

微服务引擎

# 产品介绍

文档版本 01  
发布日期 2024-10-15



版权所有 © 华为云计算技术有限公司 2024。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

---

<b>1 图解微服务引擎</b>	<b>1</b>
<b>2 什么是微服务引擎</b>	<b>2</b>
<b>3 应用场景</b>	<b>7</b>
<b>4 基本概念</b>	<b>10</b>
<b>5 ServiceComb 引擎版本支持机制</b>	<b>13</b>
<b>6 安全</b>	<b>15</b>
6.1 责任共担	15
6.2 身份认证和访问控制	16
6.3 数据保护技术	17
6.4 故障恢复	17
6.5 审计与日志	18
<b>7 产品规格</b>	<b>19</b>
<b>8 约束与限制</b>	<b>21</b>
<b>9 权限管理</b>	<b>28</b>
<b>10 与其他服务的关系</b>	<b>33</b>

# 1 图解微服务引擎



# 2 什么是微服务引擎

微服务引擎（Cloud Service Engine，CSE），是用于微服务应用的云中间件，支持华为云贡献到Apache社区的注册配置中心Servicecomb引擎、开源增强的注册配置中心Nacos引擎。用户可结合其他云服务，快速构建云原生微服务体系，实现微服务应用的快速开发和高可用运维。

- 开源开放  
微服务核心框架ServiceComb已在Apache开源，提供Spring Cloud、Service Mesh商业版。
- 可靠稳定  
核心业务CloudNative转型基础底座，经终端业务亿级用户考验。
- 专业服务  
与100多家行业合作伙伴联合提供微服务技术、工具和流程转型的咨询服务。
- 多语言  
提供Java、Go、.NET、Node.js等多语言微服务解决方案。

## Nacos 引擎

CSE Nacos是基于开源Nacos 2.x版本开发的一款微服务注册发现、配置管理平台。支持多种开发语言、框架的接入使用，具有基于DNS的服务发现能力。

### 📖 说明

Nacos引擎当前仅在华东二、中国-香港、亚太-新加坡、中东-利雅得和拉美-墨西哥城二支持。

Nacos的关键特性包括：

表 2-1 Nacos 支持特性

功能	特性
实例管理	创建实例。
	查看实例规格。
	查看实例列表。
	查看实例详情。

功能	特性
	实例删除。
	按需转包周期。
	企业项目。
规格变更	扩容变更。
连接管理	内网访问，通过ip和域名访问。
命名空间管理	查看命名空间清单。
	查看命名空间详情。
	创建命名空间。
	编辑命名空间。
	删除命名空间。
	命名空间数量限制。 <b>说明</b> 一共可创建50个命名空间。
服务管理	过滤命名空间。
	搜索服务。
	过滤空服务。
	查看服务清单。
	创建服务。
	查看服务提供者。
	查看服务订阅者。
	实例按集群区分。
	提供者支持按元数据过滤。
	支持服务节点上下线。
支持服务节点权重编辑	
配置管理	创建配置。
	导入配置。
	编辑配置。
	删除配置。
	配置灰度发布。
	单命名空间配置数量限制。
	搜索配置。

功能	特性
	查看配置清单。
	查看配置详情。
	查看历史版本。
	回滚到历史版本。
	配置内容对比。
	配置监听查询。

## ServiceComb 引擎

ServiceComb引擎采用的注册发现中心Apache ServiceComb Service Center，是一个RESTful风格的、高可用无状态的服务注册发现中心，提供微服务发现和微服务管理功能。服务提供者可以将自身的实例信息注册到服务注册发现中心，以供服务消费者发现并使用。ServiceComb引擎可无缝兼容Spring Cloud、ServiceComb等开源生态。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>

### 📖 说明

ServiceComb引擎当前仅在华东二、中国-香港、中东-利雅得、土耳其-伊斯坦布尔、亚太-新加坡和亚太-雅加达支持。

ServiceComb引擎分为1.x、2.x版本。

ServiceComb引擎2.x版本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，引擎创建完成后不支持规格变更；引擎资源独享，性能不受其他租户影响。

相较于ServiceComb引擎1.x版本，ServiceComb引擎2.x版本底层架构、功能、安全及性能全面升级，提供了独立的服务注册发现中心和配置中心，支持基于用户业务场景的定义和治理。两个版本的特性比对请参见表2-2。

表 2-2 ServiceComb 引擎 2.x 和 ServiceComb 引擎 1.x 特性比对

功能	特性		2.x	1.x	备注
引擎管理	安全性	支持安全认证	√	√	-
	可靠性	3AZ高可靠	√	√	-
微服务管理	基础能力	注册发现	√	√	-
		多框架接入	√	√	支持Spring Cloud、ServiceComb Java Chassis。

