



应用管理与运维平台

# 用户指南

发布日期 2023-06-25

---

# 目录

---

<b>1 产品介绍</b>	<b>1</b>
1.1 什么是应用管理与运维平台 (ServiceStage)	1
1.2 产品优势	2
1.3 典型应用场景	3
1.3.1 构建微服务应用	3
1.3.2 Web 应用生命周期管理	5
1.4 名词解释	6
1.5 使用限制	6
1.6 规格说明	7
1.7 权限管理	9
<b>2 快速入门</b>	<b>17</b>
<b>3 用户指南</b>	<b>22</b>
3.1 概述	22
3.2 权限管理	23
3.2.1 创建用户并授权使用 ServiceStage	23
3.2.2 创建 ServiceStage 自定义策略	24
3.2.3 设置 ServiceStage 依赖服务的权限	25
3.3 应用管理	26
3.3.1 创建应用	26
3.3.2 新建应用组件	27
3.3.2.1 应用组件概述	27
3.3.2.2 快速创建组件	30
3.3.2.3 创建微服务组件	31
3.3.2.4 创建 Web 应用组件	34
3.3.2.5 创建通用组件	36
3.3.3 应用组件部署	38
3.3.3.1 部署方式说明	38
3.3.3.2 部署组件	38
3.3.4 应用组件管理	42
3.3.5 应用高级设置	45
3.3.5.1 设置应用环境变量	45
3.3.5.2 设置应用生命周期	46

3.3.5.3 设置数据存储.....	48
3.3.5.4 设置分布式会话.....	53
3.3.6 应用组件构建.....	53
3.3.7 应用组件流水线.....	55
3.3.8 应用配置管理.....	56
3.3.8.1 创建密钥.....	57
3.3.8.2 创建配置项.....	59
3.4 环境管理.....	61
3.5 应用运维.....	63
3.5.1 维护应用组件实例.....	63
3.5.2 添加应用组件实例标签.....	65
3.5.3 配置域名映射.....	66
3.5.4 设置资源监警告警阈值.....	67
3.5.5 设置应用组件实例伸缩策略.....	68
3.5.6 设置应用组件实例调度策略.....	72
3.5.7 设置应用组件实例升级策略.....	76
3.5.8 设置应用组件自定义指标监控.....	77
3.5.9 配置应用日志策略.....	80
3.5.10 设置应用健康检查.....	82
3.6 持续交付.....	83
3.6.1 概述.....	83
3.6.2 创建源码构建任务.....	84
3.6.3 创建软件包构建任务.....	87
3.6.4 流水线管理.....	89
3.6.5 仓库授权.....	92
3.7 软件中心.....	92
3.7.1 镜像仓库.....	93
3.7.1.1 上传镜像.....	93
3.7.1.2 管理镜像.....	94
3.7.2 组织管理.....	95
3.8 基础设施.....	97
3.8.1 微服务引擎（CSE）.....	97
3.8.1.1 概述.....	97
3.8.1.2 创建微服务引擎专享版.....	97
3.8.1.3 微服务引擎（CSE）管理.....	99
3.8.1.3.1 设置微服务引擎专享版备份与恢复.....	99
3.8.1.3.2 设置微服务引擎专享版公网访问.....	100
3.8.1.3.3 查看微服务引擎接入地址.....	101
3.8.1.3.4 查看微服务引擎专享版操作日志.....	101
3.8.1.3.5 升级微服务引擎专享版.....	101
3.8.1.3.6 删除微服务引擎专享版.....	102
3.8.1.4 使用微服务仪表盘.....	103

3.8.1.5 服务治理.....	103
3.8.1.5.1 概述.....	103
3.8.1.5.2 治理微服务.....	103
3.8.1.6 配置微服务.....	109
3.8.1.7 维护微服务.....	110
3.8.2 单台虚拟机安装虚拟机 Agent.....	115
<b>4 常见问题.....</b>	<b>117</b>
4.1 如何获取 AK/SK? .....	117
4.2 如何处理修改项目名称导致的错误? .....	117
4.3 微服务和普通应用有什么不同? .....	118
4.4 如何查看应用组件部署失败的原因? .....	119
4.5 如何处理部署或升级虚拟机组件失败? .....	119
4.6 Node.js 8 软件包打包有哪些约束? .....	120
4.7 如何处理安装 Agent 失败? .....	120
4.8 如何处理 Agent 离线? .....	120
4.9 虚拟机部署的应用组件支持在什么目录写文件? .....	121
4.10 如何处理虚拟机部署的组件删除失败报 host status is not active 错误的问题? .....	121
4.11 Docker 应用程序依赖的问题.....	122
4.12 如何解决 Docker 客户端 push 镜像失败的问题? .....	122
4.13 如何获取项目名称? .....	123
4.14 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? .....	124
4.15 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? .....	124
4.16 如何处理部署组件时失败, 报错日志详情显示 connection was closed? .....	125
4.17 如何处理部署虚拟机组件时遇到 ECS 错误? .....	125
4.18 如何处理虚拟机部署端口访问不通的错误? .....	125
4.19 如何处理微服务应用名称与组件所属应用名称不同的问题? .....	126
4.20 如何解释微服务名称与组件名称不同的原因? .....	128
4.21 微服务引擎专享版数据恢复失败.....	128

# 1 产品介绍

## 1.1 什么是应用管理与运维平台（ServiceStage）

应用管理与运维平台（ServiceStage）是面向企业的应用管理与运维平台，提供应用发布、部署、监控与运维等一站式解决方案。支持Java、Go、PHP、Node.js、Python、Docker、Tomcat等运行环境。支持Web应用，以及Apache ServiceComb、Spring Cloud、Dubbo、Mesher服务网格等微服务应用和通用应用，让企业应用上云更简单。

ServiceStage主要包含如下能力：

- 应用管理：支持应用生命周期管理、环境管理。
- 微服务应用接入：支持Java Chassis、Go Chassis、Spring Cloud、Dubbo微服务框架；支持ServiceComb 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。
- 应用运维：通过日志、监控、告警支持应用运维管理。

### 应用开发

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。
- 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。

### 应用托管

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。

- 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。

## 微服务治理

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，您可以针对微服务进行相关的治理，支持的服务治理策略见表1-1。

表 1-1 服务治理策略

名称	说明
负载均衡	当出现访问量和流量较大，一台服务器无法负载的情况下，可以通过设置负载均衡的方式将流量分发到多个服务器均衡处理，从而降低时延，防止服务器过载。
限流	对服务实例设置限流，对当前服务实例的每秒请求数量超过设定的值时，当前服务实例就不再接受其他对象的调用请求。
容错	容错是服务实例出现异常时的一种处理策略，出现异常后按照定义的策略进行重试或访问新的服务实例。
降级	降级是容错的一种特殊形式，当出现服务吞吐量巨大，资源不够用等情况，可使用降级机制关掉部分不重要、性能较差的服务，避免占用资源，以保证主体业务功能可正常使用。
熔断	当由于某些原因导致服务出现了过载现象，为避免造成整个系统故障，可采用熔断来进行保护。
错误注入	错误注入策略用于测试微服务的容错能力，可以让用户知道，当出现延时或错误时，系统是否能够正常运行。
黑白名单	黑白名单是为了改变网络流量所经过的途径而修改路由信息的技术。

## 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1-2所示的优势。

表 1-2 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>

应用生命周期	传统平台	使用ServiceStage平台
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> <li>● 全手工部署</li> </ul>	开发者只需使用ServiceStage+任意源码仓库，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，微服务应用可以很好的解决上述问题。

### 价值

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

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

ServiceStage可承载微服务应用的全生命周期管理。支持Java、Go、PHP、Node.js、Docker、Tomcat等运行环境，可无侵入托管Apache ServiceComb、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提供了业内优秀的微服务应用解决方案，具有以下优势：

- 支持原生ServiceComb、Spring Cloud、Dubbo和服务 Mesh多种微服务框架，支持双栈模式（SDK和服务网格互通），无需更改业务代码直接托管上云。
- API First，支持基于Swagger的API管理。
- 支持多语言微服务，如JAVA、Go、.NET、Node.js、PHP、Python等。
- 提供服务中心、配置中心、仪表盘、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、ServiceComb 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

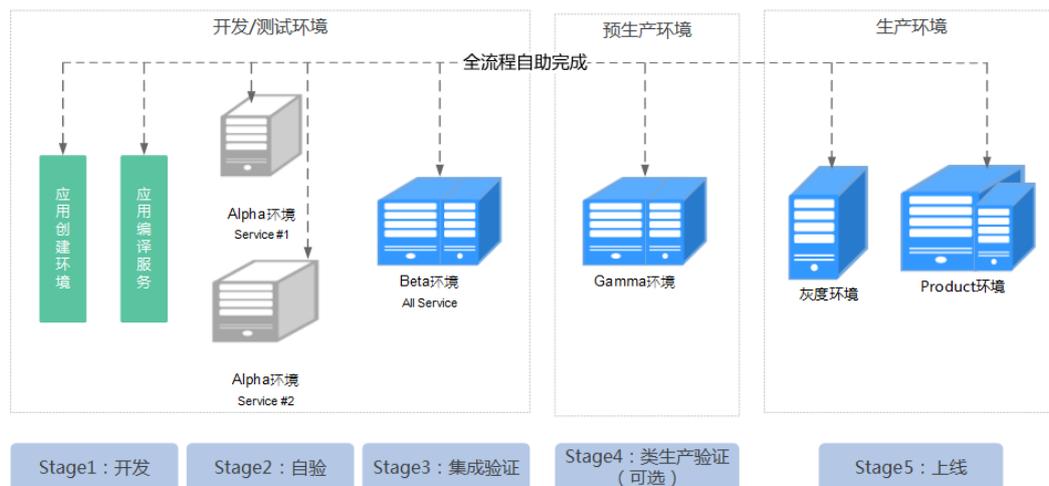
由人工执行变成了自动化执行，减少错误出现，提高工作效率。

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 1-1 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。

