

应用管理与运维平台

产品介绍

文档版本 01
发布日期 2025-01-20



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 图解应用管理与运维平台	1
1.1 ServiceStage 简介.....	2
1.2 ServiceStage 应用调度与资源管理框架.....	3
1.3 ServiceStage 微服务运行与治理框架.....	4
1.4 ServiceStage 应用智能化运维.....	5
2 什么是应用管理与运维平台	6
3 产品优势	8
4 应用场景	10
4.1 构建微服务应用.....	10
4.2 Web 应用生命周期管理.....	12
5 安全	14
5.1 责任共担.....	14
5.2 身份认证与访问控制.....	15
5.3 数据保护技术.....	16
5.4 审计与日志.....	16
5.5 服务韧性.....	17
5.6 监控安全风险.....	17
6 微服务引擎版本支持机制	18
7 约束与限制	20
8 产品规格差异	23
9 权限管理	26
10 与其他服务的关系	37
11 基本概念	39

1 图解应用管理与运维平台

1.1 ServiceStage 简介

The infographic is titled "图解ServiceStage 应用管理与运维平台" (ServiceStage Application Management and Operation Platform). It features the Huawei and CloudEye logos at the top. The main theme is "企业云化已成为不可逆转之趋势" (Enterprise cloudification has become an irreversible trend). It includes two introductory questions: "你有多久没去过银行?" (How long has it been since you went to the bank?) and "你参加过多少次剁手?" (How many times have you shopped online?). The core message is "ServiceStage 让企业应用上云更简单" (ServiceStage makes it easier for enterprise applications to go to the cloud). It is divided into three main sections: 1. "企业应用逐步迁移上云" (Enterprise applications gradually migrate to the cloud), 2. "企业应用微服务化" (Enterprise applications microservices), and 3. "企业应用统一运维监控管理" (Unified operation and maintenance and monitoring management of enterprise applications). Each section contains a diagram and descriptive text. The infographic concludes with a "ServiceStage 优势亮点" (ServiceStage Advantages) section listing: 自助按需 (Self-service on-demand), 开源开放 (Open source and open), 简单易用 (Simple and easy to use), and 敏捷高效 (Agile and efficient).

1.2 ServiceStage 应用调度与资源管理框架

3分钟了解 应用调度与资源管理框架

ServiceStage 系列海报

01/ 企业云化转型面临三大挑战

- 1 业务上线时间长**
部署新的业务需要数天
启动虚拟机 → 安装配置 → 业务上线
虚拟化场景下上线流程
- 2 分布式应用管理复杂**
应用类型多, 并部署在不同IaaS和不同数据中心。
FusionSphere, vmware
(网络、物理、虚拟化)资源, 异构IaaS, 多个数据中心/区域
- 3 企业IT资源利用率低**
由于资源管理非自动化, 导致虚拟机资源利用率低, 大多在10%~20%。
30%, 10%, 15%, 18%, 20%
1, 2, 3, 4, 12

02/ 如何解决这些挑战

- 挑战一: 如何快速把应用部署到不同的“数据中心/区域/虚拟机/容器/物理机”上去?
- 挑战二: 如何保证这些应用可以高效稳定的运行?
- 挑战三: 如何保障企业IT资源的最大化利用?

03/ ServiceStage解决方案

应用调度与资源管理技术, 通过将底层资源抽象化(包括网络、存储、计算等), 自动将应用部署到指定的机器、数据中心、打通从应用建模、编排部署到资源调度、弹性伸缩、监控自愈的生命周期管理自动化。

- 1 自动化部署, 加速业务上线效率**
应用建模 → 资源编排 → 物理部署
应用编排, 资源编排, 应用部署, 应用编排, 资源编排, 应用部署
规划应用部署位置、资源需求等
- 2 业务高可靠可用**
应用冗余和弹性部署, 避免干扰, 保障高可靠。
应用亲和性部署, 保障性能。
- 3 提升资源利用率**
资源有余, 售卖给其他应用, 提升资源利用率。
资源需求大时, 释放资源, 保障性能。

1.3 ServiceStage 微服务运行与治理框架

01/ 什么是微服务

微服务架构，将一类或单一业务单元拆分为多个小型服务构建的分布式架构，每个微服务负责提供特定的业务功能并相互协作完成业务。这种微服务架构可以独立构建完成，且支持自动化部署和持续集成部署。

第一代：单体架构

第二代：SOA架构

第三代：微服务架构

高耦合：
• 高耦合度，牵一发而动全身。

低耦合：
• 松耦合，解耦。

特点：
• 无状态运行并部署。
• 自动伸缩、升级、扩缩容能力强。

02/ 企业IT微服务架构转型，面临诸多挑战

如何基于微服务快速实现开发上线？

在不可控的部署环境下，如何保证业务可靠运行？

传统部署下的部署流程，如何快速实现部署成本降低？

在复杂的部署环境中，如何实现快速定位与恢复？

03/ 微服务治理框架应运而生

提供开放的微服务框架，配套开发工具和丰富的微服务治理能力，帮助企业快速构建微服务化。

企业DevOps人员

ServiceStage 微服务治理框架

微服务注册中心
微服务配置中心
微服务网关

微服务治理框架

华为云

微服务注册中心：微服务注册管理，冷热线一键构建部署上线。

微服务配置中心：微服务配置管理方案，应用零改造轻松上线。

微服务网关：兼容SpringCloud主流开源社区，与业界生态能力互通。

04/ 微服务典型场景

若您的业务现状满足以下任一场景描述，则可考虑微服务转型。

新应用开发

一键式快速上线

微服务治理能力持续增强

丰富的本地开发调试工具

遗留应用接入

只需完成应用ServiceMesh的部署即可自由接入Kubernetes进行微服务治理

支持多语言接入，让遗留应用也能快速上线

支持接入NodeJS、.NET等主流开发语言

持续集成

高服务可用性，支持分钟级快速上线

支持多种语言一键式Source2Deploy

支持企业级自定义开发流程

微服务治理

自动动态扩缩容、服务熔断、服务降级

自动部署、服务发现

支持微服务链路SLA实时监控、报警、熔断

实时秒级业务治理、保障业务运行不中断。

1.4 ServiceStage 应用智能化运维

3分钟了解 ServiceStage 应用智能化运维

主输入代码，寻找应用的公共代码，在微服务、容器与云原生环境下，关键业务代码行。

01/ 云时代，体验不佳是用户流失的根本原因

云时代下，用户对应用体验要求更高，体验不佳导致用户流失的根本原因。

- 应用故障? 宕机?
- 应用性能? 延迟?
- 系统资源不足?
- 网络延迟? 丢包?
- SQL执行慢?
- 服务调用失败?

