

微服务引擎

用户指南

文档版本 01
发布日期 2026-01-06



版权所有 © 华为云计算技术有限公司 2026。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 产品介绍	1
1.1 什么是微服务引擎	1
1.2 应用场景	3
1.3 产品功能	4
1.4 基本概念	6
1.5 ServiceComb 引擎版本支持机制	8
1.6 产品规格	9
1.7 约束与限制	10
1.8 权限管理	15
1.9 与其他服务的关系	19
2 快速入门	21
2.1 Spring Cloud 应用接入 ServiceComb 引擎	21
2.2 Spring Cloud 应用接入 Nacos 引擎	24
3 使用 CSE 前必读	28
4 创建用户并授权使用 CSE	29
5 ServiceComb 引擎	32
5.1 ServiceComb 引擎概述	32
5.2 创建 ServiceComb 引擎	33
5.3 管理 ServiceComb 引擎	36
5.3.1 查看 ServiceComb 引擎信息	36
5.3.2 管理 ServiceComb 引擎标签	39
5.3.3 管理 ServiceComb 引擎公网访问	40
5.3.4 管理 ServiceComb 引擎安全认证	41
5.3.5 设置 ServiceComb 引擎备份与恢复	43
5.3.6 升级 ServiceComb 引擎版本	45
5.3.7 变更 ServiceComb 引擎规格	45
5.3.8 查看 ServiceComb 引擎操作日志	46
5.3.9 删除 ServiceComb 引擎	46
5.4 使用微服务仪表盘查看微服务运行指标	47
5.5 管理微服务	47
5.5.1 查看应用	47
5.5.2 微服务管理	48

5.5.3 实例管理.....	55
5.6 业务场景治理（适用于 ServiceComb 引擎为 2.x 版本）.....	57
5.6.1 业务场景治理概述.....	57
5.6.2 创建业务场景.....	57
5.6.3 创建治理策略.....	59
5.7 微服务治理（适用 ServiceComb 引擎为 1.x 和 2.4.0+）.....	62
5.7.1 微服务治理概述.....	62
5.7.2 为微服务设置负载均衡治理策略.....	63
5.7.3 为微服务设置限流治理策略.....	64
5.7.4 为微服务设置降级治理策略.....	65
5.7.5 为微服务设置容错治理策略.....	66
5.7.6 为微服务设置熔断治理策略.....	67
5.7.7 为微服务设置错误注入治理策略.....	68
5.7.8 为微服务设置黑白名单治理策略.....	69
5.7.9 为微服务设置公钥认证.....	70
5.8 配置管理（适用于 ServiceComb 引擎 2.x 版本）.....	70
5.8.1 创建 2.x 版本 ServiceComb 引擎配置.....	70
5.8.2 管理 2.x 版本 ServiceComb 引擎配置.....	73
5.9 配置管理（适用于 ServiceComb 引擎 1.x 版本）.....	78
5.9.1 创建 1.x 版本 ServiceComb 引擎配置.....	78
5.9.2 管理 1.x 版本 ServiceComb 引擎配置.....	78
5.10 系统管理.....	79
5.10.1 系统管理概述.....	80
5.10.2 账号管理.....	80
5.10.3 角色管理.....	84
6 Nacos 引擎.....	88
6.1 Nacos 引擎概述.....	88
6.2 创建 Nacos 引擎.....	89
6.3 管理 Nacos 引擎.....	91
6.3.1 查看 Nacos 引擎详细信息.....	91
6.3.2 管理 Nacos 引擎标签.....	92
6.3.3 管理 Nacos 引擎白名单.....	93
6.3.4 扩容 Nacos 引擎.....	94
6.3.5 升级 Nacos 引擎版本.....	94
6.3.6 删除 Nacos 引擎.....	95
6.4 管理命名空间.....	95
6.5 权限控制.....	97
6.5.1 权限控制概述.....	97
6.5.2 开启/关闭安全认证.....	97
6.5.3 账号管理.....	98
6.5.4 角色管理.....	99
6.5.5 控制台资源管理.....	100

6.6 管理 Nacos 引擎服务.....	100
6.7 管理 Nacos 引擎配置.....	102
6.7.1 Nacos 引擎配置概述.....	102
6.7.2 创建 Nacos 引擎配置.....	103
6.7.3 管理 Nacos 引擎配置操作.....	104
6.7.4 管理 Nacos 引擎配置的灰度发布.....	107
6.7.5 管理 Nacos 引擎历史版本配置.....	108
6.7.6 使用 Nacos 引擎的监听查询功能.....	109
6.8 查看 Nacos 引擎运行监控.....	109
7 云审计服务支持的关键动作.....	111
7.1 云审计服务支持的 CSE 操作列表.....	111
7.2 在 CTS 事件列表查看云审计事件.....	113
8 常见问题.....	118
8.1 使用 CSE 注意事项.....	118
8.1.1 查看 ServiceComb 引擎、Nacos 引擎信息时为什么看不到 VPC、ELB 等云服务信息?	118
8.1.2 创建委托失败怎么解决?	118
8.2 Nacos 引擎.....	119
8.2.1 服务启动时注册了端口为 8080 和 9090 的实例，在服务列表中 9090 端口实例丢失，导致请求 grpc 的时候报错.....	119
8.3 ServiceComb 引擎.....	119
8.3.1 证书加载错误.....	119
8.3.2 无效头名称.....	120
8.3.3 mesher 性能损耗是多少?	120
8.3.4 连接服务中心提示“Version validate failed”	121
8.3.5 连接服务中心提示“Not enough quota”	121
8.3.6 服务名重复校验范围是什么?	121
8.3.7 为什么一定要定义服务契约?	121
8.3.8 如何解决微服务应用开发过程中微服务开发框架同 netty 版本不匹配的问题?	122
8.3.9 创建引擎提示“Duplicate cluster name”	122
8.3.10 创建引擎过程中处理接入地址步骤失败，提示 the subnet could not be found.....	123
8.3.11 SpringCloud 应用连接 ServiceComb 引擎 2.x 版本配置中心失败.....	123
8.3.12 在全局配置中配置相关的配置项修改后，服务获取的配置内容未修改成功.....	124
8.3.13 获取配置失败.....	124

1 产品介绍

1.1 什么是微服务引擎

微服务引擎（Cloud Service Engine，简称CSE）是微服务应用的云中间件，为用户提供注册发现、配置管理、服务治理等高性能和高韧性的企业级云服务；无缝兼容SpringCloud、ServiceComb、Dubbo等开源生态；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与高可用运维。

注册配置中心 Nacos 引擎

CSE注册配置中心Nacos兼容开源Nacos、Eureka客户端，具备注册发现、动态配置管理、访问权限控制、可观测等能力。可打造高可用、易管理的微服务中间件。

ServiceComb 引擎

CSE ServiceComb引擎基于Apache ServiceComb开源生态，提供一站式的微服务平台。支持使用Java-Chassis SDK、SpringCloudHuawei SDK或无侵入的Sermant Java Agent（支持标准SpringCloud和Dubbo框架）接入。接入后，用户可以轻松使用服务契约、服务治理、灰度发布、业务场景治理、服务监控、配置管理、访问控制等众多功能，实践api first开发，构建高安全、高性能、高稳定的微服务应用。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>

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1-1](#)。

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

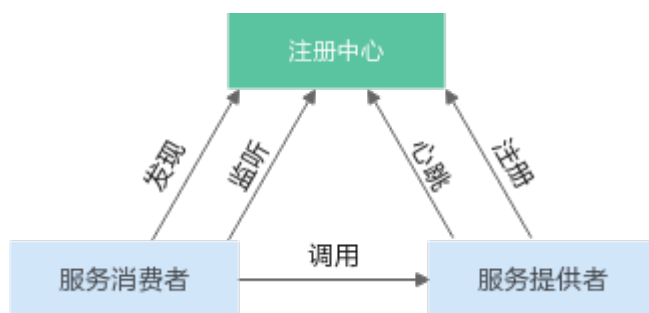
功能	特性		2.x	1.x	备注
引擎管理	安全性	支持安全认证	√	√	-
	可靠性	3AZ高可靠	√	√	-
微服务管理	基础能力	注册发现	√	√	-
		多框架接入	√	√	支持Spring Cloud、ServiceComb Java Chassis。
		无实例版本自动清理	√	x	2.3.7及以后版本，支持保留最近3个微服务版本，并自动清理无实例版本。
	性能	实例变化毫秒级推送	√	√	-
配置管理	基础能力	管理配置	√	√	-
		配置格式多样化	√	仅支持文本	2.x新增支持配置格式有：YAML、JSON、TEXT、Properties、INI、XML。
		导入导出	√	√	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	√	√	-

功能	特性	2.x	1.x	备注	
	治理策略-重试	客户端通过重试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	√	√	-
	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	√	√	-

1.2 应用场景

微服务注册发现

微服务架构首先要解决的是微服务间的通信问题，相比传统的通信总线、LB方案，注册发现机制在客户端实现负载均衡，在通信效率和弹性上更具优势。CSE为SpringCloud、Dubbo、ServiceComb等开发框架提供了高可用、稳定、免运维的服务注册中心。



微服务注册发现机制主要通过服务注册中心来实现。

- 服务注册：**微服务实例在启动时，会向服务注册中心发送注册请求，将自身的元数据（如服务名称、IP地址、端口号、版本号等）注册到注册中心。注册中心会将这些信息存储在一个内部的数据结构中，以便后续查询。服务提供者会以固定时间间隔，向注册中心发送包含服务运行状态、负载、资源消耗等健康信息的心跳消息，以此告知注册中心自身处于活动状态。注册中心接收并记录心跳消息，依据消息到达情况和内容判断服务可用性，若超过一定时间未收到心跳，会将对应服务实例状态标记为不健康，若继续超时而仍未收到，则从可用服务列表中剔除该实例。当服务实例恢复发送心跳，可重新注册。
- 服务发现：**服务消费者直接与注册中心交互，自主获取服务提供者的实例信息，并在本地实现负载均衡。在微服务架构中，服务提供者的实例可能会因为各种原

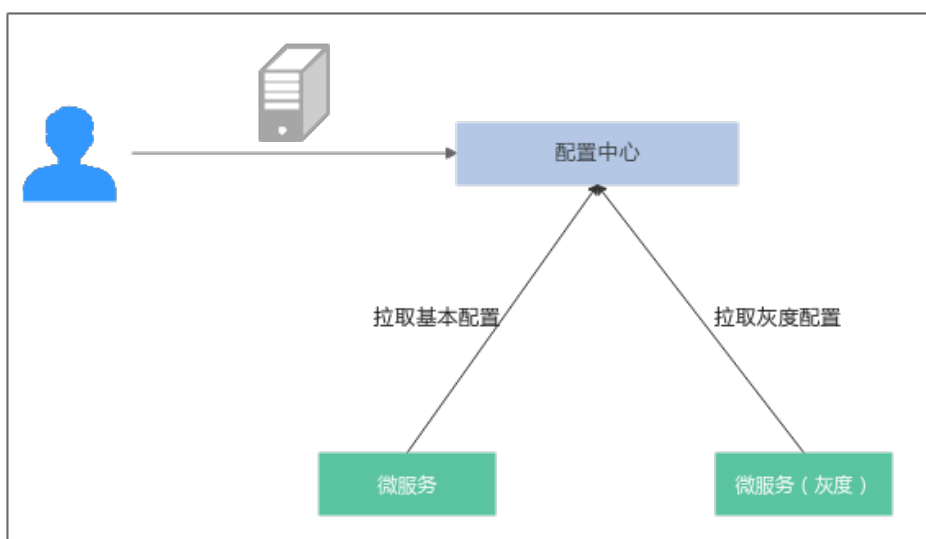
因（如扩容、缩容、故障等）发生变化。服务消费者通过监听注册中心，可以实时感知这些变化，及时更新本地的服务实例列表，确保后续的服务调用能够准确地找到可用的服务提供者。

动态配置管理

配置中心提供集中化的配置管理，实现不同环境、不同集群和不同实例的差异化配置。运行时动态变更配置，相比传统配置文件方式，更加高效和规范。

配置中心按使用方式，分为以下几种：

- 作为部署环境配置，与部署交付服务集成，如常见的SpringCloud服务的application.yaml，可以承载应用数据源，访问私钥等配置信息。
- 作为运维参数配置，与运维服务结合，如动态调整日志级别，连接池个数。
- 作为业务参数配置，如动态修改商品价格折扣，动态公告，动态中奖率等。



微服务在启动过程中，会向配置中心发送请求，获取自身的配置信息。配置中心根据微服务的请求，从存储中读取相应的配置数据，并返回给微服务。微服务在接收到配置信息后，会将其解析并加载到内存中，供应用程序使用。

在配置中心，管理员针对灰度发布需求，定义专门的灰度配置项，这些配置项与正常生产配置区分开，可关联特定标识（如灰度版本号、用户分组标识）。

1. 微服务启动时，从配置中心拉取基础配置信息，同时携带自身标识（如服务名、版本号等）及灰度相关标识（若已确定所属灰度分组）。
2. 配置中心接收到微服务请求后，依据微服务携带的标识，判断其是否属于灰度范围。若属于灰度范围，配置中心从灰度配置存储区查找并返回对应的灰度配置；若不属于，则返回正常生产配置。
3. 微服务在运行过程中，可根据配置中心的通知机制（如长连接推送、定时轮询），动态拉取灰度配置的更新内容。当灰度配置有变更时，配置中心及时通知相关微服务，微服务重新拉取并应用新的灰度配置。

1.3 产品功能

微服务引擎（Cloud Service Engine, CSE），是用于微服务应用的云中间件，支持贡献到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。

注册配置中心 Nacos 引擎

- 管理Nacos引擎。

通过微服务引擎控制台，可以查看Nacos引擎信息、扩容Nacos引擎、升级Nacos引擎、管理Nacos引擎标签和删除Nacos引擎等。详细信息请参考[管理Nacos引擎](#)。

- 使用Nacos引擎。

- 命名空间可以实现不同环境的配置的区别隔离，例如开发测试环境和生产环境的资源（如配置、服务）隔离等。不同的命名空间下，可以存在相同的Group或Data ID的配置。详细信息请参考[管理命名空间](#)。
- 微服务引擎控制台提供了对注册在Nacos上的服务进行管理的功能。详细信息请参考[管理Nacos引擎服务](#)。
- 微服务引擎控制台提供对注册在Nacos上的服务的配置进行管理。详细信息请参考[管理Nacos引擎配置](#)。

ServiceComb 引擎

- 管理ServiceComb引擎。

通过微服务引擎控制台，可以查看ServiceComb引擎信息、设置ServiceComb引擎备份与恢复、管理ServiceComb引擎公网访问、升级ServiceComb引擎版本、变更ServiceComb引擎规格、管理ServiceComb引擎安全认证、管理ServiceComb引擎标签和删除ServiceComb引擎等。详细信息请参考[管理ServiceComb引擎](#)。

- 使用ServiceComb引擎。

- ServiceComb引擎的服务中心提供了服务注册的功能。服务注册是指微服务启动的时候，将基本信息，比如所属应用、微服务名称、微服务版本、监听的地址信息等注册到服务中心。
- 配置中心用来管理微服务应用的配置。微服务连接配置中心，能够从配置中心获取配置信息及其变化。配置中心还是其他微服务管控功能的核心部件，比如服务治理规则的下发，也是通过配置中心实现的。
- 对不同的微服务开发框架，比如Spring Cloud、Java Chassis等设计了统一的基于动态配置的流量特征治理。通过在开发框架中引入相关治理组件，即可使用ServiceComb引擎的微服务治理功能。
- 为保障新特性平稳上线，可以先选择少部分用户试用，待新特性成熟以后，再让所有用户使用。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。
- 仪表盘提供一些基础的微服务运行监控能力。微服务通过SDK上报运行状态数据，上报的数据内容包括请求统计数据，比如请求数、时延、错误率等，还包括和治理有关的一些状态，比如熔断状态等。
- 同一个ServiceComb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ServiceComb引擎访问和操作权限。

1.4 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区分离，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>
配置集ID	<p>Nacos中的某个配置集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。</p>
分组	<p>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。</p>

概念	说明
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>说明</p> <ul style="list-style-type: none"> Java Chassis默认使用契约功能。 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： <pre><dependency> <groupId>com.huaweicloud</groupId> <artifactId>spring-cloud-starter-huawei-swagger</artifactId> </dependency></pre>
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。

概念	说明
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。

1.5 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。

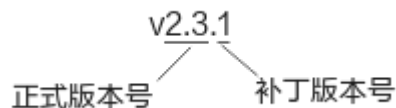
版本号说明

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。
- 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。

1.6 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 1-2 Nacos 引擎实例规格

最大微服务实例数	容量单元	客户端连接数	配置文件数
500实例	10个	1,000个	1,000个
1,000实例	20个	2,000个	2,000个
2,000实例	40个	4,000个	4,000个

说明

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1-3来选择ServiceComb引擎实例规格。

如表1-3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 1-3 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
500实例	3,000
2,000实例	12,000

1.7 约束与限制

Nacos 引擎与微服务框架版本关系

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
	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

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	1.11.11-2021.0.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2021.0.9 ● 适配的Spring Boot版本为2.7.18 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	2.8.25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 。

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

一级特性	二级特性	servicecomb-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	×	√	√	-
	优雅停机	√	√	×	-
微服务治理	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"> 生产端: JAX-RS、SpringMVC、透明RPC。 消费端: 透明RPC、RestTemplate、InvokerUtils。
	拓展	<ul style="list-style-type: none"> 支持用户自定义处理链处理流量。 支持用户扩展流量治理。 	<ul style="list-style-type: none"> 支持Spring Cloud原生扩展。 支持用户扩展流量治理。 	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, 请申请扩大配额。

- Nacos引擎实例中可创建的资源数量限制如[表1-4](#)所示。

表 1-4 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	-
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	-

- ServiceComb引擎实例中可创建的资源数量限制如表1-5所示。

表 1-5 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。

1.8 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《统一身份认证服务用户指南》的“产品简介”章节。

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1-6所示，包括了CSE的所有系统权限。

表 1-6 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1-6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1-7所示。

表 1-7 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1-8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 1-8 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 1-9。

表 1-9 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"> 查看ServiceComb引擎列表。 查看Nacos引擎列表。
cse:engine:get	查看引擎信息	<ul style="list-style-type: none"> cse:engine:list 	<ul style="list-style-type: none"> 查看ServiceComb引擎详情。 查看Nacos引擎详情。

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"> cse:engine:list cse:engine:get 	<ul style="list-style-type: none"> ServiceComb引擎开启/关闭公网访问操作。 ServiceComb引擎开启/管理安全认证操作。 ServiceComb引擎失败任务重试操作。 Nacos引擎开启/关闭安全认证。 管理Nacos权限用户与角色控制。 管理命名空间与企业项目关系。
cse:engine:upgrade	升级引擎	<ul style="list-style-type: none"> cse:engine:list cse:engine:get 	<ul style="list-style-type: none"> 升级ServiceComb引擎 升级Nacos引擎。 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"> cse:engine:list cse:engine:get vpc:ports:get vpc:ports:delete 	<ul style="list-style-type: none"> 删除ServiceComb引擎。 删除Nacos引擎。
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 	<ul style="list-style-type: none"> 创建ServiceComb引擎。 ServiceComb引擎备份/恢复任务创建操作。 创建Nacos引擎。
cse:config:modify	ServiceComb引擎服务配置管理修改	<ul style="list-style-type: none"> cse:engine:list cse:engine:get cse:config:get 	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"> cse:engine:list cse:engine:get 	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:dashboard: get	ServiceComb引擎服务仪表盘查看	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get 	ServiceComb引擎查看服务的仪表盘信息。
cse:dashboard: modify	ServiceComb引擎服务仪表盘修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:dashboard:get 	ServiceComb引擎修改服务的仪表盘配置。
cse:governance: modify	ServiceComb引擎服务治理中心修改	<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"> ● cse:engine:list ● cse:engine:get ● cse:config:get ● cse:registry:get 	查看ServiceComb引擎服务治理功能。
cse:registry:mo dify	ServiceComb引擎服务注册管理修改	<ul style="list-style-type: none"> ● cse:engine:list ● cse:engine:get ● cse:registry:get 	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"> ● cse:engine:list ● cse:engine:get 	ServiceComb引擎服务目录查看。
cse:namespace: read	Nacos引擎命名空间资源查看	<ul style="list-style-type: none"> ● cse:engine:list ● 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"> ● cse:engine:get ● cse:namespace:read 	Nacos服务列表、配置列表资源修改。

相关链接

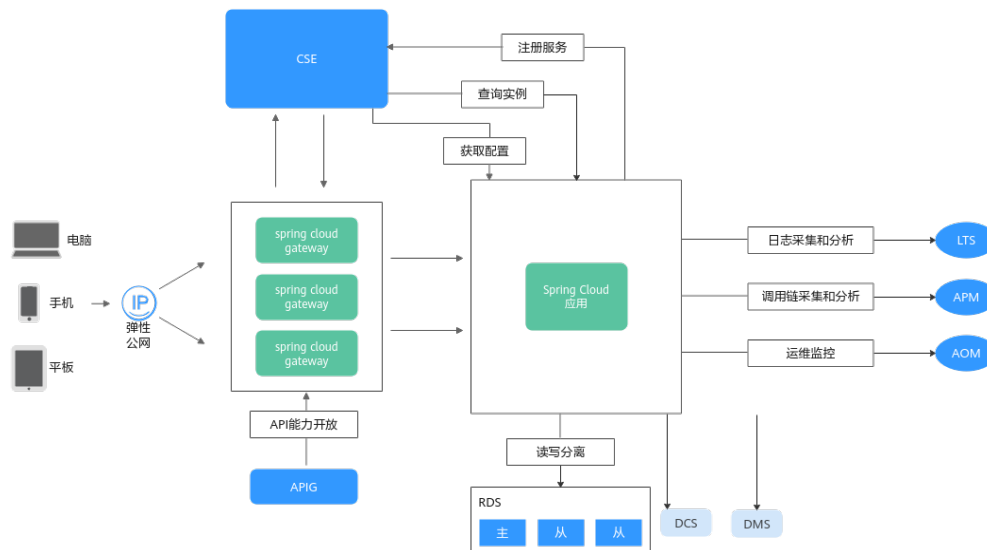
- 《统一身份认证服务用户指南》的“产品简介”章节。
- [创建用户并授权使用CSE](#)。

1.9 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。

2 快速入门

2.1 Spring Cloud 应用接入 ServiceComb 引擎

本章节通过一个demo演示如何将一个Spring Cloud应用通过Spring Cloud Huawei SDK接入ServiceComb引擎，指导您快速上手使用ServiceComb引擎。

