



# Serverless 应用引擎 最佳实践

文档版本: 20220712



# 法律声明

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。
- 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
⚠ 危险	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。	
▲ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	會告 重启操作将导致业务中断,恢复业务 时间约十分钟。
〔〕) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。
? 说明	用于补充说明、最佳实践、窍门等 <i>,</i> 不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文 件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
粗体	表示按键、菜单、页面名称等UI元素。	在 <b>结果确认</b> 页面,单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。
斜体	表示参数、变量。	bae log listinstanceid
[] 或者 [alb]	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。	switch {active stand}

# 目录

1.	弹性	06
	1.1. SAE弹性伸缩最佳实践	06
2	.网络	10
	2.1. 部署在SAE上的应用如何访问公网	10
	2.2. 常见访问公网场景示例	12
	2.3. 关于在SAE非公网测试环境下使用GTS的注意事项	18
	2.4. 单应用在SAE和ECS侧实现混部	19
	2.5. 配置自定义DNS服务器	22
	2.6. SLB支持HTTPS多域名多证书操作	24
3	.镜像	29
	3.1. 容器镜像服务ACR企业版使用指引	29
	3.2. 跨账号拉取阿里云镜像	31
	3.3. 基于FC结合SAE的自动化CICD实践	34
	3.4. 使用SAE任务实现镜像构建及S2I解决方案	43
	3.5. 制作Node.js镜像	47
	3.6. 制作PHP镜像	51
	3.7. 制作Java镜像	56
	3.8. 制作Go镜像	63
4	.监控	67
	4.1. VPC网络下的SAE应用如何接入阿里云Prometheus监控	67
5	.存储	71
	5.1. 如何避免多进程/多客户端并发写同一日志文件可能出现的异常?	71
	5.2. SAE日志导入Kafka最佳实践	73
6	.微服务	78
	6.1. 开发场景:通过端云联调实现开发和测试环境隔离	78
	6.2. 开发场景:基于消息队列RocketMQ版实现环境隔离	81

6.3. 线上场景: 基于自建Spring Cloud Gateway或Zuul网关实现全链	83
6.4. 线上场景:基于Ingress网关实现全链路灰度	84
7.Java	87
7.1. JVM参数配置说明	87
7.2. JVM内存配置最佳实践	90
7.3. 设置Java应用的启动加速	94
7.4. 为Spring Boot应用设置健康检查	98
8.应用访问阿里云数据库	01
8.1. 如何设置阿里云关系型数据库RDS白名单	01
8.2. 如何设置云数据库Redis版白名单	05
8.3. 如何设置云数据库MongoDB版白名单	07
9.将应用从Web+迁移到SAE	11

# 1.弹性

# 1.1. SAE弹性伸缩最佳实践

SAE弹性伸缩可以实现在瞬时流量波峰到来时应用自动扩容,波峰结束后自动缩容,保障应用平稳运行,具有高可靠性、免运维、低成本的特点。本文介绍通过SAE部署弹性伸缩策略的最佳实践。

### 弹性伸缩准备

- 配置应用健康检查:确保应用在弹性伸缩过程中的整体可用性,仅在启动、运行并且准备完成时才接收流量。具体操作,请参见设置健康检查。
- 配置应用生命周期管理:确保缩容时按照预期优雅下线您的应用,配置停止前处理(PreStop设置)。
   具体操作,请参见设置应用生命周期管理。
- 采用指数重试机制:为避免因弹性不及时、应用启动不及时或应用没有优雅上下线导致服务调用异常,采 用Java指数重试机制进行服务调用。
- 优化应用启动速度。
  - 软件包优化:优化应用启动时间,降低因类加载、缓存等外部因素对应用启动时长造成的影响。
  - ・ 镜像优化:精简镜像大小,减少创建实例时镜像拉取耗时,可以有的放矢地借助开源工具分析并精简镜像层信息。
  - Java应用启动优化:在SAE上创建应用时,选择Dragonwell 11环境能够开启应用加速功能。具体操作, 请参见设置Java应用的启动加速。

#### 弹性规则配置

SAE支持基础监控、应用监控多指标组合配置,您可以根据当前应用的属性(CPU敏感、内存敏感或IO敏感)灵活配置。

您可以通过查看基础监控和应用监控对应指标的历史数据(例如过去6小时、12小时、1天或7天峰值,P95 或P99数值)并预估指标目标值,借助PTS等压测工具进行压测,了解应用可以应对的并发请求数量、需要 的CPU和内存数量,以及高负载状态下的应用响应方式,以评估应用容量峰值大小。

在配置弹性伸缩策略时,您需要考虑以下因素:

- 权衡可用性与成本, 配置指标目标值。示例如下:
  - 可用性优化策略: 配置指标值为40%。
  - 可用性成本平衡策略: 配置指标值为50%。
  - 成本优化策略:配置指标值为70%。
- 考虑梳理上下游、中间件和DB等相关依赖性,并配置对应的弹性规则或限流降级手段,以确保扩容时全链路的可用性。

弹性规则配置完成后,您可以通过监控并调整弹性规则使容量接近应用实际负载。关于查看监控的具体步骤,请参见基础监控。

Java应用运行时优化是通过释放物理内存,增强内存指标与业务关联性。借助Dragonwell运行时的环境,通 过增加JVM参数开启ElasticHeap能力,支持Java堆内存的动态弹性伸缩,从而节约了Java应用在运行时实际 使用的物理内存。关于ElasticHeap的更多信息,请参见G1ElasticHeap。

推荐配置为Dragonwell+Elast icHeap Periodic uncommit模式 (自动模式)。具体操作,请参见操作步骤和设 置启动命令。

• Java环境: Dragonwell

#### Serverless 应用引擎

#### 最佳实践·弹性

配置JAR包 *			
* 应用运行环境	标出ava应用运行环境	~	
* Java环境:	《现用Springboot、Dubbo JAR的用户,应用运行环境清选择"标准Java应用运行环境"。(使用HSF JAR的用户,应用运行环境清选择"EDAS pragonwell 8	s-Contain	er-XXX" 如何选型Java环境
* 文件上传方式:	Open JDK 8		
* 上传JAR包:	Dragonwell 8	~	
*版本:	Dragonwell 11 openjdk-8u191-jdk-alpine3.9	-	
* 时区设置:	openjak-70201-jak-alpines.9 UTC+8	~	

#### • JVM参数: -XX:+ElasticHeapPeriodicUncommit

∨ 启动命令设置 设置	容器启动和运行时需要的命令 🔗 如何设置启动命令	
启动命令	控制容器运行的输入命令, 如: /run/server	
启动参数	-XX:+ElasticHeapPeriodicUncommit	] 💼
	+ 添加	

⑦ 说明 不适用配置内存指标的应用类型:采用动态内存管理进行内存分配(例如Java JVM内存管理、Glibc Malloc和Free操作)的部分应用,没有及时向操作系统释放其闲置内存,导致无法实时减少实例消耗的物理内存和新增实例消耗的平均内存,进而导致无法触发缩容。

• 最小实例数配置

确认最小实例数≥2,配置多可用区vSwitch。避免因底层节点异常导致实例驱逐或可用区无可用实例,应 用停止工作。

• 最大实例数配置

确认最大实例数≤可用区IP数。避免因配置的IP数超出限制,应用无法新增实例。

您可以在基本信息页面的应用信息区域查看当前应用的可用IP数,如果可用IP较少,请替换或新增vSwitch。具体操作,请参见验证弹性伸缩策略。

∨ 应用信息
应用名: test
应用id: b58e1761-1ebb-4781-88e9-
vSwitch: sae-demo-switch(可用ip数为218 ) 多VSwitch部署
实例规格: 1Core, 2GiB, 系统盘磁盘空间20GiB 变更规格
应用创建时间: 2020年7月27日19:12:38
应用标签:编辑标签

### 弹性伸缩指标配置

#### 内存指标配置

#### 实例数配置

#### 弹性伸缩过程

您可以在应用**概览页**页面查看当前开启弹性伸缩配置的应用,并监控当前实例数已经到达峰值的应用,对其 弹性伸缩配置重新进行评估。

实例总数异常         Warning事件次数         变更单失败率         变更单失败次数         弹性生效的应用         应用错误数         应用总请求量         」 < >           实时展示监控指标学性规则自动生效的Top N 应用列表, 按当前实例数大小排序.           应用名称         命名空间         当前实例数         设置的最大实例数            test         cn-shenzhen         2         3	💆 Top 10 关注的	应用 华南1 (深圳)	$\sim$					С
	实例总数异常	Warning事件次数	变更单失败率	变更单失败次数	弹性生效的应用	应用错误数	应用总请求量	J < >
应用名称         命名空间         当前实例数         设置的最大实例数           test         cn-shenzhen         2         3	实时展示监控指标单性	规则自动生效的Top N 应用	列表, 按当前实例数	大小排序.				
test cn-shenzhen 2 3	应用名称	命	名空间	当前	实例数	设置	的最大实例数	
	test	cn	-shenzhen	2		3		

⑦ 说明 如果单个应用需要弹出超过50个实例,需提交工单申请白名单。

您可以在应用基本信息页面的实例列表中查看实例所属可用区。弹性伸缩触发缩容后,可能会导致可用区分配不均。如果可用区不均衡,您可以通过重启实例实现再均衡。

> 默认	✓ 默认分组 状态: 這行2个实例					
显示实情	削P信息					
	实例名称	vSwitch	镜像	运行状态 ↓	运行时间	操作
	25zgz	可用区A sae-demo-switch	/sae-demo-image	● Running 未配置健康检查	7天	实时日志   Webshell   事件   重启   删除
	9bzrm	可用区A sae-demo-switch	/sae-demo-image	● Running 未配置键康检查	7天	实时日志   Webshell   事件   重启   删除

进行部署应用等变更单操作时, SAE会停止当前应用的弹性伸缩配置, 避免两种操作冲突。如果您希望变更 单完成后能够恢复弹性配置, 可以在**部署应用**页面选择**系统自动恢复**。

部署应用			
部署应用后恢复自动弹性方式 ③ 系统自动恢复 ○ 人工启用恢复			
配置镜像 * registry-vpc.cn-shenzhen.aliyuncs.com/sae-demo-image, 修改镜像			
完整镜像仓库地址	registry-vpc.cn-shenzhen.aliyuncs.com/sae-demo-image		
∨高级选项			

# 弹性到达最大值

# 可用区再均衡

# 自动恢复弹性配置

# 弹性伸缩观测

您可以在**应用事件**页面观测SAE弹性生效行为,包括查看弹性伸缩时间和动作,以此来衡量弹性伸缩策略的 有效性并按需调整。

应用事件						
总计 1 项记录,当前仅保留7天内的	应用事件。					
自动弹性(HorizontalPod  ~ 全部	部事件等级 🗸 🗸 🗸	来源名 ~	gsgs-lautoscale	2		C
事件等级 事件	件原因 来源类型	来源名		m	「次发生时间	最近发生时间
- Normal Suc	ccessfulResc HorizontalP toscaler	odAu gsgsa-a	autoscale	20	021-05-12 11:39:31	2021-05-12 11:39:31
New size: 13; reason: Current nur	umber of replicas below Spec.M	linReplicas				

# 2.网络

# 2.1. 部署在SAE上的应用如何访问公网

部署在Serverless应用引擎SAE(Serverless App Engine)上的应用,其业务通常需要获取公网资源或者跨 VPC访问,本文介绍部署在SAE上的应用如何从VPC内网环境访问公网。

### 背景信息

在业务部署过程中,可能会遇到以下几种场景需要访问公网:

- 容器在运行时需要建立公网依赖。
- 与第三方有合作,例如使用微信小程序需要上公网。
- 应用需要跨VPC或地域访问数据库。

#### 实施方案

为同一VPC下的所有应用实例配置一个公共NAT网关代理并绑定EIP,通过SNAT功能为VPC内所有无公网IP地址的应用实例配置访问公网代理。

#### ? 说明

- 如果VPC内多个vSwitch下的实例需要出公网,那么针对多个vSwitch都需要配置SNAT。
- 单个VPC内存在多个应用访问公网, 配置代理后仅需绑定1个EIP。

#### 注意事项

如果您的VPC内存在多个NAT实例,您需要确保VPC内的路由表中的路由规则是绑定了SAE所关联的NAT实例。关于修改路由表的操作步骤,请参见创建和管理路由表。

#### 步骤一: 创建NAT网关

- 1. 登录NAT网关管理控制台。
- 2. 在公网NAT网关页面,单击创建NAT网关。
- 3. 首次使用NAT网关时,在创建公网NAT网关页面**关联角色创建**区域,单击**创建关联角色**。角色创建成 功后即可创建NAT网关。

关联角色创建	建创建关联
	DOME AND
	何建NAT网关
	63480 (P ( )

关于NAT网关服务关联角色的更多信息,请参见服务关联角色。

4. 在公网NAT网关页面, 配置相关信息, 单击立即购买。

参数	说明
付费模式	默认选择为 <b>按量付费</b> ,即一种先使用后付费的付费模式。更多信息,请参见 <mark>公网</mark> NAT网关计费。
所属地域	选择部署在SAE上的应用所在的地域。

参数	说明
所属专有网络	选择部署在SAE上的应用所在的VPC ID。创建后不能修改公网NAT网关所属的VPC。
关联交换机	选择交换机ID。
计费类型	默认选择为 <b>按使用量计费</b> ,即按公网NAT网关实际使用量收费。更多信息,请参 见 <mark>公网NAT网关计费</mark> 。
计费周期	默认选择为 <b>按小时</b> ,即按使用量计费公网NAT网关的计费周期为1小时,不足1小时 按1小时计算。
实例名称	设置公网NAT网关实例的名称。 实例名称长度为2~128个字符,以英文大小字母或中文开头,可包含数字、下划线 (_)和短划线(-)。
访问模式	选择公网NAT网关的访问模式。支持以下两种模式: • VPC全通模式(SNAT):选择了VPC全通模式,在公网NAT网关创建成功后当前VPC内所有实例即可通过该公网NAT网关访问公网。 选择VPC全通模式(SNAT)后,您需要配置弹性公网IP(Elastic IP Address, 简称EIP)的相关信息。 • 稍后配置:如需稍后配置或有更多配置需求,可在购买完成后,前往控制台进行 配置。 选择稍后配置,则只购买公网NAT网关实例。 本文选择VPC全通模式(SNAT)。
弹性公网IP	<ul> <li>选择公网NAT网关的EIP。支持以下两种模式:</li> <li>选择已有:在在弹性网关IP实例下拉列表中选择已有的EIP实例绑定到公网NAT 网关实例的EIP。</li> <li>新购弹性公网IP:在公网NAT网关实例地域新购按量付费EIP实例。</li> <li>线路类型:新购EIP的线路类型默认为BGP(多线)。</li> <li>安全防护:默认选择为默认版本的安全防护,即提供不超过5 Gbps的基础 DDoS防护能力。</li> <li>带宽峰值:根据业务需要,设置弹性公网IP的带宽峰值。</li> <li>计费类型:选择EIP流量的计费类型。</li> <li>按使用流量计费:根据每小时访问公网的实际流量计费。更多信息,请参见按使用流量计费。</li> <li>按固定带宽计费:由带宽值决定每日账单价格,与实际使用的流量无关。更多信息,请参见按固定带宽计费。</li> </ul>

5. 在**确认订单**页面确认公网NAT网关的配置信息,选中服务协议并单击**确认订单**。 当出现**恭喜,购买成功!**的提示后,说明已创建成功。您可以在**公网NAT网关**页面查看已创建的公网 NAT网关实例以及绑定的EIP。

#### 步骤二: 创建SNAT条目

创建SNAT条目,为专有网络中没有公网IP地址的应用实例,提供访问公网代理服务。

- 1. 登录NAT网关管理控制台。
- 2. 在顶部菜单栏,选择公网NAT网关的地域。
- 3. 在公网NAT网关页面,找到目标公网NAT网关实例,然后在操作列单击设置SNAT。
- 4. 在SNAT管理页签,单击创建SNAT条目。
- 5. 在创建SNAT条目页面, 配置相关信息, 单击确定创建。

参数	说明
SANT条目粒度	选择 <b>交换机粒度</b> 。
	选择VPC中的交换机,该交换机下所有的应用实例均可通过SNAT功能进行公网访问。
选择交换机	⑦ 说明 如果持有公网IP的应用实例(例如已经绑定了EIP)发起互联网访问,系统将优先使用其持有的公网IP地址,而不会使用NAT网关的SNAT功能。
<b>交换机网段</b>	选择交换机后,该区域会自动显示该交换机的网段。
	选择用来提供互联网访问的公网IP地址。
选择公网IP地址	⑦ 说明 您创建SNAT条目的公网IP地址不能再用来创建SNAT条目。
条目名称	设置条目名称。支持以大小写字母或中文开头,可包含数字、下划线(_)和短划线 (-),长度为2~128个字符。

创建成功后,在SNAT条目列表处查看配置的SNAT条目。

# 2.2. 常见访问公网场景示例

本文通过具体示例演示部署在SAE上的应用如何访问公网。

#### 前提条件

您已完成以下操作:

- 开通SAE服务
- 创建专有网络和交换机
- 创建命名空间

# 背景信息

某企业需要在SAE华北2(北京)地域创建应用,并在该应用上部署WordPress服务,同时使用华东1(杭州)地域的RDS作为数据库。

# 创建并部署WordPress应用

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击应用列表,在顶部菜单栏选择地域,单击创建应用。
- 3. 在应用基本信息页签,设置应用相关信息,配置完成后单击下一步:应用部署配置。

1 应用基本信息			3		
您当前账户下	Con-beijing的应用总实例数限制为300个,目前您已使用0个,如需增加额度请提工单申请	如果您的微服务应用	实例总数超过30个,推荐使用商业版独享的注册中心。		
* 应用名称:					
* 专有网络配置:	● 自定义配置 ○ 自动配置				
* 命名空间 ?:	演进章	/	请选择VPC	~ <b>G</b>	
* vSwitch:	调选标vSwitch	/ C (支持多选)			
* 安全组:	清約/29全组全名宣询      > C				
应用实例数 ?:	2 个 (最多创建50个)				
* VCPU:	0.5Core 1Core 2Core 4Core 8Core 12Core 16Core 32Core				
* 内存:	1GiB 2GiB 4GiB				
应用描述:	介绍应用的基本情况				
				0/100	
配置费用: 参考价格,具体扣费请	田 优惠详情 以账单为准、了解计题详情				下一步:应用部署配置

#### 参数说明如下表所示。

参数名	说明
应用名称	输入应用名称。允许数字、字母、下划线(_)以及短划线(-)组合,仅允许字母 开头,不超过36个字符。
专有网络配置	选择专有网络的配置方式。 • 自定义配置:选中后,您可以为创建的应用选择所需的命名空间、VPC、 vSwitch及安全组。 • 自动配置:选中后,SAE将自动帮您配置命名空间、VPC、vSwitch及安全组,无 需手动选择。
	在下拉菜单中选择创建好的命名空间。仅 <b>自定义配置</b> 专有网络时需要选择。
命名空间	⑦ 说明 命名空间和VPC是一一映射关系,关于修改VPC的更多信息,请参 见 <mark>创建和管理专有网络</mark> 。
vSwitch	在下拉菜单中选择vSwitch。仅 <b>自定义配置</b> 专有网络时需要选择。 vSwitch数量至少需要配置1个,建议不超过3个,且每个vSwitch至少匹配1个应用 实例。

参数名	说明					
	在下拉菜单中选择安全组。仅 <b>自定义配置</b> 专有网络时需要选择。					
安全组	注意 如果您的VPC内没有创建安全组,请单击创建安全组,根据提示创建安全组。更多信息,请参见创建安全组。					
应用实例数	选择需要创建的实例个数。					
VCPU	选择需要创建的实例CPU规格。					
内存	选择需要创建的实例内存规格。					
应用描述	填写应用的基本情况,输入的描述信息不超过100个字符。					

- 4. 在应用部署配置页面,选择镜像,依据页面指示进行配置。完成设置后单击下一步:确认规格。
- 5. 在确认规格页签,查看您所创建应用的详细信息以及费用配置情况,并单击确认创建。
- 6. 验证应用部署结果。

进入**应用详情**页面,查看应用的基本信息和实例部署信息。当实例部署信息页面显示实例的运行状态 为Running时,表示应用成功发布。

- 7. 为应用绑定SLB。
  - i. 在应用详情的基本信息页面,单击应用访问设置区域的添加公网SLB访问。
  - ii. 在添加公网SLB访问页面,设置SLB的监听规则,设置完成后单击确定。

SLB服务监听规定了如何将请求转发给后端服务器。一个SLB实例至少添加一条监听规则。

添加公网SLB访问						×		
<ul> <li>1.设置公网负载均衡,能通过公网访问您的应用,您可以选择已有的SLB服务,也可以创建新的SLB。</li> <li>2.选择新创建后,系统会为您的应用自动购买一个公网SLB服务,按使用量计费,详见SLB计费详情。购买的SLB信息可以在负载均衡控制台查看。</li> </ul>								
请选择SLB: 新建 💛 SLB复用说明 🥑								
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	資费			
负载均衡SLB - 公网	地域: 华东1 公网带宽:按使用流量计费	1	按量付费	无	查看价格			
ТСР协议 НТТР协议 НТТРS协议								
网络协议	SLB端口	容器端口			操作			
ТСР	80	8080			Ī			
+添加下一条监听								
					确定	取消		

8. 通过设置的公网SLB访问WordPress应用。



# 为WordPress应用添加公网访问权限

具体信息,请参见部署在SAE上的应用如何访问公网。

# 为WordPress应用设置跨区域的RDS数据库

- 1. 购买RDS实例。
  - i. 开通RDS服务。
  - ii. RDS实例基本信息设置。

在购买RDS实例时,选择地域为**华东1(杭州)**,选择可用区为**华东1可用区K**,选择网络类型 为**专有网络**。

- 2. 配置白名单。
  - i. 登录RDS管理控制台,在左侧导航栏单击**实例列表**,选择步骤1中所购买的RDS实例,并单击实例名 称进入RDS实例基本信息页面。

ii. 在左侧导航栏,单击数据库连接,在数据库连接页面的顶部菜单栏,单击申请外网地址,并在申 请外网地址对话框中单击确定。

← rm-bp1bgy	jb45ys9 (v ante) v Beesse Failure (v ante) v
基本信息	数据病性接 切除力组织用 切除力组织用器 标志性制度 中面外用地址 中面外用地址 如何脑镜和05 ● 为什么组织不上
账号管理 数据库管理	网络提型 专有网络(VPC-/vpc.bp1b0egota3ry)) 0 数据会行提优态 (原现会全 未开造 0 图1
备份恢复	内间地址 设置白岩单后才显示地址 内间跳口 3306
数据库代理	
监控与报警	◎ 请使用内网连接串进行实例连接,连接串对应的□>地址不固定,会发生变化。
数据安全性	
日志管理	11日1月11年4月1月
SQL洞察	AliaBérédualez
学校改画 自治服务 ~	杨定概式
	回日     目目     目目     目目     目目     目目       DNS     通知P     正CS     直知P     25.8     物理P     の

申请成功后,内网地址下方会新增一项外网地址。

数据库连接	切换交换机   切换为经典网络   修改连接地址   释放外网地址   如何连接RDS ②   为什么连接不上	<u>.</u>
网络类型	专有网络( VPC: - / vpc-bp1b0tegxsta9zry ) 🕢	数据库代理状态(原高安全 未开通 🕑 模式)
内网地址	设置白名单后才显示地址	内网端口 3306
外网地址	设置白名单后才显示地址	外网端口 3306

iii. 单击**外网地址**项的设置白名单,在白名单设置页签内,单击添加白名单分组。

iv. 在添加白名单分组页面,设置分组名称和组内白名单,设置完毕后单击确定。

⑦ 说明 此处设置组内白名单为0.0.0.0/0,即允许所有的外网都可访问。

添加白名单分组	×
允许访问实例的网络类型	
● 专有网络 ● 经典网络	
* 分组名称:	
sae_demo	
* 组内白名单	
0.0.0/0	
加载ECS内网IP	还可添加999个白名单
您可指定某个IP地址或者指定某个IP段作为白名单。 指定IP地址:若输入192.168.0.1,表明允许该IP地址访问当前实例 指定IP段:若输入192.168.0.0/24,表明允许192.168.0.0到192.168 地址访问当前实例。 若要输入多个IP地址或IP段,需用英文逗号分隔。如:192.168.0.1 何定位本地设备的公网IP地址?	l。 8.0.255之间所有的IP ,192.168.0.0/24。 <b>如</b>
1 白名单IP段设置为0.0.0.0/0意味着对公网开放,请述 了测试连接设置,测试后请立即修改。	<b>慎使用。若为</b>
9 白名单IP地址设置为127.0.0.1, 表示禁止所有地址说	i问。
④ 新白名单将于1分钟后生效。	

- v. 在白名单设置页签内,单击切换高安全白名单模式(推荐),然后在弹出的确认框中单击确认 切换。
- 3. 在RDS实例左侧导航栏,单击**账号管理**,在用户账号页签内单击**创建账号**,然后按照页面提示设置账号 信息,设置完成后单击**确定**。

关于账号类型的更多信息,请参见创建数据库和账号。

创建账号	×
* 数据库账号	
sae_demo	
由小写字母、数字、下划线(_)组成,以字母开头,以字母或数字结尾,最多32个字符	
* 账号类型 🕐	
自定义权限 * 密码	
必须包含三种及以上类型:大写字母、小写字母、数字、特殊符号。长度为8~32位。特殊字符包括!@#\$%^&*0_+-:	-
* 确认密码	
备注	
	€
0/	256
备注说明最多256个字符	
创译建 取消	
	BŶ

4. 本地测试是否可以通过外网访问RDS应用。

vagrant@dev:~\$ mysql -h rm-monitor, formends end mi.rds.aliyuncs.com -u root -p Enter password: Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 535 Server version: 5.7.18-log Source distribution Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