## 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 名词解释

### 环境

环境是用于应用部署和运行的计算、存储、网络等基础设施的集合。ServiceStage把相同VPC下的基础资源（如CCE集群、ECS等）加上可选资源（如ELB、DCS等）实例组合为一个环境，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，减少具体基础设施运维管理的复杂性。

### 基础设施

在ServiceStage中，基础设施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。

## 1.5 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，最终用户在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1-3。

表 1-3 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1-4。

表 1-4 微服务引擎专享版限制（最高规格）

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

## 1.6 规格说明

表 1-5 功能说明

功能	规格说明	
管理规模	最大支持应用组件实例个数 5000个	
微服务	微服务引擎	如不能满足业务需求，请联系技术支持申请扩大配额。
	JAVA/Go语言微服务开发SDK	支持
	Spring Cloud微服务接入	
	Service Mesh微服务接入	
	服务注册中心	
	服务配置中心	
实时仪表盘		

功能		规格说明
	负载均衡	
	服务限流	
	服务降级	
	服务容错	
	服务熔断	
	错误注入	
	黑白名单	
应用生命周期管理	多语言应用管理（Java/Go/PHP/Node.js/Tomcat/Docker）	支持
	应用生命周期管理（伸缩、升级、回退、启动、停止、重启、删除）	
	应用基础监控（运行状态、CPU、内存、磁盘使用率等）	
	虚拟机部署，当前仅支持Java、Tomcat、Node.js	
	容器部署	
	访问权限控制	
	应用域名管理	
	弹性伸缩	
	事件分析	
	日志分析	
	阈值告警	
软件中心	Docker软件包管理	
	仓库权限管理	
应用基础运维	日志存储与搜索	1TB/月，存储时长30天
	日志分析	500GB
	查询日志数据API调用	10GB/月
	主机监控	200VM，指标存储时长1年
	秒级监控指标数量	2000个
	自定义指标	500个

功能	规格说明
查询指标数据API调用	500万次/月
事件和告警总条数	50万条/月，存储时长30天
智能阈值规则	50条

## 微服务引擎服务数限制说明

微服务引擎分为微服务引擎专业版和微服务引擎专享版两种规格：

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

每种规格支持最大服务数说明如下。

表 1-6 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	1000
微服务引擎专享版	100
	500
	2000

## 1.7 权限管理

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

通过IAM，您可以在云帐号中给员工创建IAM用户，并使用策略来控制员工对云资源的访问范围。例如您的员工中有负责软件开发的人员，您希望开发人员拥有ServiceStage的使用权限，但是不希望开发人员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，控制开发人员对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是提供权限管理的基础服务。关于IAM的详细介绍，请参见《IAM产品介绍》。

## ServiceStage 权限

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

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

根据授权精细程度分为角色和策略。

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

如表1-7所示，包括了ServiceStage的所有系统权限。

表 1-7 系统权限说明

系统角色/策略名称	描述	类别	依赖关系
ServiceStage FullAccess	ServiceStage服务所有权限。	系统策略	无
ServiceStage Developer	ServiceStage开发者权限，拥有应用、组件、环境的操作权限，但无审批权限和基础设施创建权限。	系统策略	无
ServiceStage ReadOnlyAccess	ServiceStage只读权限。	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

如果表1-7所列的这些权限不满足实际需求，可以在这个基础上[自定义策略](#)。

表1-8列出了ServiceStage常用操作与系统权限的授权关系，您可以参照该表选择合适的系统权限。

表 1-8 常用操作与系统权限之间的关系

操作	ServiceStage ReadOnlyAccess	ServiceStage Developer	ServiceStage FullAccess	CSE Viewer	CSE Admin
创建应用	x	√	√	x	x
修改应用	x	√	√	x	x
查询应用	√	√	√	x	x
删除应用	x	√	√	x	x
创建组件	x	√	√	x	x
查询组件	√	√	√	x	x
部署组件	x	√	√	x	x
维护组件	x	√	√	x	x
删除组件	x	√	√	x	x
创建构建工程	x	√	√	x	x
修改构建工程	x	√	√	x	x
查询构建工程	√	√	√	x	x
启动构建工程	x	√	√	x	x
删除构建工程	x	√	√	x	x
创建流水线	x	√	√	x	x

操作	ServiceStage ReadOnlyAccess	ServiceStage Developer	ServiceStage FullAccess	CSE Viewer	CSE Admin
修改流水线	x	√	√	x	x
查询流水线	√	√	√	x	x
启动流水线	x	√	√	x	x
克隆流水线	x	√	√	x	x
删除流水线	x	√	√	x	x
新建仓库授权	x	√	√	x	x
修改仓库授权	x	√	√	x	x
查询仓库授权	√	√	√	x	x
删除仓库授权	x	√	√	x	x
创建微服务引擎	x	x	√	x	√
维护微服务引擎	x	x	√	x	√
查询微服务引擎	√	√	√	√	√

操作	ServiceStage ReadOnlyAccess	ServiceStage Developer	ServiceStage FullAccess	CSE Viewer	CSE Admin
删除微服务引擎	x	x	√	x	√
微服务注册	√	√	√	x	√
微服务配置	√	√	√	x	√
微服务治理	x	√	√	x	√

### 说明

SWR未支持细粒度权限，相关权限需要另外授权。

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

- CSE细粒度权限依赖说明请参见表1-9。
- 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。

权限名称	权限描述	权限依赖	应用场景
cse:engine:delete	删除引擎	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</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> </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。
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。

权限名称	权限描述	权限依赖	应用场景
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	无	查询应用信息。
servicestage:app:create	创建应用	无	创建应用。
servicestage:app:modify	更新应用	无	更新应用。
servicestage:app:delete	删除应用	无	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	无	创建环境。
servicestage:environment:modify	更新环境	无	更新环境。
servicestage:environment:delete	删除环境	无	删除环境。
servicestage:pipeline:get	查看流水线信息	无	查看流水线信息。
servicestage:pipeline:create	创建流水线	无	创建流水线。
servicestage:pipeline:modify	修改流水线	无	修改流水线。
servicestage:pipeline:delete	删除流水线	无	删除流水线。
servicestage:pipeline:list	查看流水线列表	无	查看流水线列表
servicestage:pipeline:execute	执行流水线	无	执行流水线。

权限名称	权限描述	权限依赖	应用场景
servicestage:assembling:get	查看构建信息	无	查看构建信息。
servicestage:assembling:create	创建构建	无	创建构建。
servicestage:assembling:modify	修改构建	无	修改构建。
servicestage:assembling:delete	删除构建	无	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

## 概述

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

环境是把相同VPC下的基础资源（如CCE集群、ECS等）加上可选资源（如ELB、DCS等）组合而成，部署应用和组件时需要选择环境进行部署，选择后会自动加载此环境所包含的资源。

应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。比如：典型的ERP系统就是一个应用，它一般由会计、财务、生产控制、物流、采购、分销、库存等模块组成，这些模块之间相互有紧密的联系，每个模块就是一个组件。

组件是组成应用的某个业务特性实现，如上面的ERP系统举例。在微服务应用场景下，每个组件都有独立的软件包，可独立部署运行，部署的组件称为一个组件实例，一个组件也可以有多个组件实例形成一个集群来保障应用和组件的高可靠性。针对应用的组件实例进行启停、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。

下面会示例讲解如何搭建环境，如何创建微服务应用和组件，供用户体验ServiceStage的基本功能。

## 前提条件

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

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

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

## 准备软件包

**步骤1** 下载[微服务demo](#)源码到本地并解压。

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

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

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

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

上传对象，请参考[上传文件](#)。

----结束

## 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”。

**步骤3** 单击“确定”。

----结束

## 创建环境

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

**步骤2** 参照下表设置环境信息。

参数	参数说明
环境名称	输入环境名称（例如：test-env）。
虚拟私有云(VPC)	选择 <a href="#">前提条件</a> 中已准备好的虚拟私有云VPC。 <b>说明</b> 选定VPC后，会加载该VPC下的基础资源和可选资源供选择，不在该VPC下的资源无法选择。
基础资源	选择该VPC下的基础资源，本例使用 <a href="#">前提条件</a> 中已创建的云容器引擎(CCE)。
可选资源	选择该VPC下可选资源，本例使用 <a href="#">前提条件</a> 中已创建的微服务引擎专享版。

**步骤3** 单击“立即创建”，完成环境创建。

----结束

## 新建应用

**步骤1** 登录ServiceStage控制台，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：test-servicestage）。
2. “描述”：（可选）输入应用描述。

**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控制台，选择“应用管理 > 应用列表”。

**步骤2** 选择test-servicestage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”，“选择模板”选择“ServiceComb”。单击“下一步”。

**步骤4** 设置组件基本信息：

1. “组件名称”：输入组件名称（例如：test-cse）。
2. “上传方式”：单击“选择软件包”，选择[准备软件包](#)操作过程中已经上传到OBS桶中的servicecomb-0.0.1-SNAPSHOT.jar软件包。

**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. 其他参数使用默认配置。

**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控制台，选择“应用管理 > 应用列表”。

**步骤2** 单击已经创建的test-servicestage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test-cse，在“操作”栏单击“部署”。

**步骤4** 参照下表设置基本配置，参数前面带\*号的是必须设置的参数。

参数	参数说明
*环境	选择已经创建的环境test-env。
*部署版本	默认为1.0.0。

参数	参数说明
*部署系统	选择“云容器引擎”。
*基础资源	使用环境test-env中的CCE资源（自动加载）。
*实例数量	设置为1。
资源配额	使用默认配置。

