阿里云 容器服务

用户指南

文档版本:20180808



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

目录

法律	聿声明	I
通月	用约定	I
1 K	Kubernetes 集群	1
	1.1 概述	1
	1.2 阿里云 Kubernetes VS 自建 Kubernetes	1
	1.3 授权管理	3
	1.3.1 角色授权	4
	1.3.2 使用子账号	7
	1.3.3 创建自定义授权策略	9
	1.4 子账号Kubernetes应用权限配置指导	12
	1.5 集群管理	18
	1.5.1 创建 Kubernetes 集群	19
	1.5.2 SSH 访问 Kubernetes 集群	
	1.5.3 SSH密钥对访问Kubernetes集群	28
	1.5.4 通过 kubectl 连接 Kubernetes 集群	29
	1.5.5 添加已有节点	30
	1.5.6 扩容和缩容集群	
	1.5.7 升级集群	35
	1.5.8 删除集群	
	1.5.9 查看集群概览	40
	1.6 节点管理	41
	1.6.1 查看节点列表	42
	1.6.2 节点标签管理	43
	1.6.3 节点监控	46
	1.7 应用管理	47
	1.7.1 通过 Kubernetes Dashboard 创建应用	47
	1.7.2 使用镜像创建应用	50
	1.7.3 通过编排模板创建应用	54
	1.7.4 通过命令管理应用	58
	1.7.5 利用 Helm 简化应用部署	59
	1.7.6 创建服务	66
	1.7.7 指定节点调度	70
	1.7.8 服务伸缩	74
	1.7.9 查看服务	76
	1.7.10 更新服务	77
	1.7.11 查看容器	79
	1.7.12 变更容器配置	82

1.8 命名空间管理	
1.8.1 创建命名空间	84
1.8.2 为命名空间配置资源额度	
1.8.3 编辑命名空间	
1.8.4 删除命名空间	91
1.9 配置项	
1.9.1 创建配置项	
1.9.2 在 pod 中使用配置项	96
1.9.3 修改配置项	100
1.9.4 删除配置项	103
1.10 密钥管理	105
1.10.1 创建密钥	105
1.10.2 查看密钥	107
1.10.3 编辑密钥	108
1.10.4 删除密钥	109
1.11 路由管理	110
1.11.1 路由配置说明	110
1.11.2 通过 Web 界面创建路由	112
1.11.3 查看路由	119
1.11.4 变更路由	121
1.11.5 删除路由	122
1.12 应用目录管理	123
1.12.1 应用目录概述	
1.12.2 查看应用目录列表	124
1.13 服务目录	125
1.13.1 概述	125
1.13.2 开通服务目录	
1.14 镜像管理	127
1.14.1 查看镜像列表	
1.15 VPC下 Kubernetes 的网络地址段规划	129
1.16 负载均衡	132
1.16.1 概述	133
1.16.2 通过负载均衡(Server Load Balancer)访问服务	133
1.16.3 Ingress 监控配置	
1.16.4 Ingress 支持	141
1.17 存储管理	146
1.17.1 概述	146
1.17.2 使用阿里云云盘	147
1.17.3 安装插件	153
1.17.4 使用阿里云 NAS	157

1.17.5 使用阿里云 OSS	164
1.18 日志管理	168
1.18.1 概述	168
1.18.2 查看集群日志	168
1.18.3 通过阿里云日志服务采集 Kubernetes 集群日志	170
1.18.4 为 Kubernetes 和日志服务配置 Log4JAppender	174
1.18.5 利用 log-pilot + elasticsearch + kibana 搭建 kubernetes 日志解决方案.	179
1.19 监控管理	190
1.19.1 通过资源分组进行监控与告警	190
1.19.2 使用 Grafana 展示监控数据	194
1.19.3 与云监控集成与使用	200
1.20 安全管理	205
1.21 常见问题	206
1.21.1 收集 Kubernetes 诊断信息	206
1.21.2 集群创建失败	207
1.21.3 删除 Kubernetes 集群失败:ROS stack 无法删除	208
1.21.4 如何支持私有镜像	210
1.21.5 Helm 手动升级	210
1.21.6 如何支持私有镜像	211
2 Swarm 隹群	213
L Owarm 未相	
2.1 环境要求	213
2.1 环境要求 2.2 使用流程	213 214
 2.1 环境要求 2.2 使用流程 2.3 授权管理 	213 214 214
 2.1 环境要求	213 214 214 214
 2.1 环境要求	213 214 214 214 218
 2.1 环境要求	213 214 214 214 218 220
 2.1 环境要求	213 214 214 214 214 218 220 222
 2.1 环境要求	213 214 214 214 214 214 218 220 222 222
 2.1 环境要求	213 214 214 214 214 214 218 220 222 225 225
 2.1 环境要求	213 214 214 214 214 214 218 220 222 225 225 225 225 226
 2.1 环境要求	213 214 214 214 214 218 220 222 225 225 225 225 226 227
 2.1 环境要求	213 214 214 214 214 214 218 220 222 225 225 225 225 225 225 225 226 227 233
 2.1 环境要求	213 214 214 214 214 220 225 225 225 225 226 227 233 236
 2.1 环境要求	213 214 214 214 214 218 220 222 225 225 225 225 225 226 227 227 233 236 240
 2.1 环境要求	213 214 214 214 214 214 218 220 222 225 225 225 225 225 225 225 225
 2.1 环境要求	213 214 214 214 214 214 220 220 225 225 225 225 226 226 233 236 240 242 244
 2.1 环境要求	213 214 214 214 214 218 220 225 225 225 225 225 226 227 233 236 240 242 247
 2.1 环境要求. 2.2 使用流程	213 214 214 214 214 218 220 222 225 225 225 225 225 225 225 225
 2.1 环境要求. 2.2 使用流程	213 214 214 214 214 218 220 222 225 225 225 225 225 225 225 225

2.4.13 清理集群磁盘	250
2.4.14 登录镜像仓库	251
2.4.15 升级 Agent	252
2.4.16 升级 Docker Daemon	253
2.4.17 升级系统服务	255
2.5 节点管理	256
2.5.1 移除节点	
2.5.2 重置节点	
2.5.3 查看节点上运行的容器	259
2.6 安全组	
2.6.1 容器服务安全组规则	
2.6.2 检测安全组	263
2.6.3 重新绑定安全组	
2.7 镜像与模板管理	
2.7.1 查看镜像列表	
2.7.3 查看编排模板列表	
2.7.4 创建编排模板	
2.7.5 更新编排模板	269
2.7.6 下载编排模板	270
2.7.7 删除编排模板	271
2.7.8 另存编排模板	272
2.8 服务编排	273
2.8.1 简介	273
2.8.2 标签概览	
2.8.4 probe	
2.8.5 rolling_updates	278
2.8.6 depends	279
2.8.7 scale	
2.8.8 routing	
2.8.9 lb	282
2.8.10 日志	
2.8.11 global	
2.8.12 服务部署约束(affinity:service)	
2.8.13 external	
2.8.14 dris_options	
2.0.15 0011_KII_UISADIE	201 207
2.0.10 受里谷泱	، 201 مەم
4.0.11 谷奋里利 順皮9.0.10 古可田桃 海府	۵۵۵
2.0.10	۵۵۵
2.0.19 个文持的 Docker Compose 标签	
2.9 应用官理	

2.9.1 创建应用	
2.9.2 应用参数配置说明	296
2.9.3 限制容器资源	
2.9.4 高可用性调度	
2.9.5 指定节点调度	
2.9.6 指定多节点调度	303
2.9.7 查看应用详情	
2.9.8 启停应用	
2.9.9 变更应用配置	
2.9.10 重新部署应用	
2.9.11 删除应用	310
2.9.12 运行离线任务	
2.9.13 定时任务	315
2.9.14 系统默认应用列表	317
2.10 配置项	
2.10.1 创建配置项	318
2.10.2 修改配置项	320
2.10.3 使用配置项实现多套环境	
2.10.4 删除配置项	327
2.11 服务管理	328
2.11.1 使用说明	
2.11.2 查看服务详情	
2.11.3 启停服务	
2.11.4 变更服务配置	
2.11.5 重新调度服务	
2.11.6 删除服务	
2.12 网络管理	332
2.12.1 跨主机互联的容器网络	332
2.12.2 容器服务 VPC 网络使用指南	335
2.13 数据卷管理	341
2.13.1 概述	342
2.13.2 创建 OSSFS 数据卷	342
2.13.3 创建云盘类型的数据卷	346
2.13.4 查看或删除数据卷	349
2.13.5 使用第三方数据卷	350
2.13.6 常见问题	355
2.14 日志管理	355
2.14.1 查看日志	355
2.14.2 集成日志服务	
2.15 监控	362

2.15.1	容器监控服务	
2.15.2	2 查看监控信息	
2.15.3	9 自定义监控	
2.15.4	第三方监控方案集成	
2.15.5	· 容器自动伸缩	
2.15.6	6 节点自动伸缩	
2.15.7	' 监控指标	
2.15.8	9 弹性伸缩不起作用怎么办	
2.17 DevOps	S	
2.17.1	基于 Jenkins 的持续交付	
2.18 服务发现	现和负载均衡	
2.18.1	概述	401
2.18.2	2 简单路由(支持 HTTP/HTTPS)	401
2.18.3	6 简单路由-域名配置	406
2.18.4	简单路由-HTTP 协议变为 HTTPS 协议	
2.18.5	;简单路由-HTTP 强制跳转到 HTTPS	415
2.18.6	;集群内服务间路由和负载均衡	417
2.18.7	'负载均衡路由	
2.18.8	8 容器间的互相发现	427
2.18.9)自定义路由-使用手册	
2.18.1	0 自定义路由-简单示例	444
2.18.1	1 自定义路由-支持 TCP 协议	
2.18.1	2 自定义路由-支持多 HTTPS 证书	
2.19 发布策	略	458
2.19.1	发布策略说明	458
2.19.2	2 简单路由蓝绿发布策略	
2.19.3	9 负载均衡路由蓝绿发布策略	
3 Serverless	Kubernetes 集群	470
3.2 应用管理	<u>1</u>	

1 Kubernetes 集群

1.1 概述

Kubernetes 是流行的开源容器编排技术。为了让用户可以方便地在阿里云上使用 Kubernetes 管理 容器应用,阿里云容器服务提供了 Kubernetes 集群支持。

您可以通过容器服务管理控制台创建一个安全高可用的 Kubernetes 集群,整合阿里云虚拟化、存储、网络和安全能力,提供高性能可伸缩的容器应用管理能力,简化集群的搭建和扩容等工作,让您专注于容器化的应用的开发与管理。

Kubernetes 支持对容器化应用程序的部署、扩展和管理。它具有以下功能:

- 弹性扩展和自我修复
- 服务发现和负载均衡
- 服务发布与回滚
- 机密和配置管理

使用限制

- 目前 Kubernetes 集群只支持 Linux 容器,对 Kubernetes 的 Windows 容器的支持在计划中。
- 目前 Kubernetes 集群只支持 VPC 网络。您可以在部署 Kubernetes 集群时选择创建一个新的 VPC 或者使用已有的 VPC。
- 目前所创建的集群资源(比如:ECS、负载均衡)均为按量付费,您目前可以通过管理控制台将 按量付费实例转换成包年包月实例。未来会提供包年包月支持。

相关开源项目

- 阿里云 Kubernetes Cloud Provider 实现: https://github.com/AliyunContainerService/ kubernetes
- Flannel 的阿里云 VPC 网络驱动: https://github.com/coreos/flannel/blob/master/Documentation /alicloud-vpc-backend.md

如果您对相关项目有问题或者建议,欢迎在社区提交 Issue 或者 Pull Request。

1.2 阿里云 Kubernetes VS 自建 Kubernetes

阿里云 Kubnernetes 的优势

便捷

- 通过 Web 界面一键创建 Kubernetes 集群。
- 通过 Web 界面一键完成 Kubernetes 集群的升级。

您在使用自建 Kubernetes 集群的过程中,可能需要同时处理多个版本的集群(包括 1.8.6、1.9. 4、以及未来的 1.10)。每次升级集群的过程都是一次大的调整和巨大的运维负担。容器服务的 升级方案使用镜像滚动升级以及完整元数据的备份策略,允许您方便地回滚到先前版本。

• 通过 Web 界面轻松地实现 Kubernetes 集群的扩容和缩容。

使用容器服务 Kubernetes 集群可以方便地一键垂直伸缩容来快速应对数据分析业务的峰值。

功能	说明
网络	高性能 VPC 网络插件。支持 network policy 和流控。
	容器服务可以为您提供持续的网络集成和最佳的 网络优化。
负载均衡	支持创建负载均衡实例(公网、内网)。 如果您在使用自建 Kubernetes 集群的过程使用 自建的 Ingress 实现,业务发布频繁可能会造 成 Ingress 的配置压力并增加出错概率。容器服 务的 SLB 方案支持原生的阿里云高可用负载均 衡,可以自动完成网络配置的修改和更新。该方 案经历了大量用户长时间的使用,稳定性和可靠 性大大超过用户自建的入口实现。
存储	集成阿里云云盘、文件存储NAS、块存储EBS ,提供标准的 FlexVolume 驱动。 自建 Kubernetes 集群无法使用云上的存储资 源,阿里云容器服务提供了最佳的无缝集成。
运维	 集成阿里云日志服务、云监控 支持弹性伸缩
镜像仓库	 高可用,支持大并发。 支持镜像加速。 支持 p2p 分发。 您如果使用自建的镜像仓库,在百万级的客户端 同时拉取镜像的时候,会存在镜像仓库崩溃的可

强大

功能	说明
	能性。使用容器服务镜像仓库的专有云版本来提 高镜像仓库的可靠性,减少运维负担和升级压 力。
稳定	 专门的团队保障容器的稳定性。 每个 Linux 版本,每个 Kubernetes 版本都会 在经过严格测试之后之后才会提供给用户。
	容器服务提供了 Docker CE 兜底和推动 Docker 修复的能力。当您遇到 Docker Engine hang、 网络问题、内核兼容等问题时,容器服务可以为 您提供最佳实践。
高可用	 提供多可用区支持。 支持备份和容灾。
技术支持	 提供 Kubernetes 升级能力,新版本一键升级。 阿里云容器团队负责解决在环境中遇到的各种容器问题。

自建 Kubernetes 的成本和风险

• 搭建集群繁琐。

您需要手动配置 kubernetes 相关的各种组件、配置文件、证书、密钥、相关插件和工具,整个 集群搭建工作需要专业人员数天到数周的时间。

- 在公共云上,需要投入大量的成本实现和云产品的集成。
 和阿里云上其他产品的集成,需要您自己投入成本来实现,如日志服务、监控服务和存储管理等。
- 容器是一个系统性工程,涉及网络、存储、操作系统、编排等各种技术,需要专门的人员投入。
- 容器技术一直在不断发展,版本迭代快,需要不断的采坑、升级、测试。

1.3 授权管理

1.3.1 角色授权

在用户开通容器服务时,需要授予名称为 AliyunCSDefaultRole 和 AliyunCSClusterRole 的系统默 认角色给服务账号,当且仅当该角色被正确授予后,容器服务才能正常地调用相关服务(ECS, OSS、NAS、SLB等),创建集群以及保存日志等。

使用说明

- 如果您是在2018年1月15日之前使用过容器服务的用户,系统将默认完成角色授权,详细的授权 权限内容下面的默认角色包含的权限内容。如果您之前是通过子账号使用,您需要对子账号进行 策略升级,参见子账号策略升级。
- 2018年1月15日全面接入跨服务授权后,新用户使用主账号只有进行了跨服务授权才能使用容器
 服务产品。如果新用户需要授权子账号使用容器服务,需要自行前往RAM控制台进行授权,详
 细操作参见使用子账号。

角色授权步骤

当您进入容器服务控制台,如果之前没有正确地给服务账号授予默认角色,则会看到如下提示。
 单击同意授权。

審撮示:如需修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致CS无法获取到必要的权限。 SS请求获取访问您云资源的权限 S方是系统创建的可供CS使用的角色,授权后,CS拥有对您云资源相应的访问权限。 AllyunCSDefaultRole 描述: 容器服务默认使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS)默认角色的授权策略 AllyunCSClusterRole 描述: 容器服务集群使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS) Cluster 角色的授权策略	资源访问授权	
Si请求获取访问您云资源的权限 F方是系统创建的可供CS使用的角色,授权后,CS拥有对您云资源相应的访问权限。 AllyunCSDefaultRole 描述: 容器服务默认使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS)默认角色的授权策略 AllyunCSClusterRole 描述: 容器服务集群使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的授权策略	馨提示:如需修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致CS无法获取到必要的权限。	
Ss请求获取访问您去资源的双限 F方是系统创建的可供CS使用的角色,接仅后,CS拥有对您云资源相应的访问权限。 AllyunCSDefaultRole 描述: 容器服务默认使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS)默认角色的接权策略 AllyunCSClusterRole 描述: 容器服务集群使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的接权策略		
S方是系统创建的可供CS使用的角色,接权后,CS拥有对怨云资源相应的访问权限。 AliyunCSDefaultRole 描述: 容器服务默认使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS)默认角色的接权策略 AliyunCSClusterRole 描述: 容器服务集群使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的接权策略	CS请求获取访问您去资源的权限	
AliyunCSDefaultRole 描述:容器服务默认使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS)默认角色的授权策略 AliyunCSClusterRole 描述:容器服务集群使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS) Cluster 角色的授权策略	下方是系統创建的可供CS使用的角色,授权后,CS拥有对您云资源相应的访问权限。	
AllyunCSDefaultRole 描述: 容器服务默认使用此角色未访问您在其他云产品中的资源 权限描述: 用于容器服务(CS)默认角色的授权策略 AllyunCSClusterRole 描述: 容器服务集群使用此角色未访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的授权策略		
描述:容器服务默认使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS)默认角色的接权策略 AllyunCSClusterRole 描述:容器服务集群使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS) Cluster角色的接权策略	AliyunCSDefaultRole	
权限描述:用于容器服务(CS)默认角色的授权策略 AliyunCSClusterRole 描述:容器服务集群使用此角色未访问您在其他云产品中的资源 权限描述:用于容器服务(CS) Cluster角色的授权策略	描述:容器服务默认使用此角色来访问您在其他云产品中的资源	
AliyunCSClusterRole 描述: 容器服务集群使用此角色未访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的授权策略	权限描述:用于容器服务(CS)默认角色的授权策略	
AllyunCSClusterRole 描述: 容器服务集群使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的接权策略		
描述: 容器服务集群使用此角色来访问您在其他云产品中的资源 权限描述: 用于容器服务(CS) Cluster 角色的授权策略	AliyunCSClusterRole	
权限描述:用于容器服务(CS) Cluster 角色的授权策略	描述:容器服务集群使用此角色来访问您在其他云产品中的资源	
同会抵抗 雨迷	权限描述:用于容器服务(CS) Cluster 角色的授权策略	
同等形积 预送		
同意語知 取当		
EID2* XIZE26H	同意授权 取消	

〕 说明:

容器服务已经设置好默认的角色权限,如需修改角色权限,请前往 RAM 控制台角色管理中设置,需要注意的是,错误的配置可能导致容器服务无法获取到必要的权限。

2. 完成以上授权后,刷新容器服务控制台,然后就可以进行操作了。

如果您想查看 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色的详细策略信息,可以登录 RAM 的控制台 进行查看。

默认角色包含的权限内容

关于各个角色权限的详细信息,请参考各个产品的 API 文档。

AliyunCSDefaultRole 角色的权限内容

默认角色 AliyunCSDefaultRole 包含的主要权限信息如下:

• ECS 相关权限

权限名称(Action)	权限说明
ecs:RunInstances	查询实例信息
ecs:RenewInstance	ECS 实例续费
ecs:Create*	创建 ECS 相关资源,如实例、磁盘等
ecs:AllocatePublicIpAddress	分配公网 IP 地址
ecs:AllocateEipAddress	分配 EIP 地址
ecs:Delete*	删除机器实例
ecs:StartInstance	启动 ECS 相关资源
ecs:StopInstance	停止机器实例
ecs:RebootInstance	重启机器实例
ecs:Describe*	查询 ECS 相关资源
ecs:AuthorizeSecurityGroup	设置安全组入规则
ecs:RevokeSecurityGroup	撤销安全组规则
ecs:AuthorizeSecurityGroupEgress	设置安全组出规则
ecs:AttachDisk	添加磁盘
ecs:DetachDisk	清理磁盘
ecs:AddTags	添加标签
ecs:ReplaceSystemDisk	更换 ECS 实例的系统盘
ecs:ModifyInstanceAttribute	修改实例属性
ecs:JoinSecurityGroup	将实例加入到指定的安全组
ecs:LeaveSecurityGroup	将实例移出指定的安全组
ecs:UnassociateEipAddress	解绑弹性公网 IP
ecs:ReleaseEipAddress	释放弹性公网 IP

• VPC 相关权限

权限名称(Action)	权限说明
vpc:Describe*	查询 VPC 相关资源的信息
vpc:DescribeVpcs	查询 VPC 信息
vpc:AllocateEipAddress	分配 EIP 地址
vpc:AssociateEipAddress	关联 EIP 地址
vpc:UnassociateEipAddress	不关联 EIP 地址
vpc:ReleaseEipAddress	释放弹性公网 IP
vpc:CreateRouteEntry	创建路由接口
vpc:DeleteRouteEntry	删除路由接口

• SLB 相关权限

权限名称(Action)	权限说明
slb:Describe*	查询负载均衡相关信息
slb:CreateLoadBalancer	创建负载均衡实例
slb:DeleteLoadBalancer	删除负载均衡实例
slb:RemoveBackendServers	解绑负载均衡实例
slb:StartLoadBalancerListener	启动指定的监听服务
slb:StopLoadBalancerListener	停止指定的监听服务
slb:CreateLoadBalancerTCPListener	为负载均衡实例创建基于 TCP 协议的监听规则
slb:AddBackendServers	添加后端服务器

AliyunCSClusterRole 角色的权限内容

默认角色 AliyunCSClusterRole 包含的主要权限信息如下:

• OSS 相关权限

权限名称(Action)	权限说明		
oss:PutObject	上传文件或文件夹对象		
oss:GetObject	获取文件或文件夹对象		

权限名称(Action)	权限说明
oss:ListObjects	查询文件列表信息

• NAS 相关权限

权限名称(Action)	权限说明		
nas:Describe*	返回 NAS 相关信息		
nas:CreateAccessRule	创建权限规则		

• SLB 相关权限

权限名称(Action)	权限说明
slb:Describe*	查询负载均衡相关信息
slb:CreateLoadBalancer	创建负载均衡实例
slb:DeleteLoadBalancer	删除负载均衡实例
slb:RemoveBackendServers	解绑负载均衡实例
slb:StartLoadBalancerListener	启动指定的监听服务
slb:StopLoadBalancerListener	停止指定的监听服务
slb:CreateLoadBalancerTCPListener	为负载均衡实例创建基于 TCP 协议的监听规则
slb:AddBackendServers	添加后端服务器
slb:DeleteLoadBalancerListener	删除负载均衡实例监听规则
slb:CreateVServerGroup	创建虚拟服务器组,并添加后端服务器
slb:ModifyVServerGroupBackendServers	改变虚拟服务器组中的后端服务器
slb:CreateLoadBalancerHTTPListener	为负载均衡实例创建基于 HTTP 协议的 Listener
slb:SetBackendServers	配置后端服务器,为负载均衡实例后端的一组服 务器(ECS 实例)配置权重值
slb:AddTags	为 SLB 实例添加标签

1.3.2 使用子账号

使用子账号登录容器服务控制台并静进行相关操作之前,您需要赋予子账号相应的权限。

步骤1创建子账号并开启控制台登录

- 1. 登录访问控制管理控制台。
- 2. 单击左侧导航栏中的用户管理并单击页面右上角的新建用户。
- 3. 填写子账号的名称并单击确定。
- 4. 在用户管理页面,选择创建的子账号,单击右侧的管理。
- 5. 在Web控制台登录管理中,单击启用控制台登录。
- 6. 输入登录密码并单击确定。

步骤 2 授予子账号访问容器服务的权限

1. 在用户管理页面,选择创建的子账号,单击右侧的授权。

用户管理			新建用户 こり	刷新
用户名 🔻 teststs		搜索		
用户名/显示名	备注	创建时间		操作
teststs		2017-12-05 17:04:04	管理 授权 さかかい	删除 加入组

2. 将所需的策略授权给子账号。

编辑个人授权策略				×
添加授权策略后,该账户即具有该条约	策略的权限, 同	同一条授权策略不	能被重复添加。	
可选授权策略名称	类型		已选授权策略名称	类型
CS 管理Elasticsearch的权限	C		AliyunCSFullAccess 管理容器服务(CS)的权限	系统
AliyunCSFullAccess	自定义	>		
AliyunACSResourcesAccess_xingy aliyun container s	自定义	<		
AliyunCSReadOnlyAccess 只读访问容器服务(CS)的权限	系统			
				确定 关闭

您可以使用系统默认的授权策略。

• AliyunCSFullAccess:容器服务的管理权限。

• AliyunCSReadOnlyAccess:容器服务的只读权限。

或者根据您自己的需要自定义授权策略并授权给子账号,参见创建自定义授权策略。

步骤 3 子账号登录容器服务控制台

使用子账号登录 容器服务管理控制台。

如果您之前已经给主账号授予了 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色,子账号可以 直接登录到容器服务管理控制台,并进行相应的操作。

如果之前您没有给主账号授予 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色,则会看到如下 提示。单击同意授权。

资源访问授权	
且著提示:如需修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致CS无法获取到必要的权限。	
S请求获取访问您大资源的权限 下去早冬给创建的可进入S语用的条件, 送初午, CS用去对你一次通用中的注闭和周	
ANY SEARCH & CARLENG & CAR	
AliyunCSDefaultRole	
描述: 容器服务默认使用此角色来访问您在其他云产品中的资源	
权限描述:用于容器服务(CS)默认角色的授权策略	
AligueCCChueterPolo	
描述:容器服务集群使用此角色来访问您在其他云产品中的资源	
权限描述:用于容器服务(CS) Cluster 角色的授权策略	
同意授权 取消	

完成以上授权后,刷新容器服务控制台,然后就可以进行操作了。

1.3.3 创建自定义授权策略

容器服务提供的系统授权策略的授权粒度比较粗,如果这种粗粒度授权策略不能满足您的需要,那 么您可以创建自定义授权策略。比如,您想控制对某个具体的集群的操作权限,您必须使用自定义 授权策略才能满足这种细粒度要求。

创建自定义授权策略

在创建自定义授权策略时,您需要了解授权策略语言的基本结构和语法,相关内容的详细描述请参 考授权策略语言描述。

本文档以授予子账号查询、扩容和删除集群的权限为例进行说明。

操作步骤

1. 使用主账号登录RAM管理控制台。

- 2. 单击左侧导航栏中的策略管理并单击页面右上角的新建授权策略。
- 3. 选择一个模板,填写授权策略名称并编写您的授权策略内容。

创建授权策略				\times
STEP 1:选择权限策略模	版 STEP 2 : 编辑权限并提交		STEP 3:新建成功	
* 授权策略名称:	clusterpolicy 长度为1-128个字符,允许英文字母、数字,或"-"			
备注:				
策略内容:	<pre>1 { 2</pre>		<i>"</i>	
		上一步	新建授权策略	取消
<pre>{ "Statement": [{ "Action": ["cs:Get* "cs:Scale "cs:Delee], "Effect": "A "Resource": "acs:cs:] }], "Version": "1" }</pre>	", eCluster", teCluster" llow", [*:*:cluster/集群ID"			

其中:

• Action 处填写您所要授予的权限。

说明:

所有的 Action 均支持通配符。

- Resource 有如下配置方式。
 - 授予单集群权限

```
"Resource": [
    "acs:cs:*:*:cluster/集群ID"
]
```

- 授予多个集群权限

```
"Resource": [
    "acs:cs:*:*:cluster/集群ID",
    "acs:cs:*:*:cluster/集群ID"
]
```

- 授予您所有集群的权限

```
"Resource": [
     " * "
]
```

其中,集群ID 需要替换为您要授权的真实的集群 ID。

4. 编写完毕后,单击新建授权策略。

Action 沿胆	
0000	
CreateCluster 创建集群	
AttachInstances	有ECS实例
ScaleCluster 扩容集群	
GetClusters查看集群列表	
GetClusterByld 查看集群详情	
ModifyClusterName 修改集群名称	
DeleteCluster 删除集群	
UpgradeClusterAgent 升级集群Agent	
GetClusterLogs 查看集群的操作	目志
GetClusterEndpoint 查看集群接入点:	也址
GetClusterCerts 下载集群证书	
RevokeClusterCerts 吊销集群证书	

₹

Action	说明
BindSLB	为集群绑定负载均衡实例
UnBindSLB	为集群解绑负载均衡实例
ReBindSecurityGroup	为集群重新绑定安全组
CheckSecurityGroup	检测集群现有的安全组规则
FixSecurityGroup	修复集群的安全组规则
ResetClusterNode	重置集群中的节点
DeleteClusterNode	移除集群中的节点
CreateAutoScale	创建节点弹性伸缩规则
UpdateAutoScale	更新节点弹性伸缩规则
DeleteAutoScale	删除节点弹性伸缩规则
GetClusterProjects	查看集群下的应用
CreateTriggerHook	为应用创建触发器
GetTriggerHook	查看应用的触发器列表
RevokeTriggerHook	删除应用的触发器
CreateClusterToken	创建 Token

1.4 子账号Kubernetes应用权限配置指导

在通过集群访问凭证ConfigKube时,阿里云Kubernetes服务支持字账号Kubernetes应用权限配置,实现子账号应用操作权限的细粒度控制。

前提条件

- 您已有一个阿里云主账号,并拥有一个或若干个子账号,并获取子账号的UID。
- 您已开启容器服务,并创建了一个Kubernetes集群,参见创建Kubernetes集群
- 您已成功通过Kubectl连接到Kubernetes集群,参见通过 kubectl 连接 Kubernetes 集群。
- 你已成功在集群中部署一个Pod,本例中创建一个名为hello-world的示例Pod。

获取子账号KubeConfig

道 说明:

您需要以子账号登录容器服务管理控制台,获取集群访问凭证KubeConfig。

- 1. 登录容器服务管理控制台。
- 2. 在Kubernetes菜单下,单击左侧导航栏中的集群,进行集群列表页面。
- 3. 选择所需的集群,单击右侧的管理,进入集群管理页面,复制KubeConfig的内容。
- 4. 进入到本机Linux环境中,将KubeConfig内容复制到本机 \$HOME/.kube/config文件中。

容器服务 - Kubernetes 、	子账号授权 🔩 返回列表						
概定	选择子账号		授权资	原	>	更新授权策略	
	用户名	显示名	香注	创建时间			
集群	projekte			2018-05-16 11:32:11			授权
节点	ing a			2018-05-16 11:33:19			授权
存储卷	+			2018-07-12 16:31:03			授权
命名空间	ingening .			2018-07-18 10:21:39			授权 3
命名空间 子脉号授权 2	at Anggaing	-		2018-07-12 16:31:03 2018-07-18 10:21:39			授权 授权 3

5. 执行kubectl cluster-info命令,查看集群状态,确认集群连接正常。

```
$ kubectl cluster-info
Kubernetes master is running at https://118.178.30.xxx:6443
Heapster is running at https://118.178.30.xxx:6443/api/v1/namespaces
/kube-system/services/heapster/proxy
KubeDNS is running at https://118.178.30.xxx:6443/api/v1/namespaces/
kube-system/services/kube-dns:dns/proxy
monitoring-influxdb is running at https://118.178.30.xxx:6443/api/v1
/namespaces/kube-system/services/monitoring-influxdb/proxy
```

集群访问角色配置模板

当前容器服务创建的集群中并没有预置定制化的集群角色,主账号管理员需要登录到部署控制台中 使用模板创建的方式,或者登录到任意Master节点上通过kubect1命令手工部署需要的角色模型。



若通过kubectl连接集群的方式部署角色模型,您需要以主账号的KubeConfig作为登录凭证,否则 无法创建集群访问角色,您将需要复制两次KubeConfig内容。建议通过容器服务控制台或SSH连 接到集群的方式进行部署。

角色模型的配置方式可参考Kubernetes官方的文档https://kubernetes.io/docs/reference/accessauthn-authz/rbac/- role-and-clusterrole,根据用户的使用习惯,结合容器服务自身形态,容器服务 向您推荐以下几个集群角色(ClusterRole)模板,相关的示例编排如下:

管理员

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
name: cs:admin
rules:
```

```
- apiGroups: [""]
resources: ["pods", "pods/attach", "pods/exec", "pods/portforward", "
pods/proxy"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: [""]
resources: ["configmaps", "endpoints", "persistentvolumeclaims", "
replicationcontrollers", "replicationcontrollers/scale", "secrets", "
serviceaccounts", "services", "services/proxy"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: [""]
resources: ["bindings", "events", "limitranges", "namespaces/status
", "replicationcontrollers/status", "pods/log", "pods/status", "
resourcequotas", "resourcequotas/status"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["namespaces", "nodes", "persistentvolumes"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["apps"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "replicasets", "replicasets/scale", "statefulsets
" ]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["autoscaling"]
resources: ["horizontalpodautoscalers"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["batch"]
resources: ["cronjobs", "jobs"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["extensions"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "ingresses", "replicasets", "replicasets/scale", "
replicationcontrollers/scale"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
```

运维人员

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
name: cs:opr
rules:
- apiGroups: [""]
resources: ["pods", "pods/attach", "pods/exec", "pods/portforward", "
pods/proxy"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: [""]
resources: ["configmaps", "endpoints", "persistentvolumeclaims", "
replicationcontrollers", "replicationcontrollers/scale", "secrets", "
serviceaccounts", "services", "services/proxy"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: [""]
```

```
resources: ["bindings", "events", "limitranges", "namespaces/status
", "replicationcontrollers/status", "pods/log", "pods/status", "
resourcequotas", "resourcequotas/status"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["namespaces"]
verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "replicasets", "replicasets/scale", "statefulsets
"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
', "update", "watch"]
- apiGroups: ["autoscaling"]
resources: ["horizontalpodautoscalers"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["batch"]
resources: ["cronjobs", "jobs"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["extensions"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "ingresses", "replicasets", "replicasets/scale", "
replicationcontrollers/scale"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
```

开发人员

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
name: cs:dev
rules:
- apiGroups: [""]
resources: ["pods", "pods/attach", "pods/exec", "pods/portforward", "
pods/proxy"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: [""]
resources: ["configmaps", "endpoints", "persistentvolumeclaims", "
replicationcontrollers", "replicationcontrollers/scale", "secrets", "
serviceaccounts", "services", "services/proxy"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: [""]
resources: ["events", "namespaces/status", "replicationcontrollers/
status", "pods/log", "pods/status"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["namespaces"]
verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "replicasets", "replicasets/scale", "statefulsets
"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["autoscaling"]
```

```
resources: ["horizontalpodautoscalers"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["batch"]
resources: ["cronjobs", "jobs"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
- apiGroups: ["extensions"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "ingresses", "replicasets", "replicasets/scale", "
replicationcontrollers/scale"]
verbs: ["create", "delete", "deletecollection", "get", "list", "patch
", "update", "watch"]
```

受限人员

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
name: cs:restricted
rules:
- apiGroups: [""]
resources: ["pods", "pods/attach", "pods/exec", "pods/portforward", "
pods/proxy"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["configmaps", "endpoints", "persistentvolumeclaims", "
replicationcontrollers", "replicationcontrollers/scale", "secrets", "
serviceaccounts", "services", "services/proxy"]
verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["events", "replicationcontrollers/status", "pods/log", "
pods/status"]
verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "replicasets", "replicasets/scale", "statefulsets
" ]
verbs: ["get", "list", "watch"]
- apiGroups: ["autoscaling"]
resources: ["horizontalpodautoscalers"]
verbs: ["get", "list", "watch"]
- apiGroups: ["batch"]
resources: ["cronjobs", "jobs"]
verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
resources: ["daemonsets", "deployments", "deployments/rollback", "
deployments/scale", "ingresses", "replicasets", "replicasets/scale",
replicationcontrollers/scale"]
verbs: ["get", "list", "watch"]
```

配置子账号对应角色和角色绑定

当前控制台尚不支持子账号权限的自动化配置,同样需要主账号管理员在目标集群的部署控制台中使用模板创建子账号对应的角色绑定(按需可以是集群维度的ClusterRoleBinding或

是namespace维度的RoleBinding)。您也可登录到任意master节点上通过**kubect1**命令手工添加,

说明:

角色绑定中的subject kind为User, name是子账号对应的UID。

以受限人员的角色模板为例,演示如何为子账号配置一个受限人员的角色,并进行绑定,让该子账 号不能进行权限外的操作。

1. SSH登录到Master节点,执行vim restricted-clusterrole.yaml命令,创建

restricted-clusterrole.yaml文件。

在该文件中输入如下yaml编排。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
 name: cs:restricted
rules:
- apiGroups: [""]
 resources: ["pods", "pods/attach", "pods/exec", "pods/portforward
                          ##不允许删除Podroxy"]
", "pods/p
 verbs: ["get", "list", "watch"]
- apiGroups: [""]
resources: ["configmaps", "endpoints", "persistentvolumeclaims",
"replicationcontrollers", "replicationcontrollers/scale", "secrets
", "serviceaccounts", "services", "services/proxy"]
 verbs: ["get", "list", "watch"]
- apiGroups: [""]
 resources: ["events", "replicationcontrollers/status", "pods/log
", "pods/status"]
 verbs: ["get", "list", "watch"]
- apiGroups: ["apps"]
              ["daemonsets", "deployments", "deployments/rollback
 resources:
", "deployments/scale", "replicasets", "replicasets/scale", "
statefulsets"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["autoscaling"]
               ["horizontalpodautoscalers"]
  resources:
  verbs: ["get", "list", "watch"]
- apiGroups: ["batch"]
 resources: ["cronjobs", "jobs"]
  verbs: ["get", "list", "watch"]
- apiGroups: ["extensions"]
 resources:
             ["daemonsets", "deployments", "deployments/rollback",
 "deployments/scale", "ingresses", "replicasets", "replicasets/scale",
 "replicationcontrollers/scale"]
  verbs: ["get", "list", "watch"]
```

执行kubectl create -f restricted-clusterrole.yaml 命令,创建该ClusterRole。

2. 执行vim restricted-clusterrole-binding.yaml命令,并在该文件中输入如下yaml模

板。

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
name: read-pod-global
subjects:
- kind: User ##类型为User
name: "204299326441xxxx" ##您的子账号用户UID
apiGroup: rbac.authorization.k8s.io
roleRef:
kind: ClusterRole
name: cs:restricted
apiGroup: rbac.authorization.k8s.io
```

执行kubectl create -f restricted-clusterrole-binding.yaml 命令,创建

该ClusterRoleBinding。

3. ClusterRole和ClusterRolebinding创建完毕后,返回通过子账号ConfigKube凭证连接集群的本机

环境。尝试通过kubectl delete命令删除一个Pod。

说明:

本例中,受限人员的ClusterRole模板不允许被绑定该角色的子账号删除Pod。

\$ kubectl get pods NAME READY STATUS RESTARTS AGE hello-pod 1/1 Running 0 16s

\$ kubectl delete pod hello-pod Error from server (Forbidden): pods "hello-pod" is forbidden: User " 204299326441xxxxxx" cannot delete pods in the namespace "default"

您可以看到,以子账号登录到集群时,无法对未授权的角色进行操作。

后续增强

后续容器服务会在创建的Kubernetes集群中添加预置的集群角色,并在控制台增加权限管理页 面,提供针对子账号的集群或namespace维度的角色绑定;同时提供细粒度的面向Kubernetes模型 级别的子账号应用权限管理功能。

1.5 集群管理

1.5.1 创建 Kubernetes 集群

您可以通过容器服务管理控制台非常方便地快速创建 Kubernetes 集群。

使用须知

创建集群过程中,容器服务会进行如下操作:

- 创建 ECS,配置管理节点到其他节点的 SSH 的公钥登录,通过 CloudInit 安装配置 Kubernetes 集群。
- 创建安全组,该安全组允许 VPC 入方向全部 ICMP 端口的访问。
- 如果您不使用已有的 VPC 网络,会为您创建一个新的 VPC 及 VSwitch,同时为该 VSwitch 创 建 SNAT。
- 创建 VPC 路由规则。
- 创建 NAT 网关及 EIP。
- 创建 RAM 子账号和 AK,该子账号拥有 ECS 的查询、实例创建和删除的权限,添加和删除云盘的权限,SLB 的全部权限,云监控的全部权限,VPC 的全部权限,日志服务的全部权限,NAS 的全部权限。Kubernetes 集群会根据用户部署的配置相应的动态创建 SLB,云盘,VPC路由规则。
- 创建内网 SLB,暴露 6443 端口。
- 创建公网 SLB,暴露 6443、8443和 22 端口(如果您在创建集群的时候选择开放公网 SSH 登录,则会暴露 22 端口;如果您选择不开放公网 SSH 访问,则不会暴露 22 端口)。

前提条件

您需要开通容器服务、资源编排(ROS)服务和访问控制(RAM)服务。

登录 容器服务管理控制台、ROS 管理控制台 和 RAM 管理控制台 开通相应的服务。

▋ 说明:

容器服务 Kubernetes 集群部署依赖阿里云资源编排 ROS 的应用部署能力,所以创建 Kubernetes 集群前,您需要开通 ROS。

使用限制

- 随集群一同创建的负载均衡实例只支持按量付费的方式。
- Kubernetes 集群仅支持专有网络 VPC。

- 每个账号默认可以创建的云资源有一定的配额,如果超过配额创建集群会失败。请在创建集群前确认您的配额。如果您需要提高您的配额,请提交工单申请。
 - 每个账号默认最多可以创建 5 个集群(所有地域下),每个集群中最多可以添加 20 个
 Worker 节点。如果您需要创建更多的集群或者节点,请提交工单申请。
 - 每个账号默认最多可以创建 100 个安全组。
 - 每个账号默认最多可以创建 60 个按量付费的负载均衡实例。
 - 每个账号默认最多可以创建 20 个EIP。
- ECS 实例使用限制:
 - 一 仅支持 CentOS 操作系统。
 - 仅支持创建按量付费的 ECS 实例。

送 说明:

实例创建后,您可以通过 ECS 管理控制台将按量付费转包年包月。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏的集群,进入集群列表页面。
- 3. 单击页面右上角的创建 Kubernetes 集群。

容器服务 - Kubernetes ▼	集群列表		您最多可以创建 5 个集副	羊,每个集群最多可以活	5加 40 个节点 刷新	创建Serverless Kubernetes集群	创建Kubernetes 集群 🚽
概览	常见问题: 8 如何创新	皇集群 🔗 扩容和缩容集群 🔗 通过 kubect :	连接 Kubernetes 集群 🛛 ⑧ 通过命令管理	应用			2
▼ 集群	名称 ▼						
集群	集群名称/ID	集群关型	地域 (全部) 👻 网络类型	集群状态	创建时间	Kubernetes 版本	操作

4. 填写集群的名称。

集群名称应包含1-63个字符,可包含数字、汉字、英文字符或连字符(-)。

5. 选择集群所在的地域和可用区。

地域:	华北 1	华北 2	华东 1	华东 2	华南 1	亚太东北 1 (东京) 美国西部 1 (硅谷)
			亚太东南1(新加	美国东部1(弗吉	欧洲中部1(法兰	
	华北 3	香港	坡)	尼亚)	克福)	
可用区:	华东 1 可用区 F	-				

- 6. 设置集群的网络。
 - 网络类型:Kubernetes 集群仅支持专有网络。

- 专有网络:您可以选择自动创建(创建 Kubernetes 集群时,同步创建一个 VPC)或者使用已有(使用一个已有的 VPC)。选择使用已有后,您可以在已有 VPC 列表中选择所需的 VPC 和交换机。
 - 选择自动创建,创建集群时,系统会自动为您的 VPC 创建一个 NAT 网关。
 - 选择使用已有,如果您使用的 VPC 中当前已有 NAT 网关,容器服务会使用已有的 NAT 网关;如果 VPC 中没有 NAT 网关,系统会默认自动为您创建一个 NAT 网关。如果您 不希望系统自动创建 NAT 网关,可以取消勾选页面下方的在您选择的 VPC 内自动创建 NAT 网关。

📕 说明:

若选择不自动创建 NAT 网关,您需要自行配置 NAT 网关实现 VPC 安全访问公网环境,或者手动配置 SNAT,否则 VPC 内实例将不能正常访问公网,会导致集群创建失败。

您需要指定Pod 网络 CIDR和Service CIDR,两者都不能与 VPC 及 VPC 内已有 Kubernetes 集群使用的网段重复,创建成功后不能修改。而且 Service 地址段也不能和 Pod 地址段重复,有关 kubernetes 网络地址段规划的信息,请参考*VPC*下 *Kubernetes* 的 网络地址段规划。

7. 配置节点信息。

- 是否新增节点:目前,创建 Kubernetes 集群时,默认会创建 3 个 Master 节点(暂时不支持 用户设置 Master 节点的个数)和指定数量的 Worker 节点。您也可以在集群创建后,向集群 中添加已有节点,具体操作参见添加已有节点。
- 节点类型:目前,Kubernetes 集群仅支持创建按量付费的节点。
- 登录密码:设置节点的登录密码。
- 确认密码:确认设置的节点登录密码。
- 8. 设置 Master 节点的配置信息。

您需要选择 Master 节点的系列和规格。

ഘ		
	说明	:

- 目前仅支持 CentOS 操作系统。
- 目前仅支持创建 3 个 Master 节点。

MASTER 配置		
实例规格	4核8G(ecs.sn1ne.xlarge)	▼ 数量 3台
系统盘	SSD云盘 ▼	40 GiB 🜲

9. 设置 Worker 节点的配置信息。您可选择新增实例或添加已有实例。

6	〕 说明:
•	目前仅支持 CentOS 操作系统。
•	每个集群最多可包含 20 个 Worker 节点。如果您需要创建更多的节点,请提交工单申请。
•	每个集群最多可包含 20 个 Worker 节点。如果您需要创建更多的节点,请提交工单申请。

a. 若您选择新增实例,则需要选择 Worker 节点的系列和规格,以及需要创建的 Worker 节点的数量(本示例创建一个 Worker 节点)。

WORKER 配置										
实例规格	4核8G(ecs.sn1ne.xlarge)				• 7	数量	1	台	*	
系统盘	SSD云盘	•	40	GiB	*					
✔ 挂载数据盘	SSD云盘	-	40	GiB	*					

b. 若您选择添加已有实例,则需要预先在此地域下创建 ECS 云服务器。

Worker 实例	新增实例	添加已有实例							
	您目前可以通过 ECS 曾 如需使用包年包月实例	您目前可以通过 ECS 管理控制台格按量付费实例转换成包年包月实例。查 看详情 如需使用包年包月实例,请在ECS控制台购买后添加到集群。							

10.设置是否在所选的 VPC 内自动创建 NAT 网关。

该选项仅在您选择 使用已有 VPC 时生效。

11.设置是否开放公网 SSH 登录。

- 选择开放公网 SSH 登录,您可以 SSH 访问集群。
- 选择不开放公网 SSH 登录,将无法通过 SSH 访问集群,也无法通过 kubectl 连接 集群。如果您需要通过 SSH 访问集群实例,可以手动为 ECS 实例绑定 EIP,并配置安全组规则,开放 SSH(22)端口,具体操作参见 SSH 访问 Kubernetes 集群。

SSH登录	✔ 开放公网SSH登录
	选择不开放时,如需手动开启 SSH 访问,请参考 SSH 访问 Kubernetes 集群

12.设置是否启用云监控插件。

您可以选择在 ECS 节点上安装云监控插件,从而在云监控控制台查看所创建 ECS 实例的监控 信息。

云监控插件: 🔽 在ECS节点上安装云监控插件 在节点上安装云监控插件,可以在云监控控制台查看所创建ECS实例的监控信息

13.设置是否将节点 IP 添加到 RDS 实例的白名单。

您可以选择将所创建节点的 IP 添加到 RDS 实例的白名单中,方便 ECS 实例访问 RDS 实例。



您仅能将 ECS 实例的 IP 添加到位于同一地域同一 VPC 网络下的 RDS 实例的白名单中。

a. 单击请选择您想要添加白名单的RDS实例。

b. 在弹出的对话框中选择所需的 RDS 实例并单击确定。

14.是否启用高级选项。

a. 设置节点 Pod 数量,是指单个节点可运行 Pod 数量的上限,默认为 256,建议保持默认值。

节点Pod数量	128	-

- b. 设置是否选择自定义镜像。或不选择自定义镜像,则 ECS 实例会安装默认的 CentOS 版本。 目前您只能选择基于 CentOS 的镜像来快速部署您需要的环境,如基于 CentOS 7.2 的 LAMP 部署测试的镜像。
- **c.** 设置是否使用自定义集群**CA**。如果勾选自定义集群 CA,可以将 CA 证书添加到 kubernetes 集群中,加强服务端和客户端之间信息交互的安全性。

集群CA:	自定义集群CA			

15.单击创建集群,启动部署。



一个包含多节点的 Kubernetes 集群的创建时间一般需要十几分钟。

查看集群部署结果

集群创建成功后,您可以在容器服务管理控制台的 Kubernetes 集群列表页面查看所创建的集群。

容器服务 - Kubernetes ▼	集群列表	您最多可以创建 5 个集群,	每个集群最多可以添加 40 个节点	刷新 创建Se	erverless Kubernetes集群	创建 Kubernetes 集群	-
概览	常见问题: ②如何创建集群 ③ 扩容和缩容集群	i 🔗 通过 kubectl 连接 Kubern	etes 集群 🔗 通过命令管理应用				
* 集群	名称 ▼						
集群	集群名称/ID 集群类	地域 (全部) 型 ▼	网络类型 集群状态	创建时间	Kubernetes 版 本	2	操作
存储卷	k8s-cluster Kubert	etes 华东1	虚拟专有网络 vpc-	2018-06-27 17:48:29	1.9.7	理 <u> 查看日志</u> : 集群伸缩	控制台 更多▼

您可以单击右侧的查看日志查看集群的日志信息。

集群日志信息: k8s-cluste	「 な 返回律 類列表	刷新
资源部署详细日志请参考:	1度线条件	
时间	位思	
2018-07-09 17:19:01	c07ea7a5639924917a/756e7d3195e0d5 Start to delete node	
2018-07-09 17:19:01	c07ea7a5639924917af756e7d3195e0d5 Failed to delete node (i-bp1dmw3cynna4bsf3rfa) Api server error.	
2018-07-09 17:18:59	c07ea7a5639924917af756e7d3195e0d5 Start to delete mode	
2018-07-09 17:18:59	c07ea7a5639924917af756e7d3195e0d5 Failed to delete node (i-bp1dmw3cynna4bsf3rfb) Api server error.	
2018-07-09 17:18:40	c07ea7a5639924917af756e7d3195e0d5 Start to delete mode	
2018-07-09 17:18:40	c07ea7a5639924917af756e7d3195e0d5 Failed to delete node (i-bp1dmw3cynna4bsf3rfa) Api server error.	
2018-07-09 17:18:33	c07ea7a5639924917af756e7d3195e0d5 Start to delete node	
2018-07-09 17:18:33	c07ea7a5639924917af756e7d3195e0d5 Failed to delete node (i-bp1dmw3cynna+tbsf3rtb) Api server error.	