# 2.3. 关于在SAE非公网测试环境下使用GTS 的注意事项

在SAE的非公网测试环境(即线上正式环境)下使用GTS时,必须将URL参数配置为 https://cs2.gts.aliyuncs.com。

### 使用txc-client.jar时

使用txc-client.jar时,请按照以下示例代码配置URL参数。

```
<bean class="com.taobao.txc.client.aop.TxcTransactionScaner">
<bean class="com.taobao.txc.txt">
<bean class="com.taobao.txc.txt"/>
<bean class="com.taobao.txc.txt"/>
</bean class="com.taobao.txc.txt"/>
</bean class="com.taobao.txt"/>
</bea
```

# 使用txc-client-springcloud.jar时

使用txc-client-springcloud.jar时,请在properties配置文件中添

加 spring.cloud.txc.url=https://cs2.gts.aliyuncs.com 。

# 2.4. 单应用在SAE和ECS侧实现混部

如果您有应用需同时在SAE和ECS实例运行,那么您可以根据本文提供的混合部署方案实现。本文适用于对 ECS比较熟练的开发者查阅。

### 前提条件

您已完成以下操作:

- 为应用绑定SLB。
- 创建ECS实例并部署应用。更多信息,请参见创建ECS实例。

### 方案说明

SAE应用支持声明ECS实例,具体的方式为,将存量ECS实例ID、端口以及权重声明到SAE应用中,SAE应用在部署、扩缩容、停止、启动、重启、垂直扩缩容等场景,自动维护您在SLB后端的服务器组,将ECS实例加入SAE实例所在的虚拟服务器组中,统一对外提供服务。

SAE不会干涉ECS实例生命周期,仅提供实例上下线接口,供您优雅上下线以及集成。网络层面,需要保证 应用1的SAE和ECS实例都在一个VPC内,示意图如下。





↓ 注意 实现混部后,不允许编辑或绑定SLB,必须移除ECS实例后才能操作。如果在SAE修改了SLB 监听端口,则需重新添加ECS配置信息。

# 操作步骤

- 1. 登录传统型负载均衡CLB控制台。
- 在左侧导航栏,选择传统型负载均衡CLB(原SLB) > 实例管理,在顶部菜单栏,选择地域。
   该地域是您为SAE应用实例配置的SLB实例所在的地域。
- 3. 在**实例管理**页面,找到SAE应用使用的目标SLB实例,单击实例名称,然后单击虚拟服务器组页签。

实例	实例管理								
创建传统	<b>型负载均衡</b> 请选择标签 >>	可用区:	全部 ~ 模糊搜索 ~	请输入名称、ID或IP进行模糊搜索		Q			G = 7 (\$
	实例名称/ID		服务地址 💟	状态 🔽	监控	实例体检	第日/健康检查/后端服务器 >		攝作
	aat t Ib sae.do.not	0 0	120.2 "(公网IPv4)	✓ 运行中		ŵ	HTTP: 80 关闭	虚拟服务器组 asi/10/a5b67 ✔	些听配置向导 添加后端服务器 :
	启动 停止 释放设置	编辑标签						每页显示 20 💙	Ü共1条 < 上─页 下─页 >
?	) 说明 以	asi	为前缀的	虚拟服务器组	组,均	由SAE	自动创建。		

- 4. 找到目标虚拟服务器组,在其所在列,单击编辑。
- 5. 在编辑虚拟服务器组页面,单击继续添加。
- 6. 在我的服务器面板的选择服务器页签,选中同一VPC下的ECS实例,然后单击下一步。

我的服务器						×
	1 选择服务器	2			2 配置端口和权重	
选择后端服务器类型	云服务器ECS/弹性网卡EN	I 🗸				
搜索服务器名称、	ID或IP地址	Q 专有网络	▶ 请选择	$\sim$		
只展示可添加的多	实例 高级模式 😰 🔵					购买云服务器 🖸
✓ 云服务器	和D/名称	可用区	私网IP	公网IP/专有网络属性	状态	关联负载均衡次 数
✓ lau i-w	direct Street	深圳 可用区 A	192	47. <sup>-</sup> (公) vpc- vsw-	✓ 运行中	0
						AP
已选择 1 台服务器	下一步取消					

#### 7. 在配置端口和权重页签,按需配置端口和流量权重,然后单击添加。

我的服务器						×
(	→ 选择服务器 -			2 配置端口和	0权重	
云服务器ID/名称	地域	私网IP	端口 重置	权重 🕑 重置	操作	
lau i-v	深圳 可用区A	47. 192	80	80	添加端口 🕴 移除	
上一步 添加 月	<b>汉</b> 消					

#### ? 说明

- 。 权重越大转发的请求越多, SAE的所有实例负载均为100且不可调整。
- 如果开启会话保持,可能会造成后端服务器的请求不均匀。
- 权重设置为0,该服务器不会再接受新请求。

您可以在编辑虚拟服务器组页面查看到刚绑定的ECS实例信息。

8. 单击保存。

9. 在弹出的确认对话框中,单击确定。

# 2.5. 配置自定义DNS服务器

您可以在Linux操作系统中通过修改/etc/resolv.conf来配置自定义域名服务,同样您也可以通过SAE将相关的配置定义在配置项,然后将该配置项挂载到容器的/etc/resolv.conf文件。本文介绍如何在SAE配置自定义DNS服务器。

### 场景说明

SAE当前服务器的默认配置如下:

```
cat /etc/resolv.conf
nameserver 100.100.X.XXX
nameserver 100.100.X.XXX
```

#### 操作步骤

在SAE配置自定义DNS服务器的操作步骤如下:

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击命名空间,在顶部菜单栏选择地域,单击具体命名空间名称。
- 3. 在左侧导航栏单击配置项,在配置项页面单击创建。
- 4. 在创建配置项面板,选择手工创建,输入相关参数后单击确定。

← 创建配置项		×
* 创建方式 手工创建 文件录入		
所属命名空间		
cn-hangzhou		
* 配置项名称		
resolve		
配置项描述		
请输入描述信息,不超过255个字符。		
* 配置映射		
键	值	操作
resovle.conf	nameserver 100.100 nameserver 100.100 nameserver 10.0.	
+ 添加		
		► F
确定 取消		

#### 参数说明如下:

参数	说明
创建方式	<ul> <li>● 手工创建:本文选择手工创建方式。</li> <li>● 文件录入:具体信息,请参见管理和使用配置项(K8s)。</li> </ul>
配置项名称	自定义。
配置项描述	描述信息不超过255个字符。

参数	说明
配置映射	<ul> <li>键: 输入 resolve.conf 。</li> <li>值: 输入 resolv.conf 的文件内容。</li> <li>nameserver 100.100.X.XXX nameserver 100.100.X.XXX nameserver 10.0.X.XX</li> </ul>

5. 在**应用部署**页面,将该配置项挂载到/etc/resolv.conf文件下。具体操作,请参见在应用创建过程中注入 配置信息。

∨ 配置管	通过挂载配置文件的方式,向容器。	P注入配置信息。 🔗 如何进行配置管理			
	配置项名称	錢	挂载路径	权限	操作
	resovle 🗸	resovle.conf	✓ /etc/resolv.conf	只读	Ō
	+ 添加				

6. 通过Webshell进入容器,查看 /etc/resolv.conf文件,确认其内容是否符合预期。具体操作,请参见使用Webshell诊断应用。

在Webshell窗口执行以下命令查看文件。

cat /etc/resolv.conf

#### 预期输出。

nameserver 100.100.X.XXX nameserver 100.100.X.XXX nameserver 10.0.X.XX

# 2.6. SLB支持HTTPS多域名多证书操作

本文介绍如何给SLB实例通过HTTPS监听挂载多个证书,将来自不同域名的访问请求转发至Serverless应用引擎SAE(Serverless App Engine)同一应用实例组。

#### 前提条件

- 购买2个域名,分别为域名1和域名2。具体操作,请参见注册通用域名和域名交易。
- 上传2个SSL证书到证书服务,分别为证书1和证书2。具体操作,请参见选择阿里云签发证书。
- 创建1个SLB实例。具体操作,请参见创建和管理CLB实例。
- 创建SAE应用。具体操作,请参见创建应用。

# 背景信息

通过配置转发策略,将来自域名1和域名2的不同访问请求转发至SAE应用同一实例组,功能示意图如下:



### 步骤一:在SAE控制台配置网关路由

- 1. 登录SAE控制台。
- 2. 进入创建网关路由页面。
  - 从应用基本信息页面进入。
    - a. 在左侧导航栏单击应用列表,在顶部菜单栏选择地域,单击具体应用名称。
    - b. 在基本信息页面的网关路由设置区域,单击添加转发策略,在网关路由页面,单击新建。
  - 从命名空间详情页面进入。
    - a. 在左侧导航栏单击命名空间,在顶部菜单栏选择地域,单击具体命名空间名称。
    - b. 在命名空间详情页面的左侧导航栏,单击网关路由,在网关路由页面,单击新建。
- 3. 在新建路由规则面板, 配置相关参数。

# i. 在**配置网关入口**页签, 配置相关参数, 单击下一步。

新建路由規	RU	×
	配置网关入口	配置转发策略
* 名称:	sae-https-test	
* 网关类型:	公開 私間 通过公開開关税設的成量於開立所能发成量计是通过私用開关税設的成量不计费,但只能在当前VPC内部能发	
* SLB:	a379t	
*协议类型:	http https	
* SSL证书:	538- V	
* 访问端口:	8080	
下一步	取2尚	

#### 参数说明如下。

参数	说明
名称	设置路由规则的名称。
网关类型	选择公网。
	选择已创建的SLB实例。
SLB	⑦ 说明 SLB实例数量上限与SLB控制台上保持一致。关于SLB 实例的使用约束,请参见SLB使用说明。
协议米刑	选择https
SSL证书	选择HTTPS协议时必填。本文绑定已上传的证书1。
访问端口	设置接收请求并向后端服务或者应用进行请求转发的监听端口,端口范 围为1~65535,例如8080。
	<ul> <li>注意</li> <li>在同一个负载均衡实例内,监听端口不可重复。</li> <li>多条网关路由规则禁止使用同一个SLB的同一个端口。</li> </ul>

#### ii. 在配置转发策略页签, 配置相关参数, 单击确定。

新建路由规则							×
Ĩ	配置网关入口				配置转发策略		
• 城名规范:由英文字母、数字、中划线(-)和点 path规范:长度限制为1-80个字符。path以/?	."组成,域名不区分英3 开头,后面由字母,数	7大小写,仅支持*.example.com类型的泛 字,'-'、'/'、''、'%'、'?'、'#'、'&'、这些	域名。 标准域名如: 注符的0个或多个组	foo.example.com。 ઘ			
自定义转发策略							
城名	端口	Path 🕜		后端应用	3	容器端口	操作
www.example.com	: 8080	1		test-jar 🗸 🗸		8080	Ē
		+ 添加规则					
www.example.org	: 8080	1		test-jar 🗸 🗸		8080	Ē
		+ 添加规则					
十 添加域名							
SLB			SLB端□	后端应用		容器端口	
2379			8080	test-iar	~	8080	
			0000	con Jui		0000	
上一步 确定 取消							

#### 参数说明如下。

参数	说明
自定义转发策略	<ul> <li>域名:输入已购买的2个域名。本文假设您已购买的2个域名信息如下:</li> <li>域名1: www.example.com</li> <li>域名2: www.example.org</li> <li>Path:无需填写。</li> <li>后端应用:接收转发请求的应用,该应用须与转发请求的应用在同一个命名空间内。</li> <li>容器端口:设置访问您后端应用的容器端口,例如8080。</li> </ul>
默认转发策略	<ul> <li>后端应用:选择需要转发的后端应用。</li> <li>容器端口:设置访问您后端应用的容器端口,例如8080。</li> <li>注意 没有匹配自定义策略转发的请求,SAE会指定默认的转发策略。如果您不设置默认转发策略,会导致访问报错。</li> </ul>

您可以在网关路由页面查看新建规则。

# 步骤二:在SLB控制台添加扩展域名

- 1. 登录传统型负载均衡CLB控制台。
- 2. 在**实例管理**页面,在顶部菜单栏选择地域,单击目标实例ID。

3. 在目标实例的监听页签,单击操作列的::图标,在下拉菜单中选择扩展域名管理。

← a	379	100.044	and tool	<b>Anadra</b>	87188	<b>19.91</b>	25			创建监听	添加后端服	务器 更多	: ~
实例词	羊情 监听	虚拟服务器组	默认服务器组	主备服务器组	安全防护	监控	高精度秒级监控						
添加监	sofi											\$	C
	监听名称	前端协议	以端口 后端协	议/端口 运行	伏态 偷	建康检查状态	访问控制	监控	服务器组	操作			
	ingress/8080/Ib-	LO HTTPS:8	080 HTTP:-	<b>⊙</b> ✓ iį	新 新 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	k开启 2置	未开启 设置		[虚拟]ingress/80	修改监听配置	配置转发	策略   管理证书	:
	asi/80/379	HTTP:80	HTTP:-		5行中 #	k开启 2置	未开启 设置		[虚拟]asi/80/379	修改监听配置	配置转发	启动 停止	
		上 删除										删除 设置访问控制	
												扩展域名管理	A

4. 在**扩展域名管理**面板,单击添加扩展域名,配置相关参数,然后单击确定。

扩展域名管理		⑦ 配置扩展域名(B
● 添加扩展域名		
扩展域名列表		
域名 🚯	证书名称 (证书域名)	操作
37	37	编辑
58c .cn	58 100 cn(58 100 cn)	<b>编辑</b>   删除

#### 参数说明如下。

参数	说明				
扩展域名	输入您已购买的域名。				
ᄮᄰᄪᇩᄜᅚᆉ	选择该域名关联的证书。 本文选择您已上传的证书2。				
选拴服务畚证书	⑦ 说明 证书中的域名和您添加的扩展域名必须一致。				

您可以在扩展域名列表区域查看已扩展的域名。

# 结果验证

复制配置的域名地址,在浏览器中输入地址并回车,即可分别进入各自的应用首页。

- 域名1: https://example.com
- 域名2: https://example.org

3.镜像

# 3.1. 容器镜像服务ACR企业版使用指引

Serverless应用引擎SAE(Serverless App Engine)支持以镜像方式部署应用。当您指定部署应用的镜像文件时,如果选择阿里云容器镜像服务ACR企业版实例(ACR EE),会存在各种限制条件。本文介绍创建和部署SAE应用时,首次选择ACR EE的使用流程与注意事项。

### 背景信息

容器镜像服务ACR企业版面向企业客户,提供容器镜像、Helm Chart、Operator等符合OCI规范制品的安全托 管及高效分发服务,支持提升企业生产环境大规模分发、全球多地域分发和云原生DevSecOps分发的效率。 在使用容器镜像服务ACR企业版之前,您需要先创建ACR EE,以托管和分发您的云原生资产。更多信息,请 参见什么是容器镜像服务ACR。

SAE支持以镜像方式部署应用,包括本账号的阿里云镜像、其他阿里云账号的私有镜像、Demo镜像和公有 镜像。关于镜像的更多信息,请参见<mark>镜像最佳实践</mark>。

#### 使用流程

如果您是第一次通过ACR EE来部署SAE应用,您需要先在<mark>容器镜像服务控制台</mark>创建企业版实例。更多信息, 请参见创建企业版实例。成功创建后,您还需要为ACR EE绑定SAE应用对应的VPC和vSwitch才能成功部署。

因使用ARC EE存在多种限制条件,建议您按照以下步骤去创建或部署SAE应用,避免在操作过程中可能需要 多次跳转各个控制台。更多信息,请参见注意事项。



#### 注意事项

通过ACR EE来创建或部署SAE应用时,如果您直接在SAE控制台操作可能会遇到以下情况。

#### RAM用户无ACR EE访问权限

您需要为RAM用户授予与ACR EE相关的权限。具体操作,请参见配置仓库的RAM访问控制。

#### VPC匹配问题

● SAE未绑定或使用系统自动配置的VPC

仅支持自定义配置VPC,并且您需要为ACR EE配置专有网络的访问控制。具体操作,请参见配置专有网络的访问控制。

配置镜像*			
我的阿里云镜像	Demo镜像 公有镜像 其它阿里云账号私有镜像		
○ 镜像服务个人版	<ul> <li>         ・         ・         ・</li></ul>		
<ol> <li>注意事项:</li> <li>由于SAE中镜像拉取的策略不是Always,强烈建议不要使用 latest 作为镜像tag命名,否则会影响部署升级和回滚功能正常生效。</li> <li>镜像服务企业版尚未设置专有网络的访问控制,请前往镜像服务企业版产品控制台添加访问控制,查看详情</li> <li>镜像服务企业版需自定义设置VPC(非平台自动设置),请返回创建应用step1重新设置VPC或命名空间-基本信息菜单切换VPC。查看详情</li> </ol>			

- 创建应用: 在**应用基本信息**页签的**专有网络配置**行选择自定义配置。
- 部署应用: 在命名空间详情页面的基本信息页签切换VPC。具体操作, 请参见切换VPC。
- SAE已绑定VPC
  - 与ACR EE的VPC不匹配。

您可以选择使用ACR EE已绑定的VPC,或者修改ACR EE已绑定的VPC并替换为SAE现有的VPC。具体操作,请参见ACR EE配置VPC和SAE配置VPC。

配置镜像*			
我的阿里云镜像	Demo镜像 公有镜像	其它阿里云账号私有镜像	
○ 镜像服务个人版	● 镜像服务企业版 sae-	✓      グ      如何跨region同步     現像	
<ul> <li>注意事项:</li> <li>1. 由于SAE中镜像拉取的策略不是Always,强烈建议不要使用 latest 作为镜像tag命名,否则会影响部署升级和回滚功能正常生效。</li> <li>2. SAE中VPC请使用镜像服务企业版绑定的VPC:(vpc-),请前往镜像服务企业版产品控制台添加访问控制,查看详情</li> </ul>			

○ 与ACR EE的VPC匹配,但SAE的vSwitch非SAE推荐的可用区。

配置镜像*	
我的阿里云镜像	Demo镜像 公有镜像 其它阿里云账号私有镜像
○ 镜像服务个人版	● 镜像服务企业版 sae-
<ol> <li>注意事项:</li> <li>1. 由于SAE中镜像担</li> <li>2. SAE中vSwitch请</li> </ol>	立取的策略不是Always,强烈建议不要使用 latest 作为镜像tag命名,否则会影响部署升级和回滚功能正常生效。 使用推荐可用区:可用区D, 可用区E, 可用区F
⑦ 说明 一 用DescribeRed	个vSwitch对应一个推荐可用区。建议您在创建或部署应用前,先调 <mark>pions</mark> 接口查询目标地域下的vSwitch推荐可用区。更多信息,请参见 <mark>切换vSwitch</mark> 。

● 跨VPC部署

如果同账号下,您通过例如云企业网(Cloud Enterprise Network)等云产品将SAE的VPC A与ACR EE的 VPC B打通,实现了同地域或跨地域的网络互通,则需要使用以下方式进行部署:

- 。云效部署:使用云效2020部署应用至SAE。
- API部署:通过DeployApplication接口配置AcrInstanceId参数。

#### 跨账号部署

不支持。请提交工单或加入钉钉群反馈。

### 部署SAE应用

完成以上准备工作后,您可以按需选择技术栈语言并将应用托管到SAE。具体操作,请参见以下文档:

- 在SAE控制台使用镜像方式部署Java应用
- 在SAE控制台使用镜像方式部署PHP应用
- 在SAE控制台使用镜像方式部署多语言应用

# 3.2. 跨账号拉取阿里云镜像

SAE能够通过RAM角色扮演的方式达到跨账号拉取阿里云镜像的目的,即用户A扮演用户B的角色拉取用户B 的私有镜像。本文介绍跨账号拉取阿里云镜像的操作步骤。

#### 步骤一: 创建RAM角色

创建用户B的受信实体为阿里云账号类型的RAM角色。

- 1. 使用阿里云账号登录RAM控制台。
- 2. 在左侧导航栏,选择身份管理>角色。
- 3. 在角色页面,单击创建角色。
- 4. 在创建角色面板,选择可信实体类型为阿里云账号,然后单击下一步。
- 5. 在**配置角色**页签,输入**角色名称和备注**,选择**其他云账号**并输入用户B的阿里云账号,然后单击完成。

⑦ 说明 您可以访问安全设置页面查看阿里云账号ID。

6. 单击关闭。

#### 步骤二:为RAM角色授权

为步骤一创建的RAM角色,配置拉取用户B私有镜像的权限和信任策略。本步骤以在现有权限策略里添加镜像相关权限为例,如果您需要新建权限策略,请参见创建自定义权限策略。

- 1. 使用阿里云账号登录RAM控制台。
- 2. 在左侧导航栏,选择权限管理 > 权限策略。
- 3. 在权限策略页面,单击目标权限策略名称。
- 4. 在策略内容页签, 单击修改策略内容。
- 5. 通过可视化编辑或脚本编辑模式修改权限策略内容,然后单击下一步:编辑基本信息。

↓ 注意 请确保该RAM角色有cr.\*的相关权限。

需要添加的相关权限如下。

```
{
   "Version": "1",
   "Statement": [
       {
            "Effect": "Allow",
           "Action": [
               "cr:GetAuthorizationToken",
               "cr:ListInstanceEndpoint",
                "cr:PullRepository",
               "cr:GetRepository",
               "cr:ListRepositoryTag"
           ],
            "Resource": "*"
      }
   ]
}
```

- 6. 修改**备注**,然后单击**确定**。
- 7. 在左侧导航栏,选择身份管理 > 角色。
- 8. 在角色页面,单击目标RAM角色操作列的添加权限。
- 9. 在添加权限面板,按需添加权限,单击确定后单击完成。

### 步骤三:为RAM角色配置信任策略

在创建的用户B的RAM角色上配置信任策略,允许用户A账号对应的SAE服务账号进行角色扮演。

- 1. 使用阿里云账号登录RAM控制台。
- 2. 在左侧导航栏,选择身份管理>角色。
- 3. 在角色页面, 单击目标RAM角色名称。
- 4. 单击信任策略管理页签, 然后单击修改信任策略。
- 5. 在修改信任策略面板,将 Principal 中的 RAM 字段修改为 Service 字段,单击确定。 示例代码如下。

```
修改前
                                       修改后
                                         {
                                          "Statement": [
                                           {
                                             "Action": "sts:AssumeRole",
                                             "Effect": "Allow",
                                             "Principal": {
                                               "Service": [
                                        "123456789012****@sae.aliyuncs.com"
 {
                                            ]
    "Statement": [
                                             }
      {
                                           }
          "Action":
                                         ],
 "sts:AssumeRole",
                                          "Version": "1"
           "Effect": "Allow",
                                        }
           "Principal": {
              "RAM": [
                                       该RAM角色可以被阿里云账号
                                        (AccountID=123456789012****) 对应的SAE服务账
 "acs:ram::123456789012****:root"
                                       号 (123456789012****@sae.aliyuncs.com) 扮演。
           ]
          }
                                         ⑦ 说明 本示例中的 Service 为待跨账号
       }
                                         拉取镜像的服务。
    ],
    "Version": "1"
 }
该RAM角色可以被阿里云账号
(AccountID=123456789012****) 下授权的任何RAM
用户、RAM角色扮演。
```

### 步骤四: 部署SAE应用

您可以登录RAM控制台,在RAM角色的基本信息页面一键复制ARN。创建和部署应用时选择跨账号拉取镜像,并输入acrAssumeRoleArn配置。在配置镜像区域单击其它阿里云账号私有镜像, 在acrAssumeRoleArn文本框输入用户B的Worker RAM角色的ARN。具体操作,请参见以下文档:

- 创建应用
  - 在SAE控制台使用镜像方式部署Java应用
  - 在SAE控制台使用镜像方式部署PHP应用
  - o 在SAE控制台使用镜像方式部署多语言应用
- 部署应用

- o 分批发布应用
- 灰度发布应用

# 3.3. 基于FC结合SAE的自动化CICD实践

如果您没有持续交付流程,您可以参考本文快速搭建用于测试的持续交付环境,并通过执行 docker push {image} 命令就能免费将镜像部署到Serverless应用引擎SAE(Serverless App Engine)。如果您正在建设 持续交付流程,也可以参考本文,自定义自动触发部署镜像到SAE的方案。

### 前提条件

- 开通SAE产品,并部署一个镜像类型的应用。目前SAE支持部署所有技术栈的镜像,并针对其中的Java与 PHP技术栈,做了深度增强。如果您有兴趣体验SAE产品,您可以在SAE控制台创建SAE应用。具体操作, 请参见在SAE控制台创建应用。
- 开通容器镜像服务,并创建一个命名空间以及镜像仓库。命名空间以及镜像仓库建议是私有的,尤其是生 产环境,保证您的代码不会泄露。阿里云容器镜像服务提供免费的默认实例,可以免费创建命名空间以及 镜像仓库,并储存镜像。具体操作,请参见构建仓库与镜像。
- 开通函数计算,阿里云函数计算FC(Function Compute)是一个收费产品,但每月有100万次的免费调用 额度,您可以使用免费额度触发部署应用,无需支付其他费用。

#### 背景信息

本文以搭建一个简单的事件驱动的持续交付函数为例,全程不需要编写一行代码,实现捕获镜像提交事件, 并触发云函数,将镜像部署到SAE等操作,耗时不到10分钟。

Java函数的下载地址,请参见函数;源代码的下载地址,请参见源代码。

### 步骤一: 创建部署应用的云函数DeployTrigger

1. 创建函数。

- i. 登录函数计算控制台。
- ii. 在左侧导航栏, 单击**服务及函数**。
- iii. 在顶部菜单栏,选择地域。
- iv. 在服务列表页面,单击目标服务。
- v. 在函数管理页面,单击创建函数。

vi. 在创建函数页面,选择从零开始创建,在基本设置区域配置相关参数,然后单击创建。

请选择 <del>一种</del> 创建虽赖的方式					
从零开始创建 <ul> <li>从一个调单的 Hello World 示明开始,从头开始创建一个函数。</li> </ul>		使用容器镜像创建 选择一个容器镜像未新署念的函数。	0	使用模版创建使用示例模版,快速创建一个函数。	0
∨ 基本设置					
88	名称 DeployTrager 13/64 月時時会学母、数学、下初時時中半期後、不期以数学、中期後年後、长振在1-64 之间、				
* 运行环境	ب 8 مەدرا 👲 .				
函数触发方式	函数触起方式 通过率件触发 通过 HTTP 读求触发				
	● 通过发送 HTTP 请求能发品教协行,主要运用于快速的建 Web 服务等场景,点由这里查看更多详慎。				
实例类型 ⑦	突的笑意 ① · · · · · · · · · · · · · · · · · ·				
* 内存规格 ⑦	内存规格 ③ 512 MB ~				
手發輸入内存大小					
		的建 取消			