**步骤5** 单击“下一步 组件配置”，设置组件配置。

1. 选择“微服务引擎”，默认使用环境test-env中的微服务引擎专享版。

#### 📖 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。

2. 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。

**步骤7** 单击“部署”，部署组件。

----结束

## 确认部署结果

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 微服务引擎（CSE）”。

**步骤2** 选择**部署组件**时使用的微服务引擎专享版，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，且微服务实例数为**部署组件**时设置的实例数量，则部署成功。

----结束

## 访问应用

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

**步骤2** 单击test-servicestage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** 单击组件名称test-cse，进入组件“概览”页。

**步骤5** 单击“访问方式”。

**步骤6** 单击“TCP/UDP路由配置”区域的“添加服务”，参考下表设置参数。

参数	参数说明
服务名称	使用默认。

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

**步骤7** 单击“确定”，生成访问地址。

**图 2-1** 访问地址



**步骤8** 单击图2-1所示“访问地址”列下的访问地址，访问应用。

返回如下结果：

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

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

可以返回如下结果：

```
"ServiceStage"
```

----结束

## 应用运维

**步骤1** 登录ServiceStage控制台，选择“应用管理 > 应用列表”，单击test-servicestage应用名称，进入应用“概览”页。

**步骤2** 在“组件列表”，单击test-cse组件名称，进入组件“概览”页，在下方“环境”视图中可以查看组件版本。

**步骤3** 单击组件名称，进入组件版本详情页，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

# 3 用户指南

## 3.1 概述

应用管理与运维平台（ServiceStage）是面向企业的应用管理与运维平台，提供应用发布、部署、监控与运维等一站式解决方案。支持Java、Go、PHP、Node.js、Python、Docker、Tomcat等运行环境。支持Web应用，以及Apache ServiceComb、Spring Cloud、Dubbo、Mesher服务网格等微服务应用和通用应用，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、存储、网络等基础设施的集合。 提供环境创建、编辑、删除等功能，以卡片形式展示已有环境包含的资源信息。

类别	说明
持续交付	提供工程构建、发布等功能。 <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>● 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><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></ul>
基础设施	提供应用基础设施管理功能，如微服务引擎（CSE）、虚拟机Agent管理（VMAgent）等。 其中，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。

### 📖 说明

虚拟机Agent管理功能依赖ECS、AOM服务的安装，如果服务未安装，则功能不可使用。

## 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[ServiceStage系统策略](#)。

## 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“应用管理与运维平台 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：[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控制台，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“基础设施 > 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[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。

## 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支持单个用户最多创建可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. “描述”：（可选）输入应用描述。

**步骤3** 单击“确定”，创建应用。

----结束

#### 添加环境变量

环境是用于应用部署和运行的计算、存储、网络等基础设施的集合。ServiceStage把相同VPC下的基础资源（如CCE集群、ECS等）加上可选资源（如ELB、DCS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，减少具体基础设施运维管理的复杂性。

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。

**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”为您提供了组件类型、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
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	支持的软件包
Java8	Java Chassis	源码仓库、模板、Jar包
Tomcat8		源码仓库、模板、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
Node.js8		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

支持的运行时	支持的框架/服务网格	支持的软件包
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、模板、Jar包
Tomcat8		源码仓库、模板、War包
Docker		不需要设置该参数

## Web 应用组件说明

支持的运行时	支持的软件包
Java8	源码仓库、模板、Jar包
Nodejs8	源码仓库、模板、Zip包
Php7	源码仓库、模板、Zip包
Tomcat8	源码仓库、模板、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	支持的软件包
Java8	源码仓库、模板、Jar包
Tomcat8	源码仓库、模板、War包
Node.js8	源码仓库、模板、Zip包
Php7	源码仓库、模板、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 <a href="#">上传文件</a> 。
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 <a href="#">上传文件</a> 。
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 <a href="#">上传文件</a> 。
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。
模板	参考 <a href="#">仓库授权</a> 创建授权，设置组织和仓库名称。 ServiceStage提供了组件模板，可以根据需要选择模板进行快速创建应用，同时在设置的代码仓库中生成开发工程，说明请参考 <a href="#">模板框架说明</a> 。

## 模板框架说明

运行时	模板提供的框架	框架说明
Java8	CSE-Java (SpringMVC)	基于ServiceComb微服务开发框架，支持SpringMVC注解，使用SpringMVC风格开发微服务。
	CSE-Java (JAX-RS)	基于ServiceComb微服务开发框架，支持JAX-RS注解，使用JAX-RS模式开发微服务。
	CSE-Java (POJO)	基于ServiceComb微服务开发框架，支持接口和接口实现的开发模式，使用透明RPC开发微服务。

运行时	模板提供的框架	框架说明
Tomcat8	SpringBoot-Webapp-Tomcat	Web应用，运行于独立部署的Web服务器。
	SpringBoot-WebService-Tomcat	Web Service，运行于独立部署的Web服务器。
Nodejs8	Express	高度包容、快速而极简的Node.js Web框架。
	Koa	基于Node.js平台的下一代Web开发框架。
Php7	Laravel	简洁、优雅，为WEB创造的PHP开发框架。
	Slim	快速、轻量的微型PHP框架。

### 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”或者“模板”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[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。

参数	参数说明
*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。 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<a href="#">添加节点标签</a>。</li></ul></li><li><b>须知</b> “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。</li><li>选择“Jar包”、“War包”<ul style="list-style-type: none"><li><b>说明</b><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></li><li>1. 选择“上传方式”： 将软件包上传至OBS对象存储中，请参考<a href="#">上传文件</a>。</li><li>2. （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<a href="#">添加节点标签</a>。</li></ul></li><li>选择“模板”时，设置如下参数：<ol style="list-style-type: none"><li>选择框架。ServiceStage提供了模板框架，可以根据需要选择模板快速创建应用组件。</li><li>设置“模板代码归档”，请参考<a href="#">仓库授权</a>创建授权，设置用户名/组织和仓库名称。</li></ol></li></ul>

#### 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

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”或者“模板”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[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：

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 3-4 组件基本信息

参数	参数说明
*组件名称	组件对应的名称。

参数	参数说明
*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”： 将软件包上传至OBS对象存储中，请参考<a href="#">上传文件</a>。</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> <li>运行时为“Nodejs8”、“Php7”或者“Python3”时，选择“Zip包”。</li> </ul> <ul style="list-style-type: none"> <li>选择“模板”，设置如下参数： <ol style="list-style-type: none"> <li>选择框架，ServiceStage提供了模板框架，可以根据需要选择模板快速创建应用组件。</li> <li>设置“模板代码归档”，参考<a href="#">仓库授权</a>创建授权，设置组织和仓库名称。</li> </ol> </li> </ul> <p><b>说明</b></p> <p>当<a href="#">步骤5</a>选择的框架/服务网格是“Mesher”或者“Spring Cloud”时，本参数无效。</p>
*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。</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”或者“模板”时，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> <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<a href="#">添加节点标签</a>。</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>（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。 根据业务需要选择“组织”和“选择集群”参数。</li> </ul> <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<a href="#">添加节点标签</a>。</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”或者“模板”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[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[微服务组件说明](#)。

**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。

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”： 将软件包上传至OBS对象存储中，请参考<a href="#">上传文件</a>。</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> <li>运行时为“Nodejs8”、“Php7”或者“Python3”时，选择“Zip包”。</li> </ul> <ul style="list-style-type: none"> <li>选择“模板”，设置如下参数： <ol style="list-style-type: none"> <li>选择框架，ServiceStage提供了模板框架，可以根据需要选择模板快速创建应用组件。</li> <li>设置“模板代码归档”，参考<a href="#">仓库授权</a>创建授权，设置组织和仓库名称。</li> </ol> </li> </ul> <p><b>说明</b></p> <p>当<a href="#">步骤4</a>选择的运行时是“Python3”时，本参数无效。</p>
*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。 根据业务需要选择“编译命令”、“组织”和“选择集群”参数。</li> </ul> <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<a href="#">添加节点标签</a>。</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>（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。 根据业务需要选择“组织”和“选择集群”参数。</li> </ul> <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<a href="#">添加节点标签</a>。</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

### 3.3.2.5 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”或者“模板”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[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[微服务组件说明](#)。

**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<a href="#">上传文件</a>。</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><li>- 运行时为“Nodejs8”、“Php7”或者“Python3”时，选择“Zip包”。</li></ul> <ul style="list-style-type: none"><li>选择“模板”，设置如下参数：<ol style="list-style-type: none"><li>选择框架，ServiceStage提供了模板框架，可以根据需要选择模板快速创建应用组件。</li><li>设置“模板代码归档”，参考<a href="#">仓库授权</a>创建授权，设置组织和仓库名称。</li></ol></li></ul> <p><b>说明</b></p> <p>当<a href="#">步骤4</a>选择的运行时是“Python3”时，本参数无效。</p>
*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。 根据业务需要选择“编译命令”、“组织”和“选择集群”参数。</li></ul> <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<a href="#">添加节点标签</a>。</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>（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。 根据业务需要选择“组织”和“选择集群”参数。</li></ul> <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<a href="#">添加节点标签</a>。</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。

##### 虚拟机部署

创建的组件，仅当运行时选择了Java8、Tomcat8或者Nodejs8时，支持虚拟机部署。

#### 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[上传文件](#)。
  - 将镜像包上传至镜像仓库，请参考[上传镜像](#)。

##### 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。

参数	参数说明
*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、虚拟机部署。 详情请参见 <a href="#">部署方式说明</a> 。
*资源类型	只有“部署系统”选择“虚拟机部署”时，该参数才生效。 支持弹性云服务器。
*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。 <b>说明</b> <ul style="list-style-type: none"><li>选择虚拟机部署时无此参数。</li><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，为了保证更大程度可用，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”。 <b>说明</b> <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><li>选择虚拟机部署时无此参数。</li></ul>
组件状态	根据需要设置组件状态。 <b>说明</b> <p>“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。</p>

**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：
  - 选择镜像。支持多容器，可以单击“添加容器”增加镜像。
  - 设置“镜像版本”。
  - 输入“容器名称”。
  - （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”。
  - （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[创建负载均衡器](#)。
  - ii. (可选) 设置“HTTPS”

若开启HTTPS，单击“使用已有”选择已经创建的证书。  
若证书不存在，请单击“新创建”创建新的服务器证书。创建服务器证书请参考[创建证书](#)。
  - 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](#)。
- 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、模板时，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[创建负载均衡器](#)。
    - ii. (可选) 设置“HTTPS”