做好，服务!

02/ 云化场景下的运维面临挑战

- 云原生应用架构复杂，运维难度大。
- 多云、混合云环境，运维复杂度高。
- 容器化部署，运维难度大。
- 微服务架构，运维难度大。

03/ ServiceStage 解决方案 —— 提供统一的自动化运维监控管理

ServiceStage提供统一的自动化运维监控管理，实现运维效率提升，降低运维成本，提升用户体验。

- 统一运维平台
- 统一监控管理
- 统一告警管理
- 统一日志管理
- 统一配置管理
- 统一部署管理

04/ ServiceStage 自动化运维核心优势

- 应用拓扑分析：应用关系一目了然，故障下钻**
- 应用性能监控：支持海量性能指标，与现有系统无缝对接**
- 业务合规监控：监控合规交易的KPI数据，提升用户体验**
- 非侵入式部署：一键式安装部署，用户无感知**

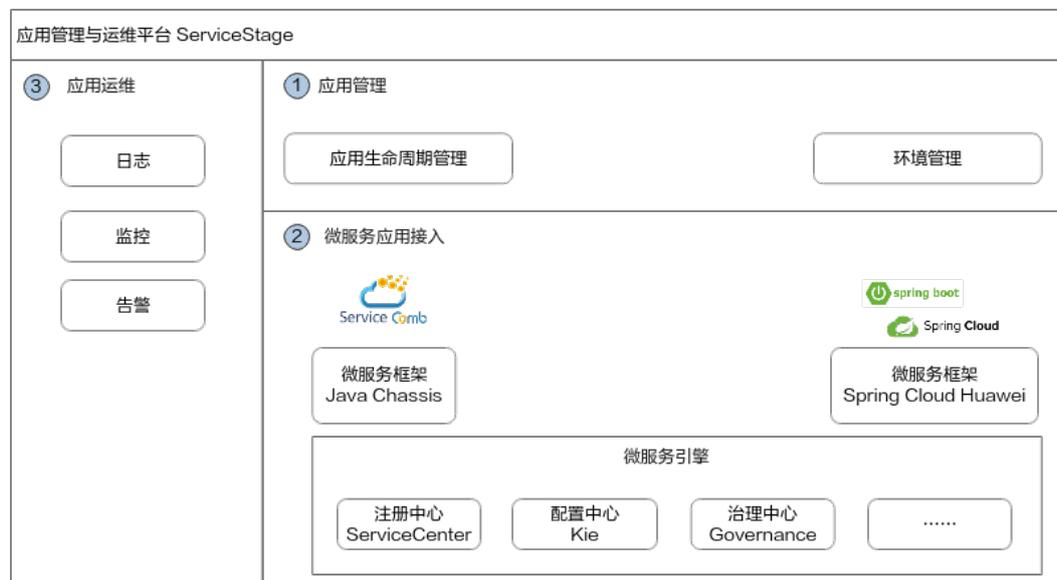
2 什么是应用管理与运维平台

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、Php、Python、Node.js、Docker、Tomcat技术栈。支持Apache ServiceComb Java Chassis（Java Chassis）、Spring Cloud等微服务应用，让企业应用上云更简单。

ServiceStage主要包含如下能力：

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

图 2-1 ServiceStage 产品功能



应用管理

- 应用生命周期管理
应用完成开发后，可以托管在ServiceStage上，为您提供完整的应用生命周期管理：

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

微服务应用接入

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

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

应用运维

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

3 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3-1所示的优势。

表 3-1 产品优势

应用生命周期	传统平台	使用ServiceStage平台
环境准备阶段	<ul style="list-style-type: none">资源获取效率低 (>1天)资源利用率低 (<30%)	<ul style="list-style-type: none">自助高效获取资源 (分钟级)按需付费 (弹性伸缩)
业务开发阶段	<ul style="list-style-type: none">架构耦合，牵一发而动全身技术单一，需要想办法用一种技术解决所有问题只能按大颗粒系统发布版本，响应周期长	<ul style="list-style-type: none">架构解耦 基于契约(Open API)的开发模式，让微服务的开发、测试、文档、协作和管控活动标准化、自动化。各种技术灵活接入 支持Java、PHP、Python、Node.js开发语言。 高性能REST/RPC微服务开发框架，提供开箱即用的工具，降低开发门槛。 提供ServiceComb、Spring Cloud、Service Mesh商业版。敏捷高效 一站式微服务治理控制台，提供微服务负载均衡、限流、降级、熔断、容错、错误注入等治理能力。 支持微服务级升级、灰度发布。
安装部署阶段	<ul style="list-style-type: none">烟囱式系统全手工部署	开发者只需使用ServiceStage+源码软件仓库，实现一键自动部署和更新。

应用生命周期	传统平台	使用ServiceStage平台
应用升级	<ul style="list-style-type: none"> 打补丁方式 手动升级 中断业务 	<p>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。</p> <p>支持灰度发布：为保障新特性平稳上线，可以通过灰度发布功能选择少部分用户试用，降低发布风险。</p>
应用运维	<ul style="list-style-type: none"> 应用崩溃，闪退 服务端响应慢 系统资源不足 故障难定位 	<ul style="list-style-type: none"> 实时图形化展示应用监控指标 CPU占用、告警、节点异常、运行日志、关键事件实时掌握。 微服务治理 支持微服务接口级SLA指标（吞吐量、时延、成功率）实时（秒级）监控和治理，保障应用运行不断服。