1. 准备工作

为用户添加操作权限、创建VPC和子网、获取demo包及本地编译构建打包环境准备工作。

2. 创建ServiceComb引擎

在创建引擎时，您可以根据实际业务需要，选择合适的引擎规格、可用区和网络等。

3. Spring Cloud接入ServiceComb引擎

本指导将演示使用一个provider服务和一个consumer服务接入ServiceComb引擎。

准备工作

1. 已获取控制台的登录账号与密码。
2. 为用户添加操作权限。
用户在创建依赖资源和ServiceComb引擎前，需要具备相应的操作权限。添加用户权限的操作，请参考[创建用户并授权使用CSE](#)。
3. 创建VPC和子网。
ServiceComb引擎运行于虚拟私有云（VPC）中，并需要绑定具体的子网。创建ServiceComb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的方法，请参考《[虚拟私有云用户指南](#)》中“[创建虚拟私有云和子网](#)”章节。如果已有可用的VPC和子网，不需要再次创建。
4. 本地编译构建打包机器环境已安装了Java JDK、Maven，并且能够访问Maven中央库。
5. 下载github的[demo源码](#)到本地并解压。


创建 ServiceComb 引擎

步骤1 登录微服务引擎控制台。

步骤2 在左侧导航栏选择“ServiceComb引擎专享版”，单击“购买ServiceComb引擎”。

步骤3 参考下表设置参数，参数前面带*号的是必须设置的参数。创建ServiceComb引擎所需参数的详细介绍请参考[创建ServiceComb引擎](#)。

表 2-1 创建 ServiceComb 引擎参数

参数	说明
*计费模式	选择计费方式，目前支持“按需计费”。按需计费是一种后付费模式，即先使用再付费，按照ServiceComb引擎实际使用时长计费
*企业项目	选择ServiceComb引擎所在的企业项目。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。此处默认选择“default”。
*选择实例数	选择引擎规格，此处选择微服务实例数为100。
*引擎类型	引擎类型为集群，其为集群模式部署，主机级容灾。
*ServiceComb引擎名称	输入ServiceComb引擎的名称，名称以字母开头，由字母、数字和-组成，且不能以-结尾，长度为3~24个字符。例如：cse-test。
*可用区	可选择1个或3个可用区。此处选择1个可用区，可提供主机级别容灾能力。
*网络	选择已创建的虚拟私有云及子网，可在下拉框中搜索和选择合适的虚拟私有云和子网。
描述	单击  ，输入引擎描述信息，例如：体验快速创建ServiceComb引擎。
*安全认证	开启了“安全认证”的ServiceComb引擎专享版，通过微服务引擎控制台提供了基于RBAC（Role-Based Access Control，基于角色的访问控制）的系统管理功能。此处选择“关闭安全认证”，关闭安全认证功能后，可以在实例创建完成后再设置开启安全认证。

步骤4 单击“立即购买”，进入引擎信息确认界面。

步骤5 单击“提交”，等待引擎创建完毕。

说明

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

----结束

Spring Cloud 接入 ServiceComb 引擎

步骤1 登录微服务引擎控制台。

步骤2 在左侧导航栏选择“ServiceComb引擎专享版”。

步骤3 单击[创建ServiceComb引擎](#)中创建的ServiceComb引擎。

步骤4 获取ServiceComb引擎的注册中心地址和配置中心地址。

在“服务发现 & 配置”区域，查看获取引擎服务注册发现地址和配置中心地址。

步骤5 修改demo中的注册中心地址和配置中心地址。

1. 在下载到本地的demo源码目录下，分别找到“\basic\consumer\src\main\resources\bootstrap.yml”和“\basic\provider\src\main\resources\bootstrap.yml”文件。
2. 添加ServiceComb引擎的注册中心地址和配置中心地址到项目配置文件中（以“\basic\consumer\src\main\resources\bootstrap.yml”为例）。

```
spring:
  application:
    name: basic-consumer
  cloud:
    servicecomb:
      discovery:
        enabled: true
        watch: false
        # 注册中心地址
        address: https://192.168.0.210:30100,https://192.168.0.246:30100
        appName: basic-application
        serviceName: ${spring.application.name}
        version: 0.0.1
        healthCheckInterval: 30
      config:
        # 配置中心地址
        serverType: kie
        serverAddr: https://192.168.0.210:30110,https://192.168.0.246:30110
```

📖 说明

使用ServiceStage部署的场景，服务注册中心地址和配置中心地址在部署过程中会自动注入，无需额外手工添加。

步骤6 打包demo源码成jar包。

1. 在demo源码根目录下，打开cmd命令，执行**mvn clean package**命令，对项目进行打包编译。
2. 编译成功后，生成如[表2-2](#)所示的两个Jar包。

表 2-2 软件包列表

软件包所在目录	软件包名称	说明
basic\consumer\target	basic-consumer-1.0-SNAPSHOT.jar	服务消费者
basic\provider\target	basic-provider-1.0-SNAPSHOT.jar	服务生产者

步骤7 部署应用。

使用ServiceStage部署微服务provider和consumer。

1. 将**步骤6**生成的JAR包上传至OBS。
2. 参考《云容器引擎用户指南》的“快速创建Kubernetes集群”章节创建CCE集群，并同ServiceComb引擎实例属于同一VPC。
3. 参考《应用管理与运维平台用户指南》中“创建环境”章节在引擎实例所在VPC下创建ServiceStage环境，并对ServiceComb引擎和CCE资源进行纳管。
4. 参考《应用管理与运维平台用户指南》中“创建并部署组件”章节部署provider和consumer微服务。

步骤8 确认部署结果。

1. **可选:** 在微服务引擎控制台页面，在左侧导航栏选择“ServiceComb引擎专享版”，单击**创建ServiceComb引擎**中创建的ServiceComb引擎。
2. 选择“微服务目录 > 微服务列表”，查看微服务basic-consumer和basic-provider的实例数量。
 - 若实例数量值不为0，则表示已经成功接入ServiceComb引擎。
 - 若实例数量为0，或者找不到basic-consumer和basic-provider服务名，则表示微服务应用接入ServiceComb引擎失败。

----结束

2.2 Spring Cloud 应用接入 Nacos 引擎

本章节通过一个demo进行全流程的微服务应用接入Nacos引擎操作演示，帮助您快速了解如何接入Nacos引擎。

1. 准备工作

为用户添加操作权限、创建VPC和子网、获取demo包及准备好本地编译构建打包环境准备工作。

2. 创建注册配置中心

在创建引擎时，您可以根据实际业务需要，选择合适的引擎规格、可用区和网络等。

3. Spring Cloud应用接入Nacos引擎

本指导将使用一个provider服务和一个consumer服务接入Nacos引擎。

准备工作

1. 已获取控制台的登录账号与密码。
2. 为用户添加操作权限。
用户在创建依赖资源和Nacos引擎前，需要具备相应的操作权限。添加用户权限的操作，请参考**创建用户并授权使用CSE**。

说明

创建注册配置中心需要保证用户具有CSE FullAccess、DNS FullAccess权限。

3. 创建VPC和子网。
Nacos引擎运行于虚拟私有云（VPC）中，并需要绑定具体的子网。创建Nacos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的方法，请参

考《虚拟私有云用户指南》中“创建虚拟私有云和子网”章节。如果已有可用的VPC和子网，不需要再次创建。

4. 本地编译构建打包机器环境已安装了Java JDK、Maven，并且能够访问Maven中央库。
5. 下载github的[demo源码](#)到本地并解压。

创建注册配置中心

步骤1 登录微服务引擎控制台。

步骤2 在左侧导航栏选择“注册配置中心”，单击“购买注册配置中心”。

步骤3 参考下表设置参数，参数前面带*号的是必须设置的参数。创建注册配置中心所需参数的详细介绍请参考[创建Nacos引擎](#)。

表 2-3 创建注册配置中心参数

参数	说明
*计费模式	选择计费方式，目前支持“按需计费”。按需计费是一种后付费模式，即先使用再付费，按照注册配置中心实际使用时长计费。
*企业项目	选择注册配置中心Nacos所在的企业项目。企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。此处默认选择“default”。
*引擎名称	输入Nacos引擎的名称，名称以字母开头，由字母、数字和-组成，且不能以-结尾，长度为3~24个字符。如输入nacos-test。
*注册配置中心类型	支持的注册配置中心类型为“Nacos”。 说明 注册配置中心集群的节点会尽可能均分到不同的可用区中，单节点故障不影响对外业务功能。注册配置中心不支持AZ级故障的容灾，可提供主机级别容灾能力。
*选择实例数	选择需要购买的容量规格。此处选择实例数为500，其容量单元为10。
版本	只能创建最新版本。
*网络	选择已创建的虚拟私有云及子网，可在下拉框中搜索和选择合适的虚拟私有云和子网。 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。

步骤4 单击“立即购买”，注册配置中心开始创建，当“运行状态”为“可用”时，注册配置中心创建完成。

----结束

Spring Cloud 应用接入 Nacos 引擎

步骤1 登录微服务引擎控制台。

步骤2 获取Nacos引擎注册发现地址。

1. 在左侧导航栏选择“注册配置中心”，单击创建的Nacos引擎实例。
2. 在“基础信息”页面的“连接信息”区域，获取注册发现地址。

步骤3 修改demo中的配置中心地址和服务注册中心地址和微服务名。

1. 在“bootstrap.properties”中配置Nacos配置中心。

在下载到本地的demo源码目录中找到“nacos-examples-master\nacos-spring-cloud-example\nacos-spring-cloud-discovery-example\nacos-spring-cloud-consumer-example\src\main\resources”添加“bootstrap.properties”文件，配置Nacos配置中心：

```
spring.cloud.nacos.config.server-addr=XXX.nacos.cse.com:8848 //Nacos的配置中心地址
spring.cloud.nacos.config.prefix=example //配置文件名前缀
spring.cloud.nacos.config.file-extension=properties //配置文件名后缀
spring.cloud.nacos.config.group=XXX //配置文件所属分组，不填默认DEFAULT_GROUP
spring.cloud.nacos.config.namespace=XXX //配置文件所属命名空间ID，不填默认public
```

更多配置详情，请参考[Nacos配置参考](#)。

2. 在“application.properties”中配置Nacos的服务注册发现地址和微服务名。

- 在下载到本地的demo源码目录中找到“nacos-examples-master\nacos-spring-cloud-example\nacos-spring-cloud-discovery-example\nacos-spring-cloud-consumer-example\src\main\resources\application.properties”文件，配置consumer服务：

```
server.port=8080
spring.application.name=service-consumer //微服务名
spring.cloud.nacos.discovery.server-addr= XXX.nacos.cse.com:8848 //Nacos的服务注册发现地址
spring.cloud.nacos.discovery.group=XXX //微服务所属分组，不填则默认DEFAULT_GROUP
spring.cloud.nacos.discovery.namespace=XXX //微服务所属命名空间ID，不填则默认public
spring.cloud.nacos.discovery.cluster-name=XXX //微服务所属集群名称，不填则默认DEFAULT
```

- 在下载到本地的demo源码目录中找到“nacos-examples-master\nacos-spring-cloud-example\nacos-spring-cloud-discovery-example\nacos-spring-cloud-provider-example\src\main\resources\application.properties”文件，配置provider服务：

```
server.port=8070
spring.application.name=service-provider //微服务名
spring.cloud.nacos.discovery.server-addr= XXX.nacos.cse.com:8848 //Nacos的服务注册发现地址
spring.cloud.nacos.discovery.group=XXX //微服务所属分组，不填则默认DEFAULT_GROUP
spring.cloud.nacos.discovery.namespace=XXX //微服务所属命名空间ID，不填则默认public
spring.cloud.nacos.discovery.cluster-name=XXX //微服务所属集群名称，不填则默认DEFAULT
```

更多配置详情，请参考[Nacos注册发现](#)。

步骤4 打包demo源码成jar包。

1. 在demo源码根目录下，打开cmd命令，执行mvn clean package命令，对项目进行打包编译。
2. 编译成功后，生成如表2-4所示的两个Jar包。

表 2-4 软件包列表

软件包所在目录	软件包名称	说明
\nacos-spring-cloud-consumer-example\target	nacos-spring-cloud-consumer-example-0.2.0-SNAPSHOT.jar	服务消费者
\nacos-spring-cloud-provider-example\target	nacos-spring-cloud-provider-example-0.2.0-SNAPSHOT.jar	服务生产者

步骤5 部署Spring Cloud应用。

将微服务Provider和Consumer部署至Nacos引擎所在VPC的ECS节点。

1. 在引擎实例所属VPC下创建一台ECS节点并登录。
2. 安装JRE，为服务提供运行环境。
3. 将[步骤4](#)生成JAR包上传至ECS节点。
4. 执行命令：`java -jar {对应jar包}`，运行生成的JAR包。

步骤6 确认部署结果。

1. **可选:** 在微服务引擎控制台页面，在左侧导航栏选择“注册配置中心”，单击[创建注册配置中心](#)中创建的Nacos引擎。
2. 选择“服务管理”，查看微服务service-consumer和service-provider的实例数量。
 - 若实例数量值不为0，则表示已经成功接入Nacos引擎。
 - 若实例数量为0，或者找不到service-consumer和service-provider服务名，则表示微服务应用接入Nacos引擎失败。

----结束

3 使用 CSE 前必读

在您需要使用微服务引擎CSE，需要满足以下条件：

1. 已注册云平台账号。
2. 当前登录账号已授权使用CSE，具体操作请参考[创建用户并授权使用CSE](#)。

约束与限制

当您没有授予任何权限时，由于CSE的使用依赖VPC、DNS等云服务，因此需要先创建云服务委托，将操作权限委托给CSE，即在“授权给CSE”弹出框中单击“确定”。CSE将在统一身份认证服务为您创建名为cse_admin_trust的委托，授权成功后，可以进入服务委托列表查看。该操作需要有Security Administrator角色权限。如不授权，将影响微服务引擎部分功能的正常使用，包括：创建引擎、升级引擎、开启/关闭安全认证。

图 3-1 创建委托



4 创建用户并授权使用 CSE

如果您需要对您所拥有的微服务引擎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，在您的账号中，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，并使用微服务引擎资源。
- 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。
- 将微服务引擎资源委托给更专业、高效的其他账号或者云服务，这些账号或者云服务可以根据权限进行代运维。

如果账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，不影响您使用微服务引擎CSE服务的其它功能。

本章节为您介绍对用户授权的方法，操作流程如[图4-1](#)所示。

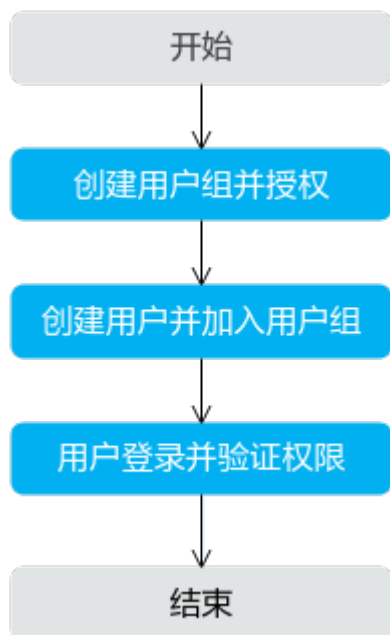
前提条件

给用户组授权之前，请您了解用户组可以添加的微服务引擎权限，并结合实际需求进行选择。

- 微服务引擎支持的系统权限，请参见[权限管理](#)。
- 若您需要对除CSE之外的其它服务授权，IAM支持服务的所有权限请参见《[系统权限](#)》。

示例流程

图 4-1 给用户授权微服务引擎权限流程



1. 创建用户组并授权
在IAM控制台创建用户组，并授予CSE“CSE ReadOnlyAccess”权限。
2. 创建用户并加入用户组。
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，验证微服务引擎的只读权限。
 - 在“服务列表”中选择“微服务引擎 CSE”，进入微服务引擎控制台，在左侧导航栏选择“ServiceComb”，单击“购买微服务引擎”，若提示权限不足，表示“CSE ReadOnlyAccess”已生效。
 - 在“服务列表”中选择除微服务引擎 CSE外的任一服务，若提示权限不足，表示“CSE ReadOnlyAccess”已生效。

