阿里云 轻量级分布式应用服务

快速入门

文档版本: 20190919

为了无法计算的价值 | [] 阿里云

<u>法律声明</u>

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读 或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用于自身的合法 合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格遵守保密义务;未经阿里云 事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分 或全部,不得以任何方式或途径进行传播和宣传。
- 3. 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有任何通知或者 提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时发布更新后的用户文档。您 应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例
•	该类警示信息将导致系统重大变更甚至 故障,或者导致人身伤害等结果。	禁止: 重置操作将丢失用户配置数据。
A	该类警示信息可能导致系统重大变更甚 至故障,或者导致人身伤害等结果。	▲ 警告: 重启操作将导致业务中断,恢复业务所需 时间约10分钟。
	用于补充说明、最佳实践、窍门等,不 是用户必须了解的内容。	道 说明: 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定。
courier 字体	命令。	执行 cd /d C:/windows 命令,进 入Windows系统文件夹。
##	表示参数、变量。	bae log listinstanceid Instance_ID
[]或者[a b]	表示可选项,至多选择一个。	ipconfig[-all -t]
{}或者{a b }	表示必选项,至多选择一个。	<pre>swich {stand slave}</pre>

目录

法律声明	I
通用约定	I
1 SAE 使用指引	
2 准备工作	2
3 普通应用入门	
3.1 入门示例:将Demo应用部署到SAE	4
4 微服务应用入门	10
4.1 微服务场景指引	10
4.2 将 Spring Cloud 应用托管到 SAE	
4.3 将 Dubbo 应用托管到 SAE	17

1 SAE 使用指引

本文介绍使用 SAE 的旅程。本旅程是根据大量用户的实际使用总结出来的一般规律,仅供参考,您可以根据实际情况灵活调整阅读文档的顺序。

单击图片中的文档标题即可直接在新窗口中打开对应文档。

2 准备工作

本文介绍您在使用 SAE 托管应用之前需要完成的准备工作,包括创建命名空间和 VPC。命名空间 可以在地域中构建隔离的资源环境,专有网络(VPC)可以在阿里云上构建一个隔离的网络环境。

开通服务并授权

- 1. 登录SAE产品主页主页,单击立即开通。
- 2. 在密码登录页面输入阿里云账号和密码,并单击登录。
- 3. 在云产品开通页勾选我已阅读并同意《 Serverless 应用引擎服务协议》,并单击立即开通。
- 4. 开通完成会在页面出现恭喜,开通成功!的提示,然后单击管理控制台进入 SAE 控制台。
- 5. 在控制台首页弹出的安全授权提示对话框中单击立即授权。
- 6. 在云资源访问授权页面会提示需要对 AliyunSAEDefaultRole 授权,单击同意授权。

创建命名空间

如果您对资源没有安全要求,可以跳过此步骤,使用地域下的默认命名空间即可。

- 1. 登录 SAE 控制台
- 2. 在左侧导航栏,选择命名空间。
- 3. 在命名空间列表页面选择地域(Region),然后在页面右上角单击创建命名空间。

1) 注意:

目前 SAE 现已开放了华北 2 (北京)、华东 1 (杭州)、华东 2 (上海)和华南 1 (深 圳)地域。

4. 在创建命名空间对话框中,设置命名空间信息,然后单击创建。

创建命名空间	×
命名空间名称:	Team
命名空间ID:	cn-beijing:
归属地域:	cn-beijing
描述:	请输入描述信息
	0/64
	创建 取消

- ·命名空间名称(必选):输入命名空间名称。
- · 命名空间 ID (必选): 命名空间 ID 的前缀已根据所选地域而确定,不可编辑,只能设置自定义部分,自定义部分仅允许输入英文字母或数字。
- · 归属地域:即您选择的地域,不可设置更改。
- ・ 描述: 输入命名空间的描述信息, 最大可以输入64个字符

创建 VPC

1. 登录专有网络控制台搭建专有网络



如果您尚未开通专有网络 VPC 服务,需要点击开通专有网络服务。进入 VPC 售卖页面,阅读 并勾选专有网络 VPC 开通协议,然后单击立即开通。

- 2. 将 VPC 同步到 SAE 控制台中。
 - a. 登录 SAE 控制台
 - b. 单击创建应用页面的 VPC 网络选项的同步按钮。
 - c. 在 VPC 网络的下拉选项中能够选中创建的 VPC 则说明 VPC 同步成功。