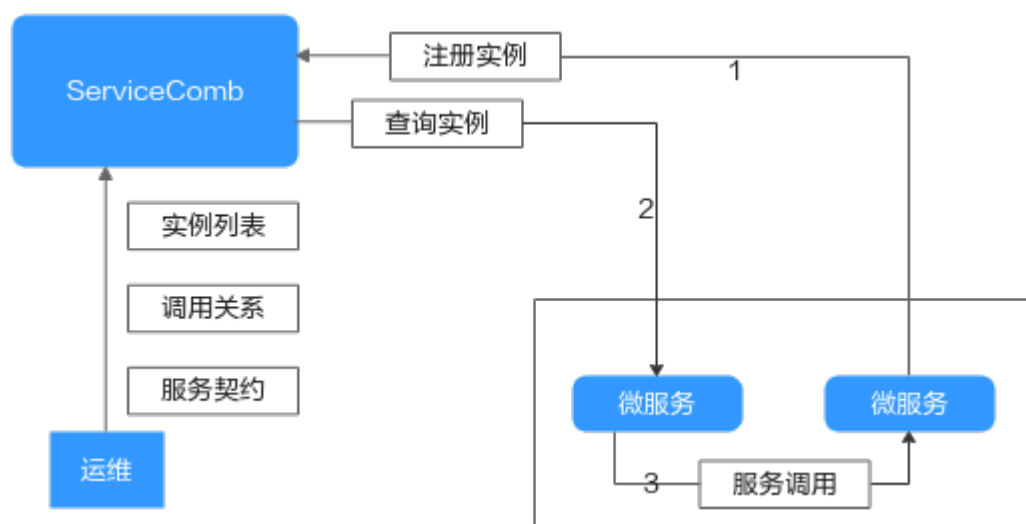
功能	特性		2.x	1.x	备注
		无实例版本自动清理	√	x	2.3.7及以后版本，支持保留最近3个微服务版本，并自动清理无实例版本。
	性能	实例变化毫秒级推送	√	√	-
配置管理	基础能力	管理配置	√	√	-
		配置格式多样化	√	仅支持文本	2.x新增支持配置格式有：YAML、JSON、TEXT、Properties、INI、XML。
		导入导出	√	√	2.x新增支持设置导入相同配置策略。
	高级特性	历史版本	√	x	-
		版本对比	√	x	-
		一键回滚	√	x	-
		配置标签	√	x	-
	性能	秒级下发	√	x	-
微服务治理	业务场景化治理	业务场景定义	√	x	-
		基于请求Method的匹配规则	√	x	-
		基于请求Path的匹配规则	√	x	-
		基于请求Headers的匹配规则	√	x	-
	治理策略-流量控制	服务端的令牌桶限流	√	√	-
	治理策略-重试	客户端通过重来保证用户业务的可用性、容错性、一致性	√	√	-
	治理策略-熔断	服务端通过熔断故障业务，防止故障蔓延到整个服务，发生大规模故障	√	√	-



功能	特性	2.x	1.x	备注	
	治理策略- 隔离仓	服务端基于信号 量控制请求并发 能力	√	x	-
开发工 具	本地轻量 化引擎	本地一键启动， 方便开发者离线 开发微服务	√	√	-

# 3 应用场景

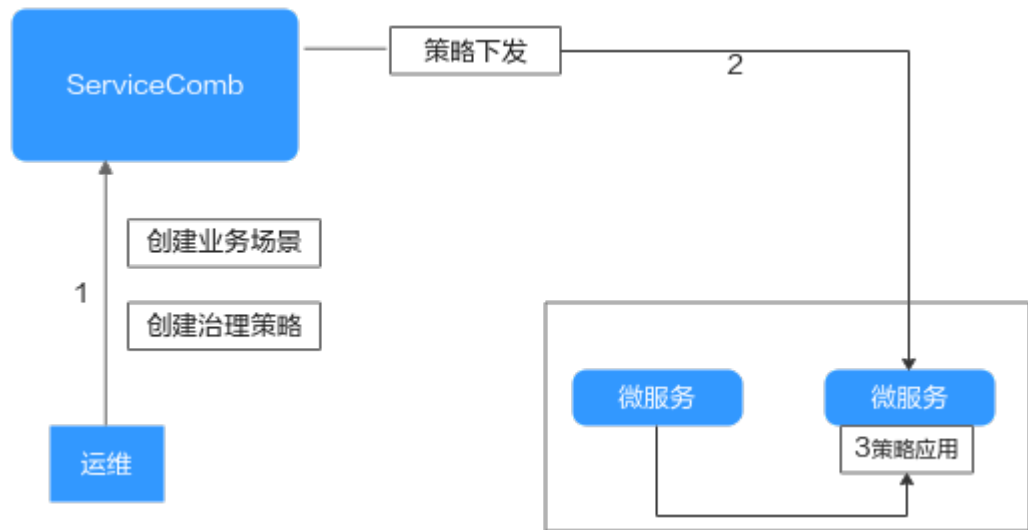
## 微服务注册发现



微服务启动时，将实例信息注册到CSE，包括实例的基本信息，如应用名称、微服务名称、版本号、服务契约、实例地址等；微服务需要调用其他微服务的接口时，从CSE查询实例信息，并将实例信息缓存到本地，缓存会通过事件通知、定时查询等机制更新；通过本地缓存的地址信息，实现微服务之间的点到点调用，当微服务存在多个实例的情况，可以配置不同的负载均衡策略，包括轮询、权重、灰度发布、同可用区优先等。