#### 参数说明如下。

参数	是否必填	操作
名称	否	填写 <b>DeployT rigger</b> 。必须以字母开头,可包含数字、字母 (区分大小写)、下划线(_)和短划线(-),不超过64个字 符。
运行环境	是	选择Java8。
函数触发方式	否	选择通过HTTP请求触发。
实例类型	否	选择 <b>弹性实例</b> 。
内存规格	是	设置函数执行内存。 <ul> <li>选择输入:在下拉列表中选择所需内存。</li> <li>手动输入:单击手动输入内存大小,可自定义函数执行内存。取值范围[128,3072],单位为MB。输入的内存必须为64 MB的倍数。</li> </ul>

创建完成后,页面会自动跳转至该函数的详情页面。

- 2. 在函数详情页面, 配置代码与触发器信息。
  - i. 单击函数代码页签,选择上传代码 > 上传JAR包,在通过ZIP、JAR包或WAR包上传代码对话框 内单击选择文件并上传函数文件,然后单击保存。

Java函数的下载地址,请参见函数。

ii. 单击触发器管理页签,在默认触发器的操作列单击编辑,在编辑触发器面板设置信息并单击确定。

编辑触发器 defaultTrigger		
触发器类型	HTTP 请求	
	HTTP 函数只能创建 HTTP 类型的触发器。如果您需要使用其他类型的触发器,请在创建函数时,选择"通过事件触发"。	
版本或别名 ②	LATEST	
是否需要认证	是否	
	①不需要身份验证,支持匿名访问,安全性低,任何人都可以发HTTP请求调用您的函数。	
* 请求方法 ⑦	POST X	/
确定	<b>双</b> 消	

参数说明如下。

参数	是否必填	操作
触发器类型	否	HTTP触发器只能在创建函数时创建。
版本或别名	否	默认为LATEST。 如果您需要创建其他版本或别名的触发器,首先您需要在函数 详情页的右上角切换到该版本或别名,然后再点击创建触发器 按钮。
是否需要认证	否	选择否,即设置为任何人都可以调用。
请求方式	是	选择 <b>POST</b> 。镜像服务的触发器发送的HTTP请求方式为 POST。

⑦ 说明 一个版本或别名的函数,最多只能创建一个HTTP类型的触发器。

3. 修改角色配置,添加调用SAE产品的权限。
i. 在**服务及函数**页面找到目标服务,在操作列单击配置,在角色配置区域,单击创建新的服务角 色,配置相关参数。

角色配置 配置服务中函数所使用的角色,	从而使函数可以获得角色所拥有的权限。	
服务角色 ⑦	sae-fc-test	~
	剧新 ○ 创建新的服务角色 [2] 注意:在 RAM 控制台创建服务所使用的角色时,需要选择"阿里云服务",并且"受信服务"需要选择"函数计算"。	
权限策略 ⑦	AliyunECSNetworkInterfaceManagementAccess AliyunSAEFullAccess FC-post-log-to-aliyun-fc-cn:	
	同新 C 新瓶茶路 L	

ii. 跳转至RAM控制台,添加SAE产品的调用权限AliyunSAEFullAccess。具体操作,请参见授予函数 计算访问其他云服务的权限。

如果您需要进一步精细化授权,您可以给角色授予部分需要部署的应用的修改权限。更多信息,请 参见<mark>权限策略和示例</mark>。

- 添加权限时,如果您已经创建了SAE读写权限的角色,您可以直接添加服务权限。具体操作,请参见授予函数计算访问其他云服务的权限。
- 如果您没有搜索到SAE读写权限的角色,您需要先登录RAM控制台创建角色并授权。具体操作, 请参见以下文档:
  - a. 创建可信实体为阿里云服务的RAM角色

创建RAM角色时,受信服务选择函数计算。

创建 RAM 角色	×
送择类型     2     配置角色     3     创建完成	
选择可信实体类型 阿里云服务	
角色类型	
● 普通服务角色 ○ 服务关联角色 【	
* 角色名称	
FC-SAE-FULL-ACCESS	
不超过64个字符,允许英文字母、数字,或"-"	
备注	
*选择受信服务	
函数计算	$\sim$
上一步 完成 关闭	

## b. 为RAM角色授权

## 为RAM角色授予AliyunSAEFullAccess权限。

忝加权限				
指定资源组的授权生效前提是该 单次授权最多支持5条策略,如	云服务已支持资源组,查看当前支持资源组的云服务。 秦绑定更多策略,请分多次进行。	[前往查看]		
授权范围				
〕 云账号全部资源				
)指定资源组				
请选择或输入资源组名称进行搜索				~
被授权主体				
FC-SAE-FULL-ACCESS@role	onaliyunservice.com X			
选择权限				
系统策略自定义策略 +	·新建权限策略		已选择 (1)	清空
SAE		G	AliyunSAEFullAccess	×
权限策略名称	备注			
AliyunSAEReadOnlyAccess	只读访问Serverless应用引擎(SAE)的权限			
AliyunSAEFullAccess	管理Serverless应用引擎(SAE)的权限			

4. 在函数详情页面, 配置环境信息与环境变量。

### i. 单击函数配置页签, 在环境信息区域单击编辑, 设置相关参数, 然后单击保存。

环境信息 配置函数的运行环境和其他关键参	χ.	
运行环境	4 Java 8	$\sim$
* 函数入口 ②	com.aliyun.serverless.DeployTrigger::handleRequest	
	●在运行环境为 Java 时,当前值的格式为 [package].[class]::[method]。如果当前值为 example.HelloFC::handleRequest,那么在函数被触发时, example 包中 HelloFC 美中的 handleRequest 函数。	将执行
*执行超时时间 ③	60	秒
* 单实例并发数 ⑦	10	

#### 参数说明如下。

参数	是否必填	操作
运行环境	否	选择Java 8。
函数入口	是	填 写com.aliyun.serverless.DeployTrigger::handleReq uest。
		设置超时时间。默认超时时间为60秒,最长为600秒。
执行超时时间	是	⑦ 说明 超过设置的超时时间,函数将以执行失败结束。
单实例并发数	是	单个实例能够并发处理的请求数。设置为10。

#### ii. 展开环境变量区域,选择使用表单编辑,单击添加变量,添加环境变量键值对。

添加ACCESS\_TOKEN环境变量,输入函数访问鉴权的自定义Token。您可以使用uuidgen命令 (oxs系统)或在互联网上寻找生成UUID的网页生成Token。

环境变量 配置函数运行环境中的环境变量。		
使用 JSON 楷式编辑 使用表单编辑		
変量 ACCESS_TOKEN + 添加変量	值 bb9e0024-cc8b-	Ē

#### ⅲ. 单击保存。

5. 获取函数的访问地址。

i. 在**服务及函数**页面找到目标函数,单击函数名称。

ii. 单击**触发器管理**页签,在触发器的配置信息区域复制函数的请求地址。

函数代码	触发器管理 函数面	置 监控指标	调用日志	链路追踪
	*/则夕 晏夕口能创建	UTTD 米刑的動始架		
		nn Fyendatar.		
创建触发器	可按触发器名称进行模糊	搜索。		Q
触发器名称 11	触发器类型 14	配置信息		
defaultTrigger	🕀 HTTP 请求	请求地址 ⑦ 请求方法 PO: 是否需要认证	https://1 ST 是	.cn-shanghai.fc.aliyuncs.com/2016-08-15/proxy/DeployTrigger.LATEST/DeployTri

⑦ 说明 为了节约成本,您可以直接使用函数计算提供的域名。函数计算每天提供1000次的域名额度,能够满足您测试环境部署场景的需求。

# 步骤二: 创建镜像推送触发器

- 1. 登录容器镜像服务控制台。
- 2. 在顶部菜单栏,选择所需地域。
- 3.
- 4. 在左侧导航栏选择**仓库管理 > 镜像仓库**,在镜像仓库页面单击目标镜像仓库。
- 5. 在仓库详情页面的左侧导航栏单击**触发器**,在**触发器**页面左上角单击创建。
- 6. 在创建触发器对话框中输入名称及触发器URL,选择触发方式为全部触发,单击确认。

创建触发器		×
类型	镜像推送触发器	
* 名称	DeploySAE 9/32	
* 触发器 URL	长度为 1-32 个字符, 只支持字母、数字、下划线 https://	
* 触发方式	● 全部触发 ○ 表达式触发 ○ Tag触发	
	确定取消	Ц.

↓ 注意 触发器URL中要包含鉴权的Token以及要部署的SAE应用ID。

# 步骤三:验证部署结果

本文的验证步骤以企业版容器镜像为例。更多信息,请参见以下文档:

- 使用企业版实例推送和拉取镜像
- 使用个人版实例推送拉取镜像
- 1. 执行以下命令,登录镜像仓库。

```
docker login --username=<镜像仓库登录名> <企业版实例名称>-registry.<regionId>.cr.aliyuncs.c om
```

#### 2. 推送镜像。

i. 执行以下命令, 查询镜像ID。

docker images

预期输出。示例如下:

REPOSITORY		TAG	IMAGE ID
CREATED	SIZE		
expresscart		latest	d6228399****
2 weeks ago	387 MB		

ii. 执行以下命令, 给镜像打标签。

docker tag <镜像ID> <企业版实例名称>-registry.<regionId>.cr.aliyuncs.com/<命名空间的名
称>/<镜像仓库的名称>:<镜像版本号>

iii. 执行以下命令, 推送镜像至企业版实例。

docker push <企业版实例名称>-registry.<regionId>.cr.aliyuncs.com/<命名空间的名称>/<镜像 仓库的名称>:<镜像版本号>

您可以观察到SAE应用会产生一条变更记录,推送的镜像被重新部署到了SAE。系统显示如下结果。

php_helloworld docker push		/php-demo:1.9
The push refers to repository		/php-demo]
3b92ecd55aec: Layer already e	xists	
2f830844d817: Layer already es	xists	
3e81cc345585: Layer already es	xists	
dd86a9ab0fc8: Layer already e	xists	
87594349fad5: Layer already es	xists	
4dcb3b64f43c: Layer already es	xists	
bb420ea46654: Layer already es	xists	
8b7ee4139985: Layer already ex	xists	
92fb81c7d22b: Layer already es	xists	
435f9e9c9e58: Layer already es	xists	
fc0b35927b5a: Layer already es	xists	
8c7a39edfbf0: Layer already e	xists	
8888f89886ba: Layer already es	xists	
f182865e86e9: Layer already es	xists	
b2bb5e569df9: Layer already es	xists	
d0fe97fa8b8c: Layer already es	xists	
1.9: digest: sha256:fb02e		size: 3658
php_helloworld		

#### 3. 结果验证。

您可以登录SAE控制台,在变更记录页面查看变更详情。

					Q 搜索文档、控制台、API、	解决方案和资源	費用 工单	备案	企业 支	持 官网	۶_	٥.	7 (	) 简体	0
应用列表 / 变更记录															
← 应用列	]表(	(php-demo)													
基本信息		变更记录													
变更记录		所有变更类型	✓ 所有変更状态	~	模糊搜索 >>	根据描述信息进行模糊	·搜索 Q								C
应用事件		创建时间	结束时间	变更类型	描述信息	变更状态		变更人		来源			操作		
日志管理	~	2020-10-15 11:23:14	2020-10-15 11:23:16	部署应用	Version: 1.9   Ima Name: php-demo	ige o 执行成I	b	主账号		pop			查看		
应用监控	~	2020-10-15 11:22:57	2020-10-15 11:22:59	部署应用	Version: 1.9   Ima Name: php-demo	ige • 执行成I	b	主账号		рор			查看		

关于Docker操作的更多信息,请参见Docker基本操作。

# 3.4. 使用SAE任务实现镜像构建及S2I解决 方案

Serverless应用引擎SAE(Serverless App Engine)支持短时任务,能够在短时间内快速创建大量计算任务, 并且在任务完成后快速释放计算资源。基于SAE短时任务实现镜像构建的实践,开发者无需构建常驻资源, 支持按量付费。本文介绍如何使用SAE任务,完成从任意Dockerfile到任意镜像仓库的Cl流程,实现S2l能力。

# 前提条件

构建镜像并上传至镜像仓库。您可以按需参考以下语言:

- 制作Java镜像
- 制作PHP镜像
- 制作Node.js镜像

### 背景信息

kaniko是一款用于构建容器镜像的谷歌开源工具,其工作原理是根据Dockerfile逐行执行命令,执行完毕后为 文件系统创建快照(Snapshot),并与上一个快照进行对比。如果内容不一致,便会创建一个新的层级, 并将所有修改都写入镜像的元数据中。当Dockerfile中每条命令执行完毕后,kaniko会将新生成的镜像推送到 指定的镜像仓库。更多信息,请参见kaniko。

除通过编写Dockerfile的方式之外,您还可以通过Source-To-Image(S2I)的方式构建Docker镜像。因此,通过SAE和kaniko,您可以借助Dockerfile构建镜像,实现S2I能力。

本文介绍的镜像构建实践支持按量付费。若镜像构建需要依赖公网拉取,则需VPC开放公网访问能力。更多 信息,请参见连接公网。

## 通过Dockerfile构建镜像

### 步骤一:在SAE控制台创建配置项

您需要创建配置项,分别对应需要构建的Dockerfile和镜像仓库的配置信息。本示例以华北2(北京)地域为例。

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击命名空间,在顶部菜单栏选择地域,单击具体命名空间名称。
- 3. 在左侧导航栏单击配置项, 在配置项页面单击创建。

4. 在创建配置项面板,选择手动创建,配置相关参数并单击确定。

#### 参数说明如下:

参数	示例值						
配置项名称	输入 build 。						
配置项描述	自定义。						
配置映射	<pre>配置以下两对键值对。 • 键: 输入 dockerfile , Dockerfile为示例Dockerfile。 • 值:     FROM registry-vpc.cn-beijing.aliyuncs.com/****/nginx:stable     CMD ["echo", "Hello DevOps"] • 键: 输入 secert , 为镜像仓库配置。 • 值: 格式如下。     {"auths":{"https://registry-vpc.cn-beijing.aliyuncs.com":     {"username":"****@test.aliyunid.com","password":"****}}</pre>						

您可以在配置项页面查看已成功创建的配置项。

# 步骤二:在SAE控制台创建任务

本示例以Java语言为例,仅介绍创建任务模板的关键参数,您可以按需选择语言。参数的更多信息,请参见任务模板管理。

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击任务模板列表,在顶部菜单栏选择地域。
- 3. 在任务模板列表页面单击创建任务模板。
- 4. 在任务基本信息页签, 配置相关参数并单击下一步: 部署配置。
- 右部署配置页签,配置相关参数并单击下一步:任务设置。
   其余配置保持默认设置。

参数	说明
技术栈语言	选择Java。
任务部署方式	选择 <b>镜像</b> 。
Java环境	仅技术栈语言选择Java时需设置。默认选择 <b>Open JDK 8</b> 。

参数	说明					
	在 <b>公有镜像</b> 页签输入镜像地址。格式如下:					
	registry-vpc. <reigonid>.aliyuncs.com/&lt;命名空间的名称&gt;/&lt;镜像仓库 的名称&gt;:&lt;镜像版本号&gt;</reigonid>					
	示例如下:					
配直镜像	<pre>registry-vpc.cn-beijing.aliyuncs.com/sae-test/kaniko- executor:1.0</pre>					
	⑦ 说明 该镜像同 gcr.io/kaniko-project/executor:v1.8.0					
	○ 启动命令: 输入 /kaniko/executor 。					
	• 启动参数:添加以下两条参数,其中 destination 为目标镜像仓库地址。					
启动命令设置	dockerfile=/workspace/Dockerfile					
	destination=registry-vpc.cn- beijing.aliyuncs.com/****/jar-test:job					
<b>エコ ccc かた ra</b>	引用创建的配置项 <b>build</b> ,并添加对应的键与挂载路径。					
配直官埋	<ul> <li>secret的挂载路径为/kaniko/.docker/config.json。</li> </ul>					

6. 在任务设置页签,按需配置定时规则和高级设置,并单击下一步:确认规格。

在确认规格页签单击确认创建。
 创建完成后,您可以在任务模板列表页面查看已创建的任务信息。

## 步骤三:执行任务

因当前SAE任务仅支持时间触发,在成功创建任务模板后,您需要采用手动执行的方式生成任务。

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击任务模板列表,在顶部菜单栏选择地域,单击具体任务名称。
- 在任务模板详情页面,单击手动执行任务,跳转至任务记录页面。
   新的镜像版本已成功推送至镜像仓库。
- 4. 验证结果。
  - 方式一: 在任务记录页面单击目标任务id, 在任务详情页面查看任务运行状态。
  - 方式二: 在目标任务左侧导航栏选择日志管理 > 实时日志, 在实时日志页面查看任务运行记录。
  - 方式三:登录容器镜像服务控制台,在目标实例镜像仓库内查看镜像版本是否已更新。

## 通过S2I构建镜像

您可以使用GitLab或者GitHub仓库托管您的代码,将本地Dockerfile改为Git中的Dockerfile。本示例以阿里云 Code为例进行验证。

#### Dockerfile示例如下:

FROM registry.cn-beijing.aliyuncs.com/hub-mirrors/maven:3-jdk-8 AS build COPY src /usr/src/app/src COPY pom.xml /usr/src/app COPY settings.xml /user/src/app/settings.xml RUN mvn -f /usr/src/app/pom.xml -s /user/src/app/settings.xml clean package -DskipTests from registry.cn-beijing.aliyuncs.com/hub-mirrors/openjdk:8-jdk-alpine VOLUME /tmp ARG JAR\_FILE=target/application.jar COPY --from=build /usr/src/app/\${JAR\_FILE} app.jar ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]

# 步骤一:设置Git Token

- 1. 登录<mark>阿里云Code</mark>。
- 2. 在左侧导航栏单击设置,然后单击个人设置页面左侧导航栏的私人令牌。
- 3. 在私人令牌页面,设置Git Token用于访问Git仓库。

# 步骤二: 创建任务

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击任务模板列表,在顶部菜单栏选择地域。
- 3. 在任务模板列表页面单击创建任务模板。
- 4. 在任务基本信息页签, 配置相关参数并单击下一步: 部署配置。
- 5. 在部署配置页签, 配置相关参数并单击下一步: 任务设置。

#### 其余配置保持默认设置。

参数	说明				
技术栈语言	选择Java。				
任务部署方式	选择 <b>镜像</b> 。				
Java环境	仅技术栈语言选择Java时需设置。默认选择 <b>Open JDK 8</b> 。				
启动命令设置	<ul> <li>启动命令:输入 /kaniko/executor 。</li> <li>启动参数:添加以下两条参数,其中 destination 为目标镜像仓库地址。         <ul> <li>context=git://code.aliyun.com/****/spring-boot.git</li> <li>⑦ 说明 上下文格式为 git://TOKEN@github.com/acme/myproj             ect.git#refs/heads/mybranch 。</li> <li>destination=registry-vpc.cn-             beijing.aliyuncs.com/****/jar-test:job</li> </ul> </li> </ul>				

参数	说明			
环境变量设置	<ul> <li>添加两条环境变量,分别标识设置的用户名和Git Token。</li> <li>类型:选择自定义。</li> <li>变量名称:分别输入 GIT_PASSWORD 和 GIT_USERNAME 。</li> <li>变量值/变量引用:分别输入步骤一设置的信息。</li> </ul>			

- 6. 在任务设置页签, 按需配置定时规则和高级设置, 并单击下一步: 确认规格。
- 在确认规格页签单击确认创建。
   创建完成后,您可以在任务模板列表页面查看已创建的任务信息。

## 步骤三:执行任务

因当前SAE任务仅支持时间触发,在成功创建任务模板后,您需要采用手动执行的方式生成任务。

- 1. 登录SAE控制台。
- 2. 在左侧导航栏单击任务模板列表,在顶部菜单栏选择地域,单击具体任务名称。
- 在任务模板详情页面,单击手动执行任务,跳转至任务记录页面。
   新的镜像版本已成功推送至镜像仓库。
- 4. 验证结果。
  - 方式一: 在任务记录页面单击目标任务id, 在任务详情页面查看任务运行状态。
  - 方式二: 在目标任务左侧导航栏选择日志管理 > 实时日志, 在实时日志页面查看任务运行记录。
  - 方式三:登录容器镜像服务控制台,在目标实例镜像仓库内查看镜像版本是否已更新。

# 3.5. 制作Node.js镜像

本文介绍如何在本地制作Node.js镜像,并以镜像方式将应用部署到Serverless应用引擎SAE(Serverless App Engine)。

# 前提条件

- 安装Docker。
- 安装Node.js。
- 开通阿里云SAE服务。
- 开通阿里云容器镜像ACR服务,个人版实例和企业版实例皆可。

### 步骤一:准备项目

- 1. 创建一个用于存放Node.js资源的项目文件夹,命名为node。
- 2. 执行以下命令,进入项目目录。

cd node

3. 创建并编辑 server.js 的文件, 文件内容如下:

```
const ronin = require( 'ronin-server' )
const mocks = require( 'ronin-mocks' )
const server = ronin.server()
server.use( '/', mocks.server( server.Router(), false, true ) )
server.start()
```

4. 创建并编辑 package.json 文件, 文件内容如下:

```
{
 "name": "node_sae_demo",
 "version": "1.0.0",
 "description": "",
 "main": "server.js",
 "scripts": {
   "test": "echo \"Error: no test specified\" && exit 1",
   "start": "node server.js"
 },
 "keywords": [],
 "author": "",
 "license": "ISC",
 "dependencies": {
   "ronin-mocks": "^0.1.4",
   "ronin-server": "^0.1.3"
 }
}
```

#### 5. 本地验证。

i. 执行以下命令, 安装依赖。

npm install --production

--production 表示只扫描 dependencies 节点下的依赖。

ii. 执行以下命令,运行项目。

node server.js

iii. 打开一个新的终端,执行以下命令,查看请求是否返回正常。

curl 'http://127.0.0.1:8000/test'

输出如下。

新终端:

{"code":"success","meta":{"total":0,"count":0},"payload":[]}

■ 现有终端,输出示例如下:

2021-\*\*-\*\*T14:36:05:1050 INFO: GET /test

# 步骤二:制作镜像

 在 *node*目录下,创建并编辑 *Dockerfile*文件,文件内容如下: 本文以Node.js 12.18.1为例。

```
FROM node:12.18.1
ENV NODE_ENV=production
WORKDIR /app
COPY ["package.json", "./"]
# 可选, 您也可以将本地的node_modules拷贝进来,而不是制作镜像的过程中再安装依赖,会拖慢镜像的构建。
# RUN COPY node_modules .
RUN npm install --production
COPY . .
# 如果没用pm2,可以不需要docker-entrypoint.sh。
COPY docker-entrypoint.sh /usr/local/bin/
CMD [ "node", "server.js" ]
```

2. 在 node 目录下, 创建并编辑 docker-ent rypoint.sh文件, 文件内容如下:

```
#!/bin/sh
# https://github.com/nodejs/docker-node/blob/main/docker-entrypoint.sh
set -e
if [ "${1#-}" != "${1}" ] || [ -z "$(command -v "${1}")" ]; then
    set -- node "$0"
fi
# 启动命令前先清理部分对Node.js无用的环境变量,避免部分框架 (PM2等) 因为环境变量过长而导致系统崩溃
o
echo "remove options"
unset JAVA_TOOL_OPTIONS
exec "$0"
```

3. 在容器镜像服务控制台创建镜像仓库。

个人版和企业版实例均适用本文的操作,本文以个人版实例为例。具体操作,请参见以下文档:

- 个人版实例: 构建仓库与镜像
- 企业版实例:使用企业版实例构建镜像
- 4. 执行以下命令,构建并推送镜像。
  - 您可以在目标镜像仓库的基本信息页面的镜像指南页签查询具体命令。更多信息,请参见以下文档:
  - 个人版实例: 使用个人版实例推送拉取镜像
  - 企业版实例:使用企业版实例推送和拉取镜像
    - i. 执行以下命令, 构建镜像。

docker build --tag node-demo:v1 .