您可以单击右侧的管理,查看集群的基本信息和连接信息。

基本信息			
集群ID:	虚拟专有网络	●运行中	地域: 华东1
连接信息			
API Server 公网连接端点	https://1 6443		
API Server 内网连接端点	https://1 6443		
Master 节点 SSH 连接地址	10.0000		
服务访问域名),(cn-hangzhou.alicontainer.com	
集群资源			
资源编排 ROS	k8s-for-cs-	891	
公网 SLB	lb-bp		
虚拟专有网络 VPC	vpc-		
NAT 网关	ngw		
通过 kubectl 连接 Kubernetes 集群 从 Kubernetes 版本页面 下载最新的 kubectl 客户端。 安装和设置 kubectl 客户端。有关详细信息,参见 安装和设置 kubectl 客户端。 配置集群凭据: KubeConfig 	ibectl		

其中:

- API Server 公网连接端点:Kubernetes 的 API server 对公网提供服务的地址和端口,可以通过 此服务在用户终端使用 kubectl 等工具管理集群。
- API Server 内网连接端点:Kubernetes 的 API server 对集群内部提供服务的地址和端口。此 IP 为负载均衡的地址,后端有3合 Master 提供服务。
- Master 节点 SSH 连接地址:可以直接通过 SSH 登录到 Master 节点,以便对集群进行日常维护。
- 服务访问域名:为集群中的服务提供测试用的访问域名。服务访问域名后缀是<cluster_id
 >.<region_id>.alicontainer.com。

例如,您可以通过 SSH 登录到 Master 节点查看集群的节点信息。

```
login as: root
root@1 _____B's password:
Welcome to Alibaba Cloud Elastic Compute Service !
[root@iZbp1d7yvpa3j183u0urllZ ~]# kubectl get node
NAME
                                              ROLES
                                                                 VERSION
                                    STATUS
                                                       AGE
cn-hangzhou.i
                                    Ready
                                                       17m
                                                                 v1.8.4
                                              <none>
cn-hangzhou.i
                                    Ready
                                                       19m
                                                                 v1.8.4
                                             master
                   24m
                                                                 v1.8.4
cn-hangzhou.i-
                                    Ready
                                             master
cn-hangzhou.i
                                    Ready
                                                       22m
                                                                 v1.8.4
                                              master
[root@iZbp1d7yvpa3j183u0url1Z ~]#
```

可以发现,一共有4个节点,包括3个 Master 节点和我们在参数设置步骤填写的1个 Worker 节点。

1.5.2 SSH 访问 Kubernetes 集群

如果您在创建集群时,选择不开放公网 SSH 访问,您将无法 SSH 访问 Kubernetes 集群,而且也 无法通过 kubectl 连接 Kubernetes 集群。如果创建集群后,您需要 SSH 访问您的集群,可以手动 为 ECS 实例绑定 EIP,并配置安全组规则,开放 SSH (22)端口。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 登录 容器服务管理控制台
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 4. 选择所需的集群并单击右侧的管理。
- 在集群资源中,找到您的公网负载均衡实例,并单击实例 ID,页面跳转至您的公网负载均衡实例的详细信息页面。

集群:kastest						
基本信息						
集群ID: c77862d73x4a4+674b44d:571+04011	虚拟专有网络	●运行中	地域:华东1	节点个数 4	集群伸缩	
连接信息						
API Server 公网连接端点	https://10					
API Server 内网连接端点	https://19					
集群迎源						
公网 SLB	lb-bp16					
虚拟专有网络 VPC	vpc-bp)					
读集群未开通公网 SSH 访问,如需访问,请点击以下任意ECS实例绑定EIP:i-bp16nbm51uyl36j595km i-bp11qftgryaa9resdm1 i-bp136jwigspx7nyoe8kt						

- 6. 单击左侧导航栏中的监听并单击页面右上角的添加监听。
- 7. 添加 SSH 监听规则。
 - a. 前端协议(端口)选择 TCP:22。
 - **b.** 后端协议(端口)选择 TCP:22。
 - C. 选择 使用服务器组 并选择虚拟服务器组。
 - d. 服务器组ID 选择sshVirtualGroup。
 - e. 单击下一步并单击确认, 创建监听。
| 添加监听 | | | | | \times |
|-----------------|-----------------------------|--------------------------------------|----------|-------------|----------|
| 1.基本配置 | | 2.健康检查配置 | > | 3.配置成功 | |
| 前端协议[端口]:🔺 | TCP ▼
端口輸入范围)
四层监听请选择 | : 22
均1-65535。
译TCP、UDP ; 七层监听 | 请选择HTTP、 | HTTPS; 查看详情 | |
| 后端协议[端口]: 🛊 | TCP
端口输入范围) | : 22
均1-65535。 | | | |
| 带宽峰值: | 不限制 配置
使用流量计费2 | 方式的实例默认不限制帮 | 带宽峰值;峰值 | 输入范围1-5000 | |
| 调度算法: | 加权轮询 | v | | | |
| 使用服务器组: 🖉 | | 七么是服务器组? | | | |
| 服务器组类型: | ◉ 虚拟服务器 | 組 🛛 🔍 主备服务 | 器组 🛛 | | |
| 服务器组ID: | sshVirtualGro | oup 🔻 | | | |
| 创建完毕自动启动监
听: | — ея | F启 | | | |
| ▼ 展开高级配置 | | | | | |
| | | | | 下一步 | 取消 |

8. 监听创建完成后,您就可以使用该负载均衡的服务地址 SSH 访问您的集群了。

K8sMasterSlbIntern ・ 返回负载均衡列表	
基本信息	^
负载均衡ID: lb-bp16j	状态: ● 运行中
负载均衡名称: K8sMa	地域: 华东 1
地址类型: 公网	可用区: 华东 1 可用区 B(主)/华东 1 可用区 D(箭)
网络类型: 经典网络	
付费信息	消费明细 释放投資 ヘ
付费方式: 按使用流量	创建时间: 2018-01-04 19:47:11
服务地址: 101	自动释放时间: 无

1.5.3 SSH密钥对访问Kubernetes集群

阿里云容器服务支持通过 SSH 密钥对的方式登录集群,保障远程SSH访问的安全性。

背景信息

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 单击右上角的创建 Kubernetes 集群。

容器服务		集群列表	約果	多可以创建 5 个集群,	每个集群最多可以添加 40 ·	个节点 刷新	由请使用Serverless Kubernetes集群	创建Kubernetes 集群 🚽	
Kubernetes	Swarm								_
101 US		常见问题: ⑧ 如何创建集群 ⑧ 扩制	斜縮容集群 🔗 通过 kubectl i	転接 Kubernetes 集群	♂ 通过命令管理应用			4	
▼ 集群	•	名称 ▼							
集群		集群名称/ID	集群类型 地域 (全	部) 👻 网络类型	集群状态	创建时间	Kubernetes 版本	握作	Έ
节点									

- 对集群登录方式进行配置,设置通过密钥的登录方式。对集群其他的参数进行配置,参见创 建Kubernetes集群,最后单击创建。
 - 1. 若您已在ECS云服务器控制台创建了密钥,只需在密钥对下拉框中进行选择。
 - 若您没有密钥,单击新建密钥对,前往ECS云服务器控制台创建密钥对,参见创建 SSH 密钥 对。

登录方式	设置密钥	设置密码		
密钥对	test11 您可以访问 ECS 控制台	新建密钥对	• C	

5. 集群创建成功后,在集群列表中找到该集群,单击集群右侧的管理,在连接信息中查 看Master节点SSH连接地址。

<	集群:test-k8s
基本信息	基本信息
	集群ID: c5
	_
	连接信息
	API Server 公网连接端点
	API Server 内网连接端点
	Master 节点 SSH 连接地址
	服务访问域名

- 下载.pem私钥文件,按照本地操作系统环境,如 Windows 和 linux 操作系统,进行相关配置,参见使用SSH密钥对连接Linux实例。以 linux 环境为例。
 - a) 找到您所下载的.pem私钥文件在本地机上的存储路径,如/root/xxx.pem
 - b) 运行命令修改私钥文件的属性: chmod 400 [.pem私钥文件在本地机上的存储路径], 如 chmod 400 /root/xxx.pem。
 - c) 运行命令连接至集群:`ssh -i [.pem私钥文件在本地机上的存储路径] root@[masterpublic-ip],其中 master-public-ip 即是 Master 节点 SSH 连接地址。如 ssh -i /root /xxx.pem root@10.10.10.100。

1.5.4 通过 kubectl 连接 Kubernetes 集群

如果您需要从客户端计算机连接到 Kubernetes 集群,请使用 Kubernetes 命令行客户端 kubectl。

操作步骤

- 1. 从Kubernetes 版本页面 下载最新的 kubectl 客户端。
- 2. 安装和设置 kubectl 客户端。

有关详细信息,参见安装和设置 kubectl。

3. 配置集群凭据。

您可以使用 scp 命令安全地将主节点的配置从 Kubernetes 集群主 VM 中的 /etc/

```
kubernetes/kube.conf 复制到本地计算机的 $HOME/.kube/config (kubectl 预期凭据所
```

在的位置)。例如:

```
mkdir $HOME/.kube
scp root@<master-public-ip>:/etc/kubernetes/kube.conf $HOME/.kube/
config
```

您可以在集群信息页面查看集群的 master-public-ip。

- a) 登录容器服务管理控制台。
- b) 单击Kubernetes进入 Kubernetes 集群列表页面。
- c) 选择所需的集群并单击右侧的管理

您可以在连接信息处查看集群的连接地址。

集群 : k8s-test					
基本信息					
集群ID: cde6	虚拟专有网络	●运行中	地域: 华东1	节点个数 4	集群伸缩
连接信息					
Master 节点 SSH 连接地址	101.37				
API Server 公网连接端点	https://				
API Server 内网连接端点	https://				

1.5.5 添加已有节点

您可以向已经创建的 Kubernetes 集群中添加已有的 ECS 实例。目前, 仅支持添加 Worker 节点。

前提条件

- 如果之前没有创建过集群,您需要先创建 Kubernetes 集群。
- 需要先把待添加的 ECS 实例添加到 Kubernetes 集群的安全组里。

背景信息

使用说明

- 默认情况下,每个集群中最多可包含 20 个 Worker 节点。如果您需要添加更多节点,请提交工 单申请。
- 添加的云服务器必须与集群在同一地域同一 VPC 下。

- 添加已有云服务器时,请确保您的云服务器有EIP(专有网络),或者相应 VPC 已经配置了
 NAT网关。总之,需要确保相应节点能正常访问公网,否则,添加云服务器会失败。
- 容器服务不支持添加不同账号下的云服务器。
- 仅支持添加操作系统为 CentOS 的节点。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在Kubernetes菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的更多 > 添加已有节点。

集群列表					您最多可以创建 5 个集群,	每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes 集群
常见问题: ♂如何创建集群 ♂ 扩容和熔容集群 名称 ▼	& 通过 kubect	连接 Kube	rrnetes 集群 🔗 通过命令管理应用					
集群名称/ID	集群类型	地域	网络类型	集群状态	创建时间	Kubernetes 版本		操作
k8stest c77862d75bda644679b48d15f1a42e831	Kubernetes	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	2018-01-04 19:47:08	1.8.4	管理	查看日志 制除 控制台 更多▼
k8s c02879fa8ea2c47d48b6c4ce76524499f	Kubernetes	华东1	虚拟专有网络	●失败 原因:	2018-01-04 19:38:32	1.8.4	管理	集群伸缩 添加已有节点

4. 进入添加节点页面,您可以选择自动添加或手动添加的方式,添加现有云服务器实例。

自动添加方式会列出当前账号下可用的 ECS 云服务器,在 Web 界面进行选择,安装部署,并 自动添加到集群中;手动添加方式要求您获取安装命令,登录到对应 ECS 云服务器上进行安 装,每次只能添加一个 ECS 云服务器。

- 5. 您可选择自动添加的方式,您可以一次性添加多个ECS云服务器。
 - a)在已有云服务器的列表中,选择所需的ECS云服务器,然后单击下一步。

选择	已有云服务器实例	填写实例信息	>		添加完成
添加方式:					
◎ 自动添加 🛛 手动添加					
容器服务添加已有ECS实例,会到	更换系统盘,磁盘ID会变更,原系统盘会被释放。				
请注意: 1. 您原系统盘的用户快照会保 2. 为保持足够的快照额度来顺 3. 您在操作前做好相关备份,	留,自动快照则根据您该系统盘属性值"自动块照是3 利克成周期性的自动技服策略,您可以删除不需要的 以免数据丢失给你這成损失。	雪随磁盘释放"的选项来判断是否保留或随磁盘 用户快照和自动快照。	删除;您可以进入磁盘列表点	击"修改属性"查看或修改属性值	
选择已有云服务器:					
实例ID ▼ 请输入实例ID:	进行精确查询 搜索				
□ 实例ID	实例名称	IP地址(ID)	可用区	网络类型	实例规格
i-bp11fbo70x9p10y9a2s6	test	- (译性) 172.16.21.239 (私有)	cn-hangzhou-b	虚拟专有网络	ecs.t5-lc2m1.nano
i-bp1glob57mdve9ba3wzn	shukun-ECS	- (译性) 172.16.21.224 (私有)	cn-hangzhou-b	虚拟专有网络	ecs.s2.xlarge
i-bp13rsab25rrzk2sb7rq [请注意,当前节点没有公 网带宽或者挂载EIP,网络 不通的情况下添加实例到 集群会失败]	iZbp13rsab25rrzk2sb7rqZ	- (弹性) 172.16.166.8 (私有)	cn-hangzhou-g	虚拟专有网络	ecs.sn2ne.large
i-bp1ek6yd7jvkxk5y1phz	ecsdocTest	47.98.48.128 (弹性) 172.16.32.254 (私有)	cn-hangzhou-e	虚拟专有网络	ecs.n1.tiny
					步一才

b) 填写实例信息,设置登录密码,然后单击下一步。

选择已有云服务器实例	填写实例信息		>	添加完成	
集群ID/名称:	当前要添加的集群信息				
登录方式:	 设置密码 				
* 密码:	 密码为8-30个字符,必须同时包含三项(大、小写字母,数字和特殊符号),	不支持\"两个符号			
* 确认密码:	•••••				
实例信息:	实例ID	实例名称			
	-12-Acres(100)	test			
		shukun-ECS			
					上一步下一步

c) 在弹出的对话框中,单击确定,选择的 ECS 云服务器会自动添加到该集群中。

确认添加E	已有实例到集群	\times
	您确定要添加已选的ECS实例到集群 e3 / test-gpu中吗 ?	
	添加已有ECS的过程中,会更换系统盘,同时磁盘 ID 会变更,原系 统盘会被释放。 1.更换系统盘后,原系统盘用户快照会被保留,自动快照随磁盘释 放。 2.为保持足够的快照数量额度完成新磁盘的自动快照策略.您可以删除 不需要的快照。 因您没有备份系统相关个人数据而造成的数据丢失,阿里云不承担责 任。	
	确定取消	Í

- 6. (可选)选择手动添加的方式。
 - a)选择所需的 ECS 云服务器,单击下一步。您一次只能添加一个 ECS 云服务器。

选择已有云服务器实例	填写实例	l信息 >		添加完成
添加方式: ◎ 自动添加 ◎ 手动添加				
手动添加已有节点一次只能选择一个云服务器				
实例ID ▼ 请输入实例ID进行精确查询 搜索				
■ 实例ID 实例名称	IP地址(ID)	可用区	网络类型	实例规格
	- (弹性) 172.16.21.239 (私有)	cn-hangzhou-b	虚拟专有网络	ecs.t5-lc2m1.nano
i-bp1glob57mdve9ba3wzn shukun-ECS	- (弹性) 172.16.21.224 (私有)	cn-hangzhou-b	虚拟专有网络	ecs.s2.xlarge
i-bp13rsab25rrzk2sb7rq [清吉忠,当約Fiz/25合公 回 阿帶愛或者性裁EIP, 网络 [Zbp13rsab25rrzk2sb7rgZ 不通的情况下添加实例到 集群会失败]	- (弹性) 172.16.166.8 (私有)	cn-hangzhou-g	虚拟专有网络	ecs.sn2ne.large
i-bp1ek6yd7jvkxk5y1phz ecsdocTest	47.98.48.128 (弹性) 172.16.32.254 (私有)	cn-hangzhou-e	虚拟专有网络	ecs.n1.tiny
				3 下一步

b) 进入实例信息页面,确认无误后,单击下一步。

选择已有云服务器实例	填写实例信息	<u> </u>	添加完成
集群ID/名称:	当前要添加的集群信息		
登录方式:	 设置密码 		
* 密码 :	•••••• 密码为8-30个字符,必须同时包含三项(大、小写字母,数字和特殊符号),	不支持\"两个符号	
* 确认密码:	•••••		
实例信息:	实例ID	实例名称	
	- to a factor of the second	test	
	10 (000 10 000 00 00 00 00 00 00 00 00 00 00	shukun-ECS	
			上一步下一步

c) 进入添加节点页面, 您需要复制其中的执行命令。

选择已有云服务器实例	\rightarrow	填写实例信息		添加完成
仅支持添加同一 VPC 内,操作系统为 C	entOS 的节点			
登录要添加的节点,执行以下命令:				
			and here a second	

- d) 登录 ECS 管理控制台,选择集群所在的地域,选择需要添加的 ECS 实例。
- e) 登录 ECS 管理控制台,选择集群所在的地域,选择需要添加的 ECS 实例。
- f) 单击 ECS 实例右侧的远程连接。进入 ECS 实例远程连接界面,输入远程连接密码并单击确定,成功后,输入上面保存的命令,单击确定,开始执行脚本。

CentOS Linux 7 (Core) Kernel 3.10.0-693.2.2.e17.	复制粘贴命令	\times	
izbp13dqaawoex9fws2wsoz lo Password: Last failed login: Tue Mar There were 5 failed login Welcome to Alibaba Cloud E Iroot@izbp13dqaawoex9fws2w	将内容粘贴至文本框中,最大支持2000个字符,暂不支持中文等非际准键重值特殊字符。 *文本内容: Curl http://aliacs-k3e-cn-hangzhou.oss-cn- hangzhou.alivuncs.com/public/pkg/run/tatach/attach_node.sh bash sdocker-version 17.06.2-cg-1 -token 5925bc.395ebt1099b320305 -endpoint 172.16.253.150.6443 - cluster-dns 172.21.0.10		
	織定	取消	

9)等待脚本执行成功,该云服务器即添加成功。您可以在集群列表页面单击集群的 ID 查看该集 群下的节点列表。查看节点是否成功添加到集群中。

1.5.6 扩容和缩容集群

通过容器服务管理控制台,您可以根据实际业务需要对 Kubernetes 集群的 Worker 节点进行扩容和 缩容。

背景信息

使用说明

- 目前仅支持用户手动进行扩容和缩容,不支持自动弹性伸缩。
- 目前,不支持集群中 Master 节点的扩容和缩容。
- 集群缩容只能缩减集群创建和扩容时增加的 Worker 节点,通过"添加已有节点"功能添加到集群中的 Worker 节点不能被缩减。
- 缩减规则是按创建时间进行的,最新扩容出来的节点会被先回收。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的集群伸缩。

容器服务		集群列表				您最多可以创建 5 个集群,	,每个集群最多可以添加 20 个节点 刷	新 创建 Kubernetes 集群 🗸
Kubernetes Swan	rm	常见问题: & 如何创建集群 & 扩容和缩容	集群 ⑧道	動过 kubectl 连接 Kubernete	5 集群 🔗 通过命	令管理应用		
		名称 •						
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	操作
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理 直看日志 控制台 2 集群伸缩 更多▼

4. 选择扩容 或者缩容,设置 Worker 节点的数量。

本示例进行集群扩容,将集群的 Worker 节点数由 1 个扩容到 4 个。

集群名称	k8s-test	
地域	华东1 可用区G	
已有Worker数	1	
伸缩	扩容 缩容	
实例规格	2 核 16 G (ecs.se1ne.large) 🔹 🔻	
伸缩数量	3 台 🜲	
伸缩后 Worker 数	4	
* 登录密码	请输入创建集群时使用的登录密码	
RDS白名单	请选择你想要添加白名单的RDS实例	

5. 填写节点的登录密码。



由于升级过程依赖登录到 ECS 来拷贝配置信息,所以集群伸缩时填写的密码必须和创建集群时 填写的密码一致。

6. 单击提交。

后续操作

伸缩完成后,通过 Kubernetes 集群的节点列表页面,您可以看到 Worker 节点的数量从 1 变成了 4。

1.5.7 升级集群

您可以通过容器服务管理控制台升级您集群的 Kubernetes 版本。

您可以在 Kubernetes 集群列表页面查看您的集群的 Kubernetes 版本。

容器服务		集群列表				您最多可以创	建 5 个集群,每个集群最多可以添加 20 1	节点 刷新	创建 Kubernetes 集群	
Kubernetes	Swarm									
▼ 集群		常见问题: Ø 如何创建集群 Ø 扩容和缩容集群	Ø 通过 kubect	i 连接 Kubernetes 集群	⌀ 通过命令管理应用					
		2.22 V								
集群										
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本		操	作
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp1gnu8br4a	●运行中	2018-04-23 11:41:46	1.9.3	1	管理 查看日志 控制台 集群伸缩 更多	•

注意事项

• 集群升级需要机器可以公网访问,以便下载升级所需的软件包。

- 集群升级 Kubernetes 过程中,可能会有升级失败的情况,为了您的数据安全,强烈建议您先打 快照然后再升级。有关 ECS 打快照的操作参见创建快照。
- 集群升级 Kubernetes 过程中,升级期间用户应用不受影响,但是不建议此时通过 kubctl 和控制 台进行集群管理。升级时间大约 5-15 分钟,升级完成后集群会变成运行中状态。

准备工作

请在集群升级前检查集群的健康状况,并且确保集群健康。

登录 Master 节点,参见SSH 访问 Kubernetes 集群和 通过 kubectl 连接 Kubernetes 集群。

1. 执行kubectl get cs命令,确保所有模块都处于健康状态。

NAME	STATUS	MESSAGE	E	RROR
scheduler	Healthy	ok		
controller-manager	Healthy	ok		
etcd-0	Healthy	{"health":	"true"}	
etcd-1	Healthy	{"health":	"true"}	
etcd-2	Healthy	{"health":	"true"}	

2. 执行 kubectl get nodes 命令,确保所有节点都处于 Ready 状态。

kubectl get nodes					
NAME		STATU	JS	ROLES	AGE
VERSION					
cn-shanghai.i-xxxxxx	Ready	master	38d		v1.9.3
cn-shanghai.i-xxxxxx	Ready	<none></none>	38d		v1.9.3
cn-shanghai.i-xxxxxx	Ready	<none></none>	38d		v1.9.3
cn-shanghai.i-xxxxxx	Ready	<none></none>	38d		v1.9.3
cn-shanghai.i-xxxxxx	Ready	master	38d		v1.9.3
cn-shanghai.i-xxxxxx	Ready	master	38d		v1.9.3

如果节点不正常可以自行修复,也可以通过提交工单,请阿里云工程师协助修复。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 4. 选择所需的集群,并单击更多>集群升级。

容器服务		集群列表					您最多可以创	建 5 个集群,每个集群最多可以添加 20 个	や点 刷新	êlija	tubernetes 集群 🛛 👻	
Kubernetes	Swarm	常见问题: 8 如何创建集群 8 名称 ▼	9 扩容和缩容集群	♂ 通过 kubect	出 连接 Kubernetes 集群	☞ 通过命令管理应用						
节点存储		集群名称/ID k8stest		地域 华东1	网络类型 虚拟专有网络 vpc-bp1gnu8br4a	集群状态 ● 运行中	创建时间 2018-04-23 11:41:46	Kubernetes 版本		管理	提作 2 查看日志 注称, 集群仲编 更多。	
 ★ 应用 部署 服务 										3	 一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	

5. 在弹出的对话框中,单击升级。

集郡	洋升级			×	
	组件	当前版本	升级		
	Kubernetes	1.8.1	升级		
	提示:	公网法词 时间下载升码彩景的	加州有		
	 集群升级需要机器可以公网访问,以便下载升级所需的软件包 集群升级Kubernetes过程中,可能会有升级失败的情况,为了您的数据安全,强烈推荐 择先打快照然后再升级的方式 				
	3. 集群升级Kubernetes过 请您在升级之前安排好相	:程中,集群上部署的服务会中断, 关事宜。升级时间大约5~30分钟,	同时无法进行集群和应用的操作, 升级完成后集群会变成运行中状态		
			×	闭	

系统开始升级 Kubernetes 的版本。

集種	洋升级			\times
	组件	当前版本	升级	
	Kubernetes	1.8.1	升级	
		210		
			关闭	9

集積	洋升级			\times
	组件	当前版本	升级	
	Kubernetes	1.8.1	升级	
	升级已经开始			
			关	म

升级完成后,您可以在 Kubernetes 集群列表页面查看集群 Kubernetes 的版本,确认升级成功。

集群列表					您最多可以创建 5 个集群	,每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes ;	集群
常见问题: 🔗 如何创建集群 🔗 扩容和缩容集群	Ø 通过 kubect	连接 Kube	rnetes 集群 🔗 通过命令管理应用						
名称 ▼									
集群名称/ID	集群类型	地域	网络类型	集群状态	创建时间	Kubernetes 版本			操作
k8stest c77862d75bda644679b48d15f1a42e831	Kubernetes	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	2018-01-04 19:47:08	1.8.4	管理	查看日志 控制台	删除 更多▼
k8s c02879fa8ea2c47d48b6c4ce76524499f	Kubernetes	华东1	虚拟专有网络	●失败 原因:	2018-01-04 19:38:32	1.8.4	管理	查看日志 控制台	删除 更多▼

1.5.8 删除集群

您可以通过容器服务管理控制台删除不再使用的集群

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的更多 > 删除。

容器服务	集群列表		您最多可以	创建 5 个集群,每个集群最多可以添加 20 个	节点 刷新 创建 Kubernetes 集群 →
Kubernetes Swarm					
▼ 集群	■光円起: ○ 川川団建築計 ○ 1 音和地	合来計 0 通辺 Kubecu 連接 Kuberne	165 米叶 《 通过命令管理应用		
_{集群}	名称 ▼				
节点	集群名称/ID	地域 网络类型	集群状态 创建时间	Kubernetes 版本	2
存储	k8stest 🔽	华东1 虚拟专有网络 "	●运行中 2018-04-	20 11:02:31 1.9.3	管理 査看日志 控制台 集群伸缩 更多▼
▼ 应用					3 删除
部署					漆加已有节点
服务					乘 時 开 级 升 级 监 控 服务

后续操作

删除集群失败

如果您在 ROS 创建的资源下手动添加了一些资源,ROS 是没有权限删除这些资源的。比如在 ROS 创建的 VPC 下手动添加了一个 VSwitch,这样在就会导致 ROS 删除时无法处理该 VPC,从 而最终删除集群失败。

容器服务提供了强制删除集群的功能。通过强制删除功能,您可以在集群删除失败后,强制删除集 群记录和 ROS 资源栈。但是,强制删除操作不会自动释放您手动创建的这些资源,您需要手动进 行释放。

集群删除失败时,会显示如下信息。

集群列表			您最多可以创建	10 个集群,每个集群	詳最多可以添加 50 个节点	刷新	创建Serverless Kubernetes集群	创建 Kubernetes 集群 🗸 👻
常见问题: S 如何创建集群 S 扩容和缩容集群	Ø 通过 kubectl 连接 Kubernetes	5 集群 (⌀ 通过命令管理应用					
名称 ▼								
集群名称/ID	集群类型	地域	网络类型	集群状态	创建时间	Kube	rnetes 版本	操作
test-k8s c7837e45e82c8465ba91c981f39b301e5	Kubernetes	华东1	虚拟专有网络 vpc-bp13cwaie5z	●删除失败	2018-05-08 17:23:27	1.9.3		管理 查看日志 控制台 集群伸缩 更多→

单击删除失败的集群对应的更多 > 删除,在弹出的对话框里,您可以看到删除失败的资源,勾选强制删除并单击确定,即可删除集群和 ROS 资源栈。



说明:

该操作不会释放这些资源,您需要手动释放资源。有关如何如果排查不能释放的资源,可参见删除 *Kubernetes* 集群失败#ROS stack 无法删除。

删除集群 -	<8stest				\times
•	确定要删除集群 k8stest ?				
	3 强制删除 仅删除集群记录和 3	资源栈,您需要手动释放以下资源			
	资源ID	资源类型	状态	更新时间	
	sg-bp1jcjz	ALIYUN::ECS::SecurityGroup	● 创建完成	2018-05-08 17:23:40	
	lb-bp14g7	ALIYUN::SLB::LoadBalancer	● 创建完成	2018-05-08 17:23:41	
	i-bp1ik2gy	ALIYUN::ECS::Instance		2018-05-11 09:43:53	
	i-bp11cxjl	ALIYUN::ECS::Instance		2018-05-11 09:43:54	
	i-bp15mv	ALIYUN::ECS::Instance		2018-05-11 09:43:56	
	eip-bp1az	ALIYUN::VPC::EIPAssociation	●创建完成	2018-05-08 17:23:51	
	ngw-bp1i	ALIYUN::ECS::NatGateway	●创建完成	2018-05-08 17:23:40	
	rsp-bp1f2	ALIYUN::SLB::VServerGroup	●删除失败	2018-05-14 16:27:02	
	eip-bp1az	ALIYUN::VPC::EIP	● 创建完成	2018-05-08 17:23:40	
	lb-bp1s3lz	ALIYUN::SLB::LoadBalancer	● 创建完成	2018-05-08 17:23:40	
	lb-bp1s3lz	ALIYUN::SLB::BackendServerAt tachment	●创建完成	2018-05-08 17:27:55	
	Kubernete d6-9e06-4 e55	ALIYUN::RAM::Role		2018-05-14 16:27:02	
	Kuberneta d6-9e06-4 e55	ALIYUN::RAM::Role	● 创建完成	2018-05-08 17:23:41	
	snat-bp12 z	ALIYUN::ECS::SNatEntry	●创建完成	2018-05-08 17:23:55	
				确定取	消

1.5.9 查看集群概览

阿里云容器服务 Kubernetes 集群提供集群概览概览功能,提供应用状态、组件状态和资源监控等功能,方便您快速了解集群的健康状态信息。

操作步骤

1. 登录容器服务管理控制台。

- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的概览,进入 Kubernetes 集群概览页面。
- 3. 选择所需的集群和命名空间,您可查看应用状态、组件状态和资源监控图表。
 - 应用状态:显示当前运行的部署、容器组和副本集的状态示意图,绿色图标代表正常,黄色
 图标代表异常。
 - 组件状态:Kubernetes 集群的组件通常部署在 kube-system 命名空间下,包括 scheduler、controller-manager和 etcd 等核心组件。
 - 资源监控:提供 CPU 和内存的监控图表。CPU 统计单位为 Cores(核),可显示小数点后3位,最小统计单位是 millcores,即一个核的 1/1000;内存的统计单位是 Gi,显示小数点后3位。更多相关信息,请参见 Meaning of CPU和 Meaning of memory。



1.6 节点管理

1.6.1 查看节点列表

您可以通过命令、容器服务管理控制台的节点列表页面或者 Kubernetes Dashboard 的节点列表页 面查看 Kubernetes 集群的节点列表。

通过命令查看

📕 说明:

使用命令查看 Kubernetes 集群的节点列表页面之前,您需要先设置通过 kubectl 连接 Kubernetes 集群。

通过 kubectl 连接到 Kubernetes 集群后,运行以下命令查看集群中的节点。

kubectl get nodes

输出示例:

```
$ kubectl get nodes
NAME
                         STATUS
                                   AGE
                                             VERSION
iz2ze2n6ep53tch701yh9zz
                         Ready
                                   19m
                                             v1.6.1-2+ed9e3d33a07093
iz2zeafr762wibijx39e5az
                         Ready
                                   7m
                                             v1.6.1-2+ed9e3d33a07093
iz2zeafr762wibijx39e5bz Ready
                                   7m
                                             v1.6.1-2+ed9e3d33a07093
iz2zef4dnn9nos8elyr32kz
                         Ready
                                   14m
                                             v1.6.1-2+ed9e3d33a07093
iz2zeitvvo8enoreufstkmz
                         Ready
                                   11m
                                             v1.6.1-2+ed9e3d33a07093
```

通过容器服务管理控制台查看

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 节点,进入节点列表页面。
- 3. 选择所需的集群,您可以查看该集群下的节点列表。

通过 Kubernetes Dashboard 查看

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的控制台,进入 Kubernetes Dashboard。

容器服务	5	集群列表				您最多可以创建 5 个集群	¥ , 每个集群最多可以添加 20 个节点	刷新 创建 Kubernetes 集群 🗸
Kubernete	s Swarm							
▼ 集群		常见问题: Ø 如何创建集群 Ø 扩容和缩容集	群 ⑧ 通过	번 kubectl 连接 Kubernetes 集	群 🕜 通过命令管	理应用		
集群	1	名称 ▼						
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	2 ⁼
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理 查看日志 控制台 集群伸缩 更多▼

4. 在 Kubernetes Dashboard 中,单击左侧导航栏中的节点。即可查看集群中的节点列表。



1.6.2 节点标签管理

您可以通过容器服务 Web 界面对节点进行标签管理,包括批量添加节点标签、通过标签筛选节点和快速删除节点标签。

关于如何使用节点标签实现节点调度,请参见指定节点调度。

前提条件

您已经成功创建一个 Kubernetes 集群,参见 创建 Kubernetes 集群。

批量添加节点标签

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 节点,进入节点列表页面。
- 4. 选择所需的集群,在页面右上角单击标签管理。

容器服务		节点列表					周	新 标签管理 集群伸缩	添加已有节点
Kubernetes	Swarm								
▼ 毎難	1	常见问题: ③ 按量付费	诗包年包月						
生形		集群: k8s-cluster ▼	3						
	•	IP地址	实例类型	角色	实例ID/名称	配置	操作系统	更新时间	操作
节点存储	2	10.000	阿里云ECS	Worker	k8s-for-cs-c98a7	ecs.sn1ne.large	centos_7_04	2018-04-23 11:58:25	监控

5. 在节点列表中,批量选择节点,然后单击添加标签。

√ 标签管理 t 返回		刷新
	IP地址	标签
Cn-hangzhou.i-	.27	
Cn-hangzhou.i-	.26	
Cn-hangzhou.i-	.29	
✓ cn-hangzhou.i-	.28	
□ ī雨加雨壶 2		

6. 在弹出的添加标签对话框中,输入标签的名称和值,然后单击确定。

添加		\times
名称	group	
值	master	
	确定	关闭

您可以在标签管理页面,看到批量节点具有相同的标签。

「「「茶管理」で近回」		周新
□ 名称	IP地址	标签
on-ha	.27	group:master 🔇
on-ha	.26	group:master 🔇
on-harmonic and the second	.29	
on-hu	.28	group:master 🛇
运 加标签		

通过标签筛选节点

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 节点,进入节点列表页面。
- 4. 选择所需的集群,在页面右上角单击标签管理。

容器服务		节点列表					周新	标签管理 集群伸缩	添加已有节点
Kubernetes	Swarm								
▼ 毎群	1	常见问题: S 按量付费转	包年包月						
集群		集群: k8s-cluster ▼	3						
	•	IP地址	实例类型	角色	实例ID/名称	配置	操作系统	更新时间	操作
节点	2	10.000	阿用于ECS	Worker	The support of the su	ers snine jarne	centos 7 04	2018-04-23 11:58:25	市台
存储			P518222003	THUR NOT	k8s-for-cs-c98a7	cestoria rendi ge	contros_7_04	2010 01 20 11:00:20	ILM

5. 选择某个节点,单击右侧的标签,如 group:worker,可通过标签来筛选节点。

您可看到通过 group:worker 标签成功筛选出所需的节点。

标签管理 ← 返回		T group:worker ③ 周新
名称	IP地址	标签
cn-hangzhou.	10.000	group:worker 🛇
汤加标签		

删除节点标签

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 节点,进入节点列表页面。
- 4. 选择所需的集群,在页面右上角单击标签管理。

容器服务		节点列表					局折	析 标签管理 集群伸缩	添加已有节点
Kubernetes	Swarm	常见问题: Ø 按量付费转	包年包月					4	
集群		集群: k8s-cluster ▼	3						
1		IP地址	实例类型	角色	实例ID/名称	配置	操作系统	更新时间	操作
存储		10.000	阿里云ECS	Worker	k8s-for-cs-c98a7	ecs.sn1ne.large	centos_7_04	2018-04-23 11:58:25	监控

5. 选择某个节点,单击标签的删除图标,如 group:worker。

标签管理 🔹 返回		刷新
— 名称	IP地址	标签
0 m	27	group:master 🔇
0 m	26	group:master 🔇
0 m-	29	group:worker
α.·····	28	group:master 🛇
■ 添加标签		

您可以看到该节点右侧的标签消失,节点标签被删除。

标签管理		周期
各称	IP地址	标签
0 cn-	27	group:master 🕲
α	26	group : master 🚳
α	29	group:worker 🛛 🚺
an-	28	group:master 🚳
□ 汤如际签		

1.6.3 节点监控

kubernetes 集群与阿里云监控服务无缝集成,您可以查看 kubernetes 节点的监控信息,了解 Kubernetes集群下 ECS 实例的节点监控指标。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 节点,进入节点列表页面。
- 3. 选择所需的集群,在该集群下选择所需的节点。
- 4. 单击监控,查看对应节点的监控信息。

容器服务	节点列表						刷新 集群伸缩	添加已有节点
Kubernetes Swarm	常见问题: ⑧ 按量付	费转包年包月						
	集群: k8s-cluster	• 3						
	IP地址	实例类型	角色	实例ID/名称	配置	操作系统	更新时间	4
存储	100.00	阿里云ECS	Worker		ecs.gn5-c8g1.2xlarge	centos_7_04	2018-04-20 11:21:10	监控
▼ 应用	10100	阿里云ECS	Master	2000	ecs.sn1ne.large	centos_7_04	2018-04-20 11:21:11	监控
影響	10.05.0501	阿里云ECS	Master		ecs.sn1ne.large	centos_7_04	2018-04-20 11:21:11	监控
发布		阿里云ECS	Master	and the second	ecs.sn1ne.large	centos_7_04	2018-04-20 11:21:11	监控

 进入云监控管理控制台,查看对应 ECS 实例的基本监控信息,包括 CPU使用率、网络流入带 宽、网络流出带宽、系统磁盘 BPS、系统盘 IOPS 等指标。



后续操作

要想查看关于操作系统级别的监控指标,需要安装云监控组件。参见主机监控概览。

Kubernetes 集群新增了关于应用分组的监控功能,您可以参见 通过资源分组进行监控与告警进行 升级。

1.7 应用管理

1.7.1 通过 Kubernetes Dashboard 创建应用

您可以通过 Kubernetes Dashboard 创建应用。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的控制台,进入 Kubernetes Dashboard。

容器服务 - Kubernetes ▼	集群列表		您最多	可以创建 5 个集群,每个	集群最多可以渗	劾 40 个节点	刷新	创建Serverless Kubernetes集群	创建 Kubernetes 集群 🛛 👻
概览	常见问题: Ø 如何创建集群 Ø 扩容和新	i容集群 🔗 通过 kubectl 连	接 Kubernetes 集群	∂ 通过命令管理应用					
▼ 集群	名称 ▼								
集群	集群名称/ID	集群类型	地域 (全部) 🔻	网络类型	集群状态	创建时间		Kubernetes 版本	2 =
节点	k8s-blockchain	Kubernetes	华东1	盧拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-06-27 17:	:48:29	1.9.7	管理 查看日志 <u>控</u> 制台 集群伸缩 更多▼

4. 在 Kubernetes Dashboard 中,单击页面右上角的创建。

<	≡ 概況		十创建
集群	工作负载		
命名空间	工作负载状态		
持久化存储卷			
角色	100.00%	100.00%	100.00%
存储类			
命名空间 default ▼	部署 	容器组	副本集

5. 在弹出的对话框中,设置应用的信息。

您可以通过以下3种方法之一创建应用:

• 使用文本创建:直接输入 YAML 或 JSON 格式的编排代码创建应用。您需要了解对应的编排 格式,如下所示是一个 YAML 格式的编排模板。

1 ap	apiVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2	
2 ki	kind: Deployment	
3 - me	metadata:	
4	name: nginx-deployment	
5 - sp	spec:	
б-	selector:	
7 -	matchLabels:	
8	app: nginx	
9	replicas: 2	
10 -	template:	
11 -	metadata:	
12 -	labels:	
13	app: nginx	
14 -	spec:	
15	Containers	
10 -	- name: nginx	
1/	Image: hginx:1.7.9	
18	ports:	
19	- containerPort. so	

- 使用配置创建:通过指定下边的设置创建应用。
 - 一应用名称:所创建应用的名称。本示例中为 nginx。
 - 容器镜像:所要使用的镜像的 URL。本示例使用的是 Docker Nginx。
 - 容器组数量:创建的应用的 pod 个数。