运维层面，可以通过CSE查看实例列表、微服务调用关系、服务契约等，帮助客户了解应用系统组成、运行状态等信息。

## 服务治理

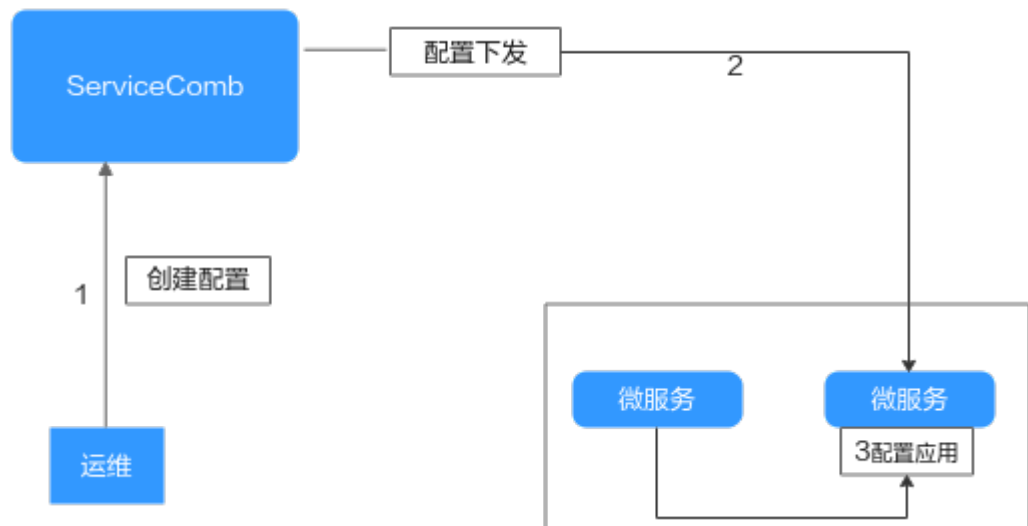


服务治理有两种开启策略：

1. 一种是开发态就配置好，这种策略适合通用型，满足全业务场景的策略，比如负载均衡策略。
2. 另外一种是在运行态修改，这种策略适合根据业务变化需要动态调整的场景，比如限流策略。

两种策略内部实现机制一般是相同的，需要配置管理系统下发治理策略和运行时SDK执行治理策略，运行时治理SDK包含在用户选择的微服务开发框架及其扩展包里面，与微服务一起编译集成。CSE控制台提供了常见的治理策略管理功能，方便用户根据业务变化，调整治理策略。

## 配置管理



配置管理集中管理微服务的配置，配置管理根据配置项的属性，比如作用域、标签等，将配置下发到具体的微服务。

配置管理提供一系列方便的功能，比如查看历史版本和配置回滚、配置导入导出、灵活的作用域管理等，满足用户管理复杂环境和海量微服务的业务诉求。

# 4 基本概念

## 通用概念

概念	说明
微服务	<p>微服务是业务概念，即提供某种服务的某个进程。</p> <p>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)。</p> <p>多个微服务组成应用程序。</p>
实例	<p>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机，可以实现多个实例同时运行。</p>
配置	<p>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。</p>

## Nacos 引擎相关概念

概念	说明
命名空间	<p>用于进行租户粒度的配置隔离。命名空间可以实现不同环境的配置的分隔离，例如开发测试环境和生产环境的资源（如配置、服务）隔离等。</p>
配置集	<p>配置项的集合称为配置集。通常一个配置文件就是一个配置集，包含了系统各个方面的配置。</p>

概念	说明
配置集ID	Nacos中的某个配置集的ID。一个系统或者应用可以包含多个配置集，每个配置集都可以被一个名称标识。
分组	Nacos中的一组配置集，是组织配置的维度之一。通过字符串对配置集进行分组，从而区分配置集ID相同的配置集。如果未填写配置分组的名称，则默认为DEFAULT_GROUP。
保护阈值	保护阈值与集群中健康实例的占比有关。如果健康实例占比小于或等于此值，会触发保护阈值，Nacos会将全部实例（健康实例 + 非健康实例）全部返回给调用者。保护阈值未触发时，Nacos只会把健康实例返回给调用者。
灰度发布	在配置正式发布前可先进行小部分的发布验证，待配置没问题后再正式发布，从而降低配置推送的风险。
权重	实例级别的配置。权重为浮点数。权重越大，分配给该实例的流量越大。
元数据	Nacos数据（如配置和服务）描述信息，如服务版本、权重等，从作用范围来看，可分为服务级别的元信息、集群的元信息及实例的元信息。