4 应用场景

4.1 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。

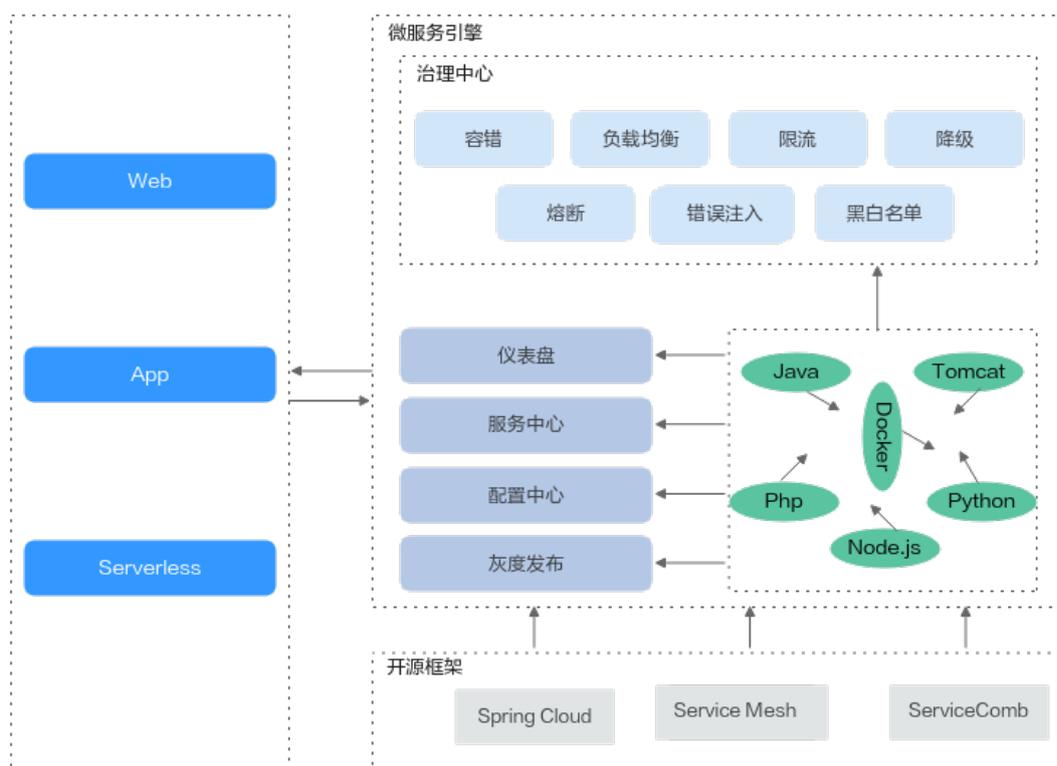
ServiceStage可承载微服务应用的全生命周期管理。支持Java、Php、Python、Node.js、Docker、Tomcat等技术栈，可无侵入托管Apache ServiceComb Java Chassis、Spring Cloud等微服务应用，另外还提供配置管理、监控运维和服务治理等更多功能，让企业微服务应用上云更简单。