3 普通应用入门

3.1 入门示例:将Demo应用部署到SAE

本文介绍如何将 SAE 所提供网页应用 Demo(用 Java 编写,已打成 WAR 包)部署到 SAE,并通 过配置公网 SLB 来实现该应用的公网访问。通过该入门示例,可以帮助您初步了解 SAE 应用托管 功能。

步骤一:开通服务

选择服务开通方式。

- ・在产品官网开通
 - ① 登录阿里云主页,在菜单栏选择产品 > 云计算基础 > 弹性计算 > Serverless > Serverless 应用引擎,进入 SAE 的产品主页。
 - 2. 在 SAE 的产品主页上,单击立即开通进入服务开通页面,请根据提示完成服务开通。
- ・在产品控制台开通
 - 1. 登录 SAE 控制台。
 - 2. 在欢迎页面单击立即开通, 根据提示完成服务开通。
- 步骤二: 创建 VPC

SAE 需要通过专有网络(VPC)构建安全隔离的网络环境。

如何创建 VPC具体请参见搭建专有网络。

道 说明:

如果您已经创建过 VPC,则可在创建 SAE 应用时,在 VPC 网络所在行中的同步按钮 ,将已有的 VPC 同步到 SAE 中。

步骤三:将 Demo 应用部署到 SAE

为了方便您快速熟悉在 SAE 部署应用,SAE 提供了 Demo 应用供您进行实践操作。该 Dmoe 应 用为 SAE 欢迎网页程序,提供 WAR 包和 JAR 包两种方式。本示例教程以 WAR 包方式部署为 例。

1. 下载 Demo 应用。

2. 创建 SAE 应用并部署。

部署 Demo 应用到 SAE 上。

a. 登录 SAE 控制台,在左侧导航树选择Serverless 应用引擎 > 应用列表。



若您是开通 SAE 服务后第一次登录控制台,系统会提示您先授权 SAE 访问您的云资源。您 只需在安全授权提示对话框中单击立即授权,然后在继而出现的云资源访问授权对话框中单 击同意授权,完成手机验证即可。

b. 在应用列表页面单击创建应用,并在应用基本信息页签内设置应用相关信息。

	应用基本信息	应用部署配置	\geq	确认规格	\geq	创建完成
您当前账户可	J用的实例总数为100个,目前已使用3个,更多额度请提交工 ^单 进行申请。					
* 应用名称:	TestDoc					
* 命名空间 ⑦:	testbj0001				\sim	
• VPC网络:						
应用实例数:	2 个 (最多创建10个)					
• 实例规格:	1Core 2GiB 2Core 4GiB 4Core 8GiB 8Core 16GiB					
应用描述:	应用描述主要介绍应用的基本情况.					
					0/100	
配置费用: ¥0/m 参考价格,具体扣费调	ì 間 					下一步:应用部署配置

- ·应用名称:输入应用名称。在本示例教程中,请输入my-sae。
- 命名空间:在下拉菜单中选择默认。该示例教程以最简单的场景为例,因此不设环境隔离的要求,您选择默认命名空间即可。
- ・ VPC 网络:
 - A. 先单击右侧的同步按钮,将已创建的 VPC 同步到 SAE 中。
 - 该过程大概需要耗时 3 分钟左右,请您耐心等待。若您在下拉选项中能够看到已创建的 VPC 和 VSwitch,则说明同步成功。
 - B. 在下拉菜单中选择要将应用部署到的VPC 和要使用的 VSwitch。
- ·应用实例数:选择要需要部署的应用实例。在该示例教程中实例数以1个为例。
- · 实例规格:单击请选择勾选所需的实例规格类型。

实例规格类型以 CPU 和 Memory 进行划分。在该示例教程中,选择最低规格### 2。

- ·应用描述:输入简短文字描述该应用的基本情况。在此您可以输入"用于入门练习的示例 应用"。
- c. 单击下一步:应用部署配置,并在应用部署配置页面,选择应用部署方式为 War 包部署 ,并按照页面指示进行配置,完成设置后单击下一步:确认价格。