## ServiceComb 引擎相关概念

概念	说明
版本	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。
契约	<p>微服务场景中的契约是指服务契约，它是基于OpenAPI规范的微服务接口约束规范，是服务端和消费端对于接口的定义。</p> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>• Java Chassis默认使用契约功能。</li> <li>• Spring Cloud默认不使用契约功能，如果Spring Cloud需要使用契约功能，需要引入以下依赖：</li> </ul> <pre>&lt;dependency&gt;   &lt;groupId&gt;com.huaweicloud&lt;/groupId&gt;   &lt;artifactId&gt;spring-cloud-starter-huawei-swagger&lt;/artifactId&gt; &lt;/dependency&gt;</pre>

概念	说明
应用	完成某项完整业务场景的软件系统。应用一般由多个微服务组成，应用里面的微服务能够相互发现和调用。
环境	服务中心建立的一个逻辑概念，比如 development、production等。不同环境里面的微服务实例逻辑隔离、无法相互发现和调用。
治理策略	微服务治理中的一个概念，指当前治理使用的某种方法。每一种治理策略都可以绑定一个业务场景，同一种策略不可以绑定多种业务场景。不同的治理策略可以绑定同一个业务场景。
业务场景	用来描述治理中的治理策略作用生效的条件，一个业务场景可以绑定多个不同的治理策略。

# 5 ServiceComb 引擎版本支持机制

本章节为您介绍ServiceComb引擎的版本支持机制。

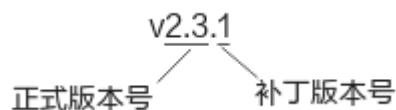
## 版本号说明

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，v2.3.1。2.3为正式版本号，1为补丁版本号。



## 版本支持机制

- 引擎创建  
只能创建最新版本的ServiceComb引擎，不支持指定版本创建引擎。
- 引擎维护  
CSE后台同时支持维护最新的3个正式版本。其余的版本，CSE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。
- 引擎升级
  - 正式版本升级，支持当前最新的3个正式版本中的2个较低版本升级到最新版本。例如，当前最新的3个正式版本为2.3、2.2、2.1，则支持由2.1、2.2升级到2.3。

### 📖 说明

当引擎升级已超出可支持升级的版本范围，例如由2.0升级到2.3，可能导致ServiceComb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，CSE后台提供补丁版本自动升级，例如由2.3.0升级到2.3.1。



## 版本约束

引擎版本升级后，不支持版本回滚。

# 6 安全

## 6.1 责任共担

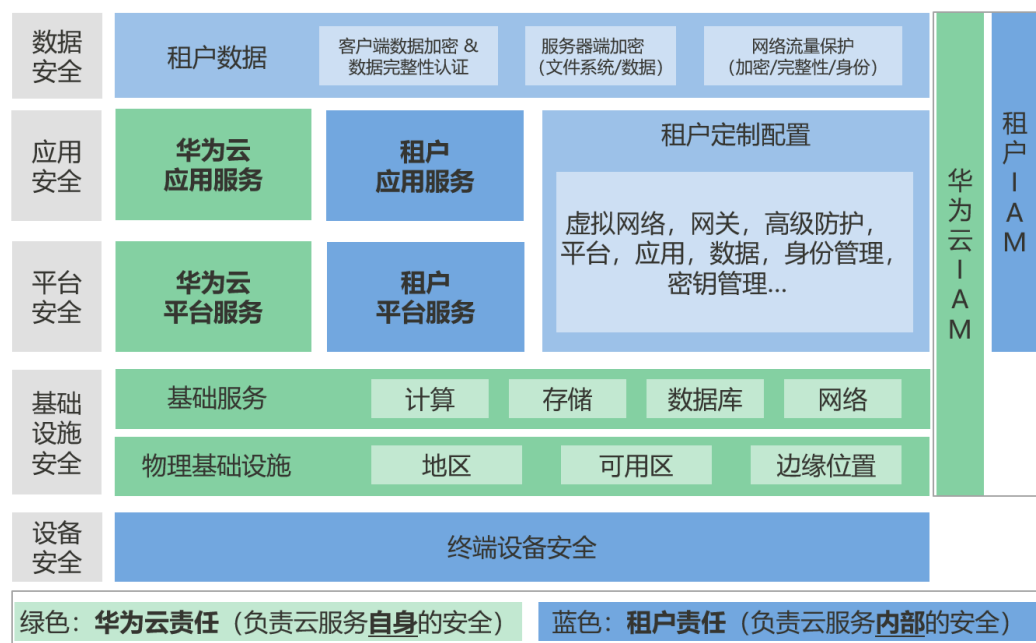
华为云秉承“将对网络和业务安全性保障的责任置于公司的商业利益之上”。针对层出不穷的云安全挑战和无孔不入的云安全威胁与攻击，华为云在遵从法律法规业界标准的基础上，以安全生态圈为护城河，依托华为独有的软硬件优势，构建面向不同区域和行业的完善云服务安全保障体系。