输出示例如下:

```
Sending build context to Docker daemon 12.82MB
Step 1/8 : FROM node:12.18.1
---> f5be1883****
.....
Step 8/8 : CMD [ "node", "server.js" ]
---> Running in e96b27dd****
Removing intermediate container e96b27dd****
---> 73028a1d****
Successfully built 73028a1d****
```

ii. 执行以下命令, 登录远端镜像仓库。

本步骤假设您使用的是阿里云ACR仓库。

docker login --username=<镜像仓库登录名> registry.<regionId>.aliyuncs.com

示例如下:

docker login --username=\*\*\*\*@188077086902\*\*\*\* registry.cn-hangzhou.aliyuncs.com

在返回结果中输入密码,如果显示 login succeeded ,则表示登录成功。如何设置密码,请参见设置镜像仓库登录密码。

iii. 执行以下命令, 给镜像打标签。

docker tag <ImageId> registry.<regionId>.aliyuncs.com/\*\*\*\*/node-demo:<镜像版本号>

- ImageId : 镜像ID。
- registry.<regionId>.aliyuncs.com/\*\*\*\*/node-demo : 镜像仓库地址。

#### 示例如下:

docker tag node-demo:vl registry.cn-hangzhou.aliyuncs.com/\*\*\*\*/node-demo:vl

iv. 执行以下命令, 推送镜像至个人版实例。

docker push registry.<regionId>.aliyuncs.com/\*\*\*\*/node-demo:<镜像版本号>

#### 示例如下:

docker push registry.cn-hangzhou.aliyuncs.com/\*\*\*\*/node-demo:v1

成功推送后,您可以登录容器镜像服务控制台,在目标镜像仓库的镜像版本页面查看推送的版本。

# 步骤三:部署镜像

1. 创建Node.js应用。

在**应用部署配置**配置向导页面,技术栈语言需选择其它语言,应用部署方式需选择镜像。具体操 作,请参见在SAE控制台使用镜像部署多语言应用。

<ul> <li>○</li> <li>□</li> <li>□</li></ul>	2 应用部者配置		- (4) STREERS
* 技术栈语言: 〇 Java	) рнр	④ 其它语言	
• 应用部署方式:			
配置镜像 'registry-vpc.cn-hangzhou.aliyuncs.cc ode-de	ma:v1		
我的阿里云镜像 Demo镜像 公有镜像 其它阿里云器	长号私有镜像		
● 镜像服务个人版 镜像服务企业版			
如果您应用所在的region和阿里云镜像仓库不在同一个region,需要跨regi	on pull镜像,请在ACR/ <sup>a</sup> 品进行配置,宣看详情。		
镜像仓库命名空间: ダイン	Q		
	类型:PRIVATE	来题:ALLHUB V1	~

⑦ 说明 除创建新应用外,您还可以修改现有应用的配置镜像。具体操作,请参见升级应用。

2. 为Node.js应用绑定公网SLB。

选择HTTP协议,并将HTTP端口设置为80,容器端口设置为8000。具体操作,请参见为应用绑定 SLB。

TCP协议	HTTP协议	HTTPS协议		
网络协议		HTTP端口	容器端口	操作
HTTP		80	8000	Ē

绑定完成后,您可以在应用访问设置区域查看公网访问地址。

∨ 应用访问设置	
● 使用SAE SLB功能之前,请先查看SLB前置约束,典型SLB场展最佳交話 自2020年4月1日,凡是在SAE产品中创建/编辑SLB后,再去SLB产品中修改监听读口、协议、SSL证书、SLB tag、虚拟服务器组合	等SLB配置约束中限制提作的资源,修改后的配置都会被SAE保存的配置信息强制覆盖,请一定注意避免。
科网访问地址: 添加科网SLB访问	公网访问地址: ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

3. 结果验证。

按照http://slbip:port/的格式,在浏览器地址栏输入步骤2设置的公网地址并回车。结果如下:



# 更多信息

如果您不想每次都通过手动方式构建、推送镜像,并部署应用,可以对接开源的Jenkins或者阿里云云效。具体信息,请参见以下文档:

- 部署Node.js应用至SAE
- 使用Jenkins创建多语言应用的持续集成

# 3.6. 制作PHP镜像

本文介绍如何制作PHP镜像并部署到Serverless应用引擎SAE(Serverless App Engine)。

## 前提条件

- 安装Docker。
- 开通阿里云SAE服务。
- 开通阿里云容器镜像ACR服务,个人版实例和企业版实例皆可。

# 背景信息

通常来说,您无需从空镜像开始安装PHP以及其他各种组件,而是可以通过dockerhub获取PHP官方镜像。官方的系统镜像是Debian或Alpine(Alpine镜像的体积小于Debian镜像),其中已经包含PHP以及一些辅助命令。官方镜像的版本命令规则如下:

- php:8.0.12-cli-bullseye
   : 表示PHP版本为8.0.12,基于系统镜像bullseye(Debian 11)进行构建。
   其中 cli 表示未开启CGI,即不能运行 fpm ,只可运行命令行。
- php:7.3.32-fpm-alpine3.13
   : 表示PHP版本为7.3.32,基于系统镜像alpine3.13进行构建。其中 fpm 表示开启了CGI,可用来运行Web服务,也可用来运行CLI命令。

## 步骤一:制作基础镜像

基础镜像里包含常用的扩展和配置,在后续每次构建时就无需重复执行这些步骤。

- 1. 创建项目文件。
  - i. 创建一个用于存放PHP资源的项目文件夹,命名为php。
  - ii. 在PHP文件夹下,创建一个用于存放基础镜像资源的项目文件夹,命名为base。
- 2. 执行以下命令,进入项目目录。

cd php/base

3. 创建并编辑 Dockerfile 文件, 文件内容如下:

```
FROM php:7.3.32-fpm-alpine3.13
LABEL MAINTAINER="sae@aliyun.com"
ENV TZ "Asia/Shanghai"
# 时区
RUN echo ${TZ} >/etc/timezone
# 可选
# COPY composer.phar /usr/local/bin/composer
# 创建www用户
RUN addgroup -g 1000 -S www && adduser -s /sbin/nologin -S -D -u 1000 -G www www
# 配置阿里云镜像源,加快构建速度。
RUN sed -i "s/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g" /etc/apk/repositories
# PHPIZE_DEPS包含gcc g++等编译辅助类库,完成后删除;pecl安装扩展。
RUN apk add --no-cache PHPIZE DEPS \setminus
   && apk add --no-cache libstdc++ libzip-dev vim
   && apk update \
   && pecl install redis-5.3.0 \
   && pecl install zip \
   && pecl install swoole \setminus
   && docker-php-ext-enable redis zip swoole\
   && apk del $PHPIZE DEPS
# docker-php-ext-install安装扩展。
RUN apk update \
    && apk add --no-cache nginx freetype libpng libjpeg-turbo freetype-dev libpng-dev l
ibjpeg-turbo-dev \
   && docker-php-ext-configure gd --with-freetype-dir=/usr/include/ --with-jpeg-dir=/u
sr/include/ --with-png-dir=/usr/include/ \
    && docker-php-ext-install -j$(nproc) gd \
    && docker-php-ext-install -j$(nproc) pdo_mysql opcache bcmath mysqli
# 在run.sh
COPY run.sh /run.sh
RUN mv "$PHP INI DIR/php.ini-production" "$PHP INI DIR/php.ini" && mkdir -p /run/nginx/
&& chmod +x /run.sh
ENTRYPOINT ["/run.sh"]
```

- ⑦ 说明 该镜像文件同时采用pecl和docker-php-ext-install安装扩展,二者差异如下:
  - pecl可以自动下载或者指定包安装扩展,安装完以后需要用docker-php-ext-enable启用扩展。
  - docker-php-ext-install可以安装的扩展有限,可以通过docker-php-ext-install help查询可以安装的扩展列表,安装以后默认启用。

#### 4. (可选)修改镜像。

⑦ 说明 步骤1的基础镜像将Nginx和PHP配置在同一个镜像内,同时启动php-fpm和nginx,符 合常见的虚拟机部署习惯,但并不满足容器提倡的单进程管理模型。您可以考虑将php-fpm和 nginx分别部署成两个应用,只需要针对上述镜像做少许修改即可。

#### 创建并编辑run.sh文件, 文件内容如下:

```
#!/usr/bin/env sh
set -e
php-fpm -D
nginx -g 'daemon off;'
```

#### 5. 执行以下命令,构建镜像。

docker build -t php-bash:v1 .

#### 输出示例如下:

```
Sending build context to Docker daemon 4.096kB
Step 1/11 : FROM php:7.3.32-fpm-alpine3.13
---> 2e127e9a****
.....
Step 11/11 : ENTRYPOINT ["/run.sh"]
---> Using cache
---> dfde0cef****
Successfully built dfde0cef****
Successfully built dfde0cef****
```

# 步骤二:制作业务镜像

您可以按需对<mark>步骤一</mark>的镜像进行部分修改,以此作为您企业内使用的基础镜像,则后续的业务镜像都可以将 其作为父镜像。

1. 在 php项目目录下, 创建 Dockerfile文件。

```
FROM php-bash:v1
# 把阿里云镜像地址换成内网地址,不开公网就可以安装各种软件。
RUN sed -i "s/mirrors.aliyun.com/mirrors.cloud.aliyuncs.com/g" /etc/apk/repositories &k
sed -i "s/https/http/g" /etc/apk/repositories
# php-fpm配置
COPY www.conf /usr/local/etc/php-fpm.d/www.conf
# Nginx配置
COPY default.conf /etc/nginx/http.d/
COPY index.php /var/www/html
```

2. 创建并编辑www.conf文件, 文件内容如下:

```
[www]
user = www
group = www
listen = 0.0.0.0:9000
pm = dynamic
pm.max_children = 100
pm.start_servers = 30
pm.min_spare_servers = 20
pm.max_spare_servers = 50
```

#### 3. 创建并编辑 def ault.conf 文件, 文件内容如下:

```
server {
  listen 80;
  root /var/www/html;
  index index.html index.htm index.php;
  location ~ .*\.(php|php5)?$
  {
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
  }
  access_log /tmp/nginx_access.log;
  error_log /tmp/nginx_error.log;
}
```

4. 创建并编辑 index.php文件, 文件内容如下:

```
<html>
<head>
<title>PHP Hello SAE!</title>
</head>
<body>
<?php echo '<h1>Hello SAE</h1>'; ?>
<?php phpinfo(); ?>
</body>
</html>
```

5. 在容器镜像服务控制台创建两个镜像仓库,分别命名为php-base和php-server。

个人版和企业版实例均适用本文的操作,本文以个人版实例为例。具体操作,请参见以下文档:

- 个人版实例: 构建仓库与镜像
- · 企业版实例: 使用企业版实例构建镜像
- 6. 构建并推送镜像。

您可以在目标镜像仓库的基本信息页面的镜像指南页签查询具体命令。更多信息,请参见以下文档:

- 个人版实例: 使用个人版实例推送拉取镜像
- 企业版实例:使用企业版实例推送和拉取镜像

i. 执行以下命令, 构建镜像。

docker build -t php-server:v1 .

#### 输出示例如下:

```
Sending build context to Docker daemon 5.12kB
Step 1/5 : FROM php-bash:v1
---> dfdeOcef****
.....
Step 5/5 : COPY index.php /var/www/html
---> Using cache
---> e2c25424****
Successfully built e2c25424****
Successfully tagged php-server:v1
```

ii. 执行以下命令,登录远端镜像仓库。

本步骤假设您使用的是阿里云ACR仓库。

docker login --username=<镜像仓库登录名> registry.<regionId>.aliyuncs.com

示例如下:

docker login --username=\*\*\*\*@188077086902\*\*\*\* registry.cn-hangzhou.aliyuncs.com

在返回结果中输入密码,如果显示 login succeeded ,则表示登录成功。如何设置密码,请参见设置镜像仓库登录密码。

iii. 执行以下命令, 给镜像打标签。

docker tag [ImageId] registry.<regionId>.aliyuncs.com/\*\*\*\*/php-server:<镜像版本号>

- ImageId : 镜像ID。
- registry.<regionId>.aliyuncs.com/\*\*\*\*/php-server : 镜像仓库地址。

#### 示例如下:

docker tag php-server:v1 registry.cn-hangzhou.aliyuncs.com/php/php-server:v1

iv. 执行以下命令, 推送镜像至个人版实例。

docker push registry.<regionId>.aliyuncs.com/php/php-server:<镜像版本号>

示例如下:

docker push registry.cn-hangzhou.aliyuncs.com/php/php-server:v1

成功推送后,您可以登录容器镜像服务控制台,在目标镜像仓库的镜像版本页面查看推送的版本。

#### 步骤三:部署镜像

1. 创建PHP应用。

在**应用部署配置**配置向导页面,**技术栈语言**需选择PHP,应用部署方式需选择**镜像**。具体操作,请参 见<mark>在SAE控制台使用镜像部署PHP应用</mark>。

#### Serverless 应用引擎

应用基本信	8	应用部署配置			
• 技术栈语言:	) Java	PHP	○ 其它语言		
•技术栈版本:	PHP-7.3 ~				
* 应用部署方式:	● 镜像	○ ZIP包部署			
配置镜像 *registry-	vpc.cn-hangzhou.aliyuncs				
我的阿里云镜像	Demo镜像 公有镜像 其它阿里云账号私有镜像				
镜像服务个人版	○ 镜像服务企业级				
< 如果您应用所在的r	egion和阿里云镜像仓库不在同一个region,需要跨region pull镜像,请在	ACR/*品进行配置,查看详情。			
镜像仓库命名空间:	→ 镜像仓库名: php	Q			
4	php-base	类型:PRIVATE	来源:ALI_HUB	请选择	~
<u>s</u>	/php-server	类型:PRIVATE	来源:ALI_HUB	۲۷.	~
					Ţ

⑦ 说明 除创建新应用外,您还可以修改现有应用的配置镜像。具体操作,请参见升级应用。

2. 为PHP应用绑定公网SLB。

选择HTTP协议,并将HTTP端口设置为80,容器端口设置为80。具体操作,请参见为应用绑定SLB。

TCP协议 HTTP	办议 HTTPS协议		
网络协议	НТТР端口	容器演口	操作
HTTP	80	80	Ē

绑定完成后,您可以在应用访问设置区域查看公网访问地址。

∨ 应用访问设置	
使用SAE SLB功能之前,请先查看SLB配置约束 典型SLB场景最佳实践 自2020年4月1日,凡是在SAE产品中创建/编辑SLB后,再去SLB产品中修改监听端口、协议、SSL证书、SLB tag、虚拟服务器组名	等SLB配置约束中限制操作的资源,修改后的配置都会被SAE保存的配置信息强制愿盖,请一定注意避免。
¥山网站的回地站上: 清赤力FAU网SLB\动问	

3. 结果验证。

按照http://slbip:port/的格式,在浏览器地址栏输入步骤2设置的公网地址并回车。结果如下:

	S PHP He	llo SAE!	×	+			
$\leftarrow \   \rightarrow$	C A 不	安全	4.20				
Hello	SAE						
					PHP Versi	on 7.3.32	

# 更多信息

您如果不想每次都是手动构建、推送、部署,也可以考虑对接开源的jenkins或者阿里云云效。具体信息,请参见以下文档:

- 部署PHP应用至SAE
- 使用Jenkins创建多语言应用的持续集成

3.7. 制作Java镜像

Spring Cloud、Dubbo、HSF框架下开发并编译的应用WAR包或JAR包,如果需要在Serverless应用引擎 SAE(Serverless App Engine)上以镜像方式部署,需要将WAR包或JAR包制作为应用镜像,以便部署时使 用。本文介绍镜像制作的使用规范、以不同代码包编译的Java应用的镜像制作过程、以及将镜像上传至镜像 仓库的操作步骤。

# 使用规范

通过Dockerfile制作自定义镜像时, SAE在运行时会自动注入以下环境变量信息。

### ↓ 注意

- 为保证应用正常运行,请勿覆盖以下配置。
- 请勿使用/home/admin/.spas\_key/default文件,该文件在POD运行时将会被覆盖。
- 请勿使用VOLUMES或USER root字段,否则应用运行时可能会导致文件丢失。

环境变量Key	描述
POD_IP	POD的IP地址。
EDAS_APP_ID	部署在SAE上应用的ID。
EDAS_ECC_ID	导入集群的ECS在应用中部署后的ID。
EDAS_PROJECT_NAME	与EDAS_ECC_ID类似,用于调用链解析。
EDAS_JM_CONTAINER_ID	与EDAS_ECC_ID类似,用于调用链解析。
EDAS_CATALINA_OPT S	中间件运行时所需的CATALINA_OPTS参数。
CATALINA_OPTS	与EDAS_CATALINA_OPTS类似,默认Tomcat启动参数。
CATALINA_HOME	安装的Tomcat路径。
PANDORA_LOCATION	安装的Pandora路径,仅HSF应用中可见。

# Spring Cloud或Dubbo应用的Dockerfile示例(基于JAR包)

FROM centos:7 MAINTAINER: SAE研发团队 #安装打包必备软件。 RUN yum -y install wget unzip telnet #准备JDK以及Tomcat系统变量。 ENV JAVA HOME /usr/java/latest ENV PATH \$PATH:\$JAVA HOME/bin ENV ADMIN HOME /home/admin #**下载安装**OpenJDK。 RUN yum -y install java-1.8.0-openjdk-devel #下载部署SAE演示JAR包。 RUN mkdir -p /home/admin/app/ && \ wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/demo/1.0/hello-edas-0.0.1-SNAPSHO T.jar -O /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar #增加容器内中文支持。 ENV LANG="en\_US.UTF-8" #增强Webshell使用体验。 ENV TERM=xterm #将启动命令写入启动脚本start.sh。 RUN mkdir -p /home/admin RUN echo 'eval exec java -jar \$CATALINA OPTS /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar' > /home/admin/start.sh && chmod +x /home/admin/start.sh WORKDIR \$ADMIN HOME CMD ["/bin/bash", "/home/admin/start.sh"]

# ? 说明

- 如果您需要使用微服务能力,则在设置启动命令时,需添加 <u>\$CATALINA\_OPTS</u>启动参数。具体 操作,请参见设置启动命令。
- 如果您需要使用自建的注册中心或MSE的注册中心,则需配置启动参数 -Dnacos.use.endpoint.
   parsing.rule=false 和 -Dnacos.use.cloud.namespace.parsing=false 。参数需要放在 -j ar 之前,否则可能会导致无法使用非SAE自带的注册中心。其他具体详情,请参见自建Nacos 服务注册中心和使用MSE的Nacos注册中心。

# Spring Cloud或Dubbo应用的Dockerfile示例(基于WAR包)

```
FROM centos:7
MAINTAINER: SAE研发团队
#安装打包必备软件。
RUN yum -y install wget unzip telnet
#准备JDK以及Tomcat系统变量。
ENV JAVA HOME /usr/java/latest
ENV CATALINA HOME /home/admin/apache-tomcat-7.0.91
ENV ADMIN HOME /home/admin
ENV PATH $PATH:$JAVA HOME/bin:$CATALINA HOME/bin
RUN mkdir -p /home/admin
#下载安装OpenJDK。
RUN yum -y install java-1.8.0-openjdk-devel
#下载安装Tomcat。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/apache-tomcat-7.0.91.tar.gz -0 /tmp/ap
ache-tomcat-7.0.91.tar.gz && \
   tar -xvf /tmp/apache-tomcat-7.0.91.tar.gz -C /home/admin && \
   rm /tmp/apache-tomcat-7.0.91.tar.gz && \
   chmod +x ${CATALINA HOME}/bin/*sh
RUN mkdir -p ${CATALINA HOME}/deploy/
#增加容器内中文支持。
ENV LANG="en US.UTF-8"
#增强Webshell使用体验。
ENV TERM=xterm
#下载部署SAE演示WAR包。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/demo/1.0/hello-edas.war -0 /tmp/ROOT.w
ar && \
   rm -rf ${CATALINA HOME}/webapps/ROOT &&\
   unzip /tmp/ROOT.war -d ${CATALINA_HOME}/deploy/ROOT/ && \
   rm -rf /tmp/ROOT.war
# 设定Tomcat安装目录为容器启动目录,并采用run方式启动Tomcat,在标准命令行输出catalina日志。
WORKDIR $ADMIN HOME
CMD ["catalina.sh", "run"]
```

# HSF应用的Dockerfile示例(基于JAR包)

FROM centos:7 MAINTAINER: SAE研发团队 #安装打包必备软件。 RUN yum install -y wget unzip telnet lsof net-tools bind-utils #准备JDK及Tomcat系统变量与路径。 ENV JAVA HOME /usr/java/latest ENV CATALINA HOME /home/admin/taobao-tomcat ENV PATH \${JAVA HOME}/bin:\${PATH} ENV ADMIN HOME /home/admin #设置EDAS-Container以及Pandora应用容器版本。 ENV EDAS CONTAINER VERSION V3.5.4 LABEL pandora V3.5.4 #下载安装OpenJDK。 RUN yum -y install java-1.8.0-openjdk-devel #创建JAVA HOME软链接。 RUN if [ ! -L "\${JAVA\_HOME}" ]; then mkdir -p `dirname \${JAVA\_HOME}` && ln -s `readlink -f /usr/lib/jvm/java` \${JAVA HOME}; fi #根据环境变量,下载安装EDAS-Container以及Pandora应用容器版本。 RUN mkdir -p \${CATALINA HOME}/deploy RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-plugins/edas.sar.\${EDAS CONTAINER \_VERSION}/taobao-hsf.tgz -0 /tmp/taobao-hsf.tgz && \ tar -xvf /tmp/taobao-hsf.tgz -C \${CATALINA\_HOME}/deploy/ && \ rm -rf /tmp/taobao-hsf.tgz #下载部署SAE演示JAR包。 RUN mkdir -p \${ADMIN HOME}/app && wget http://edas.oss-cn-hangzhou.aliyuncs.com/demoapp/fat jar-test-case-provider-0.0.1-SNAPSHOT.jar -0 \${ADMIN HOME}/app/provider.jar #将启动命令写入启动脚本start.sh。 RUN echo 'eval exec java -jar \$CATALINA\_OPTS /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar' > /home/admin/start.sh && chmod +x /home/admin/start.sh WORKDIR \$CATALINA HOME CMD ["/bin/bash", "/home/admin/start.sh"]

? 说明

- 如果您需要使用微服务能力,则在设置启动命令时,需添加<u>\$CATALINA\_OPTS</u>启动参数。具体 操作,请参见设置启动命令。
- 如果您需要使用自建的注册中心或MSE的注册中心,则需配置启动参数 -Dnacos.use.endpoint.
   parsing.rule=false 和 -Dnacos.use.cloud.namespace.parsing=false 。参数需要放在 -j ar 之前,否则可能会导致无法使用非SAE自带的注册中心。其他具体详情,请参见自建Nacos 服务注册中心和使用MSE的Nacos注册中心。

# HSF应用的Dockerfile示例(基于WAR包)

```
FROM centos:7
MAINTAINER: SAE研发团队
#安装打包必备软件。
RUN yum install -y wget unzip telnet lsof net-tools bind-utils
#准备JDK以及Tomcat系统变量与路径。
ENV JAVA HOME /usr/java/latest
ENV CATALINA HOME /home/admin/taobao-tomcat
ENV PATH ${JAVA HOME}/bin:${CATALINA HOME}/bin:${PATH}
#设置EDAS-Container以及Pandora应用容器版本。
ENV EDAS_CONTAINER_VERSION V3.5.4
LABEL pandora V3.5.4
#下载安装OpenJDK。
RUN yum -y install java-1.8.0-openjdk-devel
#创建JAVA HOME软链接。
RUN if [ ! -L "${JAVA HOME}" ]; then mkdir -p `dirname ${JAVA HOME}` && ln -s `readlink -f
/usr/lib/jvm/java` ${JAVA_HOME}; fi
#下载安装Ali-Tomcat 7.0.92到/home/admin/taobao-tomcat目录。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-container/7.0.92/taobao-tomcat-pr
oduction-7.0.92.tar.gz -0 /tmp/taobao-tomcat.tar.gz && \
   mkdir -p ${CATALINA HOME} && \
   tar -xvf /tmp/taobao-tomcat.tar.gz -C ${CATALINA HOME} && \
   mv ${CATALINA_HOME}/taobao-tomcat-production-7.0.59.3/* ${CATALINA_HOME}/ && \
   rm -rf /tmp/taobao-tomcat.tar.gz ${CATALINA HOME}/taobao-tomcat-production-7.0.59.3 &&
   chmod +x ${CATALINA HOME}/bin/*sh
#根据环境变量,下载安装EDAS-Container以及Pandora应用容器版本。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/edas-plugins/edas.sar.${EDAS CONTAINER
_VERSION}/taobao-hsf.tgz -0 /tmp/taobao-hsf.tgz && \
   tar -xvf /tmp/taobao-hsf.tgz -C ${CATALINA HOME}/deploy && \
   rm -rf /tmp/taobao-hsf.tgz
#下载部署SAE演示WAR包。
RUN wget http://edas.oss-cn-hangzhou.aliyuncs.com/demo/hello-edas.war -0 /tmp/ROOT.war && \
   unzip /tmp/ROOT.war -d ${CATALINA HOME}/deploy/ROOT && \
   rm -rf /tmp/ROOT.war
#设定Tomcat安装目录为容器启动目录,并采用run方式启动Tomcat,在标准命令行输出catalina日志。
WORKDIR ${CATALINA HOME}
CMD ["catalina.sh", "run"]
```

# 自定义设置Dockerfile

通过编辑Dockerfile文件修改运行环境配置,如更换JDK版本、Tomcat修改配置和更改运行时环境等操作。

• 更换JDK版本

在标准Dockerfile中,请参见以下示例更换其他版本的JDK。

```
# 下载安装JDK 8。
RUN wget http://edas-hz.oss-cn-hangzhou.aliyuncs.com/agent/prod/files/jdk-8u65-linux-x64.
rpm -0 /tmp/jdk-8u65-linux-x64.rpm && \
    yum -y install /tmp/jdk-8u65-linux-x64.rpm && \
    rm -rf /tmp/jdk-8u65-linux-x64.rpm
```

● 在Tomcat启动参数中添加SAE运行环境

SAE提供了JVM环境变量EDAS\_CATALINA\_OPTS,包含了运行所需的基本参数。此外Ali-Tomcat提供了自定义JVM参数配置选项JAVA\_OPTS,可以设置Xmx、Xms等参数。

```
#设置SAE应用JVM参数。
ENV CATALINA_OPTS ${EDAS_CATALINA_OPTS}
#设置JVM参数。
ENV JAVA_OPTS="\
-Xmx3550m \
-Xms3550m \
-Xmn2g"
```

• 使用独立注册中心

如果您希望在微服务应用中使用独立的注册中心,请在Java命令后增加以下两个参数。

-Dnacos.use.endpoint.parsing.rule=false

-Dnacos.use.cloud.namespace.parsing=false

#### 示例代码如下:

RUN echo 'eval exec java -Dnacos.use.endpoint.parsing.rule=false -Dnacos.use.cloud.namesp ace.parsing=false -jar \$CATALINA\_OPTS /home/admin/app/hello-edas-0.0.1-SNAPSHOT.jar'> /home/admin/start.sh && chmod +x /home/admin/start.sh

## 构建镜像

从本地命令行进入Dockerfile文件所在的目录,执行 docker build 命令制作镜像。

```
docker build -t <标签名称,最好取应用名>:<版本号> .
```

```
或
```

```
docker build -t <标签名称,最好取应用名>:<版本号> -f /path/to/custom_dockerfile_name.#适用于您创
建好的Dockerfile在其他位置或名称不为Dockerfile的场景。
```

#### 例如:

docker build -t hsf-provider:1.0.0 .

