监听端口”

应用进程的监听端口，对于 Tomcat8 运行时，默认为 8080，也支持自定义。

c. （可选）设置“JVM”

组件运行时为“Java8”、“Tomcat8”时需要设置。

输入 JVM 参数，如-Xms256m -Xmx1024m，多个参数以空格间隔，不填则使用默认值。

d. （可选）设置“Tomcat 配置”

组件运行时为“Tomcat8”时需要配置。

i. 勾选“配置参数”，弹出“Tomcat 配置”对话框。

ii. 单击“使用示例模板”，根据业务要求编辑模板文件。

iii. 单击“确定”。

e. （可选）设置“微服务引擎”

微服务类型组件需要设置该参数。

默认选择环境中添加的微服务引擎，创建微服务引擎请参考[创建微服务引擎专享版](#)。

f. （可选）设置“数据库”

选择“分布式会话”，请参考[设置分布式会话](#)。

g. （可选）设置“时区”

修改容器的时区，默认和容器节点所在 Region 的时区一致。

h. （可选）设置“高级设置”

■ 展开“高级设置 > 组件配置”，可以设置“环境变量”，请参考[设置应用组件环境变量](#)。

■ 展开“高级设置 > 部署配置”：

○ 设置“启动命令”、“生命周期”，请参考[设置应用生命周期](#)。

○ 设置“数据存储”，请参考[设置数据存储](#)。

○ 设置“调度策略”，请参考[设置应用组件实例调度策略](#)。

○ 设置“升级策略”，请参考[设置应用组件实例升级策略](#)。

■ 展开“高级设置 > 运维监控”：

○ 设置“日志采集”，请参考[配置应用日志策略](#)。

○ 设置“健康检查”，请参考[设置应用健康检查](#)。

○ 设置“性能管理”，请参考[设置应用性能管理](#)。

“部署系统”选择“云容器引擎”且环境已部署应用性能管理 APM 服务时，支持设置。

- 设置“自定义指标监控”，请参考[设置应用组件自定义指标监控](#)。  
“部署系统”选择“云容器引擎”且环境已部署应用性能管理 APM 服务时，支持设置。

步骤 6 单击“下一步 规格确认”，确认规格无误后，单击“部署”。

组件部署完成后，在应用“概览”页的“环境视图”可查看组件实例状态。

----结束

### 3.3.4 应用组件管理

组件创建或部署完成后，可以执行如下管理操作：

- [查看应用组件](#)：查看应用下已创建的组件列表。
- [部署组件](#)：部署已创建的静态组件。
- [更新组件来源](#)：更新组件的源码/软件包、版本以及环境配置。运行时为 Docker 的组件不支持本操作。
- [删除组件](#)：删除已创建组件。
- [创建应用组件流水线](#)：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。
- [查看应用组件构建](#)：查看应用组件构建工程的状态。
- [维护组件实例](#)：维护已部署的应用组件实例。
- [管理组件实例访问方式](#)：设置组件实例的访问方式。
- [查看组件运行日志](#)：支持查看、搜索、导出日志记录，用于排查和解决组件运行中出现的问题。

#### 查看应用组件

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

步骤 2 选择已经创建的应用，单击应用名称，进入应用“概览”页。

步骤 3 单击“组件列表”页签，查看该应用下已创建的组件列表。

----结束

#### 部署组件

部署组件，请参考[部署组件](#)。

#### 更新组件来源

组件创建后，可以更新组件的源码/软件包、版本以及环境配置。

运行时为 Docker 的组件不支持本操作。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

步骤 2 选择已经创建的应用，单击应用名称，进入应用“概览”页。

步骤 3 单击“组件列表”页签：

- 更新单个组件来源：选择已经创建的单个组件，在“操作”列单击“更新来源”。
- 批量更新组件来源：勾选已经创建的多个组件，单击“更新组件来源”。

步骤 4 设置“软件包”。

- 选择“源码仓库”：参考[仓库授权](#)创建授权，设置代码来源。

选择“软件包”：

单击“更换软件包”，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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 上传文件”。

步骤 5 设置“升级后版本”和“选择需升级的环境”。

步骤 6 单击“确定”，完成操作。

----结束

## 删除组件

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

步骤 2 选择已经创建的应用，单击应用名称，进入应用“概览”页。

步骤 3 单击“组件列表”页签：

- 单个删除  
选择待删除的组件，在“操作”列单击“删除”，在弹出的提示框单击“确定”。
- 批量删除  
勾选待删除的多个组件，单击“批量删除组件”，在弹出的提示框单击“确定”。

----结束

## 创建应用组件流水线

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

步骤 2 选择已经创建的应用，单击应用名称，进入应用“概览”。

步骤 3 在“组件列表”页签，单击组件名称，进入组件“概览”页。

步骤 4 选择“流水线 > 创建流水线”，创建流水线。请参考[流水线管理](#)。

### 📖 说明

- 不支持为运行时为 Docker 的组件创建流水线。

----结束

## 查看应用组件构建

- 步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。
- 步骤 2 选择已经创建的应用，单击应用名称，进入应用“概览”。
- 步骤 3 在“组件列表”页签，单击组件名称，进入组件“概览”页。
- 步骤 4 单击“构建”页签，可以查看应用组件构建工程的状态，详情请参考[创建源码构建任务](#)。

----结束

## 维护组件实例

- 步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。
- 步骤 2 选择已经创建的应用，单击应用名称，进入应用“概览”。
- 步骤 3 在“环境视图”页签下，选择“环境”：
  - 可以查看各个环境下该应用组件部署情况。
  - （可选）选择微服务类型的应用组件版本，单击“微服务管理”，进入微服务控制台进行服务治理。详情请参考[服务治理](#)。
  - 选择应用组件版本，单击“运维管理”，可以进入组件实例“概览”页，查看组件实例详情。
  - 选择应用组件版本，单击“操作”，可以选择进行组件升级、伸缩、事件查看、启动/停止、重启、回退、删除等运维操作。详情请参考[应用运维](#)。
  - 勾选“全选”或者勾选对应的应用组件，单击“升级组件”，对组件进行版本号、软件包/镜像包的变更操作。

----结束

## 管理组件实例访问方式

- 步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。
- 步骤 2 单击已经创建的应用名称，进入应用“概览”页。
- 步骤 3 在“组件列表”单击已经创建的组件名称，进入组件“概览”页。  
在对应环境卡片可以查看组件版本。
- 步骤 4 选择状态为“运行中”的组件页，单击“设置”，进入“访问方式”页，单击“添加服务”。
- 步骤 5 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	设置访问服务的名称。
访问方式	设置访问服务的方式，支持：

参数	参数说明
	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> <li>• 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。</li> </ul>
VPC 内网负载均衡	“访问方式”选择“VPC 内网访问”时有效。
*访问类型	<ul style="list-style-type: none"> <li>• “访问方式”选择“VPC 内网访问”且开启“VPC 内网负载均衡”后有效。</li> <li>• “访问方式”选择“公网访问”后有效。</li> </ul>
服务亲和	“访问方式”选择“VPC 内网访问”或者“公网访问”后有效。
端口映射	设置访问服务的“协议”、“容器端口”、“访问端口”。

步骤 6 单击“确定”。

----结束

## 查看组件运行日志

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。

步骤 2 选择待查看运行日志的应用组件所在的应用，单击应用名称，进入应用“概览”。

步骤 3 在“环境视图”页签下，选择待查看运行日志的应用组件所在的“环境”。

步骤 4 单击待查看运行日志的组件名称，进入“概览”页。

步骤 5 单击“运行日志”，参考下表执行查看组件运行日志操作。

操作名称	操作说明
查看日志	<ol style="list-style-type: none"> <li>1. 选择待查看日志的实例名称。</li> <li>2. 选择待查看的日志文件名称。</li> <li>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。</li> <li>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。</li> </ol>
导出日志	<ol style="list-style-type: none"> <li>1. 单击“导出文件”。</li> <li>2. 选择需要导出的日志记录条数。</li> <li>3. 打开已导出到本地的“log.txt”文件，查看导出的日志记录。</li> </ol>
查看 AOM	单击“查看 AOM 日志”，可在应用运维管理 AOM 中查看组件运行

操作名称	操作说明
日志	日志。 具体操作，请参考“帮助中心 > 应用运维管理 > 用户指南 > 日志管理”。

----结束

## 3.3.5 应用高级设置

### 3.3.5.1 设置应用组件环境变量

环境变量是应用运行环境中设定的一个变量，可以在应用组件部署后修改，为应用提供极大的灵活性。

本节为您介绍应用组件部署方式为云容器引擎部署时环境变量的配置方法。

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。

如果您在该应用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。

如果您需要对应用添加环境变量，请参考[添加应用环境变量](#)。

## 云容器引擎部署方式

部署应用组件时，在“基本配置”界面，“部署系统”选择“云容器引擎”时，请参考本方法添加环境变量。

**步骤 1** 在“组件配置”界面，展开“高级设置> 组件配置”。

**步骤 2** 参考[表 3-6](#) 添加环境变量。

当前支持四种类型。

表3-6 环境变量类型

环境变量类型	操作步骤
手动添加	<ol style="list-style-type: none"> <li>1. 单击“添加环境变量”，选择“手动添加”。</li> <li>2. 输入新增环境变量的“变量名称”和“变量/变量引用”。</li> </ol>
密钥导入	<ol style="list-style-type: none"> <li>1. 创建密钥，请参考<a href="#">创建密钥</a>。</li> <li>2. 单击“添加环境变量”，选择“密钥导入”。</li> <li>3. 输入“变量名称”。</li> <li>4. 在“变量/变量引用”的下拉框中选择对应的密钥名称。</li> </ol>
配置项导入	<ol style="list-style-type: none"> <li>1. 创建配置项，请参考<a href="#">创建配置项</a>。</li> <li>2. 单击“添加环境变量”，选择“配置项导入”。</li> <li>3. 输入“变量名称”。</li> </ol>

环境变量类型	操作步骤
	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	单击“从文件导入”，选择本地配置文件。 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： <pre>{"key1":"value1","key2":"value2"}</pre>

----结束

### 3.3.5.2 设置应用生命周期

部署应用组件时，在“基本配置”界面，“部署系统”选择“云容器引擎”的应用组件，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** 部署应用组件时，在“组件配置”界面，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 3-7 所示。

表3-7 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 3-8 所示，选择其中一个参数即可。

表3-8 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 <code>Command Args[1] Args[2]...</code>（<code>Command</code> 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：<code>/install.sh install_agent</code>。 这条命令表示组件部署成功后将执行 <code>install_agent</code> 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"> <li>• 路径：请求的 URL 路径，可选项。</li> <li>• 端口：请求的端口，必选项。</li> <li>• 主机地址：请求的 IP 地址，可选项，默认是应用所在的节点 IP。</li> </ul>

----结束

### 3.3.5.3 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。

部署应用组件时，在“基本配置”界面，“部署系统”选择“云容器引擎”的应用组件，支持设置数据存储。

### 使用场景

表3-9 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"> <li>• 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。</li> <li>• 高 I/O：后端存储由 SAS 存储介质提供，适用于性能相对较高、</li> </ul>

项目	场景
	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。 <ul style="list-style-type: none"> <li>• 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）。</li> </ul>
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"> <li>• 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。</li> <li>• 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。</li> </ul>
主机路径挂载	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。 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。
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 部署应用组件时，在“组件配置”界面，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 3-10 设置参数。

表3-10 云硬盘存储

参数	说明
云存储类型	选择“云硬盘”。

参数	说明
	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。
分配方式	<ul style="list-style-type: none"> <li>• 使用已有存储 选择已创建的存储。</li> <li>• 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"> <li>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。</li> <li>2. 选择存储子类型。                                     <ul style="list-style-type: none"> <li>高 IO：指由 SAS 存储介质构成的云硬盘。</li> <li>普通 IO：指由 SATA 存储介质构成的云硬盘。</li> <li>超高 IO：指由 SSD 存储介质构成的云硬盘。</li> </ul> </li> <li>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。</li> </ol> </li> </ul>
添加容器挂载	<ol style="list-style-type: none"> <li>1. 设置“挂载路径”：输入数据卷挂载到应用上的路径。 须知                             <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul> </li> <li>2. 设置“权限”。                             <ul style="list-style-type: none"> <li>• 只读：只能读应用路径中的数据卷。</li> <li>• 读写：可修改应用路径中的数据卷，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li> </ul> </li> </ol>

步骤 3 单击“确定”。

----结束

## 文件存储

步骤 1 部署应用组件时，在“组件配置”界面，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 3-11 设置参数。

表3-11 文件存储

参数	说明
----	----

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"> <li>使用已有存储 选择已创建的存储。</li> <li>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"> <li>选择存储子类型。 文件存储子类型为 NFS。</li> <li>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。</li> </ol> </li> </ul>
添加容器挂载	<ol style="list-style-type: none"> <li>设置“子路径”、“挂载路径”：输入数据卷挂载到应用上的路径。 须知                             <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul> </li> <li>设置“权限”。                             <ul style="list-style-type: none"> <li>只读：只能读应用路径中的数据卷。</li> <li>读写：可修改应用路径中的数据卷，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li> </ul> </li> </ol>

步骤 3 单击“确定”。

----结束

## 对象存储

步骤 1 部署应用组件时，在“组件配置”界面，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 3-12 设置参数。

表3-12 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

参数	说明
	档等场景。
分配方式	<ul style="list-style-type: none"> <li>使用已有存储 选择已创建的存储。</li> <li>自动分配存储 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。</li> </ul>
添加容器挂载	1. 设置“挂载路径”：输入数据卷挂载到应用上的路径。 须知 <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul> 2. 设置“权限”。 <ul style="list-style-type: none"> <li>只读：只能读应用路径中的数据卷。</li> <li>读写：可修改应用路径中的数据卷，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li> </ul>

步骤 3 单击“确定”。

----结束

## 主机路径挂载

表示在应用组件上挂载宿主机上的文件或目录。

步骤 1 部署应用组件时，在“组件配置”界面，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 3-13 设置参数。

表3-13 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。
容器挂载	1. 设置“子路径”、“挂载路径”：输入数据卷挂载到应用上的路径。 须知 <ul style="list-style-type: none"> <li>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</li> </ul>

