

微服务引擎

快速入门

文档版本 01
发布日期 2024-03-20



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

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

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

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

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

华为云计算技术有限公司

地址：贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址：<https://www.huaweicloud.com/>

目录

1 ServiceComb 引擎专享版.....	1
1.1 快速创建 ServiceComb 引擎.....	1
1.2 接入 ServiceComb 引擎.....	2
1.2.1 SpringCloud 应用通过 SpringCloudHuawei SDK 接入 ServiceComb 引擎.....	3
2 注册配置中心.....	6
2.1 快速创建 Nacos 引擎.....	6
2.2 接入 Nacos 引擎.....	7
2.2.1 Spring Cloud 应用快速接入 Nacos 引擎.....	7
2.2.2 Spring Cloud Eureka 应用快速接入 Nacos 引擎.....	9

1 ServiceComb 引擎专享版

1.1 快速创建 ServiceComb 引擎

本章节帮助您快速了解如何创建ServiceComb引擎。

前提条件





- 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[创建虚拟私有云和子网](#)。
- 当前登录账号拥有创建ServiceComb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微服务引擎](#)。

操作步骤

步骤1 进入[购买ServiceComb引擎专享版](#)页面。

步骤2 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。
*企业项目	选择微服务引擎所在的项目，可在下拉框中搜索和选择需要的企业项目。
*选择实例数	选择引擎规格。
*引擎类型	选择微服务引擎的类型。 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，例如：cse-lhy-nodelete。
*可用区	选择可用区。
*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。

参数	说明
描述	单击  , 输入引擎描述信息, 例如: 体验快速创建微服务引擎。
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3 单击“立即购买”, 进入引擎信息确认界面。

步骤4 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引擎专享版](#)。

---结束

1.2 接入 ServiceComb 引擎

1.2.1 SpringCloud 应用通过 SpringCloudHuawei SDK 接入 ServiceComb 引擎

本章节通过一个demo进行全流程的ServiceComb引擎使用操作演示，帮助您快速了解如何使用ServiceComb引擎。

📖 说明

本章节将使用一个provider服务和一个consumer服务接入ServiceComb引擎。

前提条件

- 已创建ServiceComb引擎，请参考[快速创建ServiceComb引擎](#)。
- 下载github的[demo源码](#)到本地并解压。


📖 说明

该demo的配置文件中已经完成了集成Spring Cloud Huawei的配置操作，若您需了解详细配置信息，请参考[Spring Cloud接入ServiceComb引擎](#)。

- 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。

操作步骤

步骤1 登录微服务引擎控制台。

1. 登录[华为云控制台](#)。
2. 单击 ，选择区域。
3. 单击左上角 ，在服务列表选择“微服务引擎 CSE”，进入微服务引擎控制台。

步骤2 在左侧导航栏选择“ServiceComb引擎专享版”。

步骤3 单击[前提条件](#)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。

服务发现 & 配置		微服务目录	配置管理
服务注册发现地址	https://192.168.0.210:30100,https://192.168.0.246:30100		
实例数配额		已用0/共100 (0%)	
配置中心地址	https://192.168.0.210:30110,https://192.168.0.246:30110		
配置条目配额		已用1/共600 (0%)	

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1-1所示的两个Jar包。

表 1-1 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。

- 方法一：直接将微服务provider和consumer部署至ServiceComb引擎所在VPC的ECS节点。
 - a. 请参考[购买并登录Linux弹性云服务器](#)在引擎实例所属VPC下创建一台ECS节点并登录。
 - b. 安装JRE，为服务提供运行环境。
 - c. 将步骤6生成JAR包上传至ECS节点。
 - d. 执行命令：**java -jar {对应jar包}**，运行生成的JAR包。
- 方法二：使用ServiceStage部署微服务provider和consumer。
 - a. 将步骤6生成的JAR包上传至OBS。
 - b. 参考[快速创建Kubernetes集群](#)创建CCE集群，并同ServiceComb引擎实例属于同一VPC。
 - c. 参考[创建环境](#)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。

d. 参考[创建并部署组件](#)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[前提条件](#)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 注册配置中心

2.1 快速创建 Nacos 引擎

本章节指导您根据实际业务需求创建Nacos引擎。

前提条件




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[创建虚拟私有云和子网](#)。

操作步骤

- 步骤1** 进入[购买注册配置中心](#)页面。
- 步骤2** 左侧导航栏选择“注册配置中心”。
- 步骤3** 在注册配置中心页面，单击“购买注册配置中心”。
- 步骤4** 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”。
*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。
*引擎名称	输入Nacos引擎的名称。
*注册配置中心类型	选择“Nacos”。 说明 Nacos引擎默认多可用区部署，部署在三节点上的，可提供可用区级别容灾能力。
*选择实例数	选择需要购买的容量规格。
版本	只能创建最新版本。

参数	说明
*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。
标签	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。 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。