制作完成后,执行 docker images | grep <镜像标签名称> 命令查看本地编译完成的镜像。

## 上传镜像到镜像仓库

您可以将本地生成的应用镜像上传到阿里云提供的容器镜像仓库。详细步骤,请参见管理镜像。

执行指定镜像仓库基本信息页面提供的将镜像推送到Registry相关命令,将本地镜像上传到该镜像仓库。

```
docker login --username=<当前登录的账号名> registry.<regionId>.aliyuncs.com #输入的密码为阿
里云容器镜像服务默认实例访问凭证页面设置的固定或者临时密码,并非阿里云账号密码。
docker tag <本地应用镜像ID> registry.<regionId>.aliyuncs.com/<命令空间名称>/<镜像仓库名称>:<镜像
版本号>
docker push registry.<regionId>.aliyuncs.com/<命令空间名称>/<镜像仓库名称>:<镜像版本号>
```

例如:

docker login --username=\*\*\*\*@188077086902\*\*\*\* registry.cn-hangzhou.aliyuncs.com
docker tag <ImageId> registry.cn-hangzhou.aliyuncs.com/webapps/hsf-provider:1.0.0
docker push registry.cn-hangzhou.aliyuncs.com/webapps/hsf-provider:1.0.0

基本信息 编辑							
仓库名称	-						
Repo ID		专有网络 👔	-registry-vpc.cn-hangzhou.cr.aliyuncs.com, 复制				
仓库地域	华东1 (杭州)	公网地址 🚱	-registry.cn-hangzhou.cr.aliyuncs.com 复制				
仓库类型	私有	代码仓库	https://github.com				
镜像版本	可覆盖	摘要 🕜	12				
镜像加速	未加速						
镜像指南 Chart 指南●	OCI 制品指南 ● 制品描述						
1. 登录企业版实例							
\$ docker loginuse	ername=	om					
用于登录的用户名为阿里云则	张号全名,密码为开通服务时设置的密码。						
您可以在访问凭证页面修改外	您可以在访问凭证页面修改凭证密码。						
2. 从Registry中拉取镜像							
\$ docker pull — -registry.cn-hangzhou.cr.aliyuncs.com/————————————————[镜像版本号]							
3. 将镜像推送到Registry							
\$ docker login( registry.cn-hangzhou.cr.aliyuncs.com \$ docker tag [ImageId] registry.cn-hangzhou.cr.aliyuncs.com/ :[俵像版本号] \$ docker pushregistry.cn-hangzhou.cr.aliyuncs.com/ :[俵像版本号]							
请根据实际镜像信息蓄换示例中的I[mageld]和[镜像版本号]参数。							

# 更多信息

镜像制作完成后,您可以将其以镜像的方式部署在SAE。具体操作,请参见以下文档:

- 在SAE控制台使用镜像方式部署Java应用
- 在SAE控制台使用镜像部署PHP应用
- 在SAE控制台使用镜像部署多语言应用

# 3.8. 制作Go镜像

本文介绍如何在本地制作Docker镜像,并以镜像方式将Go应用部署到Serverless应用引擎SAE(Serverless App Engine)。

# 前提条件

- 安装Docker。
- 开通阿里云SAE服务。
- 开通阿里云容器镜像ACR服务并创建实例,个人版实例和企业版实例皆可。

## 步骤一:准备项目

1. 下载Demo。

Demo内的Dockerfile内容如下:

# Golang版本; Alpine镜像的体积较小。 FROM golang:1.16.6-alpine3.14 as builder # 替换Alpine镜像,方便安装构建包。 RUN sed -i 's/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g' /etc/apk/repositories # 安装构建阶段的依赖。 RUN apk --update add gcc libc-dev upx ca-certificates && update-ca-certificates # 将代码复制到构建镜像中。 # 注意地址不要在GOPATH中。 ADD . /workspace WORKDIR /workspace # 挂载构建缓存。 # GOPROXY防止下载失败。 RUN --mount=type=cache,target=/go \ env GOPROXY=https://goproxy.cn,direct \ go build -buildmode=pie -ldflags "-linkmode external -extldflags -static -w" \ -o /workspace/gin-hello-world # 运行时镜像。 # Alpine兼顾了镜像大小和运维性。 FROM alpine:3.14 EXPOSE 8080 # 方便运维人员安装需要的包。 RUN sed -i 's/dl-cdn.alpinelinux.org/mirrors.aliyun.com/g' /etc/apk/repositories # 创建日志目录等。 # RUN mkdir /var/log/onepilot -p && chmod 777 /var/log/onepilot && touch /var/log/onepi lot/.keep # 复制构建产物。 COPY -- from=builder /workspace/gin-hello-world /app/ # 指定默认的启动命令。 CMD ["/app/gin-hello-world"]

2. 在Demo所在目录,执行以下命令,构建镜像。

```
docker build . -t gin-example
```

- 3. 本地验证。
  - i. 指定端口映射,格式为 主机端口:容器端口 ,运行镜像。

docker run -p 127.0.0.1:8080:8080/tcp gin-example

ii. 执行以下命令, 查看请求是否返回正常。

curl http://127.0.0.1:8080/

输出如下:



# 步骤二:制作镜像

1. 在容器镜像服务控制台创建镜像仓库。

个人版和企业版实例均适用本文的操作,本文以个人版实例为例。具体操作,请参见以下文档:

• 个人版实例: 构建仓库与镜像

企业版实例:使用企业版实例构建镜像

2. 构建并推送镜像。

您可以在目标镜像仓库的基本信息页面的镜像指南页签查询具体命令。更多信息,请参见以下文档:

- 个人版实例:使用个人版实例推送拉取镜像
- 企业版实例:使用企业版实例推送和拉取镜像
  - i. 执行以下命令,构建镜像。

docker build --tag go-demo:v1 .

ii. 执行以下命令, 登录远端镜像仓库。

docker login --username=<镜像仓库登录名> registry.<regionId>.aliyuncs.com

示例如下:

docker login --username=\*\*\*\*@188077086902\*\*\*\* registry.cn-hangzhou.aliyuncs.com

在返回结果中输入密码,如果显示 login succeeded ,则表示登录成功。如何设置密码,请参 见设置镜像仓库登录密码。

iii. 执行以下命令, 给镜像打标签。

docker tag <ImageId> registry.<regionId>.aliyuncs.com/\*\*\*\*/go-demo:<镜像版本号>

- ImageId : 镜像D。
- registry.<regionId>.aliyuncs.com/\*\*\*\*/go-demo : 镜像仓库地址。

```
示例如下:
```

docker tag go-demo:v1 registry.cn-hangzhou.aliyuncs.com/\*\*\*\*/go-demo:v1

iv. 执行以下命令, 推送镜像至个人版实例。

docker push registry.<regionId>.aliyuncs.com/\*\*\*\*/go-demo:<镜像版本号>

#### 示例如下:

docker push registry.cn-hangzhou.aliyuncs.com/\*\*\*\*/go-demo:v1

成功推送后,您可以登录容器镜像服务控制台,在目标镜像仓库的镜像版本页面查看推送的版本。

#### 步骤三:部署镜像

1. 创建Go应用。

在**应用部署配置**页签,**技术栈语**言需选择**其它语言,应用部署方式**需选择镜像。具体操作,请参 见<mark>在SAE控制台使用镜像部署多语言应用</mark>。

⑦ 说明 除创建新应用外,您还可以修改现有应用的配置镜像。具体操作,请参见升级应用。

2. 为Go应用绑定公网SLB。

选择HTTP协议,并将HTTP端口设置为80,容器端口设置为8080。具体操作,请参见为应用绑定 SLB。

绑定完成后,您可以在应用访问设置区域查看公网访问地址。

## 3. 验证结果。

按照*http://slbip:port/*的格式,在浏览器地址栏输入步骤2设置的公网地址并回车。结果如下:



# 更多信息

如果您不想每次都通过手动方式构建、推送镜像,并部署应用,您还可以对接开源的Jenkins或者阿里云云 效。具体信息,请参见以下文档:

- 部署Golang应用至SAE
- 使用Jenkins创建多语言应用的持续集成

# 4.监控

# 4.1. VPC网络下的SAE应用如何接入阿里云 Prometheus监控

本文介绍如何将VPC网络下的SAE应用接入阿里云Prometheus监控。

步骤一:创建SAE应用并获取VPC、交换机等信息

在SAE控制台中创建应用并暴露Metrics(可以为其他地址)服务。具体操作,请参见在SAE控制台使用WAR包部 署Java Web应用。

Serverless应用引擎SAE	Serverless应用引擎SAE / 应用列表						
概览页	应用列表						
应用列表	全部   默认(cn-hangzhou)   apaas   apaas-	全部   別以(n-hargshou)   apaas-     apaas-     ama-     prom-					
命名空间	(注应用 应用名称 > 请输入应用名	和	<b>Q</b> 标班				\$ C
配置管理 ン	应用名称	标签	命名空间	弹性策略启用状态	当前实例数 11/目标实例数 11	应用描述	操作
权限管理 ン	ARMS		cn-hanç	未设置	0/1	cheche	管理 复别
联系人管理	prom-		cn-hang:	未设置	0/1		管理 复制
	<b>0</b>	ø	cn-hangzhou:	未设置	0/1		管理 复制
	xingji-se	ø	cn-han	未设置	0/1	xingji-serverless-hz1	管理 复制
	xingji-se	ø	cn-han	未设置	0/1		管理 复制
	yougi-tes	٠	cn-hang	未设置	2/2		管理 复制
	<□ 批量停止应用 批量启动应用 批量设置	162 (82	\$中0个应用) <b>②</b>			毎页显示: 10 ~ 共6条	< 上一页 1 下一页 >

在SAE控制台的应用列表页面已获取应用的命名空间、VPC、交换机、安全组等信息。

短期列表 / 基本信息							
← 应用列表	(youqi-t )						
基本信息	基本信息		部署应用	回退历史版本 手动扩缩 自动扩缩 停止应用 更多 >>			
変更记录 应用事件	● 1.但用在VPC内部以及平衡100公司所,加減業要可用公司消费投入下還用指約12回公司,常常以約4公司施業量值多利 2.如果你的应用需要可以用の容数要求,清多考:如何改置PDS会合参考						
日志管理 🖌	基本位息 实例部署组织						
基础监控 应用监控 ~	◇ 计量数据 ♂ 了解计量方式						
微服务治理 🗸	本月CPU资源使用量 💿	本月Memory资源使用量 🛛	实时CPU资源使用量 ♀	实时Memory资源使用量 💿			
限流降振(仅支持Java) ン	1,014 <sub>核·min</sub>	2,028 GiB-min	1 核·min	2 GiB-min			
> 通知告誓 ~	▽ 応用信息						
	- mmmw 反用者 youghtes 命名空雨 on-ha						
	应用id: 40b2017e-		VPC: acs-upo				
	vSwitch: acs-vawitch (可用p数为239) \$vSwitch即著		安全编: edas.serverless.dono /vyzav 切换安全组				
	实例现格: 0.5Core, 1GiB, 系统盘磁盘空间20GiB 变更现格		运行实则数: 当前2个实例/目标2个实例				
	应用创建时间: 2022年3月29日 15:50:32		镶像地址: registry-vpc.cn-hang n-ecs.nginx-exporter				
	原用接筆: environment daily, owner:shangaban						

# 步骤二:安装Prometheus For VPC

- 1. 登录Prometheus控制台。
- 2. 在Prometheus监控页面的顶部菜单栏,选择地域,然后单击新建prometheus实例。
- 3. 在新建prometheus实例页面,单击Prometheus实例 for VPC区域。 在接入 ECS集群(VPC)面板显示当前地域下的所有VPC列表。
- 4. 单击步骤一中获取的目标VPC右侧操作列的安装。
- 5. 在**安装Promet heus应用**对话框中,输入VPC名称,选择交换机和安全组,然后单击确定。

安装Prometheus应用				
vpc-bp12j				
cmonitor-or				
请选择	$\sim$			
请选择	$\sim$			
确定取	消			
	theus应用 vpc-bp12j cmonitor-or 请选择 请选择			

# ? 说明

- VPC:为步骤一中获取的VPC。
- 交换机和安全组: 建议您选择与步骤一中获取的交换机和安全组保持一致。
- 。 安全组规则: 您还需要添加网络规则, 确保SAE应用和Prometheus监控之间的网络通畅。

### 安装成功后,对应VPC右侧**状态**列显示安装成功。

接入 ECS集群(VPC)		华东1 (杭州)	$\sim$				
ARMS使用Prometheus监控监测ECS集群(VPC)的数据。							
VPC	集群名称	交换机	安全组	状态	操作		
vpc-bp1				安装成 功	卸载		
vpc-bp1				未安装	安装		
vpc-bp1				未安装	安装		
vpc-bp*				安装成 功	卸载		
vpc-bp1r			$\{y_{1},y_{2},\cdots,y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y_{n},y$	卸载成 功	安装		

# 步骤三: 配置SAE服务发现

- 1. 在**Prometheus监控**页面的顶部菜单栏,选择地域,然后单击步骤二中安装的Prometheus实例 for VPC操作列的设置。
- 2. 在**设置**页面单击**编辑Promet heus.yaml**,在弹出的**编辑Promet heus.yaml**对话框中输入如下代码 段,创建SAE服务发现。

global:

```
scrape interval: 30s
 scrape timeout: 30s
 evaluation interval: 30s
scrape configs:
## job名称 可以自定义修改
- job name: sae-test
 honor timestamps: true
 scrape interval: 30s
 scrape timeout: 10s
 metrics_path: /metrics
 scheme: http
 ## SAE专用服务发现类型
 aliyun sae sd configs:
     ## 请将端口号调整为您的应用对应的端口号
   - port: 9913
     ## 请根据你的实际情况填写用户信息
     user id: ********
     access_key: ********
     access_key_secret: *******
     refresh interval: 30s
     ## 请根据步骤1中应用的信息填写
     region id: cn-hangzhou
     vpc_id: vpc-bp156863cbb2z17bhyzav
     namespace id: ["cn-hangzhou"]
     app name: ["test1","test2"]
     ##
     tag_filters:
       - key: environment
        values: ["daily", "publish"]
       - key: owner
         values: ["zhangshan"]
 relabel configs:
   - source_labels: [__meta_sae_private_ip]
     regex: (.*)
     target label: address
     ##将端口号调整为您的应用对应的端口号
     replacement: $1:9913
   - source_labels: [__meta_sae_namespace_id]
     regex: (.*)
     target label: namespace
     replacement: $1
   - source_labels: [__meta_sae_app_name]
     regex: (.*)
     target label: app name
     replacement: $1
   - source_labels: [__meta_sae_group_name]
     regex: (.*)
     target label: group name
     replacement: $1
   - source_labels: [__meta_sae_instance_id]
     regex: (.*)
     target label: instance id
     replacement: $1
   - source labels: [ meta sae status]
```

```
regex: (.*)
target_label: instance_status
replacement: $1
## 用户在应用上自定义的标签(格式:__meta_sae_tag_[tagName])
- source_labels: [__meta_sae_tag_owner]
regex: (.*)
target_label: owner
replacement: $1
```

# 步骤四:查看监控数据

- 1. 在左侧导航栏单击大盘列表,然后在大盘列表页面单击创建大盘,进入Grafana页面。
- 在左侧导航栏单击Explore,然后在Explore页面选择VPC对应的数据源,然后输入Metrics查询命令, 例如,up{job="sae-test"}。

```
⑦ 说明 其中job名称为步骤三中创建的。
```



检查数据采集是否符合预期,若未成功采集到数据,请您检查安全组网络规则和Promet heus.yaml配置。

# 5.存储

# 5.1. 如何避免多进程/多客户端并发写同一 日志文件可能出现的异常?

## 问题现象

文件存储 NAS 为多客户端提供了统一名字空间的文件共享读写能力,但在多进程/多客户端并发写同一个文件的场景中(典型的例如并发写同一个日志文件),各进程分别维护了独立的文件描述符及写入位置等上下文信息,而 NFS 协议本身并没有提供Atomic Append语义的支持,因此可能会出现写覆盖、交叉、串行等异常现象,推荐的解决方案有两种:

- (推荐)不同进程/客户端写入同一文件系统的不同文件中,后续分析处理时再进行归并,这个方案能够 很好地解决并发写入导致的问题,同时无需使用文件锁,不会对性能造成影响。
- 对于并发追加写同一个文件(如日志)的场景,可以使用文件锁 + seek 机制来保证写入的原子性和一致 性。但是文件锁 + seek 是一个比较耗时的操作,可能会对性能产生显著的影响。下面将对这种方式进行 一个简单的介绍,以供参考。

# flock + seek 使用方法

由于 NFS 协议本身没有提供对 Atomic Append 语义的支持,因此当并发写入同一文件末尾(如日志)时, 很可能会出现相互覆盖的情况。在 Linux 中,通过使用 flock + seek 的方式,可以在 NFS 文件系统上做到模 拟 Atomic Append,对并发追加写入同一文件提供保护和支持。

使用方式如下:

- 调用 fd = open(filename, O\_WRONLY | O\_APPEND | O\_DIRECT) 以追加写的方式打开文件,并且指定 O\_DIRECT(直写,不通过 Page Cache),获得文件描述符 fd。
- 2. 调用 flock(fd, LOCK\_EX|LOCK\_NB) 尝试获取文件锁,如果获取失败(如锁已被占用)则会返回错误,此 时可以继续重试或进行错误处理。
- 3. 文件锁获取成功后,调用 lseek(fd, 0, SEEK\_END)将 fd 当前的写入偏移定位到文件末尾。
- 4. 执行正常的 write 操作,此时写入位置应该是文件的末尾,并且由于有文件锁的保护,不会出现并发写入相互覆盖的问题。
- 5. 写操作执行完成后,调用 flock(fd, LOCK\_UN) 释放文件锁。

下面是一个简单的 C 语言示例程序, 仅供参考:

```
#define _GNU_SOURCE
#include<stdlib.h>
#include<stdio.h>
#include<fcntl.h>
#include<string.h>
#include<string.h>
#include<sys/file.h>
#include<time.h>
const char *OUTPUT_FILE = "/mnt/blog";
int WRITE_COUNT = 50000;
int do_lock(int fd)
{
    int ret = -1;
    while (1)
```

```
{
       ret = flock(fd, LOCK_EX | LOCK_NB);
      if (ret == 0)
       {
           break;
       }
      usleep((rand() % 10) * 1000);
   }
   return ret;
}
int do_unlock(int fd)
{
  return flock(fd, LOCK_UN);
}
int main()
{
       int fd = open(OUTPUT FILE, O WRONLY | O APPEND | O DIRECT);
       if (fd < 0)
        {
              printf("Error Open\n");
               exit(-1);
        }
       for (int i = 0; i < WRITE COUNT; ++i)</pre>
        {
               char *buf = "one line\n";
               /* Lock file */
               int ret = do_lock(fd);
               if (ret != 0)
               {
                     printf("Lock Error\n");
                      exit(-1);
               }
               /* Seek to the end */
               ret = lseek(fd, 0, SEEK_END);
               if (ret < 0)
                {
                      printf("Seek Error\n");
                      exit(-1);
               }
               /* Write to file */
               int n = write(fd, buf, strlen(buf));
               if (n <= 0)
                {
                      printf("Write Error\n");
                      exit(-1);
               }
               /* Unlock file */
               ret = do_unlock(fd);
               if (ret != 0)
                {
                      printf("UnLock Error\n");
                       exit(-1);
                }
```
return 0;

}

### 更详细的 flock() 使用方式可以参考以下链

接: http://www.hackinglinuxexposed.com/articles/20030616.html。

⑦ 说明 在NAS文件系统上使用 flock()需要您的Linux内核版本在2.6.12及以上,如果您的内核版本较低,请使用 fcntl()调用。

# 5.2. SAE日志导入Kafka最佳实践

在Serverless应用引擎SAE(Serverless App Engine)产生的日志导入到消息队列Kaf ka版之后,您可以将 Kaf ka中的日志统一写入阿里云Elast icsearch,便于后续的统一查看和处理。本文介绍通过SAE日志导入到 Kaf ka,再从Kaf ka将数据写入Elast icsearch的最佳实践。

### 前提条件

- 购买Kafka实例,并创建Topic和Group供SAE应用使用。具体信息,请参见创建Kafka资源。
- 创建或部署SAE应用时,开通日志收集到Kafka日志服务功能。具体信息,请参见设置日志收集至Kafka。
- 确认是否成功采集消息到Kafka。具体信息,请参见查询消息。
- 创建阿里云Elasticsearch实例。具体信息,请参见创建阿里云Elasticsearch实例。
- 创建阿里云Logstash实例。具体信息,请参见创建阿里云Logstash实例。

### 步骤一: 创建Group

本步骤介绍如何创建阿里云Elasticsearch所属的Group。

- 1. 登录消息队列Kafka版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在**实例列表**页面,单击目标实例名称。
- 4. 在左侧导航栏,单击Group管理。
- 5. 在Group 管理页面, 单击创建 Group。
- 6. 在创建 Group面板的Group ID文本框输入Group的名称,在描述文本框简要描述Group,并给Group添加标签,单击确定。
  创建完成后,在Group 管理页面的列表中显示已创建的Group。

创建元成后,在Group 管理贝面的列表中显示已创建的Grou

### 步骤二: 创建索引

本步骤介绍如何通过阿里云Elasticsearch创建索引,接收消息队列Kafka版的数据。

- 1. 登录阿里云Elasticsearch控制台。
- 2. 在左侧导航栏,单击Elasticsearch实例。
- 3. 进入目标实例。
  - i. 在顶部菜单栏处,选择资源组和地域。
  - ii. 在左侧导航栏,单击Elasticsearch实例,然后在Elasticsearch实例中单击目标实例ID。
- 4. 在左侧导航栏,选择配置与管理 > 可视化控制。
- 5. 在Kibana区域中,单击私网入口或公网入口。

- 私网入口:开启Kibana私网访问后(默认未开启),才会显示。具体操作,请参见配置Kibana公网 或私网访问白名单。
- 公网入口:开启Kibana公网访问后(默认开启),才会显示。具体操作,请参见配置Kibana公网或 私网访问白名单。

↓ 注意 首次从公网入口进入Kibana控制台且公网访问配置未修改时,系统会提示您修改配置。单击修改配置,进入Kibana配置页面修改Kibana公网访问白名单,在白名单中加入您客户端的IP地址,具体操作请参见配置Kibana公网或私网访问白名单。修改后,再次单击公网入口,即可进入Kibana控制台。

获取客户端的IP地址:如果您的客户端处在家庭网络或公司局域网中,您需要将局域网的公网出口IP地址添加到白名单中,而非客户端机器的内网机制。建议您通过淘宝IP地址库查看您当前使用的公网IP。您也可以将白名单配置为0.0.0.0/0,允许所有IPv4地址访问Kibana。此配置会导致Kibana完全暴露在公网中,增加安全风险,配置前请确认您是否可以接受这个风险。

6. 在Kibana登录页面,输入用户名和密码,单击Log in。

参数说明如下。

- ・ 用户名:默认为elastic。您也可以创建自定义用户,具体操作请参见通过Elasticsearch X-Pack角色管
   理实现用户权限管控。
- 密码:对应用户的密码。elastic用户的密码在创建实例时设定,如果忘记可重置。重置密码的注意事项和操作步骤,请参见重置实例访问密码。
- 7. 在Kibana控制台的左侧导航栏,选择Management > Dev Tools。
- 8. 执行以下命令, 创建索引。

```
PUT /elastic_test
{}
```

### 步骤三: 创建管道

本步骤介绍如何通过阿里云Logstash创建管道。成功部署管道后,数据将持续地从消息队列Kafka版导入到 阿里云Elasticsearch。

- 1. 登录阿里云Elasticsearch控制台。
- 2. 进入目标实例。
  - i. 在顶部菜单栏处,选择地域。
  - ii. 在左侧导航栏,单击Logstash实例,然后在Logstash实例中单击目标实例ID。
- 3. 在左侧导航栏,单击管道管理。
- 4. 在管道列表页面,单击创建管道。
- 5. 在创建管道任务配置向导页面, 配置相关参数。
  - i. 在Config配置页签, 配置相关参数并单击下一步。

本文的参数配置以Kaf ka数据导入Elast icsearch为例。参数的更多信息,请参见通过配置文件管理 管道。

- 管道ID: 自定义。
- Config配置:配置示例如下。

```
input {
   kafka {
   bootstrap servers => ["alikafka-pre-cn-zv*******-1-vpc.alikafka.aliyuncs.c
om:9092,alikafka-pre-cn-zv*******-2-vpc.alikafka.aliyuncs.com:9092,alikafka-pr
e-cn-zv*******-3-vpc.alikafka.aliyuncs.com:9092"]
   group_id => "elastic_group"
   topics => ["elastic test"]
   codec => json
   consumer_threads => 12
   decorate_events => true
   }
}
output {
   elasticsearch {
   hosts => ["http://es-cn-o40xxxxxxxxxm.elasticsearch.aliyuncs.com:9200"]
   index => "elastic_test"
   password => "XXX"
   user => "elastic"
   }
}
```

### 参数说明如下。

参数	说明	示例值
input		
bootstrap_servers	Kafka的VPC环境接入点。	alikaf ka-pre-cn- zv*******-1- vpc.alikaf ka.aliyuncs. com: 9092, alikaf ka- pre-cn-zv********- 2- vpc.alikaf ka.aliyuncs. com: 9092, alikaf ka- pre-cn-zv*******- 3- vpc.alikaf ka.aliyuncs. com: 9092
group_id	Group的名称。	elastic_group
topics	Topic的名称。	elastic_test
codec	解码的类型。 ⑦ 说明 建议设置为JSON格式,与SAE日 志导入到Kafka的数据格式保持一致。	json