安全性是华为云与您的共同责任，如图6-1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 6-1 华为云安全责任共担模型



## 6.2 身份认证和访问控制

### 身份认证

CSE可以通过RBAC（Role-Based Access Control，基于角色的访问控制）认证。

RBAC认证是指权限与角色相关联，您可以使用关联了admin角色权限的账号创建新账号，根据实际业务需求把合适的角色同账号关联。使用该账号的用户则具有对该微服务引擎相应的访问和操作权限，详情请参见[安全认证概述](#)。

### 访问控制

如果您需要给企业中的员工对CSE的资源设置不同的访问权限，以达到不同员工之间的权限隔离，CSE支持使用统一身份认证服务（Identity and Access Management，简称IAM）和自定义策略进行精细的权限管理。

访问控制方式	简要说明	详细介绍
IAM权限控制	IAM权限是作用于云资源的，IAM权限定义了允许和拒绝的访问操作，以此实现云资源权限访问控制。管理员创建IAM用户后，需要将用户加入到一个用户组中，IAM可以对这个组授予CSE所需的权限，组内用户自动继承用户组的所有权限。	<a href="#">IAM权限介绍</a>

访问控制方式	简要说明	详细介绍
自定义策略权限控制	同一个微服务引擎可能会有多个用户共同使用，而不同的用户根据其责任和权限，需要具备不同的微服务引擎访问和操作权限。开启了“安全认证”的微服务引擎专享版，通过微服务控制台提供了基于RBAC的系统管理功能，根据实际的业务需求对角色进行权限自定义策略控制。	<a href="#">自定义策略权限控制介绍</a>

## 6.3 数据保护技术

CSE通过多种数据保护技术保障存储在CSE中的数据安全可靠，其数据保护技术如下表所示。

数据保护手段	简要说明	详细介绍
HTTPS传输	CSE使用HTTPS协议保障信息传输安全。	<a href="#">构造请求</a>
创建多AZ引擎	CSE支持跨AZ部署，为了保障可靠性，推荐使用跨AZ部署的CSE引擎。	<a href="#">创建ServiceComb引擎</a>
多版本控制	CSE中可以存储多个配置版本，帮助您进行查看管理以及快速回滚。	<a href="#">对比配置版本</a>
配置文件加密	CSE支持配置文件加密存储，保障敏感数据安全。	<a href="#">配置文件加密方案</a>

## 6.4 故障恢复

### 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自动备份，也支持手动创建特定时间点对微服务引擎进行备份，详情请参见[设置ServiceComb引擎备份与恢复](#)。

### 多可用区

可用区指在同一区域下，电力、网络隔离的物理区域，可用区之间内网互通，不同可用区之间物理隔离。CSE支持跨可用区部署微服务引擎集群，以提供AZ级故障时的高可用性。详情请参见[创建ServiceComb引擎](#)。

## 6.5 审计与日志

### 审计

云审计服务（Cloud Trace Service, 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可记录CSE的管理事件和数据事件用于审计。CTS的详细介绍和开通配置方法，请参见[CTS快速入门](#)。CTS支持追踪的CSE管理事件和数据事件列表，请参见[云审计服务支持的CSE操作列表](#)。

### 日志

登录微服务引擎控制台，在“操作日志”区域，可以查看微服务引擎的操作日志信息，详情请参见[查看微服务引擎操作日志](#)。

# 7 产品规格

## Nacos 引擎实例规格

您可根据实际需要选择合适的Nacos引擎实例规格。

表 7-1 Nacos 引擎实例规格

微服务实例数	容量单元	客户端连接数
500实例	10个	1,000个
1,000实例	20个	2,000个
2,000实例	40个	4,000个
3,000实例	60个	6,000个
5,000实例	100个	10,000个

### 说明

- 当您需要Nacos引擎微服务实例数大于2,000时需要[提交工单](#)申请。
- 1个容量单元=50个微服务实例。

## ServiceComb 引擎实例规格

您可根据需要托管的微服务实例数量，参考[表7-2](#)来选择ServiceComb引擎实例规格。

如[表7-2](#)所示，不同微服务实例数配额的ServiceComb引擎实例会赠送相应数量的配置条目数配额和最大支持微服务版本总数。

表 7-2 ServiceComb 引擎实例规格

微服务实例数配额	配置条目数配额
100实例	600
200实例	600

微服务实例数配额	配置条目数配额
500实例	3,000
2,000实例	12,000

# 8 约束与限制

## Nacos 引擎版本兼容性说明

CSE Nacos引擎版本	兼容Nacos社区版本	兼容性
2.1.0.x	2.1.0	100%

CSE Nacos引擎很好地兼容社区Nacos 2.1.0及其以下版本，且向下兼容。例如：微服务使用开源Nacos 1.x.x版本，也可以使用CSE Nacos引擎。