优势

ServiceStage提供的微服务应用解决方案具有以下优势：

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

图 4-1 微服务应用解决方案



- 实现Spring Cloud、Java Chassis之间的互相发现。

持续集成和持续交付

应用场景

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

价值

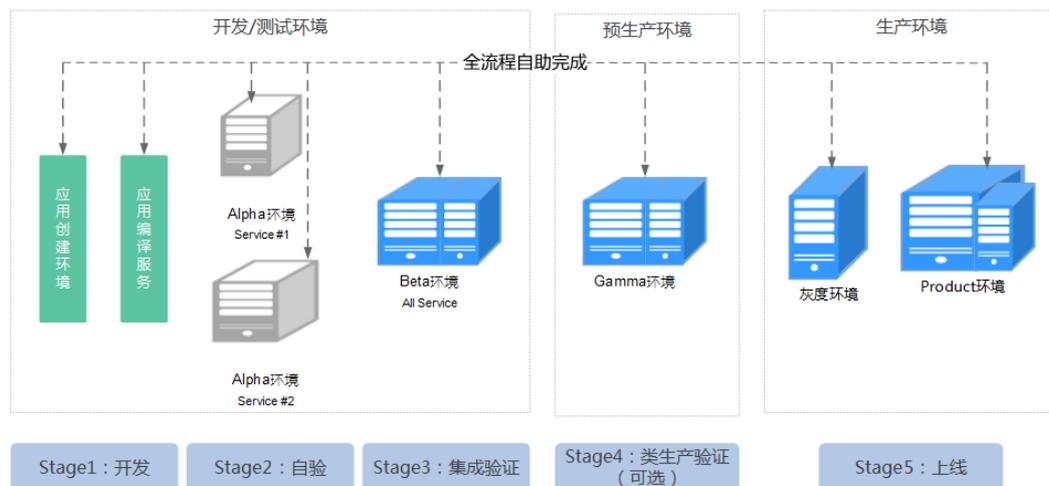
由人工执行变成了自动化执行，减少错误出现，提高工作效率。

环境及流程标准统一，利于业务扩展，降低升级改造成本。

优势

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

图 4-2 持续集成和持续交付



灰度发布

应用场景

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

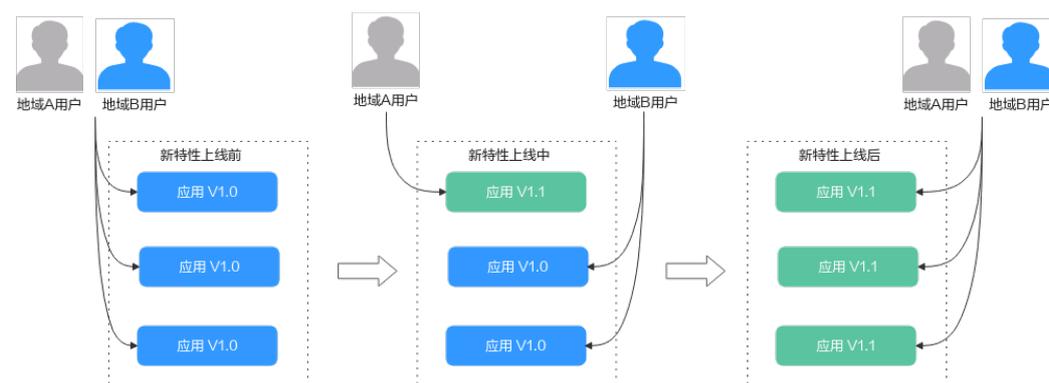
价值

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

优势

ServiceStage提供了灰度发布的能力。

图 4-3 灰度发布



4.2 Web 应用生命周期管理

典型业务应用场景

应用场景

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

价值

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

优势

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

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

5 安全

5.1 责任共担

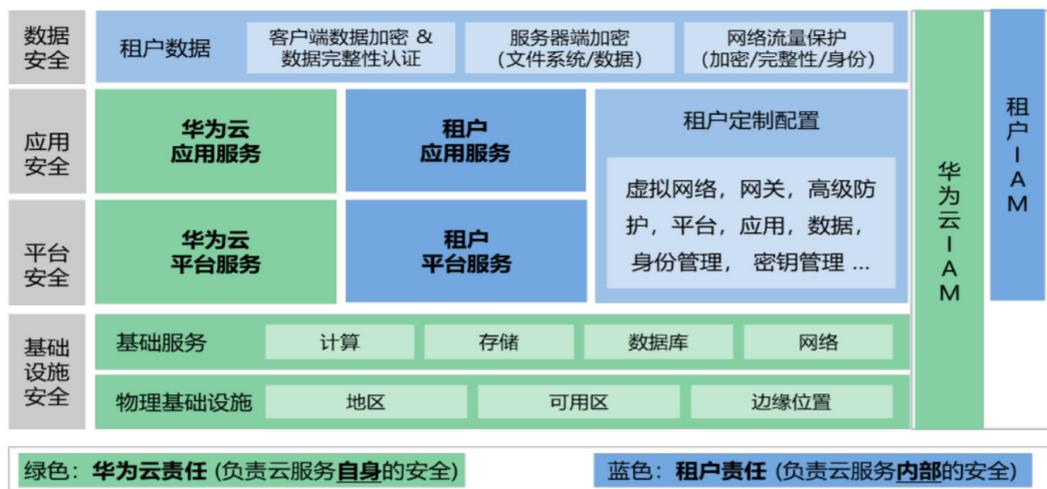
华为云秉承“将对网络和业务安全性保障的责任置于公司的商业利益之上”。针对层出不穷的云安全挑战和无孔不入的云安全威胁与攻击，华为云在遵从法律法规业界标准的基础上，以安全生态圈为护城河，依托华为独有的软硬件优势，构建面向不同区域和行业的完善云服务安全保障体系。

安全性是华为云与您的共同责任，如[图5-1](#)所示。

- 华为云：负责云服务自身的安全，提供安全的云。华为云的安全责任在于保障其所提供的IaaS、PaaS和SaaS各类各项云服务自身的安全，涵盖华为云数据中心的物理环境设施和运行其上的基础服务、平台服务、应用服务等。这不仅包括华为云基础设施和各项云服务技术的安全功能和性能本身，也包括运维运营安全，以及更广义的安全合规遵从。
- 租户：负责云服务内部的安全，安全地使用云。华为云租户的安全责任在于对使用的IaaS、PaaS和SaaS类各项云服务内部的安全以及对租户定制配置进行安全有效的管理，包括但不限于虚拟网络、虚拟主机和访客虚拟机的操作系统，虚拟防火墙、API网关和高级安全服务，各项云服务，租户数据，以及身份账号和密钥管理等方面的安全配置。

[《华为云安全白皮书》](#)详细介绍华为云安全性的构建思路与措施，包括云安全战略、责任共担模型、合规与隐私、安全组织与人员、基础设施安全、租户服务与租户安全、工程安全、运维运营安全、生态安全。

图 5-1 华为云安全责任共担模型



5.2 身份认证与访问控制

身份认证

用户访问ServiceStage的方式有两种，包括ServiceStage控制台、API。其本质都是通过ServiceStage提供的REST风格的API接口进行请求。调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权：

- Token认证：通过Token认证调用请求。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证获得操作API的权限。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。请求认证总是需要包含一个签名值，该签名值以请求者的AK/SK作为加密因子，结合请求体携带的特定信息计算而成。通过AK/SK认证方式进行认证鉴权，即使用AK/SK加密的方法来验证某个请求发送者身份。关于访问密钥的详细介绍及获取方式，请参见[访问密钥](#)。

访问控制

ServiceStage支持对资源、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，可以帮助您安全地控制云资源的访问。

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

- 关于IAM的详细介绍，请参见[IAM产品介绍](#)。
- ServiceStage支持的系统权限，请参见[权限管理](#)。

5.3 数据保护技术

为了确保您的个人数据（例如用户名、密码等）不被未经过认证、授权的实体或者个人获取，ServiceStage对用户数据的存储和传输进行加密保护，以防止个人数据泄露，保证您的个人数据安全。

收集范围

收集及产生的个人数据如表5-1所示：

表 5-1 数据收集范围

类型	收集方式	用途	是否可以修改	是否必须
仓库授权（包括账号、密码、OAuth授权、私人令牌，不同仓库支持的授权方式有所不同）	在创建仓库授权时由用户在界面输入仓库授权信息。	用于访问代码仓，代码构建时从代码仓拉取代码。	管理员权限可通过API修改。	是

数据存储安全

仓库授权信息通过安全公共组件、AES算法进行加密存储。

5.4 审计与日志

审计

云审计服务（Cloud Trace Service，以下简称CTS），是华为云安全解决方案中专业的日志审计服务，提供对各种云资源操作记录的收集、存储和查询功能，可用于支撑安全分析、合规审计、资源跟踪和问题定位等常见应用场景。

用户开通云审计服务并创建和配置追踪器后，CTS开始记录操作事件用于审计。开通方法请参考[CTS快速入门](#)。

开通云审计服务后，可[查看ServiceStage云审计日志](#)，云审计服务保存最近7天的操作日志。CTS支持追踪的ServiceStage操作列表，请参考[云审计服务支持的ServiceStage操作列表](#)。

CTS支持[配置关键操作通知](#)。您可将与ServiceStage相关的高危敏感操作，作为关键操作加入到CTS的实时监控列表中进行监控跟踪。当您使用ServiceStage服务时，如果触发了监控列表中的关键操作，那么CTS会在记录操作日志的同时实时发送通知。

日志

ServiceStage支持AOM能力，可以查看应用运行的相关日志，也可以在应用运维管理AOM中查看相关运行日志。

关于日志记录的详细介绍和配置方法，请参见[设置应用日志策略](#)。

5.5 服务韧性

- 冗余：全部服务都为无状态，利用基础设施层提供的“服务端负载均衡能力”将请求发送到不同实例当中，实现负载分摊。通过API网关对下游系统提供服务，API网关提供限流、熔断、降级等多种手段，并可对流量进行控制，保证升级不中断服务。
- 跨AZ容灾：服务实现跨AZ部署方式，将服务均匀分布在不同AZ中，服务支持跨AZ重建。当一个AZ的实例消失后，IaaS层将重新调度新的实例到其他AZ中，保证系统快速恢复到过去的负载能力，不过载。单AZ failure场景下，持久化数据满足完整性。
- 基础设施层服务提供CPU、内存、网络、磁盘等监控指标，应用层自动上报指标、日志等信息，对重要问题进行告警。
- 各服务具备限流能力，不会被大流量冲垮。
- 服务使用容器执行环境，IaaS层提供生命周期管理和容器调度。当遇到崩溃问题时，将重新调度并拉起新实例。