参数	说明
	<p>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。</p> <ul style="list-style-type: none"> <li>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主机高危文件被破坏。</li> </ul> <p>2. 设置“权限”。</p> <ul style="list-style-type: none"> <li>只读：只能读应用路径中的数据卷。</li> <li>读写：可修改应用路径中的数据卷，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li> </ul>

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 部署应用组件时，在“组件配置”界面，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 3-14 设置参数。

表3-14 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"> <li>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。</li> <li>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。</li> </ul>
容器挂载	<p>1. 设置“子路径”、“挂载路径”：输入数据卷挂载到应用上的路径。</p> <p>须知</p> <ul style="list-style-type: none"> <li>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。</li> <li>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主机高危文件被破坏。</li> </ul> <p>2. 设置“权限”。</p> <ul style="list-style-type: none"> <li>只读：只能读应用路径中的数据卷。</li> <li>读写：可修改应用路径中的数据卷，应用迁移时新写入的数</li> </ul>

参数	说明
	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 部署应用组件时，在“组件配置”界面，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 3-15 设置参数。

表3-15 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 <a href="#">创建配置项</a> 。
容器挂载	1. 设置“子路径”、“挂载路径”：输入数据卷挂载到应用上的路径。 须知 <ul style="list-style-type: none"> <li>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。</li> <li>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主机高危文件被破坏。</li> </ul> 2. 设置“权限”。 <ul style="list-style-type: none"> <li>只读：只能读应用路径中的数据卷。</li> <li>读写：可修改应用路径中的数据卷，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li> </ul>

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 部署应用组件时，在“组件配置”界面，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 3-16 设置参数。

表3-16 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 <a href="#">创建密钥</a> 。
容器挂载	1. 设置“子路径”、“挂载路径”：输入数据卷挂载到应用上的路径。 须知 <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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主机高危文件被破坏。</li> </ul> 2. 设置“权限”。 <ul style="list-style-type: none"> <li>• 只读：只能读应用路径中的数据卷。</li> <li>• 读写：可修改应用路径中的数据卷，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li> </ul>

步骤 3 单击“确定”。

----结束

### 3.3.5.4 设置分布式会话

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，可以在数据库设置中绑定分布式会话。

#### 前提条件

设置分布式会话前，需创建分布式会话，请参考“帮助中心 > 分布式缓存服务 > 快速入门 > 创建缓存实例”。

## 操作步骤

- 步骤 1 部署应用组件时，在“组件配置”界面，勾选“分布式会话”。
- 步骤 2 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。
- 结束

### 3.3.6 应用组件构建

ServiceStage 需要先将软件包构建成镜像或将源码构建成软件包，然后再部署。所以创建应用组件运行时设置应用来源后，ServiceStage 会为应用组件生成一个构建工程。

#### 📖 说明

运行时为 Docker 的组件不支持构建。

#### 查看应用组件构建

- 步骤 1 登录 ServiceStage 控制台，在“应用管理 > 应用列表”界面，单击应用名称，进入应用详情“概览”页。
- 步骤 2 在“组件列表”页签，单击组件名称，进入组件“概览”页。
- 步骤 3 单击“构建”页签，可以查看应用组件构建工程的状态。
- 结束

#### 构建工程维护操作说明

表3-17 维护操作

操作	操作说明
编辑	单击“编辑”，进入到构建工程配置界面，请参考 <a href="#">编辑软件包构建工程</a> 或 <a href="#">编辑源码构建工程</a> 。
立即构建	单击“立即构建”，启动构建工程。
查看详情/构建历史	<ul style="list-style-type: none"> <li>单击“查看其它构建记录”，在“构建记录”下查看构建历史。</li> <li>单击“日志”，可以查看本次构建记录日志。</li> <li>单击“代码检查”，可以查看代码检查概况及代码检查详细信息。</li> </ul> <p>目前支持的代码检查插件有：checkstyle、findbugs、pmd。</p> <p>说明 仅 Maven 构建工程支持代码检查。</p>

## 编辑软件包构建工程

步骤 1 登录 ServiceStage 控制台，在“应用管理 > 应用列表”界面，单击应用名称，进入应用详情“概览”页。

步骤 2 在“组件列表”页签，单击组件名称，进入组件“概览”页。

步骤 3 选择“构建”页签，单击“编辑”，进入构建工程配置界面。

步骤 4（可选）输入“描述”信息。

步骤 5 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 6 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 7 选择“镜像归档地址”。

步骤 8 选择“构建集群”，使用选择的集群进行构建任务。

步骤 9（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。

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

## 编辑源码构建工程

步骤 1 登录 ServiceStage 控制台，在“应用管理 > 应用列表”界面，单击应用名称，进入应用“概览”页。

步骤 2 在“组件列表”页签，单击组件名称，进入组件“概览”页。

步骤 3 选择“构建”页签，单击“编辑”，进入构建工程基本信息配置界面。

1. （可选）输入“描述”信息。
2. 选择“构建集群”，使用选择的集群进行构建任务。

3. （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。
4. 单击“下一步”。

步骤 4 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 5 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

### 3.3.7 应用组件流水线

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。

#### 说明

- 不支持为运行时为 Docker 的组件创建流水线。

### 创建流水线

步骤 1 登录 ServiceStage 控制台，在“应用管理 > 应用列表”界面，单击应用名称，进入应用详情“概览”页。

步骤 2 在“组件列表”页签，单击组件名称，进入组件“概览”页。

步骤 3 选择“流水线”页签，单击“创建流水线”。

步骤 4 填写流水线基本信息。

1. 输入“流水线名称”。
2. （可选）输入流水线“描述”。

步骤 5 设置流水线。

1. 添加构建任务  
自动加载该组件的构建任务。
2. 添加部署任务

单击“添加环境”，自动加载已部署的组件。

### 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 6 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

## 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，进入应用组件详情页“流水线”页签

步骤 2 选择流水线，单击“克隆”。

步骤 3 ServiceStage 自动加载克隆流水线的配置信息，可以在根据需要修改。

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升级，应用组件升级后的维护操作请参考[应用运维](#)。

## 3.3.8 应用配置管理

### 3.3.8.1 创建密钥

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。

#### 前提条件

- 已创建需要使用密钥的 CCE 集群。  
创建 CCE 集群，请参考“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 创建 CCE 集群”。
- 已创建密钥所在命名空间，请参考“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 命名空间”。

#### 创建密钥

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用配置 > 密钥”。

步骤 2 单击“创建密钥”。

步骤 3 ServiceStage 支持“可视化”和“YAML”两种方式来创建密钥。

- 方式一：可视化。参照下表设置基本信息，其中带“\*”标志的参数为必填参数。

表3-18 基本信息说明

参数	参数说明
*密钥名称	新建的密钥的名称，同一个命名空间内命名必须唯一。
*所在集群	使用新建密钥的集群。 单击“创建集群”，可以新建集群。
*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。 单击“创建命名空间”，可以新建命名空间。
*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"> <li>• <b>Opaque</b>：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。</li> <li>• <b>kubernetes.io/dockerconfigjson</b>：存放拉取私有仓库镜像所需的认证信息。</li> <li>• <b>IngressTLS</b>：存放 7 层负载均衡服务所需的证书。</li> <li>• 其他：若需要创建其他类型的密钥，请手动输入密钥类型。</li> </ul>
*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"> <li>• 当密钥为 Opaque 类型时，输入键、值。其中“值”必须使用 Base64 编码，Base64 编码方法请参考 <a href="#">Base64 编码</a>。 单击“添加更多密钥数据”，可以增加密钥数据。</li> <li>• 当密钥为 kubernetes.io/dockerconfigjson 类型时，输入“镜像仓库地址”、“用户名”和“密码”。</li> <li>• 当密钥为 IngressTLS 类型时，上传“证书文件”和“私钥文件”。</li> <li>• 当密钥为其他类型时，输入“密钥类型”及对应的“键”、“值”。</li> </ul>
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"> <li>1. 单击“添加标签”。</li> <li>2. 输入键、值。</li> </ol>

- 方式二：YAML。

#### 📖 说明

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

- a. 在“所在集群”下拉框中，选择相应的集群。
- b. （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”。
- c. 在“编排内容”中写作或者修改上传的 Secret 资源文件。

步骤 4 配置完成后，单击“创建”。

密钥列表中会出现新创建的密钥。

----结束

## 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 secret.yaml 内容如下。其中 Value 需要用 Base64，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==为编码后内容。

## 管理密钥

操作	说明
更新密钥	1. 单击目标密钥“操作”列的“更新”。 2. 根据表 3-18 更改信息。 3. 单击“更新密钥”。
删除密钥	单击目标密钥“操作”列的“删除”，根据系统提示删除配置。
批量删除密钥	1. 勾选需要删除的密钥。 2. 单击页面左上角的“删除”，根据系统提示删除密钥。
查看密钥	单击目标密钥“操作”列的“查看 YAML”，查看密钥 yaml 格式文件内容。

### 说明

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。

### 3.3.8.2 创建配置项

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。

### 前提条件

- 已创建需要使用密钥的 CCE 集群。  
创建 CCE 集群，请参考“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 创建 CCE 集群”。
- 已创建配置项所在命名空间，请参考“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 命名空间”。

### 创建配置项

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用配置 > 配置项”。

步骤 2 单击“创建配置项”。

步骤 3 ServiceStage 支持“可视化”和“YAML”两种方式来创建配置项。

- 方式一：可视化。参照下表设置新增配置参数，其中带“\*”标志的参数为必填参数。

表3-19 新建配置参数说明

参数	参数说明
*配置名称	新建的配置名称，同一个命名空间里命名必须唯一。
*所属集群	使用新建配置的集群。
*命名空间	新建配置所在的命名空间。若不选择，默认配置为 default。
描述	配置项的描述信息。
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 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。 1. 单击“添加标签”。 2. 输入键、值。

- 方式二：YAML。

#### 说明

若需要通过上传文件的方式创建资源，请确保资源描述文件已创建。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”，选择已创建的 ConfigMap 类型资源文件后，单击“打开”。
- 在“编排内容”中写作或者修改 ConfigMap 资源文件。

步骤 4 配置完成后，单击“创建”。

应用配置列表中会出现新创建的应用配置。

----结束

## 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 2MB。

文件名称为 configmap.yaml，配置示例如下：

```
apiVersion: v1
kind: ConfigMap
```

```

metadata:
  name: test-configmap
data:
  data-1: value-1
  data-2: value-2
    
```

## 管理配置项

操作	说明
更新配置项	<ol style="list-style-type: none"> <li>1. 单击目标配置项“操作”列的“更新”。</li> <li>2. 根据表 3-19 更改信息。</li> <li>3. 单击“更新配置”。</li> </ol>
删除配置项	单击目标配置项“操作”列的“删除”，根据系统提示删除配置。
批量删除配置项	<ol style="list-style-type: none"> <li>1. 勾选需要删除的配置项。</li> <li>2. 单击页面左上角的“删除配置项”，根据系统提示删除配置项。</li> </ol>
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。

### 说明

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。

## 3.4 环境管理

### 3.4.1 环境概述

环境是用于应用组件部署和运行的计算、存储、网络等基础资源的集合。ServiceStage 把相同 VPC 下的计算资源（如云容器引擎 CCE）、网络资源（如弹性负载均衡 ELB、弹性公网 EIP 等）和中间件（如分布式缓存 DCS、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。

同一个项目下，最多支持创建 300 个环境。

### 3.4.2 创建环境

在部署应用组件之前，您需要先创建环境。

### 操作步骤

步骤 1 登录 ServiceStage 控制台，进入“环境管理”，单击“创建环境”。

步骤 2 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。 开通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。
*虚拟私有云 (VPC)	选择环境资源所在 VPC。 新建 VPC，请参考“帮助中心 > 虚拟私有云 > 用户指南 > 虚拟私有云和子网 > 虚拟私有云 > 创建虚拟私有云和子网”。 说明 <ul style="list-style-type: none"> <li>• 选定 VPC 后，会加载该 VPC 下的资源供选择，不在该 VPC 下的资源无法选择。</li> <li>• 环境创建完成后，执行修改环境操作时，不支持修改 VPC。</li> </ul>
*环境类型	根据应用组件部署方式，选择环境类型。 <ul style="list-style-type: none"> <li>• <b>Kubernetes:</b> 适用于容器部署方式（云容器引擎部署），应用组件以容器镜像方式通过 Kubernetes 调度部署。</li> </ul>

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。

----结束

### 3.4.3 纳管资源

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

#### 操作步骤

步骤 1 登录 ServcieStage 控制台，进入“环境管理”，单击待操作环境名称。

步骤 2 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 3 在弹出的对话框中，勾选需要纳管的资源，单击“确定”。

### 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。

处理方法，请参考[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？](#)。

----结束

## 3.4.4 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，进入“环境管理”，单击待操作环境名称。

步骤 2 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 3 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：选择待移除的资源，在“操作”列单击“移除”。

----结束

## 3.4.5 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，进入“环境管理”。

步骤 2 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 编辑下表所示环境信息。

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

参数	参数说明
	资源按项目管理，以及项目内的资源管理、成员管理。 开通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。

步骤 4 单击“保存”，完成环境修改。

----结束

## 3.4.6 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应用组件或者部署的应用组件已经被删除。具体操作请参考[应用组件管理](#)。
- 删除环境，不会删除环境中的资源。

### 操作步骤

步骤 1 登录 ServiceStage 控制台，进入“环境管理”。

步骤 2 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

## 3.5 应用运维

### 3.5.1 维护应用组件实例

#### 伸缩应用组件

您可以根据业务需求自行定义伸缩策略，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。具体操作请参考[设置应用组件实例伸缩策略](#)。

## 启停应用组件

应用组件部署以后，可以根据需要启动或者停止。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用组件“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 进行启停操作。

- 单击待停止应用组件卡片上的“操作 > 停止”，停止状态为“运行中”、“未就绪”的应用组件。
- 单击待启动应用组件卡片上的“操作 > 启动”，启动状态为“停止”的应用组件。
- 单击待重启应用组件卡片上的“操作 > 重启”，重启状态为“运行中”、“未就绪”的应用组件。