配置中心兼容Nacos 1.0起所有的客户端版本，服务发现兼容Nacos 1.2起的所有客户端版本，因此建议使用Nacos 1.2.0之后客户端版本。Nacos 2.x.x版本新增了长连接功能，需要使用长连接能力请升级至2.x.x版本。

Nacos 2.1.0版本后，默认关闭了双写能力，因此无法支持从Nacos 1.x版本平滑升级到2.1.0+的能力，如需要使用平滑升级能力，从Nacos 1.x直接升级到Nacos 2.1.0+版本，需要在application.properties文件中配置参数  
nacos.core.support.upgrade.from.1x=true。

## Nacos 引擎与微服务框架版本关系

CSE Nacos引擎版本	Spring Cloud Alibaba版本	Spring Cloud版本	Spring Boot版本
2.1.0.x	2022.0.0.0-RC*	Spring Cloud 2022.0.0	3.0.0
	2021.0.4.0*	Spring Cloud 2021.0.4	2.6.11
	2021.0.1.0	Spring Cloud 2021.0.1	2.6.3
	2021.1	Spring Cloud 2020.0.1	2.4.2



CSE Nacos 引擎版本	Spring Cloud Alibaba版本	Spring Cloud版本	Spring Boot版本
	2.2.10-RC1*	Spring Cloud Hoxton.SR12	2.3.12.RELEASE
	2.2.9.RELEASE	Spring Cloud Hoxton.SR12	2.3.12.RELEASE
	2.2.8.RELEASE	Spring Cloud Hoxton.SR12	2.3.12.RELEASE
	2.2.7.RELEASE	Spring Cloud Hoxton.SR12	2.3.12.RELEASE
	2.2.6.RELEASE	Spring Cloud Hoxton.SR9	2.3.2.RELEASE
	2.2.1.RELEASE	Spring Cloud Hoxton.SR3	2.2.5.RELEASE
	2.2.0.RELEASE	Spring Cloud Hoxton.RELEASE	2.2.X.RELEASE
	2.1.4.RELEASE	Spring Cloud Greenwich.SR6	2.1.13.RELEASE
	2.1.2.RELEASE	Spring Cloud Greenwich	2.1.X.RELEASE
	2.0.4.RELEASE(停止维护, 建议升级)	Spring Cloud Finchley	2.0.X.RELEASE
	1.5.1.RELEASE(停止维护, 建议升级)	Spring Cloud Edgware	1.5.X.RELEASE

## ServiceComb 引擎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ServiceComb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 <b>Spring Cloud Huawei</b> 项目提供接入支持： <ul style="list-style-type: none"> <li>• 适配的Spring Cloud版本为2021.0.5</li> <li>• 适配的Spring Boot版本为2.6.13</li> </ul> Spring Cloud微服务开发框架的版本说明请参见： <a href="https://github.com/ HuaweiCloud/spring-cloud-huawei/releases">https://github.com/HuaweiCloud/spring-cloud-huawei/releases</a> 。
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： <a href="https://github.com/apache/servicecomb-java-chassis/releases">https://github.com/apache/servicecomb-java-chassis/releases</a> 。

### 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。

## Spring-cloud-huawei、Servicecomb 及 Sermant 功能对比

一级特性	二级特性	serviccomb-java-chassis	spring-cloud-huawei	sermant agent	备注
微服务网关	服务端限流	√	√	√	-
	服务端隔离仓	√	√	√	-
	客户端熔断	×	√	×	-
	客户端容错	×	√	×	-
	客户端降级	×	×	×	-
	客户端故障注入	×	×	×	-
	负载均衡策略	√	√	×	-
	灰度发布	×	√	√	-
微服务治理	优雅停机	√	√	×	-
	优雅上下线	√	√	√	-

	无损升级	√	√	√	-
	服务端限流	√	√	√	-
	客户端容错	√	√	√	-
	客户端熔断	√	√	√	-
	客户端降级	√	√	√	-
	服务端隔离仓	√	√	√	-
	客户端隔离仓	√	√	√	-
	负载均衡策略	√	√	√	-
	灰度发布	√	√	√	-
	全链路日志追踪	√	√	×	-
	服务治理状态上传	√	√	×	-
	快速失败	√	√	×	-
	故障注入	√	×	√	-
	黑白名单	√	√	×	-
注册发现	本地注册发现	√	√	×	-
	单注册-CSE	√	√	√	-
	单注册-ServiceCenter	√	√	√	-
	双注册	×	×	√	双注册指同时注册到两个注册中心，当前sermant支持同时注册到cse和宿主原生注册中心。
配置中心支持	servicecomb引擎	√	√	√	可基于配置中心下发配置，例如服务治理规则、业务配置。
	Nacos引擎	√	√	√	
	servicecomb-kie	√	√	√	
	zookeeper	×	×	√	