5.6 监控安全风险

ServiceStage支持配置应用日志策略，您可以在应用运维管理AOM中查看相关运行日志，请参考[设置应用日志策略](#)。

ServiceStage支持在部署应用组件过程中，或者应用组件已经部署后设置应用性能管理。应用性能管理可协助您快速进行应用问题定位与性能瓶颈分析，请参考[设置应用性能管理](#)。

6 微服务引擎版本支持机制

本章节为您介绍微服务引擎专享版的版本支持机制。

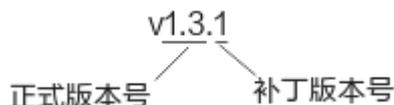
版本号说明

版本号格式为：{major}.{minor}.{patch}。

其中：

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

例如，v1.3.1。1.3为正式版本号，1为补丁版本号。



版本支持机制

- 微服务引擎创建
只能创建最新版本的微服务引擎，不支持创建指定版本微服务引擎。
- 微服务引擎维护
支持同时维护最新的3个正式版本。其余的版本，将不再提供技术支持，包括支持新的功能、社区bugfix回合、漏洞修复、升级等。
- 微服务引擎版本升级
 - 正式版本升级：
 - 1.3、1.2版本支持平滑升级到2.4及以后版本，且功能兼容。
 - 支持当前最新的3个正式版本中的2个较低版本升级到最新版本。例如，当前最新的3个正式版本为2.4、1.3、1.2，则支持由1.2、1.3升级到2.4。当引擎升级已超出可支持升级的版本范围，例如由1.0升级到1.3，可能导致微服务引擎的管理功能不可用，请谨慎操作。您可以通过[提交工单](#)，进行升级前风险评估。
 - 补丁版本升级，微服务引擎后台提供补丁版本自动升级，例如由1.3.0升级到1.3.1。

版本约束

微服务引擎版本升级后，不支持版本回滚。

7 约束与限制

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

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

虚拟机部署

应用组件采用虚拟机部署方式时，ServiceStage最多可以管理5000个虚拟机Agent，并允许其中的1500个虚拟机Agent不通过虚拟机环境代理直接和ServiceStage通信。

注册发现

微服务引擎专业版（Cloud Service Engine）限制请参见[表7-1](#)。

表 7-1 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7-2](#)。

表 7-2 微服务引擎专享版限制（最高规格）

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

微服务引擎专享版引擎类型说明

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

表 7-3 微服务引擎专享版引擎类型说明

引擎类型	适用业务场景	可用区CPU架构
集群	集群模式部署，主机级容灾，高可用。	x86或ARM，不支持混合架构。

微服务开发框架版本要求

微服务开发框架推荐版本如下表所示。

- 如果已经使用低版本的微服务开发框架构建应用，建议升级到推荐版本，以获取最稳定和丰富的功能体验。
- 如果已使用Spring Cloud微服务开发框架开发了应用，推荐使用**Spring Cloud Huawei**接入应用。
- Spring Cloud Huawei各分支版本与Spring Boot、Spring Cloud、Java Chassis及JDK编译版本的配套关系请参考**版本配套说明**。
- 如果基于开源开放和业界生态组件新开发微服务应用，可选择Spring Cloud框架。
- 如果希望使用微服务引擎提供的开箱即用的治理能力和高性能的RPC框架，可选择Java Chassis框架。

框架	推荐版本	说明
Spring Cloud Huawei	1.10.9-2021.0.x及以上	采用 Spring Cloud Huawei 项目提供接入支持： <ul style="list-style-type: none">● 适配的Spring Cloud版本为2021.0.5● 适配的Spring Boot版本为2.6.13 Spring Cloud微服务开发框架的版本说明请参见： https://github.com/huaweicloud/spring-cloud-huawei/releases 。
Java Chassis	2.7.10及以上	可以直接使用开源项目提供的软件包接入，不需要引用其他第三方软件包。 Java Chassis微服务开发框架的版本说明请参见： https://github.com/apache/servicecomb-java-chassis/releases 。

须知

系统升级、改造过程中，三方软件冲突是最常见的问题。随着软件迭代速度越来越快，传统的软件兼容性管理策略已经不适应软件的发展，您可以参考[三方软件版本管理策略](#)来解决版本冲突。

微服务引擎专享版配额说明

配额是指您在微服务引擎专享版实例中可创建的资源数量限制，具体的资源配额限制如表7-4所示。

表 7-4 微服务引擎资源配额限制

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

说明

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

8 产品规格差异

产品套餐说明

ServiceStage提供了基础版、专业版、铂金版，您可以根据需要选择，各个版本的功能说明如表8-1所示。

表 8-1 功能说明

功能		基础版	专业版	铂金版
管理规模	单个IAM账号下最大支持应用组件实例个数	100个	5000个	5000个
	单个组件支持的最大实例个数	200个		
微服务	资源包赠送微服务引擎	-	专享版（高可用）最多5个	专享版（高可用）最多10个
	Java语言微服务开发SDK	支持	支持	支持
	Spring Cloud微服务接入			
	服务注册中心			
	服务配置中心			
	实时仪表盘			
	负载均衡			
	服务限流			
	服务降级			
	服务容错			
	服务熔断			
	错误注入			
	黑白名单			

功能		基础版	专业版	铂金版
	灰度发布			
应用生命周期管理	多语言应用管理 (Java/Php/Python/Node.js/Tomcat/Docker)	支持	支持	支持
	应用生命周期管理 (灰度发布、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			
	云容器引擎部署			
	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			
持续交付	构建管理	支持	支持	支持
	源码仓库 (GitHub/GitLab/Gitee/Bitbucket/CodeArts)			
	编译任务 (Java/Php/Python/Node.js/Tomcat/Docker)			
	构建集群			
	流水线管理			
软件仓库	SWR软件包管理	支持	支持	支持
	Docker镜像包管理			
	仓库权限管理			
容器资源	虚拟机集群	支持	支持	支持
	裸金属集群			
	Windows集群			
	容器节点管理			
	容器存储管理			