- 服务:可设置为外部或内部。外部表示创建一个可以从集群外部访问的服务;内部表示创建一个集群内部可以访问的服务。
- 高级选项:您可以选择显示高级选项,对标签、环境变量等选项进行配置。此设置将流量
 负载均衡到三个 Pod。

使用文本创建	使用文件创建	使用配置创建		
应用名称 [*] nginx-test				一个以此为值的'app'标签将被添加到部署和服务上了解更多 🛛
容静镜像 "			10 / 24	
nginx				au/11世紀/ATHRERRIGHSFaresy UKL, BK, DOCKEY HUD, Google Container Kegistry 平出が4月15日家 J 新史 多 CI
容靜坦致量 * 3				部署将会被创建,由它维护集群中容器组的所需数量了解更多 2
^{服务*} 外部			*	可指定内部或外部服务请口,映射到容器所监听的满口 此服务的内部 DNS 名称将会是: nginx-test. 了城西条 ct
端口 * 80		目标端口 * 9080	10校 * TCP -	
			453.07 ¹⁰	
端口		目标端口	TCP *	
显示高级选项				
部署	取消			

- 使用文件创建:通过导入已有的 YAML 或 JSON 配置文件创建应用。
- 6. 单击部署 部署这些容器和服务。

您也可以单击显示高级选项 展开高级选项进一步配置相关参数。

后续操作

单击部署后,您可以查看应用的服务或查看应用的容器。

单击左侧导航栏中的容器,您可以通过左侧的图标查看每个 Kubernetes 对象的状态。 🕡 表示对

象仍然处于部署状态。 🌄 表示对象已经完成部署。



1.7.2 使用镜像创建应用

前提条件

创建一个 Kubernetes 集群。详情请参见创建 Kubernetes 集群。

操作步骤

- 1. 登录容器服务管理控制台。
- 在Kubernetes菜单下,单击左侧导航栏中的应用>部署,然后单击页面右上角的使用镜像创 建。
- 3. 设置应用名称、部署集群和命名空间,并单击下一步进入应用配置页面。

如果您不设置命名空间,系统会默认使用 default 命名空间。

創建应用						
应月	用基本信息	应用配置	\rangle	访问设置	\rangle	创建完成
应用名称	nginx-default 名称为1-64个字符,可包含数字、小写	"英文字符,或"-",目不能以-开头				
部署集群:	k8s-cluster	¥				
命名空间:	default	T				
						返回下一步

- 4. 设置应用的基本配置。
 - 镜像名称:您可以单击选择镜像,在弹出的对话框中选择所需的镜像并单击确定,本例中为 nginx。

您还可以填写私有 registry。填写的格式为domainname/namespace/imagename:tag

- 镜像版本:您可以单击选择镜像版本选择镜像的版本。若不指定,默认为 latest。
- 容器数量:可指定容器的数量,本示例是单容器 Pod,若指定多个容器,会启动相同数量的 Pod。

镜像名称:	nginx	镜像版本:	
	选择镜像		选择镜像版本
容器数量:	1		

- 5. 设置容器的资源限制和资源预留。
 - 资源限制:可指定该应用所能使用的资源上限,包括 CPU 和 内存两种资源,防止占用过多资源。
 - 所需资源:即为该应用预留资源额度,包括 CPU 和内存两种资源,即容器独占该资源,防止因资源不足而被其他服务或进程争抢资源,导致应用不可用。

其中,CPU资源的单位为 millicores,即一个核的千分之一;内存的单位为 Bytes,可以为 Gi、Mi 或 Ki。

资源限制:	CPU 如:500m	● 内存 如:128Mi	0
所需资源:	CPU 如:500m	● 内存 如:128Mi	0

6. 配置数据卷信息。

支持配置 hostpath 类型的数据卷。hostpath 类型的数据卷会将主机文件系统中的文件或目录挂载到 pod 中。更多信息参见 volumes 关于 hostpath 的信息。

本例中配置了一个名为 data 的数据卷,类型为 hostPath,使用宿主机的目录/*tmp*,并将 data 数据卷挂载到容器中var/lib/docker 路径下。

数据卷:	🚯 增加本地存储			
	存储卷类型	挂载源	容器路径	
		1+++0.02		
	仔唯巷尖型	注筆記法	谷箭焰住	
	云盘	 pvc-yunpan-test 	▼ /tmp	•

7. 配置环境变量。

支持通过键值对的形式为 Pod 配置环境变量。用于给 Pod 添加环境标志或传递配置等,具体请参见 Pod variable。

8. 配置容器信息。

您可以为运行在 Pod 中的容器配置 Command、Arguments 和容器启动项。

- Command 和 Args:若不配置,会使用镜像默认的设置;若配置,则会覆盖镜像里的 默认设置,并且如果只配置 Arguments,容器启动时,默认的 Command 会执行新的 Arguments。Command 和 Arguments 在 Pod 创建之后不可更改。
- 容器启动项:勾选 stdin 表示为该容器开启标准输入;勾选 tty 表示为该容器分配一个虚拟终端,以便于向容器发送信号。通常这两个选项是一起使用的,表示将终端(tty)绑定到容器的标准输入(stdin)上,比如一个交互式的程序从用户获取标准输入,并显示到终端中。
- 9. 设置自动伸缩。

您可勾选是否开启自动伸缩,为了满足应用在不同负载下的需求,容器服务支持服容器的弹性伸缩,即根据容器 CPU 和内存资源占用情况自动调整容器数量。

10.完成应用配置后,单击下一步,进入访问设置页面,即设置一个绑定后端 pod 的服务。

常见	1.问题: 🔗 限制	容器的资源 🔗 高可用性调度 🔗 通过镜像创建Nginx 🔗 通过编排模板创	建Wordpress 🔗 编排模板说明 🔗 标签说明	
		应用基本信息	访问	创建完成
超	服务:	 不创建 進以集群IP 节点第二 受数均衡 ① 内部访问 		
		名称:		
		nginx-default-svc		
访问论		端口映射:		
		靖口	容器端口(targetPort)	协议
	2	80	8080	тср то
		 添加 		3
				上一步创建

- 服务:您可以选择不创建服务,或者选择服务类型,从下面3种服务类型中进行选择。
 - 虚拟集群 IP:即 ClusterIP,指通过集群的内部 IP 暴露服务,选择该项,服务只能够在集群内部可以访问。
 - 节点端口:即NodePort,通过每个Node上的IP和静态端口(NodePort)暴露服务。NodePort服务会路由到ClusterIP服务,这个ClusterIP服务会自动创建。通过请求

 NodeIP>:<NodePort>,可以从集群的外部访问一个NodePort服务。
 - — 负载均衡:即LoadBalancer,是阿里云提供的负载均衡服务,可选择公网访问或内网访问。负载均衡可以路由到 NodePort 服务和 ClusterIP 服务。
- 名称:默认会生成一个应用名后缀 svc 的服务名称,本例中为 nginx-default-svc,您也可更 改服务的名称。
- 端口映射:您需要添加服务端口和容器端口,若您选择节点端口的服务类型,还需要自己设置节点端口,防止端口出现冲突。支持 TCP/UDP 协议。

11.完成访问配置后,单击创建。

创建成功后,进入创建完成页面,会列出应用包含的对象,您可以前往部署列表查看。

创建应用								
常见问题:			⑦ 通过镜像创建Nginx	Ø 通过编排模板创建Wordpress		☞ 标签说明		
	应用基本信	息	\rightarrow	应用配置	\rightarrow	访问设置	创建完成	
				前往部署列表 <mark>世委</mark> 创建部署 nginx-default- 创建Service nginx-defaul 创建水平扩展对象 成功	deployment 成功 ilt-svc 成功 重新创建			

12.您可以看到新建的 nginx-default-deployment 出现在部署列表下。

部署列表				刷新	使用镜像创建	使用模板创建
集群 k8s-cluster v 命名空间 default	v					
名称	标签	容器组数量	创建时间			提作
nginx-default-deployment	app:nginx	2/2	2018-07-19 16:32:54		详情 编	攝 监控 更多▼

13.单击左侧导航栏的应用 > 服务,可以看到新建的服务 nginx-default-svc 出现在服务列表下。

容器服务 - Kubernetes ▼	服务列表						刷新	创建
概览	集群 k8s-cluster 🔻 🕯	命名空间 default 🔻 3						
▼ 集群	名称	<u>送型</u> 创建时间	集群IP	内部端点	外部端点			操作
集群	kubernetes Cl	ClusterIP 2018-07-11 17:35:35	10.00	kubernetes:443 TCP		详情 更新	查看YAML	删除
节点	nginx-default-svc Lo	LoadBalancer 2018-07-19 16:34:52	11100	nginx-default-svc:80 TCP nginx-default-svc:30401 TCP	1:80	详情 更新	查看YAML	删除
命名空间								
▼ 应用 1 1								
容器组								
服务 2 Ξ								

14. 在浏览器中访问外部端点,您可访问 nginx 欢迎页。

C 1 /?spm=5176.2020520152.0.0.49	li6ddltgTUa	
	Welcome to nginx!	
	If you see this page, the nginx web server is successfully installed and working. Further configuration is required.	
	For online documentation and support please refer to <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .	
	Thank you for using nginx.	

1.7.3 通过编排模板创建应用

前提条件

```
创建一个 kubernetes 集群,参见创建 Kubernetes 集群。
```

背景信息

在容器服务 kubernetes 模板编排中,您需要自己定义一个应用运行所需的资源对象,通过标签选择 器等机制,将资源对象组合成一个完整的应用。

本例演示如何分步创建 nginx 应用。首先通过部署 deployment 创建后端 pod 资源对象,然后部署 service 将其绑定到后端 pod 上,形成一个完整的 nginx 应用。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,进入部署列表页面。

3. 单击页面右上角的使用模板创建。

容器服务		部署列表		使用	镜像创建 使用模板创建 刷新
Kubernetes	Swarm				
▼ 集群		集群 k8s-cluster ▼ 命名空间 default	v		
集群		名称 标签	容器组数量	创建时间	操作
节点			① 没有	查询到符合条件的记录	
存储					
▼ <u>应</u> 用 部署	1				
服务					
发布					
配置项					

- 4. 对模板进行相关配置,完成配置后单击创建。
 - 集群:选择目标集群。资源对象将部署在该集群内。
 - 命名空间:选择资源对象所属的命名空间,默认是 default。除了节点、持久化存储卷等底层 计算资源以外,大多数资源对象需要作用于命名空间。
 - 示例模板:阿里云容器服务提供了多种资源类型的 Kubernetes yaml 示例模板,让您快速部 署资源对象。您可以根据 Kubernetes Yaml 编排的格式要求自主编写,来描述您想定义的资 源类型。

使用模版创建		
仅支持 Kubernet	es 版本 1.8.4 及	以上的集群。对于 1.8.1 版本的集群,您可以在集群列表中进行"集群升级 "操作
	集群	k8s-cluster
1	命名空间	default v
	示例模版	Resource - basic Deployment
	模版	<pre> byiversion: apps/vibeta2 f for versions before 1.8.0 use apps/vibeta1 kind: Deployment mac: nginx-deployment-basic app: nginx repplicas: 2 aps: nginx app: nginx repplicas: 2 aps: nginx for app: nginx app: nginx for app: nginx</pre>

下面是一个 nginx 应用的 deployment 示例编排,通过该编排模板,即可快速创建一个属于

nginx 应用的 deployment。

```
apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/
vlbetal
kind: Deployment
metadata:
    name: nginx-deployment
    labels:
      app: nginx
 spec:
    replicas: 2
    selector:
     matchLabels:
       app: nginx
    template:
      metadata:
        labels:
          app: nginx
      spec:
        containers:
        - name: nginx
          image: nginx:1.7.9 # replace it with your exactly <</pre>
image_name:tags>
          ports:
          - containerPort: 80
```

5. 单击创建后。会提示部署状态信息。成功后,单击Kubernetes 控制台前往控制台查看部署进

度。

<	三 工作负载 部署					+ 创建
集群	部署					Ŧ
命名空间	名称 🗢	标签	容器组	已创建 🗢	镜像	
持久化存储卷	oginx-deployment-nodeselector	app: ngl	2 / 2	2018-02-05 15:13:03	nginx:1.7.9 docker.io/istio/proxy_debug:0.4.0	:
角色						
存储类						
命名空间						
default 👻						
概览						
工作负载						
守护进程集						

6. 再返回使用模板页面,部署一个 Service 资源对象。

容器服务提供了一个 nginx 应用的 service 示例模板。在本示例中,您只需要对该模板进行简单 修改,将访问类型修改为 LoadBalancer,即可创建一个绑定后端 pod 的 service,让您从浏览器 进行访问。



本示例中, pod 和 service 编排中 selector 的值都是 nginx,因此不必修改,实际情况下您需要 进行相应修改。

```
apiVersion: vl
                  # for versions before 1.8.0 use apps/v1beta1
kind: Service
metadata:
                           #TODO: to specify your service name
  name: my-service1
  labels:
    app: nginx
spec:
  selector:
    app: nginx
                          #TODO: change label selector to match
your backend pod
  ports:
   - protocol: TCP
    name: http
    port: 30080
                           #TODO: choose an unique port on each
node to avoid port conflict
    targetPort: 80
   type: LoadBalancer
                            ##本例中将type从Nodeport修改为LoadBalancer
```

7. 将上述编排内容输入到模板中,然后单击创建。会提示部署状态信息。若成功后,单

击Kubernetes 控制台前往控制台查看服务部署的进度。

集群	k8s-test	Ŧ
命名空间	default	¥
示例模版	Resource - Service	٣
模版	1 apiVersion: v1 2 kind: Service 3 metadata: 4 name: my-servicel #TODO: to specify your service name 5 labels: 6 app: nginx-svc 7 spec: 8 selector: 9 app: nginx #TODO: change label selector to match your backend pod 10 ports: 11 - protocol: TCP 13 port: 30080 #TODO: choose an unique port on each node to avoid port conflict 14 targetPort: 80 15 type: LoadBalancer	
		1 譯

8. 在 kubernetes Dashboard 里,您可以看到 my-service1 服务已成功部署,并暴露了外部入口。 单击外部入口的访问地址。

<		跨				+
集群	服务					
命名空间	名称 ≑	标签	集群 IP	内部入口	外部入口	已创建 ◆
节点	my-service1	app: nginx-s	1000	my-service1:30080 TCP	:30080	2018-02-05 15:48:26
角色	kubernetes	component: apiser	1000-00	kubernetes:443 TCP	. 2	2018-02-01 15:40:23
存储类		provider: kubernet		kubernetes:0 TCP		
命名空间						
default 👻						
概览						
工作负载						
守护进程集						
部署						
(£\$						
谷間辺 副本集						
副本控制器						
有状态副本集						
服务发现与负载均衡						
访问权 1						

9. 您可以在浏览器中访问 nginx 服务欢迎页面。

①:30080	
AND ADDRESS STRAINING AND	and a second
	Welcome to nainy!
	If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
	For online documentation and support please refer to <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .
	Thank you for using nginx.

1.7.4 通过命令管理应用

您可以通过命令创建应用或者查看应用的容器。

前提条件

在本地使用命令前,您需要先设置通过 kubectl 连接 Kubernetes 集群。

通过命令创建应用

可以通过运行以下语句来运行简单的容器(本示例中为 Nginx Web 服务器):

```
# kubectl run -it nginx --image=registry.aliyuncs.com/spacexnice/
netdia:latest
```

此命令将为该容器创建一个服务入口,指定 --type=LoadBalancer 将会为您创建一个阿里云负载均衡路由到该 Nginx 容器。

kubectl expose deployment nginx --port=80 --target-port=80 --type= LoadBalancer

通过命令查看容器

运行如下命令列出所有 default 命名空间里正在运行的容器。

root@master # kubectl get pods			
NAME	READY	STATUS	RESTARTS
AGE			
nginx-2721357637-dvwq3	1/1	Running	1
9h			

1.7.5 利用 Helm 简化应用部署

在 Kubernetes 中,应用管理是需求最多、挑战最大的领域。Helm 项目提供了一个统一软件打包方式,支持版本控制,可以大大简化 Kubernetes 应用分发与部署中的复杂性。

阿里云容器服务在应用目录管理功能中集成了 Helm 工具,并进行了功能扩展,支持官方

Repository,让您快速部署应用。您可以通过命令行或容器服务控制台界面两种方式进行部署。

本文档介绍 Helm 的基本概念和使用方式,演示在阿里云的 Kubernetes 集群上利用 Helm 来部署示 例应用 WordPress 和 Spark。

Helm 基本概念

Helm 是由 Deis 发起的一个开源工具,有助于简化部署和管理 Kubernetes 应用。

Helm 可以理解为 Kubernetes 的包管理工具,可以方便地发现、共享和使用为 Kubernetes 构建的 应用,它包含几个基本概念

Chart: 一个 Helm 包,其中包含了运行一个应用所需要的镜像、依赖和资源定义等,还可能包含 Kubernetes 集群中的服务定义,类似 Homebrew 中的 formula、APT 的 dpkg 或者 Yum 的 rpm 文件。

- Release:在Kubernetes集群上运行的Chart的一个实例。在同一个集群上,一个Chart可以 安装很多次。每次安装都会创建一个新的release。例如一个MySQLChart,如果想在服务器上 运行两个数据库,就可以把这个Chart安装两次。每次安装都会生成自己的Release,会有自己 的Release名称。
- Repository:用于发布和存储 Chart 的存储库。

Helm 组件

Helm 采用客户端/服务器架构,由如下组件组成:

- Helm CLI 是 Helm 客户端,可以在 Kubernetes 集群的 master 节点或者本地执行。
- Tiller 是服务器端组件,在 Kubernetes 集群上运行,并管理 Kubernetes 应用程序的生命周期。
- Repository 是 Chart 存储库, Helm 客户端通过 HTTP 协议来访问存储库中 Chart 的索引文件和 压缩包。

使用 Helm 部署应用

前提条件

• 通过 Helm 部署应用之前,利用阿里云容器服务来创建 Kubernetes 集群。参见创建 Kubernetes 集群。

在 Kubernetes 集群创建的同时, Tiller 将会被自动部署到集群之中,并且在所有的 master 节点 上自动安装 Helm CLI 以及配置指向阿里云的 Chart 存储库。

• 查看您集群中 Kubernetes 的版本。

仅支持 Kubernetes 版本 1.8.4 及以上的集群。对于 1.8.1 版本的集群,您可以在集群列表中进行集群升级 操作。

通过控制台界面部署应用

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的市场 > 应用目录,进入应用目录列表页面。
- 4. 选择一个 chart (本示例选择 WordPress),单击该 chart,进入 chart 详情页面。

● 無詳	Øo	Øo	Øo	Øo
集群 节点 存储卷	hdfs-namenode-k8s stable	jenkins 2.67 stable	kafka incubator	monocular v0.5.3 stable
命名空间 ▼ 应用	Øo	Øo	Øo	Øo
部署 容積但 服务	mssql-linux 14.0.3023.8 stable	mysql stable	mysqiha 5.7.13 incubator	nodered incubator
路由 存储声明	Øo	Øo	Øo	Øo
发布 配置项 保密字典	postgresql stable	redis 3.2.9 stable	redis-ha 4.0.8-r0 incubator	spark stable
▼市场 税象	Øo	Øo	Øo	Øo
应用目录 2 服务目录	spark-oss incubator	tiller-rest-proxy 0.7.3 stable	3 wordpress 4.8.2 stable	zookeeper incubator

- 5. 在页面右侧,填写部署的基本信息。
 - 集群:应用要部署到的集群。
 - 命名空间:选择命名空间。默认为 default。
 - 发布名称:填写应用发布的名称。本例中为 test。

说明 参数	创建
WordPress	何本社 Kubamatag ビナ 1 8 4 R.M. Láb住職 オエ
WordPress is one of the most versatile open source content management systems on the market. A publishing platform for building blogs and websites.	1.8.1版本的集群,您可以在集群列表中进行"集群升级"操作。不支持ServerlessKubernetes集群。
TI.DD.	集群 k8e.cluster
IL;DK;	<u>命名</u> 変向
<pre>\$ helm install stable/wordpress</pre>	default 🔻
	发布名称
Introduction	wordpress-default
This chart bootstraps a WordPress deployment on a Kubernetes cluster using the Helm package manager.	创建
It also packages the Bitnami MariaDB chart which is required for bootstrapping a MariaDB deployment for the database requirements of the WordPress application.	

6. 单击参数,对配置进行修改。

本示例中将持久化存储参数从 true 修改为 false。

说明	参数	创建
79 80 -	## Password for maniadhliser	A
81	<pre>## ref: https://github.com/bitnami/bitnami-docker-mariadb/blob/master/README.md#crew</pre>	ting-a
		Q支持 Kubernetes 版本 1.8.4 及以上的集群。对于
82		101 版大的生料 你可以在生料列车由进行"生料升
83		1.0.1版本的来研,应可以让来研究这个应用 来研究
84		级 媒作。 不文持ServerlessKubernetes集群。
85 -		(# 1 14
86		来群
87	##	k8s-cluster v
00 -	enabled: tave	
90	## mariadb data Persistent Volume Storage Class	
91	## If defined, storage(lass)ame: <storage(lass)< td=""><td>命名空间</td></storage(lass)<>	命名空间
92	## If set to "-", storageClassName: "", which disables dynamic provisioning	default
93		Uerduit. *
94		
95		发布名称
96		
97	storageClass: "alicloud-disk-efficiency"	wordpress-default
98	accessmode: Readwriteonce	
100	5126. 2001	
101 -	## Kubernetes configuration	创建
102	## For minikube, set this to NodePort, elsewhere use LoadBalancer	
103	##	
104	serviceType: LoadBalancer	

- 7. 配置完成后,单击部署。
- 8. 部署成功后,进入该应用的发布页面,找到test-wordpress服务,单击服务名称,进入

Kubernetes Dashboard $_{\circ}$

容器服务 - Kubernetes ▼	1	发布列表 - wordpress-default						刷新
概览	1	当前版本						
▼ 集群								
集群		发布名称: wordpress-default 命名空				部署时间: 2018-07-10 17:11:24		
节点		当前版本: 1					更新时间:	2018-07-10 17:11:24
存储卷		资源				参数		
命名空间		资源		类型				
▼ 应用 1		wordpress-default-mariadb		Secret				查看YAML
部署		wordpress-default-wordpress		Secret				查看YAML
容器组		wordpress-default-mariadb		ConfigM	ap			查看YAML
肥久	-	wordpress-default-mariadb		Persiste	ntVolumeClaim			查看YAML
NK35		wordpress-default-wordpress		Persiste	ntVolumeClaim			查看YAML
路由		wordpress-default-mariadb		Service				查看YAML
存储声明		wordpress-default-wordpress		Service				查看YAML
发布 2		wordpress-default-mariadb		Deployn	nent			查看YAML
配置项		wordpress-default-wordpress		Deployn	nent			查看YAML

9. 默认进入 test-wordpress 服务详情页面,您可获取 http/https 外部端点的地址。

容器服务 - Kubernetes ▼	服务列表							刷新	创建
概览	集群 kBs-cluster v 命名空间 default v								
▼ 集群	名称	类型	创建时间	集群IP	内部端点	外部端点			撮作
集群	kubernetes	ClusterIP	2018-06-27 17:53:49	10.0	kubernetes:443 TCP		详情 更新	查看YAML	制除
京	wordpress-default-mariadb	ClusterIP	2018-07-10 17:36:16	100.000	wordpress-default-mariadb:3306 TCP		详情 更新	查看YAML	一删除
存储卷 命名空间	wordpress-default-wordpress	LoadBalancer	2018-07-10 17:36:16	100.000	wordpress-default-wordpress:80 TCP wordpress-default-wordpress:32109 TCP wordpress-default-wordpress:443 TCP wordpress-default-wordpress:32094 TCP	:80 :443	详情 更新	查看YAML	删除
部署									
容器组 服务 2 Ξ									
10.单击上面的访问地址,进入 WordPress 博客发布页面。

通过命令行部署应用

通过命令行部署应用时,您可以 SSH 登录 Kubernetes 集群的 master 节点 (Helm CLI 已自动安装并已配置Repository)进行操作,参见SSH 访问 Kubernetes 集群。您也可以在本地安装配置 kubectl 和 Helm CLI。

本示例以在本地安装配置 kubectl 和 Helm CLI 并部署 WordPress 和 Spark 应用为例进行说明。

安装配置 kubectl 和 Helm CLI

1. 在本地计算机上安装和配置 kubectl。

参见通过 kubectl 连接 Kubernetes 集群。

若要查看 Kubernetes 目标集群的信息,键入命令 kubectl cluster-info。

2. 在本地计算机上安装 Helm。

安装方法,参见 Install Helm。

3. 配置 Helm 的 Repository。这里我们使用了阿里云容器服务提供的 Charts 存储库。

```
helm init --client-only --stable-repo-url https://aliacs-app-catalog
.oss-cn-hangzhou.aliyuncs.com/charts/
helm repo add incubator https://aliacs-app-catalog.oss-cn-hangzhou.
aliyuncs.com/charts-incubator/
helm repo update
```

Helm 基础操作

• 若要查看在集群上安装的 Charts 列表,请键入:

helm list

或者缩写

helm ls

• 若要查看存储库配置,请键入:

```
helm repo list
```

• 若要查看或搜索存储库中的 Helm charts,请键入以下任一命令:

helm search helm search 存储库名称 #如 stable 或 incubator helm search chart名称 #如 wordpress 或 spark

• 若要更新 charts 列表以获取最新版本,请键入:

helm repo update

有关 Helm 使用的详细信息,请参阅 文档。

部署 WordPress

下面我们将利用 Helm,来部署一个 WordPress 博客网站。

输入如下命令。

```
helm install --name wordpress-test --set "persistence.enabled=false,
mariadb.persistence.enabled=false" stable/wordpress
```



目前阿里云 Kubernetes 服务中还没有开启块存储的 PersistentVolume 支持,所以在示例中禁止了

数据持久化。

得到如下的结果。

```
NAME: wordpress-test
LAST DEPLOYED: Mon Nov 20 19:01:55 2017
NAMESPACE: default
STATUS: DEPLOYED
...
```

利用如下命令查看 WordPress 的 release 和 service。

helm list kubectl get svc

利用如下命令查看 WordPress 相关的 Pod,并等待其状态变为 Running。

kubectl get pod

利用如下命令获得 WordPress 的访问地址。

```
echo http://$(kubectl get svc wordpress-test-wordpress -o jsonpath='{.
status.loadBalancer.ingress[0].ip}')
```

通过上面的 URL,可以在浏览器上看到熟悉的 WordPress 站点。

也可以根据 Charts 的说明,利用如下命令获得 WordPress 站点的管理员用户和密码。

echo Username: user

```
echo Password: $(kubectl get secret --namespace default wordpress-test
-wordpress -o jsonpath="{.data.wordpress-password}" | base64 --decode)
```

如需彻底删除 WordPress 应用,可输入如下命令。

helm delete --purge wordpress-test

通过 Helm 部署 Spark

下面我们将利用 Helm,来部署 Spark 以用于大数据处理。

输入如下命令。

helm install --name myspark stable/spark

得到如下的结果。

```
NAME: myspark
LAST DEPLOYED: Mon Nov 20 19:24:22 2017
NAMESPACE: default
STATUS: DEPLOYED
...
```

利用如下命令查看 Spark 的 release 和 service。

helm list kubectl get svc

利用如下命令查看 Spark 相关的 Pod,并等待其状态变为 Running。因为 Spark 的相关镜像体积较

大,所以拉取镜像需要一定的时间。

kubectl get pod

利用如下命令获得 Spark Web UI 的访问地址。

```
echo http://$(kubectl get svc myspark-webui -o jsonpath='{.status.
loadBalancer.ingress[0].ip}'):8080
```

通过上面的 URL,可以在浏览器上看到 Spark 的 Web UI,上面显示 worker 实例当前为 3 个。

接下来,我们将利用如下命令,使用 Helm 对 Spark 应用做升级,将 worker 实例数量从 3 个变更为 4 个。请注意参数名称是大小写敏感的。

helm upgrade myspark --set "Worker.Replicas=4" stable/spark

得到如下结果。

```
Release "myspark" has been upgraded. Happy Helming!
LAST DEPLOYED: Mon Nov 20 19:27:29 2017
NAMESPACE: default
```

STATUS: DEPLOYED

利用如下命令查看 Spark 新增的 Pod,并等待其状态变为 Running。

kubectl get pod

在浏览器上刷新 Spark 的 Web UI,可以看到此时 worker 数量已经变为 4 个。

如需彻底删除 Spark 应用,可输入如下命令。

helm delete --purge myspark

使用第三方的 Chart 存储库

您除了可以使用预置的阿里云的 Chart 存储库,也可以使用第三方的 Chart 存储库(前提是网络是可达的)。使用如下命令格式添加第三方 Chart 存储库。

helm repo add 存储库名 存储库URL helm repo update

关于 Helm 相关命令的说明,您可以参阅 Helm 文档

参考信息

Helm 催生了社区的发展壮大,越来越多的软件提供商,如 Bitnami 等公司,开始提供高质量的 Charts。您可以在 https://kubeapps.com/ 中寻找和发现已有的 Charts。

1.7.6 创建服务

Kubernetes Service 定义了这样一种抽象:一个 Pod 的逻辑分组,一种可以访问它们的策略,通常称为微服务。这一组 Pod 能够被 Service 访问到,通常是通过 Label Selector 来实现的。

在 Kubernetes 中, pod 虽然拥有独立的 IP, 但 pod 会快速地创建和删除,因此,通过 pod 直接 对外界提供服务不符合高可用的设计准则。通过 service 这个抽象,Service 能够解耦 frontend (前 端)和 backend (后端)的关联,frontend 不用关心 backend 的具体实现,从而实现松耦合的微 服务设计。

更多详细的原理,请参见 Kubernetes service。

前提条件

您已经成功创建一个 Kubernetes 集群,参见创建 Kubernetes 集群。

步骤1 创建 deployment

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,单击页面右上角的使用模板创建。



4. 选择所需的集群,命名空间,选择样例模板或自定义,然后单击创建。

集群 test-gpu (default (default)) 命名空间 default (す) 示明機版 Resource - basic Deployment (kind: Deployment) ④ (kind: Deployment) ③ (metadata: ④ (non:: ngin: deployment-basic ⑤ (app: nginx) ⑦ (specie: ⑧ (non:: nginx) ⑧ (specie: ⑧ (specie: ⑧ (specie: ⑧ (specie: ○ (s			
命名空间 default 示例提版 Resource - basic Deployment 初級 1 bpiVerSion: apps/Vibeta2 # for verSions before 1.8.0 use apps/vibeta1 截版 2 kindi: Deployment - basic 5 labels: a nome: nginx-deployment-basic 5 labels: a pp: nginx 7 spec: 8 replicas: 2 9 selector: 10 matchtabels: 11 app: nginx 15 app: nginx 15 spec: 16 spec: 17 a nodeSelector: 18 env: fest-team 	集群	test-gpu	¥
示例模板 Resource - basic Deployment ▼ 模板 1 hpiVersion: appS/vibeta2 # for versions before 1.6.0 use apps/vibeta1 2 kind: Deployment * 3 metadata: * 4 mame: nginx-deployment-basic * 5 - labels: * 6 app: nginx * 7 - spec: * 10 - metadata: * 11 - metadata: * 12 - metadata: * 13 - metadata: * 14 - mame: noise *	命名空间	default	v
<pre>1 bpiVersion: apps/vibeta2 # for versions before 1.8.0 use apps/vibeta1 2 kind: Deplayment 3 estadata: 4 name: nginx.deplayment-basic 5 labels: 6 app: nginx 7 spec: 7 spec: 7 spec: 8 replicas: 2 9 selector: 10 matchiabels: 11 app: nginx 12 template: 13 matchiabels: 13 matchiabels: 14 labels: 15 app: nginx 16 spec: 17 # nodSelector: 18 # nodSelector: 19 # nodSelector: 10 # nodSelector: 10 # nodSelector: 10 # nodSelector: 11 # nodSelector: 13 # nodSelector: 14 # nodSelector: 15 # nodSelector: 16 # env: kest-team</pre>	示例模版	Resource - basic Deployment	T
<pre>20</pre>	模版	<pre>implyersion: apps/vlbeta2 # for versions before 1.8.0 use apps/vlbeta1 kind: Deployment setadsta:</pre>	

本例中,示例模板是一个 nginx 的 deployment。

```
apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/
v1beta1
kind: Deployment
metadata:
   name: nginx-deployment-basic
   labels:
     app: nginx
 spec:
   replicas: 2
   selector:
     matchLabels:
       app: nginx
   template:
     metadata:
       labels:
         app: nginx
```

```
spec:
    containers:
    - name: nginx
    image: nginx:1.7.9
exactly <image_name:tags>
    ports:
    - containerPort: 80
##需要在服务中暴露该端口
```

replace it with your

5. 进入 Kubernetes Dashboard, 查看该 deployment 的运行状态。

<	三 工作负载 部署					十 创建
東部	部署					Ŧ
* 수도미	名称 ≑	标签	容難追	日创建 🕈	現象	
時な少存在地	nginx-deployment-basic	app: ngl	2/2	2018-04-18 17:20:01	nginx:1.7.9	1
角色						
存储类						
命名空间						
default 👻						
概况						
工作负载						
定时任务						
守护进程集						
部署						
任务						

步骤2 创建服务

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 服务,进入服务列表页面。
- 4. 选择所需的集群和命名空间,单击页面右上角的创建。

容器服务	服务列表						创建	刷新
Kubernetes Swarm								
▼ 集群	莱群 test-gpu ▼ 部名全间 default							
集群	名称	类型	创建时间	集群IP	内部端点	外部端点		操作
节点	kubernetes	ClusterIP	2018-04-17 12:01:58	172.21.0.1	kubernetes:443 TCP	-	更新	删除
存储	test	ClusterIP	2018-04-18 16:48:22	172.21.12.193	test:80 TCP	-	更新	删除
▼应用 1	tf-model-ack-tensorflow-serving	LoadBalancer	2018-04-18 09:53:34	172.21.6.179	tf-model-ack-tensorflow-serving:9090 TCP tf-model-ack-tensorflow-serving:30526 TCP	118.178.117.165:9090	更新	删除
服务 2	tt	ClusterIP	2018-04-18 15:01:59	172.21.6.180	tt:8080 TCP	-	更新	删除
发布								
配置项								

5. 在弹出的创建服务对话框中,进行配置。

创建服务		\times
名称:	nginx-svc	
类型:	负载均衡 ▼ 公网访问 ▼	
关联部署:	nginx-deployment-b; 🔻	
端口映射:	●添加	
	服务端口 容器端口 协议	
	8080 80 Tr 🔻 🗢	
	创建取	消

- 名称:输入服务的名称,本例中为 nginx-vc。
- 类型:选择服务类型,即服务访问的方式,包括:
 - 虚拟集群 IP:即 ClusterIP,指通过集群的内部 IP 暴露服务,选择该值,服务只能够在集群内部可以访问,这也是默认的 ServiceType。
 - 节点端口:即 NodePort,通过每个 Node 上的 IP 和静态端口(NodePort)暴露服务。NodePort 服务会路由到 ClusterIP 服务,这个 ClusterIP 服务会自动创建。通过请求

 NodeIP>:<NodePort>,可以从集群的外部访问一个 NodePort 服务。
 - 负载均衡:即LoadBalancer,指阿里云提供的负载均衡服务(SLB),可选择公网访问或
 内网访问。阿里云负载均衡服务可以路由到 NodePort 服务和 ClusterIP 服务。
- 关联部署:选择服务要绑定的后端对象,本例中是前面创建的 nginx-deployment-basic。若不进行关联部署,则不会创建相关的 Endpoints 对象,您可自己进行绑定,参见 *services-without-selectors*。
- 端口映射:添加服务端口和容器端口,容器端口需要与后端的 pod 中暴露的容器端口一致。
- 6. 单击创建, nginx-svc 服务出现在服务列表中。

容器服务		服务列表						创建	刷新
Kubernetes	Swarm								
▼ 集群		集群 test-gpu ▼ 命名空间 default	¥						
集群		名称	类型	创建时间	集群IP	内部端点	外部端点		操作
节点		kubernetes	ClusterIP	2018-04-17 12:01:58		kubernetes:443 TCP	-	更新	删除
存储		nginx-svc	LoadBalancer	2018-04-18 17:15:29	****	nginx-svc:80 TCP nginx-svc:31334 TCP	:80	更新	删除
* 应用 部署		test	ClusterIP	2018-04-18 16:48:22	10.000	test:80 TCP		更新	删除
服务		tf-model-ack-tensorflow-serving	LoadBalancer	2018-04-18 09:53:34	10.00	tf-model-ack-tensorflow-serving:9090 TCP tf-model-ack-tensorflow-serving:30526 TCP	5:9090	更新	删除
发布		tt	ClusterIP	2018-04-18 15:01:59		tt:8080 TCP		更新	│删除
配置项									

7. 您可查看服务的基本信息,在浏览器中访问 nginx-svc 的外部端点。

C 1 //spm=5176.2020520152.0.0.49	16ddLtgTUa	
	Welcome to nginx!	
	If you see this page, the nginx web server is successfully installed and working. Further configuration is required.	
	For online documentation and support please refer to <u>pginx.org</u> . Commercial support is available at <u>nginx.com</u> .	
	Thank you for using nginx.	

至此,您完成如何创建一个关联到后端的 deployment 的服务,最后成功访问 Nginx 的欢迎页面。

1.7.7 指定节点调度

您可通过为节点设置节点标签,然后通过配置 nodeSelector 对 pod 调度进行强制约束,将 pod 调度到指定的 Node 节点上。关于 nodeSelector 的详细实现原理,请参见 *nodeselector*。

出于业务场景需要,如您需要将管控服务部署到 Master 节点;或者将某些服务部署到具有 SSD 盘的机器上。您可以采用这种方式实现指定节点调度。

前提条件

您已经成功部署一个 Kubernetes 集群,参见创建 Kubernetes 集群。

步骤1 为节点添加标签

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 节点,进入节点列表页面。
- 3. 选择所需的集群,在页面右上角单击标签管理。

容器服	务	节点列表					周身	析 标签管理 集群伸转	窗 添加已有节点
Kubernet	es Swarm								
▼ 毎難	1	常见问题: ② 按量付费	转包年包月						
生計	•	集群: k8s-cluster ▼	3						
		IP地址	实例类型	角色	实例ID/名称	配置	操作系统	更新时间	操作
节点	J 🛃				the second se				
存储		10.000	阿里云ECS	Worker	k8s-for-cs-c98a7	ecs.sn1ne.large	centos_7_04	2018-04-23 11:58:25	监控

4. 在节点列表中,选择所需节点,然后单击添加标签。本例中选择一个 worker 节点。

标签管理		刷新
二 名称	IP地址	标签
cn-hangzhou.	.27	group:master 🕲
cn-hangzhou.	.26	group : master 🚳
🗑 cn-hangzhou.	.29	
Cn-hangzhou.	.28	group : master 🚳
□ 汤如杨蓥 2		

5. 在弹出的添加标签对话框中,输入标签的名称和值,然后单击确定。

添加		×
名称	group	
值	worker	
	确定	关闭

您可以在标签管理页面,看到该节点具有 group:worker标签。

标签管理		
	IP地址	标签
G cr2	.27	group : master 🔕
2ba	-26	group : master 🔕
18cn	.29	group:worker
9nrg	28	group : master 🔕
■ 添加标签		

您也可以通过命令 kubectl label nodes <node-name> <label-key>=<label-value> 为节点添加标签。

步骤2 将 pod 部署到指定节点

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,进入部署列表页面。

3. 单击页面右上角的使用模板创建。

容器服务		部署列表				使用镜像创建使用模板创建刷新
Kubernetes	Swarm	集群 ken durter ▼ 余冬奈间 default				2
▼ 集群		Ment Kos-cluster • Mena zini ueraulu				
集群		名称	标签	容器组数量	创建时间	操作
节点		nginx-deployment-basic	app:nginx	2	2018-04-23 13:57:50	洋情 更新 删除
存储 ▼ 应用	1	test-mariadb	app:test-mariadb chart:mariadb-0.7.0 release:test heritage:Tiller	1	2018-04-23 14:12:26	洋摘 更新 翻除
部署 容器组	•	test-wordpress	app:test-wordpress chart:wordpress-0.6.13 release:test heritage:Tiller	1	2018-04-23 14:12:26	洋橋 更新 劃除

- 4. 对模板进行相关配置,部署一个 pod,完成配置后,单击创建。
 - 集群:选择所需的集群。
 - 命名空间:选择资源对象所属的命名空间,本例中是 default。
 - 示例模板:本示例选择自定义模板。

集群	k8s-cluster
命名空间	default 🔹
示例模版	自定义 ▼
模版	<pre>1 apiVersion: v1 2 kind: Pod 3 * metadata: 4 * labels: 5 name: hello-pod 6 name: hello-pod 7 * spec: 8 * containers: 9 * - image: nginx 10 * imagePullPolicy: IfNotPresent 11 * name: hello-pod 12 * ports: 13 * - containerPort: 8080 14 * protocol: TCP 15 * resources: {} 16 * securityContext: 17 * capabilities: {} 18 * privileged: false 19 * terminationMessagePath: /dev/termination-log 19 * terminationMessagePath: /dev/termination-log 20 * dnsPolicy: ClusterFirst 21 * restartPolicy: Always 22 * modeSelector: 23 * modeSelector: 24 * modeSelector: 24</pre>

本示例的编排模板如下。

```
apiVersion: v1
kind: Pod
metadata:
labels:
name: hello-pod
name: hello-pod
```

```
spec:
  containers:
    - image: nginx
      imagePullPolicy: IfNotPresent
      name: hello-pod
      ports:
         - containerPort: 8080
          protocol: TCP
      resources: {}
      securityContext:
        capabilities: {}
        privileged: false
      terminationMessagePath: /dev/termination-log
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  nodeSelector:
                                             ##注意与前面配置的节点标签
    group: worker
一致
status: {}
```

5. 单击创建 后,会提示部署状态信息。成功后,单击Kubernetes 控制台前往控制台查看部署状

态。

<	☰ 概況						+ 1	1) (P
(T. Fr	容器组						Ŧ	
命名空间	名称 \$	节点	状态 🗢	已重度	巴创建 ◆ CPU(核)	内存(字节)		
节点	S hello-pod	cn-hangzhou.i-bp1idqJ54m9at16du8cn	Running	0	2018-04-25 13:59:28	1.438 MI	≡ :	
持久化存储器	test-mariadb-9bb8f87dd-v9n6f	cn-hangzhou.i-bp1idqj54m9at16du8cn	Running	0	2018-04-23 14:12:26 0.002	138.035 Mi	≡ :	
角色	est-wordpress-6945bc9556-mpdxn	cn-hangzhou.i-bp1idqJ54m9at16du8cn	Running	0	2018-04-23 14:12:26 0.004	171.188 Mi	≡ :	
存储类	ginx-deployment-basic-6c54bd5869-l8trh	cn-hangzhou.i-bp1idq]54m9at16du8cn	Running	0	2018-04-23 13:57:50 0	1.383 Mi	≡ :	
命名空间 default ▼	ginx-deployment-basic-6c54bd5869-mw65q	cn-hangzhou.i-bp1idqj54m9at16du8cn	Running	0	2018-04-23 13:57:50 0	1.367 Mi	≡ :	
809								-
	副本集						7	

6. 您可单击 pod 名称,进入详情页,了解 pod 详情。

您可看到 pod 的标签、所处的节点 ID 等信息,表明该 pod 已经成功部署到具有 group: worker 标签的指定节点上。



1.7.8 服务伸缩

应用创建后,您可以根据自己的需求来进行服务扩容或缩容。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的控制台,进入 Kubernetes Dashboard。

容器服务		集群列表				您最多可以创建 5 个集群,	每个集群最多可以添加 20 个节点	刷新	到建 Kubernetes 集	群 👻
Kubernetes	Swarm									
▼集群		常见问题: Ø 如何创建集群 Ø 扩容和缩容	巣群 🔗 通	挝 kubectl 连接 Kubernetes	;集群 🔗 通过命	令管理应用				
集群		名称 ▼								
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本			2 最作
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理	查看日志 集群伸缩	控制台 更多▼

- 4. 在 Kubernetes Dashboard 中,单击左侧导航栏中的部署,查看部署的 deployment。
- 5. 选择所需的 deployment,单击右侧的操作图标并单击伸缩。

<	三 工作负载 部署							+ 89
集群 命名空间	CPU 使用率			内存使用率 🏮				
节点	0.146 0.130			172 Mi 153 Mi				
持久化存储卷	£ 0.096			₽ ^{114 Mi}				
角色存储类	0.085			在 76.3 Mi 会 38.1 Mi				
命名空间	09:31 09:33	09:36 09:40	D0:43 09:45	09:31	09:33	09:38 09:40	c	09:43 09:45
default 👻		et(i)				时间		
概范	部署							Ŧ
工作负载	名称 辛	标签	容器组		已创建 ≑	镜像		
FF 1	or nginx	app: ngi	1 / 1		2018-01-29 17:54:14	nginx docker.io/istio/proxy_d	debug:0.4.0	2 :
任务	wordpress	app: wordpre	2 / 1		2018-01-29 16:42:37	wordpress docker.lo/istio/proxy_d	debug:0.4.0	伸缩
容器组								删除
副本控制器								查看/编辑 YAN
有状态副本集								

6. 在弹出的对话框中,将所需容器组数修改为2,并单击确定。

此操作会扩容一个 Pod,将副本数升到 2。

Scale a 部署		
资源 wordpress 当前状态: 1 已创	将更新以反映 建, 2 需要	所需数量
所需容器组数量 2		
	取消	确完
	-10/13	HUXE

后续操作

您可以通过左侧的图标查看每个 Kubernetes 对象的状态。 🕥 表示对象仍然处于部署状态。 🌏

表示对象已经完成部署。

应用部署完成后,您可以单击某个部署的名称查看正在运行的 Web 服务的详细信息。您可以查看 部署中的副本集以及这些副本集所使用的 CPU 和 Memory 资源的相关信息。您也可以单击查看日 志 查看容器的日志。