若开启HTTPS，单击“使用已有”选择已经创建的证书。  
若证书不存在，请单击“新创建”创建新的服务器证书。创建服务器证书请参考[创建证书](#)。

- 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. 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](#)。
- d. （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。
- e. （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”。
- 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“自定义监控指标”，请参考[设置应用组件自定义指标监控](#)。
- 其他类型的组件，且[步骤4](#)中的“部署系统”为“虚拟机部署”执行以下操作：
  - a. （可选）开启“公网访问”
    - i. 设置“公网ELB”  
选择已经创建的负载均衡。  
若不存在，请单击“新增ELB”创建新的负载均衡，详情请参考[创建负载均衡器](#)。

- ii. (可选) 设置“HTTPS”  
若开启HTTPS, 单击“使用已有”选择已经创建的证书。  
若证书不存在, 请单击“新创建”创建新的服务器证书。创建服务器证书请参考[创建证书](#)。
- 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](#)。
- b. (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](#)。
- c. (可选) 设置“环境变量”  
请参考[设置应用环境变量](#)。
- d. (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。
- e. (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”。

**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[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

- 不支持为虚拟机部署的组件实例创建流水线。
- 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： <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

## 3.3.5 应用高级设置

### 3.3.5.1 设置应用环境变量

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CCE和虚拟机部署时环境变量的配置方法。

## 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. <b>可选:</b> 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><li>4. 在“变量/变量引用”的下拉框中选择对应的配置项。</li></ol>
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

## 虚拟机部署方式

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

**步骤1** 在“组件配置”界面，单击“添加环境变量”。

**步骤2** 输入“Key”和“Value”值，单击“确定”。

----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 Command Args[1] Args[2]…（Command为系统命令或者用户自定义可执行程序，如果未指定路径则在默认路径下需找可执行程序），如果需要执行多条命令，建议采用将命令写入脚本执行的方式。</p> <p>需要执行的命令示例如下：</p> <pre>exec:   command:   - /install.sh   - install_agent</pre> <p>请在执行脚本中填写： /install.sh install_agent。 这条命令表示组件部署成功后将执行install_agent安装。</p>
HTTP请求方式	<p>发起一个HTTP调用请求。配置参数如下：</p> <ul style="list-style-type: none"><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、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。</li><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。</li></ul>

项目	场景
<b>主机路径挂载</b>	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。 <b>须知</b> 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。
<b>临时路径挂载</b>	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。
<b>配置项挂载</b>	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。
<b>密钥挂载</b>	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”。 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>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>选择存储子类型。 高IO：指由SAS存储介质构成的云硬盘。 普通IO：指由SATA存储介质构成的云硬盘。 超高IO：指由SSD存储介质构成的云硬盘。</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>

参数	说明
添加容器挂载	<ol style="list-style-type: none"><li>1. 设置“挂载路径”：输入数据存储挂载到应用上的路径。 <b>须知</b><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存储。</li><li>- 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li></ul></li></ol>

**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3-11设置参数。

表 3-11 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	<ol style="list-style-type: none"><li>1. 设置“挂载路径”：输入数据存储挂载到应用上的路径。 <b>须知</b><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存储。</li><li>- 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li></ul></li></ol>

**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3-12设置参数。

表 3-12 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	<ol style="list-style-type: none"><li>设置“挂载路径”：输入数据存储挂载到应用上的路径。 <b>须知</b><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存储。</li><li>读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</li></ul></li></ol>

**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3-13设置参数。

表 3-13 配置项挂载

参数	说明
本地存储类型	选择“配置项挂载”。

参数	说明
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	<p>1. 设置“挂载路径”：输入数据存储挂载到应用上的路径。</p> <p><b>须知</b></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存储。</li> <li>- 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</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3-14](#)设置参数。

**表 3-14** 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	<p>1. 设置“挂载路径”：输入数据存储挂载到应用上的路径。</p> <p><b>须知</b></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存储。</li> <li>- 读写：可修改应用路径中的数据存储，应用迁移时新写入的数据不会随之迁移，会造成数据丢失。</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[创建Redis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 3-15 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。 目前支持的代码检查插件有：checkstyle、findbugs、pmd。</li></ul> <p><b>说明</b> 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

- 不支持为虚拟机部署的实例创建流水线。
- 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[创建CCE集群](#)。
- 已创建密钥所在命名空间，请参考[创建命名空间](#)。

#### 创建密钥

**步骤1** 登录ServiceStage控制台，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 3-16 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>- Opaque：一般密钥类型。当密钥配置文件中未作显式设定时，默认的密钥类型是Opaque。</li><li>- kubernetes.io/dockerconfigjson：存放拉取私有仓库镜像所需的认证信息。</li><li>- IngressTLS：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。

参数	参数说明
*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”中写作者或者修改上传的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==为编码后的内容。

## 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3-16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

- 已创建需要使用配置项的CCE集群。  
创建CCE集群，请参考[创建CCE集群](#)。
- 已创建配置项所在命名空间，请参考[创建命名空间](#)。

## 创建配置项

**步骤1** 登录ServiceStage控制台，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 3-17 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](#)。

- a. 在“所属集群”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ConfigMap类型资源文件后，单击“打开”。
- c. 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	1. 单击目标配置项“操作”列的“更新”。 2. 根据表3-17更改信息。 3. 单击“更新配置”。
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	1. 勾选需要删除的配置项。 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。
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

环境是用于应用部署和运行的计算、存储、网络等基础设施的集合。ServiceStage把相同VPC下的基础资源（如CCE集群、ECS等）加上可选资源（如ELB、DCS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，减少具体基础设施运维管理的复杂性。

ServiceStage支持单个用户最多可创建1000个环境。

### 创建环境

**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。
描述	环境说明信息。
*虚拟私有云 (VPC)	选择基础设施所在VPC。 新建VPC，请参考 <a href="#">创建虚拟私有云和子网</a> 。 <b>说明</b> 选定VPC后，会加载该VPC下的基础设施资源供选择，不在该VPC下的资源无法选择。
*基础资源	至少选择云容器引擎（CCE）、弹性云服务器（ECS）等基础设施中的一种资源，可以多选。
可选资源	可以根据需要选择已有弹性负载均衡（ELB）、弹性公网IP（EIP）、分布式缓存（DCS）、微服务引擎（CSE）。

**步骤3** 单击“立即创建”，完成环境创建。

环境创建成功后，可以在“环境管理”页的资源卡片查看环境信息。

----结束

## 修改环境

**步骤1** 登录ServcieStage控制台，进入“环境管理”，单击待修改环境卡片上的“编辑”。

**步骤2** 编辑环境配置信息，其中带“\*”标志的参数为必填参数。

参数	参数说明
*环境名称	输入新环境名称。
描述	环境说明信息。
*虚拟私有云 (VPC)	虚拟私有云不可修改，只能增加已选VPC下的资源，不在该VPC下的资源无法选择。
*基础资源	可以增删已有云容器引擎（CCE）、弹性云服务器（ECS）资源。
可选资源	可以增删弹性负载均衡（ELB）、弹性公网IP（EIP）、分布式缓存（DCS）、微服务引擎（CSE）等资源。

**步骤3** 单击“保存”，完成环境修改。

环境修改成功后，可以在“环境管理”页的资源卡片查看环境信息。

----结束

## 删除环境

### 📖 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。

**步骤1** 登录ServiceStage控制台，进入“环境管理”，单击已有环境卡片上的“删除”。

**步骤2** 在确认界面单击“确定”，完成环境删除。

----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”，可查看到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. 设置“HTTPS”。  
若开启HTTPS，单击“使用已有”选择已经创建的证书。  
若证书不存在，请单击“新创建”创建新的服务器证书。创建服务器证书请参考[创建证书](#)。
- 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3-18设置阈值规则参数信息。其中带“\*”标志的参数为必填参数。

表 3-18 阈值规则参数说明

参数	说明
*阈值名称	阈值规则的名称。 <b>说明</b> 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3-19中的步骤来管理阈值告警。

表 3-19 相关操作

操作	说明
修改阈值告警	当您发现当前的阈值规则设置不合理时，您可以参考如下操作对阈值规则进行修改，以便更好的满足您的业务需求。 <ol style="list-style-type: none"><li>单击阈值告警列表“操作”列的“修改”。</li><li>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>单击“修改”。</li></ol>
删除阈值告警	随着业务的变动，当您发现当前的阈值规则已不再需要时，您可以参考如下操作删除阈值规则，以便释放更多的阈值规则资源。 <ol style="list-style-type: none"><li>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>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>在下拉列表框中选择时间段。</li><li>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>单击，或在键盘上按“Enter”键。 也可单击“高级搜索”设置搜索条件，单击“搜索”进行查询。</li></ol>
查看阈值告警	在已设的连续周期内，当指标数据满足阈值条件时，系统会发送阈值告警。 在阈值告警列表中查看该告警。
查看数据不足事件	在已设的连续周期内，当没有指标数据上报时，系统会发送数据不足事件。 可单击“事件”页签，在事件列表中查看该事件。

### 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。

- 弹性伸缩：支持告警、定时、周期三种策略。配置完成后可基于资源变化、固定时间、固定周期自动触发实例的增减。

- 手动伸缩：配置完成后立即触发实例的增减。

## 优雅缩容策略

您可以设置一个缩容时间窗，用于在应用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。

**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

## 手动伸缩

**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

## 弹性伸缩

### 说明

- CCE集群，1.15以上版本不支持弹性伸缩。
- 虚机部署不支持弹性伸缩。

**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 3-20 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><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>

参数	参数说明
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。

表 3-21 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 3-22 添加周期策略

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“周期策略”。
触发时间	设置策略触发时间。

参数	参数说明
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

### 3.5.6 设置应用组件实例调度策略

ServiceStage支持丰富的调度策略，包括静态的全局调度策略，以及动态的运行时时调度策略，您可以根据需要自由组合使用这些策略来实现业务需求。

#### 概念阐述

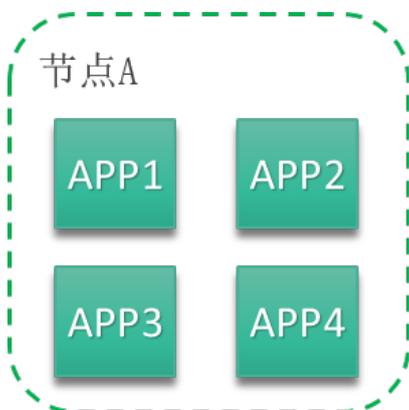
- 应用与可用区的亲和性
  - 亲和：决定应用组件部署在特定的可用区中。
  - 反亲和：决定应用组件不能部署在特定的可用区中。
- 应用与节点间的亲和性
  - 亲和：决定应用组件部署在某些特定的主机中。
  - 反亲和：决定应用组件不能部署在某些特定的主机中。