----结束

## 升级应用组件

应用组件部署以后，可以根据需要重新部署软件包，修改应用组件配置。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用组件“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可查看所有已经部署在该环境下的应用组件。

步骤 4 在待升级应用组件卡片上，选择“操作 > 升级”。

步骤 5 请根据业务需求更新配置和版本。

### 须知

如果组件版本号未修改，发生故障时将无法回退到历史版本。

步骤 6 应用高级设置请参考[应用高级设置](#)。

步骤 7 配置完成后，单击“重新部署”。

步骤 8 单击“确定”，等待组件升级完成。

----结束

## 查看实例详情

应用组件部署以后，可以在组件实例详情页查看概览、实例列表、访问方式等信息。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用组件“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页，可查看实例详情。

----结束

## 查询应用日志

应用组件启动后，可以查询应用组件日志，了解应用组件运行情况。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。可查看所有应用。

步骤 2 单击应用名称，进入到应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页。

步骤 5 单击“运行日志”，查看应用组件日志。

- 选择实例名称、指定日志文件名称和时间粒度，查看指定实例在特定时间段的日志。
- 在输入框输入内容，可对日志进行搜索。

----结束

## 回退应用组件

应用组件升级后，可以进行回退操作，将应用组件回退至目标版本。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用组件“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 在待回退应用组件卡片上，选择“操作 > 回退”。

步骤 5 选择要回退的目标版本，在“操作”列单击“回退到该版本”。

----结束

## 删除应用组件

---

### 须知

删除操作无法恢复，请谨慎操作。

---

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用组件“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 在待删除应用组件卡片上，选择“操作 > 删除”。

根据提示删除单个应用组件。

----结束

## 3.5.2 添加应用组件实例标签

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3-2 标签案例



### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页。

步骤 5 单击“标签管理”。

步骤 6 单击“添加标签”，输入“键”、“值”，单击“保存”。

#### 说明

- 新建的标签“键”名称不能和已有标签重复。
- 状态异常或者虚机部署的应用组件实例，不支持添加标签。

----结束

### 3.5.3 配置域名映射

对于开启了公网访问的应用组件，除了在 ServiceStage 定义域名外，还需在域名服务提供商处进行域名映射。

#### 前提条件

- 对于已创建的应用，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。
- 已获取应用组件绑定的 ELB 的弹性公网 IP。

#### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页。

步骤 5 设置域名：

1. 选择“访问方式 > 设置域名”，输入已获取的“应用域名”。
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“帮助中心 > 弹性负载均衡 > 证书管理 > 创建证书”。

步骤 6 在域名提供者处进行域名映射配置。

----结束

### 3.5.4 设置资源监控告警阈值

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时，可在日常运维中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，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页。

步骤 5 选择“阈值告警 > 阈值规则设置”，参考表 3-20 设置阈值规则参数信息。其中带“\*”标志的参数为必填参数。

表3-20 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（>=、<=、>、<）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"> <li>• 选择“是”（推荐选择），发送邮件或短信通知用户。</li> <li>• 选择“否”，不会发送邮件或短信通知用户。</li> </ul>

步骤 6 单击“确定”。

----结束

## 管理阈值告警

阈值规则创建完成后，您还可以按照[表 3-21](#) 中的步骤来管理阈值告警。

表3-21 相关操作

操作	说明
修改阈值告警	<p>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。</p> <ol style="list-style-type: none"> <li>1. 单击阈值告警列表“操作”列的“修改”。</li> <li>2. 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。</li> <li>3. 单击“修改”。</li> </ol>
删除阈值告警	<p>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。</p> <ol style="list-style-type: none"> <li>1. 删除阈值规则。 <ul style="list-style-type: none"> <li>• 删除单个阈值：单击阈值规则列表“操作”列的“删除”。</li> <li>• 删除单个或多个阈值规则：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”。</li> </ul> </li> <li>2. 在弹出的对话框中单击“确定”。</li> </ol>
搜索阈值告警	<ol style="list-style-type: none"> <li>1. 在下拉列表框中选择时间段。</li> <li>2. 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。</li> <li>3. 单击，或在键盘上按“Enter”键。</li> </ol> <p>也可单击“高级搜索”设置搜索条件，单击“搜索”进行查询。</p>
查看阈值告警	<p>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。</p> <p>在阈值告警列表中查看该告警。</p>
查看数据不足事件	<p>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。</p> <p>可单击“事件”页签，在事件列表中查看该事件。</p>

### 3.5.5 设置应用组件实例伸缩策略

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。

- 设置弹性伸缩策略

弹性伸缩，主要面向无状态工作负载、基于指标（CPU 利用率、内存利用率）或周期（每天、每周、每月或每年的具体时间点）进行弹性扩缩容。

### 📖 说明

- 仅 1.15 及以下版本 CCE 集群支持弹性伸缩。

- 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，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击的应用组件名称，进入应用组件实例“概览”页。

步骤 5 单击“伸缩”：

- 设置“缩容时间窗”：单击 ，修改参数，单击  生效。
- 设置“最大无效实例数”：单击 ，输入允许的最大无效实例个数；或者勾选“百分比”，输入允许的最大无效实例占比。单击“保存”。

单击“查看组件详情”，可看到实例在“升级/回滚”中。待状态为“运行中”时，表示已成功完成实例伸缩操作。

----结束

## 设置手动伸缩策略

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入到应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页。

步骤 5 单击“伸缩”，在“手动伸缩”策略下：

1. 单击 ，修改“实例数”。
2. 单击 ，实例伸缩操作即可生效。

单击“查看组件详情”，可查看到实例在“升级/回滚”中。待状态为“运行中”时，表示已成功完成实例伸缩操作。

----结束

## 设置弹性伸缩策略

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页，单击“伸缩”。

步骤 5 在“弹性伸缩”下，单击“编辑伸缩规则”：

1. 根据业务实际需要设置“冷却时间”、“最大实例数”、“最小实例数”。
2. 单击“保存”。

步骤 6 在“弹性伸缩”下，单击“添加伸缩策略”。

当前支持三种弹性伸缩策略：

- **告警策略：**支持根据 CPU/内存的设置，进行应用组件的自动伸缩。应用组件部署后即可设置，在 CPU/内存超过或少于一定值时，自动增减实例。

表3-22 添加告警策略

参数	参数说明
策略名称	请输入伸缩策略的名称。
策略类型	选择“告警策略”。
指标	选择“指标”。指标是对资源性能的数据描述或状态描述。 <ul style="list-style-type: none"> <li>● CPU 使用率：该指标用于统计测量对象的 CPU 使用率。应用实际使用的与申请的 CPU 核数量比率。</li> <li>● 物理内存使用率：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。</li> <li>● 磁盘读取速率：该指标用于统计每秒从磁盘读出的数据量。</li> <li>● 物理内存总量：该指标用于统计测量对象申请的物理内存总量。</li> </ul>

参数	参数说明
	<ul style="list-style-type: none"> <li>• 数据接收速率：该指标用于统计测量对象每秒钟接收的数据量。</li> <li>• 磁盘写入速率：该指标用于统计每秒写入磁盘的数据量。</li> <li>• 物理内存使用量：该指标用于统计测量对象实际已经使用的物理内存。</li> <li>• CPU 内核总量：该指标用于统计测量对象申请的 CPU 核总量。</li> <li>• 数据发送速率：该指标用于统计测量对象每秒钟发送的数据量。</li> <li>• 容器错包个数：该指标用于统计测量对象收到错误包的数量。</li> <li>• CPU 内核占用：该指标用于统计测量对象已经使用的 CPU 核个数。</li> </ul>
触发条件	设置“触发条件”，在对应条件下触发伸缩策略。
监控周期	设置“监控周期”，触发指标周期统计。 比如设置为 20 秒，表示每 20 秒统计一次。
连续周期	设置“连续周期”。连续周期是指监控周期内连续触发阈值的次数。 比如设置为 3，则表示指标数据连续三个监控周期达到了设定的阈值，则触发策略动作。
执行操作	选择“增加”或“减少”实例数，设置策略触发后执行的动作。

### 📖 说明

单击“收起预览”，可设置“触发条件”、“监控周期”、“连续周期”、“执行操作”参数值。

- **定时策略：**支持在特定时间点进行应用的自动伸缩。适用于秒杀周年庆等活动，例如在秒杀这个时间点增加一定数量的实例个数。

表3-23 添加定时策略

参数	参数说明
策略名称	请输入伸缩策略的名称。
策略类型	选择“定时策略”。
触发时间	设置策略触发时间。
执行操作	选择“增加”、“减少”或“设置”实例数，设置策略触发后执行的动作。

- **周期策略：**支持以天、周、月为周期的伸缩策略。适用于周期性的流量变化。

表3-24 添加周期策略

参数	参数说明
----	------

参数	参数说明
策略名称	请输入伸缩策略的名称。
策略类型	选择“周期策略”。
触发时间	设置策略触发时间。
执行操作	选择“增加”、“减少”或“设置”实例数，设置策略触发后执行的动作。

步骤 7 单击“确定”。

在弹性伸缩下，可看到策略已启动。待到满足触发条件时，弹性伸缩即会生效。

----结束

## 设置弹性伸缩-HPA 策略

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。

### 说明

- 仅 1.15 以上版本 CCE 集群支持弹性伸缩-HPA。
- 虚拟机部署不支持弹性伸缩-HPA。

步骤 1 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页，单击“伸缩”。

步骤 5 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”页面。

- CCE 集群未安装 metrics-server 插件，请执行步骤 6。
- CCE 集群已安装 metrics-server 插件，请执行步骤 8。

步骤 6 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“帮助中心 > 云容器引擎服务 > 用户指南 > 插件管理 > metrics-server”。

步骤 7 等待插件安装完成后，返回“伸缩策略配置”页面，单击“刷新”。

步骤 8 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）。

## 📖 说明

使用“CPU 利用率”或“内存利用率”作为指标，需要应用组件实例已申请“CPU 配额”或“内存配额”。否则，会伸缩异常。

如未申请，请单击“去修改申请配额”。在“更新升级”页面，修改“组件版本”，勾选并设置“CPU 配额”或“内存配额”的“申请”值，然后单击“重新部署”。等待组件升级完成后，再次进入“伸缩策略配置”页面。

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“帮助中心 > 云容器引擎服务 > 用户指南 > 插件管理 > prometheus”。
- 使用 CPU（cpu）、内存（memory）的利用率（Utilization）作为指标，需应用组件实例已申请“CPU 配额”或“内存配额”。否则，会伸缩异常。  
应用组件实例申请“CPU 配额”或“内存配额”，请在组件“更新升级”页面，修改“组件版本”，勾选并设置“CPU 配额”或“内存配额”的“申请”值，然后单击“重新部署”，等待组件升级完成。

步骤 9 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，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页，单击“伸缩”。

步骤 5 选择“伸缩策略配置”，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“帮助中心 > 云容器引擎服务 > 用户指南 > 插件管理 > prometheus”。
- 使用 CPU (cpu)、内存 (memory) 的利用率 (Utilization) 作为指标，需应用组件实例已申请“CPU 配额”或“内存配额”。否则，会伸缩异常。  
应用组件实例申请“CPU 配额”或“内存配额”，请在组件“更新升级”页面，修改“组件版本”，勾选并设置“CPU 配额”或“内存配额”的“申请”值，然后单击“重新部署”，等待组件升级完成。

步骤 6 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，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 2 单击应用名称，进入应用“概览”页。

步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 4 单击应用组件名称，进入应用组件实例“概览”页，单击“伸缩”。

- 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，选择“应用管理 > 应用列表”，可查看所有应用。
- 步骤 2 单击应用名称，进入应用“概览”页。
- 步骤 3 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。
- 步骤 4 单击应用组件名称，进入应用组件实例“概览”页，单击“伸缩”。
- 步骤 5 单击“弹性伸缩 - HPA”右侧的 。
- 步骤 6 单击“确定”，完成策略删除。

----结束

### 3.5.6 设置应用组件实例调度策略

根据应用组件的部署特性，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、存储、网络）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，往往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。这时如果他们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。

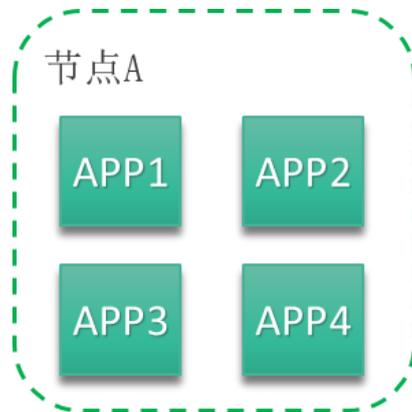
亲和性可以实现就近部署，增强网络能力实现通信上的就近路由，减少网络的损耗。

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 3-3 所示，APP1、APP2、APP3 和 APP4 部署在相同节点上，为亲和性部署。

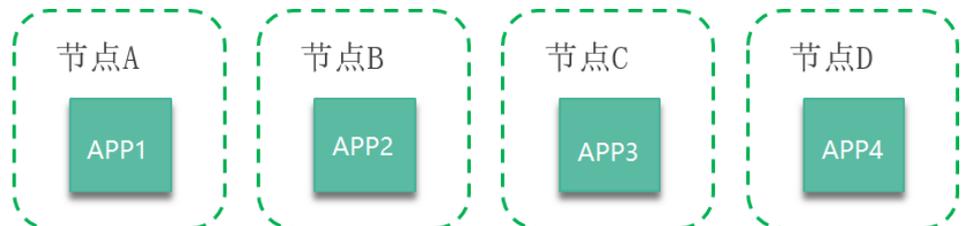
图3-3 应用间亲和



- 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 3-4 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3-4 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

“组件类型”为“通用”且运行时为“Docker”的组件，执行以下操作：

步骤 1 进入应用组件实例调度策略设置界面：

- 在部署应用组件的[组件配置](#)过程中设置应用组件实例调度策略，请执行[步骤 2](#)。
- 组件已经部署完成后设置应用组件实例调度策略，请执行[步骤 3](#)。