* 应用部署方式:	○ 镜線 ● War包部署 ⑦	Jar包部署	0	
配置War包 *				
*应用运行环境	请选择 使用Springboot, Dubbo War的用户,应用运行环境请选择"apache-tomcat-XXX		-Container-XXX"	
* Java环境:	请选择		~	
* 文件上传方式:	上传War包		~	
▲上传War包:	选择文件 			
* 版本:	0/8			
* 时区设置:	UTC+8		~	
> 环境变量设计	6 设置容器运行环境中的一些变量,便于部署后灵活变更容器配置			
> Hosts绑定设	置 设置Hosts绑定,便于通过域名访问应用 🔗 如何设置Hosts纬	淀		
> 应用健康检查	话设置 用于判断容器和用户业务是否正常运行 🔗 如何设置应用	建康检查		
> 日志收集服务	3 设置日志收集规则, 能将业务日志输出到SLS, 便于统一管理和	浙 🔗 了解更多		
			上一步:应用	基本信息 确认创建

·应用运行环境:本示例Demo应用为简单的欢迎页Java程序应用,建议选择 EDAS-Container 3.5.5。



对于 Spring 或 Dubbo 应用请选择apache-tomcat-xxx; 对于 HSF 应用请选择 EDAS-Container-XXX。

Java环境:本示例Demo应用为简单的欢迎页Java程序应用,建议选择 Open JDK 7 或
 Open JDK 8。



在正式部署时,请正确选择开发应用时使用的JDK版本。

- ・文件上传方式:选择上传War包。
- ・时区设置:选择当前应用所在时区,如UTC+8。
- · 上传 War 包:单击选择文件,上传已下载的 Demo 应用安装包。
- ·版本:设置版本号,本示例中 Dome 应用版本为 1.0。
- ・环境变量设置、Hosts 绑定设置、应用健康检查和日志收集规设置:本示例不涉及环境变量配置、Hosts 绑定、应用健康检查和日志收集,无需配置。
- d. 在确认价格页签查看您所创建应用的详细信息以及配置费用情况,并单击确认创建。
- e. 在创建完成页签,单击提示信息中的应用详情页链接,进入my-sae的应用详情页面。

my-sae 为应用基本信息设置中设置的应用名称。

f. 单击实例部署信息页签, 如果看到运行状态列显示为绿色 Running, 表示应用部署成功。

基本信息			
✔ 默认分组 状态:运行1个实例			
显示实例IP信息			
实例名称 镜像	运行状态	运行时间	操作
404981.0541 3242/10703000	Running	5天	实时日志

_	
	说明・
_	りレウリ・

应用部署大概需要1~2两分钟,请您耐心等待。

步骤四:添加公网 SLB

应用部署成功后需要为其添加公网 SLB 才能被通过公网被访问。SAE 会自动帮您代购 SLB 服务,按量计费,您只需配置应用的监听端口。

- 1. 在my-sae的应用详情页面,单击基本信息页签。
- 2. 在应用访问设置区域框,单击公网访问地址一栏的添加公网SLB访问。

3. 在弹出的添加公网 SLB 访问对话框中选择已有的 SLB,并配置默认监听端口。设置 SLB 端口 为80,设置容器端口为8080。

添加公网SLB访问						×
 1、添加后, 能通过公网访问作 2、添加前, 请检查你的阿里 	饰的应用。系统会为你的应用自动购买—个公网SLB服务,指 云账户余额是否大于100元,SLB额度(每个账户50个)是否	5使用量计费。购买的SLB信息可以在负载 记用满。账户余额不足或者SLB额度用演	《均衡控制台查看 时,都会导致添加公网SLB9	5敗.		
请选择SLB: 0请选择 V						
检查项目	状态	说明				
SLB配额检查	成功 🖉					
账户余额检查	成功 🖉					
产品类别	产品配置	数量	付费方式	购买周期	资農	
负载均衡SLB - 公网	地域: 华北2 公网带宽:按使用流量计费	1	按量付费	无	查看价格	
TCP协议 HTTPS协议						
网络协议	SLB端口	容器尚口			操作	
ТСР					•	
+添加下一条监听						
					确定	取消

〕 说明:

・SAE 支持新建 SLB,本文示例中以复用原有 SLB 为例。

· SAE 支持 TCP 协议监听和 HTTPS 监听,本文示例中以 TCP 协议监听为例。

4. 单击确定。

添加公网 SLB 需要极短的几秒钟。添加完成后,您可以在应用详情页的公网访问地址一栏看到该公 网 SLB 的 IP 和端口 。

步骤五:通过公网访问 Demo 应用

根据公网访问地址栏显示的公网 SLB 的 IP 和端口,在浏览器中按http://slbip:port/的格式 输入地址并回车,可以看到 Demo 应用的首页。

你好,SAE 用户。恭喜你成功运行了第一个应用!	
SAE 是一个围绕应用和微能务的PaaS平台,提供多样的应用发布能力和轻量级微能务解决方案,帮助用户解决在应用和服务管理过程中监控、诊断和高可用运组问题;其提供了开源软件Dubbo的商业版。 关键字:应用运维、微服务、Dubbo、SpringBoot、魔眼监控、服务治理	
官网:https://www.aliyun.com/product/sae 快速入门:https://help.aliyun.com/document_detail/120281.html	
SAE J 20	项目组 019 年
Hello, SAE User. Congratulations! You have just deployed your first application using SAE!	
SAE is a PaaS platform designed to host applications and micro-services. It provides a variety of application deploying methods and lightweight micro-service solutions, helps users to monitor and diagnose applications and servic during the whole lifecycle, and at the same time fully addresses highly availability requirements in terms of platform maintenance. SAE provides the commercial edition of open source software Dubbo. Keywords: Application operation, Micro-service, Dubbo, SpringBoot, EagleEye monitoring, Micro-service management	.es
Website : https://www.aliyun.com/product/sae Ouick_start : https://www.aliyun.com/document_ideali/120281.html	
SAE	Team 2019Y

4 微服务应用入门

4.1 微服务场景指引

SAE 支持原生 Spring Cloud 和 Dubbo 微服务框架的应用,您可以将基于原生 Spring Cloud 和 Dubbo 微服务框架开发的应用迁移部署到SAE,进行SAE微服务管理。

为什么使用 SAE 服务注册中心

SAE 服务注册中心实现了 Dubbo 所提供的 SPI 标准的 注册中心扩展,能够完整地支持 Dubbo 服务注册、路由规则、配置规则功能。

SAE 服务注册中心能够完全代替 ZooKeeper 和 Redis,作为您 Dubbo、Spring Cloud 服务的 注册中心。同时,与 ZooKeeper 和 Redis 相比,还具有以下优势:

- · SAE 服务注册中心为共享组件,节省了您运维、部署 ZooKeeper 等组件的物理设备成本。
- · SAE 服务注册中心在通信过程中增加了鉴权加密功能,为您的服务注册链路进行了安全加固。
- · SAE 服务注册中心与 SAE 其他组件紧密结合,为您提供一整套的微服务解决方案。

原生 Spring Cloud 应用

- 如果您是初次接触原生 Spring Cloud 应用,并希望在 SAE 上部署一个原生 Spring Cloud
 应用,您需要在本地完成添加依赖和配置管理等操作,然后将应用部署到 SAE。具体操作请参见初次使用原生 Spring Cloud 构建微服务应用。
- 如果您在本地开发了依赖 Eureka、Consul、ZooKeeper 等组件实现的服务注册与发现的 Spring Cloud 应用。您希望将该应用部署至 SAE,只需要将服务注册与发现的组件的依赖和 配置替换成 Spring Cloud Alibaba Nacos Discovery,无需修改任何业务代码,即可将 应用部署到 SAE 并享受 SAE 的微服务管理能力。具体操作请参见将 Spring Cloud 应用托管到 SAE。

原生 Dubbo 应用

- ·如果您是初次接触原生 Dubbo 应用,希望在 SAE 上部署一个原生 Dubbo 应用,您需在本地 完成添加依赖和配置管理等操作,然后将应用部署到 SAE。具体操作请参见初次使用Dubbo 构 建微服务应用。
- 如果您在本地开发了依赖 ZooKeeper 和 Redis 等组件实现的服务注册与发现的 Dubbo 应用。您希望将该应用部署至 SAE,只需要将服务注册与发现的组件的依赖和配置替换成 edasdubbo-extension,无需修改任何业务代码,即可将应用部署到 SAE 并享受 SAE 的微服务管 理能力。具体操作请参见#unique_11。

4.2 将 Spring Cloud 应用托管到 SAE

您在本地开发了依赖 Eureka、Consul、ZooKeeper 等组件实现的服务注册与发现的 Spring Cloud 应用。若您想将该应用部署至 SAE,需要将服务注册与发现的组件的依赖和配置替换成 Spring Cloud Alibaba Nacos Discovery,无需修改任何业务代码,可将应用部署到 SAE 并进行 SAE微服务管理。