步骤5 单击“立即购买”，进入引擎信息确认界面。

步骤6 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

2.2 接入 Nacos 引擎

2.2.1 Spring Cloud 应用快速接入 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。



本章节将使用一个provider服务和一个consumer服务接入Nacos引擎。

前提条件

- 已创建Nacos引擎，具体操作请参考[快速创建Nacos引擎](#)。
- 下载github的[demo源码](#)到本地并解压。
- 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。

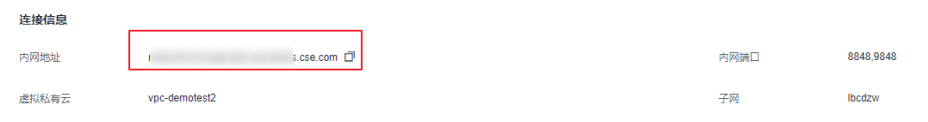
操作步骤

步骤1 登录微服务引擎控制台。

1. 登录[华为云控制台](#)。
2. 单击，选择区域。
3. 单击左上角，在服务列表选择“微服务引擎 CSE”，进入微服务引擎控制台。

步骤2 获取CSE的Nacos专享版引擎注册发现地址。

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2-1所示的两个Jar包。

表 2-1 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. 请参考[购买并登录Linux弹性云服务器](#)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[前提条件](#)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

2.2.2 Spring Cloud Eureka 应用快速接入 Nacos 引擎

本章节通过一个demo进行全流程的Spring Cloud Eureka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。

本章节将使用一个provider服务和一个consumer服务接入Nacos引擎。

前提条件


- 已创建Nacos引擎，具体操作请参考[快速创建Nacos引擎](#)。
- 下载github的[demo源码](#)到本地并解压。
- 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。

约束限制

- Nacos对Eureka的兼容，主要是兼容Eureka服务端侧的API，将服务侧注册的客户端实例信息进行保存与刷新，因此如果您仅使用Eureka作为注册中心，那么Nacos的诸多特性如命名空间和配置管理是不能使用的。
- 使用Eureka作为客户端，仅能在Nacos的“服务管理”中进行查阅，Eureka服务均使用Nacos的默认属性进行展示，即：
 - 默认命名空间：`public`。
 - 默认分组名称：`DEFAULT_GROUP`。
- 在Nacos中在“服务管理”页面创建服务的“保护阈值”设置的值属于Nacos的特性，无法作用于Eureka服务。

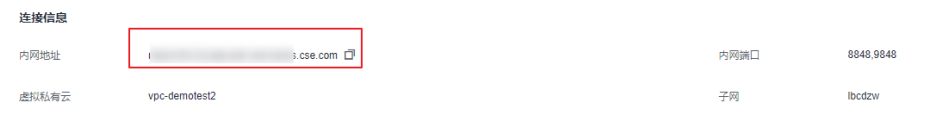
操作步骤

步骤1 登录微服务引擎控制台。

1. 登录[华为云控制台](#)。
2. 单击 ，选择区域。
3. 单击左上角 ，在服务列表选择“微服务引擎 CSE”，进入微服务引擎控制台。

步骤2 获取CSE的Nacos专享版引擎注册发现地址。

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中的注册中心地址和微服务名。

1. 在“application.properties”中配置Nacos的服务注册发现地址和微服务名。
 - 在下载本地的demo源码目录中找到“eureka-demo-master\eureka-consumer\src\main\resources\application.properties”文件，配置consumer服务：


```
server.port=9001
spring.application.name=eureka-client-consumer //微服务名
eureka.client.serviceUrl.defaultZone= XXX.nacos.cse.com:8848/nacos/eureka //Eureka的服务注册发现地址
eureka.instance.lease-renewal-interval-in-seconds=15 //服务心跳刷新间隔
eureka.client.registry-fetch-interval-seconds=15 //拉取注册中心间隔（建议与心跳间隔一致）
```
 - 在下载本地的demo源码目录中找到“eureka-demo-master\eureka-provider\src\main\resources\application.properties”文件，配置provider服务：


```
server.port=9000
spring.application.name=eureka-client-provider //微服务名
eureka.client.serviceUrl.defaultZone= XXX.nacos.cse.com:8848/nacos/eureka //Eureka的服务注册发现地址
eureka.instance.lease-renewal-interval-in-seconds=15 //服务心跳刷新间隔
eureka.client.registry-fetch-interval-seconds=15 //拉取注册中心间隔（建议与心跳间隔一致）
```

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2-2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
\eureka-consumer\target	eureka-client-consumer-1.0.0-SNAPSHOT.jar	服务消费者
\eureka-provider\target	eureka-client-provider-1.0.0-SNAPSHOT.jar	服务生产者

步骤5 部署SpringCloud应用。

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

1. 请参考[购买并登录Linux弹性云服务器](#)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[前提条件](#)中创建的Nacos引擎。
2. 选择“服务管理”，查看微服务eureka-client-consumer和eureka-client-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eureka-client-consumer和eureka-client-provider服务名，则表示微服务应用接入Nacos引擎失败。

----结束