步骤 2 在“组件配置”界面：

1. 参考下表设置应用组件实例调度策略。

目的	步骤
设置应用组件与可用区的亲和	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”。
设置应用组件与可用区的反亲和	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	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	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	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”。  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。
设置应用组件间反亲和	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”。  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。

2. 单击“下一步 规格确认”，完成组件部署，操作结束。

步骤 3 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 4 单击应用名称，进入应用“概览”页。

步骤 5 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 6 单击应用组件名称，进入应用组件实例“概览”页。

步骤 7 单击“调度策略”，填写以下参数。

目的	步骤
----	----

目的	步骤
设置应用组件与可用区的亲和	<ol style="list-style-type: none"> <li>1. 单击“添加亲和的对象”。</li> <li>2. 对象类型选择“可用区”，选择待亲和的可用区。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件与可用区的反亲和	<ol style="list-style-type: none"> <li>1. 单击“添加反亲和的对象”。</li> <li>2. 对象类型选择“可用区”，选择待亲和的可用区。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件与节点间的亲和	<ol style="list-style-type: none"> <li>1. 单击“添加亲和的对象”。</li> <li>2. 对象类型选择“节点”，选择待亲和的节点。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件与节点间的反亲和	<ol style="list-style-type: none"> <li>1. 单击“添加反亲和的对象”。</li> <li>2. 对象类型选择“节点”，选择待亲和的节点。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件间亲和	<ol style="list-style-type: none"> <li>1. 单击“添加亲和的对象”。</li> <li>2. 对象类型选择“组件”，选择待亲和的应用组件。</li> <li>3. 单击“确定”。</li> </ol> <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"> <li>1. 单击“添加反亲和的对象”。</li> <li>2. 对象类型选择“组件”，选择待反亲和的应用组件。</li> <li>3. 设置完成后，单击“确定”。</li> </ol> <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>

**步骤 8** 单击“重新部署”，完成应用组件实例调度策略设置。

----结束

其他类型的组件，执行以下操作：

**步骤 1** 进入应用组件实例调度策略设置界面：

- 在部署应用组件的[组件配置](#)过程中设置应用组件实例调度策略，请执行[步骤 6](#)。
- 组件已经部署完成后设置应用组件实例调度策略，请执行[步骤 2](#)。

**步骤 2** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

**步骤 3** 单击应用名称，进入应用“概览”页。

**步骤 4** 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

**步骤 5** 单击应用组件名称，进入应用组件实例“概览”页，单击“更新升级”。

**步骤 6** 展开“高级设置 > 部署配置”，在“调度策略”页签填写以下参数。

目的	步骤
设置应用组件与可用区的亲和	<ol style="list-style-type: none"> <li>1. 单击“添加亲和的对象”。</li> <li>2. 对象类型选择“可用区”，选择待亲和的可用区。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件与可用区的反亲和	<ol style="list-style-type: none"> <li>1. 单击“添加反亲和的对象”。</li> <li>2. 对象类型选择“可用区”，选择待亲和的可用区。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件与节点间的亲和	<ol style="list-style-type: none"> <li>1. 单击“添加亲和的对象”。</li> <li>2. 对象类型选择“节点”，选择待亲和的节点。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件与节点间的反亲和	<ol style="list-style-type: none"> <li>1. 单击“添加反亲和的对象”。</li> <li>2. 对象类型选择“节点”，选择待亲和的节点。</li> <li>3. 单击“确定”。</li> </ol>
设置应用组件间亲和	<ol style="list-style-type: none"> <li>1. 单击“添加亲和的对象”。</li> <li>2. 对象类型选择“组件”，选择待亲和的应用组件。</li> <li>3. 单击“确定”。</li> </ol> <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"> <li>1. 单击“添加反亲和的对象”。</li> <li>2. 对象类型选择“组件”，选择待反亲和的应用组件。</li> <li>3. 设置完成后，单击“确定”。</li> </ol> <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>

步骤 7 完成应用组件实例调度策略设置：

- 在部署应用组件的[组件配置](#)过程中设置应用组件实例调度策略，单击“下一步 规格确认”，完成组件部署。
- 组件已经部署完成后设置应用组件实例调度策略，单击“重新部署”。

----结束

### 3.5.7 设置应用组件实例升级策略

在部署应用组件过程中，或者组件已经部署，都可以设置升级策略。

#### 操作步骤

如果是“组件类型”为“通用”且运行时为“Docker”的组件，执行以下操作：

步骤 1 进入应用组件实例升级策略设置界面：

- 在部署应用组件的[组件配置](#)过程中设置应用组件实例升级策略，请执行[步骤 2](#)。

- 组件已经部署完成后设置应用组件实例升级策略，请执行[步骤 3](#)。

**步骤 2** 在“组件配置”界面，设置应用组件实例升级策略：

1. 选择应用组件实例的升级方式。

#### 说明

如未配置升级策略，系统默认使用滚动升级的方式。

- 滚动升级

逐步用新版本实例替换旧版本实例。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。

- 替换升级

先删除所有旧版本实例，再创建新版本实例。升级过程中业务会中断。

2. 单击“下一步 规格确认”，完成组件部署，操作结束。

**步骤 3** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

**步骤 4** 单击应用名称，进入应用“概览”页。

**步骤 5** 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

**步骤 6** 单击应用组件名称，进入应用组件实例“概览”页。

**步骤 7** 单击“更新升级”，设置应用组件实例升级策略：

1. 选择应用组件实例的升级方式。

#### 说明

如未配置升级策略，系统默认使用滚动升级的方式。

- 滚动升级

逐步用新版本实例替换旧版本实例。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。

- 替换升级

先删除所有旧版本实例，再创建新版本实例。升级过程中业务会中断。

2. 单击“重新部署”，完成应用组件实例升级策略设置。

#### ----结束

如果是其他类型的组件，执行以下操作：

**步骤 1** 进入应用组件实例升级策略设置界面：

- 在部署应用组件的[组件配置](#)过程中设置升级策略，请执行[步骤 6](#)。
- 组件已经部署完成后设置升级策略，请执行[步骤 2](#)。

**步骤 2** 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

**步骤 3** 单击应用名称，进入应用“概览”页。

**步骤 4** 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 5 单击应用组件名称，进入应用组件实例“概览”页，单击“更新升级”。

步骤 6 展开“高级设置 > 部署配置”，在“升级策略”页签，选择应用组件实例的升级方式。

#### 📖 说明

如未配置升级策略，系统默认使用滚动升级的方式。

- 滚动升级  
先安装新实例，再移除旧实例。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。
- 替换升级  
先删除旧实例，再创建新实例。升级过程中业务会中断。

步骤 7 完成应用组件实例升级策略设置：

- 在部署应用组件的[组件配置](#)过程中设置升级策略，单击“下一步 规格确认”，完成组件部署。
- 组件已经部署完成后设置升级策略，单击“重新部署”。

----结束

## 3.5.8 设置应用组件自定义指标监控

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。

在部署应用组件过程中、或者组件已经部署完成，都可以设置组件自定义指标。

### 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

“组件类型”为“通用”且运行时为“Docker”的组件，执行以下操作：

步骤 1 进入应用组件自定义指标监控设置界面：

- 在部署应用组件的[组件配置](#)过程中设置应用组件自定义指标监控，请执行[步骤 2](#)。
- 组件已经部署完成后设置应用组件自定义指标监控，请执行[步骤 3](#)。

步骤 2 在“组件配置”界面：

1. 设置应用组件自定义指标监控，填写以下参数。

参数	说明	是否必须设置

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	exporter 提供的自定义指标名称。 例如, ["cpu_usage","mem_usage"]。 <ul style="list-style-type: none"> <li>• 如果不设置, 则 ServiceStage 会获取所有的自定义指标数据。</li> <li>• 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。</li> </ul>	否

2. 单击“下一步 规格确认”, 完成组件部署, 操作结束。

**步骤 3** 登录 ServiceStage 控制台, 选择“应用管理 > 应用列表”, 可查看所有应用。

**步骤 4** 单击应用名称, 进入应用“概览”页。

**步骤 5** 在“环境视图”页签, 选择“环境”, 可以看到已经部署在该环境下的应用组件。

**步骤 6** 单击应用组件名称, 进入应用组件实例“概览”页。

**步骤 7** 单击“运维配置”, 填写以下参数。

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	exporter 提供的自定义指标名称。 例如, ["cpu_usage","mem_usage"]。 <ul style="list-style-type: none"> <li>• 如果不设置, 则 ServiceStage 会获取所有的自定义指标数据。</li> <li>• 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。</li> </ul>	否

**步骤 8** 单击“重新部署”, 完成应用组件自定义指标监控设置。

### 📖 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“帮助中心 > 应用运维管理 > 用户指南 > 资源监控 > 指标浏览”。

### ----结束

其他类型的组件，执行以下操作：

步骤 1 进入应用组件自定义指标监控设置界面：

- 在部署应用组件的[组件配置](#)过程中设置自定义指标监控，请执行[步骤 6](#)。
- 组件已经部署完成后设置自定义指标监控，请执行[步骤 2](#)。

步骤 2 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 3 单击应用名称，进入应用“概览”页。

步骤 4 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 5 单击应用组件名称，进入应用组件实例“概览”页，单击“更新升级”。

步骤 6 展开“高级设置 > 运维监控”，在“自定义指标监控”页签，填写以下参数。

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	exporter 提供的自定义指标名称。 例如，["cpu_usage","mem_usage"]。 <ul style="list-style-type: none"> <li>• 如果不设置，则 ServiceStage 会获取所有的自定义指标数据。</li> <li>• 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。</li> </ul>	否

步骤 7 完成应用组件实例的自定义指标监控策略设置：

- 在部署应用组件的[组件配置](#)过程中设置自定义指标监控策略，单击“下一步 规格确认”，完成组件部署。
- 组件已经部署完成后设置自定义指标监控策略，单击“重新部署”。

### 📖 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“帮助中心 > 应用运维管理 > 用户指南 > 资源监控 > 指标浏览”。

----结束

### 3.5.9 配置应用日志策略

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。

在部署应用组件过程中，或者应用组件已经部署，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。

#### 操作步骤

步骤 1 进入配置日志策略界面：

- 在部署应用组件的[组件配置](#)过程中配置应用日志策略，请执行[步骤 6](#)。
- 组件已经部署完成后配置应用日志策略，请执行[步骤 2](#)。

步骤 2 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 3 单击应用名称，进入应用“概览”页。

步骤 4 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 5 单击应用组件名称，进入应用组件实例“概览”页，单击“更新升级”。

步骤 6 展开“高级设置 > 运维监控”，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。

参数	参数说明
存储类型	选择存储类型。 <ul style="list-style-type: none"> <li>• “主机路径”：将主机上的路径挂载到指定的容器路径。</li> <li>• “容器路径”：日志仅输出到容器路径，无需挂载主机路径。</li> </ul>
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。
容器挂载	1. 设置“挂载路径”：输入数据卷挂载到应用上的路径。 须知 <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul> 2. 设置“主机扩展路径”。 <ul style="list-style-type: none"> <li>• None：不配置拓展路径。</li> <li>• PodUID：Pod 的 ID。</li> <li>• PodName：Pod 的名称。</li> </ul>

参数	参数说明
	<ul style="list-style-type: none"> <li>• PodUID/ContainerName: Pod 的 ID/容器名称。</li> <li>• PodName/ContainerName: Pod 名称/容器名称。</li> </ul> 3. 设置“老化周期”。 <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> <li>• 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。</li> </ul>

步骤 7 单击“确定”。

步骤 8 完成应用日志策略配置:

- 在部署应用组件的[组件配置](#)过程中配置日志策略, 单击“下一步 规格确认”, 完成组件部署。
- 组件已经部署完成后配置日志策略, 单击“重新部署”。

#### 说明

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“帮助中心 > 应用运维管理 > 用户指南 > 日志管理”。

----结束

## 3.5.10 设置应用性能管理

ServiceStage 支持在部署应用组件过程中, 或者应用组件已经部署后设置应用性能管理。

应用性能管理可协助您快速进行应用问题定位与性能瓶颈分析, 改善体验。

勾选探针会启用性能管理服务并在节点上安装探针, 产生少量资源消耗。Java 探针通过字节码增强技术进行 Java 应用调用埋点, 生成拓扑和调用链数据。

### 注意事项

- 环境需要部署对应版本的应用性能管理 APM 服务时才能开启该功能。
- JDK 工具支持 jdk7 和 jdk8 版本。
- Tomcat 服务器支持 8.x 版本。详细内容请参考“帮助中心 > 应用性能管理 > 用户指南 > 使用限制”。
- 支持容器部署的 Java 应用（运行时是 Java、Tomcat 或者基于 Java 构建的 Docker）设置应用性能管理。

## 操作步骤

“组件类型”为“通用”且运行时为“Docker”的组件，执行以下操作：

步骤 1 进入性能管理设置界面：

- 在部署应用组件的[组件配置](#)过程中设置性能管理，请执行[步骤 2](#)。
- 组件已经部署完成后设置性能管理，请执行[步骤 3](#)。

步骤 2 进入“组件配置”界面，设置性能管理：

1. 勾选“Java 探针”，设置探针相关参数。
  - “探针类型”为“APM 1.0 探针”时，在下拉框中选中“探针版本”。
  - “探针类型”为“APM 2.0 探针”时，设置如下参数：
    - 选择应用：在下拉框中选择和应用组件所在的应用同名的应用名称。如果不存在，请单击“创建应用”进行创建。创建完成后，可以登录应用性能管理控制台，在“应用列表”下查看新创建的应用名称。
    - 探针版本：在下拉框中选择“探针版本”。

### 说明

如需使用最新版本的探针，您可选择 latest。

- 探针升级策略：支持如下升级策略，默认为“重启自动升级”。
    - 重启自动升级：每次都尝试重新下载镜像。
    - 手动升级：如果本地有该镜像，则使用本地镜像，本地不存在时下载镜像。
  - 访问密钥：访问密钥会自动获取，如不能自动获取需要手动输入。
2. 单击“下一步 规格确认”，完成组件部署，操作完成。

步骤 3 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 4 单击应用名称，进入应用“概览”页。

步骤 5 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 6 单击应用组件名称，进入应用组件实例“概览”页。