背景信息

Spring Cloud Alibaba Nacos Discovery 实现了 Spring Cloud Registry 的标准接口与规 范,与 Spring Cloud 接入 Eureka、Consul、ZooKeeper 等组件实现服务注册与发现的方式一 致。

使用开源版本 Spring Cloud Alibaba Nacos Discovery 开发的应用部署到 SAE 上,可以享受商 业版的 SAE 服务注册中心的优势和能力。

商业版的 SAE 服务注册中心,与 Nacos、Eureka 和 Consul 相比,具有以下优势:

- · 共享组件,节省了您部署、运维 Nacos、Eureka 或 Consul 的成本。
- · 对服务注册和发现的调用均进行了链路加密,保护您的服务安全性,无需担心服务被未授权的应 用发现。
- · SAE 服务注册中心与 SAE 其他组件紧密结合,为您提供一套完整的微服务解决方案,包括环境 隔离、平滑上下线、灰度发布等。

前提条件

下载最新版本的 Nacos Server,并按以下步骤启动 Nacos Server。

- 1. 解压已下载的 Nacos Server 压缩包。
- 2. 进入nacos/bin目录, 启动 Nacos Server。
 - ・ Linux/Unix/Mac 系统: 执行命令sh startup.sh -m standalone。
 - · Windows 系统: 双击执行startup.cmd文件。

步骤一:获取 Demo

eureka-service-provider 和eureka-service-consumer是SAE提供的2个 Demo, 二者 均为已经接入 Eureka 服务注册与发现的 Spring Cloud 应用,您需要下载到本地完成后续操作。

- eureka-service-provider 下载>>
- eureka-service-consumer 下载>>

步骤二: provider 应用侧操作

将原来的应用托管到 SAE 中, 需要在 provider 应用程序中添加 pom 依赖, 并指定 Nacos Server 的IP地址。

1. 添加 pom 依赖。

```
打开 provider 应用的 pom.xml 文件中,将 spring-cloud-starter-netflix-eureka-
client 替换成为spring-cloud-starter-alibaba-nacos-discovery,并设置 nacos
server 的版本信息。
```

替换前:

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</
artifactId>
</dependency>
```

替换后:

```
<dependency>
        <groupId>org.springframework.cloud</groupId>
        <artifactId>spring-cloud-starter-alibaba-nacos-discovery</
artifactId>
        <version>0.9.0.RELEASE</version>
</dependency>
```

📙 说明:

·本示例中使用的版本为 Spring Cloud Greenwich,对应 spring-cloud-starter-

alibaba-nacos-discovery 版本为 0.9.0.RELEASE。

- ・ 如果您使用的是 Spring Cloud Finchley 版本, 对应 spring-cloud-starteralibaba-nacos-discovery 版本为 0.2.2.RELEASE。
- ・ 如果使用的是 Spring Cloud Edgware 版本,对应 spring-cloud-starter-alibabanacos-discovery 版本为 0.1.2.RELEASE。(Spring Cloud Edgware 版本将在 2019 年 8 月结束生命周期,不推荐使用该版本进行应用开发。)
- 2. 指定 Nacos Server 的IP地址。

打开src\main\resources路径下的文件application.properties,指定 Nacos Server 的IP地址。

修改前:

spring.application.name=service-provider

```
server.port=18081
eureka.client.serviceUrl.defaultZone=http://127.0.0.1:8761/eureka/
```

修改后:

```
spring.application.name=service-provider
server.port=18081
spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
```

其中127.0.0.1为 Nacos Server 的 IP 地址。如果您的 Nacos Server 部署在另外一台机器,则需要修改成对应的 IP 地址。如果有其它需求,可以参考配置项参考在application. properties文件中增加所需配置。

- 3. 查询应用服务。
 - a. 执行nacos-service-provider中ProviderApplication的main函数, 启动应用。
 - b. 登录本地启动的 Nacos Server 控制台http://127.0.0.1:8848/nacos,在左侧导航栏 中选择服务管理 > 服务列表,可以看到服务列表中包含了service-provider,且在详情 中可以查询该服务的详情。

📋 说明:

本地 Nacos 控制台的默认用户名和密码均为 nacos。

步骤三: consumer 应用侧操作

将原来的应用托管到 SAE 中,需要在 consumer 应用程序中添加 pom 依赖,并指定 Nacos Server 的IP地址。

1. 添加 pom 依赖。

在 consumer应用的 pom.xml 文件中, 将 spring-cloud-starter-netflix-eurekaserver 替换成 spring-cloud-starter-alibaba-nacos-discovery, 并设置 nacos server 的版本。

替换前:

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-netflix-eureka-client</
artifactId>
</dependency>
```