如果看不到资源,请耐心等待几分钟。



1.7.9 查看服务

背景信息

您在创建应用时,如果配置了外部服务,除了运行容器,Kubernetes Dashboard 还会创建外部 Service,用于预配负载均衡器,以便将流量引入到集群中的容器。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 服务,进入服务列表页面。
- 3. 您可以选择所需的集群和命名空间,查看部署的服务。

您可查看服务的名称、类型、创建时间、集群 IP 以及外部端点等信息。本例中您可看到分配给服务的外部端点(IP 地址)。您可以单击该 IP 地址访问 wordpress 博客应用。

容器服务		服务列表	3					创建剧新
Kubernetes	Swarm							
▼ 集群		集群 k8s-cluster ▼	命名空间 default	•				
集群		名称	类型	创建时间	集群IP	内部端点	外部跳点	操作
节点		kubernetes	ClusterIP	2018-04-23 11:48:29	25.5.5	kubernetes:443 TCP		更新 删除
存储		test-mariadb	ClusterIP	2018-04-23 14:12:26	1000	test-mariadb:3306 TCP	-	更新 删除
▼ 应用 ●		test-wordpress	LoadBalancer	2018-04-23 14:12:26	111.44	test-wordpress:80 TCP test-wordpress:31536 TCP test-wordpress:443 TCP test-wordpress:32142 TCP	:80 :443	更新 删除
服务	2							

您也可进入集群的 Kubernetes Dashboard。在左侧导航栏中单击服务,查看所有服务。

容器服务

1.7.10 更新服务

您可以通过容器服务管理控制台的服务列表页面或者 Kubernetes Dashboard 变更服务的配置。

通过容器服务控制台服务列表页面

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 服务,进入服务列表页面。
- 4. 选择集群和命名空间,选择所需的服务(本示例中选择 nginx-svc),单击右侧的更新。

容器服务	服务列表						创建	刷新
Kubernetes Swarm								
▼ 集群	集群 test-gpu V 茚名空间 default	<u> </u>						
集群	名称	类型	创建时间	集群IP	内部端点	外部端点		操作
节点	kubernetes	ClusterIP	2018-04-19 16:57:11		kubernetes:443 TCP		更新	删除
	nginx-default-svc	LoadBalancer	2018-04-19 15:06:19	-	nginx-default-svc:80 TCP nginx-default-svc:32194 TCP	:80	更新	│删除
部署	nginx-svc	LoadBalancer	2018-04-18 17:15:29	-	nginx-svc:80 TCP nginx-svc:31000 TCP	:80	4 更新	删除
服务 2	ps-0	ClusterIP	2018-04-19 16:19:33	None	ps-0:9000 TCP	-	更新	删除

5. 在弹出的更新对话框中,在模板中进行修改,如将 nodePort 修改为 31000,然后单击确定。



6. 在服务列表中,查看服务变化的情况。本例中, nodePort 发生变化, 如下。

服务列表						创建	刷新
集群 test-gpu ▼ 命名空间 defaul	t 🔻						
名称	类型	创建时间	集群IP	内部端点	外部端点		操作
kubernetes	ClusterIP	2018-04-17 12:01:58	10.00	kubernetes:443 TCP		更新	删除
nginx-svc	LoadBalancer	2018-04-18 17:15:29		nginx-svc:80 TCP nginx-svc:31000 TCP	2:80	更新	一删除

通过 Kubernetes Dashboard

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 4. 选择所需的集群并单击右侧的控制台,进入 Kubernetes Dashboard。

容器服务		集群列表				您最多可以创建 5 个集群	,每个集群最多可以添加 20 个节点 刷	新 创建 Kubernetes 集群 🗸
Kubernetes	Swarm		* 0.50+	ladaad XHX Kabaaataa #				
▼ 集群		第50问题: O XUPID建築群 O 1 各和增益采获	± 🛇 10010	Kubecu <u>jeg</u> Kubernetes ak	84 Q 1@17 m 4.8	理应用		
集群		名称 ▼						
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	2 🗉
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理 查看日志 控制台 集群伸缩 更多▼

- 5. 在 Kubernetes Dashboard 中,选择所需的命名空间,单击左侧导航栏中的服务。
- 6. 选择所需的服务,单击右侧的操作图标并单击查看/编辑YAML。

服务发现与负载均衡	服务					十创建
服务						Ŧ
名称 🗘	标签	集群 IP	内部端点	外部端点	已创建 ≑	
kubernetes	component: apiser provider: kubernet	100.000	kubernetes:443 TCP		2018-04-19 16:57:11	:
oginx-default-svc	-	242	nginx-default-svc:80 TCP nginx-default-svc:32194 TCP	:80 🖾	2018-04-19 15:06:19	:
orginx-svc		5	nginx-svc:80 TCP nginx-svc:31000 TCP	:80 🖾	2018-04-18 1 删除	
📀 test	-	193	test:80 TCP		2018-04-18 1 查看/約	扁辑 YAML

7. 在弹出的更新对话框中,修改配置信息,如将 nodePort 修改为 31000,然后单击更新。

修改的	Service
; = =	=
1 -	r.
1 *	p
2	NING : Service ,
J	apiversion . Vi ,
5	"name": "nainy-syc"
6	"name:n=cm=", "default"
7	"celflink": "/ani/v1/namespaces/default/services/nginy-svc"
8	"uid", "09d3e50d-42e9-11e8-add-00163e082e5e"
9	"resourceVersion": "226655".
10	"creationTimestamp": "2018-04-18T09:15:297"
11	}.
12 -	"spec": {
13 -	"ports": [
14 -	{
15	"protocol": "TCP",
16	"port": 80,
17	"targetPort": 80,
18	"nodePort": 31000
19	}
20	
21 *	"selector": {
22	"app": "nginx"
23	},
24	"clusterIP": "172.21.2.55",
25	"type": "LoadBalancer",
26	"sessionAffinity": "None",
27	"externalTrafficPolicy": "Cluster"
28	},
29 ×	"status": {
30 -	"loadBalancer": {
31 *	"ingress": [
	田澤 有利 声蛇
	· · · · · · · · · · · · · · · · · · ·

1.7.11 查看容器

您可以通过容器服务管理控制台的容器组页面或者 Kubernetes Dashboard 查看 Kubernetes 集群的容器组页面。

通过容器组页面进行查看

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 容器组,进入容器组页面。
- 4. 选择所需的集群和命名空间,选择所需的容器组,单击右侧的详情。



您可对容器组进行更新和删除操作,对于通过部署(deployment)创建的容器组,建议您通过 deployment 进行管理。

容器服务	容器组							刷新
Kubernetes Swarm								
▼ 集群	集群 kBs-cluster ▼ 命名空间 default ▼ 3							
集群	名称	状态	Pod IP	节点	创建时间			
节点	hello-pod	Running	172.20.3.24	172.16.166.29	2018-04-25 13:59:28	详情	更新	删除
存储	nginx-deployment-basic-6c54bd5869-l8trh	Running	172.20.3.7	172.16.166.29	2018-04-23 13:57:50	4 详情	更新	删除
▼ 应用 1	nginx-deployment-basic-6c54bd5869-mw65q	Running	172.20.3.8	172.16.166.29	2018-04-23 13:57:50	详情	更新	删除
部署								
容麗组 2								

5. 进入容器组的详情页,您可查看该容器组的详情信息。

容器组 - nginx-deployment-basic-6c54bd5869-l8trh					刷新	
基本信息						
名称: nginx-deployment-basic-6c54bd5869-l8trh		命名空间: default				
状态: Running		创建时间: 2018-04-23 13:57:50				
节点: cn-hangzhou.i-bp1idqj54m9at16du8cn		Pod IP: 172.20.3.7				
标签: app: nginx pod-template-hash: 2710681425						
容器						
名称	镜像			端口		
> nginx	nginx:1.7.9			TCP 80		
创建者						
名称: nginx-deployment-basic-6c54bd5869		義型: ReplicaSet				
存储						
名称	类型		详情			
default-token-2dxth	secret		defaultMode: 420 secretName: default-token-2dxth			

通过 Kubernetes Dashboard 查看容器组

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 4. 选择所需的集群并单击右侧的控制台,进入 Kubernetes Dashboard。

容器服务		集群列表				您最多可以创建 5 个集群	詳, 每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes 集群 🛛 👻
Kubernetes	Swarm	常见问题: & 如何创建集群 & 扩容和缩容集制	詳 🔗 通过 ku	ubectl 连接 Kubernetes 集	群 🔗 通过命令管	理应用			
* ^{集群}		名称 ▼							
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本		2 🗉
存储		k8s-cluster	华东1 1	盝拟专有网络 /pc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理	· 查看日志 控制台 集群伸缩 更多▼

5. 单击左侧导航栏中的容器组,查看集群中的容器组。

或者,您可以单击左侧导航栏中的**服务**,选择所需的服务并单击服务的名称,查看该服务下的容器组。

集群	服务						Ŧ
命名空间							
节点	名称 🜲	标签	集群 IP	内部入口	外部入口	已创建 ≑	
持久化存储卷	🛛 ap	app: ap	172.19.13.68	ap:80 TCP ap:31214 TCP	120.55.245.149:80	8 分钟	:
存储类	or nginx	app: nginx	172.19.8.16	nginx:80 TCP nginx:30817 TCP	120.55.150.31:80	9分钟	:
命名空间	📀 app	арр: арр	172.19.8.149	app:80 TCP app:0 TCP		26 分钟	:
default -	Mttp-svc	app: http-svc	172.19.0.140	http-svc:80 TCP http-svc:30301 TCP	-	4天	:
Overview	V kubernetes	component: apiserver provider: kubernetes	172.19.0.1	kubernetes:443 TCP kubernetes:0 TCP		4天	÷
工作负载							
守护进程集							
部署							
任务							
容器组							
副本集							
副本控制器							
有状态副本集							
服务发现与负载均衡							
访问权							
服务							
配置与存储							

6. 您可以通过左侧的图标查看每个 Kubernetes 对象的状态。 🕖 表示对象仍然处于部署状

态。 🕜 表示	对象已经完成部署	0								
集群 命名空间	CPU 使用率			内存使	用率 🔒					
 市点 持久化存储器 角色 存価拠 	0.001 0.001 0.0008 0.0005 0.0005			139 Mi 124 Mi 伊 93.0 Mi 世 62.0 Mi 昭 31.0 Mi					-	
参名空间 default →	0 22:54 22:56	23:00 23:03 时间	23:06 23:08	22:	54 22	56	23:00 时间	23:03	23:06	23:08
Overview	容器组									Ŧ
工作负载	名称 🜲	节点	状态 🗣	已重启	已创建 🗢	CPU (核)		内存 (字节)		
守护进程集	ap-b6969b5f7-4fhz8	cn-hangzhou.i- bp1izzt8ihoeootkfq1a	等待中: ContainerCreatinç	0	7 秒				₽	:
任务	ap-b6969b5f7-zcddx	cn-hangzhou.i- bp1izzt8ihoeootkfq19	Running	0	7 秒	-		-	≡	:
容積 <u>知</u> 副本集	nginx-686c8db8c9-t5w2d	cn-hangzhou.i- bp1izzt8ihoeootkfq1b	Running	0	1分钟			2.648 Mi	=	:
副本控制器 有状态副本集	oginx-686c8db8c9-2fj47	cn-hangzhou.i- bp1izzt8ihoeootkfq1a	Running	0	1 分钟			2.086 Mi	≡	:

7. 选择所需的容器组,单击容器组的名称,您可以查看容器组的详细信息以及容器组使用的 CPU 和 Memory。



8. 选择所需的容器组,单击右侧的查看日志查看容器的日志信息。

9. 您可以单击右侧的删除 删除该容器组。



1.7.12 变更容器配置

您可以通过容器服务管理控制台变更容器的配置。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 3. 选择所需的集群并单击右侧的控制台,进入 Kubernetes Dashboard。

容器服务	Swarm	集群列表				您最多可以创建 5 个集群	詳,每个集群最多可以添加 20 个节点	刷新 创建 Kubernetes 集群 →
▼ 集群	Swaini	常见问题: Ø 如何创建集群 Ø 扩容和缩容集制	詳 🖉 通过	t kubectl 连接 Kubernetes 集	群 🔗 通过命令管	理应用		
		名称 ▼						
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	2 🗉
存储		kBs-cluster	华东1	虛拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理 查看日志 控制台 集群伸缩 更多▼

- 4. 在 Kubernetes Dashboard 中,单击左侧导航栏中的容器组。
- 5. 选择所需的容器组,单击右侧的操作图标并单击查看/编辑YAML。



6. 更新服务的配置并单击更新。

修改容	器组
∋ ≓	powered by ace
1 * {	
2	"kind": "Pod",
3	"apiVersion": "v1",
4 =	"metadata": {
5	"name": "wordpress-6bbb7ffc-1d6gk",
6	"generateName": "wordpress-6bbb7ffc-",
7	"namespace": "default",
8	"selfLink": "/api/v1/namespaces/default/pods/wordpress-6bbb7ffc-ld6gk",
9	"uid": "673e913b-bd8b-11e7-9b48-00163e0fd866",
10	"resourceVersion": "887886",
11	"creationTimestamp": "2017-10-30T16:00:08Z",
12 *	"labels": {
13	"app": "wordpress",
14	"pod-template-hash": "26663997"
15	},
16 -	"annotations": {
17	"kubernetes.io/created-by": "{\"kind\":\"SerializedReference\"
	<pre>,\"apiVersion\":\"v1\",\"reference\":{\"kind\":\"ReplicaSet\"</pre>
	<pre>\"namespace\":\"default\",\"name\":\"wordpress-6bbb7ffc\",\"uid\"</pre>
	:\"798b110e-bd89-11e7-9b48-00163e0fd866\",\"apiVersion\":\"extensions\"
	<pre>\"resourceVersion\":\"887864\"}}\n"</pre>
18	}.
19 -	"ownerReferences": [
20 -	{
21	"apiVersion": "extensions/v1beta1",
22	"kind": "ReplicaSet",
23	"name": "wordpress-6bbb7ffc",
24	"uid": "798b110e-bd89-11e7-9b48-00163e0fd866",
25	"controller": true,
26	"blockOwnerDeletion": true
	取消のないと思いていた。この時代は、「「「」」の「「」」では、「」」の「「」」では、「」」の「」」では、「」」の「」」では、「」」の「」」では、「」」の「」」の「」」の「」」の「」」の「」」の「」」の「」」の「」」の「」」

1.8 命名空间管理

1.8.1 创建命名空间

前提条件

您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。

背景信息

在 Kubernetes 集群中,您可使用 Namespaces (命名空间)功能创建多个虚拟的空间,在集群 用户数量较多时,多个命名空间可以有效划分工作区间,将集群资源划分为多个用途,并通过 *resource-quotas* 对命名空间的资源进行分配。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏的集群 > 命名空间,进入命名空间列表页面。
- 3. 选择所需的集群,单击页面右上角的创建。

容器服务	命名空间			刷新创建
Kubernetes Swarm				
概览	集群 test-k8s v 3			4
▼ 集群 1	名称 标签	状态	创建时间	操作
集群	default	● 就緒	2018-05-28 17:08:35	编辑 删除
节点	kube-public	● 就绪	2018-05-28 17:08:39	编辑 制除
存储	kube-system	● 就绪	2018-05-28 17:08:36	编辑 🗌 删除
命名空间				

4. 在弹出的对话框中, 配置命名空间。

创建命名空间				\times
名称	test 长度为 1-63 个字符	于, 只能包含数字、字	母、和"-",且首尾只能是字母或数字	
标签	变量名称	变量值	操作	
	env	test	编辑 删除	
	名称	值	添加	
			确定	取消

- 名称:您需要设置命名空间的名称,本例中为 test。长度为 1-63 个字符,只能包含数字、字母、和"-",且首尾只能是字母或数字
- 标签:您可为命名空间添加多个标签。标签用于标识该命名空间的特点,如标识该命名空间 用于测试环境。

您可输入变量名称和变量值,单击右侧的添加,为命名空间新增一个标签。

- 5. 完成命名空间设置后,单击确定。
- 6. 返回命名空间列表,您可看到 test 命名空间出现在列表中,命名空间成功创建。

命名空间				刷新创建
集群 test-k8s ▼				
名称	标签	状态	创建时间	操作
default		● 就绪	2018-05-28 17:08:35	编辑 删除
kube-public		● 就绪	2018-05-28 17:08:39	编辑 🗌 删除
kube-system		● 就绪	2018-05-28 17:08:36	编辑 一删除
test	env: test	● 就绪	2018-05-31 10:46:49	编辑 删除
	命名空间 集群 test-kBs ▼ 名称 default kube-system test	☆名空间 集群 test-k8s ▼ 名称 标签 default kube-public kube-system test serve test	命名空间 集群 test-k8g ▼	命名空间 葉酢 test-k8s ▼ 名称 标签 修建时间 default ● 就编 2018-05-28 17:08:35 kube-public ● 就编 2018-05-28 17:08:39 kube-system ● 就编 2018-05-28 17:08:36 test SNY/LISE ● 数编 2018-05-28 17:08:36

1.8.2 为命名空间配置资源额度

前提条件

- 您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。
- 您已成功创建一个示例的命名空间test,参见创建命名空间。
- 连接到 Master 节点 SSH 连接地址,参见SSH访问Kubernetes集群。

背景信息

在默认的情况下,运行中的 Pod 可以无限制的使用 Node 上的 CPU 和内存,这意味着任意一个 Pod 都可以无节制地使用集群的计算资源,某个命名空间的 pod 可能会耗尽集群的资源。

命名空间的一个重要的作用是充当一个虚拟的集群,用于多种工作用途,满足多用户的使用需求,因此,为命名空间配置资源额度是一种最佳实践。

您可为命名空间配置包括 CPU、内存、Pod 数量等资源的额度,更多信息请参见 Resource Quotas。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏的应用 > 部署,然后单击右上角的使用模板创建。

容器服务		部署列表	
Kubernetes	Swarm		
概览		集群 test-k8s v	▼ 命名空间 default ▼
▼ 集群		名称	标签
集群			
节点			
存储			
命名空间			
▼ 应用	1		
部署			

3. 在使用模板创建页面中,选择所需的集群和示例命名空间 **test**,使用自定义或示例模板 Resource-ResourceQuota。

集群	test-k8s
命名空间	test
示例模版	Resource - ResourceQuota
模版	<pre>1 apiVersion: v1 2 kind: ResourceQuota # restrict resource quota for c roller, pods, service, secret, configmap 3 metadata: name: quota 5 # namespace: users-namespace # specify your namesp 6 spec: 7 hard: 8 cpu: "2" # adjust limits of cpu for your namesp 9 memory: 4Gi # adjust memory upper limits for yo 10 requests.storage: 10246 # adjust request of sto 11 persistentvolumeclaims: "50" # adjust number of 12 pods: "50" #adjust number of Pod in your namesp 13 replicationcontrollers: "10" # adjust number of 14 services: "10" # adjust number of service for y 15 secrets: "100" # adjust number of configmap : 16 configmaps: "100" # adjust number of configmap : 17 configmap = 100" # adjust number of confi</pre>

您需要根据自己集群的资源情况,以及对该命名空间的资源规划,对 ResourceQuota 模板进行 配置,本例的模板如下。

```
apiVersion: v1
kind: ResourceQuota # restrict resource quota for cpu, memory
, storage, pvc, replicationcontroller, pods, service, secret,
configmap
metadata:
  name: quota
# namespace: users-namespace # specify your namespace to apply
resource quota
spec:
  hard:
     cpu: "2" # adjust limits of cpu for your namespace
     memory: 4Gi # adjust memory upper limits for your namesapce
    requests.storage: 1024G # adjust request of storage size for
your namespace
    persistentvolumeclaims: "50" # adjust number of pvc for your
namespace
    pods: "50" #adjust number of Pod in your namespace
    replicationcontrollers: "10" # adjust number of Replicatio
nController in your namespace
     services: "10" # adjust number of service for your namespace
     secrets: "100" # adjust number of secrets for your namespace
     configmaps: "100" # adjust number of configmap for your
namespace
```

4. 您已成功为该命名空间配置了资源额度,连接到 Master 节点 SSH 连接地址,执行以下命

令,查看该命名空间的资源限额和使用情况。

Ħ	kubectl describe quota Name:	quota quota	namespace=test
	Resource	Used	Hard
	configmaps	0	100
	сри	0	2
	memory	0	4Gi
	persistentvolumeclaims	0	50
	pods	0	50
	replicationcontrollers	0	10
	requests.storage	0	1024G
	secrets	1	100
	services	0	10

1.8.3 编辑命名空间

前提条件

- 您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。
- 您已成功创建一个示例的命名空间test,参见创建命名空间。

背景信息

您可对命名空间进行编辑,对命名空间的标签进行增、删、改等操作。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏的集群 > 命名空间,进入命名空间列表页面。
- 3. 选择所需的集群,选择所需的命名空间,单击右侧的编辑。

容器服务	하名空间			刷新创建				
Kubernetes Swarm								
概览	集群 test-k8s V 3	k≅ test-k8s ▼						
▼ 康群 1	名称 标签	状态	创建时间	操作				
集群	default	● 就绪	2018-05-28 17:08:35	编辑 一 删除				
京	kube-public	● 就绪	2018-05-28 17:08:39	编辑 🗌 删除				
存储	kube-system	● 就绪	2018-05-28 17:08:36	编辑 删除				
命名空间 2 ▼ 应用	test env: test	🛑 就绪	2018-05-31 10:46:49	4 编辑 删除				

4. 在弹出的对话框中,对命名空间的标签进行修改,如将标签修改为env:test-V2

编辑命名空间				\times
名称	test 长度为 1-63 个字符 , J	录能包含数字、字母、和"-",且首即	尾只能是字母或数字	
标签	变量名称 env	变量值 test-V2	操作 <u>保存</u> 删除	
	名称	值	添加	
			确定取消	肖

5. 单击确定,返回命名空间列表,该命名空间的标签发生变化。

容器服务		命名空间				刷新创建
Kubernetes	Swarm					
概览		集群 test-k8s ▼				
▼ 集群		名称	标签	状态	创建时间	操作
集群		default		● 就緒	2018-05-28 17:08:35	编辑 删除
节点		kube-public		● 就绪	2018-05-28 17:08:39	编辑 删除
存储		kube-system		● 就绪	2018-05-28 17:08:36	编辑 删除
命名空间		test	env: test-V2	● 就绪	2018-05-31 10:46:49	编辑 删除
▼ 应用						

1.8.4 删除命名空间

您可删除不再需要的命名空间。

前提条件

- 您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。
- 您已成功创建一个示例的命名空间test,参见创建命名空间。

背景信息

📕 说明:

删除命名空间会导致其下所有的资源对象一起被删除,请慎重操作。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏的集群 > 命名空间,进入命名空间列表页面。
- 3. 选择所需的集群,选择所需的命名空间,单击右侧的删除。

容器服务	命名空间	
Kubernetes Sw	arm	
概览	集群 test-k8s	√ 3
▼ 集群 1	名称	标签
集群	default	
节点	kube-public	
存储	kube-system	
命名空间 2	test	env: test-V2
▼ 应用		

4. 在弹出的对话框中,单击确定。

提示		
	确定要删除 test 吗?	
		确定

5. 返回命名空间列表,您可看到该命名空间已被删除,其下的资源对象也会被删除。

1.9 配置项

1.9.1 创建配置项

在容器服务管理控制台上,您可以通过配置项菜单或使用模板来创建配置项。

通过配置项创建

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 配置项,进入配置文件列表。
- 3. 在配置文件列表页面,选择需要创建配置项的集群,然后单击创建配置项。

容器服务	配置文件列表			刷新创建配置项
Kubernetes Swarm				
▼ 集群	集群: k8s-cluster ▼ 3			
歩 群	配置文件名称	命名空间	创建时间	操作
节点	test-mariadb	default	2018-04-23 14:12:26	删除 修改
存储	cluster-info	kube-public	2018-04-23 11:48:31	删除 惨改
▼ 应用 1	extension-apiserver-authentication	kube-system	2018-04-23 11:48:29	删除 修改
部署	ingress-controller-leader-nginx	kube-system	2018-04-23 11:58:42	删除 修改
服务	kube-flannel-cfg	kube-system	2018-04-23 11:48:39	删除 修改
发布	kube-proxy-master	kube-system	2018-04-23 11:48:52	删除 修改
配置项 2 5	kube-proxy-worker	kube-system	2018-04-23 11:48:52	删除 修改

4. 填写配置文件的信息并单击确定。

- 命名空间:选择该配置项所属的命名空间。配置项(ConfigMap)是 kubernetes 资源对象,需要作用于命名空间。
- 配置文件名:指定配置项的文件名,名称可以包含小写字母、数字、连字符(-)或者点号(.),名称不能为空。其他资源对象需要引用配置文件名来获取配置信息。
- 配置项:填写变量名称和变量值后,需要单击右侧的添加。您也可以单击编辑配置文件 在弹出的对话框里编写配置项并单击确定。

配置文件				
*命名空间:	default			٣
* 配置文件名 称:	test-config 名称只能包含小写字母数字	" , '-'或者'.' , 名称不能为空。		
配置项:	变量名称	变量值		操作
	enemies	aliens	编辑	删除
	lives	3	编辑	删除
	名称	值		添加
	变量名不能重复 , 变量名和	1变量值不能为空。		
	编辑配置文件			
		确定 取消		

本示例中设置了 enemies 和 lives 变量,分别用于传递 aliens 和 3 这两个参数。

YAML格式的配置文件 X
<pre>1 - data: 2 enemies: aliens 3 lives: '3' 4 - metadata: 5 name: test-config 6 namespace: default 7</pre>
配置文件必须符合YAML格式
确定取消

5. 单击确定后,您可以在配置文件列表中看到 test-config 配置文件。

配置文件列表			刷新创建配置	项
集群: k8s-test ▼				
配置文件名称	命名空间	创建时间		操作
test-config	default	2018-02-05 11:03:42	删除 何	险

使用模板创建

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,进入部署列表。
- 4. 单击页面右上角的使用模板创建。

容器服务		部署列表			使用	销像创建 使用模板创建 刷新
Kubernetes	Swarm					
▼ 集群		集群 k8s-cluster	▼ 命名空间 default ·	7		
生群		名称	标签	容器组数量	创建时间	提作
节点				① 没有	查询到符合条件的记录	
存储						
▼ <u> 应用</u> 部署	1					
服务						
发布						
配置项						

- 5. 在使用模版部署的页面,设置配置文件的信息并单击部署。
 - 集群:选择需要创建配置项的集群。
 - 命名空间:选择该配置项所属的命名空间。配置项(ConfigMap)是 kubernetes 资源对象,需要作用于命名空间。
 - 示例模板:您可以选择自定义,根据 Kubernetes yaml 语法规则编写 ConfigMap 文件,或者选择示例模板 resource-ConfigMap。该示例模板的 ConfigMap 名称为 aliyun-config,包含两个变量文件 game.properties 和 ui.properties,您可以在此基础上进行修改,然后单击部署。

仅支持 Kubernetes	5 版本 1.8.4 及以上	的集群。对于 1.8.1 版本的集群,您可以在集群列表中进行"集群升级"操作	
	集群	k8s-cluster	
1	命名空间	default v	
	示例模版	Resource - ConfigMap	
	模版	<pre>1 # kubectl create configmap game-configfrom-file=docs/user-guide/configmap/kubectl 2 apiversion: v1 3 kind: ConfigMap 4 - metadata: 5 - mame: aliyun-config 6 - data: 7 - # replace your filename of properties configration and contents here. 8 - game.properties: 9 - enemies-aliens 11 - enemies.cheat.true 12 - enemies.cheat.true 13 - scret.code.ploxed=true 14 - enemies.cheat.true 15 - scret.code.aliowed=true 16 - eliowed=true 17 - color.pdod=yurple 18 - color.bddv=llow 19 - aliow.fectmode=true 20 - how.nice.to.lowef=farlyMtice 21 - special.env: env_value 21 - special.env: env_value 22 - special.env: env_value 23 - scret.code.ploxed=true 24 - special.env: env_value 25 - special.env: env_value 26 - special.env: env_value 27 - special.env: env_value 28 - special.env: env_value 29 - special.env: env_value 20 - special.env: env_value 20 - special.env: env_value 21 - special.env: env_value 23 - special.env: env_value 24 - special.env: env_value 25 - special.env: env_value 26 - special.env: env_value 27 - special.env: env_value 28 - special.env: env_value 29 - special.env: env_value 20 - special.env: env_value 20 - special.env: env_value 21 - special.env: env_value 22 - special.env: env_value 23 - special.env: env_value 24 - special.env: env_value 25 - special.env: env_value 26 - special.env: env_value 27 - special.env: env_value 28 - special.env: env_value 29 - special.env: env_value 20 - special.env: env_value 20 - special.env: env_value 21 - special.env: env_value 22 - special.env: env_value 23 - special.env: env_value 24 - special.env: env_value 25 - special.env: env_value 26 - special.env: env_value 27 - special.env: env_value 28 - special.env: env_value 29 - special.env: env_value 20 - special.env: env_value 20 - special.env: env_value 21 - special.env: env_value 21 - special.env: env_value 22 - special.env: env_value 23 - special.env: env_value 24 - special.env: env_value 24 - special.env: env_value 25 - special.env: env_value 26 - special.env: env_value 27 - special.env: env_value 28 - special.env: env</pre>	

6. 部署成功后,您可以在配置文件列表下看到 aliyun-config 配置文件。

配置文件列表			刷新创建配置项
集群: k8s-cluster ▼			
配置文件名称	命名空间	创建时间	操作
aliyun-config	default	2018-04-23 15:13:27	删除 修改

1.9.2 在 pod 中使用配置项

您可以在 Pod 中使用配置项,有多种使用场景,主要包括:

- 使用配置项定义 pod 环境变量
- 通过配置项设置命令行参数
- 在数据卷中使用配置项

更多关于配置项的信息,可以参见 Configure a Pod to Use a ConfigMap。

使用限制

您在 pod 里使用配置项时,需要两者处于同一集群和命名空间中。

创建配置项

本示例创建配置项 **special_config**,包含 SPECIAL_LEVEL: very 和 SPECIAL_TYPE: charm 两 个键值对。

使用编排模板创建配置项

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,然后单击右上角的使用模板创建。
- 3. 选择所需的集群和命名空间,选择样例模板或自定义,然后单击创建。

您可以使用如下 yaml 示例模板创建配置项。

```
apiVersion: v1
kind: ConfigMap
metadata:
   name: special-config
   namespace: default
data:
   SPECIAL_LEVEL: very
   SPECIAL_TYPE: charm
```

通过 Web 界面创建配置项

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。

- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 配置项,进入配置项列表页面。
- 4. 选择所需的集群和命名空间,然后单击右上角的使用模板创建。
- 5. 输入配置项名称,然后单击添加,输入配置项,最后单击确定。

容器服务		配置项					
Kubernetes	Swarm						
概览		集群	c15	52b4c3a188140ec8af63b34c23b	14a3		
▼ 集群		命名空间	det	fault			
集群		* 配置项名:	称: s	pecial-config			
**			名	称只能包含小写字母数字,'-'或	皆!,名称不能为空。		
제미		配置项:	3	变量名称	变量值		操作
存储			s	SPECIAL_LEVEL	very	编辑	删除
▼ 应用			s	SPECIAL TYPE	charm	编辑	删除
部署						 	
容器组			2	3称	值		添加
服务			变	量名不能重复,变量名和变量值	不能为空。		
昭由	Ξ		4	编辑配置文件			
FT FT					确定 取消		
发布							
配置项							

使用配置项定义 pod 环境变量

使用配置项的数据定义 pod 环境变量

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,然后单击右上角的使用模板创建。
- 3. 选择所需的集群和命名空间,选择样例模板或自定义,然后单击创建。

您可以在 pod 中定义环境变量,使用 valueFrom 引用 SPECIAL_LEVEL 的 value 值,从而定 义 pod 的环境变量。

下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-1
spec:
   containers:
     - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
        - name: SPECIAL_LEVEL_KEY
          valueFrom:
                                                 ##使用valueFrom来指
定env引用配置项的value值
            configMapKeyRef:
                                                 ##引用的配置文件名称
              name: special-config
              key: SPECIAL_LEVEL
                                                 ##引用的配置项key
```

```
restartPolicy: Never
```

同理,如果您需要将多个配置项的 value 值定义为 pod 的环境变量值,您只需要在 pod 中添加多个 env 参数即可。

将配置项的所有 key/values 配置为 pod 的环境变量

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,然后单击右上角的使用模板创建。
- 选择所需的集群和命名空间,选择样例模板或自定义,然后单击创建。

如果您想在一个 pod 中将配置项的所有 key/values 键值对配置为 pod 的环境变量,可以使用 envFrom 参数,配置项中的 key 会成为 Pod 中的环境变量名称。

下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
name: config-pod-2
spec:
containers:
- name: test-container
image: busybox
command: [ "/bin/sh", "-c", "env" ]
envFrom: ##引用 sepcial-config 配置文件的所有 key
/values 键值对
- configMapRef:
name: special-config
restartPolicy: Never
```

通过配置项设置命令行参数

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,然后单击右上角的使用模板创建。
- 3. 选择所需的集群和命名空间,选择样例模板或自定义,然后单击创建。

您可以使用配置项设置容器中的命令或者参数值,使用环境变量替换语法 \$(VAR_NAME)来进

行。

下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
   name: config-pod-3
spec:
   containers:
        - name: test-container
```
运行这个 pod 后, 会输出如下结果。

very charm

在数据卷中使用配置项

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 部署,然后单击右上角的使用模板创建。
- 3. 选择所需的集群和命名空间,选择样例模板或自定义,然后单击创建。

您也可以在数据卷里面使用配置项,在 volumes 下指定配置项名称,会将 key/values 的数据存储到 mountPath 路径下(本例中是 /*etc/config*)。最终生成以 key 为文件名, values 为文件内容的配置文件。

下面是一个编排示例。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-4
spec:
  containers:
     - name: test-container
       image: busybox
      command: [ "/bin/sh", "-c", "ls /etc/config/" ] ##列出该目录
下的文件名
      volumeMounts:
       - name: config-volume
        mountPath: /etc/config
  volumes:
     - name: config-volume
      configMap:
        name: special-config
```

restartPolicy: Never

运行 pod 后, 会输出配置项的 key。

SPECIAL_TYPE SPECIAL_LEVEL

1.9.3 修改配置项

您可以修改配置项的配置。



修改配置文件会影响使用该配置文件的应用。

通过配置项进行修改

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 配置项,进入配置文件列表。
- 3. 选择所需的集群和命名空间,选择需要修改的配置项并单击右侧的修改。

容器服务	配置项列表			刷新创建配置项
Kubernetes Swarm				
▼ 集群	集群 k8s-cluster V 命名空间 default V	3		
集群	配置项名称	命名空间	创建时间	操作
节点	aliyun-config	default	2018-04-23 15:13:27	删除 4
存储	test-config	default	2018-04-23 15:25:33	删除 修改
	test-mariadb	default	2018-04-23 14:12:26	删除 修改
響等				
容器组				
服务				
发布				
配置项 2				

4. 在弹出的确认对话框中,单击确定。

修改确认		\times
0	修改配置文件会影响使用该配置文件的应用,您确定要修改吗?	
	确定取	消

- 5. 修改配置项。
 - 选择需要修改的配置项并单击右侧的编辑,修改配置后单击保存。
 - 或者单击编辑配置文件,完成编辑后单击确定。

集群	and the second			
命名空间	default			
*配置项名称:	test-config			
配置项:	变量名称	变量值		操作
	enemies	aliens		编辑 删除
	lives	3		编辑 删除
	名称	值		添加
	变量名不能重复,变量	星名和变量值不能为空。		
	编辑配置文件			
		确定 耳	以消	

6. 完成配置项修改后,单击确定。

通过 Kubernetes Dashboard 进行修改

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入集群列表。
- 4. 选择所需的集群,并单击右侧的控制台。

容器服务	集群列表		您最多可以创建 5 个集群	,每个集群最多可以添加 20 个节点	刷新 创建 Kubernetes 集群 🗸
Kubernetes Swarm					
* 集群	常见问题: び 如何 団 建栗群 び 步 容和 暗容栗群	る、通过 Kubecti 注版 Kubernetes 兼件 る、通过部分	管理应用		
集群	名称 ▼				
节点	集群名称/ID ;	地域 网络类型 集群状态	创建时间	Kubernetes 版本	2 ⊧
存储	k8s-cluster	华东1 虚拟专有网络 ●运行中 ●运行中	2018-04-20 11:02:31	1.9.3	管理 查看日志 控制台 集群伸缩 更多▼

 在 Kubernetes Dashboard 页面,单击左侧导航栏中的配置与存储 > 配置字典,选择所需的配置 字典,单击右侧的操作 > 查看/编辑 YAML。

102

持久化存储卷	■ 配置与存储 配置字典
角色	
存储类	配置字典
命名空间	名称 🜲
default 🔻	test-config
概况	
工作负载	aliyun-config
定时任务	test-mariadb
守护进程集	
部署	
任务	
容器组	
副本集	
副本控制器	
有状态副本集	
服务发现与负载均衡	
访问权	
服务	
配置与存储	

6. 弹出修改配置字典对话框,对配置变量进行修改后,单击更新。



1.9.4 删除配置项

您可以删除不再使用的配置项。

通过配置项菜单进行删除

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 配置项,进入配置文件列表。
- 3. 选择目标集群,选择需要修改的配置项并单击右侧的删除。

容器服务	配置文件列表			刷新创建配置项
Kubernetes Swarm				
▼ 集群	集群: k8s-cluster ▼ 3			
集群	配置文件名称	命名空间	创建时间	操作
节点	aliyun-config	default	2018-04-23 15:13:27	删除 修改
存储	test-config	default	2018-04-23 15:25:33	4 删除 修改
▼应用 1	test-mariadb	default	2018-04-23 14:12:26	制除 修改
部署	cluster-info	kube-public	2018-04-23 11:48:31	删除 修改
服务	extension-apiserver-authentication	kube-system	2018-04-23 11:48:29	删除 修改
发布	ingress-controller-leader-nginx	kube-system	2018-04-23 11:58:42	制除 修改
配置项 2 三	kube-flannel-cfg	kube-system	2018-04-23 11:48:39	删除 修改

通过 Kubernetes Dashboard 进行删除

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。
- 4. 单击左侧导航栏中的集群,选择目标集群,单击右侧的控制台。

容器服务	集群列表		您最多可以创建 5 个集群	, 每个集群最多可以添加 20 个节点	刷新 创建 Kubernetes 集群 ▼
Kubernetes Swarm	常见问题: ③如何创建集群 ④ 扩容和缩容集群	♂通过 kubecti 连接 Kubernetes 集群 ♂ 通过命令管	理应用		
_{集群} 1	名称 *				
节点	集群名称/ID 地	地域 网络类型 集群状态	创建时间	Kubernetes 版本	2 🕫
存储	k8s-duster 4	¥东1	2018-04-20 11:02:31	1.9.3	管理 査看日志 控制台 集群伸缩 更多▼

 在 Kubernetes Dashboard 页面,选择所需的命名空间,单击左侧导航栏中的配置与存储 > 配置 字典,单击右侧的的操作按钮,在下拉框中单击删除。

节点	■ 配置与存储 配置字典	十 创建
持久化存储器		
角色	配置字典	
存储类	各称 🗣 杨蓝	日创建 🗢
命名空间	test-config .	2018 15:2 删除 3
default 🝷 🚺	aliyun-config -	2018 15:1 查看/编辑 YAML
概見	test-mariadb app: test-maria chart: mariadb-0 heritage: Trill release: t	2018-04-23
上作页载 中时/1条		
守护进程集		
部署		
任务		
容器组		
副本集		
副本控制器		
有状态副本集		
服务发现与负载均衡		
访问权		
服务		
配置与存储 配置字典		

6. 在弹出的对话框中,单击删除。

1.10 密钥管理

1.10.1 创建密钥

前提条件

您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。

背景信息

若您需要在 Kubernetes 集群中使用一些敏感的配置,比如密码、证书等信息时,建议使用密钥(secret),即保密字典。

密钥有多种类型,例如:

- Service Account:用来访问 Kubernetes API,由 Kubernetes 自动创建,并且会自动挂载到 Pod 的/*run/secrets/kubernetes.io/serviceaccount*目录中。
- Opaque: base64 编码格式的 Secret,用来存储密码、证书等敏感信息。

默认情况下,通过 Web 界面只能创建 Opaque 类型的密钥。Opaque 类型的数据是一个 map 类型,要求value 是 base64 编码格式。阿里云容器服务提供一键创建密钥的功能,自动将明文数据编码为 base64 格式。

您也可通过命令行手动创建密钥,请参见 kubernetes secret 了解更多信息。

操作步骤

1. 登录 容器服务管理控制台。

- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 保密字典,进入保密字典页面。
- 3. 选择所需的集群和命名空间,单击右上角的创建。

容器服务	0	保	密字典					刷新	创建
概览	Swarm	集群	[#] k8s-cluster ▼ 命名空间	default 🔻 3					4
▼ 集群		名	称	类型	命名空	间创建时间	1		操作
集群		de	fault-token-46hv4	kubernetes.io/service-account-token	default	2018-05	-18 17:22:31	详情 🦾 编辑	制除
节点									
存储									
命名空间									
▼应用	1								
部署									
容器组									
服务		E							
路由									
发布									
配置项									
保密字典] 2								

4. 配置新的保密字典。



命名空间	default	
* 名称	account 1	
	名称长度为 1-253 字符 , 只能包含小写	字母、数字、'-和'.'
* 数据	● 名称	值
	2	1f2d1e2e67df
	password	
		admin
	e username	
	名称只能包含数字、字母、"_"、"~"和", 团 对数据值进行 base64 编码	N
		3 政 治

1. 名称:输入密钥的名称。名称长度为 1-253 字符,只能包含小写字母、数字、'-'和'.'