步骤 7 单击“运维配置”，勾选“Java 探针”，设置探针相关参数。

- “探针类型”为“APM 1.0 探针”时，在下拉框中选中“探针版本”。
- “探针类型”为“APM 2.0 探针”时，设置如下参数：
  - 选择应用：在下拉框中选择和应用组件所在的应用同名的应用名称。如果不存在，请单击“创建应用”进行创建。创建完成后，可以登录应用性能管理控制台，在“应用列表”下查看新创建的应用名称。
  - 探针版本：在下拉框中选择“探针版本”。

### 说明

如需使用最新版本的探针，您可选择 latest。

- 探针升级策略：支持如下升级策略，默认为“重启自动升级”。
  - 重启自动升级：每次都尝试重新下载镜像。

- 手动升级：如果本地有该镜像，则使用本地镜像，本地不存在时下载镜像。
- 访问密钥：访问密钥会自动获取，如不能自动获取需要手动输入。

步骤 8 单击“重新部署”，完成应用性能管理设置。

----结束

其他类型的组件，执行以下操作：

步骤 1 进入性能管理设置界面：

- 在部署应用组件的[组件配置](#)过程中设置性能管理，请执行[步骤 6](#)。
- 组件已经部署完成后设置性能管理，请执行[步骤 2](#)。

步骤 2 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”，可查看所有应用。

步骤 3 单击应用名称，进入应用“概览”页。

步骤 4 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 5 单击应用组件名称，进入应用组件实例“概览”页，单击“更新升级”。

步骤 6 展开“高级设置 > 运维监控”，在“性能管理”页签：

勾选“Java 探针”，设置探针相关参数。

- “探针类型”为“APM 1.0 探针”时，在下拉框中选中“探针版本”。
- “探针类型”为“APM 2.0 探针”时，设置如下参数：
  - 选择应用：在下拉框中选择和应用组件所在的应用同名的应用名称。如果不存在，请单击“创建应用”进行创建。创建完成后，在下拉框中可以查看新创建的应用名称。
  - 探针版本：在下拉框中选择“探针版本”。

#### 说明

如需使用最新版本的探针，您可选择 latest。

- 探针升级策略：支持如下升级策略，默认为“重启自动升级”。
  - 重启自动升级：每次都尝试重新下载镜像。
  - 手动升级：如果本地有该镜像，则使用本地镜像，本地不存在时下载镜像。
- 访问密钥：访问密钥会自动获取，如不能自动获取需要手动输入。

步骤 7 完成应用性能管理设置：

- 在部署应用组件的[组件配置](#)过程中设置应用性能管理，单击“下一步 规格确认”，完成组件部署。
- 组件已经部署完成后设置应用性能管理，单击“重新部署”。

----结束

### 3.5.11 设置应用健康检查

健康检查是指应用组件在运行过程中，根据需要，定时检查应用健康状况。

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3-25 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 进入应用健康检查设置界面：

- 在部署应用组件的[组件配置](#)过程中设置健康检查，请执行[步骤 6](#)。
- 组件已经部署完成后设置健康检查，请执行[步骤 2](#)。

步骤 2 登录 ServiceStage 控制台，选择“应用管理 > 应用列表”。可查看所有应用。

步骤 3 单击应用名称，进入应用“概览”页。

步骤 4 在“环境视图”页签，选择“环境”，可以看到已经部署在该环境下的应用组件。

步骤 5 单击应用组件名称，进入应用组件实例“概览”页，单击“更新升级”。

步骤 6 展开“高级设置 > 运维监控”，在“健康检查”页签，根据业务实际需求设置健康检查参数。

步骤 7 完成应用健康检查设置：

- 在部署应用组件的[组件配置](#)过程中设置应用健康检查，单击“下一步 规格确认”，完成组件部署。
- 组件已经部署完成后设置应用健康检查，单击“重新部署”。

----结束

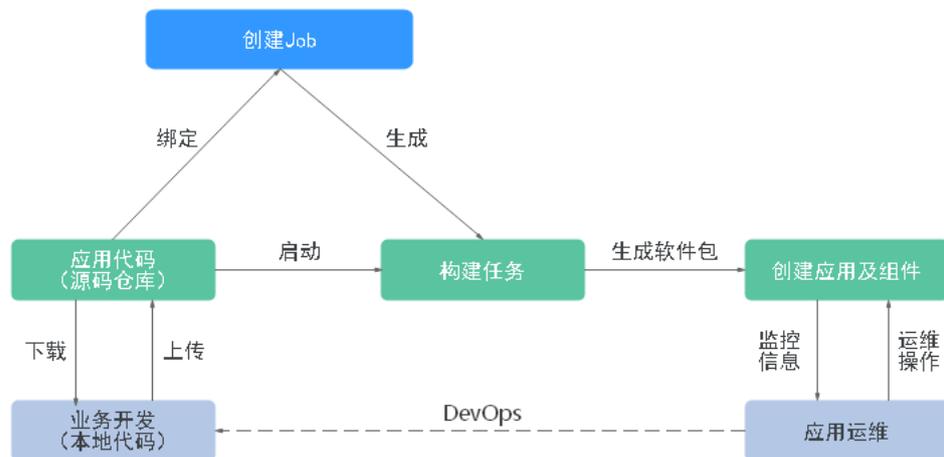
## 3.6 持续交付

### 3.6.1 持续交付概述

#### 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，归档到软件中心，后续部署应用组件时可以直接使用对应的软件包。

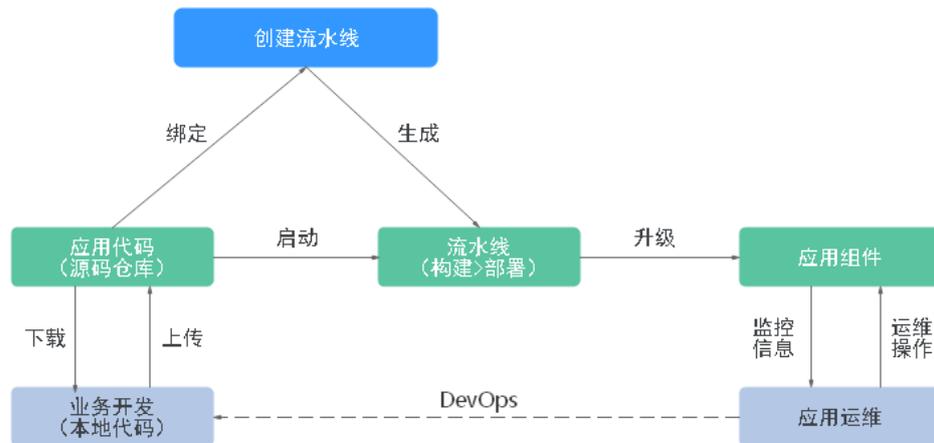
图3-5 创建构建任务



#### 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3-6 创建流水线



### 3.6.2 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“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 创建 CCE 集群”。
2. 已为构建节点绑定弹性 IP，请参考“帮助中心 > 虚拟私有云 > 用户指南 > 弹性 IP 管理 > 绑定弹性 IP 地址”。

#### 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。  
开通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[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“帮助中心 > 云容器引擎 > 用户指南 > 节点管理 > 管理节点标签”。
- 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"> <li>• 公有仓库为系统隔离，同一系统中的租户可共享资源。</li> <li>• 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。</li> </ul>
仓库组织	是指仓库的命名空间。
软件仓库	软件仓库的名称。

参数名称	参数说明
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[部署方式说明](#)。

## 维护操作

表3-26 维护操作

操作	操作说明
查看详情/构建历史	1. 单击目标构建工程名称，在“构建记录”下查看构建历史。 2. 单击构建记录名称，可以查看本次构建记录日志。 3. 单击“代码检查”，可以查看代码检查概况及代码检查详细信息。目前支持的代码检查插件有：checkstyle、findbugs、pmd。 说明 仅 Maven 构建工程支持代码检查。
立即构建	选择目标构建工程，单击“操作”列的“立即构建”。
分支/tag 构建	选择目标构建工程，单击“操作”列的“分支/tag 构建”。 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。 4. 单击“确定”。
编辑	选择目标构建工程，单击“操作”列的“更多 > 编辑”，编辑构建工程。
删除	1. 选择目标构建工程，单击“操作”列的“更多 > 删除”。 2. 单击“确定”。

### 3.6.3 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“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 创建 CCE 集群”。
2. 已为构建节点绑定弹性 IP，请参考“帮助中心 > 虚拟私有云 > 用户指南 > 弹性 IP 管理 > 绑定弹性 IP 地址”。

#### 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。

开通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“帮助中心 > 对象存储服务 > 用户指南 > 控制台指南 > 入门 > 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“帮助中心 > 云容器引擎 > 用户指南 > 节点管理 > 管理节点标签”。

**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[部署方式说明](#)。

## 维护操作

表3-27 维护操作

操作	操作说明
查看详情/构建历史	1. 单击目标构建工程名称，在“构建记录”下查看构建历史。 2. 单击构建记录名称，可以查看本次构建记录日志。
立即构建	选择目标构建工程，单击“操作”列的“立即构建”。
编辑	选择目标构建工程，单击“操作”列的“更多 > 编辑”，编辑构建工程。
删除	1. 选择目标构建工程，单击“操作”列的“更多>删除”。 2. 单击“确定”。

### 3.6.4 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。

#### 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。

开通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3-28 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 <a href="#">步骤 3.1~步骤 3.3</a> 。
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 <a href="#">步骤 3.1</a> 。
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 <a href="#">步骤 3.1~步骤 3.3</a> 。

#### 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[创建源码构建任务](#)或[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[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3-29 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"> <li>1. 选择流水线，选择“更多 &gt; 触发策略”。</li> <li>2. “触发类型”选择“自动触发”。</li> <li>3. 勾选“源码仓库”，推送代码到选择的源码仓库。</li> <li>4. 单击“确定”。</li> </ol>
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"> <li>1. 选择流水线，选择“更多 &gt; 触发策略”。</li> <li>2. “触发类型”选择“定时触发”。</li> <li>3. 设置“触发时间”。</li> <li>4. 单击“确定”。</li> </ol>
	周期触发	<p>设置触发时间段、间隔、触发周期，可以实现周期触发流水线。</p> <p>操作方法如下：</p> <ol style="list-style-type: none"> <li>1. 选择流水线，选择“更多 &gt; 触发策略”。</li> <li>2. “触发类型”选择“定时触发”。</li> <li>3. 开启“周期重复触发”。</li> <li>4. 设置“重复周期”、“触发时间”、“生效”。</li> </ol>

触发策略	触发模式	说明
		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[应用运维](#)。

### 3.6.5 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3-30 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	支持以下仓库类型： <ul style="list-style-type: none"> <li>• GitHub (<a href="https://github.com">https://github.com</a>) 授权方式支持：OAuth/私人令牌。</li> <li>• Gitee (<a href="https://gitee.com">https://gitee.com</a>) 授权方式支持：OAuth/私人令牌。</li> <li>• Bitbucket (<a href="https://bitbucket.org">https://bitbucket.org</a>) 授权方式支持：OAuth/密码/私有 Bitbucket。</li> </ul>

参数	说明
	<ul style="list-style-type: none"> <li>GitLab (<a href="https://gitlab.com">https://gitlab.com</a>)</li> </ul> 授权方式支持：OAuth/私人令牌。 说明 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： <ul style="list-style-type: none"> <li>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）”。</li> <li>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）”。</li> </ul>

步骤 3 单击“创建”，完成仓库授权创建。

----结束

## 3.7 软件中心

### 3.7.1 镜像仓库

#### 3.7.1.1 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。

#### 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。

##### 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“帮助中心 > 容器镜像服务 > 用户指南 > 容器引擎基础知识”。
- 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。

##### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“软件中心 > 镜像仓库”。

步骤 2 在“我的镜像”页面右上角，单击“页面上传”。

步骤 3 在弹出的窗口中选择镜像要上传的“组织”，单击“选择镜像文件”，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 4 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。

#### 前提条件

- 已创建组织，请参见[创建组织](#)。
- 上传镜像的容器引擎客户端版本必须为 1.11.2 及以上。

#### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“软件中心 > 镜像仓库”。

步骤 2 在“我的镜像”页面右上角，单击“客户端上传”。

步骤 3 根据界面提示，完成镜像上传。

----结束

## 3.7.1.2 管理镜像

### 获取镜像下载地址

步骤 1 登录 ServiceStage 控制台，选择“软件中心 > 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“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，选择“软件中心 > 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“编辑”，根据需要在弹出的窗口中：

- 设置“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。
- 设置“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 5 单击“确认”，完成编辑。

----结束

## 添加镜像授权

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

步骤 1 登录 ServiceStage 控制台，选择“软件中心 > 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，该用户享有对应权限。

----结束

## 删除镜像

步骤 1 登录 ServiceStage 控制台，选择“软件中心 > 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

## 3.7.2 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。同一用户可属于不同的组织。组织支持为帐户下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“软件中心 > 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“确认”。

----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“软件中心 > 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“软件中心 > 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[管理镜像](#)。

步骤 3 单击“确定”。

----结束

## 3.8 微服务引擎 CSE

### 3.8.1 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；支持双栈模式，统一接入和管理 Spring Cloud、Apache ServiceComb Java Chassis（Java Chassis）、Go Chassis、Dubbo 侵入式框架和非侵入式服务网格。

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。

### 3.8.2 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，您可以根据业务需要创建使用。

#### 前提条件

微服务引擎专享版运行于虚拟私有云，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“帮助中心 > 虚拟私有云 > 用户指南 > 虚拟私有云和子网 > 创建虚拟私有云和子网”。

如果引擎创建帐号权限为创建引擎的最小权限，如[权限管理](#)的“cse:engine:create”所示。则需要由管理员帐号为其预置 VPC 默认安全组 cse-engine-default-sg，并添加如所示规则。

添加安全组规则，请参考《虚拟私有云用户指南》中“安全组 > 添加安全组规则”章节。

表3-31 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
	1	允许	TCP：30100-30130	IPv6	::/0
	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 CSE”。
- 步骤 2 单击“购买微服务引擎”。
- 步骤 3 填写参数，其中带“\*”标志的参数为必填参数，参数说明如表 3-32 所示。