	轻量化配置中心 ( zero-config )	√	×	×	
	apollo	×	×	×	
安全特性	安全认证	√	√	×	服务实例与注册中心以及消费端与生产端之间的认证。
开发	多协议支持	√	×	×	JavaChassis针对消费与生产端支持多种通信协议，如下： <ul style="list-style-type: none"> <li>生产端：JAX-RS、SpringMVC、透明RPC。</li> <li>消费端：透明RPC、RestTemplate、InvokerUtils。</li> </ul>
	拓展	<ul style="list-style-type: none"> <li>支持用户自定义处理链处理流量。</li> <li>支持用户扩展流量治理。</li> </ul>	<ul style="list-style-type: none"> <li>支持Spring Cloud原生扩展。</li> <li>支持用户扩展流量治理。</li> </ul>	基于插件开发模式新增能力。	-

## 配额说明

配额是指您引擎实例中可创建的资源数量限制。如需使用更大配额，请[提交工单](#)申请扩大配额。

- Nacos引擎实例中可创建的资源数量限制如[表8-1](#)所示。

表 8-1 Nacos 引擎资源配额限制

资源	最大配额	是否支持修改配额	注意事项
Nacos单实例命名空间数量	50个	否	-

资源	最大配额	是否支持修改配额	注意事项
Nacos单个配置文件大小上限	100KB	否	-
Nacos单个命名空间配置总计大小	10MB	否	-
带宽（网络流出+流入之和）	2Mbit/s	否	-
Nacos单实例配置文件数量上限	10,000个	否	-

- ServiceComb引擎实例中可创建的资源数量限制如表8-2所示。

表 8-2 ServiceComb 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

- 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。

# 9 权限管理

如果您需要对CSE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，您可以在公有云账号中给员工创建IAM用户，并使用策略来控制员工对云资源的访问范围。例如，您希望您的员工中负责软件开发的人员拥有CSE的使用权限，但是不希望其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CSE，但是不允许删除的权限策略，控制其对CSE资源的使用范围。

如果公有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CSE服务的其它功能。

IAM是公有云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》。

## CSE 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。

CSE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CSE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。由于华为云各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CSE的所有系统权限。

**表 9-1** CSE 系统权限说明

系统角色/策略名称	描述	类别	依赖关系
CSE FullAccess	微服务引擎服务管理员权限。	系统策略	无
CSE ReadOnlyAccess	微服务引擎服务查看权限。	系统策略	无

如果表9-1所列权限不满足实际需求，可以在这个基础上[创建微服务引擎自定义策略](#)。  
更多CSE相关功能需要依赖的服务权限，请参见表9-2所示。

**表 9-2** CSE 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
删除、创建引擎	虚拟私有云 VPC	如果需要创建与删除引擎，需要设置 VPC ReadOnlyAccess权限。
仪表盘	应用运维管理 AOM	如果需要查看仪表盘等图表数据，需要设置AOM ReadOnlyAccess权限。
标签管理	标签管理服务 TMS	如果需要使用TMS为ServiceComb引擎、Nacos引擎添加标签，可以方便用户识别和管理拥有的ServiceComb引擎、Nacos引擎资源，需要设置TMS ReadOnlyAccess权限。
导入IAM用户	统一身份认证服务 IAM	如果需要导入IAM用户，需要设置IAM ReadOnlyAccess权限。

表9-3列出了微服务引擎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 9-3** 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	√



操作	CSE ReadOnlyAccess	CSE FullAccess
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	√

使用自定义细粒度策略，请使用管理员用户进入统一身份认证（IAM）服务，按需选择微服务引擎的细粒度权限进行授权操作。微服务引擎细粒度权限依赖说明请参见表 9-4。

表 9-4 微服务引擎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	<ul style="list-style-type: none"> <li>查看ServiceComb引擎列表。</li> <li>查看Nacos引擎列表。</li> </ul>
cse:engine:get	查看引擎信息	<ul style="list-style-type: none"> <li>cse:engine:list</li> </ul>	<ul style="list-style-type: none"> <li>查看ServiceComb引擎详情。</li> <li>查看Nacos引擎详情。</li> </ul>