- 配置密钥的数据。单击添加,在弹出的对话框中,输入密钥名称和值,即键值对。本例中创建的密钥包含两个 value, username:admin和 passwrod:1f2d1e2e67df。
- 3. 单击确定。
- 5. 默认返回保密字典页面,您可看到新建的密钥出现在列表中。

保密字典					刷	新	创建
集群 k8s-cluster 🔻 命名空间	default 🔻						
名称	类型	命名空间	创建时间				操作
account	Opaque	default	2018-05-18 18:18:20	详情		编辑	删除
default-token-46hv4	kubernetes.io/service-account-token	default	2018-05-18 17:22:31	详情		编辑	删除

1.10.2 查看密钥

您可通过阿里云容器服务 Web 界面查看已创建的密钥。

前提条件

- 您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。
- 您已创建一个密钥,参见创建密钥。

背景信息

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 保密字典,进入保密字典页面。
- 3. 选择所需的集群和命名空间,选择所需的密钥,并单击右侧的详情。

容器服务	保密字典					刷新	创建
Kubernetes Swarm	集群 test-k8s ▼ 命名空间	default 🔻 3					
(50.0) ▼ 集群	名称	类型	命名空间	创建时间	4		操作
集群	account	Opaque	default	2018-05-31 14:03:50	详	青 编辑	制除
节点	default-token-drpgj	kubernetes.io/service-account-token	default	2018-05-28 17:09:06	渊	青 编辑	制除
存储							
命名空间							
部署							
容諸祖							
路由							
发布							
配置项							
保密字典 2							

4. 进入密钥的详情页面,您可查看该密钥的基本信息,以及密钥包含的数据信息。

单击详细信息中数据名称右侧的图标,可查看数据的明文。

保密字典 - account		周新 编辑
基本信息		
名称: account		命名空间: default
创建时间: 2018-05-31 14:03:50		类型: Opaque
详细信息		
password		
username	۲	

1.10.3 编辑密钥

您可通过阿里云容器服务 Web 界面直接编辑已有密钥。

前提条件

- 您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。
- 您已创建一个密钥,参见创建密钥。

背景信息

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 保密字典,进入保密字典页面。
- 3. 选择所需的集群和命名空间,选择所需的密钥,并单击右侧的编辑。

容器服务			保密字典					刷新	êlî	建
Kubernetes	Swarm		年詳 tast k0g ▼	会名空间 dafault · ·	3					
概览			Uest-Kos *	delaur ,	Ŭ					
▼ 集群			名称	送型		命名空间	创建时间		4	操作
集群			account	Opaque		default	2018-05-31 14:03:50	洋情 編	編 #	删除
节点			default-token-drpgj	kubernetes.io/se	ervice-account-token	default	2018-05-28 17:09:06	详情 🍐 🦋	辑 一册	搬
存储										
命名空间										
▼ <u>应</u> 用	1									
部署										
容器组										
服务		÷								
路由										
发布										
配置项	_									
保密字典	2									

4. 在编辑密钥的页面,可对密钥的数据进行编辑。

编辑保密字典		
命名空间	default	
* 名称	account	
* 数据	▲ 名称	值
	 username 	admin
	password	1f2d1e2e67df
	名称只能包含数字、字母、"_"、"-"和"."	- Profess
		Attac Attac

5. 最后单击确定。

1.10.4 删除密钥

您可通过阿里云容器服务 Web 界面删除已有密钥。

前提条件

- 您已成功创建一个 Kubernetes 集群,参见创建Kubernetes集群。
- 您已创建一个密钥,参见创建密钥。

背景信息



集群创建过程中生成的密钥请勿删除。

- 1. 登录 容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 保密字典,进入保密字典页面。
- 3. 选择所需的集群和命名空间,选择所需的密钥,并单击右侧的删除。

容器服务		保密字典				刷新创建
Kubernetes	Swarm					
概览		集群 test-k8s ▼ 命名空间	default 🔻 3			
▼ 集群		名称	美型	命名空间	创建时间	4
集群		account	Opaque	default	2018-05-31 14:03:50	洋情 編輯 删除
节点		default-token-drpgj	kubernetes.io/service-account-token	default	2018-05-28 17:09:06	洋情 编辑 删除
存储						
命名空间						
	1					
部署						
容器组						
服务		-				
路由						
发布						
配置项						
保密字典	2					

4. 在弹出的对话框中,单击,完成删除。

提示		×	
•	确定要删除 account 吗?		
	确	定取消	

1.11 路由管理

1.11.1 路由配置说明

阿里云容器服务提供高可靠的 ingress controller 组件,集成了阿里云 SLB 服务,为您的 Kubernetes 集群提供灵活可靠的路由服务(Ingress)。

下面是一个 Ingress 编排示例。通过 Web 界面进行配置时,您需要对注释的参数进行配置,部分配 置需要创建依赖项,具体请参见通过 Web 界面创建路由。 也可以参考*Ingress* 支持 和 *Kubernetes Ingress*。

```
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
    annotations:
    nginx.ingress.kubernetes.io/rewrite-target: / ##注解
    name: nginx-ingress ##
Ingress 名称
```

spec:	
rules:	##路由
规则	
- host: *.xxxxxxx.cn-hangzhou.alicontainer.com	##集群
默认服务访问域名	
http:	
paths:	
- backend:	
serviceName: http-svcl	##
backend服务	
servicePort: 80	
path: /svcl	##
URL	
- host: foo.bar.com	##测试
域名	
http:	
paths:	
- backend:	
serviceName: http-svc2	
servicePort: 80	
path. /SVC2	
serviceName: http-svc3	
servicePort: 80	
path: /svc3	
tls:	## 开启
TIC	
- hosts:	
- *.xxxxxx.cn-hangzhou.alicontainer.com	
- foo.bar.com	
secretName: nginx-ingress-secret #	#使用的
secret 名称	

注解

您可以指定 ingress 的 annotation ,指定使用的 ingress controller,以及路由的规则,如常见的重 写规则。参见 *Ingress rewrite*。

例如,一个典型的重定向注解:nginx.ingress.kubernetes.io/rewrite-target: / 会 将/path路径重定向到后端服务能够识别的/ 路径上面。

规则

规则指的是授权入站连接到达集群服务的路由规则,通常指 http 规则,包括域名(虚拟主机名称)、URL 访问路径、服务及端口等。

每条 http 规则需要配置以下信息:

• **host** 配置项:比如阿里云 Kubernetes 集群服务测试域名;或虚拟主机名称,如 foo.bar. com。

- URL 路径:指定访问的 URL 路径,即 path。每个 path 都关联一个 backend (服务),在阿里 云 SLB 将流量转发到 backend 之前,所有的入站请求都要先匹配 host 和 path。
- backend 配置:即服务配置,是一个 service:port 的组合,比如 http-svc1:80
 。Ingress 的流量被转发到它所匹配的 backend。

TLS

您可以通过指定包含 TLS 私钥和证书的 secret 来加密 Ingress,实现安全的路由访问。TLS secret 中必须包含名为 tls.crt 和 tls.key 的证书和私钥。更多 TLS 的原理,请参见 *TLS*;关于如何生成 secret,请参见配置安全的路由服务。

标签

您可为 ingress 添加标签,标示该 Ingress 的特点。

1.11.2 通过 Web 界面创建路由

阿里云容器服务 Web 界面集成了路由(Ingress)服务,您可通过 Web 界面快速创建路由服务,构建灵活可靠的流量接入层。

前提条件

- 您已经成功创建一个 Kubernetes 集群,参见创建 *Kubernetes* 集群,并且集群中 Ingress controller 正常运行。
- SSH 登录到 Master 节点,参见SSH 访问 Kubernetes 集群。

步骤1 创建 deployment 和服务

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,进入路由页面。
- 3. 单击页面右上角使用模板创建。

容器服务 - Kubernetes ▼	部署列表	刷新使用编卷创建使用模坛创建
概览	集群 k8s-cluster ▼ 命名空间 default ▼	2
▼ 集群	名称 标签 容器组数量	创建时间 操作
集群		
节点	(ľ	没有查询到符合条件的记录
存储卷		
命名空间		
▼ 应用 部署		

4. 选择所需的集群和命名空间,选择样例模板或自定义,然后单击创建。

本例中,示例模板是一个 nginx 的 deployment。

集群	k8s-cluster		٣	
命名空间	default	1	Ŧ	
示例模版	自定义		٣	
横灰	<pre>1 epiVersion: extensions 2 kind: Deployment 3 - metadata: 4 name: old-nginx 5 - spect 7 - selector: 8 - match.bels: 9 run: old-nginx 10 - template: 11 - metadata: 12 labels: 13 - magenulboil: 16 - image: registr 17 - imagenulboil: 18 name: old-ngin 19 ports: 10 - containerPon 21 portocol: TO 22 restartPolicy: A 23 24 apiVersion: extensions 25 kind: Deployment 26 - metadata: 27 name: new-nginx 28 - spec: 29 replicas: 1 30 - selector: 31 - match.bels: 32 run: new-nginx 33 - template: 35 - labels: 36 sum: new-nginx 36 template: 36 sum: new-nginx 37 sum: new-nginx 38 - template: 36 sum: new-nginx 39 sum: new-nginx 30 sum: new-nginx 30 sum: new-nginx 31 labels: 32 sum: new-nginx 33 template: 34 sume: new-nginx 35 sum: new-nginx 36 sum: new-nginx 37 sume: new-nginx 38 sume: new-nginx 39 sume: new-nginx 30 sume: new-nginx 31 sume: new-nginx 33 template: 34 sume: new-nginx 35 sume: new-nginx 35 sume: new-nginx 36 sume: new-nginx 37 sume: new-nginx 38 sume: new-nginx 39 sume: new-nginx 30 sume: new-nginx 30 sume: new-nginx 31 sume: new-nginx 33 template: 34 sume: new-nginx 35 sume: new-nginx 35 sume: new-nginx 36 sume: new-nginx 37 sume: new-nginx 38 sume: new-nginx 39 sume new-nginx 30 sume n</pre>	/vibeta1 y.cn-hangzhou.aliyuncs.com/xianlu/old-nginx y: Always t: 80 p Jways /vibeta1		液加部署

5. 单击左侧导航栏中的应用 > 服务 , 进入服务列表页面 , 单击右上角创建。

本例中,为了测试复杂路由服务,您需要创建相应的多个服务,并将服务绑定到后端 deployment 上,采用 clusterIP 的服务类型。

创建服务		\times
名称:	new-svc	
类型:	虚拟集群IP ▼	
关联部署:	new-nginx 🔻	
端口映射:	●添加	
	服务端口 容器端口 协议	
	80 80 TCP v O	
注解:	● 添加	
标签:	● 添加	
	创建	取消

创建完成后,在服务列表,您可看到本示例创建的3个服务。

服务列表						局時所會成准
集群 k8s-cluster	▼ 命名空间	default 🔻				
名称	类型	创建时间	集群IP	内部端点	外部端点	操作
domain-nginx	NodePort	2018-07-11 17:43:32		domain-nginx:80 TCP domain-nginx:32347 TCP	-	洋情 更新 查看YAML 删除
kubernetes	ClusterIP	2018-07-11 17:35:35	10.00	kubernetes:443 TCP	-	详情 更新 查看YAML 删除
new-nginx	NodePort	2018-07-11 17:37:01		new-nginx:80 TCP new-nginx:32637 TCP	-	洋情 更新 查看YAML 删除
old-nginx	NodePort	2018-07-11 17:37:01		old-nginx:80 TCP old-nginx:32039 TCP	-	详情 更新 查看YAML 删除

步骤2 创建路由

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 路由,进入路由页面。

4. 选择所需的集群和命名空间,单击页面右上角的创建。

容器服务 - Kubernetes ▼	路由 (Ingress)				刷新 创建
概览	集群 k8s-cluster ▼ 命名	空间 default 🔻 3			4
▼ 集群	名称	論点	规则	创建时间	操作
集群					
节点			(1)	没有查询到符合条件的记录 ————————————————————————————————————	
存储卷					
命名空间					
▼ 应用 1					
部署					
容器组					
服务 🖸					
路由 2					

5. 在弹出的路由创建对话框中,首先配置路由名称,本例为 nginx-ingress。

名称:	nginx-ingress	

6. 对路由规则进行配置。

路由规则是指授权入站到达集群服务的规则,通常是 http 规则,配置项包括域名(虚拟主机名称)、URL 路径、服务名称和端口配置等。详细的信息,请参见路由配置说明。

本例中配置添加一条复杂的路由规则,配置集群默认的测试域名和虚拟主机名称,展示基于域名的路由服务。

规则:	▶ 添加			
	域名 test.c	5.cn-hangzhoi	u.alicontainer.com	8
	使用 *.c0 路径 /	.cn-hangzh	ou.alicontainer.com 或者 自定)	Ϋ́
	服务 💿 添加			
	名称	端口	权重 权重比例	
	domain-nginx	• 80 •	100 100.0%	•
	域名			8
	foo.bar.com			
	使用 *.c(5.cn-hangzh	ou.alicontainer.com 或者 自定.	ž.
	##12 /			
	服务 💿 添加			
	名称	靖口	权重 权重比例	
	new-nginx	▼ 80 ▼	50 50.0%	•
	old-nginx	v 80 v	50 50.0%	•

- 基于默认域名的简单路由,即使用集群的默认域名对外提供访问服务。
 - 域名配置:使用集群的默认域名,本例中是test.[cluster-id].[region-id].
 alicontainer.com。

在创建路由对话框中,会显示该集群的默认域名,域名格式是*.[cluster-id].[region-id].alicontainer.com;您也可在集群的基本信息页面中获取。

- 服务配置:配置服务的访问路径、名称以及端口。
 - 访问路径配置:指定服务访问的 URL 路径,如 /svc1,每个路径(path)都关联一 个 backend(服务),在阿里云 SLB 将流量转发到 backend 之前,所有的入站请求都 要先匹配域名和路径。
 - 服务名称及端口配置:即 backend 配置,是一个 service:port 的组合,比如 http -svc1:80。lngress的流量被转发到它所匹配的 backend。

- 基于域名的简单扇出路由。本例中使用一个虚拟的主机名称作为测试域名对外提供访问服务。若您在生产环境中,可使用成功备案的域名提供访问服务。
 - 域名配置:本例中使用测试域名 foo.bar.com。

您需要修改 hosts 文件添加一条域名映射规则。

118.178.108.143 foo.bar.com

- 服务配置:配置服务的访问路径、服务名称以及端口。

■ 访问路径配置:指定服务访问的 URL 路径,如/svc2 和 /svc3。

■ 服务名称及端口配置:本例中指定 http-svc2:80 和 http-svc2:80。

7. 配置注解。

单击重定向注解,可为路由添加一条典型的重定向注解。即nginx.ingress.kubernetes. io/rewrite-target:/,表示将/path 路径重定向到后端服务能够识别的/路径上面。 您也可单击添加按钮,输入注解名称和值,即ingress的 annotation 键值对,如常见的重写注 解,参见 rewrite annotation。

解: ◎ 添加 重定向注解		
名称	值	
nginx.ingress.kubernetes.io/rewrit	e-target /	•

- 8. 配置 TLS。勾选开启 TLS, 配置安全的路由服务。具体可参见 配置安全的路由服务。
 - 您可选择使用已有密钥。

TLS:	☑ 开启 : ● 已有密钥 ● 新建密钥	
	foo.bar	•

1. 登录 master 节点, 创建 tls.key 和 tls.crt。

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls
.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"
```

2. 创建一个 secret。

kubectl create secret tls foo.bar --key tls.key --cert tls.crt

- **3.** 执行命令 kubectl get secret,您可看到该 secret 已经成功创建。在 Web 界面可选择创建的*foo.bar*这个 secret。
- 您可选择在 TLS 界面上利用已创建的 TLS 私钥和证书,一键创建 secret。



1. 登录 master 节点, 创建 tls.key 和 tls.crt。

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls
.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"

- 2. 执行 vim tls.key 和 vim tls.crt 获取生成的私钥和证书。
- 3. 将证书和私钥的内容复制到 TLS 新建密钥的面板。
- 9. 添加标签。

标签的作用是为 ingress 添加对应的标签,标示该 Ingress 的特点。

标签:	⊙ 添加	
	名称	值
	node-role.kubernetes.io/ingress	true

10.最后单击创建,返回路由列表。

等待一段时间,可以看到一条路由。

路由 (Ingress)			刷新创建
集群 k8s-cluster v 命名	2间 default •		
名称 端点	规则	创建时间	操作
nginx-ingress	test.cf 5.cn-hangzhou.alicontainer> domain-nginx foo.bar.com/ -> new-nginx foo.bar.com/ -> old-nginx	2018-07-11 17:49:46	洋情 変更 查看YAML 删除

11.单击路由中的访问域名 test.[cluster-id].[region-id].alicontainer.com,以及

foo.bar.com/svc2 和 foo.bar.com/svc3,可访问 nginx 的欢迎页面。

(i) test.cC	a, about a strand see, as sold of the strand sector
	Welcome to nginx!
	If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
	For online documentation and support please refer to <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .
	Thank you for using nginx.
$\leftarrow \rightarrow \mathbf{G}$) foo.bar.com/?spm=5176.2020520152.0.0.509c61b1iW1N16
old	

1.11.3 查看路由

前提条件

- 您已经成功创建一个 Kubernetes 集群,参见创建 *Kubernetes* 集群,并且集群中 Ingress controller 正常运行。
- 您已经成功创建一个路由,参见通过 Web 界面创建路由。

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的 应用 > 路由,进入路由页面。
- 3. 选择所需的集群和命名空间,选择所需的路由,然后单击路由右侧的详情。



进入路由详情页面,您可查看该路由的基本信息以及路由规则。

nginx-ingress t 返回列表				刷新
基本信息				
名称:	nginx-ingress			
命名空间;	default			
创建时间:	2018-04-28T03:38:05Z			
标签:				
注解:	nginx.ingress.kubernetes.io/rewrite-target:/			
端点:	13			
规则				
域名		路径	服务名称	服务端口
test.c1	The Conhangzhou.alicontainer.com	/svc1	http-svc1	80
foo.bar.com		/svc2	http-svc2	80
foo.bar.com		/svc3	http-svc3	80

1.11.4 变更路由

前提条件

- 您已经成功创建一个 Kubernetes 集群,参见创建 *Kubernetes* 集群,并且集群中 Ingress controller 正常运行。
- 您已经成功创建一个路由,参见通过 Web 界面创建路由。

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 路由,进入路由页面。
- 3. 选择所需的集群和命名空间,选择所需的路由,然后单击路由右侧的变更。

容器服务	路由 (Ingress)		创建 易	则哪斤
Kubernetes Swarm				
概览	集群 test-k8s V 命名空间 default V 3			
▼ 集群	名称 巍点 规则	创建时间		操作
集群	test.c h-hangzhou.alicontainer> http-svc1 nginx-ingress : foo.bar.com/svc2 -> http-svc2 foo.bar.com/svc3 -> http-svc3	2018-04-28 11:38:05	详情 変更 ;	删除
节点			- 4	
存储				
▼应用 1				
部署				
容器组				
服务				
<u>路由</u> 2				

 在弹出的对话框中,对路由的相关参数进行变更,然后单击确定。本例中将 test.[clusterid].[region-id].alicontainer.com 修改为 testv2.[cluster-id].[region-id]. alicontainer.com。



后续操作

返回路由列表,您可看到该路由的其中一条路由规则发生变化。

路由	(Ingress)			创建局	剥新
集群	test-k8s	▼ 命名空间	default 🔹			
名称		読点	规则	创建时间		操作
nginx-	-ingress		testv2 c1	2018-04-28 11:38:05	洋情 変更 :	删除

1.11.5 删除路由

前提条件

- 您已经成功创建一个 Kubernetes 集群,参见创建 *Kubernetes* 集群,并且集群中 Ingress controller 正常运行。
- 您已经成功创建一个路由,参见通过 Web 界面创建路由。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 路由,进入路由页面。
- 3. 选择所需的集群和命名空间,选择所需的路由,然后单击路由右侧的删除。

容器服务		路由 (Ingress)				创建	刷新
Kubernetes	Swarm			0			
概览		集群 test-k8s ▼ 命名空	🗊 default 🔻	3			
▼ 集群		名称 端点	规则		创建时间		操作
集群		nginx-ingress :	test.c foo.bar.com/svc2 -> http-svc2 foo.bar.com/svc3 -> http-svc3	h-hangzhou.alicontainer> http-svc1	2018-04-28 11:38:05	详情 变更	删除
节点							4
存储							-
▼应用	1						
部署							
容器组							
服务							
路由	2						

4. 在弹出的对话框中,单击确定,完成删除。

确认			\times
0	删除nginx-ingress ?		
		确定	取消

1.12 应用目录管理

1.12.1 应用目录概述

微服务是容器时代的主题,应用微服务化给部署和管理带来极大的挑战。通过将庞大的单体应用拆分成一个个微服务,从而使各个微服务可被独立部署和扩展,实现敏捷开发和快速迭代。虽然微服

务带来了很大的好处,但同时,由于应用拆分成许多组件,对应着庞大数量的微服务,开发者不得 不面对这些微服务的管理问题,如资源管理、版本管理、配置管理等。

针对 Kubernetes 编排下微服务管理问题,阿里云容器服务引入 Helm 开源项目并进行集成,帮助简化部署和管理 Kubernetes 应用。

Helm 是 Kubernetes 服务编排领域的开源子项目,是 Kubernetes 应用的一个包管理工具, Helm 通过软件打包的形式,支持发布的版本管理和控制,简化了 Kubernetes 应用部署和管理的复杂 性。

阿里云应用目录功能

阿里云容器服务应用目录功能集成了 Helm,提供了 Helm 的相关功能,并进行了相关功能扩展,如 提供图形化界面、阿里云官方 Repository 等。

应用目录首页 chart 列表的信息包含:

- chart 名称:一个 Helm 包,对应一个目标应用,其中包含了运行一个应用所需要的镜像、依赖 和资源定义等。
- 版本: chart 的版本号。
- Repository:用于发布和存储 Chart 的仓库,例如官方仓库 stable、incubator 等。

各个 chart 详情页包含的信息不尽相同,例如,可能包含:

- chart 简介
- chart 详细信息
- chart 安装到集群的前提条件,如预先配置持久化存储卷(pv)。
- chart 安装命令
- chart 卸载命令
- chart 参数配置项

目前,您可以通过 helm 工具部署和管理应用目录中的 chart,具体请参见利用 *Helm* 简化应用部署。

1.12.2 查看应用目录列表

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的市场 > 应用目录,进入应用目录列表。

您可以查看该页面下 chart 列表,每个 chart 对应一个目标应用,包含一些基本信息,如应用名称、版本号和来源仓库等信息。



后续操作

您可进入单个 chart, 了解具体的 chart 信息, 可根据相关信息, 利用 Helm 工具部署应用, 具体请参见利用 Helm 简化应用部署。

1.13 服务目录

1.13.1 概述

云平台上运行的应用需要使用一些基础服务,如数据库、应用服务器等通用的基础软件。譬如一个 wordpress 应用,作为一个 web 应用,后端需要一个数据库服务,如 MariaDB。传统方式是在 wordpress 应用的编排中也创建应用依赖的 MariaDB 服务,并与 Web 应用进行集成。这种云上应 用开发的方式,就需要开发者花费精力解决所依赖的基础设施软件的部署和配置,增加应用托管和 迁移的成本。

阿里云容器服务支持并集成了服务目录的功能,该功能旨在接入和管理 Service Broker,使 kubernetes 上运行的应用可以使用 service broker 所代理的托管服务。服务目录功能将支持一系列 基础设施软件,应用开发者可以不用关心这些软件的可用性和伸缩能力,也不用对其进行管理,开 发者可以简单的将其作为服务使用,只需专注开发核心的应用程序。

服务目录通过 Kubernetes 的 Open Service Broker API 与 Service Broker 进行通信,并作为 Kubernetes API Server 的中介,以便协商首要规定(initial provisioning)并获取应用程序使用托管 服务的必要凭据。关于服务目录的具体实现原理,请参考 *service catalog*。

1.13.2 开通服务目录

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的市场 > 服务目录,在右上角选择目标集群。
- 3. 若您还未部署服务目录功能,界面上会提示您先进行安装。

容器服务		- 服务目录	: k8s-cluster 🔻
Kubernetes	Swarm		
▼ 集群		① 该集群尚未部署服务目录 , 请点击安装	3
集群		4	
节点			
存储			
▼ 应用			
部署			
服务			
发布			
配置项	Ξ		
▼ 市场 🚺			
应用目录	•		
服务目录	2		

4. 安装完成后,服务目录页面中会显示默认安装的 Service Broker,您可以进入 mariadb-broker 了 解详情。

📕 说明:

服务目录具体实现为一个扩展 API server 和一个 controller, 阿里云容器服务安装服务目录功能 后, 会创建一个 catalog 命名空间。

容器服务 Kubernetes	Swarm	服务目录			集群: k8s-cluster V
▼ 集群					
集群		₽ ⁰	Ø	Ø	Ø
节点					
存储		mariadb-broker	mysql-broker	rabbitmq-broker	spark-broker
▼ 应用					
部署					
服务					
发布					
配置项	E				
▼ 市场					
应用目录					
服务目录					

5. 在左侧导航栏中单击集群,单击目标集群右侧的控制台。

容 <mark>器服务</mark> Kubernetes Sw	/arm	集群列表				您最多可以创建 5 个集群	样,每个集群最多可以添加 20 个节点	局新 创建 Kubernetes 集群 →
▼集群		常见问题: Ø如何创建集群 Ø扩容和缩容集群	詳 🔗 通过	过 kubectl 连接 Kubernetes 集	群 🕜 通过命令管	理应用		
集群		名称 ▼						
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	2 🗉
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理 查看日志 控制台 集群伸缩 更多▼

6. 进入 Kubernetes Dashboard 页面,在命名空间中选择**catalog**,可以看到该命名空间下安装了 catalog apiserver 和 controller 相关的资源对象。

<	≡ 櫛兄						十创建
42.84	工作负载状态						
朱轩 余夕空间							
市点							
持久化存储卷	100.00%		100.00%			100.00%	
角色	登録		容器组			副本集	
存储类							
命名空间	部署						Ŧ
catalog 👻 🚺	名称 \$ 标签	2	容器组		已创建 🕈	镜像	
概况	ap	p: aliacs-service-catalog-catalog-apiser.			2018 04 04 00 24 21	registry.cn-hangzhou.aliyuncs.com/service-cata	
工作负载	anacs-service-catalog-catalog-apiserver crister	lease: aliacs-service-catal	1/1		2018-04-04 09:54:51	registry.cn-hangzhou.aliyuncs.com/service-cata	•
定时任务	ap	p: aliacs-service-catalog-catal	. / .		2010 04 04 00 24 21		
守护进程集	rel	lease: allacs-service-catal	1/1		2018-04-04 09:34:31	registry.cn-nangzhou.anyuncs.com/service-cata	
部署							
任务	容器组						Ŧ
容器组	名称 ≑	节点 状态:	÷.	已重启	已创 章 CPU(核)	内存(字节)	
副本集	 allace condex catalog catalog anisotror 86456dc580. 	cn-hangzhou.i-	ing	0	2018-04-	22 627 Mi	= :
副本控制器	anacstsenvice-catalog-catalog-apiserver-ou4boucbos-	bp1iabxy7s0uwpzmaa2e	ing .	0	09:34:32	25.037 MI	
有状态副本集 服务发现与负载均衡	Saliacs-service-catalog-catalog-controller-manager-5cf	cn-hangzhou.i- bp1iatxy7s0uwpzmaa2e Runn	ing	0	2018-04- 04 09:34:32	02 15.094 Mi	≡ :

后续操作

至此,您已经成功开通服务目录功能。接下来可以通过服务目录下的 Service Broker 来创建托管服务的实例,并将其应用到您自己的应用程序中。

1.14 镜像管理

1.14.1 查看镜像列表

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中应用 > 部署,进入部署列表。
- 3. 单击页面右上角的使用镜像创建。

容器服务		部署列表		使用	免除创建 使用模板创建 刷新
Kubernetes	Swarm				2
概览		集群 k8s-cluster ▼ 命名空间 def	fault 🔻		
▼ 伸弾		名称标签	容器组数量	创建时间	操作
JICHT					
集群				没有查询到符合条件的记录	
节点					
存储					
▼应用	1				
部署					
容器组					

4. 进入创建应用界面,输入应用名称,单击下一步。

の问题: 🔗 限(制容器的资源 🔗 高可用性调度	Ø 通过镜像创建Nginx Ø 通过编排	模板创建Wordpress	S 编排模板说明 S 标签	兑明	
应	用基本信息	应用配置	>	访问设置	>	创建完成
应用名称	test					
	名称为1-64个字符,可包含数字	字、英文字符,或"-",且不能以-开头				
部署集群:	k8s-cluster	¥				

5. 您可选择手动添加实例。

进入应用配置页面,单击选择镜像,在弹出的对话框中,查看镜像。

一创建	应用		镜像选择	全局渡索
常见	问题: 🔗 限制容	器的资源 🔗 高可用性調 立用基本信息	常用镜像 Do	ocker官方镜像 阿里云镜像 用户镜像 用户收藏
	镜像名称:	支持填写私有registr 选择镜像 1	♥ mongoDB	mongo 类型: 公开 来源: DOCKER_HUB ★756 MongoDB document databases provide high availability and easy scalability. 详情 收藏
聖记李寶	资源限制:	CPU	My <mark>SQL</mark> .	mysql 美型:公开 来源: DOCKER_HUB ★1174 MySQL is a widely used, open-source relational database management system (RDBMS). 详情 收藏
	州南又 <i>际</i> 。 数据卷:	СРU <u>1</u> 2 : 500m	NGINX	nginx 类型:公开 来源:DOCKER_HUB ★1115 Official build of Nginx. 注键 Wr# 共有8条,每页显示:9条 《 〈 1 〉 》
环境变量	环境变量:	0		施 定 取消
	Command :			

您可以查看镜像的种类或单击 全局搜索 通过镜像名称的关键字搜索镜像。

- 常用镜像: 容器服务推荐的一些常用镜像。
- Docker 官方镜像: Docker Hub 提供的官方镜像。
- 阿里云镜像: 阿里云容器 Hub 提供的镜像,包含公开镜像和私有镜像。
- 用户镜像: 用户创建的镜像。
- 用户收藏:用户可收藏其他页签下的镜像,也可以在容器镜像服务下进行收藏,都会同步到 该列表下。

1.15 VPC下 Kubernetes 的网络地址段规划

在阿里云上创建 Kubernetes 集群时,通常情况下,可以选择自动创建专有网络,使用默认的网络 地址即可。在某些复杂的场景下,需要您自主规划 ECS 地址、Kubernetes Pod 地址和 Service 地 址。本文将介绍阿里云 VPC 环境下 Kubernetes 里各种地址的作用,以及地址段该如何规划。

Kubernetes 网段基本概念

首先来看几个和 IP 地址有关的概念。

VPC 网段

您在创建 VPC 选择的地址段。只能从 10.0.0.0/8、172.16.0.0/12、192.168.0.0/16 三者当中选择一个。

交换机网段

在 VPC 里创建交换机时指定的网段,必须是当前 VPC 网段的子集(可以跟 VPC 网段地址一样,但不能超过)。交换机下面的 ECS 所分配到的地址,就是从这个交换机地址段内获取的。一个 VPC 下,可以创建多个交换机,但交换机网段不能重叠。

VPC 网段结构如下图。



Pod 地址段

Pod 是 Kubernetes 内的概念,每个 Pod 具有一个 IP 地址。在阿里云容器服务上创建 Kubernetes 集群时,可以指定Pod 的地址段,不能和 VPC 网段重叠。比如 VPC 网段用的是 172.16.0.0/12, Kubernetes 的 Pod 网段就不能使用 172.16.0.0/16,不能使用 172.17.0.0/16…,这些地址都涵盖在 172.16.0.0/12 里了。

Service 地址段

Service 也是 Kubernetes 内的概念,每个 Service 有自己的地址。同样,Service 地址段也不能和 VPC 地址段重合,而且 Service 地址段也不能和 Pod 地址段重合。Service 地址只在 Kubernetes 集群内使用,不能在集群外使用。

Kubernetes 网段和 VPC 网段关系如下图。



如何选择地址段

单 VPC+单 Kubernetes 集群场景

这是最简单的情形。VPC 地址在创建 VPC 的时候就已经确定,创建 Kubernetes 集群时,选择和 当前 VPC 不一样的地址段就可以了。

单 VPC+多 Kubernetes 集群场景

一个 VPC 下创建多个 Kubernetes 集群。在默认的网络模式下(Flannel), Pod 的报文需要 通过 VPC 路由转发,容器服务会自动在 VPC 路由上配置到每个 Pod 地址段的路由表。所有 Kubernetes 集群的 Pod 地址段不能重叠,但 Service 地址段可以重叠。

VPC 地址还是创建 VPC 的时候确定的,创建 Kubernetes 的时候,为每个 Kubernetes 集群选择一个不重叠的地址段,不仅不能和 VPC 地址重叠,也不和其他 Kubernetes Pod 段重叠。

需要注意的是,这种情况下 Kubernetes 集群部分互通,一个集群的 Pod 可以直接访问另外一个集群的 Pod 和 ECS,但不能访问另外一个集群的 Service。

VPC 互联场景

两个 VPC 网络互联的情况下,可以通过路由表配置哪些报文要发送到对端 VPC 里。以下面的场景为例,VPC 1 使用地址段 192.168.0.0/16,VPC 2 使用地址段 172.16.0.0/12,我们可以通过路由表,指定在 VPC 1 里把 172.16.0.0/12 的报文都发送到 VPC 2。



在这种情况下,VPC1里创建的 Kubernetes 集群,首先不能和 VPC1的地址段重叠,同时其地址 段也不能和要路由到 VPC2的地址段重叠。在 VPC2上创建 Kubernetes 集群也类似。这个例子中,Kubernetes 集群 Pod 地址段可以选择 10.0.0.0/8 下的某个子段。



这里要特别关注"路由到 VPC 2" 的地址段,可以把这部分地址理解成已经占用的地

址,Kubernetes集群不能和已经占用的地址重叠。

如果 VPC 2 里要访问 VPC 1 的 Kubernetes Pod,则需要在 VPC 2 里配置到 Kubernetes 的路由。

VPC 网络到 IDC 的场景

和 VPC 互联场景类似,同样存在 VPC 里部分地址段路由到 IDC ,Kubernetes 集群的 Pod 地址就 不能和这部分地址重叠。IDC 里如果需要访问 Kubernetes 里的 Pod 地址,同样需要在 IDC 端配置 到专线 VBR 的路由表。

1.16 负载均衡

1.16.1 概述

Kubernetes 集群提供了多种访问容器应用的方式,支持通过阿里云的负载均衡服务(Server Load Balancer)或者 Ingress 方式来访问内部服务的和实现负载均衡。

1.16.2 通过负载均衡 (Server Load Balancer) 访问服务

您可以使用阿里云负载均衡来访问服务。

通过命令行操作

1. 通过命令行工具创建一个 Nginx 应用。

root@master # kubectl run nginx -	image=registry	.aliyuncs.	com/acs/
netdia:latest			
root@master # kubectl get po			
NAME	READY	STATUS	RESTARTS
AGE			
nginx-2721357637-dvwq3	1/1	Running	1
65			

2. 为 Nginx 应用创建阿里云负载均衡服务,指定 type=LoadBalancer 来向外网用户暴露 Nginx

服务。

root@master # kubectl expose deployment nginx --port=80 --targetport=80 --type=LoadBalancer root@master # kubectl get svc NAME CLUSTER-IP EXTERNAL-IP PORT(S) AGE nginx 172.19.10.209 101.37.192.20 80:31891/TCP 4s

3. 在浏览器中访问 http://101.37.192.20,来访问您的 Nginx 服务。

通过 Kubernetes Dashboard 操作

1. 将下面的 yml code 保存到 nginx-svc.yml文件中。

```
apiVersion: v1
kind: Service
metadata:
   labels:
      run: nginx
   name: http-svc
   namespace: default
spec:
   ports:
      - port: 80
      protocol: TCP
      targetPort: 80
   selector:
      run: nginx
```

type: LoadBalancer

- 2. 登录容器服务管理控制台,单击目标集群右侧的控制台,进入 Kubernetes Dashboard 页面。
- 3. 单击创建,开始创建应用。



- 4. 单击上传 YAML 或 JSON 文件。选择刚才保存的nginx-svc.yml 文件
- 5. 单击上传。

这样会创建一个阿里云负载均衡实例指向创建的 Nginx 应用,服务的名称为 http-svc。

6. 在 Kubernetes Dashboard 上定位到 default 命名空间,选择服务。

可以看到刚刚创建的 http-svc 的 Nginx 服务和机器的负载均衡地址 http://114.55.79.24 :80。


7. 将该地址拷贝到浏览器中即可访问该服务。

更多信息

阿里云负载均衡还支持丰富的配置参数,包含健康检查、收费类型、负载均衡类型等参数。详细信 息参见 文档。

注释

阿里云可以通过注释的形式支持丰富的负载均衡功能。

使用已有的内网 SLB

需要指定两个annotation。注意修改成您自己的 Loadbalancer-id。

```
apiVersion: v1
kind: Service
metadata:
 annotations:
    service.beta.kubernetes.io/alicloud-loadbalancer-address-type:
intranet
    service.beta.kubernetes.io/alicloud-loadbalancer-id: your-
loadbalancer-id
 labels:
   run: nginx
 name: nginx
 namespace: default
spec:
 ports:
  - name: web
   port: 80
   protocol: TCP
   targetPort: 80
 selector:
    run: nginx
  sessionAffinity: None
  type: LoadBalancer
```

然后保存为 slb.svc 后,执行 kubectl apply -f slb.svc。

创建 HTTPS 类型的 Loadbalancer

先在阿里云控制台上创建一个证书并记录 cert-id,然后使用如下 annotation 创建一个 HTTPS 类型 的 SLB。

```
apiVersion: v1
kind: Service
metadata:
   annotations:
      service.beta.kubernetes.io/alicloud-loadbalancer-cert-id: your-
cert-id
      service.beta.kubernetes.io/alicloud-loadbalancer-protocol-port: "
https:443"
   labels:
      run: nginx
```

```
name: nginx
 namespace: default
spec:
 ports:
 - name: web
  port: 443
  protocol: TCP
   targetPort: 443
 selector:
   run: nginx
 sessionAffinity: None
 type: LoadBalancer
```



说明: 注释的内容是区分大小写的。

注释	描述	默认值
service.beta.kubernetes.io/ alicloud-loadbalancer-protocol -port	多个值之间由逗号分隔,比 如:https:443,http:80	无
service.beta.kubernetes.io/ alicloud-loadbalancer-address -type	取值可以是 internet 或者 intranet	internet
service.beta.kubernetes.io /alicloud-loadbalancer-slb- network-type	负载均衡的网络类型,取值可 以是 classic 或者 vpc	classic
service.beta.kubernetes.io/ alicloud-loadbalancer-charge- type	取值可以是 paybytraffic 或者 paybybandwidth	paybybandwidth
service.beta.kubernetes.io/ alicloud-loadbalancer-id	负载均衡实例的 ID。通过 loadbalancer-id 指定您已有的 SLB,已有 listener 会被覆盖, 删除 service 时该 SLB 不会被 删除。	无
service.beta.kubernetes.io/ alicloud-loadbalancer-backend -label	通过 label 指定 SLB 后端挂哪 些节点。	无
service.beta.kubernetes.io/ alicloud-loadbalancer-region	负载均衡所在的地域	无
service.beta.kubernetes. io/alicloud-loadbalancer- bandwidth	负载均衡的带宽	50

注释	描述	默认值
service.beta.kubernetes.io/ alicloud-loadbalancer-cert-id	阿里云上的认证 ID。您需要先 上传证书	«»
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-flag	取值是 on 或者 off	默认为 off。TCP 不需要改参数。因为 TCP 默认打开健康检查,用户不可设置。
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-type	参见 HealthCheck	
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-uri	参见 HealthCheck	
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-connect-port	参见 HealthCheck	
service.beta.kubernetes.io/ alicloud-loadbalancer-healthy- threshold	参见 HealthCheck	
service.beta.kubernetes. io/alicloud-loadbalancer- unhealthy-threshold	参见 HealthCheck	
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-interval	参见 HealthCheck	
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-connect-timeout	参见 HealthCheck	
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-timeout	参见 HealthCheck	

1.16.3 Ingress 监控配置

您可以通过开启 Ingress 默认 VTS 模块来查看 Ingress 监控数据。

通过命令行操作

1. 修改 Ingress ConfigMap 配置,增加配置项 enable-vts-status: "true"。

```
root@master # kubectl edit configmap nginx-configuration -n kube-
system
configmap "nginx-configuration" edited
```

修改后 Ingress ConfigMap 内容如下。

```
apiVersion: v1
data:
 enable-vts-status: "true" # 开启VTS模块
 proxy-body-size: 20m
kind: ConfigMap
metadata:
 annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","data":{"proxy-body-size":"20m"},"kind":"
ConfigMap", "metadata": { "annotations": { }, "labels": { "app": "ingress-
nginx" }, "name": "nginx-configuration", "namespace": "kube-system" } }
  creationTimestamp: 2018-03-20T07:10:18Z
 labels:
    app: ingress-nginx
 name: nginx-configuration
 namespace: kube-system
  selfLink: /api/v1/namespaces/kube-system/configmaps/nginx-
configuration
```

2. 验证 Ingress Nginx 正常开启 VTS 模块。

```
root@master # kubectl get pods --selector=app=ingress-nginx -n kube-
system
NAME
                                            READY
                                                       STATUS
RESTARTS
          AGE
nginx-ingress-controller-79877595c8-78gq8
                                             1/1
                                                                 0
                                                       Running
       1h
root@master # kubectl exec -it nginx-ingress-controller-79877595c8-
78gq8 -n kube-system -- cat /etc/nginx/nginx.conf | grep vhost_traf
fic_status_display
         vhost_traffic_status_display;
         vhost_traffic_status_display_format html;
```

3. 本地访问 Ingress Nginx 监控控制台。

```
    说明:
    鉴于安全考虑,默认 VTS Port 并不对外开放,这里通过 port-forward 方式来访问。
    root@master # kubectl port-forward nginx-ingress-controller-
79877595c8-78gq8 -n kube-system 18080
```

Forwarding from 127.0.0.1:18080 -> 18080

Handling connection for 18080

4. 通过 http://localhost:18080/nginx_status 来访问 VTS 监控控制台。

Nginx Vhost Traffic Status

Server main

			last					Vore	ion	Until	-		Con	nectio	ns				Requ	ests						Sh	ared me	mory		
			USL					vera	1011	opui	a	ctive	readin	ng writ	ing v	vaiting	acce	pted	handle	d T	otal	Req/s		name	•		maxSize	used	Size	usedNode
nginx-	ingress-	contro	oller-7	98775	595c8	-78g	9pg	1.	13.7	32m -	41s	7		0	1		6	93566	935	66 1	1428		vhos	t_traffic	_sta	tus	10.0 Mil	3 2	.4 KiB	
Server zones Requests Responses Traffic Cache																														
7	Req	uests			F	Resp	ons	es					Traffi	с								Cac	he							
Zone	Total R	eq/s 1	Time	1xx	2xx	3xx	4x)	5x	хТ	otal	Sent	Rc	vd S	Sent/s	Rcvd	/s Mis	s By	pass	Expire	d St	ale	Updat	ing R	evalidat	ed I	lit	Scarce	Total		
_	660	1	0ms	0	660	0		0	0	660 1	.7 Mil	3 1 4 5.4	KiB 1	I.1 KiB	503	в	0	0		0	0		0		0	0	0	0		
*	660	1	0ms	0	660	0		0	0	660 1	.7 Mil	3 1 4 5.4	KiB 1	I.1 KiB	503	В	0	0		0	0		0		0	0	0	0		
upst	ipstream-default-backend																													
S	erver	State	e Re	spons	se Tin	ne \	Neig	ht	Max	Fails	FailT	imeou	t Tota	al Requ	/s Tii	ne 1x	x 2xx	3xx	4xx 5	ixx 1	Total	Sent	Rcvd	Sent/s	Rc	vd/s	5			
172.16	6.3.6:8080) u	a		0	ms		1		0			0	0	0 0	ms	0	0 0	0 (0	0) 0 E	0 8	3 0	в	01	в			
upda	te inter	val:	1 🗘	sec																										

通过 Kubernetes Dashboard 操作

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,选择所需的集群并单击集群右侧的控制台,进入 Kubernetes Dashboard 页面。
- **3.** 编辑 kube-system 命名空间下的配置字典 nginx-configuration,增加配置项 enable-vts-status: "true"。

保存后 Ingress ConfigMap 内容如下。

```
"kind": "ConfigMap",
  "apiVersion": "v1",
  "metadata": {
    "name": "nginx-configuration",
    "namespace": "kube-system",
    "selfLink": "/api/v1/namespaces/kube-system/configmaps/nginx-
configuration",
     "creationTimestamp": "2018-03-20T07:10:18Z",
    "labels": {
       "app": "ingress-nginx"
    },
     "annotations": {
       "kubectl.kubernetes.io/last-applied-configuration": "{\"
apiVersion\":\"v1\",\"data\":{\"proxy-body-size\":\"20m\"},\"kind\":
\"ConfigMap\",\"metadata\":{\"annotations\":{},\"labels\":{\"app\":
\"ingress-nginx\"},\"name\":\"nginx-configuration\",\"namespace\":\"
kube-system\"}}\n"
 },
 "data": {
    "proxy-body-size": "20m",
```

```
"enable-vts-status": "true"
}
```

4. 本地访问 Ingress Nginx 监控控制台。

道 说明:

鉴于安全考虑,默认 VTS Port 并不对外开放,这里通过 port-forward 方式来访问。

```
root@master # kubectl port-forward nginx-ingress-controller-
79877595c8-78gq8 -n kube-system 18080
Forwarding from 127.0.0.1:18080 -> 18080
Handling connection for 18080
```

5. 通过 http://localhost:18080/nginx_status 来访问 VTS 监控控制台。

Nginx Vhost Traffic Status

Server main Connections Shared memory Requests Version Uptime active reading writing waiting accepted handled Total Reg/s Host maxSize usedSize usedN name nginx-ingress-controller-79877595c8-78gq8 1.13.7 32m 41s 93566 93566 1428 1 vhost_traffic_status 10.0 MiB 2.4 KiB 7 0 6 Server zones Requests Responses Traffic Cache Total Requests Hesponses Team Miss Bypass Expired State Updating Revisition Revisition No 660 1 0ms 0 660 0 0 0 660 Upstreams upstream-default-backend Response Time Weight MaxFails FailTimeout Requests Mesponses Total Req/s Time 1xx 2xx 3xx 4xx 5xx Total Sent Rcv/s Server 0 172.16.3.6:8080 up 0ms 1 0 0 0 B update interval: 1 📀 sec JSON | GITHUB

1.16.4 Ingress 支持

在 Kubernetes 集群中, Ingress是授权入站连接到达集群服务的规则集合,为您提供七层负载均衡能力。您可以给 Ingress 配置提供外部可访问的 URL、负载均衡、SSL、基于名称的虚拟主机等。

前置条件

为了测试复杂路由服务,本例中创建一个 nginx 的示例应用,您需要事先创建 nginx 的 deployment,然后创建多个 Service,用来观察路由的效果。实际测试请替换成自己的服务。

```
root@master # kubectl run nginx --image=registry.cn-hangzhou.aliyuncs.
com/acs/netdia:latest
root@master # kubectl expose deploy nginx --name=http-svc --port=80 --
target-port=80
root@master # kubectl expose deploy nginx --name=http-svcl --port=80
--target-port=80
```

```
root@master # kubectl expose deploy nginx --name=http-svc2 --port=80
--target-port80
root@master # kubectl expose deploy nginx --name=http-svc3 --port=80
--target-port=80
```

简单的路由服务

通过以下命令创建一个简单的 Ingress,所有对 /svc 路径的访问都会被路由到名为 http-svc 的服

务。nginx.ingress.kubernetes.io/rewrite-target: /会将/svc路径重定向到后端服务

能够识别的/路径上面。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
 name: simple
 annotations:
   nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 rules:
  - http:
      paths:
      - path: /svc
        backend:
          serviceName: http-svc
          servicePort: 80
EOF
root@master # kubectl get ing
                HOSTS
                               ADDRESS
                                                 PORTS
                                                           AGE
NAME
simple
                *
                               101.37.192.211
                                                 80
                                                           11s
```

现在访问 http://101.37.192.211/svc 即可访问到 Nginx 服务。

基于域名的简单扇出路由

如果您有多个域名对外提供不同的服务,您可以生成如下的配置达到一个简单的基于域名的扇出效

果。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: simple-fanout
spec:
 rules:
  - host: foo.bar.com
   http:
     paths:
      - path: /foo
        backend:
          serviceName: http-svc1
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
```

```
- host: foo.example.com
    http:
      paths:
       - path: /film
        backend:
          serviceName: http-svc3
          servicePort: 80
EOF
root@master # kubectl get ing
NAME
                HOSTS
                               ADDRESS
                                                 PORTS
                                                            AGE
                               101.37.192.211
simple-fanout
                 *
                                                 80
                                                            11s
```

这时您可以通过 http://foo.bar.com/foo 访问到 http-svc1 服务;通过 http://foo.bar .com/bar 访问到 http-svc2 服务;通过 http://foo.example.com/film 访问到 httpsvc3 服务。



- 如果是生产环境,您需要将您的这个域名指向上面返回的 ADDRESS 101.37.192.211。
- 如果是测试环境测试,您可以修改 hosts 文件添加一条域名映射规则。

101.37.192.211 foo.bar.com 101.37.192.211 foo.example.com

简单路由默认域名

如果您没有域名地址也没有关系,容器服务为 Ingress 服务绑定了一个默认域名,您可以通过这个 域名来访问服务。域名的格式如下:*.[cluster-id].[region-id].alicontainer.com。 您可以直接在控制台集群基本信息页获取到该地址。

您可以通过下面的配置借助该默认域名暴露两个服务。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
 name: shared-dns
spec:
 rules:
  - host: foo.[cluster-id].[region-id].alicontainer.com ##替换为您集群
默认的服务访问域名
   http:
     paths:
      - path: /
       backend:
         serviceName: http-svc1
         servicePort: 80
 - host: bar.[cluster-id].[region-id].alicontainer.com ##替换为您集群
默认的服务访问域名
   http:
     paths:
     - path: /
```

```
backend:
          serviceName: http-svc2
          servicePort: 80
EOF
root@master # kubectl get ing
NAME
                HOSTS
                              ADDRESS
                                               PORTS
                                                          AGE
shared-dns
            foo.[cluster-id].[region-id].alicontainer.com,bar.[
cluster-id].[region-id].alicontainer.com
                                                      47.95.160.171
80
          40m
```