- 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。

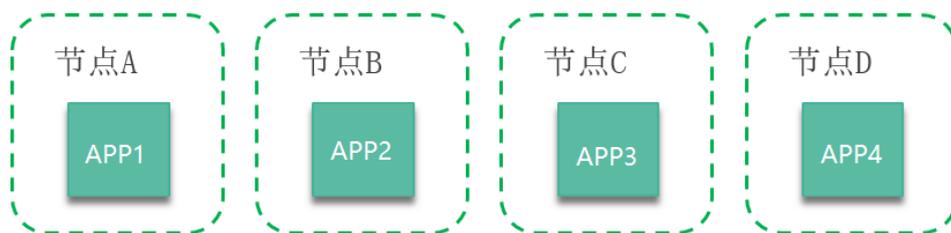
**图 3-3** 应用间亲和



- 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	<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>

目的	步骤
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”。 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。</li> </ol>
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”。 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。</li> </ol>

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
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”。 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。</li> </ol>

目的	步骤
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”。 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。</li> </ol>

**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”，可查看到所有应用。

**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”。 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。</li> </ol>

目的	步骤
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”。 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。</li></ol>

**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

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。

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

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。

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
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"]。 - 如果不设置，则ServiceStage会获取所有的自定义指标数据。 - 如果设置，例如设置为["cpu_usage","mem_usage"]，则ServiceStage会对自定义指标进行过滤，只获取cpu_usage、mem_usage指标数据。	否

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[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”, 可查看到所有应用。

**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

参数	说明	是否必须设置
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[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”, 可查看到所有应用。

**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	<ol style="list-style-type: none"> <li>设置“挂载路径”：输入数据存储挂载到应用上的路径。  <b>须知</b> <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“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> <li>PodUID/ContainerName：Pod的ID/容器名称。</li> <li>PodName/ContainerName：Pod名称/容器名称。</li> </ul> </li> <li>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> </li> </ol>

**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[查看日志文件](#)。

----结束

## 3.5.10 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。

## 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

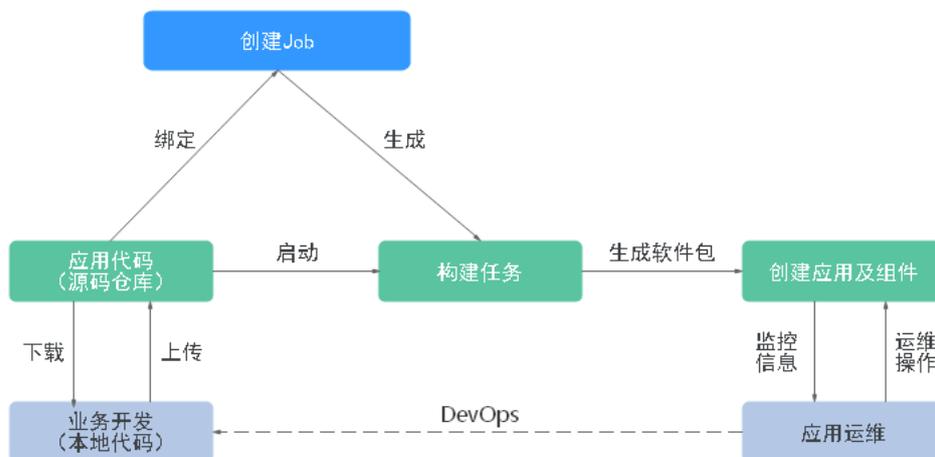
## 3.6 持续交付

### 3.6.1 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。

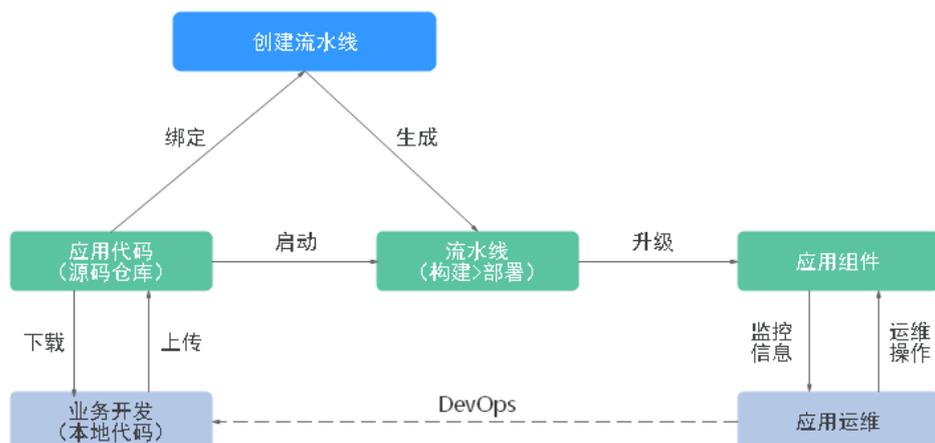
图 3-5 创建构建任务



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

## 操作步骤

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

**步骤2** 设置基本信息：

1. 输入“名称”。
2. （可选）输入工程“描述”。
3. 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。
4. 设置“构建集群”，选择用于构建的集群。
5. （可选）选择“过滤节点标签”，可以通过节点标签将构建任务下发到固定节点上。新增过滤标签请参考[添加节点标签](#)。
6. 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。
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 3-23 维护操作

操作	操作说明
查看详情/构建历史	<ol style="list-style-type: none"><li>单击目标构建工程名称，在“构建记录”下查看构建历史。</li><li>单击构建记录名称，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。目前支持的代码检查插件有：checkstyle、findbugs、pmd。</li></ol> <p><b>说明</b> 仅Maven构建工程支持代码检查。</p>
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	选择基于源码构建的目标构建工程，单击“操作”列的“分支/tag构建”。 <ol style="list-style-type: none"><li>选择“分支/Tag”。</li><li>从下拉框中选择相应的“分支”或“Tag”。</li><li>输入对应分支/Tag的CommitId。</li><li>单击“确定”。</li></ol>
编辑	选择用户创建的目标构建工程，单击“操作”列的“更多 > 编辑”，编辑构建工程。
删除	<ol style="list-style-type: none"><li>选择用户创建的目标构建工程，单击“操作”列的“更多 &gt; 删除”。</li><li>单击“确定”。</li></ol>

### 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

- 已经创建集群，请参考[创建CCE集群](#)操作。
- 已为构建节点绑定弹性IP，请参考[为弹性云服务器申请和绑定弹性公网IP](#)。

#### 操作步骤

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

**步骤2** 输入“Job名称”。

**步骤3** （可选）输入Job“描述”。

**步骤4** 设置“包来源”。

支持以下上传方式：

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

单击“选择软件包”，选择对应的软件包。

**步骤5** 选择构建方式。

- 系统默认
  - a. 选择基础镜像语言，需与[步骤4](#)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4](#)中选择的软件包编译语言一致。

**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”。

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

**步骤9** 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[应用组件部署](#)。

## 维护操作

表 3-24 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. （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 3-25 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</a> ~ <a href="#">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> 。

模板名称	说明	操作说明
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</a> ~ <a href="#">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”。  
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 3-26 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。

触发策略	触发模式	说明
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”、“生效时间”、“触发周期”。</li> <li>5. 单击“确定”。</li> </ol>

## 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 3-27 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 授权方式支持：OAuth/私人令牌。</li><li>● Gitee 授权方式支持：OAuth/私人令牌。</li><li>● Bitbucket 授权方式支持：OAuth/密码/私有Bitbucket。</li><li>● GitLab 授权方式支持：OAuth/私人令牌。</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[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** 单击“页面上传”，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。

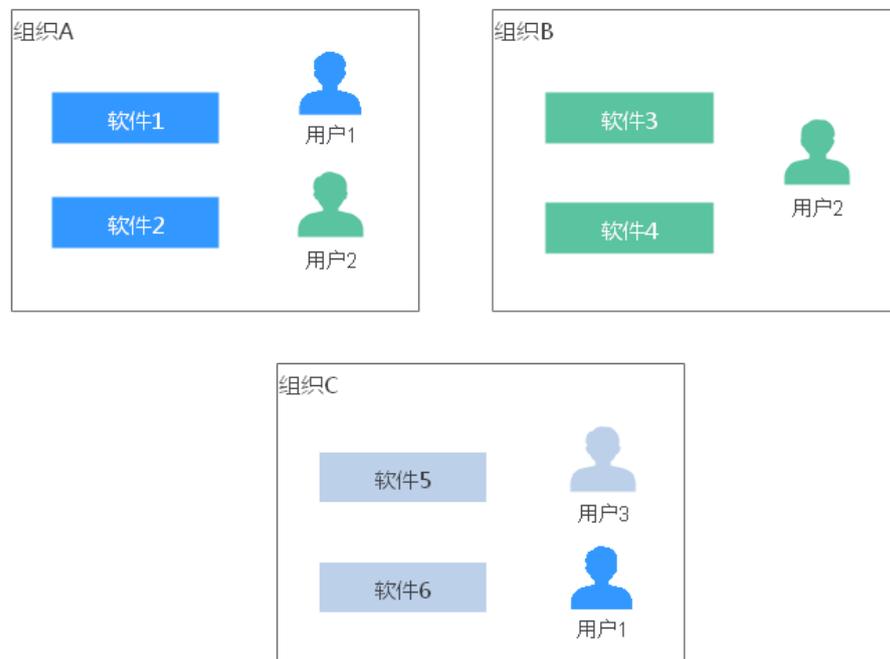
----结束

## 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）。

图 3-7 组织



## 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 基础设施

### 3.8.1 微服务引擎（CSE）

#### 3.8.1.1 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/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。支持用户自定义规格和自定义特性，您可以根据业务需要定制相应实例数的微服务引擎。
- 微服务引擎专业版不支持多AZ。
- 微服务引擎专享版在创建时可以设置多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。