权限名称	权限描述	权限依赖	应用场景
cse:engine:modify	修改引擎	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	<ul style="list-style-type: none"> <li>ServiceComb引擎开启/关闭公网访问操作。</li> <li>ServiceComb引擎开启/管理安全认证操作。</li> <li>ServiceComb引擎失败任务重试操作。</li> <li>Nacos引擎开启/关闭安全认证。</li> <li>管理Nacos权限用户与角色控制。</li> <li>管理命名空间与企业项目关系。</li> </ul>
cse:engine:upgrade	升级引擎	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	<ul style="list-style-type: none"> <li>升级ServiceComb引擎</li> <li>升级Nacos引擎。</li> </ul> 升级的操作包括：版本升级操作、规格变更操作。
cse:engine:delete	删除引擎	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>vpc:ports:get</li> <li>vpc:ports:delete</li> </ul>	<ul style="list-style-type: none"> <li>删除ServiceComb引擎。</li> <li>删除Nacos引擎。</li> </ul>
cse:engine:create	创建引擎	<ul style="list-style-type: none"> <li>cse:engine:get</li> <li>cse:engine:list</li> <li>ecs:cloudServerFlavors:get</li> <li>vpc:vpcs:get</li> <li>vpc:vpcs:list</li> <li>vpc:subnets:get</li> <li>vpc:ports:get</li> <li>vpc:ports:create</li> </ul>	<ul style="list-style-type: none"> <li>创建ServiceComb引擎。</li> <li>ServiceComb引擎备份/恢复任务创建操作。</li> <li>创建Nacos引擎。</li> </ul>
cse:config:modify	ServiceComb引擎服务配置管理修改	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> <li>cse:config:get</li> </ul>	ServiceComb引擎全局配置功能与治理功能涉及的配置修改。
cse:config:get	ServiceComb引擎服务配置管理查看	<ul style="list-style-type: none"> <li>cse:engine:list</li> <li>cse:engine:get</li> </ul>	ServiceComb引擎查看服务的配置。

权限名称	权限描述	权限依赖	应用场景
cse:governance:modify	ServiceComb引擎服务治理中心修改	<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ServiceComb引擎服务治理。
cse:governance:get	ServiceComb引擎服务治理中心查看	<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ServiceComb引擎服务治理功能。
cse:registry:modify	ServiceComb引擎服务注册管理修改	<ul style="list-style-type: none"> <li>● cse:engine:list</li> <li>● cse:engine:get</li> <li>● cse:registry:get</li> </ul>	ServiceComb引擎服务修改。
cse:registry:get	ServiceComb引擎服务注册管理查看	<ul style="list-style-type: none"> <li>● cse:engine:list</li> <li>● cse:engine:get</li> </ul>	ServiceComb引擎服务目录查看。
cse:namespace:read	Nacos引擎命名空间资源查看	cse:engine:get	Nacos服务列表、配置列表资源查看。
cse:namespace:write	Nacos引擎命名空间资源修改	<ul style="list-style-type: none"> <li>● cse:engine:get</li> <li>● cse:namespace:read</li> </ul>	Nacos服务列表、配置列表资源修改。

## 相关链接

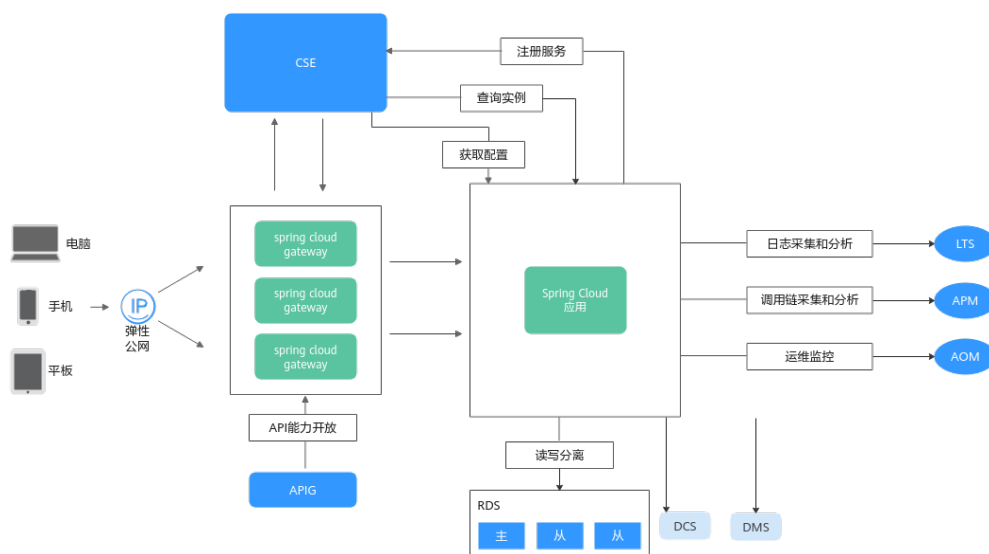
- [IAM产品介绍](#)。
- [创建用户并授权使用微服务引擎](#)。

# 10 与其他服务的关系

云原生架构模式下，完成业务功能需要很多服务相互配合：

- CSE一般和数据库、缓存和消息中间件同时使用，完成业务功能的开发。
- AOM、APM、LTS等工具，则为业务提供运维能力，帮助检测业务故障、分析故障原因。

以Spring Cloud为例，典型的云原生架构和技术选型如下：



云原生架构和DevOps通常是密不可分的，配合使用ServiceStage，可以实现云原生环境的管理、部署流水线，简化微服务应用部署到容器CCE的流程。