表3-32 参数说明

参数	说明
*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。 开通企业项目，创建并启用新的企业项目后可以使用，默认选择 default。
*规格	选择微服务引擎规格。 须知 引擎创建成功后，不支持规格变更。
引擎类型	选择微服务引擎的类型。 <ul style="list-style-type: none"> <li>• 集群 集群模式部署，主机级容灾。</li> </ul>

参数	说明
	<ul style="list-style-type: none"> <li>• 单机 单节点部署，不支持容灾。</li> </ul> 须知 “引擎类型”为“单机”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；不支持变更引擎类型。 如选择使用，请先阅读约束说明并勾选“我已知晓如下约束”。
*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。 <ul style="list-style-type: none"> <li>• “引擎类型”选择“集群”时，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"> <li>- 选择 1 个可用区，可提供主机级别容灾能力。</li> <li>- 选择 3 个可用区，可提供可用区级别容灾能力。</li> </ul> </li> <li>• “引擎类型”选择“单机”时，为引擎选择 1 个可用区。</li> </ul>
*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	输入引擎描述信息。 <ol style="list-style-type: none"> <li>1. 单击 ，输入引擎描述信息。</li> <li>2. 单击 ，保存描述。</li> </ol>
安全认证	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。 <ul style="list-style-type: none"> <li>• 选择“开启安全认证”：                             <ol style="list-style-type: none"> <li>1. 根据业务需要确认需开启安全认证后，勾选“我已知晓：开启安全认证后，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。”。</li> <li>2. 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。</li> </ol> </li> <li>• 选择“关闭安全认证”： 无需配置帐户名、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。</li> </ul>

**步骤 4** 单击“下一步”，进入确认微服务引擎规格的页面。确认无误后，单击“提交”，开始创建微服务引擎。

- 创建微服务引擎专享版大约需要 10~30 分钟。
- 创建成功后，微服务引擎的“状态”为“可用”。

- 如果创建失败，可单击 ，在弹出菜单选择“重试”，重新创建。

#### 说明

如果重试失败，可以删除创建失败的微服务引擎。删除微服务引擎专享版，请参考[删除微服务引擎专享版](#)。

如果未及时删除重试失败的微服务引擎，会占用计算资源。系统会在 UTC 时间每日 18:00 统一清理资源。

----结束

## 3.8.3 微服务引擎（CSE）管理

### 3.8.3.1 设置微服务引擎专享版备份与恢复

用户可自定义备份策略，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

- 此功能只适用于微服务引擎专享版。
- 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 CSE”。

步骤 2 单击微服务专享版的引擎名称，进入微服务引擎的“基本信息”页面。

步骤 3 选择“备份与恢复”页签，单击“备份策略”，配置备份参数。

表3-33 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。
开始时间	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 CSE”。
- 步骤 2 单击专享版的微服务引擎名称，进入微服务引擎的“基本信息”页面。
- 步骤 3 选择“备份与恢复”页签，单击“手动备份”，配置备份参数。

表3-34 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！

- 步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。
- 步骤 2 单击专享版的微服务引擎名称，进入微服务引擎的“基本信息”页面。
- 步骤 3 选择“备份与恢复”页签，单击指定备份数据后面“操作”列下的“恢复”。
- 步骤 4 勾选“我已阅读并充分了解该风险”，单击“确定”，恢复备份数据。

单击“恢复记录”，可查看恢复状态。

----结束

### 3.8.3.2 设置微服务引擎专享版公网访问

状态“可用”的微服务引擎专享版，绑定公网 IP 后可提供公网访问微服务引擎专享版的能力。

#### 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。

### 开启公网访问

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击待开启公网访问的微服务引擎名称，进入微服务引擎的“基本信息”页面。

步骤 3 开启“公网访问”开关，在弹出的对话框勾选“我已知晓安全风险”，单击“确定”。

步骤 4 单击“弹性 IP 地址”后的下拉框。在下拉框中选择可用弹性 IP，单击下拉框后侧的“√”。

若下拉框中无可用弹性 IP，请单击“创建弹性 IP”，根据界面提示创建。

----结束

### 关闭公网访问

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击待关闭公网访问的微服务引擎名称，进入微服务引擎的“基本信息”页面。

步骤 3 关闭“公网访问”开关。

步骤 4 在弹出对话框单击“确定”。

----结束

### 3.8.3.3 查看微服务引擎服务注册发现地址和配置中心地址

本章节指导您查看微服务引擎的服务注册发现地址和配置中心地址。

#### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 选择待查看的微服务引擎，查看服务注册发现地址、配置中心地址。

单击服务注册发现地址、配置中心地址后的，可以复制地址信息。

----结束

### 3.8.3.4 查看微服务引擎专享版操作日志

对微服务引擎的创建、升级、删除、变更等操作会在后台运行，您可以在列表中查看任务执行状态。

#### 操作步骤

- 步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。
- 步骤 2 单击专享版的微服务引擎名称，进入微服务引擎的“基本信息”页面。
- 步骤 3 选择“其他操作”页签，单击指定的操作类型名称，查看操作日志详情。

----结束

### 3.8.3.5 升级微服务引擎专享版版本

创建微服务引擎专享版时，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，您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。
- 微服务引擎专享版的版本升级约束，请参见[微服务引擎版本支持机制](#)。

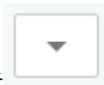
#### 背景介绍

升级过程中，高可用和非高可用引擎对外表现不同：

- 高可用引擎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。
- 非高可用引擎只有一个服务实例，升级过程中服务会中断，导致不能进行服务注册、服务发现和配置变更。需要评估会不会对业务产生影响。如果查到了空实例，会继续使用本地缓存。如果是调用服务注册中心、配置中心接口来进行注册发现，需要使用缓存降级。

#### 操作步骤

- 步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

- 步骤 2 选择可升级且状态可用的微服务引擎，单击 ，在弹出菜单选择“升级”。

也可单击微服务引擎名称，在详情页面的右上方，单击“升级”。

- 步骤 3 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 4 单击“确定”，进行升级。

如果升级失败，可单击 ，在弹出菜单选择“重试”，再次进行升级。

----结束

### 3.8.3.6 删除微服务引擎专享版

如不再使用微服务引擎专享版，可执行删除操作。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

#### 须知

删除引擎后数据无法恢复，请谨慎操作。

### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 选择待删除的微服务引擎专享版，单击引擎名称，进入引擎“基本信息”页面：

1. 在页面的右上方，单击“删除”。
2. 在弹出的对话框中输入“DELETE”，单击“确定”。

----结束

### 3.8.3.7 管理微服务引擎专享版安全认证

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[系统管理](#)。

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。请参考如下方法进行处理：

- **Java Chassis:** 请参考“帮助中心 > 微服务云应用平台 > 开发指南 > 对接微服务应用 > Java Chassis 接入 CSE”中的配置安全认证操作。
- **Spring Cloud:** 请参考“帮助中心 > 微服务云应用平台 > 开发指南 > 微服务开发指南 > 对接微服务应用 > Spring Cloud 接入 CSE”中的配置安全认证操作。

- **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 CSE”。

**步骤 2** 选择待操作的微服务引擎，单击引擎名称，进入引擎“基本信息”页面。

**步骤 3** 在“安全”区域，开启安全认证开关。

- 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“升级引擎至新版本”。
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，单击引擎名称，进入引擎“基本信息”页面。
5. 在“安全”区域，开启安全认证开关。

**步骤 5** 在微服务引擎控制台的“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，单击“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**角色管理**，根据业务需要，创建角色。

**步骤 7** (可选) 参考**帐号管理**，根据业务需要，创建帐号。

**步骤 8** 在微服务引擎控制台的“系统管理”页面，单击“开启安全认证”。

**步骤 9** 根据提示确认已经完成所有配置后，勾选“确保已配置”。

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 CSE”。

步骤 2 选择待操作的微服务引擎，单击引擎名称，进入引擎“基本信息”页面。

步骤 3 在“安全”区域，关闭安全认证开关。

步骤 4 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。
- 关闭安全认证后，开启安全认证后该引擎下创建的帐号不会被删除。

----结束

## 3.8.4 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（当前只有 Java Chassis 和 Go Chassis 支持仪表盘地址自动发现功能）。
- 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。  
配置 monitor 地址，请参考“帮助中心 > 微服务云应用平台 > 开发指南 > 微服务开发指南 > 使用微服务引擎功能 > 使用仪表盘”章节。

### 操作步骤

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 进入微服务引擎页面，选择待操作的微服务引擎，单击“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

#### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)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

## 3.8.5 服务治理

### 3.8.5.1 服务治理概述

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[治理微服务](#)进行服务治理的相关操作。本章节描述的服务治理只适用于 Java Chassis 开发框架。

### 3.8.5.2 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“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、熔断和错误注入等策略的配置，具体说明见下表。

名称	说明
负载均衡	<ul style="list-style-type: none"> <li>• 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提供者多个实例的策略，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。</li> <li>• 前提条件 作用于微服务消费者，需要微服务应用集成负载均衡模块，启用 <code>loadbalance</code> 处理链。 配置示例如下：  <pre> servicecomb:   handler:   chain:     Consumer:                     </pre> </li> </ul>

名称	说明
	<p>default: <b>loadbalance</b></p> <p>在 POM 中添加如下依赖:</p> <pre>&lt;dependency&gt; &lt;groupId&gt;org.apache.servicecomb&lt;/groupId&gt; &lt;artifactId&gt;handler-loadbalance&lt;/artifactId&gt; &lt;version&gt;\${project.version}&lt;/version&gt; &lt;/dependency&gt;</pre>
限流	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 前提条件 作用于微服务提供者，需要微服务应用集成流量控制模块，启用 <b>qps-flowcontrol-provider</b> 处理链。 配置示例如下：</li> </ul> <pre>servicecomb:   handler:     chain:       Provider:         default: <b>qps-flowcontrol-provider</b></pre> <p>在 POM 中添加如下依赖:</p> <pre>&lt;dependency&gt; &lt;groupId&gt;org.apache.servicecomb&lt;/groupId&gt; &lt;artifactId&gt;<b>handler-flowcontrol-qps</b>&lt;/artifactId&gt; &lt;version&gt;\${project.version}&lt;/version&gt; &lt;/dependency&gt;</pre>
降级	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 前提条件 作用于微服务消费者，需要微服务应用集成降级模块，启用 <b>bizkeeper-consumer</b> 处理链。 配置示例如下：</li> </ul> <pre>servicecomb:   handler:     chain:       Consumer:         default: <b>bizkeeper-consumer</b></pre> <p>在 POM 中添加如下依赖:</p> <pre>&lt;dependency&gt; &lt;groupId&gt;org.apache.servicecomb&lt;/groupId&gt; &lt;artifactId&gt;<b>handler-bizkeeper</b>&lt;/artifactId&gt; &lt;version&gt;\${project.version}&lt;/version&gt; &lt;/dependency&gt;</pre>

名称	说明
容错	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 前提条件 作用于微服务消费者，需要微服务应用集成容错模块，启用 <b>loadbalance</b> 处理链。 配置示例如下： <pre>servicecomb:   handler:     chain:       Consumer:         default: <b>loadbalance</b></pre>           在 POM 中添加如下依赖： <pre>&lt;dependency&gt;   &lt;groupId&gt;org.apache.servicecomb&lt;/groupId&gt;   &lt;artifactId&gt;handler-loadbalance&lt;/artifactId&gt;   &lt;version&gt;\${project.version}&lt;/version&gt; &lt;/dependency&gt;</pre> </li> </ul>
熔断	<ul style="list-style-type: none"> <li>• 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。 熔断提供了手工熔断和自动熔断两种策略。手工熔断会设置熔断状态，强制返回；自动熔断需要结合错误率等判断是否熔断。</li> <li>• 前提条件 作用于微服务消费者，需要微服务应用集成熔断模块，启用 <b>bizkeeper-consumer</b> 处理链。 配置示例如下： <pre>servicecomb:   handler:     chain:       Consumer:         default: <b>bizkeeper-consumer</b></pre>           在 POM 中添加如下依赖： <pre>&lt;dependency&gt;   &lt;groupId&gt;org.apache.servicecomb&lt;/groupId&gt;   &lt;artifactId&gt;<b>handler-bizkeeper</b>&lt;/artifactId&gt;   &lt;version&gt;\${project.version}&lt;/version&gt; &lt;/dependency&gt;</pre> </li> </ul>
错误注入	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 前提条件</li> </ul>

名称	说明
	<p>作用于微服务消费者，需要微服务应用集成错误注入模块，启用 <code>fault-injection-consumer</code> 处理链。</p> <p>配置示例如下：</p> <pre>servicecomb:   handler:     chain:       Consumer:         default: <b>fault-injection-consumer</b></pre> <p>在 POM 中添加如下依赖：</p> <pre>&lt;dependency&gt;   &lt;groupId&gt;org.apache.servicecomb&lt;/groupId&gt;   &lt;artifactId&gt;<b>handler-fault-injection</b>&lt;/artifactId&gt;   &lt;version&gt;\${project.version}&lt;/version&gt; &lt;/dependency&gt;</pre>
黑白名单	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 前提条件 只有启用了公钥认证，设置的黑白名单才能生效，请参考<a href="#">公钥认证</a>。</li> </ul>

## 设置负载均衡

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

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	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	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"> <li>• 会话保持时间：会话保持的限制时间，0-86400，单位为秒。</li> <li>• 失败次数阈值：访问失败次数，0-10。当微服务访问下属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。</li> </ul>

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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

配置项	说明	范围
限流对象	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。
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	0-99999 的整数。

配置项	说明	范围
	的值，当前服务实例不再接受该对象的请求。	

#### 📖 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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

#### 📖 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

配置项	配置项说明
降级对象	选择需要降级的微服务与降级方法。
降级策略	<ul style="list-style-type: none"> <li>• 开启：开启降级。</li> <li>• 关闭：关闭降级。</li> </ul>

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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