微服务引擎实例规格说明

微服务引擎分为微服务引擎专业版和微服务引擎专享版两种产品规格：

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

引擎支持的最大实例规格说明如下。

表 8-2 微服务引擎实例规格说明

引擎类型	微服务实例数配额	配置条目数配额
微服务引擎专业版	20	-
微服务引擎专享版	100	600
	200	600
	500	3,000
	2,000	12,000

9 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，可以帮助您安全地控制云资源的访问。

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

如果云账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服务的其他功能。

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

ServiceStage 系统权限

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9-1所示，包括了ServiceStage的所有系统权限。推荐使用系统策略，系统角色仅用于兼容已有的权限配置。

表 9-1 ServiceStage 系统权限

系统角色/策略名称	描述	类别	依赖关系
ServiceStage FullAccess	应用管理与运维平台所有权限。	系统策略	无
ServiceStage ReadOnlyAccess	应用管理与运维平台只读权限。	系统策略	无
ServiceStage Development	应用管理与运维平台开发者权限。 拥有应用、组件、环境的操作权限，但无审批权限和基础设施创建权限。	系统策略	无
CSE FullAccess	微服务引擎管理员权限。	系统策略	无
CSE ReadOnlyAccess	微服务引擎查看权限。	系统策略	无
ServiceStage Administrator	应用管理与运维平台管理员，拥有该服务下的所有权限。	系统角色	Tenant Guest、Server Administrator、CCE Administrator、APM Administrator。
ServiceStage Operator	应用管理与运维平台操作员，拥有该服务下的只读权限。	系统角色	Tenant Guest
ServiceStage Developer	应用管理与运维平台开发者，拥有该服务下的所有权限，但无基础设施创建权限。	系统角色	Tenant Guest

如果表9-1所列的这些权限不满足实际需求，您可以参考表9-2和表9-3，在这个基础上自定义策略。

其中，“√”表示支持，“x”表示不支持。

表 9-2 ServiceStage 常用操作与系统权限之间的关系

操作	ServiceStage ReadOnlyAccess	ServiceStage Development	ServiceStage FullAccess
创建应用	x	√	√

操作	ServiceStage ReadOnlyAccess	ServiceStage Development	ServiceStage FullAccess
修改应用	x	√	√
查询应用	√	√	√
删除应用	x	√	√
创建组件	x	√	√
查询组件	√	√	√
部署组件	x	√	√
维护组件	x	√	√
删除组件	x	√	√
创建构建工程	x	√	√
修改构建工程	x	√	√
查询构建工程	√	√	√
启动构建工程	x	√	√
删除构建工程	x	√	√
创建流水线	x	√	√
修改流水线	x	√	√
查询流水线	√	√	√
启动流水线	x	√	√
克隆流水线	x	√	√
删除流水线	x	√	√
新建仓库授权	x	√	√
修改仓库授权	x	√	√
查询仓库授权	√	√	√
删除仓库授权	x	√	√

表 9-3 CSE 常用操作与系统权限之间的关系

操作	CSE ReadOnlyAccess	CSE FullAccess
创建微服务引擎	x	√
维护微服务引擎	x	√
查询微服务引擎	√	√
删除微服务引擎	x	√
创建微服务	x	√
查询微服务	√	√
维护微服务	x	√
删除微服务	x	√
创建微服务配置	x	√
查询微服务配置	√	√
编辑微服务配置	x	√
删除微服务配置	x	√
创建微服务治理策略	x	√
查询微服务治理策略	√	√
编辑微服务治理策略	x	√
删除微服务治理策略	x	√

细粒度权限

📖 说明

- SWR未支持细粒度权限，相关权限需要另外授权。
- 创建“计费模式”为“包月/包年”的微服务引擎专享版时：
 - 创建订单不支付，创建该引擎的用户需拥有“BSS Operator”权限，即费用中心（BSS）操作员，拥有账号中心和资源中心的所有菜单项执行任意操作的权限，在费用中心有相关的只读权限。
 - 创建订单并支付，创建该引擎的用户需拥有“BSS Administrator”权限，即费用中心（BSS）管理员，拥有该服务下的所有权限。

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

- CSE细粒度权限依赖说明请参见[表9-4](#)。
- ServiceStage细粒度权限依赖说明请参见[表9-5](#)。

表 9-4 CSE 细粒度权限依赖说明

权限名称	权限描述	权限依赖	应用场景
cse:engine:list	列出所有引擎	无	查看引擎列表。
cse:engine:get	查看引擎信息	cse:engine:list	查看引擎详情，仅微服务引擎专享版支持。
cse:engine:modify	修改引擎	<ul style="list-style-type: none"> cse:engine:list cse:engine:get 	修改引擎的操作包括：开启/关闭公网访问操作、开启/关闭安全认证操作、引擎失败任务重试操作，仅微服务引擎专享版支持。
cse:engine:upgrade	升级引擎	<ul style="list-style-type: none"> cse:engine:list cse:engine:get 	升级引擎的操作包括：引擎版本升级操作，仅微服务引擎专享版支持。
cse:engine:delete	删除引擎	<ul style="list-style-type: none"> cse:engine:list cse:engine:get vpc:ports:get vpc:ports:delete 	删除引擎，仅微服务引擎专享版支持。
cse:engine:create	创建引擎	<ul style="list-style-type: none"> cse:engine:get cse:engine:list ecs:cloudServerFlavors:get vpc:vpcs:get vpc:vpcs:list vpc:subnets:get vpc:ports:get vpc:ports:create 	创建引擎的操作包括：创建引擎操作，引擎备份/恢复任务创建操作，仅微服务引擎专享版支持。
cse:config:modify	服务配置管理修改	<ul style="list-style-type: none"> cse:engine:list cse:engine:get cse:config:get 	全局配置功能与治理功能涉及的配置修改。
cse:config:get	服务配置管理查看	<ul style="list-style-type: none"> cse:engine:list cse:engine:get 	查看服务的配置。