替换后:



- 本示例中使用的版本为 Spring Cloud Greenwich, 对应 spring-cloud-starteralibaba-nacos-discovery 版本为 0.9.0.RELEASE。
- ・如果您使用的是 Spring Cloud Finchley 版本, 对应 spring-cloud-starteralibaba-nacos-discovery 版本为 0.2.2.RELEASE。
- ・ 如果使用的是 Spring Cloud Edgware 版本,对应 spring-cloud-starter-alibabanacos-discovery 版本为 0.1.2.RELEASE。(Spring Cloud Edgware 版本将在 2019 年 8 月结束生命周期,不推荐使用该版本进行应用开发。)
- 2. 修改配置。

打开src\main\resources路径下的文件application.properties,指定 Nacos Server 的IP地址。

修改前:

```
spring.application.name=service-consumer
server.port=18082
eureka.client.serviceUrl.defaultZone=http://127.0.0.1:8761/eureka/
```

修改后:

```
spring.application.name=service-consumer
server.port=18082
spring.cloud.nacos.discovery.server-addr=127.0.0.1:8848
```

```
其中127.0.0.1为 Nacos Server 的IP地址。如果您的 Nacos Server 部署在另外一台机
器,则需要修改成对应的 IP 地址。如果有其它需求,可以参考配置项参考在application.
properties文件中增加配置。
```

3. 查询应用服务。

- a. 执行eureka-service-provider中ConsumerApplication.java, 启动应用。
- b. 登录本地启动的 Nacos Server 控制台http://127.0.0.1:8848/nacos,在左侧导航栏 中选择服务管理 > 服务列表,可以看到服务列表中已经包含了service-consumer,且在 详情中可以查询该服务的详情。

圓 说明:

本地 Nacos 控制台的默认用户名和密码均为 nacos。

步骤四: 查看调用结果

在本地测试消费者对提供者的服务调用结果。启动服务,执行 IP + port / echo-rest / 自定义变量 或 IP + port / echo-feign / 自定义变量 查看调用结果。

- Linux/Unix/Mac 系统:执行curl http://127.0.0.1:18082/echo-rest/{自定义变量}
 量}或curl http://127.0.0.1:18082/echo-feign/{自定义变量}。
- Windows系统: 在浏览器中输入http://127.0.0.1:18082/echo-rest/{自定义变量}或
 http://127.0.0.1:18082/echo-feign/{自定义变量}。

步骤五:将应用部署到 SAE

1. 在应用的pom.xml文件中添加如下配置,然后执行 mvn clean package 命令将本地的程序 编译成可执行的 JAR 包。

```
<build>
<plugins>
<plugins>
<plugins
<pre>
```

2. 参照部署微服务应用到 SAE,分别将在步骤二: provider 应用侧操作和步骤二: provider 应 用侧操作修改依赖配置后的两个应用包部署至 SAE。



· SAE 暂不支持创建空应用,故第一次部署需在控制台完成。

·如果使用 JAR 包部署,在应用部署配置时应用运行环境必需选择标准 Java 应用运行环境。

·如果使用 WAR 包部署,在应用部署配置时应用运行环境必需选择apache-tomcat-XXX。

当您将应用部署到 SAE 时, SAE 服务注册中心会以高优先级去自动设置 Nacos Server 服务端 地址和服务端口,以及 namespace、access-key、secret-key、context-path 等信息。您 无需进行任何额外的配置,原有的配置内容可以选择保留或删除。

步骤六:结果验证

- 1. 为 consumer 应用绑定公网 SLB,并在浏览器键入所设置的公网访问地址,进入应用首页。
- 2. 在应用首页发起调用请求, 然后登录 SAE 控制台, 进入消费者应用详情页面。
- 在左侧导航栏选择应用监控 > 应用总览,查看服务调用数据总览。当能够监测到调用数据则说明 服务调用成功。