CSE 自定义策略

如果系统预置的CSE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：《统一身份认证服务用户指南》的“创建自定义策略”。本章为您介绍常用的CSE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"
    }
  ]
}
```

Deny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：

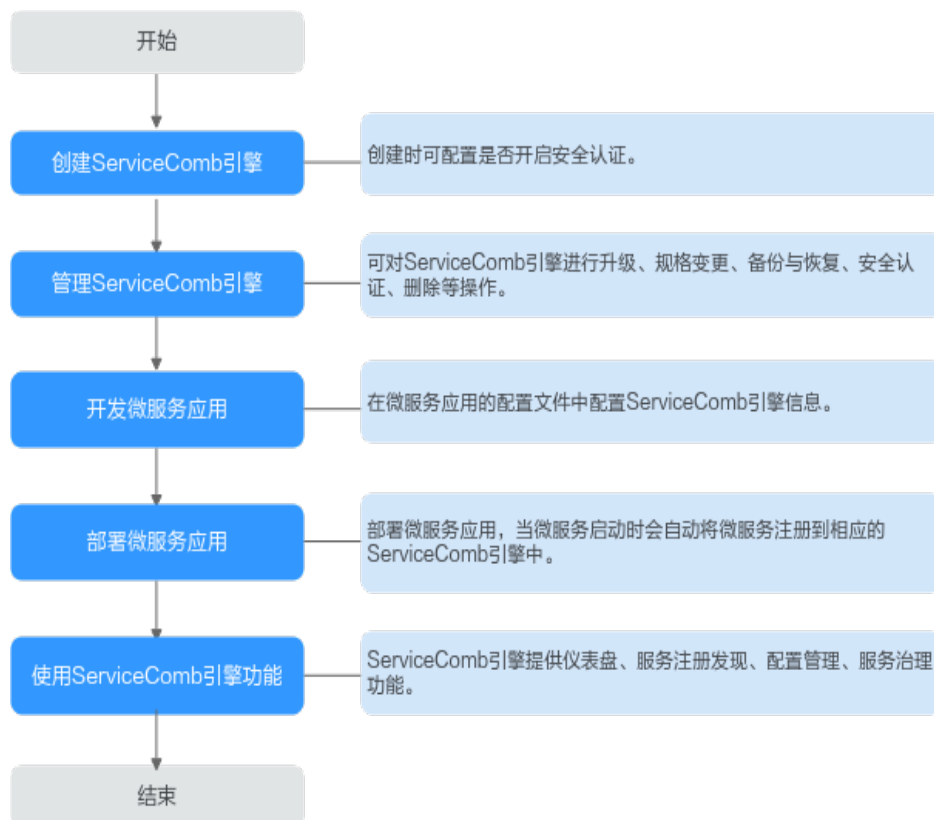
1. 使用新创建的用户登录，登录方法选择“IAM用户”。
 - “租户名”为该IAM用户所属账号的名称。
 - “IAM用户名”和“IAM用户密码”为以“租户名”在IAM创建用户时输入的用户名和密码。
2. 在微服务引擎控制台，创建微服务引擎，返回403错误，表示权限配置正确并已生效。

5 ServiceComb 引擎

5.1 ServiceComb 引擎概述

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

ServiceComb引擎的业务使用流程如下：



1. 创建ServiceComb引擎。

2. 参考[管理ServiceComb引擎](#)对ServiceComb引擎进行升级、规格变更、备份与恢复、安全认证和删除等操作。
3. 在开发完成的微服务应用配置文件中配置ServiceComb引擎的信息，包括配置中心、注册中心、仪表盘和治理策略等。
4. 完成微服务部署，当微服务启动时会自动注册到ServiceComb引擎中。
5. 参考[使用微服务仪表盘查看微服务运行指标~系统管理](#)，使用ServiceComb引擎提供的功能，对微服务进行相应管理。

5.2 创建 ServiceComb 引擎

本章节指导您根据实际业务需求创建ServiceComb引擎。

约束与限制

- ServiceComb引擎使用期间，请勿停用企业项目，否则会导致该引擎在引擎列表中不可见，影响正常使用。
- 引擎创建完成后，不支持变更虚拟私有云。
- 每个项目默认最多可以创建5个ServiceComb引擎。ServiceComb引擎同Nacos引擎共享配额数，最多不超过5个。

前提条件

- 当前登录账号拥有创建微服务引擎的权限。账号权限授权与绑定，请参考[创建用户并授权使用CSE](#)。
- ServiceComb引擎运行于虚拟私有云，创建ServiceComb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《虚拟私有云用户指南》中“创建虚拟私有云和子网”章节。
- 如果引擎创建账号的权限为创建引擎的最小权限，如[CSE权限](#)中的“cse:engine:create”所示。则需要由主账号为其预置VPC默认安全组cse-engine-default-sg，并添加如[表5-1](#)所示规则。添加安全组规则，请参考《虚拟私有云用户指南》中“添加安全组规则”章节。

表 5-1 默认安全组 cse-engine-default-sg 规则说明

方向	优先级	策略	协议端口	类型	源地址
入方向	1	允许	ICMPV6 : 全部	IPv6	::/0
	1	允许	TCP : 30100-30130	IPv6	::/0
	1	允许	全部	IPv6	cse-engine-default-sg
	1	允许	全部	IPv4	cse-engine-default-sg
	1	允许	TCP : 30100-30130	IPv4	0.0.0.0/0
	1	允许	ICMP : 全部	IPv4	0.0.0.0/0

方向	优先级	策略	协议端口	类型	源地址
出方向	100	允许	全部	IPv4	0.0.0.0/0
	100	允许	全部	IPv6	::/0





创建 ServiceComb 引擎

步骤1 登录微服务引擎控制台。

步骤2 在左侧导航栏选择“ServiceComb引擎专享版”，单击“创建ServiceComb引擎”。

步骤3 参考下表设置参数，参数前面带*号的是必须设置的参数。

参数	说明
*企业项目	<p>选择ServiceComb引擎所在的项目，可在下拉框中搜索和选择需要的企业项目。</p> <p>企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。</p> <p>参考《企业管理用户指南》的“创建企业项目”章节，创建并启用新的企业项目后可以使用，默认选择default。</p> <p>ServiceComb引擎创建完成后，可将ServiceComb引擎资源迁出当前所在企业项目，迁入到新的企业项目，具体操作请参考《企业管理用户指南》的“迁出企业项目资源”章节与《企业管理用户指南》的“为企业项目迁入资源”章节。</p>
*选择实例数	<p>选择微服务实例数配额。您可根据需要托管的微服务实例数量，选择相应的实例数，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。</p>
*引擎类型	<p>选择ServiceComb引擎的类型。</p> <p>引擎类型为集群，其为集群模式部署，主机级容灾。</p>
*ServiceComb引擎名称	<p>输入ServiceComb引擎的名称，名称以字母开头，由字母、数字和-组成，且不能以-结尾，长度为3~24个字符。ServiceComb引擎名称不允许命名为“default”。</p>
*可用区	<p>选择可用区。</p> <p>根据环境可用区数量，为引擎选择1个或者3个可用区。</p> <ul style="list-style-type: none"> 选择1个可用区，可提供主机级别容灾能力。 选择3个可用区，可提供可用区级别容灾能力。 <p>说明</p> <ul style="list-style-type: none"> 同一个区域内的可用区内网互通。 多可用区可加强容灾能力。

参数	说明
*网络	<p>为引擎选择虚拟私有云及其子网，可以为您的引擎构建隔离的、自主配置和管理的虚拟网络环境。</p> <ul style="list-style-type: none"> 使用已创建的VPC，可在下拉列表中搜索和选择当前账号已创建的虚拟私有云和子网。 使用新的VPC，在下拉列表中单击“创建新虚拟私有云”，创建新的虚拟私有云，具体操作请参考《虚拟私有云用户指南》中“创建虚拟私有云和子网”章节。
描述	<p>单击，输入引擎描述信息，可输入任意字符，长度为0 ~ 255个字符。</p>
标签	<p>用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。</p> <p>您可以单击“ 添加标签”，在“添加标签”弹框输入标签键和标签值，添加标签，标签的命名规则请参见管理标签。在“添加标签”弹框，可单击“ 新增标签”同时添加多个标签，也可单击标签后的，删除该标签。</p>
安全认证	<p>开启了“安全认证”的ServiceComb引擎专享版，通过微服务引擎控制台提供了基于RBAC（Role-Based Access Control，基于角色的访问控制）的系统管理功能。</p> <ul style="list-style-type: none"> 选择“开启安全认证”： <ol style="list-style-type: none"> 根据业务需要确认是否需要开启“编程接口安全认证”。开启编程接口安全认证后，需要在微服务的配置文件中添加对应用户的账号密码，否则服务无法注册到引擎。 关闭编程接口安全认证，微服务的配置文件中无需配置账号密码即可将服务注册到引擎，效率性能更高，建议用于VPC内访问时使用。 输入root账号的“密码”，并在“再次输入密码”输入框输入密码进行确认。 密码请妥善保管，以免遗失。 选择“关闭安全认证”： <p>关闭安全认证功能，可以在实例创建完成后再设置开启。</p>

步骤4 单击“立即创建”，进入引擎信息确认界面。

步骤5 单击“提交”，当引擎状态为“可用”时，表示创建成功。

📖 说明

- ServiceComb引擎创建完成，大约需要10-30分钟。
- ServiceComb引擎创建成功后，“状态”为“可用”。查看ServiceComb引擎状态，请参考[查看ServiceComb引擎信息](#)。
- 如果ServiceComb引擎创建失败，可在操作日志页面上查看失败原因并处理后可进行以下操作：
 - 可在“ServiceComb引擎信息”区域，单击“重试”重新创建。
 - 如果重试失败，可删除创建失败的ServiceComb引擎，删除ServiceComb引擎，请参考[删除ServiceComb引擎](#)。

----结束

5.3 管理 ServiceComb 引擎

5.3.1 查看 ServiceComb 引擎信息

您可单击待查看的引擎名称，进入引擎详细信息页面，获取基本信息、服务注册发现地址、配置中心地址、实例数配额和配置条目配额。

查看 ServiceComb 引擎基本信息

在“ServiceComb引擎信息”区域，可以查看如[表5-2](#)所示的引擎信息，了解引擎的基本配置。

步骤1 登录微服务引擎控制台。

步骤2 在左侧导航栏选择“ServiceComb引擎专享版”。

步骤3 单击待查看的引擎。

步骤4 在“ServiceComb引擎信息”区域，查看如[表5-2](#)所示的ServiceComb引擎信息。

表 5-2 引擎详细信息说明

引擎信息	信息说明
引擎名称	创建ServiceComb引擎 时输入的引擎名称，单击📄可复制。
引擎ID	引擎的ID，单击📄可复制。

引擎信息	信息说明
状态	引擎的状态，包括： <ul style="list-style-type: none"> ● 创建中 ● 可用 ● 不可用 ● 配置中 ● 删除中 ● 升级中 ● 变更中 ● 创建失败 ● 删除失败 ● 升级失败 ● 变更失败 ● 未知
引擎版本	引擎的版本号。
引擎类型	创建ServiceComb引擎 时选择的引擎类型。
可用区	创建ServiceComb引擎 时选择的可用区。
标签	显示为ServiceComb引擎添加的标签，也可单击“标签管理”，根据需要对标签进行相应操作，具体操作可参见 管理ServiceComb引擎标签 。
描述	创建ServiceComb引擎 时输入的引擎描述信息。

---结束

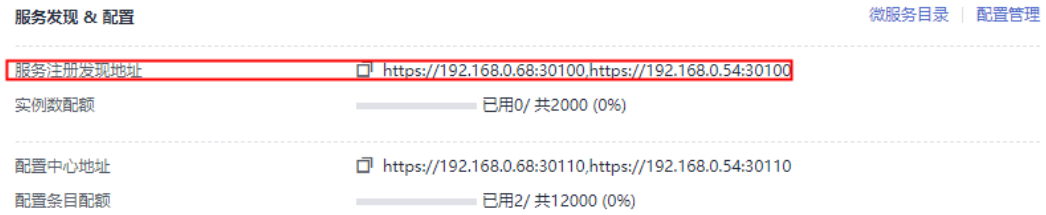
获取 ServiceComb 引擎服务注册发现地址

注册发现地址是服务注册与发现的核心，实现服务动态管理。微服务启动时，通过注册中心地址将自身元数据（如IP、端口、服务契约、版本）上报至ServiceComb的服务中心，实现服务的自动注册，避免硬编码地址，简化运维。服务消费者通过该地址从注册中心查询可用服务实例列表，结合负载均衡（如加权轮询）选择实例调用，支持动态感知服务上下线（通过心跳检测），确保故障实例自动剔除，提升系统容错性。

当引擎创建完成后，注册发现地址不可修改。

步骤1 单击待查看的引擎。

步骤2 在“服务发现 & 配置”区域，查看获取引擎服务注册发现地址。



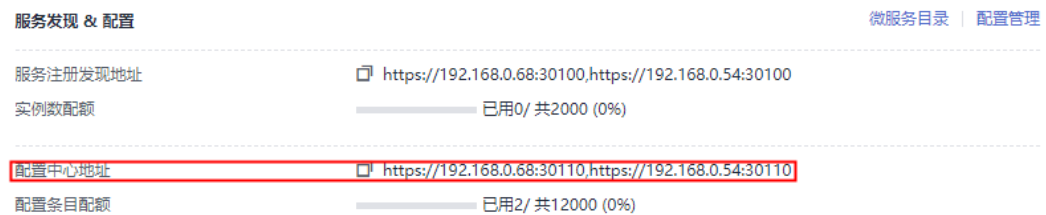
----结束

获取 ServiceComb 引擎配置中心地址

配置中心集中管理微服务配置，微服务接入ServiceComb引擎时，需要在配置文件中配置引擎的配置中心地址，ServiceComb通过配置中心地址与配置中心建立长连接，当配置中心检测到配置变更时，主动推送给微服务实例，无需重启即可刷新配置，实现动态配置管理。

步骤1 单击待查看的引擎。

步骤2 在“服务发现 & 配置”区域，查看获取引擎的配置中心地址。



说明

- 当ServiceComb引擎为1.x版本时，其配置中心地址端口号为30103。

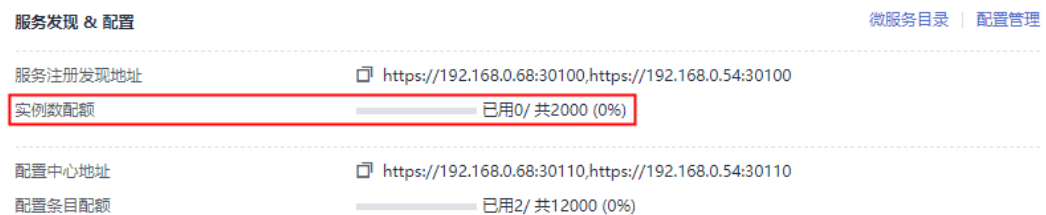
----结束

查看 ServiceComb 引擎实例数配额

您可在此处查看ServiceComb引擎的实例数配额及配额使用情况。

步骤1 单击待查看的引擎。

步骤2 在“服务发现 & 配置”区域，查看引擎的实例数配额及配额使用情况。



----结束

查看 ServiceComb 引擎配置条目配额

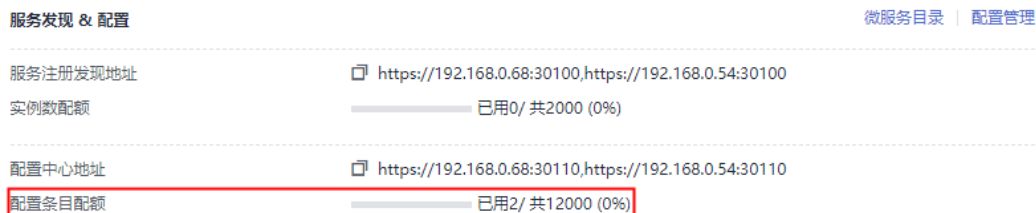
您可在此处查看ServiceComb引擎的配置条目配额及配额使用情况。

说明

配置条目配额只适用于ServiceComb引擎版本为2.x。

步骤1 单击待查看的引擎。

步骤2 在“服务发现 & 配置”区域，查看引擎的配置条目配额及配额使用情况。



---结束

5.3.2 管理 ServiceComb 引擎标签

标签是ServiceComb引擎的标识，为ServiceComb引擎添加标签，可以方便用户识别和管理拥有的ServiceComb引擎资源。

如您的组织已经设定ServiceComb引擎的相关标签策略，则需按照标签策略规则为引擎添加标签。标签不符合标签策略的规则，则可能会导致引擎创建失败，请联系组织管理员了解标签策略详情。

您可以在创建ServiceComb引擎时添加标签，也可以在ServiceComb引擎创建完成后，在引擎的详情页添加标签，您最多可以给引擎添加20个标签。另外，您还可以修改和删除标签。

标签共由两部分组成：标签键和标签值，其中，标签键和标签值的命名规则如表5-3所示。

表 5-3 标签命名规则

标签信息	规则
标签键	<ul style="list-style-type: none"> 不能为空。 对于同一个实例，Key值唯一。 长度不超过128个字符。 只能包含任意语种字母、数字、空格和特殊字符（_ . : = + - @ ）。 不能以空格开头和结尾，且不能以_sys_开头。
标签值	<ul style="list-style-type: none"> 长度不超过255个字符。 只能包含任意语种字母、数字、空格和特殊字符（_ . : = + - @ ）。

约束与限制

新增和修改标签会影响引擎业务十秒左右，请在业务低峰期时增加或修改。

管理标签

步骤1 单击待操作的引擎的名称，进入该引擎的详细信息页面。

步骤2 在“ServiceComb引擎信息”区域的“标签”参数处，您可以根据实际需要，执行以下操作：

- 新增标签
 - a. 单击“标签管理”，弹出“编辑标签”窗口。
 - b. 单击“⊕ 新增标签”，您可在输入框分别输入标签键和标签值。
 - c. 单击“确定”，为实例添加标签成功。
- 修改标签
 - a. 单击“标签管理”，弹出“编辑标签”窗口。
 - b. 您可在原有的输入框中修改标签键与标签值信息。
 - c. 单击“确定”，标签修改成功。
- 删除标签

单击标签所在行的⊖，在弹出的确认框中单击“确定”，删除该标签。

---结束

5.3.3 管理 ServiceComb 引擎公网访问

ServiceComb引擎支持公网访问，有利于将微服务架构从内网的“封闭环境”拓展到公网的“开放生态”，不仅满足互联网业务、跨云协同等场景需求，还通过技术架构的开放性提升了系统的弹性、可扩展性和灾备能力。同时，结合安全与治理机制，公网访问在“开放”与“可控”之间实现平衡，为企业数字化转型中“内外联动”的业务模式提供了关键技术支持。

约束与限制

- 未开启安全认证的ServiceComb引擎无认证鉴权能力，开放到公网面临安全风险，增加系统的脆弱性。如：配置、服务信息等数据资产可能会被窃取。
- 请不要在生产环境和安全要求较高的网络环境中使用该功能。

前提条件

在绑定公网IP前，需已创建可用的弹性公网IP。创建弹性公网IP，请参考《弹性公网IP用户指南》的“获取EIP”章节。

绑定公网 IP

未绑定公网IP的ServiceComb引擎，绑定公网IP后可提供公网访问引擎的能力。

步骤1 单击待操作的引擎。

步骤2 在“网络配置 & 安全”区域，单击“绑定公网IP”。



步骤3 阅读弹出的对话框中的安全风险提示，勾选“我已知晓安全风险”。

步骤4 在“选择公网IP”下拉列表，选择待绑定的公网IP。只能选择与ServiceComb引擎同企业项目下的公网IP。

步骤5 单击“确定”，完成绑定。

----结束

解绑公网 IP

已绑定公网IP的ServiceComb引擎，可通过解绑公网IP取消公网访问引擎的能力。

步骤1 单击待操作的引擎。

步骤2 在“网络配置 & 安全”区域，单击“解绑公网IP”。



步骤3 在弹出的对话框中单击“确定”，完成解绑。

----结束

5.3.4 管理 ServiceComb 引擎安全认证

同一个ServiceComb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ServiceComb引擎访问和操作权限。开启了“安全认证”的ServiceComb引擎专享版，根据用户接入引擎使用的账号所关联的角色，赋予该用户不同的ServiceComb引擎访问和操作权限。

安全认证具体内容，请参考[系统管理](#)。

您可根据实际业务需要，对ServiceComb引擎专享版执行开启或关闭安全认证操作：

- 开启安全认证**

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

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

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

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

约束与限制

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

开启安全认证

步骤1 单击待操作的引擎。

步骤2 在“网络配置 & 安全”区域，单击“开启安全认证”。

- 如果引擎版本低于1.2.0，执行**步骤3**。
- 如果引擎版本为1.2.0及以上版本，执行**步骤4**。

步骤3 升级引擎至1.2.0或以上版本。

1. 单击“升级”。
2. 选择“升级后版本”，查看版本说明，根据需要决定是否升级到该版本后，单击“确定”。
3. 在刚升级成功的ServiceComb引擎的“网络配置 & 安全”区域，单击“开启安全认证”。

步骤4 在“系统管理”页面开启安全认证。

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

步骤5 （可选）参考[角色管理](#)，根据业务需要，创建角色。

步骤6 （可选）参考[账号管理](#)，根据业务需要，创建账号。

步骤7 在“系统管理”页面单击“设置安全认证”，根据实际业务需要设置安全配置。

- 选择开启“控制台安全认证”，请执行**步骤9**。
开启控制台安全认证后，进入微服务引擎控制台界面，根据当前登录控制台的权限判断是否需要使用账号、密码登录。登录账号用户只能查看、配置有权限的服务。
- 选择开启“编程接口安全认证”，请执行**步骤8**。
开启编程接口安全认证，会自动同步开启“控制台安全认证”。
开启编程接口安全认证后，需要在微服务的配置文件中添加对应用户的账号密码，否则服务无法注册到引擎。
关闭编程接口安全认证，微服务的配置文件中无需配置账号密码即可将服务注册到引擎，效率性能更高，建议用于VPC内访问时使用。

步骤8 配置SDK，对于已部署但未配置安全认证参数的微服务组件，先为组件配置微服务安全认证的账号名和密码，再升级组件。

步骤9 单击“确定”。

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

----结束

关闭安全认证

关闭安全认证后，开启安全认证时该引擎下创建的账号不会被删除。

步骤1 单击待操作的引擎。

步骤2 在“网络配置 & 安全”区域，单击“关闭安全认证”。

步骤3 单击“确认”，等待ServiceComb引擎更新完成，引擎状态由“配置中”变为“可用”，关闭安全认证成功。

----结束

5.3.5 设置 ServiceComb 引擎备份与恢复

微服务引擎提供备份与恢复功能，可对ServiceComb引擎的数据，包括微服务、契约、配置、账号角色信息进行备份与恢复。

您可自定义备份策略，对ServiceComb引擎进行定期备份，也支持手动对ServiceComb引擎进行备份。

约束与限制

- 每个ServiceComb引擎最多有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。
- 备份数据恢复会使用备份的历史数据（包括微服务、契约、配置、账号角色信息）覆盖ServiceComb引擎当前数据，可能导致微服务、服务实例错乱，动态配置丢失等问题，请谨慎操作！
- 若ServiceComb引擎开启了安全认证，则备份数据中包含其账号信息，建议关闭安全认证后再进行恢复备份数据，否则恢复后可能存在访问ServiceComb引擎认证失败的问题。

自动备份

步骤1 单击待操作的引擎。

步骤2 在“备份与恢复”区域，单击“自动备份设置”，配置备份参数。

表 5-4 自动备份参数说明

参数	说明
自动备份	关闭自动备份后将会删除之前设置的备份策略，需要重新设置。
备份周期	设置备份周期。备份周期指进行自动备份操作的时间间隔规律。可选择周一到周日中的一天或多天作为备份执行的日期，被选中的日期便会触发自动备份。 开启“自动备份”后生效。

参数	说明
触发时间	设置备份任务开启的时间，仅支持整点，即在设置的备份周期的所有日期的该时间点触发自动备份。 开启“自动备份”后生效。

步骤3 单击“确定”，完成备份策略的配置。

设置备份策略后，将在预设时间之后1小时内触发备份任务。

----结束

手动备份

步骤1 单击待手动备份数据的引擎。

步骤2 在“备份与恢复”区域，单击“新增手动备份”，配置备份参数。

表 5-5 手动备份参数说明

参数	说明
名称	输入备份任务的名称，名称以字母开头，由字母、数字和-组成，且不能以-结尾，长度为3 ~ 24个字符。创建备份任务后不能修改名称。
备注	(可选)添加描述信息，可输入任意字符，长度为0 ~ 255个字符。

步骤3 单击“确定”，立即执行备份任务，在备份任务列表，生成一个备份任务，执行结果由“执行中”变为“成功”，手动备份成功。

----结束

恢复备份数据

备份数据恢复会使用备份的历史数据（包括微服务、契约、配置、账号角色信息）覆盖ServiceComb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 单击待操作的引擎。

步骤2 在“备份与恢复”区域，单击指定备份数据后面“操作”列的“恢复备份”。

1. 勾选“我已经阅读并充分了解该风险”。
2. 单击“确定”，恢复备份数据。可在“备份与恢复”区域单击“恢复记录”，查看恢复状态。
3. 恢复成功后，选择“配置管理”，在配置管理页查看配置是否恢复成功。

----结束

删除备份数据

步骤1 单击待删除备份数据的引擎。

步骤2 在“备份与恢复”区域，单击指定备份数据右侧“操作”列的“删除”，在弹出的对话框中输入“DELETE”，单击“确定”。

----结束

5.3.6 升级 ServiceComb 引擎版本

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

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

约束与限制

- ServiceComb引擎升级过程中微服务与引擎的连接会出现闪断，已运行的微服务的业务不受影响。建议您在升级ServiceComb引擎时，尽量避开微服务的升级、重启、变更等操作。
- 当升级完成后，不支持版本回退。

升级 ServiceComb 引擎版本

步骤1 单击待升级的引擎“操作”列的“升级”，也可单击待升级的引擎，在“ServiceComb引擎信息”区域，单击“升级”。

步骤2 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3 单击“确定”，进行升级，引擎状态由“升级中”变为“可用”，则表示升级完成。如果升级失败，可单击“重试”，尝试再次进行升级。

----结束

5.3.7 变更 ServiceComb 引擎规格

ServiceComb引擎专享版规格支持在线自动变更。当前只支持扩容，不支持缩容场景。变更时业务侧感知到会有闪断，对业务本身没影响。变更期间不支持注册新服务。

约束与限制

变更时业务侧感知到会有闪断，对业务本身没影响。变更期间不支持注册新服务。

变更 ServiceComb 引擎规格

步骤1 在待操作的引擎“操作”列，选择“更多 > 规格变更”，也可单击待操作的引擎，在“ServiceComb引擎信息”区域，单击“规格变更”。

步骤2 在“ServiceComb引擎规格变更”页面，选择目标规格。

步骤3 单击“立即变更”，确认无误后，单击“提交”，引擎状态由“变更中”变为“可用”，表示变更完成。

----结束

5.3.8 查看 ServiceComb 引擎操作日志

在“操作日志”区域，可以查看ServiceComb引擎的操作日志信息，记录用户对引擎的操作行为，有助于安全合规和审计。


查看 ServiceComb 引擎操作日志

步骤1 单击待查看的引擎。

步骤2 在“操作日志”区域，查看ServiceComb引擎的操作日志信息。



序号	任务类型	操作结果	用户名	启动时间	结束时间	详细信息
1	创建	成功	gmsw_user_bms11@huawei.com	2021/04/16 20:47:55 GMT+08:00	2021/04/16 21:00:32 GMT+08:00	创建引擎 更多

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

----结束

5.3.9 删除 ServiceComb 引擎

如不再使用ServiceComb引擎，可执行删除操作。支持删除处于如下状态的ServiceComb引擎：

- 可用
- 不可用
- 创建失败
- 变更失败
- 升级失败
- 未知

约束与限制

- 删除引擎后数据无法恢复，请谨慎操作。
- 对于1.x版本引擎，缺少cse_admin_trust委托时，删除引擎，租户侧资源DNS、VPC、安全组会残留，需要租户自行删除。

删除 ServiceComb 引擎

步骤1 单击待删除的引擎“操作”列的“删除”，也可单击待删除的引擎，在“ServiceComb引擎信息”区域，单击“删除”。

步骤2 在弹出的对话框中输入“DELETE”，单击“确定”。

说明

若删除失败，可执行“强制删除”。

----结束

5.4 使用微服务仪表盘查看微服务运行指标

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。

约束与限制

- 当ServiceComb引擎为1.x或2.4.0及以上版本时，支持此功能。
- 如果微服务应用部署在ServiceStage上，部署应用时需要设置微服务引擎，应用会自动获取服务注册发现地址、配置中心地址和仪表盘地址，不需要配置monitor地址，就可以使用仪表盘功能。
- 如果是本地启动微服务应用注册到ServiceComb引擎，需要手工配置monitor地址，才可以使用仪表盘功能。
- 当使用Spring Cloud Huawei框架接入时，仪表盘上暂不支持查看中位数时延、90th时延、99th时延。

查看微服务运行指标

步骤1 单击待操作的引擎。

步骤2 单击“仪表盘”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

说明

- 首次连接ServiceComb引擎，请输入root账号名及**创建ServiceComb引擎**时输入的密码。
- 创建账号请参考**新增账号**。

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

单击“查看示例图”，可以查看运行指标参数含义。

步骤5 选择排序方式，筛选出的微服务会按照指定方式进行排序。

----结束

5.5 管理微服务

5.5.1 查看应用

在应用列表页签下，显示当前ServiceComb引擎的所有应用，支持使用应用名称搜索目标应用，也可以通过选择环境筛选应用。

查看应用列表

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“应用列表”，查看该引擎下当前账号的所有应用的详细信息。

支持使用应用名称搜索目标应用，也可以选择环境，筛选应用。

----结束

5.5.2 微服务管理

在微服务列表页签下，可对微服务进行创建、查看、清理无实例版本、动态配置、灰度发布和删除微服务操作。

创建微服务

创建新的微服务，可用于测试或恢复引擎中误删的微服务。当自动注册到ServiceComb的微服务被误删后，可通过手动创建微服务恢复误删的微服务，其中需使用与原服务完全相同的微服务名称、微服务应用、版本和微服务环境。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 选择“微服务列表 > 创建微服务”，参考下表配置微服务参数。参数前面带*号的是必须设置的参数。

参数	说明
*微服务名称	输入微服务名称，例如：myServiceName。以数字、字母开头结尾，由数字、字母和特殊字符（_-.）组成，且特殊字符不能连续出现，长度为1 ~ 128字符。
*微服务应用	输入微服务所属应用的名称，微服务间通过应用隔离的。以数字、字母开头结尾，由数字、字母和特殊字符（_-.）组成，且特殊字符不能连续出现，长度为1 ~ 160字符。
*版本	输入微服务的版本号，默认为1.0.0。 微服务版本号格式为“X.Y.Z”，“X.Y.Z.B”型，其中X、Y、Z、B为数字且范围在0-32767，长度为3-46个字符。

参数	说明
*微服务环境	选择微服务所在环境，用于隔离微服务数据，包括：版本、实例。
描述	输入微服务描述信息。

步骤5 单击“确定”。

创建完成后，微服务会显示在微服务列表。

----结束

查看微服务列表

显示当前ServiceComb引擎的所有微服务，支持使用微服务名称搜索目标服务，也可以选择环境、应用筛选微服务。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“微服务列表”，查看该引擎下当前账号下的所有微服务。

支持使用微服务名称搜索目标服务，也可以选择环境、应用筛选服务。

----结束

查看微服务详情

在微服务详情页可以分别查看实例列表、被调用服务、调用服务、动态配置、服务契约等信息，也可对微服务级配置进行**动态配置**和**灰度发布**。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“微服务列表”下待查看的微服务名称，进入微服务详情页查看实例列表、被调用服务、调用服务、动态配置、灰度发布和服务契约等信息。

----结束

清理无实例版本

清理实例数为0的微服务版本，即删除没有运行中实例的服务版本，主要用于优化资源利用、提升治理效率和保障系统稳定性。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 选择“微服务列表 > 清理无实例版本”，勾选待清理的无实例微服务版本。

支持使用微服务名称搜索目标服务，也可以选择环境、应用筛选微服务。

步骤5 单击“确定”。

----结束

动态配置

可以创建微服务级配置，也可对微服务级配置进行查看历史版本、编辑、禁用和删除操作。

配置项使用明文存储，请勿附带敏感数据。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“微服务列表”，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：

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

---结束

灰度发布

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

- 基于ServiceComb Java Chassis框架开发的微服务需要在POM中添加依赖darklaunch或handler-router，并且在配置文件microservice.yaml中增加servicecomb.router.type=router。
- 基于Spring Cloud Huawei框架开发的微服务需要在POM中增加依赖spring-cloud-starter-huawei-router。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的VDC用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

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

步骤5 单击“添加发布规则”，弹出“创建新规则”弹框。

- 按照权重方式添加灰度规则。
 - “发布规则”选择“权重”，显示权重规则设置页签。
 - 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。以英文字母开头，支持数字、字母和特殊字符(.@_-),长度为3~24个字符。
作用域	<ul style="list-style-type: none"> ▪ 选择规则作用的版本。 ▪ 勾选“是否添加自定义版本”，根据界面提示添加新版本。
规则配置	配置所选版本的引流权重，流量会按照配置的百分比分配给所选择的服务版本。

- 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。
 - “发布规则”选择“自定义”，显示自定义规则设置页签。
 - 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。以英文字母开头，支持数字、字母和特殊字符 (.@_-),长度为3 ~ 24个字符。
作用域	<ul style="list-style-type: none"> ▪ 选择规则作用的版本。 ▪ 勾选“是否添加自定义版本”，根据界面提示添加新版本。
规则配置	<p>设置匹配规则的参数，使用darklaunch实现灰度能力时，设置的是policyCondition内容，使用handler-router实现灰度能力时，设置的是Headers中参数。</p> <p>Spring Cloud Huawei框架开发的微服务设置的Headers参数请参见设置Header参数。</p> <ul style="list-style-type: none"> ▪ 参数名 根据契约的参数名或Header自定义的key作为参数名设置。 ▪ 规则 通过选择匹配符和契约的Key或Header的Key对应的Value确定符合该规则请求将被分配到微服务版本中。 <p>说明</p> <ul style="list-style-type: none"> ○ 当匹配符选~时，可使用*或者?字符的模糊匹配规则自定义规则值，*表示任意长度字符，?表示一个字符。比如，字段Name的规则值配置为*1000时，表示Name后4位为1000的都能匹配上。 ○ 当匹配符没有选~时，*或者?字符只是普通字符，没有模糊匹配的作用。

c. 单击“确定”，完成自定义规则配置和灰度发布。

----结束

下发灰度发布规则样例如下：

- ServiceComb Java Chassis框架开发的微服务，在ServiceComb引擎页面下发规则是基于darklaunch依赖下发的，通过配置自定义方式添加灰度规则。

创建新规则

* 发布规则

权重 自定义

* 规则名称

作用域

* 版本 是否添加自定义版本

规则配置

* 参数名

* 规则

• 参数符合 `[name=11111]` 条件的请求将会分配到 0.0.1 微服务版本中

确定 取消

其中，在规则配置中，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。

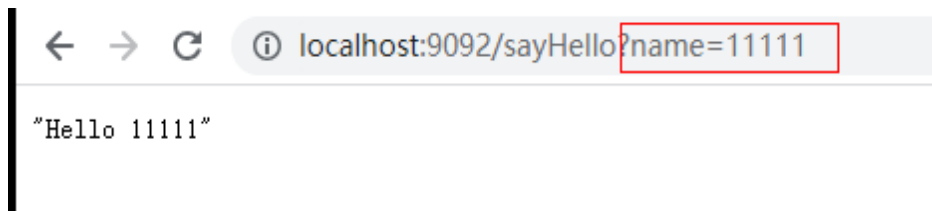
Consumer...

Swagger Yaml

⚠ 请您在服务契约中慎用敏感信息，或进行敏感信息加密，以免造成信息泄露。例如：用户