权限名称	权限描述	权限依赖	应用场景
cse:governance:modify	服务治理中心修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:config:get ● cse:config:modify ● cse:registry:get ● cse:registry:modify ● cse:governance:get 	创建与修改服务治理。
cse:governance:get	服务治理中心查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:config:get ● cse:registry:get 	查看服务治理功能。
cse:registry:modify	服务注册管理修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get 	服务修改。
cse:dashboard:modify	dashboard管理修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get ● cse:dashboard:get ● cse:registry:modify 	仪表盘修改。
cse:dashboard:get	dashboard管理查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get 	仪表盘查看。
cse:registry:get	服务注册管理查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get 	服务目录查看。

📖 说明

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

表 9-5 ServiceStage 细粒度权限依赖说明

权限名称	权限描述	权限依赖	应用场景
servicestage:app:get	查询应用信息	servicestage:app:list	查询应用信息。

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

权限名称	权限描述	权限依赖	应用场景
servicestage:pipeline:get	查看流水线信息	<ul style="list-style-type: none"> servicestage:pipeline:list servicestage:asssembling:get servicestage:asssembling:list 	查看流水线信息。
servicestage:pipeline:create	创建流水线	<ul style="list-style-type: none"> servicestage:pipeline:list servicestage:pipeline:get servicestage:asssembling:create servicestage:asssembling:get servicestage:asssembling:list 	创建流水线。
servicestage:pipeline:modify	修改流水线	<ul style="list-style-type: none"> servicestage:pipeline:get servicestage:pipeline:list servicestage:asssembling:modify servicestage:asssembling:get servicestage:asssembling:list 	修改流水线。
servicestage:pipeline:delete	删除流水线	<ul style="list-style-type: none"> servicestage:pipeline:get servicestage:pipeline:list servicestage:asssembling:get servicestage:asssembling:list servicestage:asssembling:delete 	删除流水线。
servicestage:pipeline:list	查看流水线列表	<ul style="list-style-type: none"> servicestage:asssembling:get servicestage:asssembling:list 	查看流水线列表

权限名称	权限描述	权限依赖	应用场景
servicestage:pipeline:execute	执行流水线	<ul style="list-style-type: none"> servicestage:pipeline:get servicestage:pipeline:list servicestage:assembling:modify servicestage:assembling:get servicestage:assembling:list servicestage:app:get servicestage:app:list servicestage:app:modify 	执行流水线。
servicestage:assembling:get	查看构建信息	servicestage:assembling:list	查看构建信息。
servicestage:assembling:create	创建构建	<ul style="list-style-type: none"> servicestage:assembling:get servicestage:assembling:list 	创建构建。
servicestage:assembling:modify	修改构建	<ul style="list-style-type: none"> servicestage:assembling:get servicestage:assembling:list 	修改构建。
servicestage:assembling:delete	删除构建	<ul style="list-style-type: none"> servicestage:assembling:get servicestage:assembling:list 	删除构建。
servicestage:assembling:list	查看构建列表	无	查看构建列表。

ServiceStage 控制台功能依赖的角色或策略

如果IAM用户需要在ServiceStage控制台拥有相应功能的查看或使用权限，请确认已经对该用户所在的用户组设置了ServiceStage Administrator、ServiceStage FullAccess或ServiceStage ReadOnlyAccess权限，再参考表9-6增加依赖服务的角色或策略。

表 9-6 ServiceStage 控制台依赖服务的角色或策略

控制台功能	依赖服务	需配置角色/策略
<ul style="list-style-type: none"> 总览 告警 运维监控 	应用运维管理 AOM	<ul style="list-style-type: none"> IAM用户设置了ServiceStage Administrator权限后，需要增加AOM FullAccess权限后才能访问总览、告警、运维监控中的数据图表。 支持设置了IAM ReadOnlyAccess和ServiceStage FullAccess或ServiceStage ReadOnlyAccess权限的IAM用户直接访问总览、告警、运维监控中的数据图表。
性能管理	应用性能管理 APM	如果需要使用Java探针，需要设置AOM FullAccess和APM FullAccess权限。
组件管理	弹性伸缩组 AS	如果需要使用AS资源用于虚拟机环境下部署组件，需要设置AutoScaling FullAccess权限。
	云容器引擎 CCE	如果需要使用CCE资源用于容器环境下部署组件，需要设置CCE FullAccess权限。
	弹性云服务器 ECS	如果需要使用ECS资源用于虚拟机环境下部署组件，需要设置ECS ReadOnlyAccess权限。
	对象存储服务 OBS	如果部署的组件来源于存储在OBS的软件包，需要设置OBS ReadOnlyAccess权限。
微服务引擎	微服务引擎 CSE	如果需要为微服务组件绑定CSE用于服务注册、服务治理、配置管理，需要设置CSE FullAccess权限。
分布式缓存	分布式缓存服务 DCS	如果需要为容器环境下部署的组件绑定DCS用于在应用运行时读取环境变量来获取分布式缓存的相关信息，需要设置DCS ReadOnlyAccess权限。
数据存储	云硬盘服务 EVS	如果容器环境下部署的组件需要使用云硬盘存储方式存储数据，需要设置EVS ReadOnlyAccess权限。
	弹性文件服务 SFS	如果容器环境下部署的组件需要使用极速文件存储方式存储数据，需要设置SFS Turbo ReadOnlyAccess权限。
	对象存储服务 OBS	如果容器环境下部署的组件需要使用对象存储方式存储数据，需要设置OBS ReadOnlyAccess权限。
云数据库	云数据库 RDS	如果需要为容器环境下部署的组件绑定RDS用于应用数据持久化存储，需要设置RDS ReadOnlyAccess权限。

控制台功能	依赖服务	需配置角色/策略
<ul style="list-style-type: none"> 组件VPC内网访问 组件域名访问 	弹性负载均衡 ELB	如果需要为组件设置VPC内网访问或域名访问可以使用组件提供的服务，需要设置ELB ReadOnlyAccess权限。
组件公网访问	NAT网关 NAT	如果需要为组件设置通过NAT网关公网访问可以使用组件提供的服务，需要设置NAT ReadOnlyAccess权限。
	弹性公网IP EIP	如果需要为组件设置通过EIP公网访问可以使用组件提供的服务，需要设置EIP ReadOnlyAccess权限。
	弹性负载均衡 ELB	如果需要为组件设置通过ELB公网访问可以使用组件提供的服务，需要设置ELB ReadOnlyAccess权限。
组件日志	云日志服务 LTS	如果需要通过对接LTS，查看、搜索、导出组件运行LTS日志记录用于排查和解决组件运行中出现的问题，需要设置LTS FullAccess权限。
阈值告警	消息通知服务 SMN	如果需要将容器环境下部署组件产生的阈值告警消息由消息通知服务SMN发送通知给用户，需要设置SMN ReadOnlyAccess权限。
镜像仓库	容器镜像仓库 SWR	如果容器环境下部署的组件来源于存储在SWR的镜像包，需要设置SWR FullAccess权限。
标签管理	标签管理服务 TMS	如果需要使用TMS对组件等管理对象设置标签用于管理和选择，需要设置TMS ReadOnlyAccess权限。
环境管理	虚拟私有云 VPC	VPC用于创建环境时，将用于组件部署和运行的计算、网络、中间件等基础资源隔离在同一个虚拟网络环境中。需要设置VPC ReadOnlyAccess权限。