配置项参考

配置项	Кеу	默认值	说明
服务端地址	spring.cloud.nacos .discovery.server- addr	无	Nacos Server 启动监 听的 IP 地址和端口。
服务名	spring.cloud.nacos. discovery.service	\${spring.applicatio n.name}	给当前的服务命名。
网卡名	spring.cloud.nacos. discovery.network- interface	无	当 IP 未配置时,注册 的 IP 为此网卡所对应 的 IP 地址。如果此项 也未配置,则默认取第 一块网卡的地址。
注册的 IP 地址	spring.cloud.nacos. discovery.ip	无	优先级最高
注册的端口	spring.cloud.nacos. discovery.port	-1	默认情况下不用配 置,系统会自动探测。
命名空间	spring.cloud. nacos.discovery. namespace	无	常用场景之一是不同 环境的注册的区分隔 离,例如开发测试环境 和生产环境的资源(如 配置、服务)隔离等。

配置项	Кеу	默认值	说明
Metadata	spring.cloud.nacos. discovery.metadata	无	使用 Map 格式配 置,用户可以根据自己 的需要自定义一些和服 务相关的元数据信息。
集群	spring.cloud.nacos .discovery.cluster- name	DEFAULT	配置成 Nacos 集群名 称。
接入点	spring.cloud.nacos. discovery.enpoint	UTF-8	地域的某个服务的入口 域名,通过此域名可以 动态地拿到服务端地 址,此配置在部署到 EDAS 时无需填写。
是否集成 Ribbon	ribbon.nacos. enabled	true	一般不需要修改

更多关于 Spring Cloud Alibaba Nacos Discovery 的信息请参见开源版本的 Spring Cloud Alibaba Nacos Discovery 文档。

4.3 将 Dubbo 应用托管到 SAE

如果需要将一个原生 Dubbo 应用托管到 SAE,只需在本地应用程序中修改服务注册与发现组件的 依赖和服务注册中心配置,无需修改任何代码,可将原生 Dubbo 应用部署到 SAE,实现服务注 册、发现及调用。

前提条件

·参考配置轻量级配置中心下载轻量配置中心将其启动并进行host配置。

送明:

为了便于本地开发, SAE 提供了一个包含了 SAE 服务注册中心基本功能的轻量级配置中心。基 于轻量级配置中心开发的应用无需修改任何代码和配置就可以部署到云端的 SAE 中。推荐使用 最新版本。

・ 匹配相应的Dubbo 版本。

SAE 目前支持的 Dubbo 版本为 2.5.3~2.7.0。

- 如果您使用的 Dubbo 版本为 2.5.3~2.6.x,对应的 edas-dubbo-extension 的版本为 1.0.
 6。
- 如果您使用的 Dubbo 版本为 2.7.0,对应的 edas-dubbo-extension 的版本为 2.0.2。

操作步骤

1. 本地开发

Dubbo 应用使用的注册中心为 ZooKeeper, 在本地开发时需要将依赖由 ZooKeeper 替换为 SAE 注册中心,并在配置文件中将 ZooKeeper 的配置修改为 SAE 注册中心。

- a) 修改依赖和配置
 - A. 在应用程序的pom.xml文件中将注册中心的依赖由 ZooKeeper 修改为 SAE 注册中心。

原生 Dubbo 应用中的注册中心为 ZooKeeper, 依赖如下:

需要修改为 SAE 注册中心。

B. 在src/main/resources路径下的配置文件(如application.properties)中将

ZooKeeper 的配置修改为 SAE 注册中心。

之前的配置为 ZooKeeper 的地址:

dubbo.registry.address = zookeeper://127.0.0.1:2181

修改为 SAE 注册中心的地址:

dubbo.registry.address = edas://127.0.0.1:2181

- 2. 本地验证
 - a) 启动应用。
 - b) 在本地打开浏览器,访问本地轻量配置中心控制台(http://127.0.0.1:8080)。
 - c) 在左侧导航栏单击服务列表, 在服务列表页面的提供者列表和调用者列表中分别查看是否存在 Provider 和 Consumer 应用。
 - d) 依据应用的业务逻辑,检查应用的注册中心由 ZooKeeper 替换为 SAE 注册中心后,其服务 是否能够调用成功。

3. 将应用部署到 SAE

a) 在应用的pom.xml文件中添加如下配置, 然后执行 mvn clean package 将本地的程序打成可执行的 JAR 包。

```
<build>
<plugins>
<plugins>
<plugins>

<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-maven-plugin</artifactId>
<executions>

<goals>
<goal>repackage</goal>
</goals>
</plugin>
</plugin>
</plugins>
</build>
```

b) 根据您要部署的集群类型,参考应用部署概述部署应用。