```
11 application/json
12 produces:
13 - "application/json"
14 paths:
15   /sayHello:
16     get:
17       operationId: "sayHello"
18       parameters:
19         - name: "name"
20           in: "query"
21             required: true
22             type: "string"
23       responses:
24         "200":
25           description: "response of 200"
26           schema:
27             type: "string"
28
```

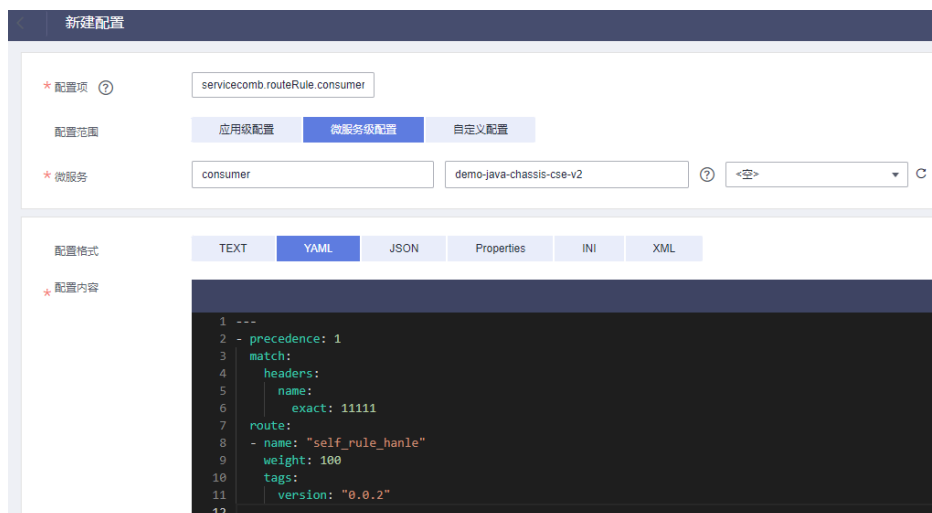
通过选择匹配符和契约的Key对应的Value确定符合该规则请求将被分配到微服务版本中。



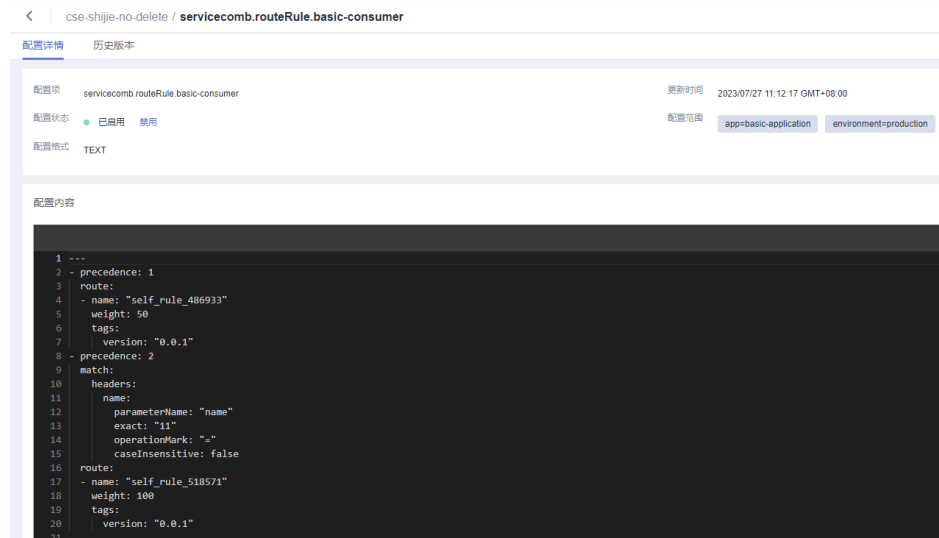
成功下发的规则内容如下：配置项为cse.darklaunch.policy.\${serviceName}。



- ServiceComb Java Chassis框架开发的微服务，依赖handler-router下发的灰度规则需要手动在配置中心下发，配置项为：servicecomb.routeRule.\${serviceName}，内容如下：



- Spring Cloud Huawei框架开发的微服务，ServiceComb引擎页面下发灰度规则如下：



删除微服务

根据实际业务需求，删除不再使用的微服务。

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

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行[步骤4](#)。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行[步骤4](#)，其他用户请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“微服务列表”，进入微服务列表：

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

步骤5 在输入框输入“DELETE”确认删除，单击“确定”。

----结束

5.5.3 实例管理

CSE支持对注册到ServiceComb引擎中的微服务下的所有实例进行查看和修改状态操作。

查看实例列表

显示当前ServiceComb引擎的所有实例，支持使用微服务名称搜索目标实例，也可以选择环境、应用筛选实例。

步骤1 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“实例列表”，进入实例列表查看引擎的所有实例。

支持使用微服务名称搜索目标实例，也可以选择环境、应用筛选实例。

----结束

修改实例状态

“实例状态”是微服务实例信息的一部分，用于反映微服务实例的状态。

在《应用管理与运维平台用户指南》的“创建部署组件”中通过绑定ServiceComb引擎同步的微服务实例不支持状态修改。

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ServiceComb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 单击待操作的引擎。

步骤2 单击“微服务目录”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“实例列表”，选择目标实例，修改微服务实例状态。

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

----结束

5.6 业务场景治理（适用于 ServiceComb 引擎为 2.x 版本）

5.6.1 业务场景治理概述

ServiceComb引擎针对不同的微服务开发框架，比如Spring Cloud、Java Chassis等设计了统一的基于动态配置的流量特征治理。通过在开发框架中引入相关治理组件，即可使用微服务引擎的微服务治理功能。

ServiceComb引擎业务场景治理分为创建业务场景和创建治理策略两个步骤。这两个步骤可以在微服务部署之前操作，比较适合您独立规划治理功能。

约束与限制

- 业务场景治理功能适用于ServiceComb引擎2.x版本。
- 若您需要删除正在使用的治理策略，需要注意以下信息：
 - 存在的风险：治理策略失效，降低微服务系统抵抗异常流量的能力，当出现异常流量时，可能发生调用分布不均衡、微服务雪崩等问题。
 - 注意事项：请在业务低谷期实施，在实施前确认无异常流量，如大规模调用、调用超时等。

治理策略说明

支持限流、熔断、重试和隔离仓等策略的配置，具体说明见下表。

治理策略名称	说明
限流	面对流量风暴，或可预知的流量冲击，对非重点业务场景进行限流，防止瞬时流量过大造成服务和数据崩溃，导致服务不可用。
重试	当服务遇到一些非致命性的错误（如偶尔超时）时，可以通过重试的方式来避免服务的最终失败。
隔离仓	面对大规模并发流量风暴，或可预知的流量冲击，对并发流量进行控制，防止瞬时并发流量过大造成服务和数据崩溃，导致服务不可用。
熔断	当某业务场景的错误率超过设定阈值时，为了保证整体业务系统的可用性，在此后的一分钟内该业务场景下的所有请求会被拒绝。然后再以50%的比率接受服务请求并统计业务的错误率，直至该业务场景下的错误率降低到设定阈值以下。

5.6.2 创建业务场景

您可根据自身业务流程和需求创建对应的业务场景，可针对性地实施治理策略，保障业务稳定、高效运行。

前提条件

- 已参考《应用管理与运维平台用户指南》的“创建部署组件”完成应用部署。
- 您需要了解待治理的微服务接口设计，根据接口特征创建业务场景。

创建业务场景

步骤1 单击待操作的引擎。

步骤2 选择“业务场景治理”。

📖 说明

当ServiceComb引擎版本为2.0.0及以上且小于2.4.0时，选择“微服务治理”。


- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 选择“业务场景 > 创建业务场景”，参考下表设置参数。

参数名称	参数说明
业务场景名称	输入业务场景名称。支持输入任意字符，长度不超过20个字符。
所属环境	选择微服务环境。
所属应用	选择待创建业务场景所属的应用。
匹配规则	<p>通过设定Method、Path和Headers规则，筛选出符合特定条件的请求。只有满足这些规则请求，才会被纳入到所创建的业务场景中，便于针对特定类型请求进行集中治理。</p> <p>可单击“添加匹配规则”，设置流量标记：</p> <ul style="list-style-type: none"> • Method：选择流量的请求特征标记方法，支持GET、PUT、POST、DELETE、PATCH请求。该规则必须配置。 • Path：设置流量请求URI所包含的特征。该规则必须配置。 • Headers：单击“添加Headers规则”，设置流量请求头所包含的特征。该规则可根据需要选择配置。

步骤5 单击“确定”，完成业务场景创建。**创建业务场景会自动生成以“servicecomb.matchgroup.”开头的配置。**

- 单击业务场景所在行的，可查看匹配规则详情。
- 单击业务场景所在行“操作”列的“编辑”，可对业务场景进行编辑。
- 单击业务场景所在行“操作”列的“删除”，可删除该业务场景。

----结束

5.6.3 创建治理策略

治理策略是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。

前提条件

- 已参考[创建业务场景](#)完成业务场景创建。
- 待治理的微服务开发框架需要启用基于动态配置的流量特征治理功能。如果没有启用，仍然可以使用微服务治理功能，但是不会产生治理效果。

创建治理策略

步骤1 单击待操作的引擎。

步骤2 选择“业务场景治理”。

📖 说明

当ServiceComb引擎版本为2.0.0及以上且小于2.4.0时，选择“微服务治理”。

- 未开启安全认证的引擎，请执行[步骤4](#)。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行[步骤4](#)，其他用户请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 在“治理策略”页签，单击“创建治理策略”。

步骤5 选择治理方式，单击“创建治理”，设置参数。

- 限流

参数名称	参数说明
治理策略名称	输入治理策略的名称。支持输入任意字符，长度不超过32个字符。
业务场景	设置治理策略适用的业务场景： <ul style="list-style-type: none"> - 单击“选择业务环境”，选择已创建好的业务场景名称。 - 单击“创建业务场景”，具体操作请参考创建业务场景。
单位请求数	设定请求数及时间段的值。 当限流对象对当前服务实例的设定时间段内的请求数量超过设定的值，超过部分将会被限流，返回错误码429。

- 重试

参数名称	参数说明
治理策略名称	输入治理策略的名称。支持输入任意字符，长度不超过32个字符。

参数名称	参数说明
业务场景	设置治理策略适用的业务场景： - 单击“选择业务环境”，选择已创建好的业务场景名称。 - 单击“创建业务场景”，具体操作请参考 创建业务场景 。
响应错误码	选择响应错误码，用于定义哪些错误类型会触发重试。
重试次数	设置重试次数。
重试策略	选择重试策略： - 固定间隔重试：每次重试的时间间隔是固定不变的。例如，若设置重试间隔时间为500毫秒，那么无论进行多少次重试，相邻两次重试之间的时间间隔始终保持500毫秒。 - 指数间隔重试：每次重试的时间间隔会根据算法随机取值，一般是随着重试次数增加，间隔时间呈指数级增长趋势。例如，第一次重试间隔可能是100毫秒，第二次可能是200毫秒，第三次可能是400毫秒。
重试间隔时间	设置重试的间隔时间。 - “重试策略”选择“固定间隔重试”，设置重试的固定间隔时间，取值范围为1 ~ 60,000之间的正整数，单位为（s/ms）。 - “重试策略”选择“指数间隔重试”，设置重试的基准时间，取值范围为1 ~ 60,000之间，单位为（s/ms）。

• 隔离仓


参数名称	参数说明
治理策略名称	输入治理策略的名称。支持输入任意字符，长度不超过32个字符。
业务场景	设置治理策略适用的业务场景： - 单击“选择业务环境”，选择已创建好的业务场景名称。 - 单击“创建业务场景”，具体操作请参考 创建业务场景 。
最大并发数	根据实际系统的实际业务处理能力，设置业务的最大并发数。
阻塞计时	设置阻塞计时时间，当请求超出最大并发数时，超出阻塞计时时间范围后，该请求会被丢弃。取值范围为1 ~ 300,000之间的正整数，单位为（s/ms）。

• 熔断

参数名称	参数说明
治理策略名称	输入治理策略的名称。支持输入任意字符，长度不超过32个字符。

参数名称		参数说明
业务场景		设置治理策略适用的业务场景： <ul style="list-style-type: none"> - 单击“选择业务环境”，选择已创建好的业务场景名称。 - 单击“创建业务场景”，具体操作请参考创建业务场景。
熔断范围	滑动窗口类型	选择滑动窗口类型。滑动窗口为一个范围，在这个范围内的调用超过调用次数基线时，才会根据熔断开启条件判断是否开启熔断。 <ul style="list-style-type: none"> - 时间：以时间为维度划定窗口范围。 - 请求数：以请求数量为维度划定窗口。
	滑动窗口大小	设置滑动窗口的大小。 <ul style="list-style-type: none"> - “滑动窗口类型”为“时间”：最近n秒/分钟时间范围内的调用会被记录和统计。 - “滑动窗口类型”为“基于计数”：最近n次的调用会被记录和统计。 其中，n为您设置的滑动窗口的大小。
	调用次数基线	设置调用次数基线，即开启统计调用错误率至少需要达到的调用数量。 <p>例如，设置“调用次数基线”为10，为统计错误率，则至少要记录10个调用。</p>
熔断开启条件	错误率阈值熔断	勾选“错误率阈值熔断”时生效，设置“错误率阈值”，即调用错误的百分比。 <p>当调用错误率不小于错误率阈值时，发生熔断，返回响应码429。</p>
	慢请求熔断	勾选“当慢请求达到您定义的比例时，开启熔断”时生效，设置以下参数： <ul style="list-style-type: none"> - 慢请求定义：慢请求阈值定义，响应时间超过该阈值的请求都是慢请求。 - 慢请求比例：慢请求阈值，达到指定的慢请求比例时，发生熔断，返回响应码429。

步骤6 单击“创建”，治理策略开始生效。

在治理策略列表，单击业务场景所在行的：

- 单击治理策略所在行“操作”列的“启用策略”，可启动已关闭的治理策略，使治理策略生效。
- 单击治理策略所在行“操作”列的“关闭策略”，可关闭已启动的治理策略。
- 单击治理策略所在行“操作”列的“编辑”，可对治理策略进行编辑。
- 单击治理策略所在行“操作”列的“删除”，可删除已关闭的治理策略。

----结束

5.7 微服务治理（适用 ServiceComb 引擎为 1.x 和 2.4.0+）

5.7.1 微服务治理概述

如果使用微服务框架开发应用，在应用托管后启动应用，微服务会自动注册到对应的 ServiceComb 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。

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请求流量、故障处理和负载均衡等的相关治理。

治理策略说明

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

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

名称	说明
黑白名单	<p>该服务治理策略只适用于Java Chassis开发框架接入的微服务。</p> <ul style="list-style-type: none"> 应用场景 基于公钥认证机制，ServiceComb引擎提供了黑白名单功能。通过黑白名单，可以控制微服务允许其他哪些服务访问。 Java Chassis开发框架接入的微服务治理 只有启用了公钥认证，设置的黑白名单才能生效，请参考为微服务设置黑白名单治理策略。

5.7.2 为微服务设置负载均衡治理策略

微服务一般会部署多个实例，负载均衡控制微服务消费者访问微服务提供者的多个实例的策略，以达到流量均衡的目的。策略包括轮询、随机、响应时间权值、会话粘滞等。

前提条件

已参考[创建微服务](#)完成微服务创建，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。

设置负载均衡

步骤1 单击待操作的引擎。

步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行[步骤4](#)。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行[步骤4](#)，其他用户请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“负载均衡”。

步骤5 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。
响应时间权值	<p>此配置适用于Java Chassis开发框架接入的微服务。</p> <p>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。</p>

策略名	策略说明
会话粘滞	<p>此配置适用于Java Chassis开发框架接入的微服务。</p> <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"> 会话保持时间：会话保持的限制时间，取值范围为0 ~ 86400，单位为秒。 失效次数阈值：访问失败次数，取值范围为0 ~ 10。当微服务访问下实例的失败次数或会话保持时间超过设定的值时，微服务不再访问该实例。

步骤6 单击“确定”保存配置。

----结束

5.7.3 为微服务设置限流治理策略

用于控制访问微服务的请求量大小，避免由于流量冲击对系统造成破坏。

前提条件

已参考[创建微服务](#)完成微服务创建，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。

设置限流

步骤1 单击待操作的引擎。

步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行[步骤4](#)。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行[步骤4](#)，其他用户请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“限流”。

步骤5 单击“新增”，限流配置项如下表所示。

配置项	说明
限流对象	<p>此配置适用于Java Chassis开发框架接入的微服务。</p> <p>设置的限流规则所作用的目标范围。</p>
上游微服务	<p>此配置适用于Spring Cloud开发框架接入的微服务。</p> <p>微服务架构中，各个微服务相互协作完成业务功能。从下拉选项框中选择调用当前微服务的其他微服务。</p>

配置项	说明
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。取值范围为1 ~ 99999。 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6 单击“确定”保存配置。

----结束

5.7.4 为微服务设置降级治理策略

用于控制微服务调用其他微服务的时候，强制返回缺省值或者抛出异常，而不将请求发送到目标微服务，以达到屏蔽对目标微服务的访问和降低其压力的目的。

前提条件

已参考[创建微服务](#)完成微服务创建，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。

设置降级

步骤1 单击待操作的引擎。


步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“降级”。

步骤5 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。 当Java Chassis开发框架接入的微服务时，则针对选定的微服务内的某一个或几个特定方法作为降级对象。
指定请求路径	此配置适用于Spring Cloud开发框架接入的微服务。 可单击  ，通过设置Method（如GET、POST、PUT等）、Path（请求路径）以及Headers（请求头）等参数，可以筛选出特定的请求集合，精准定位需要执行降级操作的请求。

配置项	配置项说明
降级策略	<ul style="list-style-type: none"> • 开启：开启降级，则会根据设置的参数，进行降级操作。 • 关闭：关闭降级，则不会发生降级治理。

步骤6 单击“确定”，保存配置。

----结束

5.7.5 为微服务设置容错治理策略

当微服务消费者访问提供者出现异常，比如实例网络不通等，需要将请求转发到其他可用的实例。这里的容错，常被称为重试。

前提条件

已参考[创建微服务](#)完成微服务创建，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。

设置容错

步骤1 单击待操作的引擎。

步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行[步骤4](#)。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行[步骤4](#)，其他用户请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“容错”。

步骤5 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
下游微服务	<p>此配置适用于Spring Cloud开发框架接入的微服务。</p> <p>设置本微服务调用下游微服务的容错规则，下拉菜单可直接选择下游微服务。</p>
容错对象	<p>此配置适用于Java Chassis开发框架接入的微服务。</p> <p>该应用依赖的应用或方法，下拉菜单可直接选择。</p>
是否开启容错	<p>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。</p> <p>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。</p>

配置项	配置项说明
容错策略	<p>当“是否开启容错”配置项设置为“开启”时需配置。</p> <p>Spring Cloud开发框架接入的微服务，需设置：</p> <ul style="list-style-type: none"> ● 尝试同一个微服务实例次数。 ● 尝试新的微服务实例次数。 <p>Java Chassis开发框架接入的微服务，需设置：</p> <ul style="list-style-type: none"> ● Failover 在不同服务器上重新尝试建立连接。 ● Failfast 不再重新尝试建立连接，即请求失败时会立即返回失败结果。 ● Failback 在同一个服务器上重新尝试建立连接。 ● custom <ul style="list-style-type: none"> - 尝试同一个服务器次数：尝试与同一个服务器重新建立连接的次数。 - 尝试新的服务器次数：尝试与新的服务器建立连接的次数。

步骤6 单击“确定”，保存配置。

----结束

5.7.6 为微服务设置熔断治理策略

当微服务消费者访问提供者出现异常，比如实例网络不通、请求超时等，并且异常积累到一定的程度，需要停止访问提供者，返回一个异常或者缺省值，防止雪崩效应。

前提条件

已参考[创建微服务](#)完成微服务创建，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。

设置熔断

步骤1 单击待操作的引擎。

步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“熔断”。

步骤5 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
下游微服务	此配置适用于Spring Cloud开发框架接入的微服务。 设置本微服务调用下游微服务的熔断规则，在下拉选项中选择下游微服务。
熔断对象	此配置适用于Java Chassis开发框架接入的微服务。 该应用调用的服务和方法，下拉菜单可直接选择。
指定请求路径	此配置适用于Spring Cloud开发框架接入的微服务。 可单击  ，通过设置Method（如GET、POST、PUT等）、Path（请求路径）以及Headers（请求头）等参数，可以筛选出特定的请求集合，精准定位需要执行熔断操作的请求。
熔断时间窗	熔断的持续时间，该时间窗内不再响应请求。单位为ms。
失败率	在设定的窗口请求数内，请求失败的比例。例如设置窗口请求数为20，失败率为50%，当这20个请求中有10个（50%）失败时，就会触发熔断条件。
窗口请求数	触发条件，窗口收到的请求数。“失败率”和“窗口请求数”的条件需同时满足才会触发熔断。

步骤6 单击“确定”，保存配置。

----结束

5.7.7 为微服务设置错误注入治理策略

错误注入可以模拟一个调用失败，主要用于功能验证、故障场景演示等场景。**该服务治理策略只适用于Java Chassis开发框架接入的微服务。**

前提条件

已参考[创建微服务](#)完成微服务创建，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。

设置错误注入

步骤1 单击待操作的引擎。

步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“错误注入”。

步骤5 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"> • 延迟 • 错误
协议	访问微服务出现延时或错误的协议： <ul style="list-style-type: none"> • Rest • Highway
触发概率	访问微服务出现延时或错误的概率。
延迟时间	访问微服务出现延时的时长，“类型”选择为“延迟”时才需要配置。
http错误码	访问微服务出现错误的HTTP错误码，“类型”选择为“错误”时才需要配置。该错误码为HTTP标准的错误码。

步骤6 单击“确定”，保存配置。

----结束

5.7.8 为微服务设置黑白名单治理策略

基于公钥认证机制，ServiceComb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。**该服务治理策略只适用于Java Chassis开发框架接入的微服务。**

前提条件

- 已参考[创建微服务](#)完成微服务创建，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。
- 微服务已启用了公钥认证，设置的黑白名单才能生效，请参考[为微服务设置公钥认证](#)。

设置黑白名单

步骤1 单击待操作的引擎。

步骤2 选择“微服务治理”。

- 未开启安全认证的引擎，请执行[步骤4](#)。