这时您可以通过 http://foo.[cluster-id].[region-id].alicontainer.com/ 访问到 http-svc1 服务;通过 http://bar.[cluster-id].[region-id].alicontainer.com 访 问到 http-svc2 服务。

配置安全的路由服务

支持多证书管理,为您的服务提供安全防护。

1. 准备您的服务证书。

如果没有证书,可以通过下面的方法生成测试证书。

域名与您的 Ingress 配置要一致。

root@master # openssl req -x509 -nodes -days 365 -newkey rsa:2048 keyout tls.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"

上面命令会生成一个证书文件 tls.crt、一个私钥文件tls.key。

然后用该证书和私钥创建一个名为foo.bar 的 Kubernetes Secret。创建 Ingress 时需要引用

这个 Secret。

```
root@master # kubectl create secret tls foo.bar --key tls.key --cert
tls.crt
```

2. 创建一个安全的 Ingress 服务。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
    name: tls-fanout
spec:
    tls:
        - hosts:
        - foo.bar.com
        secretName: foo.bar
rules:
        - host: foo.bar.com
        http:
            paths:</pre>
```

```
- path: /foo
        backend:
          serviceName: http-svcl
          servicePort: 80
       path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
EOF
root@master # kubectl get ing
NAME
                HOSTS
                               ADDRESS
                                                 PORTS
                                                            AGE
                *
                               101.37.192.211
tls-fanout
                                                 80
                                                            11s
```

按照 基于域名的简单扇出路由 中的注意事项,配置 hosts 文件或者设置域名来访问该 tls 服务。

您可以通过 http://foo.bar.com/foo 访问到 http-svc1 服务;通过 http://foo.bar. com/bar 访问到 http-svc2 服务。

您也可以通过 HTTP 的方式访问该 HTTPS 的服务。Ingress 默认对配置了 HTTPS 的 HTTP 访问重定向到 HTTPS 上面。所以访问 http://foo.bar.com/foo 会被自动重定向到 https ://foo.bar.com/foo。

通过 Kubernetes Dashboard 部署 Ingress

1. 将下面的 yml code 保存到 nginx-ingress.yml 文件中。

```
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
   name: simple
spec:
   rules:
    - http:
        paths:
        - path: /svc
        backend:
        serviceName: http-svc
        servicePort: 80
```

- 2. 登录容器服务管理控制合,在 Kubernetes 菜单下,在集群列表页面中,单击目标集群右侧的控制合,进入 Kubernetes Dashboard 页面。
- 登录容器服务管理控制台,在Kubernetes菜单下,在集群列表页面中,单击目标集群右侧的控制台,进入Kubernetes Dashboard页面。
- 4. 单击创建,开始创建应用。



- 5. 单击上传 YAML 或 JSON 文件。选择刚才保存的 nginx-ingress.yml 文件。
- 6. 单击上传。

这样就创建了一个 Ingress 的七层代理路由到 http-svc 服务上。

7. 在 Kubernetes Dashboard 上定位到 default 命名空间,选择访问权。

可以看到您刚刚创建的 Ingress 资源及其访问地址 http://118.178.174.161/svc。

<	服务发现与负载均衡 访问权			+ 创建
集群	访问权			Ŧ
命名空间	名称 ≑ 2	入口	巳创建 ≑	
节点	simple	8.174.161 ∅	2018-01-24 18:29:30	
行人化行储仓	gateway	.113.231	2018-01-23 11:57:52	
存储类				
命名空间				
default 👻				
概览				
工作负载				
守护进程集				
部署				
任务				
容器组				
副本集				
副本控制鑽				
有4/20mm/中果 服务发现与负载均衡				
访问权 1				

8. 打开浏览器输入该地址即可访问前面创建的 http-svc 服务。

1.17 存储管理

1.17.1 概述

容器服务支持 kubernetes Pod 自动绑定阿里云云盘、NAS、 OSS 存储服务。

目前,支持静态存储卷和动态存储卷。每种数据卷的支持情况如下:

阿里云存储	静态数据卷	动态数据卷
阿里云云盘	您可以通过以下两种方式使用 云盘静态存储卷: • 直接通过 volume 使用 • 通过 PV/PVC 使用	支持
阿里云 NAS	 NAS 静态存储卷又支持下面两种使用方式: 通过 flexvolume 插件使用 通过 volume 方式使用 使用 PV/PVC 通过 Kubernetes 的 NFS 驱动使用 	支持
阿里云 OSS	 您可以通过以下两种方式使用 OSS 静态存储卷: 直接使用 volume 方式 使用 PV/PVC 	不支持

1.17.2 使用阿里云云盘

您可以在阿里云容器服务 Kubernetes 集群中使用阿里云云盘存储卷。

目前,阿里云云盘提供两种 Kubernetes 挂载方式:

• 静态存储卷

您可以通过以下两种方式使用云盘静态存储卷:

- 直接通过volume使用
- 通过 PV/PVC 使用
- 动态存储卷

静态存储卷

您可以直接通过 volume 使用阿里云云盘存储卷或者通过 PV/PVC 使用阿里云云盘存储卷。

前提条件

使用云盘数据卷之前,您需要先在 ECS 管理控制台上创建云盘。有关如何创建云盘,参见创建云

盘。

使用说明

- 云盘为非共享存储,只能同时被一个 pod 挂载。
- 使用云盘存储卷前需要先申请一个云盘,并获得磁盘 ID。参见 创建云盘。
- volumeId: 表示所挂载云盘的磁盘 ID; volumeName、PV Name要与之相同。
- 集群中只有与云盘在同一个可用区(Zone)的节点才可以挂载云盘。

直接通过 volume 使用

使用disk-deploy.yaml文件创建 Pod。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: nginx-disk-deploy
spec:
 replicas: 1
 template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-flexvolume-disk
        image: nginx
        volumeMounts:
          - name: "d-bp1j17ifxfasvts3tf40"
            mountPath: "/data"
      volumes:
        - name: "d-bp1j17ifxfasvts3tf40"
          flexVolume:
            driver: "alicloud/disk"
            fsType: "ext4"
            options:
              volumeId: "d-bp1j17ifxfasvts3tf40"
```


如果您所使用的集群是多 AZ 的模式,使用云盘的时候需要进行 AZ 调度,即把 Pod 调度到与云盘 相同的 AZ。

具体配置为在 yaml 中对 pod 添加 nodeSelector,如下:

```
nodeSelector:
    failure-domain.beta.kubernetes.io/zone: cn-hangzhou-b
```

通过 PV/PVC 使用

步骤1创建云盘类型的PV

您可以使用 yaml 文件或者控制台界面创建云盘类型的 PV。

通过 yaml 文件创建 PV

使用 disk-pv.yaml 文件创建 PV。

送明:

pv name 要与阿里云盘 ID 相同。

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: d-bp1j17ifxfasvts3tf40
 labels:
    failure-domain.beta.kubernetes.io/zone: cn-hangzhou-b
    failure-domain.beta.kubernetes.io/region: cn-hangzhou
spec:
 capacity:
   storage: 20Gi
 storageClassName: disk
 accessModes:
    - ReadWriteOnce
  flexVolume:
   driver: "alicloud/disk"
    fsType: "ext4"
    options:
      volumeId: "d-bp1j17ifxfasvts3tf40"
```

通过控制台界面创建云盘数据卷

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 存储,进入数据卷列表页面。
- 3. 选择所需的集群,单击页面右上角的创建。

容器服务	数据卷列表							刷新创建
Kubernetes Swarm	集群: k8s-cluster 🔻							4
生形	名称	总量	访问模式	回收策略	状态	存储类型	创建时间	攝作
节点	d-bp1gyzplgtreo4zpvwda	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-efficiency	2018-04-23 14:12:29	删除
存储 2								

- 4. 在创建数据卷对话框中, 配置数据卷的相关参数。
 - 数据卷类型:本示例中为云盘。
 - 数据卷名:创建的数据卷的名称。数据卷名必须和 云盘 ID 一致。
 - 数据卷总量:所创建数据卷的容量。注意不能超过磁盘容量。
 - 访问模式:默认为 ReadWriteOnce。
 - 云盘 ID:您可以选择与集群属于相同地域和可用区下处于待挂载状态的云盘。

- 文件系统类型:您可以选择以什么数据类型将数据存储到云盘上,支持的类型包括 ext4、ext3、xfs、vfat。默认为 ext4。
- 标签:为该数据卷添加标签。

创建数据卷		\times
数据卷类型:	● 云盘 ○ NAS ○ OSS	
访问模式:	ReadWriteOnce	
云盘ID:	d-l 选择云盘	
文件系统类型:	ext4	7
标签:	● 添加标签	
		创建取消

5. 完成配置后,单击创建。

步骤 2 创建 PVC

使用 disk-pvc.yaml 文件创建 PVC。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
   name: pvc-disk
spec:
    accessModes:
        - ReadWriteOnce
   storageClassName: disk
   resources:
        requests:
        storage: 20Gi
```

步骤 3 创建 Pod

使用disk-pod.yaml文件创建 pod。

```
apiVersion: v1
kind: Pod
metadata:
    name: "flexvolume-alicloud-example"
spec:
```

```
containers:
    - name: "nginx"
    image: "nginx"
    volumeMounts:
        - name: pvc-disk
        mountPath: "/data"
volumes:
    - name: pvc-disk
    persistentVolumeClaim:
        claimName: pvc-disk
```

动态存储卷

动态存储卷需要您手动创建 StorageClass,并在PVC中通过 storageClassName 来指定期望的云盘*****

类型。

创建 StorageClass

```
kind: StorageClass
apiVersion: storage.k8s.io/vlbetal
metadata:
    name: alicloud-disk-common-hangzhou-b
provisioner: alicloud/disk
parameters:
    type: cloud_ssd
    regionid: cn-hangzhou
    zoneid: cn-hangzhou-b
```

参数说明:

- provisioner: 配置为 alicloud/disk,标识使用阿里云云盘 Provsioner 插件创建。
- type:标识云盘类型,支持 cloud、cloud_efficiency、cloud_ssd、available 四种类型;其中 available 会对高效、SSD、普通云盘依次尝试创建,直到创建成功。
- regionid:期望创建云盘的区域。
- zoneid: 期望创建云盘的可用区。

创建服务

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
    name: disk-common
spec:
    accessModes:
        - ReadWriteOnce
    storageClassName: alicloud-disk-common-hangzhou-b
    resources:
        requests:
        storage: 20Gi
----
kind: Pod
apiVersion: v1
metadata:
```

```
name: disk-pod-common
spec:
    containers:
    - name: disk-pod
    image: nginx
    volumeMounts:
        - name: disk-pvc
        mountPath: "/mnt"
    restartPolicy: "Never"
    volumes:
        - name: disk-pvc
        persistentVolumeClaim:
        claimName: disk-common
```

默认选项

在多 AZ 的集群中,需要您手动创建上述 StorageClass,这样可以更准确的定义所需要云盘的 AZ 信息;

集群默认提供了下面几种 StorageClass,可以在单 AZ 类型的集群中使用。

- alicloud-disk-common:普通云盘。
- alicloud-disk-efficiency:高效云盘。
- alicloud-disk-ssd:SSD云盘。
- alicloud-disk-available:提供高可用选项,先试图创建高效云盘;如果相应AZ的高效云盘资源售尽,再试图创建SSD盘;如果SSD售尽,则试图创建普通云盘。

使用云盘创建多实例StatefulSet

使用 volumeClaimTemplates 的方式来创建,这样会动态创建多个 PVC 和 PV 并绑定。

```
apiVersion: v1
kind: Service
metadata:
 name: nginx
 labels:
   app: nginx
spec:
 ports:
  - port: 80
   name: web
 clusterIP: None
 selector:
   app: nginx
apiVersion: apps/v1beta2
kind: StatefulSet
metadata:
 name: web
spec:
 selector:
   matchLabels:
      app: nginx
 serviceName: "nginx"
```

```
replicas: 2
template:
  metadata:
    labels:
      app: nginx
  spec:
    containers:
     - name: nginx
      image: nginx
      ports:
       containerPort: 80
        name: web
      volumeMounts:
      - name: disk-common
        mountPath: /data
volumeClaimTemplates:
- metadata:
   name: disk-common
  spec:
    accessModes: [ "ReadWriteOnce" ]
    storageClassName: "alicloud-disk-common"
    resources:
      requests:
        storage: 10Gi
```

1.17.3 安装插件

通过下面的 yaml 配置部署阿里云 Kubernetes 存储插件。

如果您的 Kubernetes 集群是在 2018 年 2 月 6 日之前创建的,那么您使用数据卷之间需要先安装 阿里云 Kubernetes 存储插件;如果您的 Kubernetes 集群是在 2018 年 2 月 6 日之后创建的,那 么您可以直接使用数据卷,不需要再安装阿里云 Kubernetes 存储插件。

使用限制

目前支持 CentOS 7 操作系统。

注意事项

- 使用 flexvolume 需要 kubelet 关闭 --enable-controller-attach-detach 选项。默认阿 里云 Kubernetes 集群已经关闭此选项。
- 在 kube-system 用户空间部署 flexvolume。

验证安装完成:

在 master 节点上:

 执行命令:kubectl get pod -n kube-system | grep flexvolume。输出若干(节点 个数)Running 状态的 Pod 列表。 执行命令:kubectl get pod -n kube-system | grep alicloud-diskcontroller。输出一个 Running 状态的 Pod 列表。

安装步骤

安装 Flexvolume:

```
apiVersion: apps/v1 # for versions before 1.8.0 use extensions/v1beta1
kind: DaemonSet
metadata:
 name: flexvolume
 namespace: kube-system
  labels:
   k8s-volume: flexvolume
spec:
  selector:
   matchLabels:
     name: acs-flexvolume
  template:
    metadata:
      labels:
        name: acs-flexvolume
    spec:
      hostPID: true
      hostNetwork: true
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      containers:
      - name: acs-flexvolume
        image: registry.cn-hangzhou.aliyuncs.com/acs/flexvolume:v1.9.7
-42e8198
        imagePullPolicy: Always
        securityContext:
          privileged: true
        env:
        - name: ACS_DISK
          value: "true"
        - name: ACS_NAS
          value: "true"
        - name: ACS_OSS
          value: "true"
        resources:
          limits:
            memory: 200Mi
          requests:
            cpu: 100m
            memory: 200Mi
        volumeMounts:
        - name: usrdir
          mountPath: /host/usr/
         name: etcdir
          mountPath: /host/etc/
         name: logdir
          mountPath: /var/log/alicloud/
      volumes:
      - name: usrdir
        hostPath:
```

```
path: /usr/
- name: etcdir
hostPath:
path: /etc/
- name: logdir
hostPath:
path: /var/log/alicloud/
```

安装 Disk Provisioner:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-common
provisioner: alicloud/disk
parameters:
 type: cloud
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-efficiency
provisioner: alicloud/disk
parameters:
 type: cloud_efficiency
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-ssd
provisioner: alicloud/disk
parameters:
 type: cloud_ssd
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-available
provisioner: alicloud/disk
parameters:
 type: available
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: alicloud-disk-controller-runner
rules:
  - apiGroups: [""]
   resources: ["persistentvolumes"]
   verbs: ["get", "list", "watch", "create", "delete"]
  - apiGroups: [""]
   resources: ["persistentvolumeclaims"]
   verbs: ["get", "list", "watch", "update"]
  - apiGroups: ["storage.k8s.io"]
   resources: ["storageclasses"]
   verbs: ["get", "list", "watch"]
  - apiGroups: [""]
   resources: ["events"]
   verbs: ["list", "watch", "create", "update", "patch"]
___
```

```
apiVersion: v1
kind: ServiceAccount
metadata:
 name: alicloud-disk-controller
 namespace: kube-system
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: run-alicloud-disk-controller
subjects:
  - kind: ServiceAccount
    name: alicloud-disk-controller
    namespace: kube-system
roleRef:
 kind: ClusterRole
 name: alicloud-disk-controller-runner
 apiGroup: rbac.authorization.k8s.io
kind: Deployment
apiVersion: extensions/vlbetal
metadata:
 name: alicloud-disk-controller
 namespace: kube-system
spec:
 replicas: 1
  strategy:
    type: Recreate
  template:
   metadata:
      labels:
        app: alicloud-disk-controller
    spec:
      tolerations:
      - effect: NoSchedule
        operator: Exists
        key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      nodeSelector:
         node-role.kubernetes.io/master: ""
      serviceAccount: alicloud-disk-controller
      containers:
        - name: alicloud-disk-controller
          image: registry.cn-hangzhou.aliyuncs.com/acs/alicloud-disk-
controller:v1.9.3-ed710ce
          volumeMounts:
            - name: cloud-config
              mountPath: /etc/kubernetes/
            - name: logdir
              mountPath: /var/log/alicloud/
      volumes:
        - name: cloud-config
          hostPath:
            path: /etc/kubernetes/
        - name: logdir
          hostPath:
```

path: /var/log/alicloud/

1.17.4 使用阿里云 NAS

您可以在容器服务 Kubernetes 集群中使用阿里云 NAS 数据卷。

目前阿里云 NAS 提供两种 Kubernetes 挂载方式:

• 静态存储卷

其中,静态存储卷又支持下面两种使用方式:

- 通过 flexvolume 插件使用
 - 通过 volume 方式使用
 - 使用 PV/PV
- 通过 Kubernetes 的 NFS 驱动使用
- 动态存储卷

前提条件

使用 NAS 数据卷之前,您需要先在 NAS 管理控制台上创建文件系统,并在文件系统中添加 Kubernetes 集群的挂载点。创建的 NAS 文件系统需要和您的集群位于同一 VPC。

静态存储卷

您可以通过阿里云提供的 flexvolume 插件使用阿里云 NAS 文件存储服务或者通过 Kubernetes 的 NFS 驱动使用阿里云 NAS 文件存储服务。

通过 flexvolume 插件使用

使用 flexvolume 插件,您可以通过 volume 方式使用阿里云 NAS 数据卷或者通过 PV/PVC 方式使用阿里云 NAS 数据卷。

送 说明:

- NAS 为共享存储,可以同时为多个 Pod 提供共享存储服务。
- server:为 NAS 数据盘的挂载点。
- path:为连接 NAS 数据卷的挂载目录,支持挂载 NAS 子目录,且当子目录不存在时,会自动 创建子目录并挂载。
- vers:定义 nfs 挂载协议的版本号,支持 4.0。

mode:定义挂载目录的访问权限,注意挂载 NAS 盘根目录时不能配置挂载权限。当 NAS 盘中数据量很大时,配置 mode 会导致执行挂载非常慢,甚至挂载失败。

通过 volume 方式使用

使用 nas-deploy.yaml 文件创建 Pod。

```
apiVersion: v1
kind: Pod
metadata:
 name: "flexvolume-nas-example"
spec:
  containers:
    - name: "nginx"
      image: "nginx"
      volumeMounts:
        - name: "nas1"
         mountPath: "/data"
 volumes:
    - name: "nas1"
      flexVolume:
        driver: "alicloud/nas"
        options:
          server: "0cd8b4a576-grs79.cn-hangzhou.nas.aliyuncs.com"
          path: "/k8s"
          vers: "4.0"
```

使用 PV/PVC

步骤1创建PV

您可以使用 yaml 文件或者通过阿里云容器服务控制台界面创建 NAS 数据卷。

• 使用 yaml 文件创建 PV

使用 nas-pv.yaml 文件创建 PV。

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: pv-nas
spec:
  capacity:
    storage: 5Gi
  storageClassName: nas
  accessModes:
    - ReadWriteMany
  flexVolume:
    driver: "alicloud/nas"
    options:
      server: "Ocd8b4a576-uih75.cn-hangzhou.nas.aliyuncs.com"
      path: "/k8s"
      vers: "4.0"
```

• 通过控制台界面创建 NAS 数据卷

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 存储,进入数据卷列表页面。
- 3. 选择所需的集群,单击页面右上角的创建。

容器服务	数据卷列表							刷新创建	2
Kubernetes Swarm	集群: k8s-cluster 🔻 3							4	
#群	名称	总量	访问模式	回收策略	状态	存储类型	创建时间	5	₿ f fE
节点	d-bp1gyzplgtreo4zpvwda	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-efficiency	2018-04-23 14:12:29	m	制除
存储 2									

- 4. 在创建数据卷对话框中, 配置数据卷的相关参数。
 - 数据卷类型:本示例中为NAS。
 - 数据卷名:创建的数据卷的名称。数据卷名在集群内必须唯一。本例为 pv-nas。
 - 数据卷总量:所创建数据卷的容量。注意不能超过磁盘容量。
 - 访问模式:默认为 ReadWriteMany。
 - 挂载点域名:集群在 NAS 文件系统中挂载点的挂载地址。
 - 子目录: NAS 路径下的子目录, 以 / 开头, 设定后数据卷将挂载到指定的子目录。
 - 如果 NAS 根目录下没有此子目录,会默认创建后再挂载。
 - 您可以不填此项,默认挂载到 NAS 根目录。
 - 权限:设置挂载目录的访问权限,例如:755、644、777等。
 - 只有挂载到 NAS 子目录时才能设置权限,挂载到根目录时不能设置。
 - 您可以不填此项,默认权限为 NAS 文件原来的权限。
 - 标签:为该数据卷添加标签。

创建数据卷		\times
数据卷类型:	◯ 云盘 ● NAS ◯ OSS	
数据卷名:	pv-nas	
总量:	20Gi	
访问模式:	ReadWriteMany	
挂载点域名:	.nas.aliyuncs.com	
子目录:	如:/xxx,默认为根目录	
权限:	如:755,默认为NAS权限	
标签:	●添加标签	
	创建	取消

5. 完成配置后,单击创建。

步骤 2 创建 PVC

使用 nas-pvc.yaml 文件创建 PVC。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
   name: pvc-nas
spec:
    accessModes:
        - ReadWriteMany
   storageClassName: nas
   resources:
        requests:
        storage: 5Gi
```

步骤 3 创建 Pod

使用nas-pod.yaml文件创建 pod。

```
apiVersion: v1
kind: Pod
```

```
metadata:
    name: "flexvolume-nas-example"
spec:
    containers:
        - name: "nginx"
            image: "nginx"
            volumeMounts:
                - name: pvc-nas
                      mountPath: "/data"
volumes:
        - name: pvc-nas
            persistentVolumeClaim:
                claimName: pvc-nas
```

通过 Kubernetes 的 NFS 驱动使用

步骤 1 创建 NAS 文件系统

登录 文件存储管理控制台, 创建一个 NAS 文件系统。

登录 文件存储管理控制台,创建一个 NAS 文件系统。



创建的 NAS 文件系统需要和您的集群位于同一地域。

假设您的挂载点为 055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com。

步骤 2 创建 PV

您可以使用编排模板或者通过阿里云容器服务控制台界面创建 NAS 数据卷。

• 使用编排模板

使用nas-pv.yam1文件创建 PV。

执行以下命令创建一个类型为 NAS 的 PesistentVolume。

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: PersistentVolume
metadata:
   name: nas
spec:
   capacity:
    storage: 8Gi
   accessModes:
        - ReadWriteMany
   persistentVolumeReclaimPolicy: Retain
   nfs:
        path: /
        server: 055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com
EOF</pre>
```

• 通过控制台界面创建 NAS 数据卷

具体操作参见使用PV/PVC中关于通过控制台界面创建 NAS 数据卷的内容。

步骤 2 创建 PVC

创建一个 PersistentVolumeClaim 来请求绑定该 PersistentVolume。

```
root@master # cat << EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
    name: nasclaim
spec:
    accessModes:
        - ReadWriteMany
    resources:
        requests:
        storage: 8Gi
EOF</pre>
```

步骤 3 创建 Pod

创建一个应用来申明挂载使用该数据卷。

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
     name: mypod
spec:
     containers:
       - name: myfrontend
         image: registry.aliyuncs.com/spacexnice/netdia:latest
         volumeMounts:
         - mountPath: "/var/www/html"
           name: mypd
     volumes:
       - name: mypd
         persistentVolumeClaim:
           claimName: nasclaim
EOF
```

至此,您就将 NAS 远程文件系统挂载到了您的 Pod 应用当中了。

动态存储卷

使用动态 NAS 存储卷需要您手动安装驱动插件,并配置 NAS 挂载点。

安装插件

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
   name: alicloud-nas
provisioner: alicloud/nas
---
apiVersion: v1
```

```
kind: ServiceAccount
metadata:
  name: alicloud-nas-controller
 namespace: kube-system
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: run-alicloud-nas-controller
subjects:
  - kind: ServiceAccount
    name: alicloud-nas-controller
    namespace: kube-system
roleRef:
  kind: ClusterRole
  name: alicloud-disk-controller-runner
  apiGroup: rbac.authorization.k8s.io
kind: Deployment
apiVersion: extensions/vlbetal
metadata:
  name: alicloud-nas-controller
  namespace: kube-system
spec:
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: alicloud-nas-controller
    spec:
      tolerations:
      - effect: NoSchedule
        operator: Exists
        key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      nodeSelector:
         node-role.kubernetes.io/master: ""
      serviceAccount: alicloud-nas-controller
      containers:
        - name: alicloud-nas-controller
          image: registry.cn-hangzhou.aliyuncs.com/acs/alicloud-nas-
controller:v1.8.4
          volumeMounts:
          - mountPath: /persistentvolumes
            name: nfs-client-root
          env:
            - name: PROVISIONER_NAME
              value: alicloud/nas
            - name: NFS_SERVER
              value: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            - name: NFS_PATH
              value: /
      volumes:
        - name: nfs-client-root
          nfs:
            server: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
```

```
path: /
```

使用动态存储卷

```
apiVersion: apps/vlbetal
kind: StatefulSet
metadata:
 name: web
spec:
  serviceName: "nginx"
  replicas: 2
  volumeClaimTemplates:
  - metadata:
      name: html
    spec:
      accessModes:
        - ReadWriteOnce
      storageClassName: alicloud-nas
      resources:
        requests:
          storage: 2Gi
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
       - name: nginx
        image: nginx:alpine
        volumeMounts:
        - mountPath: "/usr/share/nginx/html/"
          name: html
```

1.17.5 使用阿里云 OSS

您可以在阿里云容器服务 Kubernetes 集群中使用阿里云 OSS 数据卷。

目前, 仅支持 OSS 静态存储卷, 不支持 OSS 动态存储卷。您可以通过以下方式使用 OSS 静态存储卷:

- 直接使用 volume 方式
- 使用 PV/PVC

前提条件

使用 OSS 静态存储卷之前,您需要先在 OSS 管理控制台上创建 Bucket。

使用说明

- OSS 为共享存储,可以同时为多个 Pod 提供共享存储服务。
- bucket:目前只支持挂载 Bucket,不支持挂载 Bucket 下面的子目录或文件。
- url: OSS endpoint, 挂载 OSS 的接入域名。

- akld: 用户的 access id 值。
- akSecret:用户的 access secret 值。
- otherOpts: 挂载 OSS 时支持定制化参数输入,格式为: -o *** -o ***。

注意事项

如果您的 Kubernetes 集群是在 2018 年 2 月 6 日之前创建的,那么您使用数据卷之间需要先安装安装插件。使用 OSS 数据卷必须在部署 flexvolume 服务的时候创建 Secret,并输入 AK 信息。

使用 OSS 静态卷

直接使用 volume 方式

使用oss-deploy.yaml 文件创建 Pod。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: nginx-oss-deploy
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-flexvolume-oss
        image: nginx
        volumeMounts:
          - name: "oss1"
            mountPath: "/data"
      volumes:
        - name: "oss1"
          flexVolume:
            driver: "alicloud/oss"
            options:
              bucket: "docker"
              url: "oss-cn-hangzhou.aliyuncs.com"
              akId: ***
              akSecret: ***
              otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

使用 PV/PVC(目前不支持动态 pv)

步骤1创建PV

您可以使用 yaml 文件或者通过容器服务控制台界面创建 PV。

使用 yaml 文件创建 PV

使用oss-pv.yam1文件创建 PV。

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: pv-oss
spec:
 capacity:
   storage: 5Gi
 accessModes:
    - ReadWriteMany
  storageClassName: oss
  flexVolume:
    driver: "alicloud/oss"
    options:
      bucket: "docker"
      url: "oss-cn-hangzhou.aliyuncs.com"
      akId: ***
      akSecret: ***
      otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

通过控制台界面创建 OSS 数据卷

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群 > 存储,进入数据卷列表页面。
- 3. 选择所需的集群,单击页面右上角的创建。

容器服务	数据卷列表							刷新创建
Kubernetes Swarm	集群: k8s-cluster ▼ 3							4
	名称	总量	访问模式	回收策略	状态	存储类型	创建时间	摄作
节点	d-bp1gyzplgtreo4zpvwda	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-efficiency	2018-04-23 14:12:29	删除
存储 2								

- 4. 在创建数据卷对话框中,配置数据卷的相关参数。
 - 数据卷类型:本示例中为 OSS。
 - 数据卷名:创建的数据卷的名称。数据卷名在集群内必须唯一。本例为 pv-oss。
 - 数据卷总量:所创建数据卷的容量。
 - 访问模式:默认为 ReadWriteMany。
 - AccessKey ID、AccessKey Secret:访问 OSS 所需的 AccessKey。
 - Bucket ID: 您要使用的 OSS bucket 的名称。单击选择Bucket,在弹出的对话框中选择所 需的 bucket 并单击选择。
 - 访问域名:如果 Bucket 和 ECS 实例位于不同地域(Region),请选择外网域名;如果位于相同地域,需要根据集群网络类型进行选择,若是 VPC 网络,请选择VPC域名,若是经典网络,请选择内网域名。

• 标签:为该数据卷添加标签。

创建数据卷	\times
数据卷类型:	◯ 云盘 ◯ NAS ◉ OSS
数据卷名:	pv-oss
总量:	20Gi
访问模式:	ReadWriteMany
AccessKey ID :	
AccessKey Secret :	
可选参数:	
	更多参数的填写格式可以 参考该文档
Bucket ID :	选择Bucket
访问域名:	◎ 内网域名 ◎ 外网域名 ◎ vpc域名 Ø
标签:	●添加标签
	创建取消

5. 完成配置后,单击创建。

步骤 2 创建 PVC

使用oss-pvc.yaml文件创建 PVC。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
   name: pvc-oss
spec:
   storageClassName: oss
   accessModes:
        - ReadWriteMany
   resources:
        requests:
```

storage: 5Gi

步骤 3 创建 Pod

使用oss-pod.yaml 创建 Pod。

```
使用 OSS 动态卷
```

目前暂不支持。

1.18 日志管理

1.18.1 概述

阿里云容器服务 Kubernetes 集群提供给您多种方式进行应用的日志管理。

- 通过阿里云容器服务提供的开源 Fluentd-pilot 项目,通过阿里云日志服务采集 Kubernetes 集群日志,您可以方便地使用日志服务采集应用日志,从而更好地利用阿里云日志服务提供给您的各种日志统计分析等功能。
- 为 Kubernetes 和日志服务配置 Log4JAppender。
- 利用 log-pilot + elasticsearch + kibana 搭建 kubernetes 日志解决方案。

1.18.2 查看集群日志

背景信息

您可以通过容器服务的简单日志服务查看集群的操作日志。

操作步骤

1. 登录容器服务管理控制台。

2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,进入 Kubernetes 集群列表页面。

3. 选择所需的集群并单击右侧的查看日志。

容器服务		集群列表				您最多可以创	建 5 个集群,每个集群最多可以添加 20 个	节点 刷新	创建 Kubernetes 集群 🛛 👻
Kubernetes	Swarm	常见问题: ② 如何创建集群 ③ 扩容和缩容集群	🖉 通过 kuber	tl 连接 Kubernetes 集群	⊘ 通过命令管理应用				
集群	1	名称 ¥							
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本		2 操作
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp1gnu8br4a	●运行中	2018-04-23 11:41:46	1.9.3	1	查理 <u>查看日志</u> 控制台 集群伸缩 更多▼

您可以查看该集群的操作信息。

集群日志信息: k8s-cluster ★ 返回集群列:	æ	刷新	
资源部署详细日志请参考: 资源线事件			
时间	信息		
2018-04-23 11:59:27	c98a72d1cf17549e28aeb94a09b5b1eb4 Set up k8s DNS configuration successfully		
2018-04-23 11:58:23	c98a72d1cf17549e28aeb94a09b5b1eb4 Stack CREATE completed successfully:		
2018-04-23 11:58:23	c98a72d1cf17549e28aeb94a09b5b1eb4 Start describeStackInfo		
2018-04-23 11:58:22	c98a72d1cf17549e28aeb94a09b5b1eb4 Start describeStackInfo		
2018-04-23 11:41:48	c98a72d1cf17549e28aeb94a09b5b1eb4 Successfully to CreateStack		
2018-04-23 11:41:48	c98a72d1cf17549e28aeb94a09b5b1eb4 Start to wait stack ready		
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4 Start to getNatGatway		
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4 Start to create cluster task		
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4 Start to CreateK8sCluster		
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4 Start to CreateStack		
2018-04-23 11:41:45	c98a72d1cf17549e28aeb94a09b5b1eb4 Start to validateCIDR		
2018-04-23 11:41:41	c98a72d1cf17549e28aeb94a09b5b1eb4 Start create cluster certificate		

1.18.3 通过阿里云日志服务采集 Kubernetes 集群日志

日志服务支持通过 Logtail 采集 Kubernetes 集群日志,本文主要介绍 Logtail 的 DaemonSet 部署步骤。

配置流程



步骤一部署 Kubernetes DaemonSet

步骤二 配置机器组

在日志服务控制台创建自定义标识机器组,后续该 Kubernetes 集群伸缩无需额外运维。

步骤三 创建采集配置

在日志服务控制台创建采集配置,所有采集均为服务端配置,无需本地配置。

步骤一 部署 Kubernetes DaemonSet

1. 连接到您的 Kubernetes 集群。

参见通过 kubectl 连接 Kubernetes 集群。

- **2.** 配置参数。
 - a. 单击下载 日志服务 YAML 文件模板,用 vi 编辑器打开。
 - b. 将 env 环境变量一节中所有的 \${your_xxxx} 替换成您的真实值。

参数	参数说明
----	------
\${your_region_name}	地域名称。请替换为您创建的日志服务 project 所在的地域。地域名称请参考 <i>Logtail</i> 安装所选region名。
-----------------------------	---
\${your_aliyun_user_id}	用户标识,请替换为您的阿里云主账号用户 ID。主账号用户 ID 为字符串形式。有关如何 查看 ID,参见用户组标识配置。
\${your_machine_group_name}	您集群的机器组标识,此部分由您自己命 名,取值范围为 [0-9a-zA-Z],具体请参考 自定义机器组。



说明:

- 您的主账号需要开启 AccessKey,参见五分钟快速入门。
- 请您不要修改模板中的 volumeMounts 和 volumes 部分,否则会造成 Logtail 无法正常工作。
- 您可以自定义配置 Logtail 容器的启动参数,只需保证如下几点即可:

- 启动时,必须具备3个环境变量:ALIYUN_LOGTAIL_USER_DEFINED_ID、

ALIYUN_LOGTAIL_USER_ID、ALIYUN_LOGTAIL_CONFIG。

- 必须将 Docker 的 Domain Socket 挂载到/var/run/docker.sock。
- 如果您需要采集其他容器或宿主机文件,需要将根目录挂载到 Logtail 容器的/ logtail_host 目录。
- 3. 部署 Logtail 的 DaemonSet。

示例:

您可以通过 kubect1 get ds -n kube-system 查看您的 Logtail agent 的运行状态。

步骤二 配置机器组

- 1. 参见五分钟快速入门开通日志服务并创建 Project。
- 2. 在日志服务控制台的机器组列表页面单击 创建机器组。
- **3.** 选择用户自定义标识,将您上一步配置的ALIYUN_LOGTAIL_USER_DEFINED_ID填入用户自定 义标识内容框中。

•		技家	Q 🛓 🔼	费用	т#			支持与服务	简体中文 🙆
k8s-log_c12bs	创建Logstore			×					志城: 华东1
Logstore列表	・Logstore名称: Logstore属性 -	kBs-stidout						查看Er	idpoint 创建
请输入Logstore名进行模糊查	* WebTracking:	Web Tracking Thttl: 实际性 连 印 年 2 社社 的第一人	C/Android/ADD						
Logstore名称		Web Hability が起こた (部数) 访问信息、默认关闭(部数)			お消費	日本派日本	(費模式 (投递	查询分析	銀行
config-operation-log	数据保存时间:	30 自定义数据保存时间支持1-3650天,如需要永久不 保存"	序储请开启"永久	3	e va	MaxCom	pute OSS	查询	修改]删除
	* Shard数目:	2 0 什么是分区 (Shard) ?				共有	「1条,每页显	示: 10条 。	< 1 × ×
	• 计费:	參考计费中心说明							
			》说 取消						

配置完成一分钟后,在机器组列表页面单击右侧的查看状态,即可看到已经部署 Logtail DaemonSet 节点的心跳状态。具体参见机器组配置中的查看状态。

步骤三 创建采集配置

请根据您的需求在控制台创建 Logtail 采集配置,采集配置步骤请参考:

- 容器文本文件#推荐#
- 容器标准输出 #推荐#
- 文本日志

默认宿主机根目录挂载到 Logtail 容器的/logtail_host 目录,配置路径时,您需要加上此前 缀。例如需要采集宿主机上/home/logs/app_log/目录下的数据,配置页面中日志路径设置 为/logtail_host/home/logs/app_log/。

- Syslog
- 自定义插件

其他操作

查看 Kubernetes 集群中 Logtail DaemonSet 状态

您可以执行命令 kubectl get ds -n kube-system 查看 Logtail 运行状态。



Logtail 默认的 namespace 为 kube-system。

如何调整 Logtail 资源限制

默认 Logtail 最多占用单核 40% CPU 和 200M 的内存,如需提升处理速度,需调整两部分参数:

- yaml 模板中 resources 下的 limits 和 requests。
- Logtail 启动配置文件,文件路径为 yaml 模板中的 ALIYUN_LOGTAIL_CONFIG 环境变量,修改 方式参见配置启动参数。

强制更新 Logtail DaemonSet

修改logtail-daemonset.yaml文件后,执行如下命令进行更新:

```
kubectl --namespace=kube-system delete ds logtail
kubectl apply -f ./logtail-daemonset.yaml
```



强制更新期间可能会出现数据重复。

查看 Logtail DaemonSet 的配置信息

kubectl describe ds logtail -n kube-system

查看 Logtail 的版本号信息、IP、启动时间等

```
示例如下:
```

```
[root@iZbpldsu6v77zfb40qfbiaZ ~]# kubectl get po -n kube-system -l k8s
-app=logtail
NAME
                READY
                           STATUS
                                     RESTARTS
                                                AGE
logtail-gb92k
                1/1
                          Running
                                     0
                                                 2h
logtail-wm7lw
                1/1
                          Running
                                     0
                                                 4d
[root@iZbpldsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-
system cat /usr/local/ilogtail/app_info.json
ł
   "UUID" : ""
   "hostname" : "logtail-gb92k",
   "instance_id" : "0EBB2B0E-0A3B-11E8-B0CE-0A58AC140402_172.20.4.
2_1517810940"
   "ip" : "172.20.4.2"
   "logtail_version" : "0.16.2",
```

```
"os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:
13 UTC 2017; x86_64",
    "update_time" : "2018-02-05 06:09:01"
}
```

查看 Logtail 的运行日志

Logtail 运行日志保存在/usr/local/ilogtail/ 目录下,文件名为ilogtail.LOG,轮转文件 会压缩存储为ilogtail.LOG.x.gz。

示例如下:

[root@iZbpldsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kubesystem tail /usr/local/ilogtail/ilogtail.LOG [2018-02-05 06:09:02.168693] [build/release64/sls/ [INFO] [9] ilogtail/LogtailPlugin.cpp:104] logtail plugin Resume:start [2018-02-05 06:09:02.168807] [INFO] [build/release64/sls/ [9] ilogtail/LogtailPlugin.cpp:106] logtail plugin Resume:success [2018-02-05 06:09:02.168822] [INFO] [build/release64/sls [9] start add existed check point /ilogtail/EventDispatcher.cpp:369] events, size:0 [2018-02-05 06:09:02.168827] [INFO] [9] [build/release64/sls /ilogtail/EventDispatcher.cpp:511] add existed check point events, success count:0 size:0 cache size:0 event size:0

重启某个 Pod 的 Logtail

示例如下:

```
[root@iZbpldsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-
system /etc/init.d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 9
stop success
[root@iZbpldsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-
system /etc/init.d/ilogtaild start
ilogtail is running
```

1.18.4 为 Kubernetes 和日志服务配置 Log4JAppender

Log4j 是 Apache 的一个开放源代码项目。Log4j 由三个重要组件构成:日志信息的优先级、日志信息的输出目的地、日志信息的输出格式。通过配置 Log4jAppender,您可以控制日志信息输送的目的地是控制台、文件、GUI 组件、甚至是套接口服务器、NT 的事件记录器、UNIX Syslog 守护进程等。

本文介绍在不需要修改应用代码的前提下,通过配置一个 yaml 文件,将阿里云容器服务 Kubernetes 集群中产生的日志输出到阿里云日志服务。此外,通过在 Kubernetes 集群上部署一个 示例 API 程序,来进行演示。

前提条件

• 您已经开通容器服务,并且创建了 Kubernetes 集群。

本示例中,创建的 Kubernetes 集群位于华东1地域。

• 启用 AccessKey 或 RAM,确保有足够的访问权限。本例中使用 AccessKey。

步骤 1 在阿里云日志服务上配置 Log4jAppender

- 1. 登录 日志服务管理控制台。
- 2. 单击页面右上角的创建 Project,填写 Project 的基本信息并单击确认进行创建。

本示例创建一个名为 k8s-log4j, 与 Kubernetes 集群位于同一地域(华东1)的 Project。

ren		
	说明	1

在配置时,一般会使用与 Kubernetes 集群位于同一地域的日志服务 Project。因为当 Kubernetes 集群和日志服务 Project 位于同一地域时,日志数据会通过内网进行传输,从而避免了因地域不一致而导致的数据传输外网带宽费用和耗时,从而实现实时采集、快速检索的最 佳实践。

创建Project	×
* Project名称: k8s-log4j	
注释: <u>k8s-log4j</u> demo	
不支持<>"最多包含64个字符 * 所属地域: 华东1 ▼	
确认	取消

创建完成后, k8s-log4j 出现在 project 列表下,单击该 project 名称,进入 project 详情页面。
 默认进入日志库页面,单击右上角的创建。

日志库 1	k8s-log4j返回	Project列表						地域:华东1
▼ LogHub - 实时采集	Logstore列表						查看Endpoin	t创建
[文档] 接入指南								
Logtail配置	请输入Logstore名进行模糊	禮 搜索						2
Lootail机器组	i i bith					日志消费模式		10.16
▼ LoaHub - 如时消费	Logstore名称	数据接入问导	监控 日志米集模式		日志消费	日志投递	查询分析	採作
[又归] /月贺作用								
消费组管理				ALLER AND COSTOLE, THE RIVE				
 Search/Analytics 								
快速查询						共有0条,每页5	眎:10条 « 、	1 > »
报警配置								

5. 填写日志库配置信息并单击确认。

本示例创建名为 k8s-logstore 的日志库。

创建Logstore		\times
* Landara (24).	k9a logatoro	
* Logstore 名称:	K85-logstore	
Logstore属性		
* WebTracking:	WebTracking功能支持快速采集各种浏览器以及 iOS/Android/APP访问信息,默认关闭(帮助)	
* 数据保存时间:	30 目前Loghub保存时间和索引已经统一,数据生命周 期以Loghub设置为准(单位:天)	
* Shard数目:	2 ▼ 什么是分区(Shard)?	
* 计费:	参考计费中心说明	
	确认	取消

6. 创建完毕后,页面会提示您创建数据接入向导。



7. 选择自定义数据下的log4jAppender,根据页面引导进行配置。

本示例使用了默认配置,您可根据日志数据的具体使用场景,进行相应的配置。

自定义数据							
	文本文件	SysLog	Web Tracking	API/SDK	✓ log4jAppender	Producer	
			Logtail自定	义播件			
							下一步

步骤 2 在 Kubernetes 集群中配置 log4j

本示例使用 demo-deployment 和 demo-Service 示例 yaml 文件进行演示。

1. 连接到您的 Kubernetes 集群。

具体操作参见SSH 访问 Kubernetes 集群 或 通过 kubectl 连接 Kubernetes 集群。

2. 获取*demo-deployment.yam*1文件并配置环境变量 JAVA_OPTS 设置 Kubernetes 集群日志的 采集。

demo-deployment.yaml 文件的示例编排如下。

```
apiVersion: apps/v1beta2
kind: Deployment
metadata:
   name: log4j-appender-demo-spring-boot
   labels:
        app: log4j-appender
spec:
    replicas: 1
   selector:
        matchLabels:
        app: log4j-appender
template:
        metadata:
```

其中:

- -Dproject:您所使用的阿里云日志服务 Project 的名称。本示例中为 k8s-log4j。
- -Dlogstore: 您所使用的阿里云日志服务 Logstore 的名称。本示例中为 k8s-logstore。
- -Dendpoint:日志服务的服务入口,用户需要根据日志 Project 所属的地域,配置自己的服务入口,参见服务入口进行查询。本示例中为 cn-hangzhou.log.aliyuncs.com。
- -Daccess_key_id: 您的 AccessKey ID。
- -Daccess_key: 您的 AccessKey Secret。
- **3.** 在命令行中执行以下命令, 创建 deployment。

```
kubectl create -f demo-deployment.yaml
```

4. 获取demo-Service.yaml文件,并运行以下命令创建 service。

您不需要修改demo-Service.yaml中的配置。

```
kubectl create -f demo-service.yaml
```

步骤 3 测试生成 Kubernetes 集群日志

您可以使用 kubect1 get 命令查看资源对象部署状况,等待 deployment 和 service 部署成功

后,执行 kubect1 get svc 查看 service 的外部访问 IP,即 EXTERNAL-IP。

```
$ kubectl get svc
NAME TYPE CLUSTER-IP EXTERNAL-IP
PORT(S) AGE
log4j-appender-demo-spring-boot-svc LoadBalancer 172.21.XX.XX
120.55.XXX.XXX 8080:30398/TCP 1h
```