配置项	配置项说明
容错对象	该应用依赖的应用或方法，下拉菜单可直接选择。
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 说明 当“是否开启容错”配置项设置为“开启”时配置。	<ul style="list-style-type: none"> <li>• Failover 在不同服务器上重新尝试建立连接。</li> <li>• 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。</li> <li>• Failback 在同一个服务器上重新尝试建立连接。</li> <li>• custom                             <ul style="list-style-type: none"> <li>- 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。</li> <li>- 尝试新的服务器次数：尝试与新的服务器建立连接的次数。</li> </ul> </li> </ul>

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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

配置项	配置项说明
熔断对象	该应用调用的服务或方法，下拉菜单可直接选择。
触发条件	<ul style="list-style-type: none"> <li>• 手动熔断 即刻触发熔断，且不再调用。</li> <li>• 取消熔断 将已经熔断的微服务实例取消熔断，继续调用。</li> <li>• 自动熔断                             <ul style="list-style-type: none"> <li>- 熔断时间窗：熔断的持续时间，该时间窗内不再响应请求。</li> <li>- 失败率：触发条件，窗口请求的失败率。</li> <li>- 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。</li> </ul> </li> </ul>

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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

#### 📖 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

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"> <li>• 延迟</li> <li>• 错误</li> </ul>
协议	访问微服务出现延时或错误的协议： <ul style="list-style-type: none"> <li>• Rest</li> <li>• Highway</li> </ul>
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。
触发概率	访问微服务出现延时或错误的概率。

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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

#### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务治理”。

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。

配置项	配置项说明
类型	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul>
匹配规则	使用正则表达式表示。 例如“匹配规则”设置为 <code>data*</code> ，指在黑名单下匹配到的名称以 <code>data</code> 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 <code>data</code> 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了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>
```

### 3.8.6 配置微服务

您可以使用 ServiceStage 提供的全局配置功能，对微服务进行配置。

添加全局配置后，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](#)。

#### 全局配置

全局配置提供微服务间的公共配置，如日志级别、运行参数等。全局配置添加以后，如果微服务没有自定义相同的配置项，会将全局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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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“密码”，单击“确定”。

#### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“全局配置”，可进行如下操作：

操作	步骤
导出配置	单击“全部导出”，导出所有全局配置项。
导入配置	1. 单击“导入”。 2. 单击“...”，选择目标文件。 3. 单击“上传文件”，批量导入配置项。 说明 每次导入的配置项条目数量最多为 150 条。

操作	步骤
添加配置	1. 单击“创建配置”，弹出“创建配置”对话框。 2. 选择微服务环境，并输入“配置项”和“值”。 3. 单击“确定”保存配置。
修改配置	1. 单击目标配置项对应“操作”列的“编辑”。 2. 在编辑框输入“值”。 3. 单击“确定”保存配置修改。
删除配置	1. 选择所需删除的配置。 2. 单击“操作”列的“删除”。 3. 在“删除配置”对话框中单击“确定”，删除全局配置。
批量删除	1. 选择所需删除的配置。 2. 单击配置项列表上方的“删除”，批量删除全局配置项。

----结束

### 3.8.7 维护微服务

您可通过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。

“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，也可以选择环境筛选应用。
- 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、应用筛选服务。
- 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。

#### 动态配置

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

#### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务目录”。

步骤 5 单击微服务名称。

步骤 6 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
导出配置	在“所有作用域”下拉列表选择一个作用域，单击“全部导出”，导出当前作用域的 json 格式配置文件。 作用域的格式如下： <ul style="list-style-type: none"> <li>• 微服务名称@所属应用</li> <li>• 微服务名称@所属应用#版本号</li> </ul>
导入配置	1. 单击“导入”，选择“作用域”。 作用域的格式如下： <ul style="list-style-type: none"> <li>• 微服务名称@所属应用</li> <li>• 微服务名称@所属应用#版本号</li> </ul> 2. 单击“...”，选择目标文件。 3. 单击“上传文件”，批量导入配置项。 4. 单击“关闭”。
创建配置	1. 单击“创建配置”，选择配置“作用域”。 2. 输入“配置项”。 3. 输入“值”。 4. 单击“确定”，保存配置。
修改配置	1. 单击目标配置项对应“操作”列的“编辑”。 2. 在“值”编辑框输入新值。 3. 单击“确定”，保存配置修改。
删除配置	1. 单击目标配置项对应“操作”列的“删除”。 2. 单击“确定”，删除配置。

----结束

## 灰度发布

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

#### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务目录”。

步骤 5 单击微服务名称，进入详情页面，单击“灰度发布”。

步骤 6 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。
  - 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"> <li>• 选择规则作用的版本。</li> <li>• 勾选“是否添加自定义版本”，根据界面提示添加新版本。</li> </ul>
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比率平均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。
    - 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"> <li>• 选择规则作用的版本。</li> <li>• 勾选“是否添加自定义版本”，根据界面提示添加新版本。</li> </ul>
规则配置	<ul style="list-style-type: none"> <li>• 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。</li> <li>• 规则 契约的 Key 对应的 Value。</li> </ul> <p>说明</p> <ul style="list-style-type: none"> <li>• 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</li> </ul>

配置项	说明
	<p>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。</p> <ul style="list-style-type: none"> <li>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。</li> </ul>

c. 单击“确定”，完成自定义规则配置和灰度发布。

----结束

## 删除微服务

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务目录”。

步骤 5 选择需要删除的微服务，单击“删除”，根据提示删除对应微服务。

### 说明

- 当微服务实例个数为 0 时，可直接删除微服务。
- 当微服务实例个数不为 0 时，删除微服务后过一段时间微服务将会重新注册到服务中心。

----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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。

- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务目录”。

步骤 5 单击微服务名称，进入微服务详情页。

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、灰度发布、服务契约等信息。

----结束

## 查看服务契约

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 📖 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务目录”。

步骤 5 单击微服务名称，进入微服务详情页。

步骤 6 单击“服务契约”，可以查看服务契约。

----结束

## 添加标签

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 📖 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“服务目录”。

步骤 5 单击微服务名称，进入详情页面，单击标签后的“标签管理”。

步骤 6 单击“添加标签”，输入“键”和“值”。

步骤 7 单击“确定”，保存修改。

----结束

## 修改微服务实例状态

“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 CSE”。

步骤 2 单击微服务引擎的“查看控制台”。

- 未开启安全认证的微服务引擎专享版或微服务引擎专业版，请执行[步骤 4](#)。
- 开启安全认证的微服务引擎专享版，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密码，单击“确定”。

### 说明

- 首次进入微服务控制台，请输入 root 帐号名及[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 进入微服务控制台。

步骤 5 选择“服务目录 > 实例列表”。

步骤 6 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  
单击“操作”列的“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

## 3.8.8 系统管理

### 3.8.8.1 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 内部微服务的权限管理机制。
  - 如果您通过 ServiceStage 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。
  - 当前系统管理功能仅支持微服务权限管理，暂不支持配置管理、服务治理。
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[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[角色管理](#)。
    - 角色权限说明，请参见[表 3-35](#)。

表3-35 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。

### 3.8.8.2 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3-36 帐号管理操作说明

操作	说明
新增帐号	<p>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。</p> <p>ServiceStage 支持管理 1000 个帐号。</p>
查看权限	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	<p>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。</p> <p>须知</p> <ul style="list-style-type: none"> <li>• 密码修改后，需及时更新微服务认证配置。</li> <li>• Java Chassis：请参考“帮助中心 &gt; 微服务云应用平台 &gt; 开发指南 &gt; 微服务开发指南 &gt; 对接微服务应用 &gt; Java Chassis 接入 CSE”中的配置安全认证操作。</li> <li>• Spring Cloud：请参考“帮助中心 &gt; 微服务云应用平台 &gt; 开发指南 &gt; 微服务开发指南 &gt; 对接微服务应用 &gt; Spring Cloud 接入 CSE”中的配置安全认证操作。</li> <li>• 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。</li> </ul>
重置密码	<p>根据业务需求或者安全规定，可使用已登录微服务引擎的帐号对该微服务引擎下的其他帐号密码进行重置。</p> <p>须知</p> <ul style="list-style-type: none"> <li>• 密码重置后，需及时更新微服务认证配置。</li> <li>• Java Chassis：请参考“帮助中心 &gt; 微服务云应用平台 &gt; 开发指南 &gt; 微服务开发指南 &gt; 对接微服务应用 &gt; Java Chassis 接入 CSE”中的配置安全认证操作。</li> <li>• Spring Cloud：请参考“帮助中心 &gt; 微服务云应用平台 &gt; 开发指南 &gt; 微服务开发指南 &gt; 对接微服务应用 &gt; Spring Cloud 接入 CSE”中的配置安全认证操作。</li> <li>• 密码重置后，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。</li> </ul>
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。

## 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 CSE”。

**步骤 2** 进入微服务引擎页面，选择待新增帐号的微服务引擎，单击“查看控制台”。

**步骤 3** 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“帐号名称”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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

**步骤 4** 选择“系统管理 > 帐号管理 > 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 5** 单击“确定”，完成新帐号创建。

----结束

## 查看权限

**步骤 1** 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

**步骤 2** 进入微服务引擎页面，选择待查看帐号的微服务引擎，单击“查看控制台”。

**步骤 3** 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[创建微服务引擎专享版](#)时输入的密码。

- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 帐号管理”，单击帐号列表中待查看帐号关联的角色名称。

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 CSE”。

步骤 2 进入微服务引擎页面，选择待编辑帐号的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 帐号管理”，单击待编辑帐号“操作”列的“编辑帐号”。

步骤 5 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 6 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 CSE”。

步骤 2 进入微服务引擎页面，选择待修改帐号密码的微服务引擎，单击“查看控制台”。

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[创建微服务引擎专享版](#)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 4 选择“系统管理 > 帐号管理”，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 5 单击“保存”，完成密码修改。

----结束

## 重置密码

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 进入微服务引擎页面，选择待重置帐号密码的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 帐号管理”，选择待重置密码的帐号名，单击“操作”列的“重置密码”。

1. 输入“新密码”和“确认密码”。
2. 查看提示信息确认需要重置密码后，勾选“我已确认知晓”。

步骤 5 单击“保存”，完成密码重置。

----结束

## 删除帐号

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 进入微服务引擎页面，选择待删除帐号的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 帐号管理”，单击待删除帐号“操作”列的“删除”。

步骤 5 在输入框输入“DELETE”，单击“确定”。

----结束

### 3.8.8.3 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 3-37 所示的操作。

表3-37 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。
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"> <li>角色删除后无法恢复，请谨慎操作。</li> <li>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<a href="#">编辑帐号</a>。</li> </ul>
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 CSE”。

步骤 2 进入微服务引擎页面，选择待创建角色的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 角色管理”，单击“创建角色”。

步骤 5 输入新角色名称。

#### 📖 说明

角色创建后，角色名称不可修改。

步骤 6 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 3-38 进行设置。

表3-38 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</p> <p>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"> <li>• 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。</li> </ul> <p>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。</p> <ul style="list-style-type: none"> <li>• 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。</li> </ul> <p>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。</p> <ul style="list-style-type: none"> <li>• 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”。</li> </ul> <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”。</p> <p>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 7 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 CSE”。

步骤 2 进入微服务引擎页面，选择待编辑角色的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 角色管理”，单击待编辑角色“操作”列的“编辑”。

步骤 5 根据实际业务需求，修改“服务组”和“权限动作”。

步骤 6 单击“保存”，完成角色编辑。

----结束

## 删除角色

---

### 须知

- 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](#)。
- 

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 进入微服务引擎页面，选择待删除角色的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 角色管理”，单击待删除角色“操作”列的“删除”。

步骤 5 在输入框输入“DELETE”，单击“确定”。

----结束

## 查看角色

步骤 1 登录 ServiceStage 控制台，选择“微服务引擎 CSE”。

步骤 2 进入微服务引擎页面，选择待查看角色的微服务引擎，单击“查看控制台”。

步骤 3 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[创建微服务引擎专享版](#)时输入的密码。
- 创建帐号请参考[新增帐号](#)。

步骤 4 选择“系统管理 > 角色管理”，点击待查看的角色前的 ，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

## 3.9 云审计服务支持的关键操作

### 3.9.1 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。

帮助中心 > 云审计 > 计费说明 > 申请云审计服务后，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3-39 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

操作名称	资源类型	事件名称
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
删除环境	environment	deleteEnvironment

### 3.9.2 查看云审计日志

查看 ServiceStage 云审计日志，请参考帮助中心 > 云审计 > 用户指南 > 云审计服务管理 > 查看追踪事件。

## 3.10 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

ServiceStage 支持容器和虚机两种组件部署方式。

- 设置容器部署组件监控及告警

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，[设置应用组件自定义指标监控](#)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[设置资源监报告警阈值](#)，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。

- 容器部署组件支持的监控指标  
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 3-40](#)。

表3-40 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。	$\geq 1$	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。	$\geq 0$	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)
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。	$\geq 0$	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)。	$\geq 0$	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。	$\geq 0$	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。	$\geq 0$	千字节/秒 (Kilobytes/Second)
下行 Pps (recvPackRate)	每秒网卡接收的数据包个数。	$\geq 0$	个/秒 (Packets/Second)

监控指标	指标含义	取值范围	单位
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。	$\geq 0$	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。	$\geq 0$	字节/秒 (Bytes/Second)
下行包错率 (recvErrPackRate)	每秒网卡接收的错误包个数。	$\geq 0$	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。	$\geq 0$	字节/秒 (Bytes/Second)
上行包错率 (sendErrPackRate)	每秒网卡发送的错误包个数。	$\geq 0$	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。	$\geq 0$	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。	$\geq 0$	个 (Packets)
线程数 (threadsCount)	该指标用于统计主机中当前创建的线程数量。	$\geq 0$	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。	$\geq 0$	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。	$\geq 0$	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。	$\geq 0$	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"> <li>• 0: 表示正常</li> <li>• 1: 表示异常</li> </ul>	无

监控指标	指标含义	取值范围	单位
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。	$\geq 0$	兆字节 (Megabytes)

# 4 常见问题

## 4.1 环境管理

### 4.1.1 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。

## 4.2 应用管理

### 4.2.1 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，选择“应用管理 > 应用列表”。
- 步骤 2 单击应用名称，进入应用“概览”页。
- 步骤 3 在“组件列表”，单击部署失败的组件名称，进入组件“概览”页。
- 步骤 4 选择状态异常的组件版本，单击组件名称，进入组件实例“概览”页。
- 步骤 5 单击“实例列表”，单击实例名称前的 ，再单击“事件”。
- 步骤 6 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

## 4.2.2 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-klxsp	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-c9b57b546-6vj5m	创建中	0.8Core / 2Core	0.78GiB / 4GiB		11分钟		2019/08/14 17:50:31 GMT+08:00	删除

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	警告	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。

## 4.2.3 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。

## 4.2.4 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"/>
```