- 开启安全认证的引擎，当登录控制台的用户为[导入IAM账号](#)中导入的用户，请执行[步骤4](#)，其他用户请执行[步骤3](#)。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击需要治理的微服务。单击“黑白名单”。

步骤5 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"> ● 黑名单：表示根据匹配规则匹配到的微服务都不允许访问当前服务。 ● 白名单：表示根据匹配规则匹配到的微服务允许访问当前服务。
匹配规则	使用正则表达式表示。 例如“匹配规则”设置为data*，指在黑名单下匹配到的名称以data开头的服务不允许访问当前服务，或者在白名单下匹配到的名称以data开头的服务允许访问当前服务。

步骤6 单击“确定”，保存配置。

----结束

5.7.9 为微服务设置公钥认证

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

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

启用公钥认证步骤如下：

1. 公钥认证需要在消费者、提供者都启用。

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

2. 在pom.xml中增加依赖：

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

5.8 配置管理（适用于 ServiceComb 引擎 2.x 版本）

5.8.1 创建 2.x 版本 ServiceComb 引擎配置

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

约束与限制

- 配置项使用明文存储，请勿附带敏感数据。
- 当引擎规格限定的配置条目配额即将用完时，为了保证尽量可用，引擎允许超出剩余配额的新建配置项操作同时成功。请尽快扩容引擎，避免后续新建配置失败。
- 对于编辑或删除配置项，若配置项已被微服务使用，可能导致微服务读取不到配置或读取错误配置，引发业务异常。因此在操作修改或删除配置项前请备份配置。

创建应用级配置

将新建的配置关联到某一应用，并添加应用名称和所在环境标签。

📖 说明

配置项使用明文存储，请勿附带敏感数据。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“新建配置”，参考下表设置配置参数。参数前面带*号的是必须设置的参数。

参数名称	参数说明
*配置项	输入配置项。 配置项为该配置的全局ID，在编码阶段通过配置项对配置进行索引及操作。建议采用类Java包命名方式（如cse.service.registry.address）的命名规则来保证配置项的可读性和唯一性。 创建应用级配置时不允许创建以“servicecomb.matchGroup.”开头的配置项，此开头的配置会与创建业务场景治理生成的配置冲突，导致业务场景无法显示。
配置范围	选择“应用级配置”。
*应用	1. 选择或输入应用名称。 2. 选择环境。
配置格式	选择配置格式，支持TEXT、YAML、JSON、Properties、INI、XML六种常见配置格式在线编辑。默认为：TEXT。
*配置内容	输入配置内容。
是否启用	选择是否启用配置： <ul style="list-style-type: none"> • 现在启用：创建成功后，配置立即生效。 • 暂不启用：创建成功后，配置暂不生效。

步骤5 单击“立即创建”，启用该配置项。

----结束

创建微服务级配置

将新建的配置关联到某一微服务，并添加微服务名称、应用名称和环境。

说明

配置项使用明文存储，请勿附带敏感数据。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“新建配置”，参考下表设置配置参数。参数前面带*号的是必须设置的参数。

参数名称	参数说明
*配置项	输入配置项。 配置项为该配置的全局ID，在编码阶段通过配置项对配置进行索引及操作。建议采用类Java包命名方式（如cse.service.registry.address）的命名规则来保证配置项的可读性和唯一性。
配置范围	选择“微服务级配置”。
*微服务	1. 选择或者输入微服务名称。 2. 选择或输入应用名称。 3. 选择环境。
配置格式	选择配置格式，支持TEXT、YAML、JSON、Properties、INI、XML六种常见配置格式在线编辑。默认为：TEXT。
*配置内容	输入配置内容。
是否启用	选择是否启用配置： <ul style="list-style-type: none">• 现在启用：创建成功后，配置立即生效。• 暂不启用：创建成功后，配置暂不生效。

步骤5 单击“立即创建”，启用该配置项。

----结束

创建自定义配置

当应用级配置、微服务级配置不能满足使用需求时，可自定义配置文件。

📖 说明

配置项使用明文存储，请勿附带敏感数据。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“新建配置”，参考下表设置配置参数。参数前面带*号的是必须设置的参数。

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

步骤5 单击“立即创建”，启用该配置项。

----结束

5.8.2 管理 2.x 版本 ServiceComb 引擎配置

编辑配置

ServiceComb引擎支持根据业务需要，编辑配置信息，可在不重启服务的前提下，动态更新配置。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击待编辑的配置项右侧“操作”列的“编辑”，也可单击待编辑的配置项名称，在配置详情页面，单击“编辑”。

步骤5 在“配置内容”输入框输入配置信息，单击“保存”编辑配置项完成。

----结束

导入配置

ServiceComb引擎支持在控制台导入本地配置文件，格式为json文件，减少手动创建配置的工作量。当新建的开发、测试、预发环境需要复用生产环境的配置，或服务Comb引擎需要进行跨云或跨平台迁移时，可预先将生产环境的配置导出，再将配置文件导入目标引擎，批量完成配置初始化，缩短环境搭建；当配置被误删、篡改、ServiceComb引擎故障导致配置丢失，可通过导入备份配置文件，快速重建服务配置环境。

说明

- 当配置文件导入成功后，实际导入到环境的配置信息取决于在操作时设置的对相同配置的处理策略，如终止导入、跳过和覆盖。
- 导入前需备份目标环境现有配置，优先增量导入，核心服务配置需小范围验证后再批量导入。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击右上角的“导入”，根据需要导入不同文件格式，参考下表设置导入参数。

表 5-6 导入文件格式为 V2.0

参数名称	参数说明
文件格式	可选择导入文件的格式，默认是V2.0。
导入至特定环境	<ul style="list-style-type: none"> 关闭：导入的配置不改变环境标签。 开启：把配置导入至特定环境，将会改变环境标签。在下拉列表中选择特定环境。
相同配置	<ul style="list-style-type: none"> 终止导入：导入过程中，如果遇到和系统中相同的配置，导入终止。 跳过：导入过程中，如果遇到和系统中相同的配置，该配置会被跳过，继续导入其余配置。 覆盖：导入过程中，如果遇到和系统中相同的配置，该配置的值会被替换。

参数名称	参数说明
配置文件	单击“导入文件”，选择待导入配置文件。 只支持导入json格式的配置文件且文件大小不超过2MB。

表 5-7 导入文件格式为 V1.0

参数名称	参数说明
文件格式	可选择导入文件的格式，选择文件格式为V1.0。
*导入至特定环境	在下拉列表中选择微服务环境。
微服务名称	在下拉列表中选择将配置导入到的微服务。
微服务版本	在下拉列表中选择将配置导入到的微服务的版本。
*配置文件	单击“导入文件”，选择待导入配置文件。 只支持导入json格式的配置文件且文件大小不超过2MB。

步骤5 单击“关闭”，完成导入。

----结束

导出配置

ServiceComb引擎支持导出选定的配置文件到本地，防止配置中心数据丢失（如服务器故障、人为误删除），定期导出配置作为恢复依据。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 勾选待导出的配置项，单击“导出”。

- 单击配置项上方的“导出”，在弹出框中单击“导出”，配置文件导出成功。
- 单击右上方的“导出”，在“导出配置”弹出框中，选择导出配置的文件格式，默认为V2.0，单击“确定”，配置文件导出成功。

📖 说明

当选择导出配置的文件格式为V1.0时，需在下拉框中选择微服务环境。

----结束

查看配置详情

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击待查看的配置项，进入配置项详情页面，可查看配置详细信息。

----结束

查看历史版本配置

可查看不同历史版本的配置。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击配置项右侧“操作”列的“查看历史版本”，也可在配置详情页面，单击“历史版本”，进入“历史版本”页面可查看该配置项的历史版本信息，在该页面也可对比配置版本和回滚版本操作。

----结束

对比配置版本

提供了不同历史版本之间差异比对。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击待对比的配置项名称。

步骤5 单击“历史版本”。

步骤6 在左侧“历史版本”列表选择待查看的历史版本。在“历史版本”列表中最多可显示**100个历史版本**。

在右侧“配置文件”列，可查看历史版本和当前版本的差异。

----结束

回滚版本

从当前版本回滚到选择的历史版本。

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击待操作的配置项名称。

步骤5 单击“历史版本”。

步骤6 在左侧“历史版本”列表，选择待回滚到的历史版本。

步骤7 在右侧“配置文件”列，单击“回滚到此版本”。

----结束

禁用配置项

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 在待禁用的配置项右侧“操作”列选择“更多 > 禁用”。

步骤5 在弹出框中单击“确定”禁用该配置项。

----结束

删除配置项

步骤1 单击待操作的引擎名称。

步骤2 选择“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 在待删除的配置项右侧“操作”列选择“更多 > 删除”，也可单击待删除的配置项名称，在配置详情页面，单击“删除”。

步骤5 在弹出框中单击“确定”删除该配置项。

----结束

5.9 配置管理（适用于 ServiceComb 引擎 1.x 版本）

5.9.1 创建 1.x 版本 ServiceComb 引擎配置

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

创建配置

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

说明

配置项使用明文存储，请勿附带敏感数据。

步骤1 单击待操作的引擎名称。

步骤2 单击“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“创建配置”。

步骤5 在“创建配置”页面，选择微服务环境，并输入“配置项”和“值”。

步骤6 单击“确定”完成配置添加。

----结束

5.9.2 管理 1.x 版本 ServiceComb 引擎配置

编辑配置

步骤1 单击待操作的引擎名称。

步骤2 单击“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击待操作配置项“操作”列的“编辑”，编辑配置项的值。

步骤5 单击“确定”，配置项修改。

----结束

导入配置

步骤1 单击待操作的引擎名称。

步骤2 单击“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“导入”。

步骤5 选择微服务环境，单击“导入文件”，选择待导入配置文件。每次导入的配置项条目数量最多为150条。

步骤6 单击“关闭”完成配置文件导入。

----结束

导出配置

步骤1 单击待操作的引擎名称。

步骤2 单击“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 单击“全部导出”，配置文件导出成功。

----结束

删除配置

步骤1 单击待操作的引擎名称。

步骤2 单击“配置管理”。

- 未开启安全认证的引擎，请执行**步骤4**。
- 开启安全认证的引擎，当登录控制台的用户为**导入IAM账号**中导入的用户，请执行**步骤4**，其他用户请执行**步骤3**。

步骤3 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。

步骤4 选中待操作的配置项，单击“删除”，也可单击待操作配置项“操作”列的“删除”。

步骤5 在弹出框单击“确定”，删除配置项。

----结束

5.10 系统管理

5.10.1 系统管理概述

同一个ServiceComb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ServiceComb引擎访问和操作权限。

开启了“安全认证”的ServiceComb引擎专享版，通过微服务控制台提供了基于RBAC（Role-Based Access Control，基于角色的访问控制）的系统管理功能。

开启了“安全认证”的ServiceComb引擎专享版，支持Spring Cloud、Java Chassis微服务框架正常接入。

说明

- 基于RBAC的系统管理功能与IAM权限管理无关，仅是CSE内部的权限管理机制。
 - 如果您通过微服务引擎控制台操作ServiceComb引擎，必须同时具备IAM和RBAC的操作权限，且IAM权限优先级要高于RBAC权限。
 - 如果您通过API接口或者微服务框架操作ServiceComb引擎，则只需具备RBAC相关权限。
1. 您可以使用关联了admin角色权限的账号创建新账号，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ServiceComb引擎的相应的访问和操作权限。
 - 创建开启了“安全认证”的ServiceComb引擎专享版时，系统自动创建1个关联了admin角色权限的root账号。不能编辑、删除root账号。
 - 您可以使用创建该ServiceComb引擎的root账号或者该ServiceComb引擎下关联了admin角色权限的账号创建新账号。创建和管理账号，请参考[账号管理](#)。
 2. 您可以使用关联了admin角色权限的账号创建自定义角色，根据业务需求把合适的ServiceComb引擎访问和操作权限赋予该角色。
 - 系统默认内置两种角色：管理员（admin）、开发者（developer）。不能编辑、删除内置角色。
 - 您可以使用创建该ServiceComb引擎的root账号或者该ServiceComb引擎下关联了admin角色权限的账号创建自定义角色。创建和管理角色，请参考[角色管理](#)。
 - 角色权限说明，请参见[表5-8](#)。

表 5-8 角色权限说明

角色	权限说明
admin	具有该ServiceComb引擎下所有微服务、账号和角色的所有操作权限。
developer	具有该ServiceComb引擎下所有微服务的所有操作权限。
自定义角色	根据实际业务需求创建角色，给角色分配相应微服务、配置的操作权限。

5.10.2 账号管理

您可以使用ServiceComb引擎下关联了admin角色权限的账号登录该微服务引擎控制台，根据实际业务需求创建新账号或对该引擎下已创建的指定账号进行管理操作。

表 5-9 账号管理操作说明

操作	说明
新增账号	根据实际业务需求创建新账号，把合适的角色同账号关联，使用该账号的用户则具有对该ServiceComb引擎的相应的访问和操作权限。 最多可创建1000个账号，包括新增账号和导入的IAM账号。
导入IAM账号	CSE支持导入IAM账号，把合适的角色同IAM账号关联，使用该IAM账号的用户则具有对该微服务引擎的相应的访问和操作权限。 导入的IAM账号如果要通过编程接口安全认证将微服务应用接入引擎，需要先将导入的IAM账号 重置密码 ，再用新密码配置安全认证参数。 使用该IAM账号登录CSE控制台，进入开启了安全认证的微服务引擎控制台时，不需要输入账号密码。 CSE支持管理1000个账号，包括新增账号和导入的IAM账号。
查看账号角色权限	查看指定账号关联的角色权限配置。
编辑账号	根据实际业务需求，对账号进行增加、删除角色操作。不能编辑root账号。
修改密码	根据业务需求或者安全规定，可修改已登录ServiceComb引擎的账号密码。 说明 <ul style="list-style-type: none"> 登录ServiceComb引擎的用户为admin角色时，允许修改自身及非admin角色用户的密码，且需要输入原密码。 登录ServiceComb引擎的用户为非admin角色时，只允许修改自身的密码，且需要输入原密码。 若在SDK中使用了该账号密码注册微服务，修改账号密码可能会影响使用该微服务业务运行（无法注册到ServiceComb引擎），将导致业务系统受损，请谨慎操作。 密码修改后，需及时更新微服务认证配置。 密码修改后，可能会因密码错误超过3次导致账号锁定，需要等待15分钟锁定状态才会释放。
重置密码	根据业务需求或者安全规定，可使用已登录ServiceComb引擎的账号对该ServiceComb引擎下的其他账号密码进行重置。 <ul style="list-style-type: none"> 若在SDK中使用了该账号密码注册微服务，重置账号密码可能会影响使用该微服务业务运行（无法注册到ServiceComb引擎），将导致业务系统受损，请谨慎操作。 密码重置后，需及时更新微服务认证配置。 密码重置后，可能会因密码错误超过3次导致账号锁定，需要等待15分钟锁定状态才会释放。
删除账号	根据实际业务需求，删除不再使用的账号。不能删除root账号。 若在SDK中使用了该账号密码注册服务，删除账号会影响该服务业务运行（无法注册到引擎），将导致业务系统受损，请谨慎操作。

新增账号

创建新账号前，可先根据实际业务需要[创建角色](#)。

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”，在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的账号名及其密码，单击“确定”。

📖 说明

首次连接ServiceComb引擎，请输入root账号名及[创建ServiceComb引擎](#)时输入的密码。

步骤3 在“账号管理”页签，单击“新增账号”，参考下表设置账号参数。

参数名称	参数说明
账号名称	输入新账号名称。 账号创建后，账号名称不可修改。
账号角色	根据实际业务需求，选择账号角色。 一个账号下最多可关联5个角色。
密码	输入账号密码。
确认密码	再次输入账号密码。

步骤4 单击“确定”，完成新账号创建。

----结束

导入 IAM 账号

导入IAM账号前，可先根据实际业务需要[创建角色](#)。

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。

步骤3 选择“账号管理 > 导入IAM账号”。

步骤4 勾选待导入的IAM账号名称，根据实际业务需求，选择账号角色。一个账号下最多可关联5个角色。

步骤5 单击“确认导入”。

被导入的账号不支持通过密码登录。如想使用导入的iam账号通过编程接口安全认证将微服务应用接入引擎，请先进行[重置密码](#)。

----结束

查看账号角色权限

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

- 步骤2** 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。
- 步骤3** 单击账号列表中待查看账号“角色”列的角色名称，在弹出页面可以查看账号关联的角色名称及权限配置情况。
- 结束

编辑账号

- 步骤1** 单击待操作的开启了安全认证的ServiceComb引擎。
- 步骤2** 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。
- 步骤3** 在“账号管理”页签，单击待编辑账号“操作”列的“编辑账号”。
- 步骤4** 根据实际业务需求，选择“账号角色”。一个账号下最多可关联5个角色。
- 步骤5** 单击“保存”，完成账号编辑。
- 结束

修改密码

- 步骤1** 单击待操作的开启了安全认证的ServiceComb引擎。
- 步骤2** 单击“系统管理”。
- 步骤3** 在弹出的“安全认证”对话框输入账号名及其密码，单击“确定”。
- 步骤4** 在“账号管理”页签，选择登录该ServiceComb引擎的账号名，单击“操作”列的“修改密码”。
1. 输入“原密码”、“新密码”和“确认密码”。
 2. 查看提示信息确认需要修改密码后，勾选“我已确认知晓”。

说明

也可单击“系统管理”页面右上角的“修改密码”，修改当前登录账号的密码。

- 步骤5** 单击“保存”，完成密码修改。
- 结束

重置密码

- 步骤1** 单击待操作的开启了安全认证的ServiceComb引擎。
- 步骤2** 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。
- 步骤3** 在“账号管理”页签，选择待重置密码的账号名，单击“操作”列的“重置密码”。
1. 输入“新密码”和“确认密码”。
 2. 查看提示信息确认需要重置密码后，勾选“我已确认知晓”。

步骤4 单击“保存”，完成密码重置。

----结束

删除账号

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。

步骤3 在“账号管理”页签，单击待删除账号“操作”列的“删除”。

步骤4 在弹出的对话框中输入“DELETE”，单击“确定”。

----结束

5.10.3 角色管理

除了系统中默认提供的两种角色：管理员（admin）和开发者（developer）无法进行操作外，您可以使用该ServiceComb引擎下关联了admin角色权限的“账号”登录微服务引擎控制台，根据实际的业务需求对角色进行相关操作。

创建角色

根据实际业务需求创建新角色，设置该角色在不同服务组和配置组对应的权限动作。

最多可创建100个角色。

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。

步骤3 在“角色管理”页签，单击“创建角色”。

步骤4 输入新角色名称。角色创建后，角色名称不可修改。



步骤5 对权限进行配置。

1. 设置“权限组”。

a. 设置服务权限。

- 选择“全部服务”。
可以对该ServiceComb引擎的所有微服务资源实施相应的权限动作。
- 选择“自定义服务组”，可按照[表5-10](#)进行设置。



表 5-10 自定义服务组操作

操作名称	操作说明
新增匹配规则	<p>单击“添加服务组匹配规则”，根据实际业务需要，选择“应用”、“环境”、“服务”三个参数值来匹配规则过滤该角色可以实施权限动作的微服务。</p> <ul style="list-style-type: none"> 自定义服务组最多可以同时设置20条微服务匹配规则。 当自定义服务组设置了多条匹配规则时，只要微服务满足其中任意一条匹配规则，角色就对该微服务有操作权限。 <p>应用名称、环境名称和服务名称是微服务的三个参数：</p> <ul style="list-style-type: none"> 如果单条匹配规则只设置了一个参数，则角色对与该参数值相匹配的微服务有操作权限。 例如添加匹配规则“环境:production”，那么该角色只对环境名称是“production”的微服务有操作权限。 如果单条匹配规则设置了多于一个参数，则角色对与所有参数值都相匹配的微服务有操作权限。 例如添加匹配规则“环境:production”“应用:abc”，那么该角色对环境名称是“production”且应用名称是“abc”的微服务有操作权限。 在启用自动发现的情况下，微服务会通过注册中心查询注册中心、配置中心、仪表盘等服务的实例地址。对微服务授予查询权限时，需要包含应用default的权限，添加匹配规则“应用:default”。 <p>设置微服务匹配规则后，单击“确定”。</p>
编辑匹配规则	<p>单击待编辑的匹配规则后的“”，即可根据实际业务需要，重新设置该条匹配规则的“服务组”和“权限动作”。</p> <p>重新设置服务组匹配规则后，单击“确定”。</p>
删除匹配规则	<p>单击待删除的匹配规则后的“”，即可根据实际业务需要，删除该条服务组匹配规则。</p>

b. 设置配置权限。

- 选择“全部配置”。
可以对该ServiceComb引擎的所有配置资源实施相应的权限动作。
- 选择“自定义配置组”，可按照[表5-11](#)进行设置。

表 5-11 自定义配置组操作

操作名称	操作说明
新增匹配规则	<p>单击“添加配置组匹配规则”，根据实际业务需要，选择“应用”、“环境”、“服务”三个参数值来匹配规则过滤该角色可以实施权限动作的配置。当应用级配置、微服务级配置不能满足使用需求时，也可以通过增加自定义匹配规则来匹配配置的自定义标签过滤该角色可以实施权限动作的配置。</p> <ul style="list-style-type: none"> 自定义配置组最多可以同时设置20条配置匹配规则。 当自定义配置组设置了多条匹配规则时，只要配置满足其中任意一条匹配规则，角色就对该配置有操作权限。 <p>应用名称、环境名称和服务名称是配置的三个参数：</p> <ul style="list-style-type: none"> 如果单条匹配规则只设置了一个参数，则角色对与该参数值相匹配的配置有操作权限。 例如添加匹配规则“环境:production”，那么该角色只对环境名称是“production”的配置有操作权限。 如果单条匹配规则设置了多于一个参数，则角色对与所有参数值都相匹配的配置有操作权限。 例如添加匹配规则“环境:production”“应用:abc”，那么该角色对环境名称是“production”且应用名称是“abc”的配置有操作权限。 <p>设置配置匹配规则后，单击“确定”。</p>