#### 3.8.1.2 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。

#### 操作步骤

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击页面上方的“创建微服务引擎”。

**步骤3** 填写参数，参数说明如表3-28所示。

表 3-28 参数说明

参数	说明
微服务引擎名称	输入微服务引擎的名称，引擎创建后不能修改名称。
描述（可选）	输入引擎描述信息。
规格	选择引擎实例规格。 <ul style="list-style-type: none"><li>● 高可用 最多可选两个同CPU架构的可用区。 服务更可靠，可用于生产环境。</li><li>● 非高可用 只能选择一个可用区。 非高可用引擎在节点故障后引擎将不可用，影响部分业务功能；非高可用引擎创建完成后不可以再切换到高可用引擎。 建议仅在业务开发测试阶段使用。</li></ul>
可用区	选择可用区。 <ul style="list-style-type: none"><li>● 已创建的微服务引擎不支持更换可用区。</li><li>● 同一个区域内的可用区内网互通。</li></ul>
微服务实例数	选择微服务引擎支持的最大微服务实例数。
虚拟私有云	选择已创建的虚拟私有云。 虚拟私有云可以为您的引擎构建隔离的、用户自主配置和管理的虚拟网络环境。
子网	选择子网。

**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**

### 3.8.1.3 微服务引擎（CSE）管理

#### 3.8.1.3.1 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“基础设施 > 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 3-29 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。
开始时间	设置备份任务开启的时间，仅支持整点。

**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小时内触发自动备份任务。

----结束

#### 手动备份

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 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 3-30 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”，立即执行手动备份任务。

----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“基础设施 > 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

### 3.8.1.3.2 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“基础设施 > 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。

**步骤4** 在弹出的对话框勾选警告信息，单击“确定”。

**步骤5** 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“基础设施 > 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

### 3.8.1.3.3 查看微服务引擎接入地址

微服务引擎专享版创建完成之后才可查看，微服务引擎的接入地址查看方法如下。

#### 操作步骤

- 步骤1** 登录ServiceStage控制台，选择“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 选择待查看的微服务引擎，可以查看或单击复制服务注册发现地址。

----结束

### 3.8.1.3.4 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“基础设施 > 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

### 3.8.1.3.5 升级微服务引擎专享版

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，用户可以选择升级引擎版本。

---

#### 须知

仅微服务引擎专享版支持升级引擎版本。升级后，不支持版本回退。

---

## 背景介绍

升级过程中，高可用和非高可用对外表现不同：

- 高可用版本支持两个实例滚动升级，服务不中断。但是可能存在两个接入地址中一个不可用的情况，从而导致该实例不可用，需要支持快速切换到另外一个可用

实例。当前，ServiceComb SDK、Go Chassis和Mesher都支持快速切换实例。如果是调用服务注册中心、配置中心接口来进行注册发现的，需要实现实例切换。

- 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。ServiceComb SDK、Go Chassis和Mesher都做了可靠性增强，断服过程中，客户端会有缓存。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的，需要使用缓存降级。

## 操作步骤

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 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

### 3.8.1.3.6 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。

---

#### 须知

删除引擎后数据无法恢复，请谨慎操作。

---

## 背景介绍

支持删除处于如下状态的微服务引擎专享版：

- 可用
- 不可用
- 创建失败
- 变更失败
- 升级失败

## 操作步骤

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 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

### 3.8.1.4 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中心地址、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。

#### 操作步骤

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 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”。

**步骤3** 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4** 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

### 3.8.1.5 服务治理

#### 3.8.1.5.1 概述

如果使用微服务框架开发应用，在应用托管后启动应用，微服务会自动注册到对应的微服务引擎，您可以到微服务引擎控制台进行服务治理的相关操作。服务治理只适用于Java Chassis、Go Chassis开发框架。

目前ServiceStage提供了名称为“Cloud Service Engine”的微服务引擎专业版，您可以直接使用。也可以创建微服务引擎专享版，详情请参考[创建微服务引擎专享版](#)。

#### 3.8.1.5.2 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	<p>当出现访问量和流量较大，一台服务器无法负载的情况下，可以通过设置负载均衡的方式将流量分发到多个服务器均衡处理，从而优化响应时长，防止服务器过载。</p> <p>可以通过新增规则配置负载均衡策略，设置参数支持轮询、随机、响应时间权值、会话粘滞等多种负载均衡策略。</p>
限流	<p>限流主要解决微服务之间的流量分配问题，保证微服务在自己的资源池运行，互不影响。</p> <ul style="list-style-type: none"><li>● 当限流对象对当前服务实例的每秒请求数量超过设定的值，当前服务实例不再接受该对象的请求。</li><li>● 常用的检测方法是请求超时、流量过大等。</li><li>● 设置参数包括限流对象、QPS阈值等。</li></ul>
降级	<p>降级是容错的一种特殊形式，当出现服务吞吐量巨大，资源不够用等情况，可使用降级机制关掉部分不重要、性能较差的服务，避免占用资源，以保证主体业务功能可正常使用。</p>
容错	<p>容错是服务调用者访问服务实例，服务实例出现异常时的一种处理策略，出现异常后按照定义的策略进行重试或访问新的服务实例。</p>
熔断	<p>当发现由于某些原因导致服务出现了过载现象，为避免造成整个系统故障，可采用熔断来进行保护。</p> <p>熔断在服务请求处理出现异常时产生作用。进入熔断状态后，hystrix会认为被请求的服务已经无法处理请求，在第一时间截断请求直接返回错误给调用者。</p> <p>hystrix每隔一段时间会尝试访问后端服务，如果服务恢复正常，会退出熔断状态，恢复正常的请求访问。</p>
错误注入	<p>错误注入策略用于测试微服务的容错能力，可以让用户知道，当出现延迟或错误时，系统是否能够正常运行。</p> <p>错误注入通过延迟、错误等方式，供用户测试微服务的容错能力。</p>
黑白名单	<p>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。</p> <p>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>。</p>

## 设置负载均衡

- 步骤1** 登录ServiceStage控制台，选择“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务治理”。

**步骤4** 单击需要治理的微服务。

**步骤5** 单击“负载均衡”。

**步骤6** 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	<p>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。</p> <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7** 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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务治理”。

**步骤4** 单击需要治理的微服务。

**步骤5** 单击“限流”。

**步骤6** 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的值，当前服务实例不再接受该对象的请求。	0-99999的整数。

### 📖 说明

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

**步骤7** 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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务治理”。

**步骤4** 单击需要治理的微服务。

**步骤5** 单击“降级”。

**步骤6** 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7** 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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务治理”。

**步骤4** 单击需要治理的微服务。

**步骤5** 单击“容错”。

**步骤6** 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。

配置项	配置项说明
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7** 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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务治理”。

**步骤4** 单击需要治理的微服务。

**步骤5** 单击“熔断”。

**步骤6** 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7** 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“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。
- 步骤3** 选择“服务治理”。
- 步骤4** 单击需要治理的微服务。
- 步骤5** 单击“错误注入”。
- 步骤6** 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7** 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“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。
- 步骤3** 选择“服务治理”。
- 步骤4** 单击需要治理的微服务。
- 步骤5** 单击“黑白名单”。
- 步骤6** 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data*，指在黑名单下匹配到的名称以data开头的服务不允许访问当前服务，或者在白名单下匹配到的名称以data开头的服务允许访问当前服务。

**步骤7** 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>
```

### 3.8.1.6 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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 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。
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

### 3.8.1.7 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：显示当前用户的所有应用，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：显示当前用户的所有微服务，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：显示当前用户的所有实例，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“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。
- 步骤3** 选择“服务目录”。
- 步骤4** 单击微服务名称。
- 步骤5** 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	<ol style="list-style-type: none"><li>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></li><li>2. 单击“...”，选择目标文件。</li><li>3. 单击“上传文件”，批量导入配置项。</li><li>4. 单击“关闭”。</li></ol>
创建配置	<ol style="list-style-type: none"><li>1. 单击“创建配置”，选择配置“作用域”。</li><li>2. 输入“配置项”。</li><li>3. 输入“值”。</li><li>4. 单击“确定”，保存配置。</li></ol>
修改配置	<ol style="list-style-type: none"><li>1. 单击目标配置项对应“操作”列的“编辑”。</li><li>2. 在“值”编辑框输入新值。</li><li>3. 单击“确定”，保存配置修改。</li></ol>
删除配置	<ol style="list-style-type: none"><li>1. 单击目标配置项对应“操作”列的“删除”。</li><li>2. 单击“确定”，删除配置。</li></ol>

----结束

## 灰度发布

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务目录”。

**步骤4** 单击微服务名称，进入详情页面，单击“灰度发布”。

**步骤5** 单击“添加发布规则”。

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

配置项	说明
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。 <b>说明</b><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。</li><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></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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务目录”。

**步骤4** 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“基础设施 > 微服务引擎（CSE）”。

**步骤2** 单击微服务引擎的“查看控制台”。

**步骤3** 选择“服务目录”。

**步骤4** 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“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。
- 步骤3** 选择“服务目录”。
- 步骤4** 单击微服务名称，进入微服务详情页。
- 步骤5** 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“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。
- 步骤3** 选择“服务目录”。
- 步骤4** 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5** 单击“添加标签”，输入“键”和“值”。
- 步骤6** 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“基础设施 > 微服务引擎（CSE）”。
- 步骤2** 单击微服务引擎的“查看控制台”。
- 步骤3** 进入微服务控制台。
- 步骤4** 选择“服务目录 > 实例列表”。
- 步骤5** 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

## 3.8.2 单台虚拟机安装虚拟机 Agent

将组件部署到虚拟机上，需要安装Agent，将主机节点纳管后，后台才能与主机节点通信。

虚拟机Agent的状态及说明，请参见表3-31。

表 3-31 虚拟机 Agent 状态说明

Agent状态	说明
缺少Agent，请先安装	ECS节点上没有安装虚拟机Agent，需安装。
a.b.c	虚拟机Agent已经安装，当前处于正常状态。 a.b.c表示虚拟机Agent的版本号。例如：1.3.15。
Agent离线	Agent已经安装，但处于离线状态，不能正常工作。
CCE集群中的ECS，跟随集群纳管到环境中，无需单独纳管	ECS节点属于CCE集群的节点，不能再安装虚拟机Agent。

本章节指导您在单台虚拟机上安装虚拟机Agent。

### 操作步骤

**步骤1** 登录ServiceStage控制台，选择“基础设施 > 虚拟机Agent管理”。

**步骤2** 找到需要安装Agent的虚拟机，在右侧单击“安装”，弹出Agent安装窗口。

**步骤3** 选择“授权模式”。

授权Agent可以使用您的身份认证信息，获取应用的部署、升级、启动、停止等任务，并执行任务。

您可以选择“AK/SK”模式进行授权。

“授权模式”选择“AK/SK”：输入“AK”、“SK”。AK、SK获取请参考[如何获取AK/SK?](#)

**步骤4** 根据实际业务需要，确认是否勾选“自动添加应用访问端口”。

**步骤5** 单击“复制命令”，复制窗口下方自动生成的命令，即为Agent安装命令。

“AK/SK”模式，命令示例如下：