参数	说明	示例值
	消费线程数。	12
consumer_threads	⑦ 说明 建议与Topic的分区数保持一 致。	
decorate_events	是否包含消息元数据。默认值为 false 。	true
output		
hosts	Elasticsearch的访问地址。您可在Elasticsearch实 例的基本信息页面获取。	http://es-cn- o40xxxxxxxxxxwm. elasticsearch.aliyunc s.com:9200
index	索引的名称。	elastic_test
password	访问Elasticsearch的密码。您在创建Elasticsearch 实例时设置的密码。	XXX
user	访问Elasticsearch的用户名。您在创建 Elasticsearch实例时设置的用户名。	elastic

ii. 在管道参数配置页签, 配置相关参数, 单击保存并部署。

本示例以默认配置为准,您可以按需修改。

6. 在提示对话框,单击确认。

### 步骤四: 搜索数据

您可以在Kibana控制台搜索通过管道导入阿里云Elasticsearch的数据,确认数据是否导入成功。

- 1. 登录阿里云Elasticsearch控制台。
- 2. 进入目标实例。
  - i. 在顶部菜单栏处,选择资源组和地域。

ii. 在左侧导航栏,单击Elasticsearch实例,然后在Elasticsearch实例中单击目标实例ID。

- 3. 在左侧导航栏,选择配置与管理 > 可视化控制。
- 4. 在Kibana控制台的左侧导航栏,选择Management > Dev Tools。

登录Kibana控制台的具体步骤,请参见创建索引步骤5~7。

5. 执行以下命令,搜索数据。

```
GET /elastic_test/_search
{}
```

返回示例如下。

	1-	[
	2	"took" : 7,
	3	"timed_out" : false,
	4 -	"_shards" : {
	5	"total" : 1,
	6	"successful" : 1,
	7	"skipped" : 0,
	8	"failed" : 0
	9 -	},
1	.0 -	"hits" : {
1	.1 +	"total" : {
1	.2	"value" : 63,
1	.3	"relation" : "eq"
1	.4 -	},
1	.5	"max_score" : 1.0,
1	.6 -	"hits" : [
1	.7 -	
1	.8	"_index" : "elastic_test",
1	.9	"_type" : "_doc",
2	0	,
2	1	
2	2 -	source": {
2	3	"message" : """14-Mar-2022 20:17:13.962 INF0 [Thread-23] org.apache.coyote
		.AbstractProtocol.pause Pausing ProtocolHandler ["http-nio-8080"]""",
2	.4	"file" : "/home/admin/apache-tomcat-8.5.42/logs/catalina.2022-03-14.log",
2	25	"host" : "test-kafka-7d1d9a88-2a5c-4838-
2	26	"topic" : "test",
2	.7	"@version" : "1",
2	.8	"@timestamp" : "2022-03-14T12:17:19.673Z"
2	.9 -	
3	60 -	] },
3	1 -	-{
3	2	
3	3	"_type" : "_doc",
3	4	,,
3	5	"scope" · 1 0

# 6.微服务

# 6.1. 开发场景:通过端云联调实现开发和测 试环境隔离

本文通过模拟真实的调用链路,为您演示如何自动实现Serverless应用引擎SAE(Serverless App Engine)本 地开发环境与测试环境的流量隔离,方便您快速开发联调,提高开发效率。

### 前提条件

- 在Intellij IDEA中安装和配置Cloud Toolkit
- 在应用所在VPC内创建一台可使用SSH登录的ECS,用于建立端云互联通道。更多信息,请参见通过控制台 使用ECS实例(快捷版)云服务器ECS快速入门。

### ↓ 注意

- 。 ECS与应用必须处于同一VPC内。
- SSH通道需要使用密码方式登录,暂不支持使用密钥对登录。

### 背景信息

以下图中Spring Cloud框架的微服务应用为例,您无需修改任何业务代码,只需设置流量标签(Tag),通 过端云联调,将流量通过不同的路径分配到SAE测试环境或本地开发环境,自动实现流量隔离。



### 步骤一: 将应用部署至SAE测试环境

部署Demo应用,分别命名为spring-cloud-zuul、spring-cloud-a、spring-cloud-b、spring-cloud-c。具体操作,请参见将Spring Cloud应用托管到SAE。

⑦ 说明 建议使用镜像方式或JAR包方式部署应用,并配置启动参数 Dnacos.use.endpoint.parsing.rule=false 和 -Dnacos.use.cloud.namespace.parsing=false 才能
 使用已购买的注册中心。

### 步骤二:配置端云互联

更多信息,请参见使用Cloud Toolkit实现端云互联(Intellij IDEA)。

- 1. 启动Intellij IDEA。
- 2. 在顶部菜单栏中选择Tools > Alibaba Cloud > Preferences...。
- 3. 在Settings对话框中选择Alibaba Cloud Toolkit > Microservice。
- 4. 在Microservice页面下配置端云互联的相关参数。
  - i. Proxy:购买一台跟SAE应用在同一VPC下的ECS(带公网IP)作为代理机器,添加至Proxy页面。

		Prefere	nces		
Q.	Alibaba Cloud To	olkit > Microservice > Pr	оху		Reset
Appearance & Behavior	您可在此页面配置·	一级或多级代理,并在端云互联	或远程debug中使用;		
▼ Alibaba Cloud Toolkit	Profile SI	nenZhen-pre		AddProfile	Remove Profile
Appearance & Behavior	Profile Detail				
Accounts	Profile Name	ShenZhen-pre			
SSH Profile	代理米刑	土朷端口	/环墙	影中	Add
Host Tag	SSH	1/0%m (	/×1.3%	root	Aud
► Docker					Edit
▼ Microservice					Remove
Microservice					
Proxy					
Apsara Stack					
MSE					
Import & Export Settings					
► Code					
Кеутар					
► Editor					
Plugins 🕜					
► Version Control					
Build, Execution, Deployment					
Languages & Frameworks					
▶ Tools					
Other Settings					

ii. Microservice:在Microservice页面配置代理和已购买的MSE注册中心。



### iii. MSE:在MSE页面配置相关参数。

		Preferences					
	Alibaba Cloud Toolkit 🔸 N	licroservice > MSE					Reset
► Appearance & Behavior ▼ Alibaba Cloud Toolkit	MSE(微服务引擎)产品配置 ☑ 开启微服务治理						
Appearance & Behavior	License Key:	APON DEALORA		8-68			
Accounts SSH Profile	App Name:	cn-beijing-nacos-sprir	ng-p	provider-			
Host Tag	Tag:	local					
► Docker ▼ Microservice	Agent地址: —	深圳 🔻		:n-shenzhen.aliyuncs.com/ms/	ArmsAgent.zip	清除本地缓行	
	☑ 开启RPC灰度			☑ 开启消息灰度     ☑ 开启	标签染色		
Apsara Stack							
MSE							
Import & Export Settings							
▶ Code							
► Editor							
Plugins 🕖							
► Version Control							
Build, Execution, Deployment							
Languages & Frameworks							
► Tools							
<ul> <li>Other Settings</li> </ul>							
					Cancel	Apply	ок

### a. 输入License Key和App Name。

您可以登录SAE的Websell,执行以下命令获取License Key和App Name。登录Webshell的 具体操作,请参见查看Webshell。

env | grep java

### 输出示例如下:

JAVA\_TOOL\_OPTIONS=-javaagent:/home/admin/.opt/ArmsAgent/arms-bootstrap-1.7.0-SN APSHOT.jar -Darms.licenseKey=iioe7jcnuk@a0bcdaec24f\*\*\*\* -Darms.appId=8641dbc9-8 663-4c47-95f7-683fa6dd\*\*\*\* -Darms.agent.env=Serverless -Darms.agent.args= -Dpr ofiler.micro.service.canary.enable=true -Dprofiler.micro.service.metadata.repor t.enable=true -Dprofiler.micro.service.http.server.enable=true -Dmsc.appName=cn -shenzhen-test-sc-gateway-\*\*\*\*\* -Dmsc.licenseKey=iioe7jcnuk@460d62050fa\*\*\*\*

- b. 设置标签。在Tag栏输入local。
- c. 下载Agent到本地。
- 5. 单击Apply后再单击OK。
- 6. 启动本地应用,验证联调。
   在Info对话框如果看到端云互联成功启用的提示,证明端云互联功能已生效。

•	<b>lnfo</b> 模块 A:端云互联已启用 * 此提示可通过高级配置自动关闭;	
		ОК

### 步骤三:发起流量调用进行测试

如果您发往云原生网关的Request是HTTP请求,您需要在请求的Header中添加\_\_\_x-mse-tag=[local]\_,配置完成后流量会自动在指定的环境内完成闭环。

⑦ 说明 此处的Key为 x-mse-tag 是固定值,其参数值需要和环境标签保持一致。

环境标签配置完成后,使用curl命令发起流量调用请求,返回结果如下:

• 主链路:

curl http://47.106.XX.XX:8080/A/a

• 调用结果:

A[10.0.1.160] -> B[10.0.1.161] -> C[10.0.1.162]

● 分支链路:

curl http://47.106.XX.XX:8080/A/a \--header 'x-mse-tag: local'

● 调用结果:

A[127.0.0.1] -> B[127.0.0.1] -> C[10.0.1.162]

# 6.2. 开发场景:基于消息队列RocketMQ版 实现环境隔离

本文介绍如何在Serverless应用引擎SAE (Serverless App Engine) 上基于阿里云

消息队列Rocket MQ版

实现环境隔离,您可以在不修改任何业务代码的情况下,实现异步场景下的流量控制。

### 背景信息

Producer		MQ S	Server		Cor	nsumer
消息发送方	TopicA ————————————————————————————————————	Message		gray消费组	Message	▶ 消息消费者
	(env=gray)		SQL 92 filter			
消息发送方	 用户属性 (env=base)	Message		base消费组	Message	▶ 消息消费者

- 支持Rocket MQ版本: 4.2.0及以上版本。
- 支持Pull和Push两种模式。
- 需要在服务端配置 enablePropertyFilter=true 并重启服务端。

### 准备工作

### 部署SAE Demo应用

- 1. 下载Demo。
- 2. 部署主干链路应用。

部署主干链路应用(A、B、C)。具体操作,请参见将Spring Cloud应用托管到SAE。

3. 部署灰度链路应用(A-gray、B-gray、C-gray), 在启动命令中添加启动参数 -Dalicloud.service.t ag=gray 用于区别主干链路应用。

⑦ 说明 部署时,如果需要使用独立的注册中心,需要添加启动参数 -

Dnacos.use.endpoint.parsing.rule=false -Dnacos.use.cloud.namespace.parsing=false 。

### 部署RocketMQ

• 消息的生产者和消息的消费者,需要同时开启Rocket MQ消息灰度,消息的灰度功能才能生效。消息类型 目前只支持Rocket MQ,包含开源版本(Apache Rocket MQ)和阿里云商业版(

### 消息队列Rocket MQ版

)。

- 如果您使用开源Rocket MQ,则Rocket MQ Server和Rocket MQ Client都需要使用4.5.0及以上版本。更多 信息,请参见Apache Rocket MQ。
- 如果您使用阿里云

消息队列RocketMQ版

- ,需要使用铂金版,且Ons Client使用1.8.0.Final及以上版本。更多信息,请参见快速入门概述。
- 开启Rocket MQ消息灰度后,消息的Consumer Group会被修改。例如原来的Consumer Group 为 group1 ,环境标签为 gray ,开启Rocket MQ消息灰度后,则group会被修改成 group1\_gray , 如果您使用的是阿里云

消息队列Rocket MQ版

- , 请提前创建好Group。
- 默认使用SQL92的过滤方式,如果您使用开源RocketMQ,需要在服务端开启SQL92过滤功能(即在 brok er.conf 中配置 enablePropertyFilter=true )。

### 步骤一:为应用开启RocketMQ消息灰度

Demo中的spring-cloud-c、spring-cloud-a应用分别为消息的生产者和消费者,通过添加启动参数 - Dprofiler.micro.service.mq.gray.enable=true ,在SAE上为其开启Rocket MQ消息灰度。

⑦ 说明 Rocket MQ消息灰度功能的开启和关闭,需要在SAE控制台重新部署应用后才能生效。

### 步骤二:引入流量并进行验证

Demo应用的结构图如下,应用之间的调用,既包含了Spring Cloud的调用,也包含了Dubbo的调用,覆盖 了当前市面上最常用的两种微服务框架。其中C应用会生产出Rocket MQ消息,由A应用进行消费,A应用在 消费消息时,也会发起新的调用。这些应用都是基础的Spring Cloud、Dubbo和Rocket MQ的标准用法。



Demo的调用链路为: spring-cloud-zuul应用在收到/A/dubbo的请求时,会把请求转发给spring-cloud-a, 然后spring-cloud-a通过Dubbo协议去访问spring-cloud-b, spring-cloud-b也通过Dubbo协议去访问spring-cloud-c, spring-cloud-c在收到请求后,会生产一个消息,并返回自己的环境标签和IP地址。这些生产出来的消息会由spring-cloud-a应用消费, spring-cloud-a应用在消费消息的时候,会通过spring cloud去调用spring-cloud-c,并且将结果输出到自己的日志中。

```
#访问/A/dubbo的时候
#返回值为A[10.25.xx.xx] -> B[10.25.xx.xx] -> C[10.25.xx.xx]
#同时,A应用在接收到消息之后,输出的日志如下:
c.a.mse.demo.service.MqConsumer: topic:TEST_MQ,producer:C[10.25.xx.xx],invoke result:A[10.2
5.xx.xx] -> B[10.25.xx.xx] -> C[10.25.xx.xx]
```

您可以登录SAE控制台,查看spring-cloud-a应用的日志并验证配置。基线环境可以同时消费灰度(gray)和基线(base)环境生产出来的消息,而且在消费对应环境消息时产生的Spring Cloud调用分别路由到灰度 (gray)和基线(base)环境中。具体操作,请参见查看实时日志。

# 6.3. 线上场景:基于自建Spring Cloud Gateway或Zuul网关实现全链路灰度

本文通过示例演示如何在SAE上实现全链路灰度发布,可以在不需要修改您的任何业务代码的情况下,实现全链路流量控制。

### 准备工作

本文通过示例为您演示ALB网关全链路灰度功能。假设应用的架构由ALB网关以及后端的微服务架构(Spring Cloud)组成,后端调用链路有3个微服务:购物车(A)、交易中心(B)、库存中心(C),可以通过客户端或HTML来访问后端服务,这些服务之间通过Nacos注册中心实现服务发现。

### 部署SAE Demo应用

- 1. 下载Demo。
- 2. 部署主干链路应用。

部署主干链路应用(A、B、C)。具体操作,请参见将Spring Cloud应用托管到SAE。

3. 部署灰度链路应用(A-gray、B-gray、C-gray), 在启动命令中添加启动参数 -Dalicloud.service.t ag=gray 用于区别主干链路应用。

⑦ 说明 部署时,如果需要使用独立的注册中心,需要添加启动参数 Dnacos.use.endpoint.parsing.rule=false -Dnacos.use.cloud.namespace.parsing=false 。

### 部署Spring Cloud Gateway或Zuul网关

应用可以使用相同域名,通过不同路径流量转发实现请求路由分发。具体操作,请参见为应用配置网关路由 (CLB)。

### 为流量配置特定Header实现全链路灰度

有些客户端无法改写域名,希望能在访问 www.base.com 时,通过传入不同的Header来路由到灰度环境。 例如下图中,通过添加 x-mse-tag:gray 这个Header,来访问灰度(gray)环境。



### 结果验证。

- 访问 www.base.com 路由到基线 (base) 环境。
  - 执行以下curl命令:

curl -H"Host:www.base.com" http://106.14.XX.XX/a

○ 执行结果:

A[172.18.XX.XX] -> B[172.18.XX.XX] -> C[172.18.XX.XX]

- 如果想访问灰度环境,只需要在请求中增加一个Header: x-mse-tag:gray 。
  - 执行以下curl命令:

curl -H"Host:www.base.com" -H"x-mse-tag:gray" http://106.14.XX.XX/a

○ 执行结果:

Agray[172.18.XX.XX] -> Bgray[172.18.XX.XX] -> Cgray[172.18.XX.XX]%

可以看到CLB Ingress根据这个Header直接路由到了A的灰度(gray)环境中。

# 6.4. 线上场景:基于Ingress网关实现全链 路灰度

本文介绍如何通过ALB Ingress网关,在不需要修改您的任何业务代码的情况下,为您的SAE应用实现全链路 流量控制。

### 背景信息

微服务架构下,有些开发需求会导致微服务调用链路上的多个微服务同时发生改动,通常每个微服务都会有 灰度环境或分组来接收灰度流量。此时希望进入上游灰度环境的流量,也能进入下游灰度的环境中,确保1 个请求始终在灰度环境中传递,即使这个调用链路上有一些微服务没有灰度环境。通过SAE提供的全链路灰 度能力,能够在不需要修改任何您的业务代码的情况下,轻松实现上述能力。

### 准备工作

本文通过示例为您演示ALB网关全链路灰度功能。假设应用的架构由ALB网关以及后端的微服务架构(Spring Cloud)组成,后端调用链路有3个微服务:购物车(A)、交易中心(B)、库存中心(C),可以通过客户端或HTML来访问后端服务,这些服务之间通过Nacos注册中心实现服务发现。

### 部署SAE Demo应用

- 1. 下载Demo。
- 2. 部署主干链路应用。

部署主干链路应用(A、B、C)。具体操作,请参见将Spring Cloud应用托管到SAE。

3. 部署灰度链路应用(A-gray、B-gray、C-gray), 在启动命令中添加启动参数 -Dalicloud.service.t ag=gray 用于区别主干链路应用。

⑦ 说明 部署时,如果需要使用独立的注册中心,需要添加启动参数 Dnacos.use.endpoint.parsing.rule=false -Dnacos.use.cloud.namespace.parsing=false 。

### 部署Ingress网关路由

应用可以使用相同域名,通过不同路径流量转发实现请求路由分发。具体操作,请参见为应用配置网关路由 (ALB)和为应用配置网关路由(CLB)。

### 为流量配置特定Header实现全链路灰度

有些客户端无法改写域名,希望能在访问 www.base.com 时,通过传入不同的Header来路由到灰度环境。 例如下图中,通过添加 x-mse-tag:gray 这个Header,来访问灰度(gray)环境。



### 结果验证。

- 访问 www.base.com 路由到基线 (base) 环境。
  - 执行以下curl命令:

curl -H"Host:www.base.com" http://106.14.XX.XX/a

○ 执行结果:

A[172.18.XX.XX] -> B[172.18.XX.XX] -> C[172.18.XX.XX] %

- 如果入口应用A没有灰度(gray)环境,访问到A的基线(base)环境,又需要在A→B的时候进入灰度环境,则可以通过增加一个特殊的Header: x-mse-tag
   網如 gray 。
  - 执行以下curl命令:

curl -H"Host:www.base.com" -H"x-mse-tag:gray" http://106.14.XX.XX/a

○ 执行结果:

A[172.18.XX.XX] -> Bgray[172.18.XX.XX] -> Cgray[172.18.XX.XX]%

可以看到,首先进入了A的基线(base)环境,但是A→B的时候又重新回到了灰度(gray)环境。

⑦ 说明 您在ALB网关中配置好规则,当某个应用需要灰度发布时,只需要在灰度环境中部署好应用,灰度流量会进入到灰度节点中。如果验证无误,则将灰度的镜像发布到基线环境中;如果一次变更中有多个应用需要灰度发布,则把他们都加入到灰度环境中即可。

# 7.Java 7.1. JVM参数配置说明

本文列出了JVM启动参数相关信息。

### 调优栈内存

堆大小典型配置参数

配置参数	说明	示例
-Xmx	设置最大堆大小。	Xmx3550m / 设置JVM最大可用 内存为3550 MB。
-Xms	设置JVM初始内存。	-Xms3550m ,设置JVM初始内存 为3550 MB。此值建议与 -Xmx 相 同,避免每次垃圾回收完成后JVM重 新分配内存。
-Xmn2g	设置年轻代大小。	-Xmn2g ,设置年轻代大小为2 GB。整个JVM内存大小=年轻代大小 +年老代大小+持久代大小。持久代 一般固定大小为64 MB,所以增大年 轻代后,将会减小年老代大小。此值 对系统性能影响较大,Sun官方推荐 配置为整个堆的3/8。
-Xss	设置线程的栈大小。	-Xss128k , 设置每个线程的栈 大小为128 KB。 ⑦ 说明 JDK 5.0版本以后每 个线程栈大小为1 MB, JDK 5.0 以前版本每个线程栈大小为256 KB。请依据应用的线程所需内 存大小进行调整。在相同物理 内存下,减小该值可以生成更 多的线程。但是操作系统对一 个进程内的线程个数有一定的 限制,无法无限生成,一般在 3000个~5000个。
-XX:NewRatio=n	设置年轻代和年老代的比值。	-XX:NewRatio=4,设置年轻代 (包括Eden和两个Survivor区)与年 老代的比值(除去持久代)。如果设 置为4,那么年轻代与年老代所占比 值为1:4,年轻代占整个堆栈的 1/5。

配置参数	说明	示例
-XX:SurvivorRatio=n	年轻代中Eden区与两个Survivor区的 比值。	-XX:SurvivorRatio=4 ,设置 年轻代中Eden区与Survivor区的大小 比值。如果设置为4,那么两个 Survivor区与一个Eden区的比值为 2:4,一个Survivor区占整个年轻代 的1/6。
-XX:MaxPermSize=n	设置持久代大小。	-XX:MaxPermSize=16m , 设置 持久代大小为16 MB。
- XX:MaxTenuringThreshold=n	设置垃圾最大年龄。	<ul> <li>XX:MaxTenuringThreshold=0</li> <li>,设置垃圾最大年龄。</li> <li>如果设置为0,那么年轻代对象不经过Survivor区,直接进入年老代。对于年老代比较多的应用,提高了效率。</li> <li>如果将此值设置为较大值,那么年轻代对象会在Survivor区进行多次复制,增加了对象在年轻代的存活时间,增加在年轻代即被回收的概率。</li> </ul>

### 调优回收器GC

### 吞吐量优先的GC典型配置参数

配置参数	说明	示例
-XX:+UseParallelGC	选择垃圾收集器为并行收集器。	-Xmx3800m -Xms3800m -Xmn2g -Xss128k -XX:+UseParallelGC - XX:ParallelGCThreads=20 , -XX:+UseParallelGC 此配置仅 对年轻代有效,即在示例配置下,年 轻代使用并发收集,而年老代仍旧使 用串行收集。
	配置并行收集器的线程数,即同时多 少个线程一起进行垃圾回收。	-Xmx3800m -Xms3800m -Xmn2g -Xss128k -XX:+UseParallelGC -
-XX:ParallelGCThreads	⑦ 说明 此值建议配置与处 理器数目相等。	XX:ParallelGCThreads=20 , -XX:ParallelGCThreads=20 表 示配置并行收集器的线程数为20 本

配置参数	说明	示例
	配置年老代垃圾收集方式为并行收 集。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseParallelGC
-XX:+UseParallelOldGC	⑦ 说明 JDK 6.0支持对年老 代并行收集。	-XX:ParallelGCThreads=20 - XX:+UseParallelOldGC , - XX:+UseParallelOldGC 表示对 年老代进行并行收集。
-XX:MaxGCPauseMillis	设置每次年轻代垃圾回收的最长时间,如果无法满足此时间,JVM会自 动调整年轻代大小,以满足此值。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseParallelGC - XX:MaxGCPauseMillis=100 , -XX:MaxGCPauseMillis=100 设 置每次年轻代垃圾回收的最长时间为 100 ms。
- XX:+UseAdaptiveSizePolicy	设置此选项后,并行收集器自动选择 年轻代区大小和相应的Survivor区比 例,以达到目标系统规定的最低响应 时该间或者收集频率,该值建议使用 并行收集器时,并且一直打开。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k -XX:+UseParallelGC -XX:MaxGCPauseMillis=100 - XX:+UseAdaptiveSizePolicy

### 响应时间优先的GC典型配置参数

配置参数	说明	示例
-XX:+UseConcMarkSweepGC	设置年老代为并发收集。 ⑦ 说明 配置了 - XX:+UseConcMarkSweepGC ,建议年轻代大小使用 - Xmn 设置。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k - XX:ParallelGCThreads=20 - XX:+UseConcMarkSweepGC - XX:+UseParNewGC
-XX:+UseParNewGC	设置年轻代为并行收集。 可与CMS收集同时使用。JDK 5.0以 上版本,JVM根据系统配置自行设 置,无需再设置此值。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k - XX:ParallelGCThreads=20 - XX:+UseConcMarkSweepGC - XX:+UseParNewGC

### 最佳实践·Java

配置参数	说明	示例		
- XX:CMSFullGCsBeforeCompactio n	由于并发收集器不对内存空间进行压 缩、整理,所以运行一段时间以后会 产生"碎片",使得运行效率降低。 此值设置运行多少次GC以后对内存 空间进行压缩、整理。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k - XX:+UseConcMarkSweepGC - XX:CMSFullGCsBeforeCompactio n=5 - XX:+UseCMSCompactAtFullColle ction , - XX:CMSFullGCsBeforeCompactio n=5 ,表示运行GC5次后对内存空 间进行压缩、整理。		
	打开对年老代的压缩。	-Xmx3550m -Xms3550m -Xmn2g -Xss128k -		
- XX:+UseCMSCompactAtFullColle ction	⑦ 说明 该值可能会影响性 能,但是可以消除碎片。	XX:+UseConcMarkSweepGC - XX:CMSFullGCsBeforeCompactio n=5 - XX:+UseCMSCompactAtFullColle ction		