编辑匹配规则	<p>单击待编辑的匹配规则后的“”，即可根据实际业务需要，重新设置该条匹配规则的配置组和“权限动作”。重新设置配置组匹配规则后，单击“确定”。</p>
删除匹配规则	<p>单击待删除的匹配规则后的“”，即可根据实际业务需要，删除该条配置组匹配规则。</p>

2. 设置“权限动作”。

按照实际的业务需求，设置角色对选择的服务组、配置组可实施的权限动作，可以勾选多个。

- 全选：对该服务组、配置组资源可实施增加、删除、修改和查询四种动作。
- 增加：对该服务组、配置组资源可实施增加的动作。
- 删除：对该服务组、配置组资源可实施删除的动作。

如果只勾选了删除，将无法通过界面对服务组、配置组资源进行删除，必须同时勾选查询。

- 修改：对该服务组资源可实施修改的动作。

如果只勾选了修改，将无法通过界面对服务组、配置组资源进行修改操作，必须同时勾选查询。

- 查询：对该服务组、配置组资源可实施查询的动作。

步骤6 单击“创建”，创建角色。

----结束

编辑角色

根据实际业务需求，修改已创建角色的权限配置。

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。

步骤3 在“角色管理”页签，单击待编辑角色“操作”列的“编辑”。

步骤4 根据实际业务需求，修改服务组、配置组和“权限动作”。

步骤5 单击“保存”，完成角色编辑。

----结束

删除角色

根据实际业务需求，删除不再使用的角色。

- 角色删除后无法恢复，请谨慎操作。
- 删除角色前要先确认该角色没有被账号关联。取消角色同账号之间的关联，请参考[编辑账号](#)。

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。

步骤3 在“角色管理”页签，单击待删除角色“操作”列的“删除”。在弹出的对话框中输入“DELETE”，单击“确定”。

- 角色删除后无法恢复，请谨慎操作。
- 删除角色前要先确认该角色没有被账号关联。取消角色同账号之间的关联，请参考[编辑账号](#)。


----结束

查看角色

可按照角色名称的关键字查看该ServiceComb引擎下已创建的角色。

步骤1 单击待操作的开启了安全认证的ServiceComb引擎。

步骤2 选择“系统管理”。在弹出的“安全认证”对话框输入该ServiceComb引擎下关联了admin角色权限的“账号名称”及其“密码”，单击“确定”。

步骤3 在“角色管理”页签，单击待查看角色名称前的，展开待查看角色的详细信息。

展开后可以查看角色的服务组、配置组和“权限动作”。

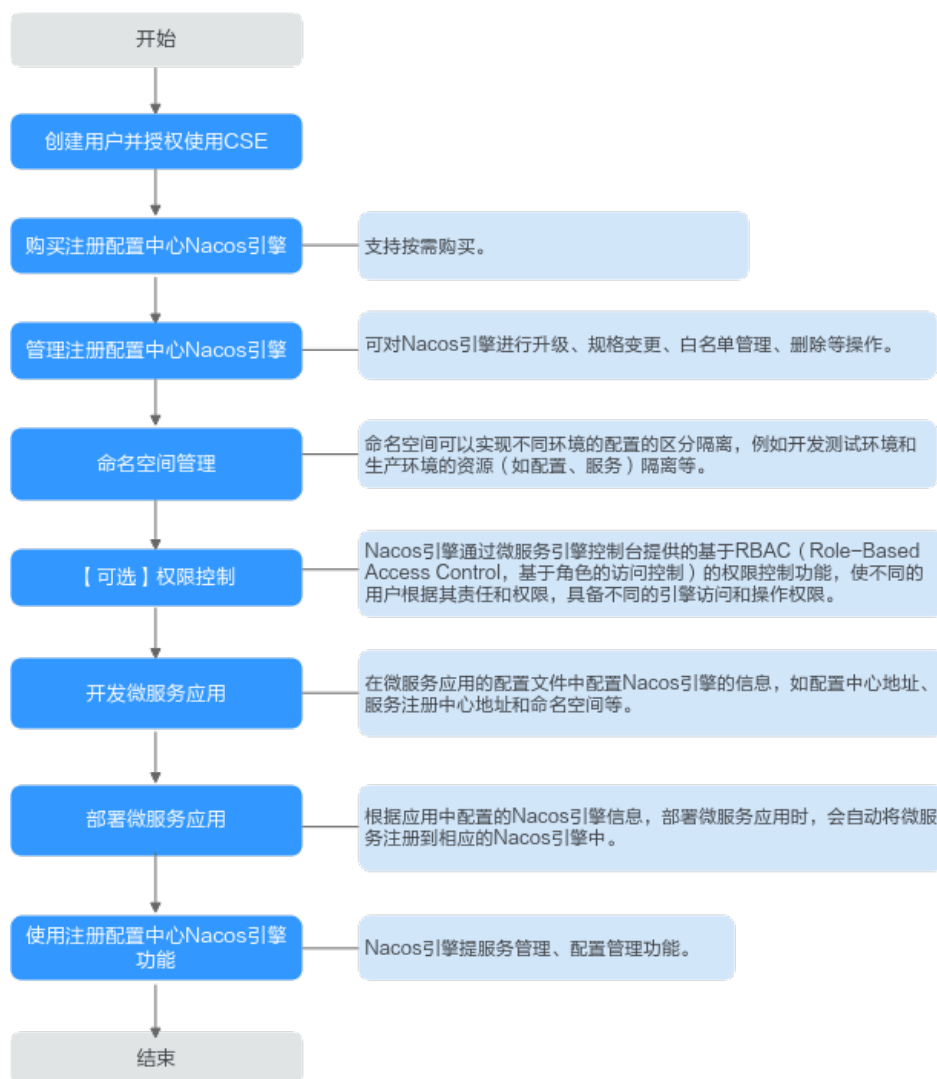
----结束

6 Nacos 引擎

6.1 Nacos 引擎概述

CSE注册配置中心Nacos兼容开源Nacos、Eureka客户端，具备注册发现、动态配置管理、访问权限控制、可观测等能力。可打造高可用、易管理的微服务中间件。Nacos是微服务架构中服务治理和配置管理的基础设施，尤其适合需要动态调整、高可用、多环境适配的复杂分布式系统。

注册配置中心Nacos的业务使用流程：



1. [创建用户并授权使用CSE](#)。
2. [创建注册配置中心](#)。
3. 参考[管理Nacos引擎](#)对Nacos引擎进行升级、规格变更、备份与恢复和删除等操作。
4. [命名空间管理](#)，实现不同环境的配置的区分隔离。
5. [【可选】权限控制](#)，实现不同用户根据责任和权限，对不同引擎具备访问和操作权限。
6. 开发微服务应用，在开发的微服务配置文件中，配置Nacos引擎的相关信息，如配置中心地址、服务注册中心地址和命名空间等。
7. 完成微服务部署，当微服务启动时会自动注册到Nacos引擎中。
8. 参考[管理Nacos引擎服务](#)~[管理Nacos引擎配置](#)，使用Nacos引擎提供的服务管理和配置管理功能。

6.2 创建 Nacos 引擎

注册配置中心Nacos引擎，核心是解决微服务架构中注册发现和配置管理的两大痛点，同时提供轻量化的服务治理的能力。控制台支持一键式创建CSE Nacos引擎，免去运维

成本，您只需关注业务本身。本章节指导您根据实际业务需求创建注册配置中心类型为Nacos的引擎。

约束与限制

- 注册配置中心集群的节点会尽可能均分到不同的可用区中，单节点故障不影响对外业务功能。
- 创建Nacos引擎时，不支持使用共享VPC，否则会创建引擎失败。
- Nacos引擎创建完成后，不支持变更虚拟私有云。
- 按需购买引擎时，可能会由于底层资源不足而导致创建失败，请及时删除引擎，避免当底层资源满足时，引擎恢复可用，从而导致扣费。
- 每个项目默认最多可以创建5个ServiceComb引擎。ServiceComb引擎同Nacos引擎共享配额数，最多不超过5个。

前提条件

- 当前登录账号拥有创建微服务引擎的权限。账号权限授权与绑定，请参考[创建用户并授权使用CSE](#)。
- Nacos运行于虚拟私有云，创建前，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《虚拟私有云用户指南》中“创建虚拟私有云和子网”章节。
- 创建Nacos需要保证用户具有CSE FullAccess、DNS FullAccess权限。




创建注册配置中心

步骤1 登录微服务引擎控制台。

步骤2 在左侧导航栏选择“注册配置中心”，单击“购买注册配置中心”。

步骤3 参考下表设置参数，参数前面带*号的是必须设置的参数。

参数	说明
*计费模式	选择计费方式，目前支持“按需计费”。按需计费是一种先使用再付费的计费模式，适用于无需任何预付款或长期承诺的用户。
*企业项目	选择Nacos所在的项目，可在下拉框中搜索和选择需要的企业项目。 企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。 参考，创建并启用新的企业项目后可以使用，默认选择default。 注册配置中心Nacos引擎创建完成后，可将Nacos引擎资源迁出当前企业项目，迁入到新的企业项目，具体操作请参考《企业管理用户指南》的“迁出企业项目资源”章节和《企业管理用户指南》的“为企业项目迁入资源”章节。
*引擎名称	输入Nacos引擎的名称，名称以字母开头，由字母、数字和-组成，且不能以-结尾，长度为3~24个字符。名称不允许命名为“default”。
*注册配置中心类型	选择注册配置中心类型“Nacos”。

参数	说明
*选择实例数	选择需要的容量规格。
版本	只能创建最新版本。
*网络	<p>为引擎选择虚拟私有云及子网，可以为您的引擎构建隔离的、自主配置和管理的虚拟网络环境。</p> <ul style="list-style-type: none"> 使用已创建的VPC，可在下拉列表中搜索和选择当前账号已创建的虚拟私有云和子网。 使用新的VPC，可单击“控制台”进入“虚拟私有云”页面，创建新的VPC，具体操作请参考《虚拟私有云用户指南》中“创建虚拟私有云和子网”章节。
标签	<p>用于标识云资源，当您拥有相同类型的许多云资源时，可以使用标签按各种维度（例如用途、所有者或环境）对云资源进行分类。</p> <p>您可以单击“ 添加标签”，在“添加标签”弹框输入标签键和标签值，添加标签，标签的命名规则请参见管理标签。在“添加标签”弹框，可单击“ 新增标签”同时添加多个标签，也可单击标签后的，删除该标签。</p>

步骤4 单击“立即购买”，进入引擎信息确认界面。

步骤5 单击“提交”，引擎开始创建，当“运行状态”为“可用”时，引擎创建完成。

----结束

6.3 管理 Nacos 引擎

6.3.1 查看 Nacos 引擎详细信息

本章节介绍如何在微服务引擎控制台查看Nacos引擎的详细信息。



查看 Nacos 引擎详细信息

步骤1 在左侧导航栏选择“注册配置中心”。

步骤2 单击待查看的Nacos实例名称。只有引擎运行状态为“可用”才可单击实例名称进入详细信息页面。

步骤3 查看如[表6-1](#)所示的Nacos引擎信息。

表 6-1 引擎详细信息说明

信息类型	参数	说明
基础信息	名称	创建Nacos引擎时输入的引擎名称，单击  可复制。
	ID	引擎的ID，单击  可复制。

信息类型	参数	说明
	运行状态	引擎的状态。
	注册配置中心类型	注册配置中心的类型为Nacos。
	版本	引擎的版本号。
	容量规格	创建Nacos引擎 时选择的引擎规格。当为小规格引擎时可单击右侧“扩容”进行扩容操作，具体操作请参考 扩容Nacos引擎 。
	企业项目	创建Nacos引擎 时选择的企业项目。
	计费方式	创建Nacos引擎 时选择的计费方式。当前只支持按需计费。
	创建时间	创建Nacos引擎 的时间。
连接信息	内网地址	Nacos引擎的内网地址。
	内网端口	Nacos引擎的内网端口。
	虚拟私有云	创建注册配置中心 时选择的虚拟私有云。
	子网	创建注册配置中心 时选择的子网。
	访问白名单	Nacos支持白名单控制。允许配置多个IP地址/地址段，不在地址段范围内的ip不允许访问。白名单访问的操作请参考 管理Nacos引擎白名单 。
更多设置	标签	显示为Nacos引擎添加的标签，也可单击“标签管理”，根据需要可对标签进行相应操作，具体操作可参见 管理Nacos引擎标签 。

----结束

6.3.2 管理 Nacos 引擎标签

标签是Nacos引擎的标识，为Nacos引擎添加标签，可以方便用户识别和管理拥有的Nacos引擎资源。

如您的组织已经设定Nacos引擎的相关标签策略，则需按照标签策略规则为引擎添加标签。标签不符合标签策略的规则，则可能会导致引擎创建失败，请联系组织管理员了解标签策略详情。

您可以在创建Nacos引擎时添加标签，也可以在Nacos引擎创建完成后，在引擎的详情页添加标签，您最多可以给引擎添加20个标签。另外，您还可以进行修改和删除标签。

标签共由两部分组成：标签键和标签值，其中，标签键和标签值的命名规则如表6-2所示。

表 6-2 标签命名规则

标签信息	规则
标签键	<ul style="list-style-type: none"> 不能为空。 对于同一个实例，Key值唯一。 长度不超过128个字符。 只能包含任意语种字母、数字、空格和特殊字符（_ . : = + - @ ）。 不能以空格开头和结尾，且不能以_sys_开头。
标签值	<ul style="list-style-type: none"> 长度不超过255个字符。 只能包含任意语种字母、数字、空格和特殊字符（_ . : = + - @ ）。

约束与限制

新增和修改标签会影响Nacos引擎业务十秒左右，请在业务低峰期时增加或修改。

管理标签

步骤1 单击待操作的引擎的名称，进入该引擎的详细信息页面。

步骤2 在“更多设置”区域的“标签”参数处，您可以根据实际需要，执行以下操作：

- 新增标签
 - 单击“标签管理”，弹出“编辑标签”窗口。
 - 单击“⊕ 新增标签”，您可在输入框分别输入标签键和标签值。
 - 单击“确定”，为实例添加标签成功。
- 修改标签
 - 单击“标签管理”，弹出“编辑标签”窗口。
 - 您可在原有的输入框中修改标签键与标签值信息。
 - 单击“确定”，标签修改成功。
- 删除标签

单击标签所在行的⊖，在弹出的确认框中单击“确定”，删除该标签。

---结束

6.3.3 管理 Nacos 引擎白名单

本章节主要介绍如何管理Nacos引擎的白名单，如果需要指定的IP地址才能访问Nacos引擎，您需要将指定的IP地址加入到白名单中。


如果引擎没有添加任何白名单或停用白名单功能，所有与引擎所在VPC互通的IP地址都可以访问该引擎。

设置白名单

步骤1 单击待查看的Nacos引擎名称，进入Nacos引擎详细信息页面。

📖 说明

只有引擎运行状态为“可用”才可单击引擎名称进入详细信息页面。

步骤2 在“连接信息”区域，可单击，在“设置访问白名单”弹出框中输入“IP地址/地址段”的值。可以同时输入多个，有多个白名单请用“,”进行分隔。

📖 说明

每个引擎最多可以添加20个IP地址/地址段。

- 若需修改或删除IP地址/地址段，可在“设置访问白名单”弹出框中修改或删除已有的IP地址/地址段。
- 若需添加IP地址/地址段，可在“设置访问白名单”弹出框中添加新的IP地址/地址段。

步骤3 单击“确定”，当引擎状态由“配置中”变成“可用”时，白名单功能生效。

----结束

6.3.4 扩容 Nacos 引擎

随着业务流量激增，业务规模扩张，为保障服务注册发现、配置管理的稳定性和高性能，可对Nacos引擎进行规格变更。

Nacos引擎容量支持在线扩容，且只有低容量的引擎支持此操作。

约束与限制

扩容时会有短时间的接口请求失败，Nacos框架一般有重试功能，失败后会重新调接口。建议在您自己的变更维护时间窗内进行扩容操作。

扩容 Nacos 引擎

步骤1 在待扩容的Nacos引擎实例“操作”列，选择“更多 > 规格变更”，也可单击待扩容的Nacos引擎，在Nacos引擎的“基础信息”页面，单击“基础信息”区域“容量规格”后的“规格变更”。

步骤2 在“注册配置中心扩容”页面，选择要扩容的容量。

步骤3 单击“立即变更”，确认无误后，单击“提交”。待实例运行状态由“变更中”变成“可用”时，扩容成功。

----结束

6.3.5 升级 Nacos 引擎版本

创建Nacos引擎时，使用的是当前最新的引擎版本。当后续再发布新版本的引擎时，用户可以升级到引擎最新版本。


约束与限制

- Nacos引擎升级过程中微服务与引擎的连接会出现闪断，已运行的微服务的业务不受影响。建议您在升级Nacos引擎时，尽量避开微服务的升级、重启、变更等操作。
- 当前只能升级到最新的版本。

升级 Nacos 引擎版本

步骤1 单击可升级的Nacos引擎“版本”列的。

说明

当引擎已为最新版本，则在该引擎“版本”列不存在。

步骤2 在弹出框中确认当前版本及升级后版本，单击“确定”，进行升级。

如果升级失败，可单击“重试”，尝试再次进行升级。

----结束

6.3.6 删除 Nacos 引擎

如不再使用Nacos引擎，可执行删除操作。支持删除处于如下状态的Nacos引擎：

- 可用
- 不可用
- 创建失败
- 变更失败
- 升级失败
- 未知

约束与限制

删除引擎后数据无法恢复，请谨慎操作。

删除 Nacos 引擎

步骤1 在待删除的Nacos引擎实例“操作”列，选择“更多 > 删除”，也可单击待删除的Nacos引擎，在Nacos引擎的“基础信息”页面，单击右上角的“删除”，在弹出的对话框中输入“DELETE”，单击“确定”。

说明

若删除失败，可执行“强制删除”。

----结束

6.4 管理命名空间

命名空间可以实现不同环境的配置的区分离，例如开发测试环境和生产环境的资源（如配置、服务）隔离等。不同的命名空间下，可以存在相同的Group或Data ID的配置。

约束与限制

- 接入Nacos的SDK中填写的是命名空间ID，命名空间的名称仅为控制台上查看时使用的标识。
- 当您的业务SDK使用了一个未在Nacos服务端创建的命名空间ID进行服务注册发现，也是可以正常完成注册发现，但是无法在注册配置中心的服务管理中查看到该服务，需要您创建完对应的命名空间后，才能正常查看，具体操作请参考[创建命名空间](#)。

前提条件

已创建Nacos引擎实例，具体操作可参考[创建Nacos引擎](#)。

创建命名空间

当创建实例时，会自动生成一个默认的命名空间public（保留空间），该命名空间不可编辑和删除。您可以直接使用此命名空间实现资源和服务的隔离。一共可以创建50个命名空间。

- 步骤1** 单击待操作的Nacos引擎实例名称。
- 步骤2** 在左侧导航栏选择“命名空间”，单击“创建命名空间”。
- 步骤3** 在弹出的创建命名空间对话框中设置参数如下，其中带*的配置项为必填。

表 6-3 命名空间参数

参数	说明
*命名空间名称	可自定义填写命名空间名称，支持除@、#、\$、%、^、&和*外的任意字符，且不超过128个字符。
命名空间ID	仅支持大小写字母、数字、短划线（-）和下划线（_），且不超过128个字符。命名空间ID必须唯一。 创建时若未输入ID值，则由系统随机生成。

- 步骤4** 单击“确定”，命名空间创建完成。

----结束

编辑命名空间

- 步骤1** 单击待操作的Nacos引擎实例名称。
- 步骤2** 在左侧导航栏选择“命名空间”。
- 步骤3** 单击待编辑的命名空间名称后“操作”列的“编辑”，可编辑“命名空间名称”。

说明

自动生成的命名空间public不可编辑。

步骤4 单击“确定”编辑完成。

----结束

删除命名空间

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“命名空间”。

步骤3 单击待删除的命名空间名称后“操作”列的“删除”。

步骤4 在弹出框中单击“确定”删除完成。

----结束

6.5 权限控制

6.5.1 权限控制概述

同一个Nacos引擎可能会有多个用户共同使用，开启了“安全认证”的Nacos引擎，通过微服务控制台提供的基于RBAC（Role-Based Access Control，基于角色的访问控制）的权限控制功能，使不同的用户根据其责任和权限，具备不同的引擎访问和操作权限。

开启了“安全认证”的Nacos引擎，支持微服务正常接入。

说明

- 只有引擎版本为2.1.0.1及以上版本支持此功能，若版本低于2.1.0.1，可参考[升级Nacos引擎版本](#)升级到最新版本。
- 当Nacos引擎版本为2.1.0升级到2.1.0.1及以上版本时，需要先开启安全认证初始化密钥信息，才可使用权限控制功能。
- Eureka兼容的实例不支持安全认证。

6.5.2 开启/关闭安全认证

约束与限制

- 开启安全认证后，控制台仅能看到有权限访问的命名空间。
- 开启安全认证后，没有配置用户名密码的客户端将无法访问Nacos实例，请谨慎操作。
- 关闭安全认证后，无法对各个用户进行权限控制，客户端无需密码即可访问Nacos实例，访问控制台可以看到全部的命名空间，请谨慎操作。

开启安全认证

Nacos引擎默认关闭安全认证。可通过控制台开启安全认证。

步骤1 单击待开启安全认证的Nacos引擎。

步骤2 在左侧导航栏选择“权限控制”。

步骤3 单击“设置安全认证”，开启“编程接口安全认证”。

步骤4 单击“确定”，等待Nacos引擎更新完成，引擎状态由“配置中”变为“可用”，开启安全认证成功。

----结束

关闭安全认证

Nacos引擎开启安全认证后，若不再需要，可在控制台上关闭安全认证。

步骤1 单击待关闭安全认证的Nacos引擎。

步骤2 在左侧导航栏选择“权限控制”。

步骤3 单击“设置安全认证”。在“安全配置”页面，关闭“编程接口安全认证”。

步骤4 在弹出框中单击“确定”，待该引擎的状态变为“可用”时，该引擎关闭安全认证成功。

----结束

6.5.3 账号管理

您可以登录微服务引擎控制台，根据实际业务需求为Nacos引擎创建新账号或对该Nacos引擎下已创建的指定账号进行管理操作。

新增账号

根据实际业务需求创建新账号，把合适的角色同账号关联，使用该账号的用户则具有对该Nacos引擎的相应的访问和操作权限。

步骤1 单击待操作的Nacos引擎。

步骤2 在左侧导航栏选择“权限控制”。

步骤3 在“账号管理”页签，单击“新增账号”，参考下表设置账号参数。

参数名称	参数说明
账号名称	输入新账号名称。账号创建后，账号名称不可修改。
密码	输入账号密码。
确认密码	再次输入账号密码。

步骤4 单击“确定”，完成新账号创建。

----结束

重置密码

为安全考虑，您可通过控制台，重置账号的密码。

- 步骤1 单击待操作的Nacos引擎。
 - 步骤2 在左侧导航栏选择“权限控制”。
 - 步骤3 在“账号管理”页签，单击待操作的账号“操作”栏的“重置密码”。
 - 步骤4 输入新密码和确认密码，勾选“我已确认知晓”，单击“保存”，账号密码重置成功。
- 结束

删除账号


- 步骤1 单击待操作的Nacos引擎。
 - 步骤2 在左侧导航栏选择“权限控制”。
 - 步骤3 在“账号管理”页签，单击待删除账号“操作”列的“删除”，在弹出的对话框中输入“DELETE”，单击“确定”。
- 结束

6.5.4 角色管理

您可以登录微服务引擎控制台，根据实际的业务需求对Nacos引擎的角色进行创建、编辑、删除和查看的操作。支持按命名空间或更细粒度的权限控制。

创建角色

- 步骤1 单击待操作的Nacos引擎。
- 步骤2 在左侧导航栏选择“权限控制”。
- 步骤3 在“角色管理”页签，单击“创建角色”。
- 步骤4 输入新角色名称。角色创建后，角色名称不可修改。
- 步骤5 设置关联用户，在下拉框中选择**新增账号**中创建的用户。
- 步骤6 添加权限配置。**权限之间是与的关系，即该角色同时拥有此处配置的权限。**

单击“ 添加权限配置”，在“命名空间”下拉框选择命名空间，在“权限动作”下拉框选择动作权限，包括只读、只写和读写。可同时添加多条权限配置，也可单击权限配置“操作”列的“删除”，删除某一条权限配置。

- 步骤7 单击“创建”，创建角色。
- 结束

编辑角色

- 步骤1 单击待操作的开启了安全认证的Nacos引擎。
- 步骤2 在左侧导航栏选择“权限控制”。
- 步骤3 在“角色管理”页签，单击待编辑角色“操作”列的“编辑”。
- 步骤4 根据实际业务需求，修改“命名空间”和“权限动作”。

步骤5 单击“编辑”，完成角色编辑。

----结束

删除角色

角色删除后无法恢复，请谨慎操作。

步骤1 单击待操作的Nacos引擎。

步骤2 在左侧导航栏选择“权限控制”。

步骤3 在“角色管理”页签，单击待删除角色“操作”列的“删除”。在弹出的对话框中输入“DELETE”，单击“确定”。

----结束

6.5.5 控制台资源管理


Nacos引擎支持命名空间关联企业项目，关系为N:1，即N个命名空间可关联到一个企业项目。

默认在[创建命名空间](#)中创建的命名空间是没有关联企业项目，可以通过编辑企业项目操作，将命名空间同企业项目进行关联。

编辑企业项目时，只支持更换企业项目，不可将企业项目置空。

步骤1 单击待操作的Nacos引擎。

步骤2 在左侧导航栏选择“权限控制”。

步骤3 在“控制台资源管理”页签下，单击待操作的命名空间右侧“企业项目”的，在“编辑企业项目”弹出框中，在下拉框中选择企业项目，单击“确定”，编辑成功。

----结束


6.6 管理 Nacos 引擎服务

微服务引擎控制台提供了对注册在Nacos上的服务进行管理的功能。

前提条件

已创建Nacos引擎实例。

创建服务

您可以通过控制台创建服务，刚创建的服务为空服务（即提供者数量为0），在服务列表中默认显示空服务，若您不想再显示空服务，可单击“隐藏空服务”后的 隐藏空服务。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“服务管理”。

步骤3 在“命名空间”下拉框中选择命名空间，在“命名空间ID”框中自动填充ID。

📖 说明

当选择的命名空间为“public”即保留空间，其命名空间ID默认为空。

步骤4 单击“创建服务”，在“创建服务”弹出框中输入配置项信息如下，其中带*的配置项为必填。

表 6-4 配置项

配置项	说明
*服务名称	可自定义输入服务名称。支持输入数字、大小写字母和特殊字符（_-:），长度不超过236个字符。
分组	可设置服务所属分组。支持输入数字、大小写字母和特殊字符（_-:），长度不超过128个字符。
*保护阈值	该服务下的健康实例占总实例数比例不足阈值时，触发保护阈值。取值范围为0~1，默认为0。

步骤5 单击“确定”，服务创建完成。

----结束

查看服务列表

步骤1 单击待查看的Nacos引擎实例名称。

步骤2 选择“服务管理”，在“命名空间”下拉框中选择命名空间，在“命名空间ID”框中自动填充ID。

📖 说明

当选择的命名空间为“public”即保留空间，其命名空间ID默认为空。

步骤3 查看该引擎该命名空间下的所有服务。

支持使用服务名称和分组名称搜索目标服务。

📖 说明

搜索目标服务时，模糊查询支持的字符有：,\$*+.[?。

----结束

查看服务详情

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“服务管理”。


步骤3 单击待查看的服务名，可查看服务详情。

- 查看服务基本信息，包括服务名称、命名空间名称、服务分组、命名空间ID、保护阈值和集群数。

- 在“实例信息”页签，可查看实例信息，包括实例的IP、端口、所属集群、健康状态、上下线状态、权重和元数据，也可进行[实例操作](#)，如根据元数据搜索实例、对实例进行上下线和修改权重等操作。
- 在“订阅者”页签，可查看所有订阅当前服务的客户端实例列表。列表中可查看订阅者和客户端的版本。

---结束

实例操作

- 根据元数据搜索：在“实例信息”页签，在“集群列表”中选择某集群，在“元数据搜索”输入框中分别输入元数据的key及value值，单击“添加筛选”可显示符合条件的实例信息。也可单击“清空”清空搜索数据。
- 实例上下线：在“实例信息”页签，单击待操作的实例右侧“操作”栏的“上线”或“下线”可将该实例进行上下线，待操作完成，其实例的“上下线状态”也会同步更新。
- 修改实例权重：在“实例信息”页签，鼠标放至待修改权重值的实例的“权重”列，单击 修改权重值，其取值范围为1~99，修改完成后，单击“确定”。

说明

若使用Nacos的权重功能，进行流量负载分摊，需要在客户端注册Nacos提供的NacosRule为Bean。