相关链接

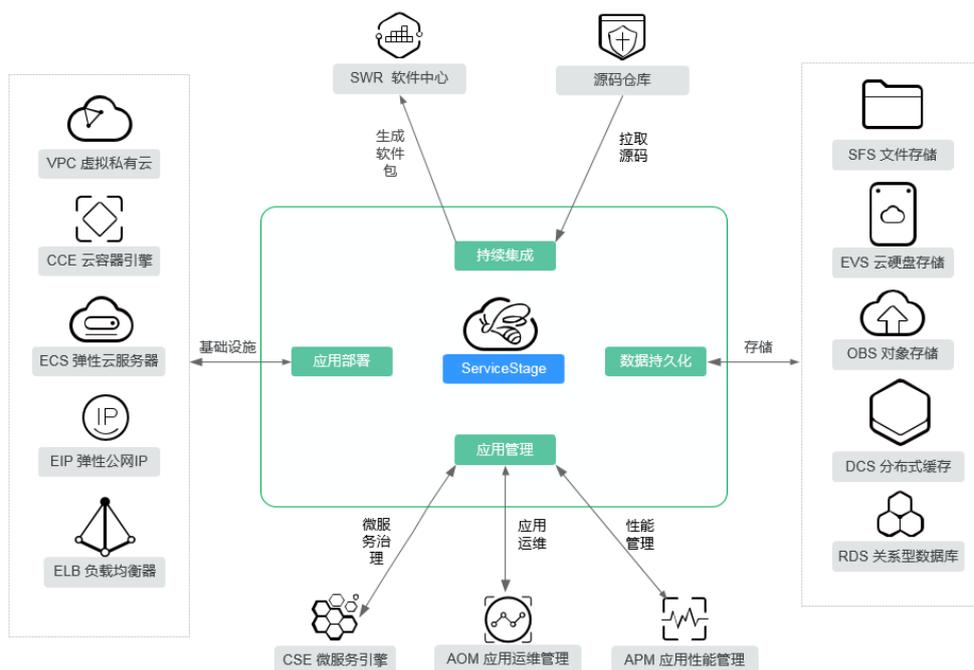
- [IAM产品介绍](#)
- [创建用户并授权使用ServiceStage](#)

10 与其他服务的关系

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

使用ServiceStage可以系统的体验到众多云服务的功能精髓，ServiceStage集成云服务如图10-1所示。

图 10-1 ServiceStage 集成的云服务



- ServiceStage实现了与源码仓库的对接（如CodeArts、Gitee、GitHub、GitLab、Bitbucket），绑定源码仓库后，可以直接从源码仓库拉取源码进行构建。
- ServiceStage集成了部署源管理功能，可以将构建完成的软件包（或者镜像包）归档对应的仓库和组织。
- ServiceStage集成了相关的基础资源（如VPC、CCE、ECS、EIP、ELB），在部署应用时可以直接使用已有或者新建所需的基础资源。

- ServiceStage集成了微服务引擎，进入ServiceStage控制台可以进行微服务治理相关的操作。
- ServiceStage集成了应用运维管理及应用性能管理服务，可以进行应用运维及性能监控相关的操作。
- ServiceStage集成了存储、数据库、缓存等服务，通过简单配置即可实现数据持久化存储。

11 基本概念

环境

环境是用于组件部署和运行的计算（如云容器引擎 CCE、弹性云服务器 ECS等）、网络（如弹性负载均衡 ELB、弹性IP EIP等）和中间件（如分布式缓存 DCS、云数据库 RDS等）等基础资源的集合。ServiceStage把多种基础资源组合为一个环境，如：开发环境、测试环境、预生产环境和生产环境。按环境维度来管理资源、部署组件，可以减少具体基础资源运维管理的复杂性。

基础资源

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

应用

应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。

组件

组件是组成应用的某个业务特性实现，以代码或者软件包为载体，可独立部署在环境下运行。

技术栈

技术栈包含组件运行所依赖的操作系统、框架、运行时，由技术栈名称、类型、状态、版本等属性构成。其中，版本号遵从[语义化版本控制规范](#)。ServiceStage为您提供并管理技术栈生命周期，您只需聚焦具体业务开发，提升您的应用托管体验。

ServiceComb

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

微服务

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

说明

在ServiceStage中，微服务对应组件。

微服务实例

一个微服务的最小运行和部署单元，通常对应一个应用进程。

全链路流量控制

全链路流量控制，将应用组件的相关版本隔离成一个独立的运行环境（即泳道），通过设置流控路由规则，将满足规则的请求流量路由到目标版本应用。

泳道

泳道用于为相同版本组件定义一套隔离环境。只有满足了流控路由规则的请求流量才会通过标签控制流量转发到对应泳道的目标服务组件。一个组件只能属于一个泳道，一个泳道可以包含多个组件。

泳道组

泳道组是一组泳道的集合，用于区分不同的组织或场景。

基线泳道

基线泳道关联应用中所有的基线版本组件，当微服务调用链中不存在某个组件的时候，会默认访问基线泳道中的组件。

灰度泳道

灰度泳道用于部署组件灰度版本。

发布单

发布单是应用发布的入口，提供单组件发布、批量发布、批量升级和批量克隆功能，且支持组件依赖编排。

配置

在ServiceStage中，配置就是一个文件。通过创建统一的配置文件，您可以将环境和应用的系统变量（例如环境关联的IP、端口、数据库地址，应用关联的应用名称等）填充到配置中，生成一个配置文件。组件关联配置文件进行部署时，系统变量会自动替换为实际值，通过文件挂载实现一次配置、多环境使用。