### 用于辅助的GC典型配置参数

配置参数	说明
-XX:+PrintGC	用于输出GC日志。
-XX:+PrintGCDetails	用于输出GC日志。
	用于输出GC时间戳(JVM启动到当前日期的总时长的时间戳形式)。示例如 下:
-XX:+PrintGCTimeStamps	0.855: [GC (Allocation Failure) [PSYoungGen: 33280K- >5118K(38400K)] 33280K->5663K(125952K), 0.0067629 secs] [Times: user=0.01 sys=0.01, real=0.00 secs]
	用于输出GC时间戳(日期形式)。示例如下:
-XX:+PrintGCDateStamps	2022-01-27T16:22:20.885+0800: 0.299: [GC pause (G1 Evacuation Pause) (young), 0.0036685 secs]
-XX:+PrintHeapAtGC	在进行GC前后打印出堆的信息。
-Xloggc:/logs/gc.log	日志文件的输出路径。

# 7.2. JVM内存配置最佳实践

本文介绍如何在容器环境下配置JVM堆参数大小。

### 背景信息

当您的业务是使用Java开发,且设置的JVM堆空间过小时,程序会出现系统内存不足OOM (Out of Memory)的问题。特别是在容器环境下,不合理的JVM堆参数设置会导致各种异常现象产生,例如应用堆大小还未到达设置阈值或规格限制,就因为OOM导致重启等。

### 通过-XX:MaxRAMPercentage限制堆大小(推荐)

- 在容器环境下, Java只能获取服务器的配置, 无法感知容器内存限制。您可以通过设置 -xmx 来限制JVM 堆大小, 但该方式存在以下问题:
  - 当规格大小调整后,需要重新设置堆大小参数。
  - 当参数设置不合理时,会出现应用堆大小未达到阈值但容器OOM被强制关闭的情况。

⑦ 说明 应用程序出现OOM问题时,会触发Linux内核的OOM Killer机制。该机制能够监控占用过 大内存,尤其是瞬间消耗大量内存的进程,然后它会强制关闭某项进程以腾出内存留给系统,避免 系统立刻崩溃。

### ● 推荐的JVM参数设置。

-XX:+UseContainerSupport -XX:InitialRAMPercentage=70.0 -XX:MaxRAMPercentage=70.0 -XX:+Pri ntGCDetails -XX:+PrintGCDateStamps -Xloggc:/home/admin/nas/gc-\${POD\_IP}-\$(date '+%s').log -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/admin/nas/dump-\${POD\_IP}-\$(date '+ %s').hprof

参数	说明
-XX:+UseContainerSupport	使用容器内存。允许JVM从主机读取cgroup限制,例如可用的CPU和RAM, 并进行相应的配置。当容器超过内存限制时,会抛出OOM异常,而不是强制 关闭容器。
-XX:InitialRAMPercentage	设置JVM使用容器内存的初始百分比。建议与 – XX:MaxRAMPercentage 保持一致,推荐设置为70.0。
-XX:MaxRAMPercentage	设置JVM使用容器内存的最大百分比。由于存在系统组件开销,建议最大不超过75.0,推荐设置为70.0。
-XX:+PrintGCDetails	输出GC详细信息。
-XX:+PrintGCDateStamps	输出GC时间戳。日期形式,例如2019-12-24T21:53:59.234+0800。
-Xloggc:/home/admin/nas/g c-\${POD_IP}-\$(date '+%s').l og	GC日志文件路径。需保证Log文件所在容器路径已存在,建议您将该容器路 径挂载到NAS目录或收集到SLS,以便自动创建目录以及实现日志的持久化 存储。
-XX:+HeapDumpOnOutOfMemor yError	JVM发生OOM时,自动生成DUMP文件。
-XX:HeapDumpPath=/home/ad min/nas/dump-\${POD_IP}-\$(da te '+%s').hprof	DUMP文件路径。需保证DUMP文件所在容器路径已存在,建议您将该容器 路径挂载到NAS目录,以便自动创建目录以及实现日志的持久化存储。

### 参数说明如下。

? 说明

- 使用 -XX:+UseContainerSupport 参数需JDK 8u191+、JDK 10及以上版本。
- JDK 11版本下日志相关的参数 -XX:+PrintGCDetails 、 -XX:+PrintGCDateStamps 、 -X1 oggc:\$LOG\_PATH/gc.log 参数已废弃,请使用参数 -Xlog:gc:\$LOG\_PATH/gc.log 代替。
- o Dragonwell 11暂不支持 \${POD IP} 变量。
- 如果您没有将/*home/admin/nas*容器路径挂载到NAS目录,则必须保证该目录在应用启动前已存在,否则将不会产生日志文件。

### 通过-Xms-Xmx限制堆大小

- 您可以通过设置 -xms 和 -xmx 来限制堆大小,但该方式存在以下两个问题:
  - 当规格大小调整后,需要重新设置堆大小参数。
  - 当参数设置不合理时,会出现应用堆大小未达到阈值但容器OOM被强制关闭的情况。

⑦ 说明 应用程序出现OOM问题时,会触发Linux内核的OOM Killer机制。该机制能够监控占用过 大内存,尤其是瞬间消耗大量内存的进程,然后它会强制关闭某项进程以腾出内存留给系统,避免 系统立刻崩溃。

### ● 推荐的JVM参数设置。

-Xms2048m -XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:/home/admin/nas/gc -\${POD\_IP}-\$(date '+%s').log -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/admin /nas/dump-\${POD IP}-\$(date '+%s').hprof

### 参数说明如下。

参数	说明
-Xms	设置JVM初始内存大小。建议与 - Xmx 相同,避免每次垃圾回收完成后 JVM重新分配内存。
-Xmx	设置JVM最大可用内存大小。为避免容器OOM,请为系统预留足够的内存大小。
-XX:+PrintGCDetails	输出GC详细信息。
-XX:+PrintGCDateStamps	输出GC时间戳。日期形式,例如2019-12-24T21:53:59.234+0800。
-Xloggc:/home/admin/nas/g c-\${POD_IP}-\$(date '+%s').l og	GC日志文件路径。需保证Log文件所在容器路径已存在,建议您将该容器路 径挂载到NAS目录或收集到SLS,以便自动创建目录以及实现日志的持久化 存储。
-XX:+HeapDumpOnOutOfMemor yError	JVM发生OOM时,自动生成DUMP文件。
-XX:HeapDumpPath=/home/ad min/nas/dump-\${POD_IP}-\$(da te '+%s').hprof	DUMP文件路径。需保证DUMP文件所在容器路径已存在,建议您将该容器 路径挂载到NAS目录,以便自动创建目录以及实现日志的持久化存储。

### • 推荐的堆大小设置。

内存规格大小	JVM堆大小
1 GB	600 MB
2 GB	1434 MB
4 GB	2867 MB
8 GB	5734 MB

### 通过ossutil下载堆转储文件

- 1. 挂载容器日志目录至NAS。具体操作,请参见设置NAS存储。
- 2. 设置JVM参数。

其中Dump文件路径/home/admin/nas为NAS挂载目录:

-Xms2048m -XMx2048m -XX:+PrintGCDetails -XX:+PrintGCDateStamps -Xloggc:/home/admin/nas/ gc-\${POD\_IP}-\$(date '+%s').log -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/home/a dmin/nas/dump-\${POD\_IP}-\$(date '+%s').hprof

3. 当应用发生OOM时,会生成堆转储文件到NAS挂载目录,您可以利用ossutil工具,将该Dump文件下载 到本地进行分析。具体操作,请参见通过日志上传下载诊断应用。

### 常见问题

1. 容器出现137退出码的含义是什么?

当容器使用内存超过限制时,会出现容器OOM,导致容器被强制关闭。此时业务应用内存可能并未达到 JVM堆大小上限,所以不会产生Dump日志。建议您调小JVM堆大小的上限,为容器内其他系统组件预留 足够多的内存空间。

top	-	16:24	4:55 u	p 7	min,	, 0 use	rs, loa	d aver	age	: 0.0	02, 0	.06, 0.02		
Tas	ks:	3	total	,	1 ru	inning,	2 slee	ping,	0	stop	pped,	0 zombie	3	
%Cp	u(s	): (	0.7 us	,	0.0 \$	sy, 0.0	ni, 99.	3 id,	0.	0 wa,	0.0	0 hi, 0.0	si, <b>0.</b> 0	) st
KiB	Mei	m :	40827	20	total	L, 54	076 free	, 392	689	6 use	ed,	101748 but	ff/cache	
KiB	Swa	ap:		0	total	L,	0 free	,		0 use	ed.	<b>82080</b> ava	ail Mem	
	PID	USEI	R	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND	
	1	root	t	20	0	6265840	3.6g	21080	S	0.7	92.5	0:14.52	java	
	96	root	t	20	0	15264	3348	3088	S	0.0	0.1	0:00.00	sh	
	101	root	£	20	0	59596	4056	3536	R	0.0	0.1	0:00.03	top	
{"s	tat	us":	"Failu	re"	,"mes	ssage":"	command	termin	ate	d wit	h nor	n-zero exit	t code",'	"reas
on"	:"No	onZei	roExit	Cod	le","o	letails"	:{"cause	s":[{"	rea	son":	Exit	tCode", "mes	ssage":"1	137"}
]}}														

2. 堆大小和规格内存的参数值可以相同吗?

不可以。因为系统自身组件存在内存开销,例如使用SLS进行日志收集时会占用一小部分的内存空间, 所以不能将JVM堆大小设置为和规格内存大小相同的数值,需要为这些系统组件预留足够的内存空间。

3. 在JDK 8版本下设置 -XX:MaxRAMPercentage 值为整数时报错怎么处理?

这是JDK 8的一个Bug。具体信息,请参见Java Bug Database。

例如,在JDK 8u191版本下,设置 -XX:MaxRAMPercentage=70 ,这时JVM会启动报错。



### 解决方案如下:

o 方式一: 设置 -XX:MaxRAMPercentage 为 70.0 。

⑦ 说明 如果您使用了 -XX:InitialRAMPercentage 或 -XX:MinRAMPercentage ,参数值同 样不可设置为整数,需按照方式一的形式来设置。

○ 方式二:升级JDK版本至JDK 10及以上版本。

### 相关文档

● JVM参数配置说明

# 7.3. 设置Java应用的启动加速

SAE对Java应用在部署过程中的不同阶段的启动效率做了一系列优化与提升。本文介绍如何通过设置,提升 Java应用的启动效率。

### 前提条件

设置启动加速的Java应用必须为JAR包或者WAR包部署。

### 提升应用启动时的效率

如果您希望提升应用的启动速度,您可以参考以下步骤,在创建应用时选择Dragonwell 11环境,并在启动 命令设置中开启应用加速。本章节以在应用部署时设置相关参数为例,如果您希望为已经部署的应用提高启 动效率,您可以参考以下文档:

- 应用部署完成后挂载NAS
- 应用部署完成后配置启动命令

应用运行效果如下所示。更多信息,请参见Dragonwell Benchmark。



1. 登录SAE控制台。

- 2. 在左侧导航栏单击应用列表,在顶部菜单栏选择地域,单击创建应用。
- 3. 在应用基本信息页签,设置应用相关信息,配置完成后单击下一步:应用部署配置。
- 4. 在应用部署配置页签, 配置相关参数。

✓	2	3	4
应用基本信息	应用邰署配置		
* 技术栈语言:         ● Java	○ PHP	○ 其它语言	
* 应用部署方式:	<ul><li>WAR包錄石</li></ul>	○ JAR包部石 ?	
配置WAR包 *			
* 应用运行环境	apache-tomcat-8.5.42 使用Springboot, Dubbo War的用户,应用运行环境遴选提"apache-tomcat-XXX"。使用H	SF War的用户,应用运行环境请选择"EDAS-Container-XXX"	
* Java环境:	Dragonwell 11	✓ 如何选型Java环境	
* 文件上传方式:	上传WAR包	~	
* 上传WAR包:	选择文件		
	下载示例WAR包		
* 版本:	1608800008749 13/16 使用时间载为版本号		
* 时区设置:	UTC+8	$\sim$	

参数说明如下表所示。

参数名	说明
技术栈语言	选择Java。
应用部署方式	选择WAR包部署或者JAR包部署。本文中以WAR包 部署为例。
应用运行环境	选择您需要的应用环境,例如apache-tomcat- 8.5.42。
Java环境	选择Dragonwell 11。
文件上传方式	<ul> <li>可选择上传WAR包或WAR包地址。</li> <li>● 上传WAR包:单击选择文件,选择待部署WAR包。</li> <li>● WAR包地址:输入WAR包的存放地址。</li> <li>○ 注意 应用部署程序包名仅允许字母、数字,及短划线(-)、下划线(_)两个特殊符号。</li> </ul>
版本	设置应用版本号,您可以选择输入版本号或者单击 <b>使</b> 用时间戳为版本号将时间戳作为应用版本号。
时区设置	选择当前应用所在时区,例如UT C+8。

5. (可选)设置持久化存储,达到跨实例加速的效果。

i. 展开持久化存储区域, 打开应用NAS存储开关。

▶ 持久化存储 持久化存储数据、日志等 🔗 如何	设置持久化存储	
NAS文件系统 Beta OSS对象有	储	
启用NAS存储:		
存储类型: NAS文件系统		
使用已有的NAS文件系统: 293fb 293fb	● 注戦源: 293fb	~ C
挂载目录	容器路径	操作
1	/	Ē
请输入挂载目录,如/或者xx/xxx	请输入挂载容器路径,如/tmp	Ē
+添加		

ii. 在使用已有的NAS文件系统所在行的下拉列表中选择待挂载的NAS,并设置挂载源和容器路径。

6. 展开启动命令设置区域,选中开启应用启动加速(Qucikstart),设置持久化目录。

✓ 启动命令设置 设置Java应用」	自动和运行时需要的命令 🔗 如何设置启动命令
系统默认启动命令	CATALINA_OPTS="\$CATALINA_OPTS \$Options" catalina.sh run
	启动命令格式说明:java \$JAVA_OPTS \$CATALINA_OPTS [-Options] org.apache.catalina.startup.Bootstrap "\$@" start
options设置	-XX:+UnlockExperimentalVMOptions -XX:+UseWisp2 -XX:+UseG1GC -Xquickstart:verbos JVM参数设置说明
options 快捷设置	
✓ 开启微服务性能提	升(Wisp2 协程) ③
✓ 开启应用内存优化	(G1) 💿
✓ 开启应用启动加速	(Quickstart) 💿
持久化目录:	/tmp

参数说明如下表所示。

参数	说明
系统默认启动命令	SAE默认的启动命令。

参数	说明
options设置	配置JVM参数。关于参数详情,请参见JVM参数配置说 明、Tuning Java Virtual Machines和JVM Tuning: How to Prepare Your Environment for Performance Tuning。 如果您需要使用应用的远程调试功能,请配置以下命 令: - agentlib:jdwp=transport=dt_socket,addr ess=9000,server=y,suspend=n • transport:远程调试间的数据传输方式。 • address:远程调试的地址。与开启远程调试时设 置的调试端口保持一致,远程调试的详细说明,请 参见远程调试。
args设置	配置标准输出和错误输出的重定向命令,例如 1>>/t mp/std.log>&1 。
options快捷设置:只有Java环境为Dragonwell时可	以设置。
开启微服务性能提升(Wisp 2协程)	默认开启,开启后可以提升运行时多线程性能。
开启应用内存优化(G1)	默认开启,开启后可以针对多CPU与大容量内存场景, 降低GC时间,适用于GC需要优化、大数据等场景。
开启应用启动加速(Quickstart)	只有Java环境为Dragonwell 11时可以设置。选 中开启应用启动加速(Quickstart)并设置持久化 目录后,可以提升应用启动效率。
	开启应用加速后需要设置,设置的 <b>持久化目录</b> 推荐为

持久化目录

### 7. 单击下一步:确认规格。

8. 在确认规格页签,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。

9. 您可以通过以下方式验证配置是否生效。

。 方法一:

NAS存储的目录或者子目录,达到跨实例间的加速效 果。关于如何设置NAS存储,请参见<mark>设置NAS存储</mark>。 在应用的**变更记录**页面中查看应用变更详情,如果显示**执行成功**,则表示部署成功,即配置已生 效。

。 方法二:

在应用基本信息页面的实例部署信息页签查看实例的运行状态。如果运行状态显示为绿色的Running,表示应用部署成功,即配置已生效。

### 提升应用运行时的效率

如果您希望提升应用运行时的效率,您可以在在创建应用时选择Dragonwell环境,并在启动命令设置中开启 微服务性能提升(Wisp2协程)。具体步骤,请参见提升应用启动时的效率。

∨ 启动命令设置	设置Java应用启	記动和运行时需要的命令 🥜 如何设置启动命令	
系统默认	后动命令	CATALINA_OPTS="\$CATALINA_OPTS \$Options" catalina.sh run	
		 启动命令格式说明:java \$JAVA_OPTS \$CATALINA_OPTS [-Options] org.apache.catalina.startup.B	Bootstrap "\$@" start
optionsi	设置	-XX:+UnlockExperimentalVMOptions -XX:+UseWisp2 -XX:+UseG1GC -Xquickstart:verbos	JVM参数设置说明
options	快捷设置		
☑ 开启	微服务性能提升	┼ (Wisp2 协程)  ◎	
☑ 开启	应用内存优化	(61) 💿	
✓ 开启	应用启动加速	(Quickstart)	
持久化目	录:	/tmp	

应用运行效果如下所示。更多信息,请参见Dragonwell官方实时Benchmark。

jso	n					•
	-	OpenJ	ок	Dragonwell+V	/isp2	
	10000					
	1000					
tps/rt						
	1000					₩
	0-	-	-	100	-	1010

# 7.4. 为Spring Boot应用设置健康检查

对于Spring Boot的应用,除了使用HTTP或TCP端口检测来进行应用健康检查之外,您也可以使用Actuator 组件实现定制化健康检查。

### 背景信息

Actuator组件是Spring Boot提供的用来对应用系统进行自省和监控的功能模块,借助于Actuator,您可以 很方便地查看并统计应用系统的某些监控指标。您也可以通过Actuator组件自定义您的健康检查程序。更多 信息,请参见Spring Boot Actuator官方文档。

### 操作步骤

1. 在Maven中添加所需依赖。

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-actuator</artifactId>
</dependency>
```

2. 设置 application.properties 配置文件,显示健康检查详细信息。

```
management.endpoints.web.base-path= / #Actuator 2.0之前默认基础访问路径是"/"(则健康检查为``
/heath"),2.0之后默认访问路径是"/actuator" (则健康检查为``/actuator/heath")。
management.endpoint.health.show-details=always #显示健康检查详细信息,默认为never,即不显示
o
```

- 3. 您可以通过Actuator组件提供的自动配置的健康指示器或者自定义检查程序对应用进行检查。
  - 通过自动配置的健康检查器。

Actuator有些自动配置加载的健康检查指示器(HealthIndicator),例如若应用中使用了Redis、 MongoDB,那么RedisHealthIndicator以及MongoHealthIndicator就会被作为健康检查的一部分。更 多关于自动加载的配置信息,请参见自动配置加载。

您也可以禁用所有自动配置的健康指示器,或者禁用某个指示器的健康检查。

management.health.defaults.enable=false #禁用所有默认健康检查指示器。
management.health.mongo.enable=false #禁用MongoDB健康检查指示器。

 ↓ 注意 请您根据应用,选择需要开启的健康检查指示器。如果您开启了全部的指示器,可能 会导致健康检查持续失败。

○ 通过自定义检查程序。此时您可以通过/health/custom路径来单独获取这个健康指示器的结果。

新建*CustomHealthIndicator.java*文件,输入代码,实现您业务中特定的检查内容。例如检查数据库 连接是否正常,线程池状态等。示例代码如下所示:

```
@Component
public class CustomHealthIndicator extends AbstractHealthIndicator {
    @Override
    protected void doHealthCheck(Health.Builder builder) throws Exception {
        # 实现您业务中特定的检查内容。
        if (checkSomething()) {
            builder.up().withDetail("Item", "xxx").withDetail("error", "null");
        } else {
            builder.down().withDetail("Item", "xxx").withDetail("error", "xxxErrorCode
");
        }
}
```

4. 设置完成后,运行应用,进行健康检查。

### ○ 通过直接访问默认端口进行健康检查。示例命令如下:

curl 127.0.0.21:8080/health/custom #custom为健康检查指示器类名前缀,请以您定义的类名为准。

? 说明

- Actuator 2.0版本之前默认基础访问路径是/,则健康检查路径为/health。
- Actuator 2.0版本之后默认访问路径是/actuator,则健康检查路径 为/actuator/heath。

### 返回示例如下:

```
{"status":"UP","details":{"custom":{"status":"UP","details":{"Item","xxx","error","nu
ll"}}}
```

### 状态说明如下:

- UP: HTTP状态码为200, 说明健康检查成功。
- DOWN: HTTP状态码为503, 说明健康检查失败。

↓ 注意 如果这个检查路径中包含了多个健康指示器的结果(例如检查的路径是/health),
 只要有一个指示器的结果为DOWN, HTTP状态码就会是503, 即健康检查失败。

### • 在SAE控制台配置健康检查。具体步骤,请参见设置健康检查。

应用实	E例存活检查(Liveness配置) 应用业务就绪检查(Readines	s記畫)
检查容器是否正常,	不正常则重启实例	
注意事项: 使用自定义镜器	象部署应用的用户,需确保镜像内有curl与telnet工具(http请求检查通道	Icurl命令,TCP调口检查通过telnet命令)。避免因镜像不支持部分命令行工具,导致健康检查失败
检查方式:	● HTTP请求检查 ○ TCP端口检查 ○ 执行命令检查	
* 路径	/health/custom	
* 端口	8080	转化为执行命令 系统自动时先倒填写参数转化为执行命令提交到后台。 ##\$\$\$\$10
◇ 高级设置		1 resp=\$(curl -I -m 1 -o /dev/null -s -w %{http_co
返回内容关键字 包含		<pre>1 test presp tege you want test presp if you 3 then echo "check http://127.0.0.1:8080/health/cu 4 else echo "check http://127.0.0.1:8080/health/cu 5 fi 6 resps{Courl -m 1 http://127.0.0.1:8080/health/cu 7 if test -m "sponse contain UP"</pre>
协议 🖲 НТТР  〇	нттря	9 else echo ", but response does not contain UP"; 10 fi 11 oxit 0
延迟时间(秒):	表示应用启动之后多久开始探测,默认值10秒	
超时时问(秒):	表示探测超时时间,默认值1秒	
检查周期 (秒)	表示探测超时时间,默认值30秒	

# 8.应用访问阿里云数据库 8.1.如何设置阿里云关系型数据库RDS白名 单

应用在Serverless应用引擎SAE(Serverless App Engine)托管后,如果您需要访问阿里云关系型数据库 RDS(Relational Database Service),则需要为应用设置RDS白名单。本文以RDS MySQL实例为例,介绍如 何在不同场景下设置RDS IP白名单。

### 前提条件

- 购买RDS实例
- 组合购买公网NAT网关和弹性公网IP
- 设置部署在SAE的应用访问公网

### 背景信息

RDS支持MySQL、SQL Server、PostgreSQL和MariaDB TX引擎。更多信息,请参见云数据库RDS简介。

关于如何设置白名单,请参见以下文档:

- RDS MySQL设置白名单
- RDS PostgreSQL设置白名单
- RDS MariaDB设置白名单
- RDS PPAS设置白名单

### 场景一:应用访问本VPC内的RDS数据库

- 1. 获取SAE应用的VPC和vSwitch的IP地址。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击具体应用名称。
  - iii. 获取IP地址。
    - VPC: 在基本信息页签的应用信息区域内,单击VPC名称,跳转至专有网络管理控制台,然后 在基本信息页签,复制并保存IPv4网段。
    - vSwitch: 在基本信息页签的应用信息区域内,单击vSwitch名称,跳转至专有网络管理控制 台,然后在交换机基本信息页面,复制并保存IPv4网段。
- 2. 访问RDS实例列表,在上方选择地域,然后单击目标实例ID。
- 3. 在左侧导航栏选择数据安全性。
- 4. 确认IP白名单模式。

⑦ 说明 较早创建的实例可能采用高安全模式。新创建的实例都采用通用模式。

通用白名单模式

白名单设置	102-807 IS. TOK
添加白名单分组	当前为通用白名单模式。该模式下,白名单可通过经典网络和专有网络访问实例。
> default	
高安全白名单模式	
白名单设置	106
添加白名单分组 🚱	网络隔离模式: 高安全白名单模式。以下白名单明确区分经典网络及专有网络。
> group 经典网络	

5. 单击default分组右侧的修改。

⑦ 说明 如有需要,也可以单击添加白名单分组,并自定义一个分组名称。

数据库管理	添加白名单分组 ④ 当前为通用白名单模式。该模式下,白名单可通过经典网络和专有网络访问实例。	
备份恢复	∽ defaut	修改
数据库连接	127.0.0.1	
或 19 年代 12 监控与报警	● 提示: 1、RDS IP白名单支持P段的格式(如X.X.X.X/X), 2、设置为127.0.0.1表示禁止所有地址访问, 白名单设置说明	
数据安全性		

6. 在弹出的修改白名单分组对话框中,将步骤1复制的VPC和vSwitch的IP地址配置在白名单中,然后单击确定。

修改白名单分组	×	
允许访问实例的网络类型	◎ 专有网络 ◎ 经典网络	
* 分组名称:	default	
* 组内白名单	192.168	
加载ECS内网IP	还可添加998个白名单	1
您可指定某个IP地址或者指定IP地址:若输入192.1 指定IP地址:若输入192.168 若要输入多个IP地址或IPF IP地址?	皆定某个IP段作为白名单。 68.0.1,表明允许该IP地址访问当前实例。 3.0.0/24,表明允许192.168.0.0到192.168.0.255之间所有的IP地址访问当前实例。 段,需用英文逗号分隔。如:192.168.0.1,192.168.0.0/24。 <mark>如何定位本地设备的公网</mark>	
<ul> <li>白名单IP段设置为 试后请立即修改。</li> <li>白名单IP地址设置</li> </ul>	90.0.0.0/0意味着对公网开放,请谨慎使用。若为了测试连接设置,测 3为127.0.0.1,表示禁止所有地址访问。	
• 新白名单将于1分	钟后生效。	
	确定取消	
⑦ 说明 单个实例 有空格。	最多添加1000个IP地址或IP段,以英文逗号分隔多个IP地址,	且逗号前后不

完成配置后,您部署在SAE上的应用便可访问本VPC内的RDS数据库。

### 场景二:应用跨VPC或跨地域访问RDS数据库

不同VPC和不同地域之间属于完全的逻辑隔离,故常规情况下不能跨VPC和跨地域访问RDS数据库。若您的应用想跨VPC或跨地域访问RDS数据库,请按照下面步骤进行相关配置。

- 1. 获取SAE应用的弹性公网IP和vSwitch的IP地址。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击具体应用名称。
  - iii. 在基本信息页签的应用信息区域内,单击vSwitch名称,跳转至专有网络管理控制台,然后在交换机基本信息页面,复制并保存IPv4网段。
  - iv. 在左侧导航栏选择NAT网关 > 公网NAT网关。
  - v. 在公网NAT网关页面,复制并保存目标实例的弹性公网IP。
- 2. 访问RDS实例列表,在上方选择地域,然后单击目标实例ID。
- 3. 在左侧导航栏选择数据安全性。
- 4. 确认IP白名单模式。

⑦ 说明 较早创建的实例可能采用高安全模式。新创建的实例都采用通用模式。

# 通用白名单模式 白名单设置 添加白名单分组 ● 当前为通用白名单模式 该模式下,白名单可通过经典网络和专有网络访问实例。 ◆ default 高安全白名单模式 「白名单设置 添加白名单分组 ● 网络隔离模式: 高安全白名单模式。以下白名单明确区分经典网络及专有网络。 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

5. 单击default分组右侧的修改。

⑦ 说明 如有需要,也可以单击添加白名单分组,并自定义一个分组名称。

数据库管理	添加白名单分组 当前为通用白名单模式。该模式下,白名单可通过经典网络和专有网络动向实例。	
备份恢复	✓ default	修改
数据库连接	127.0.0.1	
数据库代理		
监控与报警	● 提示: 1、RDS IP白名单支持P段的格式(加X.X.X.X/X), 2、设置为127.0.0.1表示禁止所有地址访问,白名单设置说明	
数据安全性		

6. 在弹出的修改白名单分组对话框中,将步骤2复制的弹性公网IP和vSwitch的IP地址配置在白名单中,然 后单击确定。

修改白名单分组	×	
允许访问实例的网络类型	<ul> <li>专有网络</li> <li>经典网络</li> </ul>	
* 分组名称:	default	
* 组内白名单	192.168. ,116.62.	
加载ECS内网IP		
您可指定某个IP地址或者指 指定IP地址:若输入192.1 指定IP段:若输入192.168 若要输入多个IP地址或IP的 IP地址?	皆定某个IP段作为白名单。 68.0.1,表明允许该IP地址访问当前实例。 3.0.0/24,表明允许192.168.0.0到192.168.0.255之间所有的IP地址访问当前实例。 段,需用英文逗号分隔。如:192.168.0.1,192.168.0.0/24。 <mark>如何定位本地设备的公网</mark>	
<ul> <li>自名单IP段设置为 试后请立即修改。</li> <li>自名单IP地址设置</li> <li>新白名单将于1分</li> </ul>	g0.0.0.0/0意味着对公网开放,请谨慎使用。若为了测试连接设置,测 3为127.0.0.1,表示禁止所有地址访问。 钟后生效。	
	确定取消	
<ol> <li>说明 单个实例</li> <li>右空格</li> </ol>	最多添加1000个IP地址或IP段,以英文逗号分隔多个IP地址,	且逗号前后不能

完成配置后,您部署在SAE上的应用便可跨VPC、跨地域访问RDS数据库。

### 更多信息

除为IP地址添加白名单的方式外,您还可以通过添加安全组的方式来访问RDS数据库。具体信息,请参见以下文档:

- RDS MySQL设置安全组
- RDS PostgreSQL设置安全组
- RDS MariaDB T X 设置安全组

## 8.2. 如何设置云数据库Redis版白名单

应用在Serverless应用引擎SAE(Serverless App Engine)托管后,如果您需要访问云数据库Redis版 (ApsaraDB for Redis),则需要设置Redis白名单。本文介绍如何在不同场景下设置Redis白名单。

### 场景一:应用访问本VPC内的Redis数据库

- 1. 获取SAE应用的VPC和vSwitch的IP地址。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击具体应用名称。

iii. 获取IP地址。

- VPC: 在基本信息页签的应用信息区域内,单击VPC名称,跳转至专有网络管理控制台,然后 在基本信息页签,复制并保存IPv4网段。
- vSwitch: 在基本信息页签的应用信息区域内,单击vSwitch名称,跳转至专有网络管理控制 台,然后在交换机基本信息页面,复制并保存IPv4网段。
- 2. 访问Redis实例列表,在上方选择地域,然后单击目标实例ID。
- 3. 在左侧导航栏, 单击白名单设置。
- 4. 在default默认安全组,单击修改。

⑦ 说明 您也可以单击添加白名单分组创建一个新的分组。分组名称长度为2~32个字符,由小写字母、数字或下划线组成,需以小写字母开头,以小写字母或数字结尾。

5. 在弹出的修改白名单分组对话框中,将步骤1复制的VPC和vSwitch的IP地址配置在白名单中,然后单击确定。

修改白名单分	组	×
* 分组名称	default	
* 添加方式	● 手动添加 ○ 加载ECS私网IP	
* 组内白名单	192.168. 192.168.	
		确定取消
? 说明	单个实例最多添加1000个IP地址或IP段,以英文	、逗号分隔多个IP地址,且逗号前后不能

完成配置后,您部署在SAE上的应用便可访问本VPC内的Redis数据库。

### 场景二:应用跨VPC或跨地域访问Redis数据库

不同VPC和不同地域之间逻辑上完全隔离,故常规情况下不能跨VPC和跨地域访问Redis数据库。若您的应用 想跨VPC或跨地域访问Redis数据库,请按照以下步骤进行相关配置。

1. 前提准备。

有空格。

购买NAT网关和弹性公网IP组合包并保证SAE应用可以访问公网。具体步骤,请参见部署在SAE上的应用 如何访问公网。

- 2. 获取SAE应用的弹性公网IP和vSwitch的IP地址。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击具体应用名称。
  - iii. 在基本信息页签的应用信息区域内,单击vSwitch名称,跳转至专有网络管理控制台,然后在交换机基本信息页面,复制并保存IPv4网段。
  - iv. 在左侧导航栏选择NAT网关 > 公网NAT网关。
  - v. 在公网NAT网关页面,复制并保存目标实例的弹性公网IP。

- 3. 访问Redis实例列表,在上方选择地域,然后单击目标实例ID。
- 4. 在左侧导航栏,单击白名单设置。
- 5. 在default 默认安全组,单击修改。

⑦ 说明 您也可以单击添加白名单分组创建一个新的分组。分组名称长度为2~32个字符,由小写字母、数字或下划线组成,需以小写字母开头,以小写字母或数字结尾。

6. 在弹出的修改白名单分组对话框中,将步骤2复制的弹性公网IP和vSwitch的IP地址配置在白名单中,然 后单击确定。

修改白名单分组		×
* 分组名称	default	
* 添加方式	● 手动添加 ○ 加戰ECS私网IP	
* 组内白名单	192.168	
	确定 取消	v and

⑦ 说明 单个实例最多添加1000个IP地址或IP段,以英文逗号分隔多个IP地址,且逗号前后不能 有空格。

完成配置后,您部署在SAE上的应用便可跨VPC、跨地域访问Redis数据库了。

### 更多信息

除为IP地址添加白名单的方式外,您还可以通过添加安全组的方式来访问Redis数据库。具体信息,请参见通 过ECS安全组设置Redis白名单。

# 8.3. 如何设置云数据库MongoDB版白名单

应用在Serverless应用引擎SAE(Serverless App Engine)托管后,如果您需要访问云数据库MongoDB版 (ApsaraDB for MongoDB),则需要设置MongoDB白名单。本文介绍如何在不同场景下设置MongoDB白 名单。

### 场景一:应用访问本VPC内的MongoDB数据库

1. 获取SAE应用的VPC和vSwitch的IP地址。

- i. 登录SAE控制台。
- ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击具体应用名称。
- iii. 获取IP地址。
  - VPC: 在基本信息页签的应用信息区域内,单击VPC名称,跳转至专有网络管理控制台,然后 在基本信息页签,复制并保存IPv4网段。
  - vSwitch: 在基本信息页签的应用信息区域内,单击vSwitch名称,跳转至专有网络管理控制 台,然后在交换机基本信息页面,复制并保存IPv4网段。

- 2. 登录MongoDB管理控制台。
- 3. 根据实例类型,在左侧导航栏,单击副本集实例列表、分片集群实例列表或Serverless实例列表。
- 4. 在页面左上角,选择实例所在的资源组和地域。
- 5. 单击目标实例ID或目标实例所在行操作列的管理。
- 6. 根据实例类型,执行对应操作。
  - 单节点实例、副本集实例和分片集群实例
     在目标实例页面的左侧导航栏,单击数据安全性 > 白名单设置。
  - Serverless实例

在目标实例页面的左侧导航栏,单击安全设置。

7. 在添加白名单分组区域,单击default分组操作列的并选择手动修改。

⑦ 说明 如有需要,也可以单击添加白名单分组,并自定义一个分组名称。

8. 在手动修改面板的允许访问IP名单文本框内输入步骤1复制的VPC和vSwitch的IP地址,然后单击确定。

≕动修改	×
1 0.0.0.0/0代表不设IP访问的限制,数据库将会有高安全风险。建订务器外网IP/IP段设为可访问权限。	义仅将您的Web服
分组名	
IPv4 IPv6	
允许访问IP名单	
192.168. 192.168.	
	NEW E
	<b></b>
確定 取消	
1000	

完成配置后,您部署在SAE上的应用便可访问本VPC内的MongoDB数据库。
## 场景二:应用跨VPC或跨地域访问MongoDB数据库

不同VPC和不同地域之间逻辑上完全隔离,故常规情况下不能跨VPC和跨地域访问MongoDB数据库。若您的应用想跨VPC或跨地域访问MongoDB数据库,请按照以下步骤进行相关配置。

1. 前提条件。

购买NAT网关和弹性公网IP组合包并保证SAE应用可以访问公网。具体操作,请参见部署在SAE上的应用 <mark>如何访问公网</mark>。

- 2. 获取SAE应用的弹性公网IP和vSwitch的IP地址。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击具体应用名称。
  - iii. 在基本信息页签的应用信息区域内,单击vSwitch名称,跳转至专有网络管理控制台,然后在交换机基本信息页面,复制并保存IPv4网段。
  - iv. 在左侧导航栏选择NAT网关 > 公网NAT网关。
  - v. 在公网NAT网关页面,复制并保存目标实例的弹性公网IP。
- 3. 登录MongoDB管理控制台。
- 4. 根据实例类型,在左侧导航栏,单击副本集实例列表、分片集群实例列表或Serverless实例列表。
- 5. 在页面左上角,选择实例所在的资源组和地域。
- 6. 单击目标实例ID或目标实例所在行操作列的管理。
- 7. 根据实例类型,执行对应操作。
  - 单节点实例、副本集实例和分片集群实例
     在目标实例页面的左侧导航栏,单击数据安全性>白名单设置。
  - Serverless实例

在目标实例页面的左侧导航栏,单击安全设置。

8. 在添加白名单分组区域,单击default分组操作列的并选择手动修改。

⑦ 说明 如有需要,也可以单击添加白名单分组,并自定义一个分组名称。

9. 在**手动修改**面板的**允许访问IP名单**文本框内输入步骤2复制的弹性公网IP和vSwitch的IP地址,然后单击确定。

手动修改	×
0.0.0.0/0代表不设IP访问的限制,数据库将会有高安全风险。建议仅将您务器外网IP/IP段设为可访问权限。	题的Web服
分组名 default	
IPV4 IPV6 允许访问IP名单 192.168116.62	
	N EW
確定取消	
⑦ 说明 单个实例最多添加1000个IP地址或IP段,以英	文逗号分隔多·

完成配置后,您部署在SAE上的应用便可跨VPC、跨地域访问MongoDB数据库。

# 更多信息

有空格。

除为IP地址添加白名单的方式外,您还可以通过添加安全组的方式来访问MongoDB数据库。具体信息,请参见MongoDB设置安全组。

# 9.将应用从Web+迁移到SAE

Serverless应用引擎SAE(Serverless App Engine)是面向应用的Serverless PaaS平台,帮助PaaS层用户免运维laaS、按量计费、低门槛微服务上云,将Serverless架构与微服务架构完美结合。本文介绍将不同应用从Web应用托管服务(Web App Service,简称Web+)平台迁移到SAE的操作步骤与常见问题。

## 背景信息

对于使用RDS、Redis等数据库的应用,迁移到SAE后对应的数据库可继续使用,并且可以通过Secret方式配置数据库连接(需要保证网络互通)。

# 将Tomcat、Java、PHP应用从Web+迁移到SAE

1. 使用阿里云账号登录SAE购买页。

⑦ 说明 SAE支持免费开通,但开通时您的账户余额需要大于0。

- 2. 在SAE控制台创建应用。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击**创建应用**。
  - iii. 在应用基本信息页签,填入名称、选择网络配置(可选自动配置)与规格,单击下一步:应用部 署配置。

1 应用基本信息				▲ 创建完成
您当前账户下	:n-shanghai的应用总实例数限制为300个,目前您已使用0个,如需增加额度请提]	单申请。 如果您的微服务质	<b>2用实例总数超过30个,推荐使用商业版独享的注册中心。</b>	
* 应用名称:				
* 专有网络配置:				
* 命名空间 ?	清选择	$\sim$	$rac{1}{2}$	
* vSwitch:	游选择vSwitch	~ C (支持多选)		
* 安全组:	溶輸入安全组全名重词 く C			
应用实例数 ?:	2 个 (最多创建50个)			
* VCPU:	0.5Core 1Core 2Core 4Core 8Core 12Core 16Core 32Core			
* 内存:	1GiB 2GiB 4GiB			
应用描述:	介绍应用的基本情况			
			٩,	00
配置费用: 参考价格,	AND DESCRIPTION OF THE OWNER.			下一步:应用部署配置

- iv. 在应用部署配置页签, 技术栈语言可选Java或PHP。
  - Java和Tomcat应用选择Java技术栈(以Tomcat应用为例)

根据实际情况选择运行环境(Tomcat版本)与JDK版本,并上传部署包(原Web+ Tomcat应用 选择WAR包部署,原Java应用选择JAR包部署)。

✓ 应用基本信息	2 应用部署配置		(A) MRS2N
* 技术栈语言:	🔿 РНР	◯ 其它语言(如Python、O	C++、Go、.NET、Node.js等)
* 应用部署方式: 🛛 撬缴	الله WARte	西 🕜 三 asignation of a state of	
配置WAR包 *			
* 应用运行环境	apache-tomcat-8.5.42 使用Springboot、Dubbo War的用户,应用运行环境遴选	く 摩'apache-tomcat-XXX"。使用HSF War的用户,应用运行环境遴选择"EDAS-Cont	ainer-3004"
* Java环境:	Open JDK 8	~	如何逃型ava环境
* 文件上传方式:	上傳WAR包	~	
*上传WAR包:	选择文件 下载示例MAR包		
* 版本:	1650440789550 13/16 使用时间最为版	专	
* 时区设置:	UTC+8	~	
配置费用: ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	a second data		上一步:应用基本信息 下一步:输入规格

■ PHP应用选择PHP技术栈

✓ - 应用基本信息	ļ.		2 应用部署配置	3 	
* 技术栈语言:	🔿 Java		PHP	○ 其它语言(如Python、C+	+、Go、.NET、Node.js等)
* 技术栈版本:	PHP-7.3	$\sim$			
* 应用部署方式:	() 續像		● ZIP包部器		
配置 Zip 包 *					
• 运行组件:					
◆ PHP 环境:	请选择		$\sim$		
PHP扩展					
PHP PECL	扩展				
* 软件包:					
* 文件上传方式	t:	上传Zip包		~	
* 上传Zip包:	[	选择文件 下载Zip包样例			
* 版本:		1644910200184 13/16	使用时间戳为版本号		
* 时区设置:		UTC+8		~	

根据需要选择相应的环境(PHP版本)与扩展,并上传ZIP包。

#### v. 按需配置以下高级设置。

* 上传WAR包:	hello-sae.war (3.94M)	×	^			
	下統元的MARQ					
* 版本:	1650440789550 13/16 使用时间减少版本等					
* 时区设置:	UTC+8 ~					
> 启动命令设置 设置Java应用启动和设	if9#需要的命令 🔗 如何设置周动命令					
> Java Tomcat 设置 设置 lava Tomcat读	间、应用访问路径、编码推过等。 🔗 如何设置 Java Tomcat参数					
> 环境变量设置 设置容器运行环境中的	9-些变量,便于都電振気能改更容器能置。 🔗 如何设置环境变量					
> Hosts绑定设置 请遵循标准Hosts格示	以其可能置,设置Hosts绑定,便于逾过城名访问应用 🔗 如何设置Hosts绑定					
> 國內用總導給意及選 用于判断容認知用戶业务是否正常运行 🥜 如何设置应用總導給應						
> 如用山/人公開动荷设置   如用山/人公開设置,如此用即開过程中下報公開依赖或应用即開始设置做否小程序自名单等。 🥜 如用设置应用出公用访问						
> 血用生命與時當環设置 生命與期降本主义,管理应用容器在运行前和关闭前的一些动作,如环境准备、优雅下线等 🔗 如何设置应用生命间的						
>日本な実験務は登里日本の無規則、総府业务日本輸出部に5、使于施一管理和分析の知識の構成である実施						
> 持久以存補。 设置持久以 存储数据 🔗 如何设置持久以存储						
> 配置管理 通过挂板规型文件的方式,向客都中注入规型信息。 🔗 如何进行配置管理						
> 做服务无损上下线 实现 Spring Clou	4(Cubbo 服务统升级/面包/扩播前时,消费捐献品对别断服务列表,保证定服无后和通应。 🔗 设置的服务无法上下线					
配置裁用:	a nerves from	上一步:应用基本信息	下一步: 确认规格			

更多信息,请参见高级配置。

- vi. 单击下一步:确认规格,在确认规格页签,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。
- 3. 在SAE控制台应用详情页面,添加应用对外访问方式。
  - i. 应用创建完成后,单击目标应用,进入应用详情页面。
  - ii. 在基本信息页面的应用访问设置区域,根据原本Web+应用使用的负载均衡器类型选择使用公网负载均衡访问或者私网负载均衡访问,并且可以选择新建SLB或者使用已有SLB。具体操作,请参见为应用绑定SLB。

访问方式配置完成后,即可通过访问方式配置对应的地址访问Web应用。

⑦ 说明 若原本的Web+应用使用域名方式访问,迁移到SAE后,可以通过阿里云云解析产品进行域名设置。

# 将Node.js、Go、Python、ASP.NET Core、Ruby、Native应用从Web+迁 移到SAE

1. 使用阿里云账号登录SAE购买页。

⑦ 说明 SAE支持免费开通,但开通时您的账户余额需要大于0。

- 2. 在SAE控制台创建应用。
  - i. 登录SAE控制台。
  - ii. 在左侧导航栏单击**应用列表**,在顶部菜单栏选择地域,单击创建应用。

 iii. 在应用基本信息页签,填入名称、选择网络配置(可选自动配置)与规格,单击下一步:应用部 署配置。

1 应用基本信息			(3)	
您当前账户】	「cn-shanghai的应用总实例数限制为300个,目前您已使用0个,如需增加	1额度请提工单申请。 如果您的微服务/	立用实例总数超过30个,推荐使用商业版独享的注册中心。	
* 应用名称:				
* 专有网络配置:				
* 命名空间 ?	请选择	$\sim$	ia题率vpc ~ C	
* vSwitch:	请选择vSwitch	~ C (支持多选)		
* 安全组:	请输入安全组全名查询	~ G		
应用实例数?:	2 个 (最多创建50个)			
* VCPU:	0.5Core 1Core 2Core 4Core 8Core 12Core 16Core	32Core		
* 内存:	1GiB 2GiB 4GiB			
应用描述:	介绍应用的基本博兄			
			00	
配置费用: 参考价格,	and the second lines			下一步: 应用部署配置

iv. 在应用部署配置页签, 技术栈语言选择其它语言。

此时将默认使用自定义镜像的方式部署应用。因此,需要您将原来Web+应用先构建成容器镜像, 才能在SAE上通过该方式重新部署。具体操作,请参见制作Node.js镜像。

✓ - 应用基本信息	1	2 应用部署配置		3 确认规格	(4) (1)))))
* 技术栈语言:	🔿 Java	○ рнр	• 其它	语言(如Python、C++、Go、.NET、Node.js等)	
* 应用部署方式:	<ul> <li>6</li> <li>6</li> </ul>				
配置镜像*					
我的阿里云镜像	Demo镜像 公有镜像 其它阿里云账号	私有镜像			
● 續像服务个人」	版 ( 编像服务企业版 如何跨region同步镜像				
注意事项: 由于SAE中镜	像拉取的策略不是Always,强烈建议不要使用 latest 作为境份	mag命名,否则会影响部署升级和回滚功能	正常生效。		
镜像仓库命名空间	]: · · · · · · · · · · · · · · · · · · ·	ICFA: Q			
4	6	类型PRIVATE	来源:ALI_HUB	请选择	~
4	6	类型:PRIVATE	来源AU_HUB	演选择	~
配置费用: 参考价格,	CONTRACTOR OF THE OWNER OF THE	-		<u></u>	应用基本信息 下一步:确认规格

#### v. 按需配置以下高级设置。

<u>4</u>	类型PRIVATE	来源:ALI_HUB	请选择	~			
<u> 4</u>	类型:PRIVATE	来源:ALI_HUB	调选择	~			
				く 1 下一页 >			
> 启动命令设置 设置容器启动和运行时需要的命令 🔗 如何说	建置启动命令						
> 环境变量设置 设置容器运行环境中的一些变量,便于部署后员	活变更容器配置。 🔗 如何设置环境变量						
> Hosss编定设置 通道领标准Hosss指式填写能型,设置Hosss编定,使于通过域名访问应用 🥜 如何设置Hosss编定							
> 📮 应用键制绘图设置 用于判断者器和用户业务是否正常运行 🥜 如何设置应用键制绘图							
> 血用生命原则管理设置 生命间明脚本定义,管理应用启器在运行前和关闭的的一些动作,如环境普督、优推下线等 ♂ 如何设置应用生命间明							
> 应用出/入公网访问设置 应用出/入公网设置,如应用部署过程	中下载公网依赖或应用部署完设置微信小程序白名单等	。 🔗 如何设置应用出公网访问					
> 日本收集服务 设置日本收集规则,能将业务日本输出到SLS,使于统一管理和D分析 🥜 如何设置日本收集							
> 持久化存储 设置持久化存储数据							
> 配置管理 通过挂载配置文件的方式,向容器中注入配置信息。							
				上一步:应用基本信息 下一步:确认规格			

更多信息,请参见<mark>高级配置</mark>。

- vi. 单击下一步:确认规格,在确认规格页签,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。
- 3. 在SAE控制台应用详情页面,添加应用对外访问方式。
  - i. 应用创建完成后,单击目标应用,进入应用详情页面。
  - ii. 在基本信息页面的应用访问设置区域,根据原本Web+应用使用的负载均衡器类型选择使用公网负载均衡访问或者私网负载均衡访问,并且可以选择新建SLB或者使用已有SLB。具体操作,请参见为应用绑定SLB。

访问方式配置完成后,即可通过访问方式配置对应的地址访问Web应用。

## 常见问题

#### 如何处理产品下线的退费与补偿?

Web+产品的费用说明,请参见使用说明。

SAE产品可按量付费,也可购买资源包进行抵扣。费用说明,请参见计费概述。

#### Web+不同技术栈类型的应用对应哪种SAE技术栈?

- Web+ Tomcat技术栈对应SAE Java技术栈,WAR包部署。
- Web+ Java技术栈对应SAE Java技术栈, JAR包部署。
- Web+ PHP技术栈对应SAE PHP技术栈, ZIP包部署。
- Web+ Node.js、Go、Python、ASP.NET Core、Ruby、Native等技术栈对应SAE其他语言,镜像部署方式。

#### Web+应用中有使用RDS、Redis等数据库,在SAE中是否支持呢?

如果原来Web+应用中有使用RDS、Redis等数据库,迁移到SAE时需要事先在对应RDS、Redis云数据库控制 台创建相应数据库。如果是已有数据库,则只需要将新部署的应用连接到已有数据库上,无需迁移相关数 据。

⑦ 说明 若原本的Web+应用使用域名方式访问,迁移到SAE后,可以通过阿里云云解析产品进 行域名设置。

### Web+应用迁移至SAE,对应用代码需要进行哪些改造?

- 对于使用Tomcat、Java、PHP技术栈的应用,应用代码不需要任何修改,可直接在SAE上进行应用发布。
- 对于使用Node.js、Go、Python、ASP.NET Core、Ruby、Native等技术栈的应用,应用代码不需要任何修改,但您需要将应用先构建成容器镜像,推荐使用阿里云容器镜像服务(ACR),在SAE上通过镜像的方式发布应用。

#### 配置迁移过程中,对业务运行有什么影响?

迁移过程中,Web+业务应用访问并不受影响,但需要使用SLB等产品进行流量切换。

#### 配置迁移的时间周期大概需要多久?

您可以基于现有的JAR、WAR包或者新构建的容器镜像,在SAE控制台重新发布一次即可。配置迁移的步骤, 请参见将Tomcat、Java、PHP应用从Web+迁移到SAE。