```
export AGENT_INSTALL_URL=https://${Region_Name}-servicestage-vmapp.obs.${Region_Name}.${Domain_Name}/vmapp/agent/agent-install.sh;if [ -f `which curl` ];then curl -# -O -k ${AGENT_INSTALL_URL};else wget --no-check-certificate ${AGENT_INSTALL_URL};fi;bash agent-install.sh ${AK}${SK} ${Project_ID} ${Version} ${Region_Name} ${Flag}
```

- 其中AGENT\_INSTALL\_URL为Agent安装地址。
- \${AK}/\${SK}为访问密钥值。
- \${Region\_Name}为区域名称。
- \${Domain\_Name}为全局域名。
- \${Project\_ID}为项目ID，如何获取项目ID请参考 [如何获取项目名称?](#)。
- \${Version}为版本号，使用latest，自动去下载最新版本。
- \${Flag}为布尔值，表示是否自动添加应用访问端口。true表示是；false表示否。

**步骤6** 参照界面提示登录虚拟机，执行安装命令。

----结束

# 4 常见问题

## 4.1 如何获取 AK/SK?

### 说明

请以应用所需权限用户登录ServiceStage控制台。

- 步骤1** 登录ServiceStage控制台。
- 步骤2** 鼠标移动到用户名，在下拉菜单选择“我的凭证”。
- 步骤3** 在导航栏，单击“访问密钥”。
- 步骤4** 单击“新增访问密钥”，通过身份认证后成功创建AK/SK。
- 步骤5** 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

## 4.2 如何处理修改项目名称导致的错误?

如果使用了微服务引擎专业版，由于是通过项目来隔离的，所以需要感知项目变化。

当修改了项目名称后，本项目下使用专业版的服务中心进行注册发现的微服务都需要升级，否则可能出现“Project id or name is not existed”的错误。

解决方案如下：

- 如果微服务通过ServiceStage部署，请参考[升级应用组件](#)升级该微服务组件（无需修改任何配置参数）。
- 如果微服务不通过ServiceStage部署，请将微服务配置中的项目修改为实际值，并升级微服务。
  - 项目名称对应的配置项，请参见[表4-1](#)。
  - 项目名称获取方法，请参考[如何获取项目名称？](#)。

表 4-1 项目名称对应的配置项

微服务框架	配置项	配置所在文件
Java Chassis	servicecomb.credentials.project	microservice.yml
Spring Cloud Huawei	spring.cloud.servicecomb.credentials.project	application.yml
Dubbo ServiceComb	dubbo.servicecomb.credentials.project	dubbo.properties
Go Chassis	servicecomb.credentials.project	chassis.yaml或auth.yaml

## 4.3 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与周边微服务进行配合，推荐“基于契约”的开发方式。	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。
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。	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。	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。

## 4.4 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”页面。

**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** 单击未就绪的组件名称，进入组件实例“概览”页。

**步骤5** 选择“实例列表”，单击实例名称前的。

**步骤6** 单击“事件”页签，查看类型非正常的事件。

例如事件类型为“告警”，查看该告警的“描述”，判断应用失败原因。

----结束

## 4.5 如何处理部署或升级虚拟机组件失败？

将组件部署在虚拟机上的时候，部署或者升级失败，可以通过如下步骤确认失败的原因并解决问题。

**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”页面。

**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** 单击未就绪的组件名称，进入组件实例“概览”页，获取并记录组件实例的名称。

**步骤5** 选择“更新升级”，在更新升级页面，获取并记录组件实例的版本号。

**步骤6** 选择“实例列表”，单击启动失败或者安装失败的实例名称前的。

**步骤7** 单击“事件”页签，查看报错信息。如提示端口已经被占用，则可以重新部署修改端口。

**步骤8** 如果**步骤6**获取的日志信息不够明确，则需要到ECS主机上获取该组件实例的日志。

1. 单击实例“所在节点”，跳转到主机详情页。如果有绑定弹性公网IP的话，可以单击“弹性公网IP”的页签，获取节点的弹性公网IP后通过SSH客户端工具连接到主机，或者直接单击左上角的“远程登录”，登录主机执行如下命令：

```
cd /var/log/application/{component_instance_name}/{version}
```

其中，{component\_instance\_name}用**步骤4**获取的组件实例名代替，{version}用**步骤5**获取的版本号代替。

然后执行ls命令，再执行cd命令进入ls后显示的子目录内。

2. 在上一步进入的目录内获取详细的错误日志。  
如果是Tomcat8运行时的组件，则查看catalina.out的日志文件，查看tomcat启动过程中的报错。  
其他类型的应用可以看start\_app.log，查看启动过程中的报错。
- 结束

## 4.6 Node.js 8 软件包打包有哪些约束？

ServiceStage支持将Node.js8的zip包直接安装在虚拟机上。用户只需要将业务的zip包上传后部署，ServiceStage就会在虚拟机上自动安装Node.js8运行时的基础软件，并且自动执行npm run命令启动用户的Node.js8组件。

用户打包的zip包中不能包含有start.sh、stop.sh、install.sh、pre-install.sh、uninstall.sh、appspec.node.yml以及agent\_template.yml这些名称的文件，以免跟基础软件包中包含的文件冲突。

## 4.7 如何处理安装 Agent 失败？

### 问题描述

安装Agent时，显示安装失败。提示：

```
auth token valid fail: User does not have valid roles to write!
```

### 处理方法

- 步骤1 获取AK/SK，请参考[如何获取AK/SK?](#)。
- 步骤2 检查安装命令中的AK/SK与步骤1获取的AK/SK是否一致。

----结束

## 4.8 如何处理 Agent 离线？

获取新的AK/SK并更新虚拟机Agent配置的AK/SK，然后重启虚拟机Agent。

具体操作步骤如下：

- 步骤1 在ECS节点界面通过VNC登录节点，或者给节点绑定弹性IP后通过SSH客户端登录节点。
- 步骤2 执行cd /opt/servicestage-agent/命令，编辑servicestage-agent.conf，修改该配置文件中的AK和SK，修改完后保存退出。
- 步骤3 执行以下命令完成agent的重启，其中x.x.x要替换成实际环境中servicestage-agent的最高版本：

```
cd /opt/servicestage-agent/servicestage-agent-x.x.x
su agent
./servicestage-agent.sh restart
```

----结束

## 4.9 虚拟机部署的应用组件支持在什么目录写文件？

虚拟机部署的应用组件只支持在应用组件的运行目录下写文件，比如写日志文件或者解压包等。

应用组件的运行目录为安装虚拟机应用组件的ECS上的/opt/application/\${appName}/\${appVersion}/\${instanceId}目录路径下。其中：

- \${appName}为组件实例名。
- \${appVersion}为组件实例的版本号。
- \${instanceId}为实例id。

同时，虚拟机部署的应用组件只支持在应用组件的运行目录下写文件只针对新安装的组件实例或者升级后的组件实例生效，对于原先已经部署的组件实例的目录还是维持原来的权限。

## 4.10 如何处理虚拟机部署的组件删除失败报 host status is not active 错误的问题？

### 问题描述

虚拟机部署组件删除失败，在任务详情界面单击“查看详情”，看到的错误信息示例如下：

```
{
  "statusCode": 400,
  "jsonBody": {
    "error_code": "SVCSTG.VMAPP.4001020",
    "error_msg": "4001020",
    "error_detail": "host status is not active: abb3d0a4-f715-4932-b7ec-6dd917f65778,4f68e35b-6e08-48d0-bd3a-1151be19efa5"
  }
}
```

其中：

- 错误码为：SVCSTG.VMAPP.4001020。
- 详细错误信息为：host status is not active: abb3d0a4-f715-4932-b7ec-6dd917f65778 ， “abb3d0a4-f715-4932-b7ec-6dd917f65778,4f68e35b-6e08-48d0-bd3a-1151be19efc6” 是部署组件的两个弹性云服务器的ID。

### 解决方法

- 步骤1** 登录云服务器控制台，单击“弹性云服务器”。
- 步骤2** 在弹性云服务器列表，使用报错信息中的弹性云服务器ID搜索部署组件的弹性云服务器。
- 步骤3** 查看弹性云服务的状态是否为“运行中”。
  - 是，执行**步骤2**，搜索报错信息中的下一个弹性云服务器。
  - 否，执行**步骤4**。

**步骤4** 根据弹性云服务器的状态：

- 在“操作”列选择“更多 > 开机”或者“更多 > 重启”，将弹性云服务器的状态恢复为“运行中”。
- 如果不再使用该弹性云服务器，在“操作”列选择“更多 > 删除”，删除该弹性云服务器。

**步骤5** 待报错信息中所有弹性云服务器已按照**步骤2~步骤4**执行完操作后，重新在ServiceStage界面执行删除组件操作。

----结束

## 4.11 Docker 应用程序依赖的问题

### 问题描述

在微服务docker里面运行node程序，这个程序依赖node-gyp，怎么在程序编译之前安装好这些依赖？

#### 说明

由于环境没有外网权限，无法拉取到外网的依赖，需要事先编译好。

### 解决方法

**步骤1** 将您的工程编译好。

**步骤2** 将编译好的工程整个打包成zip包。

**步骤3** 将打包好的软件包上传到obs，再使用ServiceStage的应用管理功能即可。

#### 说明

在创建组件的时候，运行时选择Nodejs8。

----结束

## 4.12 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 xxx/test/busybox:latest
```