在本示例中,通过运行login命令来测试生成 Kubernetes 集群日志。其中 K8S_SERVICE_IP 即为 EXTERNAL-IP。



您可以在 GitHub log4j-appender-demo 中查看完整的 API 集合。

curl http://\${K8S_SERVICE_IP}:8080/login?name=bruce

步骤4在阿里云日志服务中查看日志

登录 日志服务管理控制台。

进入对应的 Project 的详情页面,选择对应的日志库 k8s-logstore,并单击右侧的查询分析 - 查询,查看 Kubernetes 集群输出的日志,如下所示。

<	B k8s-logstore (眉于k8s	s-log4j)					分享	查询分析属性	另存为快速查询	另存为报警
标签列表	请输入关键字进行搜索				0	1分钟 🗸 🗸	2018-03-06 16:3	7:50 ~ 2018-03-06	8 16:38:50	搜索
i k8s-logstore 新建标签 🕑	0 37分50秒 37分559 原始日志 统计图	秒 38 ₉₀ 0 1表	089 389058	3 8 公10 8 3 8 公15 8 日志总条数:1	38分20秒 38分 查询状态:结果精确	258) 38 63 08	38935	8 38534	089 38 <u>6</u> 345	ŧ
	快速分析	<	时间▲▼	内容 ▼						4
	您还没有指定字段查 询,赶紧活加吧(查書 帮助)	1 Q	03-06 16:38:39	source: 172.20.3.8 topic: level: INFO location: com.allyun.log4jappenderder message: User login successfully.req thread: http-nio-6080-exec-7 time: 2018-03-06T08:38+0000	no. UserController.login(Us JestID=hIAgSGKGyA nam	erController.java:17) e=bruce				

日志的输出内容对应上面的命令。本示例演示了将示例应用产生的日志输出到阿里云日志服务的操作。通过这些操作,您可以在阿里云上配置 Log4JAppender,并通过阿里云日志服务,实现日志实时搜集、数据过滤、检索等高级功能。

1.18.5 利用 log-pilot + elasticsearch + kibana 搭建 kubernetes 日志解决方案

开发者在面对 kubernetes 分布式集群下的日志需求时,常常会感到头疼,既有容器自身特性的原因,也有现有日志采集工具的桎梏,主要包括:

- 容器本身特性:
 - 采集目标多:容器本身的特性导致采集目标多,需要采集容器内日志、容器 stdout。对于容器内部的文件日志采集,现在并没有一个很好的工具能够去动态发现采集。针对每种数据源都有对应的采集软件,但缺乏一站式的工具。
 - 弹性伸缩难:kubernetes 是分布式的集群,服务、环境的弹性伸缩对于日志采集带来了很大的困难,无法像传统虚拟机环境下那样,事先配置好日志的采集路径等信息,采集的动态性以及数据完整性是非常大的挑战。
- 现有日志工具的一些缺陷:

- 一缺乏动态配置的能力。目前的采集工具都需要事先手动配置好日志采集方式和路径等信息,因为它无法能够自动感知到容器的生命周期变化或者动态漂移,所以它无法动态地去配置。
- 一日志采集重复或丢失的问题。因为现在的一些采集工具基本上是通过 tail 的方式来进行日志
 采集的,那么这里就可能存在两个方面的问题:一个是可能导致日志丢失,比如采集工具在
 重启的过程中,而应用依然在写日志,那么就有可能导致这个窗口期的日志丢失;而对于这
 种情况一般保守的做法就是,默认往前多采集 1M 日志或 2M 的日志,那么这就又会可能引起日志采集重复的问题。
- 未明确标记日志源。因为一个应用可能有很多个容器,输出的应用日志也是一样的,那么当我们将所有应用日志收集到统一日志存储后端时,在搜索日志的时候,我们就无法明确这条日志具体是哪一个节点上的哪一个应用容器产生的。

本文档将介绍一种 Docker 日志收集工具 log-pilot,结合 Elasticsearch 和 kibana 等工具,形成一套 适用于 kubernetes 环境下的一站式日志解决方案。

log-pilot 介绍

log-Pilot 是一个智能容器日志采集工具,它不仅能够高效便捷地将容器日志采集输出到多种存储日志后端,同时还能够动态地发现和采集容器内部的日志文件。

针对前面提出的日志采集难题,log-pilot 通过声明式配置实现强大的容器事件管理,可同时获 取容器标准输出和内部文件日志,解决了动态伸缩问题,此外,log-pilot 具有自动发现机制, CheckPoint 及句柄保持的机制,自动日志数据打标,有效应对动态配置、日志重复和丢失以及日志 源标记等问题。

目前 log-pilot 在 Github 完全开源,项目地址是 *https://github.com/AliyunContainerService/log-pilot*。您可以深入了解更多实现原理。

针对容器日志的声明式配置

Log-Pilot 支持容器事件管理,它能够动态地监听容器的事件变化,然后依据容器的标签来进行解析,生成日志采集配置文件,然后交由采集插件来进行日志采集。

在 kubernetes 下, Log-Pilot 可以依据环境变量 aliyun_logs_\$name = \$path 动态地生成日志 采集配置文件,其中包含两个变量:

• \$name 是我们自定义的一个字符串,它在不同的场景下指代不同的含义,在本场景中,将日志 采集到 ElasticSearch 的时候,这个 \$name 表示的是 Index。

- 另一个是 \$path,支持两种输入形式,stdout 和容器内部日志文件的路径,对应日志标准输出和 容器内的日志文件。
 - 第一种约定关键字 stdout 表示的是采集容器的标准输出日志,如本例中我们要采集 tomcat 容器日志,那么我们通过配置标签 aliyun.logs.catalina=stdout 来采集 tomcat 标准 输出日志。
 - 第二种是容器内部日志文件的路径,也支持通配符的方式,通过配置环境变量 aliyun_log s_access=/usr/local/tomcat/logs/*.log来采集 tomcat 容器内部的日志。当然如果 你不想使用 aliyun 这个关键字,Log-Pilot 也提供了环境变量 PILOT_LOG_PREFIX 可以指定 自己的声明式日志配置前缀,比如 PILOT_LOG_PREFIX: "aliyun,custom"。

此外,Log-Pilot 还支持多种日志解析格式,通过 aliyun_logs_\$name.format=<format>标 签就可以告诉 Log-Pilot 在采集日志的时候,同时以什么样的格式来解析日志记录,支持的格式包括:none、json、csv、nginx、apache2 和 regxp。

Log-Pilot 同时支持自定义 tag,我们可以在环境变量里配置 aliyun_logs_\$name.tags="K1= V1,K2=V2",那么在采集日志的时候也会将 K1=V1 和 K2=V2 采集到容器的日志输出中。自定义 tag 可帮助您给日志产生的环境打上 tag,方便进行日志统计、日志路由和日志过滤。

日志采集模式

本文档采用 node 方式进行部署,通过在每台机器上部署一个 log-pilot 实例,收集机器上所有 Docker 应用日志。

该方案跟在每个 Pod 中都部署一个 logging 容器的模式相比,最明显的优势就是占用资源较少,在 集群规模比较大的情况下表现出的优势越明显,这也是社区推荐的一种模式。



前提条件

您已经开通容器服务,并创建了一个 kubernetes 集群。本示例中,创建的 Kubernetes 集群位于华东1地域。

步骤1 部署 elasticsearch

- **1.** 连接到您的 Kubernetes 集群。具体操作参见通过创建*Kubernetes*集群 或 *SSH* 访问 *Kubernetes* 集群。
- 2. 首先部署 elasticsearch 相关的资源对象。

```
vim elasticsearch.yaml
```

3. 然后输入如下所示的编排模板,该编排模板包含一个 elasticsearch-api 的服务、elasticsearch-discovery 的服务和 elasticsearch 的状态集。这些对象都会部署在 kube-system 命名空间下。

```
apiVersion: v1
kind: Service
metadata:
   name: elasticsearch-api
   namespace: kube-system
   labels:
        name: elasticsearch
spec:
        selector:
        app: es
        ports:
        - name: transport
```

```
port: 9200
     protocol: TCP
apiVersion: v1
kind: Service
metadata:
  name: elasticsearch-discovery
   namespace: kube-system
   labels:
    name: elasticsearch
spec:
  selector:
    app: es
  ports:
   - name: transport
     port: 9300
     protocol: TCP
apiVersion: apps/vlbetal
kind: StatefulSet
metadata:
  name: elasticsearch
  namespace: kube-system
   labels:
    kubernetes.io/cluster-service: "true"
spec:
  replicas: 3
                             #高可用,允许宕机一台节点以满足容灾需求
   serviceName: "elasticsearch-service"
   selector:
     matchLabels:
      app: es
   template:
     metadata:
       labels:
        app: es
     spec:
       tolerations:
       - key: node-role.kubernetes.io/master
         effect: NoSchedule
       serviceAccountName: admin
       initContainers:
       - name: init-sysctl
         image: busybox:1.27
         command:
         - sysctl
         - -w
         - vm.max_map_count=262144
         securityContext:
           privileged: true
       containers:
       - name: elasticsearch
         image: registry.cn-hangzhou.aliyuncs.com/cqz/elasticsearch:
5.5.1
         ports:
         - containerPort: 9200
          protocol: TCP
         - containerPort: 9300
          protocol: TCP
         securityContext:
           capabilities:
             add:
               - IPC_LOCK
```

- SYS_RESOURCE
limita:
memory: 4000Mi
requests:
cpu: 100m
memory: 2000Mi
env: #部分参数可根据您的
雪雨进行修改 雪雨进行修改
而女儿11 哆叹 - name: "http host"
value: " $0 \ 0 \ 0$ "
- name: "network.host"
value: " eth0 "
- name: "cluster.name"
value: "docker-cluster"
<pre>- name: "bootstrap.memory_lock"</pre>
value: "false"
- name: "discovery.zen.ping.unicast.hosts"
value: "elasticsearch-discovery"
- name: "discovery.zen.ping.unicast.hosts.resolve_timeout
"
value. "Jus" - name: "discovery zen ping timeout"
value: "6s"
- name: "discovery.zen.minimum master nodes"
value: "2"
- name: "discovery.zen.fd.ping_interval"
value: "2s"
- name: "discovery.zen.no_master_block"
value: "write"
- name: "gateway.expected_nodes"
Value. "2" - name: "gateway expected master nodes"
value: "1"
- name: "transport.tcp.connect timeout"
value: "60s"
- name: "ES_JAVA_OPTS"
value: "-Xms2g -Xmx2g"
livenessProbe:
tcpSocket:
port: transport
InitialDelaySeconds: 20
volumeMounts:
- name: es-data
mountPath: /data
terminationGracePeriodSeconds: 30
volumes:
- name: es-data
hostPath:
path: /es-data

4. 部署成功后, kube-system 命名空间下会出现相关对象,执行以下命令查看运行情况。

<pre>\$ kubectl get svc,StatefulSet</pre>	-n=kube-system		
NAME	TYPE	CLUSTER-IP	
EXTERNAL-IP PORT(S)		AGE	
svc/elasticsearch-api	ClusterIP	172.21.5.134	<none></none>
9200/TCP	22h		
svc/elasticsearch-discovery	ClusterIP	172.21.13.91	<none></none>
9300/TCP	22h		

•••			
NAME	DESIRED	CURRENT	AGE
statefulsets/elasticsearch	2	2	22h

步骤2 部署 log-pilot 和 kibana 服务

1. 部署 log-pilot 日志采集工具,编排模板如下所示。

```
- -
apiVersion: extensions/vlbetal
kind: DaemonSet
metadata:
  name: log-pilot
  namespace: kube-system
   labels:
     k8s-app: log-pilot
     kubernetes.io/cluster-service: "true"
spec:
   template:
     metadata:
       labels:
         k8s-app: log-es
         kubernetes.io/cluster-service: "true"
         version: v1.22
     spec:
       tolerations:
       - key: node-role.kubernetes.io/master
         effect: NoSchedule
       serviceAccountName: admin
       containers:
       - name: log-pilot
         image: registry.cn-hangzhou.aliyuncs.com/acs-sample/log-
pilot:0.9-filebeat
         resources:
           limits:
             memory: 200Mi
           requests:
             cpu: 100m
             memory: 200Mi
         env:
           - name: "FILEBEAT_OUTPUT"
             value: "elasticsearch"
           - name: "ELASTICSEARCH_HOST"
             value: "elasticsearch-api"
           - name: "ELASTICSEARCH_PORT"
             value: "9200"
           - name: "ELASTICSEARCH_USER"
             value: "elastic"
           - name: "ELASTICSEARCH_PASSWORD"
             value: "changeme"
         volumeMounts:
         - name: sock
           mountPath: /var/run/docker.sock
         - name: root
           mountPath: /host
           readOnly: true
         - name: varlib
           mountPath: /var/lib/filebeat
         - name: varlog
           mountPath: /var/log/filebeat
         securityContext:
```

```
capabilities:
      add:
      - SYS ADMIN
terminationGracePeriodSeconds: 30
volumes:
- name: sock
 hostPath:
    path: /var/run/docker.sock
- name: root
 hostPath:
    path: /
- name: varlib
 hostPath:
    path: /var/lib/filebeat
    type: DirectoryOrCreate
- name: varlog
 hostPath:
    path: /var/log/filebeat
    type: DirectoryOrCreate
```

2. 部署 kibana 服务,该编排示例包含一个 service 和一个 deployment。

```
apiVersion: v1
kind: Service
metadata:
                                             #kibana 服务的名称
  name: kibana
  namespace: kube-system
   labels:
     component: kibana
spec:
  selector:
    component: kibana
  ports:
   - name: http
                                              #kibana 服务暴露的端口
    port: 80
    targetPort: http
   type: NodePort
apiVersion: apps/vlbetal
kind: Deployment
metadata:
  name: kibana
  namespace: kube-system
   labels:
     component: kibana
spec:
  replicas: 1
   selector:
     matchLabels:
      component: kibana
   template:
    metadata:
       labels:
         component: kibana
     spec:
       containers:
       - name: kibana
         image: registry.cn-hangzhou.aliyuncs.com/acs-sample/kibana:
5.5.1
         env:
         - name: CLUSTER_NAME
```

```
value: docker-cluster
- name: ELASTICSEARCH_URL
value: http://elasticsearch-api:9200/ #端口
要与 elasticsearch-api 服务暴露的端口一致
resources:
limits:
cpu: 1000m
requests:
cpu: 1000m
ports:
- containerPort: 5601
name: http
```

步骤3 部署测试应用 tomcat

在 elasticsearch + log-pilot + Kibana 这套日志工具部署完毕后,现在开始部署一个日志测试应用 tomcat,来测试日志是否能正常采集、索引和显示。

编排模板如下。

```
apiVersion: v1
kind: Pod
metadata:
 name: tomcat
 namespace: default
 labels:
   name: tomcat
spec:
 containers:
  - image: tomcat
   name: tomcat-test
   volumeMounts:
    - mountPath: /usr/local/tomcat/logs
     name: accesslogs
    env:
     - name: aliyun_logs_catalina
      value: "stdout"
##采集标准输出日志
     - name: aliyun_logs_access
       value: "/usr/local/tomcat/logs/catalina.*.log"
 ## 采集容器内日志文件
 volumes:
    - name: accesslogs
      emptyDir: {}
```

tomcat 镜像属于少数同时使用了 stdout 和文件日志的 Docker 镜像,适合本文档的演示。在上面的编排中,通过在 pod 中定义环境变量的方式,动态地生成日志采集配置文件,环境变量的具体说明如下:

- aliyun_logs_catalina=stdout表示要收集容器的 stdout 日志。
- aliyun_logs_access=/usr/local/tomcat/logs/catalina.*.log 表示要收集容器内 /usr/local/tomcat/logs/目录下所有名字匹配 catalina.*.log 的文件日志。

在本方案的 elasticsearch 场景下,环境变量中的 \$name 表示 Index,本例中 \$name即是 catalina 和 access。

步骤 4 将 kibana 服务暴露到公网

上面部署的 kibana 服务的类型采用 NodePort,默认情况下,不能从公网进行访问,因此本文档配置一个 ingress 实现公网访问 kibana,来测试日志数据是否正常索引和显示。

1. 通过配置 ingress 来实现公网下访问 kibana 。本示例选择简单的路由服务来实现,您可参考 *Ingress* 支持 获取更多方法。该 ingress 的编排模板如下所示。

apiVersion: extensions/vlbetal kind: Ingress metadata: name: kibana-ingress	
namespace: kube-system	#要与 kibana 服务处
于同一个 namespace	
spec:	
rules:	
- http:	
paths:	
- path: /	
backend:	
serviceName: kibana	#输入 kibana 服务的
名称	
servicePort: 80	#输入 kibana 服务暴
露的端口	

2. 创建成功后,执行以下命令,获取该 ingress 的访问地址。

\$	kubectl	get	ingress	-n=kube-system			
1	JAME		HOSTS	ADDRESS	PORTS	AGE	
shared-dns		*	120.55.150.30	80	5m		

3. 在浏览器中访问该地址,如下所示。

	+ → C 🔺 不安全 https://1											
kibana	0 hits	New Save Open Share < 🔿 Last 15 minutes 🗲										
KIDalla	Search (e.g. status:200 AND exter	usion:PHP) Uses lucene query syntax										
Ø Discover	Add a filter 🕇											
Uisualize	* - 3											
🕥 Dashboard	Selected Fields	No results found 🙂										
😨 Timelion	? _source	Unfortunately i could not find any results matching your search. I tried really hard. I looked all over the place and frankly, I just couldn't find anything good. Help me, help you. Here are some										
🔑 Dev Tools	Available Fields	ideas:										
🄅 Management		Expand your time range										
		I see you are looking at an index with a date field. It is possible your query does not match anything in the current time range, or that there is no data at all in the currently selected time range. Click the button below to open the time picker. For future reference you can open the time picker by clicking on the Otime picker button in the top right corner of your screen.										
		Refine your query										

 4. 单击左侧导航栏中的management ,然后单击Index Patterns > Create Index Pattern。具体 的索引名称会在 \$name 变量后缀一个时间字符串,您可以配合通配符 * 进行创建。本例中使用 \$name* 来创建 Index Pattern。

您也可以执行以下命令,进入 elasticsearch 对应的 pod,在 index 下列出 elasticsearch 的所有 索引。

\$ kubectl get pods -n=kube-system 对应的 pod	#找到 elasticsearch
<pre> \$ kubectl exec -it elasticsearch-</pre>	1 bash #进入
elasticsearch 的一个 pod	
•••	
<pre>\$ curl 'localhost:9200/_cat/indice</pre>	s?v' ## 列出所有家
弓	
health status index	uuid pri rep
docs.count docs.deleted store.size	pri.store.size
green open .kibana	x06jj19PS4Cim6Ajo51PWg 1 1
4 0 53.6kb	26.8kb
green open access-2018.03.19	txd3tG-NR6-guqmMEKKzEw 5 1
143 0 823.5kb	411.7kb
green open catalina-2018.03.19	ZgtWd16FQ7qqJNNWXxFPcQ 5 1
143 0 915.5kb	457.5kb

5. 索引创建完毕后,单击左侧导航栏中的Discover,然后选择前面创建的 Index,选择合适的时间 段,在搜索栏输入相关字段,就可以查询相关的日志。



至此,在阿里云 Kubernetes 集群上,我们已经成功测试基于 log-pilot、elasticsearch 和 kibana 的 日志解决方案,通过这个方案,我们能有效应对分布式 kubernetes 集群日志需求,可以帮助提升运 维和运营效率,保障系统持续稳定运行。

1.19 监控管理

1.19.1 通过资源分组进行监控与告警

前提条件

- 如果之前没有创建过集群,您需要创建Kubernetes集群。
- Kubernetes 版本需要在 1.8.4 及以上,若集群版本过低,您可以先对集群进行升级,然后通过 升级监控服务的方式快速建立资源报警分组。

背景信息

在 IT 系统基础设施运维中,监控告警一直是保证可靠性和安全性的基础,有助于日常运维、系统监测、故障排除和调试。

在 Kubernetes 场景下,传统的容器监控方案通过静态配置化的监控 agent 或中心化的 server 进行 资源监控和告警,会遇到很大的难题。例如,由于容器更多的是在资源池中调度,宿主机部署监控 agent 会造成缺乏必要信息来识别监控对象;容器的生命周期与传统应用相比而言会更加短暂,而 由容器抽象的上层概念如 kubernetes 中的 ReplicaSet、Deployment 等则没有太好的办法从采集的 数据中进行反向的抽象,造成单纯的容器监控数据无法有效地进行监控数据的聚合和告警,一旦应 用的发布可能会导致原有的监控与报警规则无法生效。

阿里云容器服务 Kubernetes 与云监控进行了深度集成,用应用分组来抽象逻辑概念,实现逻辑的 概念和物理概念在监控数据、生命周期上面的统一。此外,阿里云云监控服务提供丰富的功能特性 和自定义工具,帮助您快速实现 Kuberbetes 资源监控和告警的最佳实践。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的集群,在集群列表下找到目标集群。
- 查看 kubernetes 版本和集群 ID 等信息,单击集群右侧的更多 > 升级监控服务,快速建立资源 报警分组。

送 说明: 若您的集群版本低于 1.8.4,请先单击集群升级进行升级。

容器服务		集群列表			您最多可以创建	皇 5 个集群,每个集群最多可以添加 20 个节点	新 创建 Kubernetes 集群 👻
Kubernetes ▼ 集群 集群	Swarm	常见问题: の如何创建集群 のず容和熔容集群 名称 マ	♂通过 kubecti 连接 Kubernetes 集群	⌀ 通过命令管理应用			
节点存储		集群谷称/ID k8e-duster 🖍	地域 网络类型 络东1 虚拟专有网络 vpc-bp1gnu8br4a	集群状态 ● 运行中	创建时间 2018-04-23 11:41:46	Kubernetes 版本	管理 查看日志 投 集群伸缩 更多
 ▼ 应用 部署 服务 							 部除 添加已有节点 集群升级 <u>予切違边限例</u>

 登录 云监控服务控制台。在左侧导航栏中单击应用分组,在应用分组中,可以看到包含集群 ID 信息的 kubernetes 资源分组。

云监控	应用分组					€ 刷新	创建组 当前版本: 升级版本,	免费版 获得更多配额和功能	升级
概览	業型 ▼ 请输入要搜索的组名(支持模糊搜索)	搜索	Ę						
	分组名称 / 分组ID	健康状况 🛛	类型	服务器总数 🖉	资源类型总数 🛛	不健康实例数②	创建时间		操作
主机监控	test / 27016	٢	自建	2	2	0	2017-10-22 03:28:43	管理	更多,
事件监控	k8s-ce7t L-master / 53473	•	kubernetes	3	2	0	2018-04-03 09:56:01	管理	更多,
自定义监控	k8s-ce7tworker / 53474	•	kubernetes	1	1	0	2018-04-03 09:56:01	管理	更多,
日志监控							共 3条	10 V « < 1	> >>
▶ 站点监控专业版									

单击分组名称,进入具体的分组页面,您可以查看组内的各项监控视图。以 kubernetes 的 master 分组为例,默认显示分组的拓扑视图。

Kubernetes 节点从职能上分为 Worker 和 Master 两种不同的节点。Master 节点上面通常会部署 管控类型的应用,整体的资源要求以强鲁棒性为主;而 Worker 节点更多的承担实际的 Pod 调 度,整体的资源以调度能力为主。当你创建资源报警分组时,容器服务会为你自动创建两个资源 分组,一个是 Master 组,一个是 Worker 组。Master 组中包含了 Master 节点以及与其相关的 负载均衡器;Worker 组包含了所有的工作节点。

<	k8s-ce7b578f229754f4783ce693839bdf031-master ± 返回应用分组	1小时	2小时 6小时	寸 12小时	1天 3	沃 7天 14天	2018-04-03	3 05:42:25	- 2018-04-03 🗰
组内资源					-				
■ 监控视图					℃刷新	新建报警规则	修改组	删除组	应用模板到组
故障列表	A =								
事件监控									
可用性监控		.							
日志监控				负载均衡					
自定义监控									
报警历史)						
报警规则			y .						
	L		7 .	B务器ECS					
	未留好望						毎行展示 3 イ	1	添加监控图表

 您若想了解分组内详细的机器信息,请单击列表视图的图标,您可以查看分组内的各云产品的详 细信息。

<	k8s-c	-master 主返	回应用分组	1小时 2小时 6小时	12小时 1天	3天 7天 14天	2018-04-03 05:42	2:25 - 2018-04-03 🗰		
组内资源										
• 监控视图	_						℃ 刷新	新建报警规则	修改组 删除结	8 应用模板到组
故障列表										
事件监控	云服务器ECS (3)	进入:	云服务器ECS 控制台	负载均衡 (1)		进入 负载均衡 控制台				
可用性监控	实例名称	健康状况 🛛	资源描述	实例名称	健康状况 🙆	资源描述				
日志监控	10 C			lb-	0.T#W+					
自定义监控	Table Contraction of Contract	● 正常状态	.10							
报警历史					共 1条, 毎页显示10条 (1 →					
报警规则	232 areas	♥正常状态	.11							
	222 communities	●正常状态	.12							
		共 3条, 每页显示1	.0条 < 1 >							
					查看全部					
	监控图表							[每行展示 3 个图 🔹	添加监控图表

7. 单击右下角的添加监控图表,您可以添加更多的监控图表。

添加图表	×
1 选择图表类型	
折线图 面积图 热力图 饼图	
2 选择监控顶	
云产品监控 日志监控 自定义监控	
云服务器ECS ▼ 云服务器ECS Y轴显示范围: 0 au	to
无数据	
监控项: 插件无心跳 v status v	
工物加益性权	
发布	

8. 在左侧导航栏中单击监控视图,您可以查看分组内各云产品的详细监控指标。



- 在左侧导航栏中单击报警规则,该页面列出当前分组中已有的报警规则列表,默认会在 Master 分组中设置所有节点的核心组件的健康检查。
 - 1. 您可以单击新建报警规则,根据业务需求,来创建属于该分组的报警规则。

<	k8s-c		er 🔹 i 🦉	回应用分组	1小时 2小时 6小	时 12小时 1天 3天 1	天 14天 2018-04-	33 07:41:52 - 2018-04-03 🔳
组内资源 • 监控视图	请输入报警规则名称			搜索		○刷新新建	报警规则修改组	删除组 应用模板到组
故障列表	规则名称	状态 (全部) -	启用	維度	报警规则	产品名称 (全部) 👻	2 Jak	操作
事件监控 可用性监控	 kube-proxy- TelnetLatency.Average 	。 ●正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 平均值>1000 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	查看 修改 │ 禁用
日志监控 自定义监控 招感历史	kube-proxy- TelnetStatus.Value	●正常状态	已启用	分组维度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 监控值>400 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	章看 修改│ 禁用
报警规则 1	kubelet- TelnetLatency.Average	e ●正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 平均值>1000 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	查看 修改 禁用
	kubelet- TelnetStatus.Value	●正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 监控值>400 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	查看 修改│ 禁用
=	□ 启用 禁用	删除					共 4条 2	0 ▼ « < 1 > ≫

- 2. 进入报警规则页面,您需要设置报警规则。
 - 您可选择报警关联对象,如云服务器 ECS。
 - 您可选择是否使用模板创建报警规则。若选择使用模板创建报警规则,您可以在选择模板下拉框中选择已有的报警模板;或者可以单击创建报警模板,创建新的自定义报警模板,参见,然后再进行选择。
 - 您可设置通知方式,如通过钉钉、邮件、短信的方式在第一时间获取到 Kubernetes 的集群状态。

创建报誓	登规则 t 返回										
	大联资源										
	产品:	元服务器ECS ▼									
	资源范围:	应用分组 ● 远择应用分组时,支持使用报警模板。点击 查看报警模板最佳实践									
	分组:	k8s- *									
2	设置报警规则										
	使用模板:	 ● 是 ○ 否 									
	选择模板:	est_monitor ¥ 创建报警模板									
		test_monitor_agent_heal 插件无心貌									
		test_monitor_CPUUtilizat CPU使用率 * Im * 连续3次* >= * 90 % *									
	通道沉默时 间:	24/jsj • Ø									
	生效时间:	00:00 · Z23:59 ·									
3	通知方式										
	通知对象:	日选组 1 个									
		云账号报警联系人									
	报警级别:	Warning ("https://withewith									
		◎ Info (邮箱+旺旺+\$]钉机器人)									

3. 最后单击确认,本例中新建的报警规则会出现在规则列表下。

<	k8s-cedcc8dcbe5b3466d802f86a	4f96cc97e-master	1小时 2小时 6小时 12小时	1天	3天 7天 14天	2018-04-13	09:45:18 -	2018-04	-13 🗰		
组内资源●监控视图	请输入报警规则名称			搜索		こ刷新	新建报警规则	修改组	删除组	应用模	板到组
故障列表	规则名称	状态 (全部) -	启用	維度 (全部) *	报誓规则		产品名称 (全部) 👻	通知对象			操作
事件监控 可用性监控	test_monitor_CPUUtilization	♥正常状态	已启用	分组維度:k8s- cedcc8dcbe5b3466d802f86a4f96cc97e- master	1分钟 CPU使用率 平均值>=90 % 连续 3 次 则报警		云服务器ECS	云账号报警题 系人 查看	关修改	(禁用	查看 删除
日志监控 自定义监控 招幣历史	■ 常用基础模板 _agent_heartbeat	●正常状态	已启用	分组维度:k8s- cedcc8dcbe5b3466d802f86a4f96cc97e- master	插件无心跳事件报警		云服务器ECS	云账号报警题 系人 查看	关 修改	(禁用	查看 删除
报警规则	□ 常用基础模板 _diskusage_utilization	♥正常状态	已启用	分组维度:k8s- cedcc8dcbe5b3466d802f86a4f96cc97e- master	1分钟 磁曲使用率 平均值>90 % 连续 3 次 则报警		云服务器ECS	云账号报警题 系人 查看	关 修改	(禁用	查看 删除

后续操作

在左侧导航栏中,您可以探索更多符合自己资源监控需求的功能特性,如故障列表、事件监控、可 用性监控、日志监控等。

1.19.2 使用 Grafana 展示监控数据

前提条件

- 您已经成功部署一个 Kubernetes 集群,参见创建Kubernetes集群。
- 本示例使用的 Grafana 的镜像地址是 registry.cn-hangzhou.aliyuncs.com/acs/ grafana:5.0.4,内置了相关监控模板。

背景信息

在 kubernetes 的监控方案中, Heapster+Influxdb+Grafana 的组合相比 prometheus 等开源方案而 言更为简单直接。而且 Heapster 在 kubernetes 中承担的责任远不止监控数据的采集,控制台的监 控接口、HPA的 POD 弹性伸缩等都依赖于 Heapster 的功能。因此 Heapster 成为 kubernetes 中一 个必不可少的组件,在阿里云的 Kubernetes 集群中已经内置了 Heapster+Influxdb 的组合,如果需 要将监控的数据进行展示,只需要配置一个可用的 Grafana 与相应的 Dashboard 即可。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Kubernetes 菜单下,单击左侧导航栏中的应用 > 部署,进入部署列表页面。
- 3. 单击页面右上角的使用模板创建。

容器服务 - Kubernetes ▼	部審列表	刷新 使用镜像创建 使用模板创建
概览	集群 k8s-duster v 命名空间 default v	2
▼ 集群	名称 标签 容器组数量 创建时间	操作
集群		
节点	(1) 没有查询到符合条件的记录	
存储卷		
命名空间		
▼ 应用 【1		
部署		

4. 对模板进行相关配置,部署 Grafana 的 deployment 和 service,完成配置后,单击创建。

- 集群:选择所需的集群。
- 命名空间:选择资源对象所属的命名空间,必须是kube-system。
- 示例模板:本示例选择自定义模板,其中包含一个 deployment 和 service。



本示例的编排模板如下。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
  name: monitoring-grafana
  namespace: kube-system
spec:
  replicas: 1
   template:
    metadata:
       labels:
         task: monitoring
        k8s-app: grafana
     spec:
       containers:
       - name: grafana
        image: registry.cn-hangzhou.aliyuncs.com/acs/grafana:5.0.4
        ports:
         - containerPort: 3000
          protocol: TCP
         volumeMounts:
         - mountPath: /var
          name: grafana-storage
         env:
         - name: INFLUXDB_HOST
           value: monitoring-influxdb
       volumes:
       - name: grafana-storage
        emptyDir: {}
apiVersion: v1
kind: Service
metadata:
  name: monitoring-grafana
```

```
namespace: kube-system
spec:
  ports:
  - port: 80
    targetPort: 3000
type: LoadBalancer
  selector:
    k8s-app: grafana
```

5. 完成部署后,返回部署页面,选择所需集群,然后选择 kube-system,查看其下部署的应用。

容器服务 - Kubernetes ▼	部署列表				刷新使用镜像创建使用模板创建
概览	集群 k8s-cluster v 命名空间	kube-system 🔻 🙎			
▼ 集群	名称	标签	容器组数量	创建时间	部
集群	alicloud-disk-controller	app:alicloud-disk-controller	1/1	2018-06-27 17:53:56	详情 编辑 监控 更多▼
节点 存储券	alicloud-monitor-controller	k8s-app:alicloud-monitor-controller task:monitoring	1/1	2018-06-28 14:06:30	洋情 编辑 监控 更多▼
命名空间	default-http-backend	app:default-http-backend	1/1	2018-06-27 17:53:56	详情 │ 编辑 │监控│更多▼
▼应用	heapster	k8s-app:heapster task:monitoring	1/1	2018-06-27 17:53:56	详情 编辑 监控 更多▼
部署	kube-dns	k8s-app:kube-dns	1/1	2018-06-27 17:53:51	详情 编辑 监控 更多▼
容器组 服务	monitoring-grafana	k8s-app:grafana task:monitoring	1/1	2018-07-10 16:04:34	详情 编辑 监控 更多▼
路由	monitoring-influxdb	k8s-app:influxdb task:monitoring	1/1	2018-06-27 17:53:56	详情 编辑 监控 更多★

6. 单击 monitoring-grafana 的名称,查看部署状态,等待运行状态变为 running。

创建monitoring-grafana t 返回列表 制新				
基本信息				
名称:	monitoring-grafana			
命名空间:	kube-system			
创建时间:	2018-04-25T07:32:02Z			
标签:	k8s-app:grafana task:monitoring			
注解:	deployment.kubernetes.io/revision:1			
选择器:	k8s-app:grafana task:monitoring			
策略:	RollingUpdate			
状态:	已更新:1个,不可用:0个,共计:1个			
容器组 关联服务				
名称		状态	镜像	事件
monitoring-grafana-675dc8448c-nq	4s	Running	registry.cn-hangzhou.aliyuncs.com/acs/grafana:5.0.4	

 P. 单击左侧导航栏中的应用 > 服务,进入服务列表,选择所需的集群和命名空间(kubesystem),查看外部端点。

这个地址是通过 LoadBalancer 类型的 service 自动创建的,对于要求更安全访问策略的开发者 而言,建议考虑添加 IP 白名单或者使用配置证书等方式增强安全性。

-							
容器服务 - Kubernetes ▼	服务列表						刷新 创建
概览	集群 k8s-cluster	▼ 命名空间	kube-system 🔻 3				
▼ 集群	名称	类型	创建时间	集群IP	内部端点	外部端点	操作
集群	default-http-backend	ClusterIP	2018-06-27 17:53:56	1000	default-http-backend:80 TCP	-	详情 更新 查看YAML 删除
売	heapster	ClusterIP	2018-06-27 17:53:56	10.00	heapster:80 TCP	-	详情 更新 查看YAML 删除
存储卷 命名空间	kube-dns	ClusterIP	2018-06-27 17:53:51		kube-dns:53 UDP kube-dns:53 TCP	-	详情 更新 查看YAML 删除
▼ 应用 1	monitoring-grafana	LoadBalancer	2018-07-10 16:04:34	-	monitoring-grafana:80 TCP monitoring-grafana:32746 TCP	80	详情 更新 查看YAML 删除
部署	monitoring-influxdb	ClusterIP	2018-06-27 17:53:56	10.000	monitoring-influxdb:8086 TCP		详情 更新 查看YAML 删除
容器组 2 正	nginx-ingress-lb	LoadBalancer	2018-06-27 17:53:56	-	nginx-ingress-lb:80 TCP nginx-ingress-lb:30883 TCP nginx-ingress-lb:443 TCP nginx-ingress-lb:22380 TCP	80 443	详情 更新 查看YAML 删除

8. 选择 monitoring-grafana 服务,单击右侧的外部端点,登录 Grafana 监控界面。

默认的 Grafana 的用户名和密码都是 admin,建议在登录后先修改为更复杂的密码。



9. 您可选择内置的监控模板 , 查看 Pod 和 Node 的监控 Dahsboard。

本示例使用的 Grafana 版本内置了两个模板,一个负责展示节点级别的物理资源,一个负责展示 Pod 相关的资源。开发者也可以通过添加自定义的 Dashboard 的方式进行更复杂的展现,也可以基于 Grafana 进行资源的告警等。

Ô	Kube	rnetes Node <u>∦</u>	註控・	
	node_name	cn-hangzhou.i-t		•
+	🗸 Dashboa	ard Row		
Ļ		Uptime		CPU Cores
*	1.	84 day		2
	✓ History			
	108			Filesystem Available
	0.5.8			
	0 B			No data points
	-0.5 B			
	-1.0 B			
		15:00	15:10	15:20
				Memory Utilization
	92.500%	\bigwedge		





91.500%

91.000%

📲 Kubernetes Pod监控 🗸			ult 🖄 🖞	☆ 🖹 🌣 < Ə >	② Last 1 hour Refresh every 5s 2
namespace default	test-wordpress-6945bc9556-mpdxn ▼				
✓ Status					
Uptime	CPU Usage Rate	Memory Usage	Network Tx Rate	Network Rx Rate	
2.07 day		181 MB	1.815 kBps	7.61 kBps	
 Resource Limit and Request 	¢ ±				
CPU Limit	CPU Request	Memory Limit	Memory Request		
0	300	0 B	537 MB		
↓ History					
45 40 35 30 25 1500 1500	CPU Usage Rate		182.5 MB 182.0 MB 181.5 MB 181.0 MB	Memory Usage	
— cpu/usage_rate.mean	15.20 15.30	U.40 10.50	— memory/usage.mean	15.20 15.30	13.40 15.50

1.19.3 与云监控集成与使用

前提条件

请先检查kube-system命名空间下是否已经部署了alicloud-monitor-controller,若未部署,请进行旧版本集群升级。

使用方式

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的部署,进入部署列表页面,单击监控,或者在内置的Kubernetes Dashboard的部署页面中单击监控。

容器服务 - Kubernetes ▼	部署列表	部署列表			
概览	集群 K8S-cs v 命名空间	kube-system 🔻 🙎			
▼ 集群	名称	标签	容器组数量	创建时间	3 操作
集群	alicloud-disk-controller	app:alicloud-disk-controller	1/1	2018-06-27 17:53:56	洋情 编辑 监控 更多▼
节点	alicloud-monitor-controller	kBs-app:alicloud-monitor-controller task:monitoring	1/1	2018-06-28 14:06:30	洋情 编辑 监控 更多▼
命名空间	default-http-backend	app:default-http-backend	1/1	2018-06-27 17:53:56	洋情 编辑 监控 更多▼
▼ 应用 1	heapster	kBs-app:heapster task:monitoring	1/1	2018-06-27 17:53:56	洋情 编辑 监控 更多▼
部署	kube-dns	k8s-app:kube-dns	1/1	2018-06-27 17:53:51	洋情 │ 编辑 │监控│更多▼

<	➡ 工作负载 部署					十 创建
集群	部署					Ŧ
命名空间	名称 🗢	标签	容器组	已创建 ≑	镜像	
持久化存储卷	monitoring-grafana	k8s-app: grafana task: monitoring	1 / 1	2018-07-10 16:04:34	registry.cn-hangzhou.aliyuncs.com/acs,	:
角色存储类	alicloud-monitor-controller	k8s-app: alicloud-monitor-controller task: monitoring	1/1	2018-06-28 14:06:30	registry.cn-hangzhou.aliyuncs.com/acs,	:
<u>会々内間</u>	V tiller-deploy	app: helm name: tiller	1/1	2018-06-27 17:53:58	registry-vpc.cn-hangzhou.aliyuncs.com	3 :
kube-system	alicloud-disk-controller	app: alicloud-disk-controller	1/1	2018-06-27 17:53:56	registry-vpc.cn-hangzhou.aliyur 监控	Ĩ
概况	leapster	k8s-app: heapster task: monitoring	1/1	2018-06-27 17:53:56	registry.cn-hangzhou.aliyuncs.c 伸缩	
工作负载	monitoring-Influxdb	k8s-app: influxdb task: monitoring	1/1	2018-06-27 17:53:56	egistry-vpc.cn-hangzhou.aliyur 删除	
之切过另	oginx-ingress-controller	app: ingress-nginx	1/1	2018-06-27 17:53:56	查看/ registry-vpc.cn-hangzhou.aliyur	编辑 YAML
部署 2	default-http-backend	app: default-http-backend	1/1	2018-06-27 17:53:56	registry-vpc.cn-hangzhou.aliyuncs.com	:

此时会跳转到云监控的相应的应用分组详情页面。

<	k8-c				
组内资源					
● 监控视图	2019 Jun Unit 12143 1A JA /A 14A = Hallingtin · · · · · · · · · · · · · · · · · · ·				

3. 应用分组支持分组和实例两个维度的监控。





4. 如需告警设置,分组级别的指标以group开头,实例级别的指标以pod开头。

关联资源		
产品:	容器服务Kubernetes版	
资源范围:	应用分组	● 选择应用分组时,支持使用报警模板。 点击 查書报警模板最佳实践
分组:	k8s-c07ea7a5639924917af756e7d3f95e	
设置报警规则		
使用模板:	◎ 是 ● 否	1.00
规则名称:	Deployment CPU利用率监控	
规则描述:	group.cpu.usage_rate 5分錄	● 急计 ▼ >= ▼ 調直
	I	0.00
十添加报警规则	group.cpu.limit	-0.50
通道沉默时 间:	group.cpu.request	-1.00
连续几次超过 阈值后报警:	group.cpu.usage_rate	14:10:00 14:15:00 14:23:20 14:31:40 14:40:00 14:50:0(● group.cpu.usage_rate—Sum—应用分组—152343
生效时间:	group.disk.io_read_bytes	

旧版本集群升级

- 1. 登录 容器服务管理控制台。
- 在Kubernetes菜单下,单击左侧导航栏中的应用>部署,进入部署列表页面,单击右上角的使用模板创建。

容器服务 - Kubernetes ▼	部署列表				刷新(用镜像创建	使用模板创建
概览	集群 k8s-test v 命名	G空间 default ▼					2
▼ 集群	名称	标签	容器组数量	创建时间			操作
集群							
节点			① 没有查询到符合条件的记录				
存储							
命名空间							
▼ 应用 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1							

3. 选择所需的集群,Kube-system命名空间,使用以下的示例模板,然后单击创建。

Ê	说明:		
根据自	己的集群替换REGION与CLUSTER_	_ID,	并重新部署Heapster的yaml编排。

集群	k8s-test
命名空间	kube-system 🔹
示例模版	Ê定义 ▼
欄版	<pre>implyersion: extensions/vibetal tind: Deployment implyersion: extensions/vibetal tind: Deployment implyersion: Deployment</pre>
	的識

heapster示例编排模板如下。若集群中已有旧版本的heapster,您也可登录到Kubernetes集

群,执行kubectl apply -f xxx.yaml进行更新。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
  name: heapster
 namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: heapster
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      serviceAccount: admin
      containers:
      - name: heapster
        image: registry.##REGION##.aliyuncs.com/acs/heapster-amd64:
v1.5.1.1
        imagePullPolicy: IfNotPresent
        command:
        - /heapster
         --source=kubernetes:https://kubernetes.default
        - --historical-source=influxdb:http://monitoring-influxdb:
8086
        - --sink=influxdb:http://monitoring-influxdb:8086
```

- --sink=socket:tcp://monitor.csk.##REGION##.aliyuncs.com: 8093?clusterId=##CLUSTER_ID##&public=true