```
@Bean
NacosRule nacosRule() {
    return new NacosRule();
}
```

并在application.properties配置文件中增加配置。

```
xxx-service.ribbon.NFLoadBalancerRuleClassName=com.alibaba.cloud.nacos.ribbon.NacosRule
```

其中xxx-service为客户端的服务名称，即
spring.application.name=xxx-service

删除服务

- 仅允许删除空服务，当实例数量不为0时，不支持删除。
- 服务如果保持空服务状态大于1分钟，Nacos会自动删除此服务。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“服务管理”，单击待删除的服务名右侧“操作”列的“删除”。

步骤3 在弹出框中单击“确定”，删除服务完成。

---结束

6.7 管理 Nacos 引擎配置

6.7.1 Nacos 引擎配置概述

在Nacos引擎中，配置管理是核心功能之一，主要用于解决微服务架构中配置的集中管理、动态更新、环境隔离等问题。它通过统一的配置中心，无需修改代码或重启服务即可实时调整应用行为，提升系统的灵活性和可维护性。

Nacos配置管理核心功能：

- 集中配置管理
 - 将应用配置统一存储在Nacos服务器，避免配置分散在代码或本地文件中，便于集中维护和管控。
 - 支持按应用、环境（开发、测试、生产）、集群等维度分组管理配置，提升配置隔离性。
- 动态配置更新

配置修改后可实时推送到应用，无需重启服务，实现动态刷新（需结合应用端SDK支持）。
- 配置版本管理

自动记录配置修改历史，支持版本回滚，避免误操作导致的配置故障。
- 配置监听与推送

应用通过SDK监听Nacos配置变化，Nacos主动推送更新，确保配置实时生效。
- 多环境支持

支持不同环境（如dev、test、prod）的配置隔离，同一应用在不同环境可加载不同配置。

6.7.2 创建 Nacos 引擎配置

在Nacos中创建配置是实现微服务配置集中管理、动态更新和多环境隔离的基础，避免配置分散混乱，提升开发运维效率与系统稳定性。本章节指导您如何创建Nacos引擎的配置文件。

须知

配置项使用明文存储，请自行加密敏感数据。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 在“命名空间”下拉框中选择命名空间，在“命名空间ID”框中自动填充ID。

📖 说明

当选择的命名空间为“public”即保留空间，其命名空间ID默认为空。

步骤4 单击“创建配置”在弹出框中输入如下配置项参数，带*的参数项为必填参数。

表 6-5 配置项参数

配置项	参数说明
*配置集ID	配置集ID是组织划分配置的维度之一。Data ID通常用于组织划分系统的配置集。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有意义的名称标识。Data ID通常采用类Java包的命名规则保证全局唯一性。此命名规则非强制。 支持输入数字、大小写字母和特殊字符（_-.:），长度不超过255个字符。
配置分组	Nacos中的一组配置集，是组织配置的维度之一。 支持输入数字、大小写字母和特殊字符（_-.:），长度不超过128个字符。
命名空间	创建配置所属的命名空间。
配置格式	Nacos支持YAML、Properties、TEXT、JSON、XML、HTML六种常见配置格式在线编辑。默认为：TEXT。
*配置内容	输入配置内容。 配置内容不能大于100KB，如配置内容过大，请将此配置拆分成多个小配置。调整配置内容大小可能会影响Nacos稳定性，请谨慎操作。
描述	可自定义输入描述。支持输入任意字符，长度不超过128个字符。
所属应用	输入所属应用名称。支持输入数字、大小写字母和特殊字符（_-.:），长度不超过128个字符。
标签	输入标签。支持输入数字、大小写字母和特殊字符（_-.:），长度不超过64个字符。

步骤5 单击“发布”创建完成。

----结束


6.7.3 管理 Nacos 引擎配置操作

查询 Nacos 引擎配置

CSE Nacos支持通过配置集ID、分组名称、所属应用和标签进行查询配置。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 在配置列表上方筛选框中，可通过筛选配置集ID、分组名称、所属应用和标签，单击  筛选出符合条件的配置。

----结束

查看 Nacos 引擎配置详情

在CSE控制台可查看Nacos引擎配置的详细信息。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 单击待查看的配置集ID，进入“配置详情”页面查看配置详细信息。在“配置内容”区域，可单击“查找”查询配置信息。

----结束

编辑 Nacos 引擎配置

配置处在灰度发布状态时，需要回退或正式发布后才可进行编辑。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 通过以下任意一种方式实现编辑操作。

- 单击待编辑的配置集ID右侧“操作”列的“编辑”。
- 单击待编辑的配置集ID，在配置详情页面，单击“编辑”。

步骤4 在“编辑配置”页面可修改配置内容、配置格式、描述、所属应用及标签，单击“发布”弹出“配置内容对比”弹框，查看历史版本和当前版本内容差异。

步骤5 单击“发布”编辑配置完成。编辑配置页面还提供了灰度发布特性，具体操作参见[管理Nacos引擎配置的灰度发布](#)。

----结束

导入 Nacos 引擎配置

注册配置中心Nacos引擎支持在控制台导入配置文件（格式为zip包），减少手动创建配置的工作量。当新建的开发、测试、预发环境需要复用生产环境的配置，或Nacos引擎需要进行跨云或跨平台迁移时，可预先将生产环境的配置导出，再将配置文件导入目标引擎，批量完成配置初始化，缩短环境搭建；当配置被误删、篡改、Nacos集群故障导致配置丢失，可通过导入备份配置文件，快速重建服务配置环境。

说明

- 当配置文件导入成功后，实际导入到环境的配置信息取决于在操作时设置的对相同配置的处理策略，如终止导入、跳过和覆盖。
- 导入前需备份目标环境现有配置，优先增量导入，核心服务配置需小范围验证后再批量导入。
- 若需修改配置，建议将预先导出的配置文件，原样导入到目标引擎后，再在控制台上修改配置，修改操作请参考[编辑Nacos引擎配置](#)。

- 步骤1** 单击待操作的Nacos引擎实例名称。
- 步骤2** 在左侧导航栏选择“配置管理 > 配置列表”。
- 步骤3** 单击“导入配置”，参考下表设置导入参数。

图 6-1 导入配置

导入配置

相同配置

配置文件

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

- 步骤4** 单击“关闭”，完成导入。

说明

- 当“相同配置”选择为“终止导入”时，导入过程中，如果遇到和系统中相同的配置，则会弹出“终止导入”弹框，单击“确定”终止导入。
- 当“相同配置”选择为“跳过”时，导入过程中，如果遇到和系统中相同的配置，该配置会被跳过，继续导入其余配置，会弹出“导入成功”弹框，显示具体导入的配置信息，单击“确定”导入完成。

----结束

导出 Nacos 引擎配置

将Nacos中的配置导出为文件（properties、yaml、JSON 等），作为线下备份存储，防止因Nacos服务故障、误操作或配置误删导致的数据丢失。在引擎迁移时，导出原引擎配置并批量导入新引擎，避免手动重建配置的繁琐工作。

- 步骤1** 单击待操作的Nacos引擎实例名称。
- 步骤2** 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 选中待导出的配置，单击“导出”。

📖 说明

- 也可单击“全部导出”可将所有配置项信息导出到本地。
- 请您尽量分开导出，保证导出的配置文件大小不超过2MB。

图 6-2 导出配置



步骤4 在弹出框中单击“导出”配置项则导出到本地。

----结束

删除 Nacos 引擎配置

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 通过以下任意一种方式实现删除操作。

- 在待删除的配置集ID右侧“操作”列的“删除”。
- 可选中待删除的配置集ID，单击上方的“删除”。

步骤4 单击“确定”，配置删除成功。

----结束

6.7.4 管理 Nacos 引擎配置的灰度发布


CSE Nacos配置中心支持灰度发布，即在配置正式发布前可先进行小部分的发布验证，待配置确保无误后再正式发布，从而降低配置推送的风险。

配置灰度发布

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 在待灰度发布的配置项右侧，单击“操作”列的“编辑”。

步骤4 在“编辑配置”页面，单击  打开灰度配置。

步骤5 在输入框中选择待灰度推送的IP地址或手动输入灰度发布的实例IP，单击回车。

📖 说明

可同时配置多个实例IP。

步骤6 单击“发布灰度版本”弹出“配置内容对比”弹框，对比历史版本配置和当前配置内容。

步骤7 单击“发布灰度版本”灰度发布完成。

----结束

查看灰度版本配置

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 在正在灰度发布的目标配置项右侧，单击“操作”列的“编辑”。

步骤4 在“编辑配置”页面的“灰度版本”页签可查看灰度版本配置，也可进行“回退灰度发布”和“正式发布”，具体操作可参考[相关操作](#)。

----结束

相关操作

- 回退灰度发布：在“编辑配置”页面的“灰度版本”页签，可单击“回退灰度发布”则取消灰度发布，回退到历史版本。
- 正式发布：在“编辑配置”页面的“灰度版本”页签，可单击“正式发布”，在“配置内容对比”弹框中确认配置信息，单击“发布灰度版本”，则灰度版本将成为正式版本发布。

6.7.5 管理 Nacos 引擎历史版本配置

CSE Nacos支持查看配置的历史版本详情及回滚历史版本。

查看历史版本

通过查看历史版本配置，可以清晰地了解配置的变更过程，便于进行审计和追溯。当配置修改后出现问题，如服务报错、性能下降等，可以通过查看历史版本，对比当前配置与之前正常版本的差异，快速定位到问题所在的配置项和修改操作。同时，Nacos支持一键回滚到稳定版本，能够迅速恢复配置，减少故障对业务的影响。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 通过以下任意一种方式进入“历史版本”页面。单击某个时间段的历史版本的配置集ID，可查看该配置项的历史版本信息。**历史版本最多可保留30天。**

- 在某配置集ID右侧“操作”列选择“更多 > 历史版本”进入该配置的“历史版本”页面。
- 单击该配置集ID进入“配置详情”页面，切换“历史版本”页签，进入该配置的“历史版本”页面。

----结束

回滚历史版本

CSE Nacos提供回滚历史版本功能，帮助您在改错配置的情况下能够快速恢复，从而降低微服务系统在配置管理可能存在的风险。

步骤1 单击待操作的Nacos引擎实例名称。

步骤2 在左侧导航栏选择“配置管理 > 配置列表”。

步骤3 通过以下任意一种方式进入“历史版本”页面。

- 在某配置集ID右侧“操作”列选择“更多 > 历史版本”进入该配置的“历史版本”页面。
- 单击该配置集ID进入“配置详情”页面，切换“历史版本”页签，进入该配置的“历史版本”页面。

步骤4 单击待回滚的历史版本“操作”列的“回滚”，弹出“回滚历史版本详情”页面。

📖 说明

只有“操作类型”为“更新”的配置，才支持回滚操作。

步骤5 在“配置内容”区域，单击“回滚到此版本”，在弹出框中，单击“确认”回滚成功。

----结束

6.7.6 使用 Nacos 引擎的监听查询功能

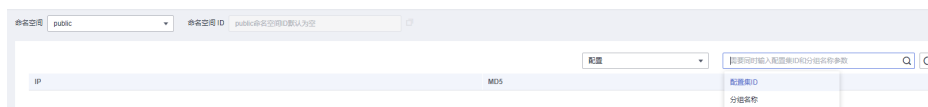
CSE Nacos提供监听查询的能力，即修改配置后，需要查看修改后的配置信息是否已推送到监听该配置的机器上，以便帮助您更好地检查配置变更是否推送到客户端。

步骤1 单击待操作的Nacos引擎实例名称。

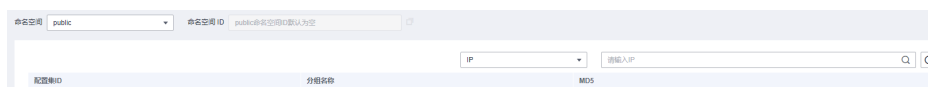
步骤2 在左侧导航栏选择“配置管理 > 监听查询”。

步骤3 在“命名空间”下拉框中选择命名空间，在下拉框中选择查询条件，单击🔍查询监听信息。

- 下拉框中选择“配置”，则在输入框中输入配置集ID和分组名称，查询该配置推送到的机器及推送状态。



- 下拉框中选择“IP”，则在输入框中输入监听配置的机器的IP，查询该机器监听的所有配置。



----结束

6.8 查看 Nacos 引擎运行监控

在使用Nacos引擎过程中，您可在CSE控制台提供的运行监控页面，观察Nacos引擎配置中心、注册中心等相关的常用指标，实时掌握Nacos集群的运行状态、提前识别性能瓶颈与故障风险、保障服务注册发现和配置管理功能的稳定可用，是微服务架构运维的关键环节。本文介绍如何查看Nacos引擎的监控指标数据。

约束与限制

查看Nacos运行监控数据需保证用户具有AOM FullAccess权限。

查看 Nacos 引擎运行监控

步骤1 单击待查看的Nacos引擎。

步骤2 在左侧导航栏选择“运行监控”。可以查看该Nacos的一些监控指标。

- 在“概览”页签，可以查看微服务实例数和配置数。且通过图表显示微服务实例数使用比例和配置使用比例。
 - 微服务实例数使用比例是指“已接入的微服务实例数”占“建议接入的最大微服务实例数”的比例。当比例较高时，请及时扩容。
 - 配置使用比例是指“已创建的配置数”占“允许配置的最大配置数”的比例。
- 在“配置中心监控”页签，可以在右上角的下拉框中选择时间查看某一时段的引擎的配置中心的监控数据，包括配置数、配置中心长连接数、写请求频率、读请求频率、写请求平均响应时间、读请求平均响应时间和配置推送耗时。支持选择的时间为近30分钟、近1小时、近6小时、近1天和近一周，默认显示近30分钟的监控数据。
- 在“注册中心监控”页签，可以在右上角的下拉框中选择时间查看某一时段的引擎的注册中心的监控数据，包括微服务数、微服务实例数、写请求频率、读请求频率、写请求平均响应时间、读请求平均响应时间和服务推送耗时。支持选择的时间为近30分钟、近1小时、近6小时、近1天和近一周，默认显示近30分钟的监控数据。

----结束

7 云审计服务支持的关键动作

7.1 云审计服务支持的 CSE 操作列表

CSE通过云审计服务（Cloud Trace Service，简称CTS）为您提供云服务资源的操作记录，记录内容包括您从管理控制台或者API发起的云服务资源操作请求以及每次请求的结果，供您查询、审计和回溯使用。

如果您需要收集、记录或者查询Nacos引擎和ServiceComb引擎的操作日志，需要先。通过云审计服务可查看Nacos和ServiceComb引擎最近7天的操作记录，支持记录的操作日志见[表7-1](#)和[表7-2](#)。

表 7-1 云审计服务支持的 Nacos 引擎操作列表

操作类型	资源类型	事件名称
创建引擎	engine	CreateEngineJob
删除引擎	engine	DeleteEngineJob
创建服务	service	createService
修改服务	service	modifyService
删除服务	service	deleteService
发布配置	config	publishConfig
删除配置	config	deleteConfig
创建命名空间	namespace	createNamespace
修改命名空间	namespace	modifyNamespace
删除命名空间	namespace	deleteNamespace

表 7-2 云审计服务支持的 ServiceComb 引擎操作列表

操作类型	资源类型	事件名称
创建引擎	engine	createEngine
删除引擎	engine	deleteEngine
升级或变更引擎	engine	upgradeOrModifyEngine
创建引擎备份任务	engine	createEngine_backup
删除引擎备份任务	engine	deleteEngine_backup
创建引擎恢复任务	engine	createEngine_recovery
创建引擎备份策略	engine	createEngine_backup_strategy
删除引擎备份策略	engine	deleteEngine_backup_strategy
更新引擎备份策略	engine	updateEngine_backup_strategy
更新灰度发布规则	engine	ModifyDarklaunch
删除灰度发布	engine	DeleteDarklaunch
修改配置项	engine	ModifyConfig
新增配置项	engine	CreateConfig
删除配置项	engine	DeleteConfig
更新治理规则	engine	ModifyGovern_policy
更新微服务	engine	modifyMicroservice
创建微服务	engine	createMicroservice
删除微服务	engine	deleteMicroservice
创建微服务标签	engine	createMicroserviceTag
更新微服务标签	engine	updateMicroserviceTag
删除微服务标签	engine	deleteMicroserviceTag
创建微服务规则	engine	createMicroserviceRule
更新微服务规则	engine	updateMicroserviceRule
删除微服务规则	engine	deleteMicroserviceRule
创建微服务契约	engine	createMicroserviceSchema
更新微服务契约	engine	updateMicroserviceSchema

操作类型	资源类型	事件名称
删除微服务契约	engine	deleteMicroserviceSchema
更新微服务依赖关系	engine	updateMicroserviceDependency
更新微服务属性	engine	updateMicroserviceProperty
更新微服务	engine	updateMicroservice
更新监控阈值	engine	updateThreshold
更新自定义规则	engine	updateItem_meta
删除自定义规则	engine	DeleteItem_meta
执行配置项清理	engine	executeConfig_cleanup
更新微服务实例状态	engine	updateInstanceStatus
更新微服务实例属性	engine	updateInstanceProperty
创建微服务实例	engine	createInstance
删除微服务实例	engine	deleteInstance

7.2 在 CTS 事件列表查看云审计事件

场景描述

云审计服务能够为您提供云服务资源的操作记录，记录的信息包括发起操作的用户身份、IP地址、具体的操作内容的信息，以及操作返回的响应信息。根据这些操作记录，您可以很方便地实现安全审计、问题跟踪、资源定位，帮助您更好地规划和利用已有资源、甄别违规或高危操作。

什么是事件

事件即云审计服务追踪并保存的云服务资源的操作日志，操作包括用户对云服务资源新增、修改、删除等操作。您可以通过“事件”了解到谁在什么时间对系统哪些资源做了什么操作。

约束与限制

- 用户通过云审计控制台只能查询最近7天的操作记录，过期自动删除，不支持人工删除。如果需要查询超过7天的操作记录，您必须配置转储到对象存储服务（OBS）或云日志服务（LTS），才可在OBS桶或LTS日志流里面查看历史事件信息。否则，您将无法追溯7天以前的操作记录。
- 用户对云服务资源做出创建、修改、删除等操作后，1分钟内可以通过云审计控制台查询管理类事件操作记录，5分钟后才可通过云审计控制台查询数据类事件操作记录。

- CTS新版事件列表不显示数据类审计事件，您需要在旧版事件列表查看数据类审计事件。

在 CTS 新版事件列表查看审计事件





- 步骤1** 登录控制台，单击左上角 ，选择“管理与部署 > 云审计服务 CTS”，进入云审计服务页面。
- 步骤2** 单击左侧导航栏的“事件列表”，进入事件列表信息页面。
- 步骤3** 在列表上方，可以通过筛选时间范围，查询最近1小时、最近1天、最近1周的操作事件，也可以自定义最近7天内任意时间段的操作事件。
- 步骤4** 事件列表支持通过高级搜索来查询对应的操作事件，您可以在筛选器组合一个或多个筛选条件。

表 7-3 事件筛选参数说明

参数名称	说明
事件名称	操作事件的名称。 输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。 各个云服务支持审计的操作事件的名称请参见《云审计服务用户指南》的“支持审计的服务及操作列表”章节。 示例：updateAlarm
云服务	云服务的名称缩写。 输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。 示例：IAM
资源名称	操作事件涉及的云资源名称。 输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。 当该事件所涉及的云资源无资源名称或对应的API接口操作不涉及资源名称参数时，该字段为空。 示例：ecs-name
资源ID	操作事件涉及的云资源ID。 输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。 当该资源类型无资源ID或资源创建失败时，该字段为空。 示例：{虚拟机ID}
事件ID	操作事件日志上报到CTS后，查看事件中的trace_id参数值。 输入的值需全字符匹配，不支持模糊匹配模式。 示例：01d18a1b-56ee-11f0-ac81-*****1e229

参数名称	说明
资源类型	操作事件涉及的资源类型。 输入的值区分大小写，需全字符匹配，不支持模糊匹配模式。 各个云服务的资源类型请参见《云审计服务用户指南》的“支持审计的服务及操作列表”章节。 示例：user
操作用户	触发事件的操作用户。 下拉选项中选择一个或多个操作用户。 查看事件中的trace_type的值为“SystemAction”时，表示本次操作由服务内部触发，该条事件对应的操作用户可能为空。
事件级别	下拉选项包含“normal”、“warning”、“incident”，只可选择其中一项。 <ul style="list-style-type: none"> • normal代表操作成功。 • warning代表操作失败。 • incident代表比操作失败更严重的情况，如引起其他故障等。


步骤5 在事件列表页面，您还可以导出操作记录文件、刷新列表、设置列表展示信息等。

- 在搜索框中输入任意关键字，按下Enter键，可以在事件列表搜索符合条件的数据。
- 单击“导出”按钮，云审计服务会将查询结果以.xlsx格式的表格文件导出，该.xlsx文件包含了本次查询结果的所有事件，且最多导出5000条信息。
- 单击按钮，可以获取到事件操作记录的最新信息。
- 单击按钮，可以自定义事件列表的展示信息。启用表格内容折行开关，可让表格内容自动折行，禁用此功能将会截断文本，默认停用此开关。

步骤6 （可选）在新版事件列表页面，单击右上方的“返回旧版”按钮，可切换至旧版事件列表页面。

----结束

在 CTS 旧版事件列表查看审计事件

步骤1 登录控制台，单击左上角，选择“管理与部署 > 云审计服务 CTS”，进入云审计服务页面。

步骤2 单击左侧导航栏的“事件列表”，进入事件列表信息页面。

步骤3 用户每次登录云审计控制台时，控制台默认显示新版事件列表，单击页面右上方的“返回旧版”按钮，切换至旧版事件列表页面。

步骤4 在页面右上方，可以通过筛选时间范围，查询最近1小时、最近1天、最近1周的操作事件，也可以自定义最近7天内任意时间段的操作事件。


步骤5 事件列表支持通过筛选来查询对应的操作事件。


表 7-4 事件筛选参数说明

参数名称	说明
事件类型	<p>事件类型分为“管理事件”和“数据事件”。</p> <ul style="list-style-type: none"> 管理类事件，即用户对云服务资源新建、修改、删除等操作事件。 数据类事件，即OBS服务上报的OBS桶中的数据的操作事件，例如上传数据、下载数据等。
云服务	在下拉选项中，选择触发操作事件的云服务名称。
资源类型	<p>在下拉选项中，选择操作事件涉及的资源类型。</p> <p>各个云服务的资源类型请参见《云审计服务用户指南》的“支持审计的服务及操作列表”章节。</p>
筛选类型	<p>筛选类型分为“资源ID”、“事件名称”和“资源名称”。</p> <ul style="list-style-type: none"> 资源ID：操作事件涉及的云资源ID。当该资源类型无资源ID，或资源创建失败时，该字段为空。 事件名称：操作事件的名称。各个云服务支持审计的操作事件的名称请参见《云审计服务用户指南》的“支持审计的服务及操作列表”章节。 资源名称：操作事件涉及的云资源名称。当事件所涉及的云资源无资源名称，或对应的API接口操作不涉及资源名称参数时，该字段为空。
操作用户	<p>触发事件的操作用户。</p> <p>下拉选项中选择一个或多个操作用户。</p> <p>查看事件中的trace_type的值为“SystemAction”时，表示本次操作由服务内部触发，该条事件对应的操作用户可能为空。</p>
事件级别	<p>可选项包含“所有事件级别”、“Normal”、“Warning”、“Incident”，只可选择其中一项。</p> <ul style="list-style-type: none"> Normal代表操作成功。 Warning代表操作失败。 Incident代表比操作失败更严重的情况，如引起其他故障等。

步骤6 选择完查询条件后，单击“查询”。

步骤7 在事件列表页面，您还可以导出操作记录文件和刷新列表。

- 单击“导出”按钮，云审计服务会将查询结果以CSV格式的表格文件导出，该CSV文件包含了本次查询结果的所有事件，且最多导出5000条信息。
- 单击按钮，可以获取到事件操作记录的最新信息。

步骤8 在需要查看的事件左侧，单击展开该记录的详细信息。

步骤9 在需要查看的记录右侧，单击“查看事件”，会弹出一个窗口显示该操作事件结构的详细信息。

步骤10 （可选）在旧版事件列表页面，单击右上方的“体验新版”按钮，可切换至新版事件列表页面。

----结束

相关文档

- 关于事件结构的关键字段详解，请参见“《云审计服务用户指南》>事件参考>事件结构”章节和“《云审计服务用户指南》>事件参考>事件样例”章节。

8 常见问题

8.1 使用 CSE 注意事项

8.1.1 查看 ServiceComb 引擎、Nacos 引擎信息时为什么看不到 VPC、ELB 等云服务信息？

问题描述

用户在查看ServiceComb引擎、Nacos引擎信息时看不到VPC、ELB等云服务信息。

问题原因

当前用户没有云服务（如VPC、ELB等）所在企业项目的查看权限，请联系租户管理员授权后再查看。

8.1.2 创建委托失败怎么解决？

问题描述

用户进入微服务引擎控制台，提示需要开通服务并授权，单击授权后提示授权创建失败。

原因分析

用户无权限创建委托或用户委托配额不足。

解决方法

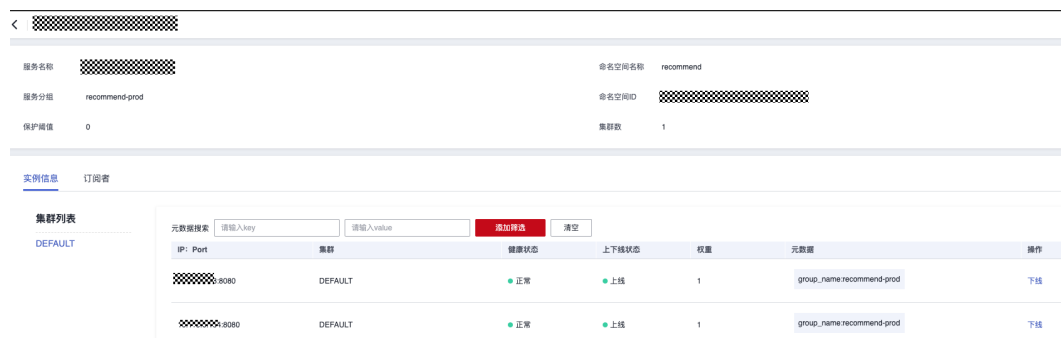
- 若用户进入IAM控制台时，系统提示权限不足，说明用户无权限创建委托，请参考《统一身份认证服务用户指南》的“创建委托时提示权限不足怎么办”章节进行处理。
- 若用户有权限创建委托，则登录统一身份认证服务控制台，在统一身份认证服务的左侧导航窗中，选择“委托”，查看当前是否还可以创建委托，当可创建委托数量为0时，可删除委托或调整委托配额后进行创建。

8.2 Nacos 引擎

8.2.1 服务启动时注册了端口为 8080 和 9090 的实例，在服务列表中 9090 端口实例丢失，导致请求 grpc 的时候报错

问题描述

服务启动时分别注册了端口为8080和9090的两个实例，对应http和grpc的端口，但是服务发现列表只能看到8080的实例，导致请求grpc的时候报错。



可能原因

使用了go sdk的client连接Nacos时，在一个微服务同时注册了两个实例，当前Nacos并不支持。

解决方法

同一个微服务不要同时注册两个实例。

8.3 ServiceComb 引擎

8.3.1 证书加载错误

问题描述