#### 说明

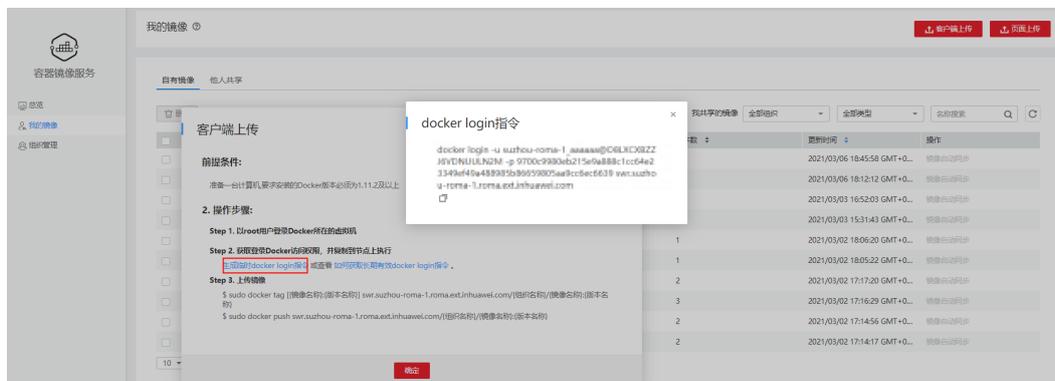
- xxx为租户或用户准备上传仓库的地址。
- test为组织。
- busybox为镜像名称。
- latest表示镜像最新版本。

上传失败，Docker客户端出现如下提示：

unauthorized: authentication required

## 解决方法

**步骤1** 进入“容器镜像服务”页面，选择“我的镜像 > 客户端上传 > 生成临时docker login指令”，拷贝该命令。



**步骤2** 单击“组织管理”，创建组织，例如组织名为“test”。

**步骤3** 登录数据面节点，执行**步骤1**拷贝的命令。

**步骤4** 执行如下命令，向本租户或用户下有操作权限的组织下上传镜像，例如**步骤2**创建的组织“test”，上传一个名为busybox的镜像。

```
docker push xxx/test/busybox:latest
```

### 说明

- xxx为租户或用户准备上传仓库的地址，为**步骤1**中拷贝命令的最后面，例如：swr.{regionID}.{external\_global\_domain\_name}。
- test为该租户或用户下有操作权限的组织。
- latest表示镜像最新版本。

**步骤5** 上传成功后，显示如下：

```
The push refers to a repository [xxx/test/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

## 4.13 如何获取项目名称？

**步骤1** 登录ServiceStage控制台。

**步骤2** 单击用户名，在下拉菜单选择“我的凭证”。

**步骤3** 单击“API凭证”，在项目列表查看项目名称和项目ID。

----结束

## 4.14 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，微服务的注册中心地址使用微服务引擎注册中心的IPv4地址，可以注册成功并正常启动。

如果修改微服务的注册中心地址为微服务引擎注册中心的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[修改环境](#)。

**步骤2** 重新部署应用，请参考[部署组件](#)。

----结束

## 4.15 如何处理操作微服务引擎专享版时遇到非微服务引擎本身错误？

### 问题描述

在对微服务引擎专享版执行创建、删除、变更规格、升级等操作时，可能会遇到非微服务引擎本身的错误。

例如，在创建微服务引擎专享版作时，集群部署失败，报错如下：

```
{"error_code":"SVCSTG.00500400","error_message":{"kind":"Status","apiVersion":"v1","metadata":{},"status":{"Failure"},"code":400,"errorCode":"CCE.01400013","errorMessage":"Insufficient volume quota.\","error_code":"CCE_CM.0307","error_msg":"Volume quota is not enough"},"message":"volume quota checking failed as [60/240] insufficient volume size quota"},"reason":"QuotaInsufficient\n"}}
```

### 解决方法

页面上展示的错误信息中已经包含相应服务的错误码，根据错误码和错误信息联系相应服务的技术支持工程师提供支持。

## 4.16 如何处理部署组件时失败，报错日志详情显示 connection was closed?

### 问题描述

在ServiceStage中部署应用组件时，在部署任务详情界面报错，单击查看日志详情显示报错如下：

```
{
  "Message": "ECF0006",
  "Detail": "io.vertx.core.VertxException:Connection was closed"
}
```

### 解决方法

可能是网络波动导致请求的时候连接被关闭，可以等一段时间后再重新部署。

如果重新部署失败，请联系技术支持工程师协助解决。

## 4.17 如何处理部署虚拟机组件时遇到 ECS 错误?

### 问题描述

在ServiceStage服务应用管理部署组件的基本配置阶段，当组件“部署系统”选择“虚拟机部署”，“资源类型”选择“弹性云服务器”，部署时可能会遇到ECS服务不可用问题。

例如，在组件部署时调用ECS接口超时，查看日志详情报错如下：

```
{
  "statusCode": 500,
  "jsonBody": {
    "error_code": "SVCSTG.VMAPP.5001002",
    "error_msg": "read ECS host 471ff77a-c827-41d5-941d-4fea8aaa56ef fail TIMEOUT."
  }
}
```

### 解决方法

**步骤1** 重新部署组件，查看部署是否成功。

- 是，处理结束。
- 否，执行**步骤2**。

**步骤2** 联系技术支持工程师协助解决。

----结束

## 4.18 如何处理虚拟机部署端口访问不通的错误?

### 问题描述

在ServiceStage服务应用管理部署组件的基本配置阶段，当组件“部署系统”选择“虚拟机部署”，“资源类型”选择“弹性云服务器”，部署时可能会遇到容器端口无法访

问的问题。使用curl -kv http://\${部署应用组件的弹性云服务器节点IP}:\${容器端口}命令访问容器端口时会提示访问超时。

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

## 解决方法

- 步骤1 登录云服务器控制台，单击“弹性云服务器”。
- 步骤2 在弹性云服务器列表选择部署组件的弹性云服务器，单击名称进入“基本信息”。
- 步骤3 在“安全组”页签，单击“更改安全组”。
  - 如果该端口规则已经在已有安全组中存在，可直接勾选该安全组。
  - 如果该端口规则在已有安全组中不存在，请单击“新建安全组”，创建安全组，自行配置相关规则后再勾选该新建安全组。
- 步骤4 再次使用curl -kv http://\${部署应用组件的弹性云服务器节点IP}:\${端口}命令访问容器端口，确认问题是否已解决。

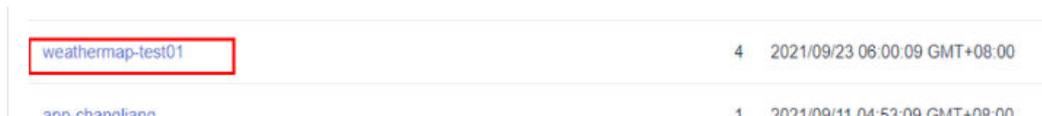
----结束

## 4.19 如何处理微服务应用名称与组件所属应用名称不同的问题？

### 问题描述

在ServiceStage控制台创建并部署微服务类型组件后，在部署该微服务组件的微服务引擎控制台的“服务目录 > 微服务列表”下查看该微服务的应用名称，如图4-2。可能会遇到与该微服务组件所在的应用名称不一致的问题，如图4-1所示。

图 4-1 应用管理中的应用名称



weathermap-test01	4	2021/09/23 06:00:09 GMT+08:00
ann.chanliann	1	2021/09/11 04:53:09 GMT+08:00

图 4-2 微服务引擎的服务目录中的应用名称



## 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 单击遇到问题的微服务组件所在的应用名称，进入应用“概览”。
- 步骤3 单击“环境变量”，在“环境”下拉列表选择组件部署环境。
- 步骤4 选择“环境变量 > 添加环境变量”，配置全局环境变量。
  - Spring Cloud框架：参考表4-2输入“Key”、“Value”的值。

表 4-2 Spring Cloud 框架全局环境变量

Key	Value
spring_cloud_serviccomb_discovery_appName	输入微服务组件所在的应用名称

- Java Chassis框架：参考表4-3输入“Key”、“Value”的值。

表 4-3 Java Chassis 框架全局环境变量

Key	Value
serviccomb_service_application	输入微服务组件所在的应用名称

- 步骤5 参考[应用组件部署](#)，重新部署该应用下已创建的组件。  
等待组件部署完成，状态更新为“运行中”。
- 步骤6 选择“基础设施 > 微服务引擎（CSE）”。
- 步骤7 选择部署了微服务组件的微服务引擎，单击“查看控制台”。

**步骤8** 选择“服务目录 > 微服务列表”，可以看到微服务应用名称跟应用管理中的应用名称一致。

----结束

## 4.20 如何解释微服务名称与组件名称不同的原因？

### 现象描述

在ServiceStage控制台创建并部署微服务类型组件后，在部署该微服务组件的微服务引擎控制台的“服务目录 > 微服务列表”下查看微服务名称，可能会遇到与该微服务组件的名称不一致的问题。

### 原因分析

微服务引擎中显示的微服务名称是微服务的配置文件中设置的，和业务调用相关，是由微服务开发者开发时设定的。在ServiceStage控制台创建并部署微服务类型组件时，组件名称是由用户自定义的。

该现象对于用户业务无影响，无需处理。

## 4.21 微服务引擎专享版数据恢复失败

### 问题描述

微服务引擎专享版数据恢复失败。

### 解决方案

数据恢复失败可能导致引擎不可用，引起用户业务故障，请联系技术支持工程师，按故障处理步骤处理，优先恢复引擎功能。