## 4.2.5 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一下“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。

## 4.2.6 如何体验 ServiceStage 的源码部署功能?

如表 4-3 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[部署组件](#)去体验 ServiceStage 的源码部署功能。

表4-1 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	<a href="https://github.com/servicestage-template/ServiceComb-SpringMVC">https://github.com/servicestage-template/ServiceComb-SpringMVC</a>
ServiceComb-JAX-RS	Java	<a href="https://github.com/servicestage-template/ServiceComb-JAX-RS">https://github.com/servicestage-template/ServiceComb-JAX-RS</a>
ServiceComb-POJO	Java	<a href="https://github.com/servicestage-template/ServiceComb-POJO">https://github.com/servicestage-template/ServiceComb-POJO</a>
SpringBoot-WebService	Java	<a href="https://github.com/servicestage-template/SpringBoot-WebService">https://github.com/servicestage-template/SpringBoot-WebService</a>
SpringBoot-Webapp-Tomcat	Java	<a href="https://github.com/servicestage-template/SpringBoot-Webapp-Tomcat">https://github.com/servicestage-template/SpringBoot-Webapp-Tomcat</a>
nodejs-express	Node.js	<a href="https://github.com/servicestage-template/nodejs-express-4-16">https://github.com/servicestage-template/nodejs-express-4-16</a>
nodejs-koa	Node.js	<a href="https://github.com/servicestage-template/nodejs-koa-2-5-2">https://github.com/servicestage-template/nodejs-koa-2-5-2</a>

demo 名称	语言类型	GitHub 代码仓库地址
php-laravel	PHP	<a href="https://github.com/servicestage-template/php-laravel-v5-6-28">https://github.com/servicestage-template/php-laravel-v5-6-28</a>
php-slim	PHP	<a href="https://github.com/servicestage-template/php-slim-3-10-0">https://github.com/servicestage-template/php-slim-3-10-0</a>
Gosimple	Golang	<a href="https://github.com/servicestage-template/Gosimple">https://github.com/servicestage-template/Gosimple</a>

## 4.3 持续交付

### 4.3.1 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。

### 4.3.2 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 服务搭建在公有云内部网络，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，请参考帮助中心 > 虚拟私有云 > 用户指南 > 安全性 > 安全组 > 为安全组添加安全组规则。

#### 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[创建源码构建任务](#)，获取“构建集群”和“过滤节点标签”。

图4-1 获取集群名称和标签信息

The screenshot shows a web interface for cluster management. At the top, there is a section for '构建集群' (Create Cluster) with a dropdown menu showing 'cce-#集群名称' and a 'C' icon. Below it, a note says '如果没有集群, 请点击 新建集群 创建完成, 刷新后再选择'. Underneath is a table for '过滤节点标签' (Filter Node Tags) with columns '键' (Key), '值' (Value), and '操作' (Action). The '键' column has a dropdown with 'node.kubernetes.io/barem...' selected. The '值' column has a dropdown with 'false' selected. There is a '刷新' (Refresh) button in the '操作' column. At the bottom, a note says '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 3 获取该集群下有该标签的节点弹性 IP。

- 应用组件构建
  - a. 请参考[查看应用组件构建](#), 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, 请参考帮助中心 > 虚拟私有云 > 用户指南 > 安全性 > 安全组 > 为安全组添加安全组规则。

----结束

### 4.3.3 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[创建源码构建任务](#), 获取“构建集群”和“过滤节点标签”。

图4-2 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. 请参考[查看应用组件构建](#)，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，请参考帮助中心 > 虚拟私有云 > 用户指南 > 安全性 > 安全组 > 为安全组添加安全组规则。

----结束

### 4.3.4 使用 ServiceStage 构建失败

软件工程构建失败的原因很多，可以按照以下场景排查定位。

#### 出现拉取不到代码的场景

一般原因可能有三种。

- 如果是在自己的节点上构建应用，可能是该节点没有绑定弹性 ip，如上图所示“192.168.x.x”的节点没有绑定弹性 ip，解决方法：请参考“帮助中心 > 虚拟私有云 > 用户指南 > 弹性 IP 管理 > 绑定弹性 IP 地址”。
- 授权信息过期，代码源的私人令牌权限范围不够，或者授权信息已被移除等，例如：devcloud 的代码源，拉取不到代码，可能是创建授权的时候，用户名对应的密码输错了，导致拉取不到代码。解决方法：[仓库授权](#)即可。
- 自己搭建的代码源仓库，和构建的节点网络不通，例如：在集群 A 的某一个节点上搭建了一个私有的 bitbucket，使用集群 B 构建，但是集群 B 和集群 A 不是同一个 vpc，内网不通，导致构建拉取不到代码。解决方法：打通网络。

#### 构建的代码依赖自己的私有 maven 仓库

有以下两种解决方案。

- 在自己的项目的根目录下增加 settings.xml 文件，在 settings.xml 指定自己的私有 maven 仓库地址（如果自己的私有 maven 仓库时需要认证的，则需要 settings.xml 配置上自己的认证信息，用户名密码等）。
- 在自己项目的 pom.xml 文件中，指定自己的私有 maven 仓库。

## 4.4 软件中心

### 4.4.1 如何解决上传软件包失败的问题？

#### 问题描述

上传满足系统要求的软件包后，系统提示“无权限访问，请联系管理员”。

#### 解决方法

步骤 1 在谷歌浏览器中，选择“更多工具 > 清除浏览数据”。

步骤 2 在弹出的清除浏览数据窗口中，保持默认勾选，单击“清除数据”。

----结束

### 4.4.2 如何解决 Docker 客户端 push 镜像失败的问题

#### 问题描述

在后台 Docker 登录成功以后，使用 Docker 客户端上传镜像包时，例如执行如下命令上传：

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

#### 📖 说明

- 10.125.54.133:20202 为租户或用户准备上传仓库的 IP 和端口号。
- test1 为 namespace。

上传失败，Docker 客户端出现如下提示：

```
unauthorized: authentication required
```

#### 解决方法

步骤 1 租户或用户以正确的 AK/SK 登录 Docker 客户端。

步骤 2 向本租户或用户下有操作权限的 namespace 上传镜像，或者更换到新的 namespace。

- 执行如下命令，向本租户或用户下有操作权限的 namespace 上传镜像。

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

#### 📖 说明

- 10.125.54.133:20202 为为租户或用户准备上传仓库的 IP 和端口号。
- test2 为该租户或用户下有操作权限的 namespace。
- 执行如下命令，更换到新的 namespace。

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

#### 📖 说明

- 10.125.54.133:20202 为为租户或用户准备上传仓库的 IP 和端口号。
- test3 为新的 namespace。

步骤 3 上传成功后，显示如下：

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

出现异常，请联系技术支持工程师。

----结束

## 4.4.3 下载 SWR 软件包失败

### 问题描述

创建 tomcat 应用时，界面显示创建失败。经查后台 tomcat 日志，原因为下载 swr 中的软件包时认证失败。手工下载 swr 中的软件包，报 401。

### 解决方法

把镜像设置为公开即可，私有包会导致因权限问题无法被拉取。

## 4.5 基础设施

### 4.5.1 退订服务器是否影响现有程序运行？

#### 问题描述

退订服务器是否影响现有程序运行？

#### 解决方法

- 容器部署，退订服务器后服务实例将会在 CCE 集群内重新调度。

## 4.5.2 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。

步骤 2 重新部署应用，请参考。

----结束

## 4.5.3 如何将组件构建集群由共享构建集群切换为私有构建集群？

### 问题描述

ServiceStage 提供的共享构建集群为所有用户共用，从安全和资源隔离的角度考虑，并不安全。

建议您在共享构建集群下线前，尽快将共享构建集群上的应用配置到私有构建集群，私有构建集群会提供更加安全可靠的服务。

### 前置操作

1. 已经创建集群，请参考“帮助中心 > 云容器引擎 > 用户指南 > 集群管理 > 创建 CCE 集群”。
2. 已为 CCE 集群节点绑定弹性 IP，请参考“帮助中心 > 虚拟私有云 > 用户指南 > 弹性 IP 管理 > 绑定弹性 IP 地址”。

### 解决方法

步骤 1 登录 ServiceStage 控制台，选择“持续交付 > 构建”。

步骤 2 选择“所有项目”、“所有创建类别”、“共享集群”和“所有状态”进行过滤，根据过滤结果判断是否需要切换。

- 如果过滤结果为空，操作结束。
- 如果过滤结果不为空，执行步骤 3。



步骤 3 进入编辑构建工程界面：

- “创建类别”为“系统创建”：选择“应用管理 > 应用组件”，单击待操作的组件名称，选择“构建 > 编辑”。
- “创建类别”为“用户创建”：在当前页面待操作构建名称所在列操作行，选择“更多 > 编辑”。

步骤 4 编辑构建工程。

- 以源码部署的组件，请参考[编辑源码构建工程](#)编辑构建工程。
- 以软件包部署的组件，请参考[编辑软件包构建工程](#)。

其中：

1. “构建集群”选择“使用自己的构建集群”。
2. 在“构建集群”下拉列表选择私有构建集群。

----结束

## 4.6 应用运维

### 4.6.1 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。
- 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。请参考进行相关配置。

## 4.6.2 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。

## 4.6.3 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。

## 4.7 应用开发问题

### 4.7.1 微服务和普通应用有什么不同？

微服务是一种架构模式，其核心是将一个单体应用分成多个部分进行开发。所以微服务架构的应用程序，其本质上是一个分布式应用。

基于微服务架构构建的应用程序，可以让业务变化更快，整体系统可靠性更高。

类型	微服务	普通应用
开发	每个微服务的体量相对较小，业界的 two pizza 团队和“2 周即可全部重写全部代码”等都可以作为微服务划分的参考。在开发时期，需注意服务接口的定义以及与周边微服务进行配合，“基于契	普通应用逻辑复杂、模块耦合、代码臃肿、修改难度大、版本迭代效率低下。

类型	微服务	普通应用
	约”的开发方式是非常推荐的。 微服务开发，请参考“帮助中心 > 微服务云应用平台 > 开发指南 > 微服务开发指南 > 开发微服务应用”章节。	
部署	微服务组成的应用系统通常比较复杂，在一次性部署的时候，需要进行编排部署。 微服务应用部署，请参考 <a href="#">部署组件</a> 。	普通应用可能会比较大，构建和部署时间也相应地比较长，不利于频繁部署，阻碍持续交付。在移动应用开发中，这个问题会显得尤为严重。
运维	在原来的指标监控、日志收集之外还非常强调治理。其核心理念是在运行时期通过对线上系统的各种调整以达到系统整体健康度要求的效果。 应用运维，请参考 <a href="#">应用运维</a> 。	普通应用线上问题修复周期长，任何一个线上问题修复都需要对整个应用系统进行全面升级。

## 4.7.2 部署在云上的微服务如何进行排错？

对于问题的定界，可以使用微服务仪表盘，通过仪表盘可以看到系统内所有微服务及其实例的实时运行情况，找到没有正常工作的节点。

找到问题节点后，可以通过 APM 查看问题节点的应用日志来分析具体问题。

## 4.7.3 如何解决获取依赖失败的问题？

配置 maven 镜像源之后，获取依赖失败，如下图所示。

图4-3 获取依赖失败

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

对于企业内部需要使用代理访问外网的情况，需要配置 Maven 代理，可以在用户目录（windows 中如 C:\Users\yang\*\*\*\）下的.m2 目录中 setting.xml（用户配置）或 Maven 安装目录下的 conf 目录中 setting.xml（系统全局配置）里配置代理来实现。

找到 setting.xml 文件中的标签对，在其内配置代理信息，参考如下样例。

```
<proxies>
```

```
<!-- proxy
  | Specification for one proxy, to be used in connecting to the
network.
  |
  -->
<proxy>
  <id>自定义proxyid, 多个proxy配置间不可重复</id>
  <active>true</active>
  <protocol>http</protocol>
  <username>proxy认证帐号</username>
  <password>proxy认证密码</password>
  <host>企业的代理地址</host>
  <port>代理地址端口号</port>
</proxy>

<proxy>
  <id>自定义proxyid, 多个proxy配置间不可重复</id>
  <active>true</active>
  <protocol>http</protocol>
  <username>proxy认证帐号</username>
  <password>proxy认证密码</password>
  <host>企业的代理地址</host>
  <port>代理地址端口号</port>
</proxy>

</proxies>
```

## 4.7.4 服务名重复校验范围是什么？

### 问题描述

服务名重复校验范围是什么？

### 解决方法

服务名重复校验范围是微服务名称、微服务应用、微服务版本和微服务环境。

是一个微服务的主键，标识一个唯一的微服务。

请确保主键不重复。

## 4.7.5 为什么一定要定义服务契约？

企业级系统规模普遍较大，微服务组件众多，所以对服务间接口进行统一管理是企业的关键需求。微服务引擎通过契约管理满足这一需求。

管理角度：通过契约管理，企业中的接口管理者可以统一定义微服务的契约文件（符合接口描述标准的接口定义文件），从而做到规范并协调多个开发团队的接口开发，降低沟通成本且避免后期的混乱。

开发角度：在微服务开发的时候，不同团队甚至不同 ISV 间，可以基于统一的契约文件开发同一应用或系统，从而方便整体系统一致性的维护。具体表现在，单体应用中模间是代码级调用，在编译期就可以解决 API 不兼容问题，修复成本也极低。微服务解耦后，服务间变为了远程调用，接口不一致通常发现时间较晚，会造成更大的修复成本。有了契约可以保证架构师设计契约，严格审查变更，并反向生成代码，保证兼容性。

另外，对于规模较小、统一管理要求不高的系统，产品支持从接口代码自动生成契约文件。