```
019/02/21 09:04:16 read ca cert file failed
019/02/21 09:04:16 Init chassis fail: read ca cert file failed
019/02/21 09:04:16 chassis init failed.
019/02/21 09:04:16 SetUp:RoomService init fail: read ca cert file failed
```

看到此错误时，说明verifyPeer选项被设置为true，却没有配置证书，go-chassis默认为false，不会自己更改配置。

解决方案

- 如果本来并不打算配置证书的，说明是开发者在开发或者持续集成的某个环节中自行更改的，请仔细排查代码从提交到部署环节中，是谁打开的此选项。

- 缺少的证书，配上即可。

8.3.2 无效头名称

问题描述

```

{"level": "[0:3]mERROR", "timestamp": "2018-12-19 10:28:43.145 +08:00", "file": "handler/transport_handler.go:46", "msg": "Call got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \"\\\""]"}
{"level": "[0:3]mERROR", "timestamp": "2018-12-19 10:28:43.145 +08:00", "file": "handler/transport_handler.go:46", "msg": "Call got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \"\\\""]"}
{"level": "[0:3]mERROR", "timestamp": "2018-12-19 10:28:43.146 +08:00", "file": "handler/transport_handler.go:46", "msg": "Call got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \"\\\""]"}
{"level": "[0:3]mERROR", "timestamp": "2018-12-19 10:28:43.146 +08:00", "file": "handler/transport_handler.go:46", "msg": "Call got Error, err [Post http://172.16.0.140:31007/v1/domains: net/http: invalid header field name \"\\\""]"}

```

此问题与go-chassis无关。

解决方案

请仔细排查业务代码，尤其是定制的go chassis handler，是否有传入空header，或者不规范的header name。

8.3.3 mesher 性能损耗是多少？

服务网格技术实际利用了网络流量劫持的方式来管理服务间流量，除了mesher本身内部的逻辑处理会耗时之外，还会引起额外的用户态和内核态间转换（CPU会有额外消耗），而前者相对于后者性能影响极小，因此性能损耗基本取决于网络中传输的payload大小。以http协议举例，影响传输速度的就是header、body等内容的大小。mesher一次端到端调用中的延迟为1ms，一个典型的用户测试过自己真实的业务调用，加上mesher后，延迟高了4ms，在用户可接受范围内。

以下测试结果为加入mesher前后的性能测试对比，使用的payload很小，就是字符串helloworld，但是加入了一定的代码以增加服务端的计算时间来模仿业务代码执行耗时。

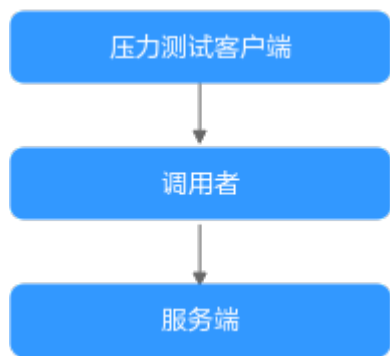


表 8-1 测试结果

指标	使用前	使用后
TPS	1749	1496
Latency	2.8ms	3.34ms
CPU	50%	100%
Concurrency	5	5

从以上结果可以看出mesher本身性能损耗很低，性能的主要瓶颈是在业务代码，如果增大payload内容，性能将会进一步降低。

建议在初期选型与POC时，使用该技术进行业务间的调用以测试真实性能损耗。

8.3.4 连接服务中心提示“Version validate failed”

异常消息

```
{"errorCode":"400001","errorMessage":"Invalid parameter(s)","detail":{"Version validate failed, rule: {Length: 64,Length: ^[a-zA-Z0-9_\\-]*$}}}
```

问题原因

使用了新版本的SDK，却连接服务中心老版本的SDK。

排查方法

检查服务中心的版本。可以从云平台下载最新版本的服务中心，或者从ServiceComb官网下载最新版本的服务中心。

8.3.5 连接服务中心提示“Not enough quota”

异常消息

```
{"errorCode":"400100","errorMessage":"Not enough quota","detail":{"no quota to create instance, ..."}}
```

问题原因

没有足够的额度增加服务实例。

排查方法

登录微服务引擎控制台，可以看到实例个数的额度。如果发现页面有额度，需要检查下代码配置的服务中心地址和区域信息。

8.3.6 服务名重复校验范围是什么？

问题描述

服务名重复校验范围是什么？

解答

服务名重复校验范围是微服务名称、微服务应用、微服务版本和微服务环境。

是一个微服务的主键，标识一个唯一的微服务。

请确保主键不重复。

8.3.7 为什么一定要定义服务契约？

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

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

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

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

8.3.8 如何解决微服务应用开发过程中微服务开发框架同 netty 版本不匹配的问题？

问题描述

开发微服务应用时，运行日志提示如下错误：

```
"Caused by: java.lang.NoSuchMethodError:
io.netty.handler.codec.http.websocketx.WebSocketClientHandshakerFactory.newHandshaker(Ljava/net/
URI;Lio/netty/handler/codec/http/websocketx/WebSocketVersion;Ljava/lang/String;ZLio/netty/handler/codec/
http/HttpHeaders;IJJ)Lio/netty/handler/codec/http/websocketx/WebSocketClientHandshaker;"
```

原因分析

通常是由于某个第三方软件引入了不匹配的版本依赖。

解决方法

可在开发环境下使用 `mvn dependency:tree` 命令查看依赖树，排查微服务开发框架同 netty 版本是否匹配。

例如，ServiceComb 2.0.1 开发框架所匹配的 netty 依赖版本为 4.1.45.Final。

使用 maven 管理复杂依赖关系，请参考：https://servicecomb.apache.org/cn/docs/maven_dependency_management/。

8.3.9 创建引擎提示“Duplicate cluster name”

异常消息

创建引擎时，报错提示：

```
{"error_code":"SVCSTG.00500500","error_message":{"kind":"Status":"Failure"...}message":"duplicate
cluster name","reason":"Conflict"}
```

问题原因

CSE引擎依赖的CCE服务出现问题，此时引擎的创建、删除、配置等功能都将不可用。

解决方案

可联系运维人员进行解决。

8.3.10 创建引擎过程中处理接入地址步骤失败，提示 the subnet could not be found

问题描述

创建引擎过程中，处理接入地址步骤失败，报错提示：

```
{"error_code":"SVCSTG.00500404","error_message":{"code":"VPC.0202","message":"Query resource by id xxx fail.the subnet could not be found."}}
```

问题原因

用户的项目未对CSE云服务进行委托授权。

解决方案

- 当您使用从ServiceStage发放的微服务引擎实例时，如想在CSE中创建新实例，需要对CSE云服务进行授权，具体操作可参考《微服务引擎用户指南》的“创建用户并授权使用CSE”章节。
- 当您没有授予任何权限时，由于CSE使用依赖VPC云服务，因此需要先参考《统一身份认证用户指南》的“创建委托”章节创建云服务委托cse_admin_trust，将操作权限委托给CSE。

8.3.11 SpringCloud 应用连接 ServiceComb 引擎 2.x 版本配置中心失败

问题描述

代码中配置了配置中心但是无法获取到配置项。

```
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1277)
at org.springframework.boot.SpringApplication.run(SpringApplication.java:1236)
at com.huawei.financialsolutionassetcenter.console.ConsoleApplication.main(ConsoleApplication.java:29)
Caused by: java.lang.IllegalArgumentException: Create breakpoint : Could not resolve placeholder 'cse.example.key1' in value "${cse.example.key1}"
at org.springframework.util.PropertyPlaceholderHelper.parseStringValue(PropertyPlaceholderHelper.java:186)
```

问题原因

配置文件中配置中心的类型和地址不正确。

排查过程

1. 引擎使用的2.x版本但是配置中心使用的是1.x版本的key。
2. 配置文件指定了环境，但是代码中获取的时候配置环境的key不正确。

解决方案

1. 将配置文件中配置中心的类型改为kie，即
spring.cloud.servicecomb.config.serverType: kie。
2. 在配置文件中修改环境的key，即参数server.env。

8.3.12 在全局配置中配置相关的配置项修改后，服务获取的配置内容未修改成功

问题现象

全局配置修改后未生效，日志中未打印关键字changed。

```
2021-11-15 10:13:33.710 INFO 6212 --- [ntloop-thread-0] .a.s.ConfigCenterConfigurationSourceImpl : Config value cache changed; action:set, item:[servicecomb.credentials.akskEn
2021-11-15 10:13:33.710 INFO 6212 --- [ntloop-thread-0] o.a.s.config.client.ParseConfigUtils : Updating remote config is done, revision has changed from default:9602643 to
2021-11-15 10:13:36.538 INFO 6212 --- [ctator-poller-0] o.a.s.m.c.publish.DefaultLogPublisher :
verts:
instances:
  name      eventLoopContext-created
  registry  0
  transport 0
  config-center 0
```

问题原因

没有配置config-cc的依赖。

排查过程

1. 在application.yaml中查看是否有该配置中心的地址。
2. 在pom.xml文件中是否有config-cc的依赖。

解决方案

在pom.xml文件中添加config-cc的依赖。

```
<dependency>
<groupId>org.apache.servicecomb</groupId>
<artifactId>config-cc</artifactId>
</dependency>
```

8.3.13 获取配置失败

问题现象

微服务在接入相应的微服务开发框架（如spring-cloud-huawei、java-chassis）后，微服务通过SDK调用查询配置接口到ServiceComb引擎获取配置项失败。

问题原因

微服务与注册中心间的连接因网络、CPU等其他因素发生抖动时，可能会导致请求异常。

解决方案

微服务框架具有自愈能力，拉取配置失败后会自动进行重试，一般情况下不会导致业务异常。您可以查看下次获取配置是否成功，若否，请联系技术支持人员。