4. 更新完毕后,进入Kubernetes 控制台,在kube-system命名空间中,可看到这两

个Deployment处于运行中,即升级完毕。

#8#	kube-flannel-ds	app: flannel tier: node	4 / 4	2018-05-28 17:08:49	registry-vpc.on-hangzhou.aliyuncs.com/acs/fiannel:v0.8.0 registry-vpc.on-hangzhou.aliyuncs.com/acs/fiannel:v0.8.0	
命名空间						
节点	部響					7
持久化存储卷	名称 \$	标签	容器组	已创建 🗢	領像	
角色	alicloud-monitor-controller	k8s-app: alicloud-monitor-controller task: monitoring	1/1	2018-06-26 17:10:32	registry.cn-hangzhou.aliyuncs.com/acs/alicloud-monitor-contr-	
存储类	alibaba-log-controller	kubernetes.lo/cluster-service: true	1/1	2018-06-22 17:27:08	${\it registry.cn-hangzhou.aliyuncs.com/log-service/alibabacloud-lc}$:
命名空间	Stiller-deploy	app: helm name: tiller	1/1	2018-05-28 17:08:52	registry-vpc.cn-hangzhou.aliyuncs.com/acs/tiller:v2.8.2	÷
kube-system *	alicloud-disk-controller	app: alicloud-disk-controller	1/1	2018-05-28 17:08:49	registry-vpc.cn-hangzhou.aliyuncs.com/acs/alicloud-disk-contr	8
概况	default-http-backend	app: default-http-backend	1/1	2018-05-28 17:08:49	${\it registry-vpc.cn-hangzhou.allyuncs.com/acs/defaultbackend:} 1.$:
工作负载	S heapster	k8s-app: heapster task: monitoring	1/1	2018-05-28 17:08:49	registry.cn-hangzhou.allyuncs.com/acs/heapster-amd64:v1.5.	:

alicloud-monitor-controller 的示例编排如下, 部署alicloud-monitor-controller。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: alicloud-monitor-controller
 namespace: kube-system
spec:
 replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: alicloud-monitor-controller
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      hostNetwork: true
      tolerations:
      - effect: NoSchedule
        operator: Exists
        key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      serviceAccount: admin
      containers:
      - name: alicloud-monitor-controller
        image: registry.##REGION##.aliyuncs.com/acs/alicloud-monitor-
controller:v1.0.0
        imagePullPolicy: IfNotPresent
        command:
        - /alicloud-monitor-controller
        - agent
        - --regionId=##REGION##
        - --clusterId=##CLUSTER_ID##
        - --loqtostderr
        - -v = 4
```

对于不清楚自己REGION信息的开发者,可以通过如下的方式快速查询,打开ECS控制台,选择自 己集群所在的地域,页面地址URL中最后一段即是REGION。

G	≩全 https://ecs.console.aliyun.com/#/server/region/cn-hangzhou											
管理控	制台 产品与	∋服务 ▼	₩ 华东1(杭州) ▼	搜索		Q	4 24	费用	工单			
云服务	器 ECS	144	实例列表 2									
概览 实例	1	-	 选择实例属性项搜索,或者输入关键字识别搜索 		Q	标签						

1.20 安全管理

授权管理

与 Swarm 集群一样, Kubernetes 集群支持集群级别的操作的子账号授权。

详细操作参见使用子账号。

全链路 TLS 证书

在容器服务提供的 Kubernetes 集群中,存在的以下通信链路,均会进行 TLS 证书校验,以保证通 信不被窃听或篡改。

- 位于 Worker 节点上的 kubelet 主动连接位于 Master 节点上的 apiserver 时
- 位于 Master 节点上的 apiserver 主动连接位于 Worker 节点上的 kubelet 时

在初始化过程中,发起初始化的 Master 节点会通过 SSH 隧道的方式连接到其他节点的 SSH 服务(22 端口)进行初始化。

原生 Secret&RBAC 支持

Kubernetes Secret 用于存储密码、OAuth Token、SSH Key 等敏感信息。明文地将这些敏感信息 写在 Pod YAML 文件中或者写在 Dockerfile 固化到镜像中,会有信息泄露的可能,使用 Secret 能 有效地避免这些安全隐患。

详细信息参见Secret。

Role-Based Access Control (RBAC) 使用 Kubernetes 内置的 API 组来驱动授权鉴权管理,您可以通过 API 来管理不同的 Pod 对应到不同的角色,以及各自的角色拥有的访问权限。

详细信息参见Using RBAC Authorization。

网络隔离

Kubernetes 集群中不同节点间 Pod 默认是可以互相访问的。在部分场景中,不同业务之间不应该 网络互通,为了减少风险,您需要引入网络隔离(Network Policy)。在 Kubernetes 集群中,您可 以使用 Canal 网络驱动实现网络隔离支持。

镜像安全扫描

Kubernetes 集群可使用容器镜像服务进行镜像管理,镜像服务支持镜像安全扫描。

镜像安全扫描可以快速识别镜像中存在的安全风险,减少 Kubernetes 集群上运行的应用被攻击的可能性。

详细描述参见 镜像安全扫描。

安全组与公网访问

每个新建的集群会被默认分配一个新的、安全风险最小化的安全组。该安全组对于公网入方向仅允许 ICMP。

创建集群默认不允许公网 SSH 连入,您可以参考SSH 访问 Kubernetes 集群 配置通过公网 SSH 连入到集群节点中。

集群节点通过 NAT 网关访问公网,可进一步减少安全风险。

1.21 常见问题

1.21.1 收集 Kubernetes 诊断信息

1. 在 master 节点下载诊断脚本,并增加运行权限。

```
curl -o /usr/local/bin/diagnose_k8s.sh http://aliacs-k8s-cn-hangzhou
.oss-cn-hangzhou.aliyuncs.com/public/diagnose/diagnose_k8s.sh
chmod u+x /usr/local/bin/diagnose_k8s.sh
```

2. 执行诊断脚本。

```
diagnose_k8s.sh
.....
+ echo 'please get diagnose_1514939155.tar.gz for diagnostics'
##每次执行诊断脚本,产生的日志文件的名称不同
please get diagnose_1514939155.tar.gz for diagnostics
+ echo '请上传 diagnose_1514939155.tar.gz'
```
请上传 diagnose_1514939155.tar.gz

3. 上传产生的日志。

```
cd /usr/local/bin
ls -ltr|grep diagnose_1514939155.tar.gz
志文件名
```

##注意替换为生成的日

1.21.2 集群创建失败

查看失败原因

您可以通过查看集群的创建事件来查看集群创建失败的原因。

登录 ROS 管理控制台

选择集群所在的地域,选择所需的集群并单击右侧的管理,单击左侧导航栏中的事件,将鼠标移动 到失败事件上查看具体的失败报错信息。

概范	直看生成的资源					
资源	资源名称	关联资源ID	资源关型	资源状态	状态描述	事件发生时间
事件	k8s_dev	dcb38ca9-fb8b-4471-b415-604cc2	ALTYUN::ROS::Stack	国滚完成	Stack ROLLBACK compl	2017-05-21 10:47:57
欄板	k8s_vpc	+	ALTYUN::ECS::VPC	删除完成	state changed	2017-05-21 10:47:57
	k8s_vpc		ALTYUN::ECS::VPC	删除中	state changed	2017-05-21 10:47:56
	k8s_master_cloudinit	k8s_master_cloudinit_wait_cond	ALTYUN::ROS::WaitConditionHandle	删除完成	state changed	2017-05-21 10:47:55
	k8s_master_cloudinit	k8s_master_cloudinit_wait_cond	ALTYUN::ROS::WaltConditionHandle	删除中	state changed	2017-05-21 10:47:54
	k8s_node_cloudinit_w	k8s_node_cloudinit_wait_cond_h	ALTYUN::ROS::WaitConditionHandle	影除完成	state changed Resource CREATE failed: Resoo	2017-05-21 10:47:54
Ξ	k8s_node_cloudinit_w	k8s_node_cloudinit_wait_cond_h	ALTYUN::ROS::WaitConditionHandle	激除中	nseException: resources.k8s_vp c: VPC quota exceeded. Code:	2017-05-21 10:47:53
	k8s_dev	dcb38ca9-fb8b-4471-b415-604cc2	ALTYUN::ROS::Stack	回線中	QuotaExceeded.Vpc RequestId: 5947330C-4888-4E63-9789-0E DB3F4A7294	2017-05-21 10:47:53
	k8s_dev	dcb38ca9-fb8b-4471-b415-604cc2	ALTYUN::ROS::Stack	创建失败	Resource CREATE fail	2017-05-21 10:47:52
	k8s_node_cloudinit_w	k8s_node_cloudinit_wait_cond_h	ALTYUN::ROS::WaitConditionHandle	创建完成	state changed	2017-05-21 10:47:52
					共有16条。	, 每页显示:10条 《 < 1 2 > »

上图中的报错信息显示由于 VPC 达到配额导致集群创建失败。

失败异常码及解决方法

Code: QuotaExceeded.Eip , Message: Elastic IP address quota exceeded

解决方法:释放多余的 EIP 或者提交 VPC 工单提高 EIP 限额。

 The maximum number of SLB instances is exceeded. Code: ORDER.QUANTITY_I NVALID

解决方法:释放多余的 SLB 实例或者提交 SLB 工单提高 SLB 限额。

Resource CREATE failed: ResponseException: resources.k8s_vpc: VPC quota exceeded
 Code: QuotaExceeded.Vpc

解决方法:释放多余的 VPC 或者提交 VPC 工单提高 VPC 限额。

• Resource CREATE failed: ResponseException: resources.k8s_master_1: The specified image does not support cloud-init. Code: ImageNotSupportCloudInit

解决方法:使用自定义镜像创建集群,自定义镜像必须是基于最新的 centos 公有云镜像。

 Status Code: 403 Code: InvalidResourceType.NotSupported Message: This resource type is not supported;

解决方法: ECS 没有库存或者类型不支持。

• Your account does not have enough balance. Code: PAY.INSUFFICIENT_BALANCE 解决方法:账户余额不足,请充值。

1.21.3 删除 Kubernetes 集群失败: ROS stack 无法删除

问题原因

用户在 ROS 创建的资源下手动添加了一些资源(比如在 ROS 创建的 VPC 下手动添加了一个 VSwitch), ROS 是没有权限删除这些资源的。这就会导致 ROS 删除 Kubernetes 资源时无法处理 该 VPC,最终导致删除失败。

- 说明:

有关创建 Kubernetes 集群时 ROS 自动创建的资源,参见创建Kubernetes集群。

解决办法

1. 集群删除失败时(集群的状态显示删除失败),跳转到 ROS 管理控制台。

集群列表					您最多可以创建 5 个集群,	每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes 集群
常见问题: & 如何创建集群 & 扩容和缩容集群	Ø 通过 kubectl	主接 Kubern	etes 集群 🔗 通过命令管理应用					
名称 ▼								
集群名称/ID	集群类型	地域	网络类型	集群状态	创建时间	Kubernetes 版本		操作
k8stest c77862d75bda644679b48d15f1a42e831	Kubernetes	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●删除失败	2018-01-04 19:47:08	1.8.4	管理	1 查看日志 删除 控制台 更多▼

2. 集群删除失败时(集群的状态显示删除失败),跳转到 ROS 管理控制台。

集群列表					您最多可以创建 5 个集群	, 每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes 集群
常见问题: ⑧ 如何创建集群 ⑧ 扩容和缩容集群	Ø 通过 kubectl i	车接 Kuberr	netes 集群 🛛 ⑧ 通过命令管理应用					
名称 ▼								
集群名称/ID	集群类型	地域	网络类型	集群状态	创建时间	Kubernetes 版本		操作
k8stest c77862d75bda644679b48d15f1a42e831	Kubernetes	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●删除失败	2018-01-04 19:47:08	1.8.4	管理	2 查看日志 删除 控制台 更多▼

3. 选择集群所在的地域,找到集群对应的资源栈 k8s-for-cs-{cluster-id},可以看到其状态为删除失败。

资源栈列表	华北1 华北2 华北 美国东部1(弗吉尼亚)	3 华北 5 华东 1 美国西部 1 (硅谷)	华东 2 华南 1 中东东部 1 (迪拜)	香港 亚太东 欧洲中部1()	R北 1 (东京) 法兰売福)	亚太东南 1 (新加坡)	亚太东南 2 (悉尼)	亚太东南3 (吉隆坡)		新建资源栈 🔹	2刷新
<u> </u> 次迎加入ROS	5交流群进行讨论和反馈,¥ ▼ 请输入资源栈名进行	街群号:11783495。 音词	旺旺群号:149600 搜索	6086.							
名称	· ·	₩~ (A	所有) - 超	时 (分钟)	失败回滚	状态描述			创建时间	ex.18	操作
k8s-for-cs-c77	7862d75bda644679b				否	Resource DELETE f	ailed: ResponseExcep	ntio	2018-01-04 19:47:10	管理	
test		● 105×-	·····································		是	Resource DELETE f	ailed: ResponseExcep	tio	2017-05-22 14:25:14	管理	更多▼ 删除 更多▼
									共有3条 ,每页显示:1	0条 « < 1	> >>

4. 单击资源栈的名称进入资源栈详情页面,单击左侧导航栏中的资源。

您可以看到哪些资源删除失败了。本示例中负载均衡下的 VSwitch 删除失败。

<	资源列表							€刷新
概览	注意:当删除资源栈时,包年	包月类型的资源将会被忽略!						
资源	资源名称	资源ID	资源类型	状态	状态描述	创建日期	更新日期	操作
事件	k8s_api_server_liste	lb-bp1kbi9oewtiy30azwlr6	ALIYUN::SLB::Listener	🔵 删除完成	state changed	2017-10-26 14:33:55	2018-01-03 11:14:14	详情
模板	k8s_dashboard_listen	lb-bp1kbi9oewtiy30azwlr6	ALIYUN::SLB::Listener	● 删除完成	state changed	2017-10-26 14:33:55	2018-01-03 11:14:14	详情
	k8s_vswitch	vsw-bp19v3gdubfcbt7b1sniy	ALIYUN::ECS::VSwitch	● 删除失败	ResponseException: r	2017-10-26 14:33:55	2018-01-15 16:35:21	详情

5. 进入删除失败的资源所在产品的控制台,并找到该资源。

本示例中,登录 VPC 管理控制台,找到集群所在的 VPC,并在该 VPC 下找到删除失败的 VSwitch。

交换机列表							一个专有网络最多只能创]建24个交换机	€刷新	创建交换机
交换机ID ▼ 请输入交换机ID进行转	蒋确查询	搜索								
交换机 ID/名称	ECS实例数	网段	状态	可用区	可用私有IP数	创建时间	默认交换机	描述		操作
vsw-bp19v3gdubfcbt7b1sniy -	4	192.168.0.0/16	可用	华东 1 可用区 F	65515	2017-10-26 14:34:07	否			編辑 删除 创建实例▼
							共有1条,4	再页显示:10条	« <	1 > »

6. 单击 VSwitch 右侧的删除 尝试手动删除。

本示例中,由于 VSwitch 下还有资源未释放,所以删除失败。

错误提示		\times
\otimes	该交换机下还有资源未释放,无法删除,请释放该 VPC 下的云产品资源后再删除 VPC。	
	确	Ē

手动释放该 VSwitch 下的资源,然后再次尝试删除该 VSwitch。

7. 使用类似的方法手动删除 Kubernetes 集群下所有删除失败的资源,然后再次尝试删除 Kubernetes 集群。

1.21.4 如何支持私有镜像

```
kubectl create secret docker-registry regsecret --docker-server=
registry-internal.cn-hangzhou.aliyuncs.com --docker-username=abc@
aliyun.com --docker-password=xxxxxx --docker-email=abc@aliyun.com
```

其中:

- regsecret:指定密钥的键名称,可自行定义。
- —docker-server:指定 Docker 仓库地址。
- —docker-username: 指定 Docker 仓库用户名。
- —docker-password:指定 Docker 仓库登录密码。
- —docker-email:指定邮件地址(选填)。

yml 文件加入密钥参数。

```
containers:
- name: channel
  image: registry-internal.cn-hangzhou.aliyuncs.com/abc/test:1.0
ports:
- containerPort: 8114
imagePullSecrets:
- name: regsecret
```

其中:

- imagePullSecrets 是声明拉取镜像时需要指定密钥。
- regsecret 必须和上面生成密钥的键名一致。
- image 中的 Docker 仓库名称必须和 --docker-server 中的 Docker 仓库名一致。

详情信息参见官方文档 使用私有仓库。

1.21.5 Helm 手动升级

登录到Kubernetes集群master节点,参见通过 kubectl 连接 Kubernetes 集群。

执行以下命令。

helm init --tiller-image registry.cn-hangzhou.aliyuncs.com/acs/tiller: v2.8.2 --upgrade

其中镜像地址可使用对应region的vpc域名,比如杭州region的机器就可以替换为registry-vpc.cn-hangzhou.aliyuncs.com/acs/tiller:v2.8.2。

然后等待tiller健康检查通过,就可以通过执行helm version查看版本升级情况。



这里只会升级Helm 服务端版本,客户端可以通过直接下载对应的client binary 使用。

Helm 2.8.2 client 下载地址: https://github.com/kubernetes/helm/releases/tag/v2.8.2。目前阿里云

支持的最新版本为2.8.2。

Helm 客户端和服务端版本都升级完毕后,执行helm version命令,可看到如下信息。

```
#helm version
Client: &version.Version{SemVer:"v2.8.2", GitCommit:"a80231648a
1473929271764b920a8e346f6de844", GitTreeState:"clean" }
Server: &version.Version{SemVer:"v2.8.2", GitCommit:"a80231648a
1473929271764b920a8e346f6de844", GitTreeState:"clean "}
```

1.21.6 如何支持私有镜像

```
kubectl create secret docker-registry regsecret --docker-server=
registry-internal.cn-hangzhou.aliyuncs.com --docker-username=abc@
aliyun.com --docker-password=xxxxxx --docker-email=abc@aliyun.com
```

其中:

- regsecret:指定密钥的键名称,可自行定义。
- —docker-server:指定 Docker 仓库地址。
- —docker-username: 指定 Docker 仓库用户名。
- —docker-password:指定 Docker 仓库登录密码。
- —docker-email:指定邮件地址(选填)。

yml 文件加入密钥参数。

```
containers:
    - name: foo
    image: registry-internal.cn-hangzhou.aliyuncs.com/abc/test:1.0
imagePullSecrets:
```

- name: regsecret

其中:

- imagePullSecrets 是声明拉取镜像时需要指定密钥。
- regsecret 必须和上面生成密钥的键名一致。
- image 中的 Docker 仓库名称必须和 --docker-server 中的 Docker 仓库名一致。

详情信息参见官方文档 使用私有仓库。

2 Swarm 集群

2.1 环境要求

创建 swarm 集群时,不管是新建节点,还是将已有 ECS 实例添加到集群中,集群对机器环境有一系列相关依赖。

ECS 机器规格配置

- 添加的云服务器必须与集群在同一地域并使用相同类型的网络。
- 添加已有云服务器时,请确保您的云服务器有 EIP (专有网络);否则,添加云服务器会失败。
- 容器服务不支持添加不同账号下的云服务器。

Docker Engine 要求

允许 Docker Engine 版本在 1.12 以下。确保 Docker Engine daemon 运行在每个节点上。

如果您选择 手动添加。如果机器之前安装过 Docker Engine,会提示您是否删除。如果 ECS 曾经 加入过某个集群,必须先删除 Docker Engine。

删除命令如下所示:

- Ubuntu: apt-get remove -y docker-engine, rm -fr /etc/docker/ /var/lib/ docker /etc/default/docker
- Centos : yum remove -y docker-engine , rm -fr /etc/docker /var/lib/docker

操作系统

容器服务的节点对系统有要求,推荐您使用 Ubuntu 16.04 和 Centos 7.0 64位系统。我们对这 2 个系统版本进行了非常严格的稳定性和兼容性测试。

安全组配置

对于 2017 年 2 月 28 日之后创建的容器服务集群,默认创建的安全组已经做了加固。阿里云容器 服务未设置出方向的安全组,只会设置入方向的安全组规则,参见 容器服务安全组规则。开放的规 则如下。

安全组	作用	配置策略
80、443 端口	方便用户业务的 Web 服务,可 以根据自己的需求选择放开或 者关闭	默认开启,可选择放开或关闭

安全组	作用	配置策略
ICMP 规则	用于节点间通信,方便排查问 题。有些工具也依赖 ICMP	默认开启
VPC 安全组授权对象	容器网段的基础地址,与创建 VPC 集群时选择的容器初始网 段有关,保证容器之间可以互 相通信。	自主配置网段

2.2 使用流程

完整的容器服务使用流程包含以下步骤:



步骤1:授予账号相应的角色。

详细信息参见角色授权。

步骤2:创建集群。

您可以选择集群的网络环境,设置集群的节点个数和配置信息。

步骤3:通过镜像或编排模板创建应用。

您可以使用已有的镜像或编排模板,或者新建镜像或者编排模板。

如果您的应用由多个镜像承载的服务组成,可以选择通过编排模板创建应用。

步骤 4: 查看部署后应用的状态和相应的服务、容器信息。

2.3 授权管理

2.3.1 角色授权

在用户开通容器服务时,需要授予名称为 AliyunCSDefaultRole 和 AliyunCSClusterRole 的系统默 认角色给服务账号,当且仅当该角色被正确授予后,容器服务才能正常地调用相关服务(ECS, OSS、NAS、SLB等),创建集群以及保存日志等。

使用说明

- 如果您是在2018年1月15日之前使用过容器服务的用户,系统将默认完成角色授权,详细的授权 权限内容下面的默认角色包含的权限内容。如果您之前是通过子账号使用,您需要对子账号进行 策略升级,参见子账号策略升级。
- 2018年1月15日全面接入跨服务授权后,新用户使用主账号只有进行了跨服务授权才能使用容器 服务产品。如果新用户需要授权子账号使用容器服务,需要自行前往RAM控制台进行授权,详 细操作参见使用子账号。

角色授权步骤

当您进入容器服务控制台,如果之前没有正确地给服务账号授予默认角色,则会看到如下提示。
 单击同意授权。

一云资源访问授权	
温馨提示:如需修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致CS无法获取到必要的权限。	×
CS请求获取访问您去资源的权限	
下方是系统创建的可供CS使用的角色,接仅后,CS拥有对您云资源相应的访问权限。	
AlivunCSDefaultRole	~
描述:容器服务默认使用此角色来访问您在其他云产品中的资源	
权限描述:用于容器服务(CS)默认角色的授权策略	
AliyunCSClusterRole	<u>~</u>
描述: 容器服务集群使用此角色未访问您在其他云产品中的资源	
权限描述:用于容器服务(CS) Cluster 角色的授权策略	
利約	



容器服务已经设置好默认的角色权限,如需修改角色权限,请前往 RAM 控制台角色管理中设置,需要注意的是,错误的配置可能导致容器服务无法获取到必要的权限。

2. 完成以上授权后,刷新容器服务控制台,然后就可以进行操作了。

如果您想查看 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色的详细策略信息,可以登录 *RAM* 的控制台 进行查看。

默认角色包含的权限内容

关于各个角色权限的详细信息,请参考各个产品的 API 文档。

AliyunCSDefaultRole 角色的权限内容

默认角色 AliyunCSDefaultRole 包含的主要权限信息如下:

• ECS 相关权限

权限名称(Action)	权限说明
ecs:RunInstances	查询实例信息
ecs:RenewInstance	ECS 实例续费
ecs:Create*	创建 ECS 相关资源,如实例、磁盘等
ecs:AllocatePublicIpAddress	分配公网 IP 地址
ecs:AllocateEipAddress	分配 EIP 地址
ecs:Delete*	删除机器实例
ecs:StartInstance	启动 ECS 相关资源
ecs:StopInstance	停止机器实例
ecs:RebootInstance	重启机器实例
ecs:Describe*	查询 ECS 相关资源
ecs:AuthorizeSecurityGroup	设置安全组入规则
ecs:RevokeSecurityGroup	撤销安全组规则
ecs:AuthorizeSecurityGroupEgress	设置安全组出规则
ecs:AttachDisk	添加磁盘
ecs:DetachDisk	清理磁盘
ecs:AddTags	添加标签
ecs:ReplaceSystemDisk	更换 ECS 实例的系统盘
ecs:ModifyInstanceAttribute	修改实例属性
ecs:JoinSecurityGroup	将实例加入到指定的安全组
ecs:LeaveSecurityGroup	将实例移出指定的安全组
ecs:UnassociateEipAddress	解绑弹性公网 IP
ecs:ReleaseEipAddress	释放弹性公网 IP

• VPC 相关权限

权限名称(Action)	权限说明
vpc:Describe*	查询 VPC 相关资源的信息

权限名称(Action)	权限说明
vpc:DescribeVpcs	查询 VPC 信息
vpc:AllocateEipAddress	分配 EIP 地址
vpc:AssociateEipAddress	关联 EIP 地址
vpc:UnassociateEipAddress	不关联 EIP 地址
vpc:ReleaseEipAddress	释放弹性公网 IP
vpc:CreateRouteEntry	创建路由接口
vpc:DeleteRouteEntry	删除路由接口

• SLB 相关权限

权限名称(Action)	权限说明
slb:Describe*	查询负载均衡相关信息
slb:CreateLoadBalancer	创建负载均衡实例
slb:DeleteLoadBalancer	删除负载均衡实例
slb:RemoveBackendServers	解绑负载均衡实例
slb:StartLoadBalancerListener	启动指定的监听服务
slb:StopLoadBalancerListener	停止指定的监听服务
slb:CreateLoadBalancerTCPListener	为负载均衡实例创建基于 TCP 协议的监听规则
slb:AddBackendServers	添加后端服务器

AliyunCSClusterRole 角色的权限内容

默认角色 AliyunCSClusterRole 包含的主要权限信息如下:

• OSS 相关权限

权限名称(Action)	权限说明
oss:PutObject	上传文件或文件夹对象
oss:GetObject	获取文件或文件夹对象
oss:ListObjects	查询文件列表信息

• NAS 相关权限

权限名称(Action)	权限说明
nas:Describe*	返回 NAS 相关信息
nas:CreateAccessRule	创建权限规则

• SLB 相关权限

权限名称(Action)	权限说明
slb:Describe*	查询负载均衡相关信息
slb:CreateLoadBalancer	创建负载均衡实例
slb:DeleteLoadBalancer	删除负载均衡实例
slb:RemoveBackendServers	解绑负载均衡实例
slb:StartLoadBalancerListener	启动指定的监听服务
slb:StopLoadBalancerListener	停止指定的监听服务
slb:CreateLoadBalancerTCPListener	为负载均衡实例创建基于 TCP 协议的监听规则
slb:AddBackendServers	添加后端服务器
slb:DeleteLoadBalancerListener	删除负载均衡实例监听规则
slb:CreateVServerGroup	创建虚拟服务器组,并添加后端服务器
slb:ModifyVServerGroupBackendServers	改变虚拟服务器组中的后端服务器
slb:CreateLoadBalancerHTTPListener	为负载均衡实例创建基于 HTTP 协议的 Listener
slb:SetBackendServers	配置后端服务器,为负载均衡实例后端的一组服 务器(ECS 实例)配置权重值
slb:AddTags	为 SLB 实例添加标签

2.3.2 使用子账号

使用子账号登录容器服务控制台并静进行相关操作之前,您需要赋予子账号相应的权限。

步骤1创建子账号并开启控制台登录

- 1. 登录访问控制管理控制台。
- 2. 登录访问控制管理控制台。
- 3. 单击左侧导航栏中的用户管理并单击页面右上角的新建用户。
- 4. 填写子账号的名称并单击确定。

- 5. 在用户管理页面,选择创建的子账号,单击右侧的管理。
- 6. 在web控制台登录管理中,单击启用控制台登录。
- 7. 输入登录密码并单击确定。

步骤 2 授予子账号访问容器服务的权限

1. 在用户管理页面,选择创建的子账号,单击右侧的授权。

用户管理			新建用户	こ刷新
用户名 🔻 teststs	搜索	1		
用户名/显示名	备注	创建时间		操作
teststs		2017-12-05 17:04:04	管理 授权	删除 加入组

2. 将所需的策略授权给子账号。

编辑个人授权策略						\times
添加授权策略后,该账户即具有该条。	衰略的权限	, 同·	一条授权策略不	能被重复添加。		
可选授权策略名称	类型	l		已选授权策略名称		类型
CS		ζ		AliyunCSFullAccess		系统
管理Elasticsearch的权限	2000	•		管埋容器服务(CS)的权限		
AliyunCSFullAccess	自定义		>			
AliyunACSResourcesAccess_xingy aliyun container s	自定义		<			
AliyunCSReadOnlyAccess 只读访问容器服务(CS)的权限	系统					
		Ŧ				
					确定	关闭

您可以使用系统默认的授权策略。

- AliyunCSFullAccess:容器服务的管理权限。
- AliyunCSReadOnlyAccess:容器服务的只读权限。

或者根据您自己的需要自定义授权策略并授权给子账号,参见创建自定义授权策略。

步骤 3 子账号登录容器服务控制台

使用子账号登录 容器服务管理控制台。

如果您之前已经给主账号授予了 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色,子账号可以 直接登录到容器服务管理控制台,并进行相应的操作。

如果之前您没有给主账号授予 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色,则会看到如下 提示。单击同意授权。

-资源访问授权	
薯提示:如需修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致CS无法获取到必要的权限。	
で通じな可能につる人を通知なりに	
与有关和代码中心在这些解剖的代表 下方是系统创建的可供CS使用的角色,授权后,CS拥有对您云资源相应的访问权限。	
AlianoCCDafaultBala	
AllyuncsDelautkole 描述:容器服务默认使用此角色来访问您在其他云产品中的资源	
权限描述:用于容器服务(CS)默认角色的授权策略	
AlivunCSClusterRole	
—————————————————————————————————————	
权限描述:用于容器服务(CS) Cluster 角色的授权策略	
同意核权 取消	
Variable Color	

完成以上授权后,刷新容器服务控制台,然后就可以进行操作了。

2.3.3 子账号策略升级

容器服务在2018年1月15日全面升级安全授权管理,基于STS提供跨服务授权,为您提供更加安全的服务。如果您是在2018年1月15日之前使用过容器服务的用户,系统将默认完成授权,详细的授权权限内容参见角色授权。如果您之前是通过子账号使用,您需要重新授予子账号使用容器服务的权限。

容器服务提供了自动升级子账号策略的功能,通过此功能容器服务会自动授予您的子账号 AliyunCSReadOnlyAccess 权限。您也可以选择通过RAM管理控制台来手动为您的子账号授权。

升级子账号策略

- 1. 使用主账号登录容器服务管理控制台。
- 2. 使用主账号登录容器服务管理控制台。
- 3. 在概览页面,单击右上角的子账号升级。

容器服务		概览 集群中应用、服务和容器的数目	(包含了管理服务所需容器的数目)			子账号升级
Swarm	Kubernetes					
概览		集群名称 : swarm				●运行中 ✓
应用						
服务		1	4	5	7	
集群		节点	应用	服务	容器	

4. 在弹出的确认对话框中,单击确定。

请确认		\times
0	是否对子账号策略进行升级?	
	确定	取消

升级过程中,容器服务会为您的子账号进行相应的角色授权。

如果升级失败,对话框中会显示升级失败的子账号。

请确认		\times
	是否对子账号策略进行升级? 以下子账号升级策略失败: test_cs ^② test_cs2 ^②	
		确定取消

您可以再次单击子账号升级尝试升级。或者通过RAM管理控制台手动对子账号进行授权。

通过RAM为子账号授权

- 1. 使用主账号登录容器服务管理控制台。
- 2. 使用主账号登录容器服务管理控制台。
- 3. 单击左侧导航栏中的用户管理。
- 4. 选择所需的子账号并单击右侧的授权。

访问控制 RAM	用户管理			新建用户		こ刷新
概览 用户管理	用户名 🔻 请输入用户名进行楼糊查询	搜索				
	用户名/显示名	酱注	创建时间	2		操作
策略管理	k		2017-05-22 14:25:14	管理	授权	删除 加入组
角色管理 设置			2017-10-09 15:52:01	管理	授权	删除 加入组

5. 选择所需的策略,并单击确定。

编辑个人授权策略					×
添加授权策略后,该账户即具有该条领	衰略的权限,	同-	一条授权策略不	能被重复添加。	
可选授权策略名称	类型			已选授权策略名称	类型
CS		Q		AliyunCSReadOnlyAccess	系统
管理Elasticsearch的权限		•		只读访问容器服务(CS)的权限	
AliyunCSFullAccess	自定义		>		
AliyunACSResourcesAccess_xingy aliyun container s	自定义		<		
AliyunCSFullAccess 管理容器服务(CS)的权限	系统	i.			
		Ŧ			
				确定	关闭

容器服务提供两个系统策略:

- AliyunCSFullAccess:容器服务的管理权限。
- AliyunCSReadOnlyAccess:容器服务的只读权限。

您也可以根据自己的需求创建自定义授权策略并授权给子账号,参见创建自定义授权策略。

2.3.4 创建自定义授权策略

容器服务提供的系统授权策略的授权粒度比较粗,如果这种粗粒度授权策略不能满足您的需要,那 么您可以创建自定义授权策略。比如,您想控制对某个具体的集群的操作权限,您必须使用自定义 授权策略才能满足这种细粒度要求。

创建自定义授权策略

在创建自定义授权策略时,您需要了解授权策略语言的基本结构和语法,相关内容的详细描述请参 考授权策略语言描述。

本文档以授予子账号查询、扩容和删除集群的权限为例进行说明。

操作步骤

- 1. 使用主账号登录RAM管理控制台。
- 2. 单击左侧导航栏中的策略管理并单击页面右上角的新建授权策略。

X

> { "Statement": [{ "Action": ["cs:Get*", "cs:ScaleCluster", "cs:DeleteCluster",

, "Effect": "Allow", "Resource": [

}], "Version": "1"

"acs:cs:*:*:cluster/cb2f4d _____"

上一步

新建授权策略

取消

3. 选择一个模板,填写授权策略名称并编写您的授权策略内容。

1

45678

10

11 12 13

14 }

授权策略格式定义

策略内容:

```
{
   "Statement": [{
    "Action": [
        "cs:Get*",
        "cs:ScaleCluster",
        "cs:DeleteCluster"
    ],
    "Effect": "Allow",
    "Resource": [
        "acs:cs:*:*:cluster/集群ID"
    ]
   }],
   "Version": "1"
}
```

其中:

• Action 处填写您所要授予的权限。

说明:

所有的 Action 均支持通配符。

• Resource 有如下配置方式。

- 授予单集群权限

```
"Resource": [
"acs:cs:*:*:cluster/集群ID"
]
```

- 授予多个集群权限

```
"Resource": [
"acs:cs:*:*:cluster/集群ID",
"acs:cs:*:*:cluster/集群ID"
]
```

- 授予您所有集群的权限

```
"Resource": [
"*"
]
```

其中,集群ID 需要替换为您要授权的真实的集群 ID。

4. 编写完毕后,单击新建授权策略。

表 2-1: 容器服务RAM Action

Action	说明
CreateCluster	创建集群
AttachInstances	向集群中添加已有ECS实例
ScaleCluster	扩容集群
GetClusters	查看集群列表
GetClusterById	查看集群详情
ModifyClusterName	修改集群名称
DeleteCluster	删除集群
UpgradeClusterAgent	升级集群Agent
GetClusterLogs	查看集群的操作日志
GetClusterEndpoint	查看集群接入点地址
GetClusterCerts	下载集群证书
RevokeClusterCerts	吊销集群证书
BindSLB	为集群绑定负载均衡实例

Action	说明
UnBindSLB	为集群解绑负载均衡实例
ReBindSecurityGroup	为集群重新绑定安全组
CheckSecurityGroup	检测集群现有的安全组规则
FixSecurityGroup	修复集群的安全组规则
ResetClusterNode	重置集群中的节点
DeleteClusterNode	移除集群中的节点
CreateAutoScale	创建节点弹性伸缩规则
UpdateAutoScale	更新节点弹性伸缩规则
DeleteAutoScale	删除节点弹性伸缩规则
GetClusterProjects	查看集群下的应用
CreateTriggerHook	为应用创建触发器
GetTriggerHook	查看应用的触发器列表
RevokeTriggerHook	删除应用的触发器
CreateClusterToken	创建 Token

2.4 集群管理

2.4.1 集群简介

一个集群(cluster)指容器运行所需要的云资源组合,关联了若干云服务器节点、负载均衡等云资源。

集群创建

您可以通过多种方式创建一个集群:

方法一:创建一个集群,并同时创建若干个云服务器。

您可以通过容器服务直接创建一个包含若干个新云服务器的集群。

详细信息参见创建集群。

通过此方式创建的云服务器皆为按量付费,如需要包年包月的云服务器,可以单独购买再通过接下 来的 方法二 操作。 方法二:创建一个零节点的集群并添加已有的云服务器。

1. 创建一个零节点的集群。

如果您已经在云服务器 ECS 上购买了若干个云服务器,可以在容器服务上创建一个零节点的集群。

操作方式同 方法一,您只需要选择添加已有节点。

2. 添加已有的云服务器。

您可以通过以下两种方法将已有云服务器添加到容器服务中。

• 重置云服务器的镜像,将其自动加入集群。

此种方式会重置云服务器的镜像和系统盘,需要谨慎。但是这种方式加入的服务器比较干 净。

• 在云服务器上执行脚本,将云服务器手动加入集群。

此种方式适合于不希望重置云服务器的镜像。

详细信息参见添加已有节点。

集群管理

支持集群扩容、删除、清理、连接等操作。更多详细的内容,请参考:

- 搜索集群
- 扩容集群
- 下载集群证书
- 清理集群磁盘
- 删除集群

2.4.2 集群的生命周期

表 2-2: 集群状态说明

状态	说明
待激活(Inactive)	集群创建成功,但是集群中没有节点
初始化中(Initial)	集群正在申请相应的云资源
运行中(Running)	集群申请云资源成功

状态	说明
更新中(Updating)	集群升级 Agent 版本
伸缩中(Scaling)	变更集群的节点数量
创建失败(Failed)	集群申请云资源失败
删除中(Deleting)	集群删除中
删除失败(DeleteFailed)	集群删除失败
已删除(Deleted,该状态用户不可见)	集群删除成功

图 2-1: 集群状态流转



2.4.3 创建集群

您可以在创建集群的时候同时指定云服务器的配置和个数,也可以创建一个零节点的集群,之后再 绑定其他云服务器。



如果您选择创建一个零节点集群,创建完成后,集群会处于"待激活"状态,添加云服务器后就可以 激活集群(变为"运行中"状态)。有关如何向集群中添加已有云服务器,参见添加已有节点。

使用须知

容器服务在创建集群的过程中,会进行如下操作:

- 如果您勾选自动创建负载均衡,系统会创建负载均衡,并配置 80->9080 监听。
- 创建安全组,安全组的规则如下:

VPC 网络入方向:

<	alicloud-	cs-auto-cre	at 💩 k8s_v	rpc /			教我设置	3 返回 添加安全组规则	快速创建规则 添加Cla	ssicLink安全组规则
安全组规则 安全组内实例列表	入方向	出方向							± 导入规则	▲ 导出全部规则
	授权策略	协议类型	端口范围	授权类型	授权对象	描述	优先级	创建时间		操作
	允许	全部	-1/-1	地址段访问	172.20.0.0/16	-	100	2018-04-04 16:14:09	修改描述	売隆 删除
	允许	全部 ICMP	-1/-1	地址段访问	0.0.0/0	-	100	2018-04-04 16:14:08	修改描述	売隆 删除
	允许	自定义 TCP	80/80	地址段访问	0.0.0/0	-	100	2018-04-04 16:14:07	修改描述	売隆 删除
	允许	自定义 TCP	443/443	地址段访问	0.0.0/0		100	2018-04-04 16:14:07	修改描述	克隆 删除

- 如果您已经开通了 RAM 服务,系统会创建 RAM 子账号。
- 如果您选择创建节点,系统会创建 ECS。同时为 ECS 分配公网 IP (如果是 VPC 网络,则会分 配 EIP,同时会创建相应的路由规则)。
- 集群创建过程中,容器服务会使用您设置的登录密码配置 ECS 节点。

容器服务不会保存该密码。

• VPC 节点配置失败时,容器服务会收集节点创建初始化的标准输出信息。您可以在集群的日志 信息中查看。

限制说明

- 随集群一同创建的负载均衡实例只支持按量付费的方式。
- 每个账号默认可以创建的云资源有一定的配额,如果超过配额创建集群会失败。请在创建集群前确认您的配额。如果您需要提高您的配额,请提交工单申请。
 - 每个账号默认最多可以创建5个集群(所有地域下),每个集群中最多可以添加20个节点。
 - 每个账号默认最多可以创建 100 个安全组。
 - 每个账号默认最多可以创建 60 个按量付费的负载均衡实例。
 - 每个账号默认最多可以创建 20 个EIP。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航中的集群,单击右上角的创建集群。

容器服务	Guarm	集群列表							您最多可以自	创建 5 个集群,每个集群最多可以	添加 20 个节点 刷新	创建Swarm集都	¥ •
概览	Swarm	常见问题: & 如何	创建集群 🔗 如何消	和已有云服务	器 🔗 跨可用[◎ 安成	记志服务 🔗 通过Doc	er客户端连接集群				2	
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🖉	节点个数	创建时间	Docker版本			操作
集群	1						您当前没有	集群,可以去 创 建集 群	ŧ				
节点													
网络													

- 3. 设置集群的基本信息。
 - 集群名称:要创建的集群的名称。可以包含 1~63 个字符,包括数字、中文字符、英文字符 和连字符(-)。

送 说明:

集群名称在同一个用户和同一个地域下必须唯一。

- 地域: 所创建集群将要部署到的地域。
- 可用区:集群的可用区。

ഘ		
	说明	:

您可以根据您的服务器分布情况,选择不同的地域和可用区。

* 集群名称:							
	名称为1-64个字符,	可包含数字、汉字、	英文字符,或"-"				
地域:							
	华北 1	华北 2	华东 1	华东 2	华南 1	亚太东北 1 (东京)	美国西部1(硅谷)
	欧洲中部 1 (法兰克						
	福)	亚太东南 2 (悉尼)					
可用区:	华东 1 可用区 F	•					

4. 设置集群的网络类型。目前支持专有网络 VPC 网络类型。

专有网络 VPC 需要配置相关信息。

网络类型:	专有网络	
	vpc-bp1kcm36tevkpms97 🔻	vsw-bp195kr4zv98jvzcrv5 ▼
容器起始网段	172.20.0.0/16	查看已有网段 ⑦
	不能与 VPC 使用的网段重复,创 可选值为: - 192.168.0.0/16-18 - 172.19-30.0.0/16-18 - 10.0.0.0/16-18 系统保留私网地址: 172.16/17/15]建成功后不能修改。 18/31.0.0/16

专有网络 VPC 支持您基于阿里云构建一个隔离的网络环境,您可以完全掌控自己的虚拟网络,包括自由 IP 地址范围、划分网段、配置路由表和网关等。

您指定一个 VPC、一个 VSwitchld 和容器的起始网段(Docker 容器所属的子网网段,为了便于 IP 管理,每个虚拟机的容器属于不同网段,容器子网网段不能和虚拟机网段冲突)。为了防止 网络冲突等问题,建议您为容器集群建立属于自己的 VPC/VSwitchld。

5. 添加节点。

是否新增节点	创建节点	添加已有节点

您可以在创建集群的同时创建若干个节点,或者创建一个零节点集群并添加已有云服务器。有关如何添加已有云服务器的详细信息,参见添加已有节点。

- 创建节点
 - 1. 设置节点的操作系统。

操作系统:	CentOS 7.4 64位	• 0	

目前支持的操作系统包括 Ubuntu 16.04 64 位和 CentOS 7.4 64 位。

2. 设置云服务器的实例规格。



您可选择不同的实例规格和数量,并指定数据盘的容量(云服务器默认带有 20G 大小的 系统盘)和登录密码。集群创建过程中,容器服务会使用您设置的 登录密码 配置 ECS 节 点,但是容器服务不会保存该密码。

📃 说明:

- 如果您选择了数据盘,它会被挂载到 /var/lib/docker 目录,用于 Docker 镜像和 容器的存储。
- 从性能和管理考虑,建议您在宿主机挂载独立的数据盘,并利用 Docker 的 volume 对 容器的持久化数据进行管理。

• 添加已有节点

您可以单击下边的选择已有实例将已有的云服务器添加到集群中,或者直接单击创建集群等 集群创建完成后再通过集群列表页面添加已有云服务器,参见添加已有节点。

6. 配置 EIP。

当您将网络类型设置为 VPC 时,容器服务会默认给每一个专有网络下的云服务器配置一个 EIP。如果不需要,您可以勾选不保留公网EIP复选框,但是需要额外配置 SNAT 网关。

每个账号最多可申请 20 个EIP。当您创建集群时选择使用 VPC 网络且选择系统自动创建 EIP 时,如果您账号下的 EIP 已达到配额,创建集群会失败。

EIP :	□ 不保留公网EIP
	不保留公网EIP,完成实例初始化后会释放EIP,可以使用阿里云提供的 NAT网关产品实现VPC安全访问公网环境,您也可以自行配置SNAT(请参考以下文档)。未配置 SNAT会导致VPC不能正常访问公网,会影响集群创建和应用部署等。请参考: Linux系统配置SNAT

7. 创建一个负载均衡实例。

负载均衡:	✓ 自动创建负载均衡
	创建集群会默认创建一个公网负载均衡实例,计费类型为 按量付费

目前创建集群会默认创建一个负载均衡实例。您可以通过这个负载均衡实例访问集群内的容器应 用。所创建的负载均衡实例为按量付费实例。

8. 在 ECS 节点上安装云监控插件。

您可以选择在节点上安装云监控插件,从而在云监控控制台查看所创建 ECS 实例的监控信息。

云监控插件: C 在ECS节点上安装云监控插件						
在节点上安装云监控插件,	可以在云监控控制台查看所创建ECS实例的监控信息					

9. 将节点 IP 添加到 RDS 实例的白名单。

您可以选择将所创建节点的 IP 添加到 RDS 实例的白名单中,方便 ECS 实例访问 RDS 实例。



- 建议您选择创建节点时进行该项配置。
- 如果您选择添加已有节点,对该项进行配置。在集群创建界面就添加已有 ECS 实例,配置 可用;在创建零节点集群后添加 ECS 实例,该配置不生效。
- 您仅能将 ECS 实例的 IP 添加到位于同一地域的 RDS 实例的白名单中。
- a. 单击请选择您想要添加白名单的RDS实例。

RDS白名单: 请选择你想要添加白名单的RDS实例

b. 在弹出的对话框中选择所需的 RDS 实例并单击确定。

添加到RDS实例的白名单			×
☑ 实例Id	引擎版本	可用区	网络类型
m-m5ekyxwn71w8s2mfx	MySQL 5.6	cn-qingdao-b	经典网络
			确定取消

10.单击创建集群。

集群创建成功后,如果需要单独配置云服务器或负载均衡,可以去相应的控制台进行相关操作。

后续操作

您可以查看集群创建日志。在集群列表页面,选择所创建的集群并单击查看日志。

容器服务		集群列表						您最多可以创	皇 5 个集群,每个集群最多。	J以添加 20 个节点)	剧新	创建Swarm集	₩ •
Kubernetes	Swarm												
概览		常见问题: ③ 如何创建集群 ⑧ 如何添加已有	云服务器 🔗 🛛	跨可用区节	5点管理 🔗 集成日志服务 🔗 通过Do	ocker客户端连	妾集群						
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本			攝作
集群		test-swarm	阿里云集群	华东1	虚拟专有网络	●运行中	健康 😂	1	2018-04-04 16:14:07	17.06.2-ce	管理	査看日志 监控	删除 更多 -
节点													
网络													

您可以在创建的集群中创建应用。有关创建应用的详细信息,参见创建应用。

参考文档

如果集群创建失败,您可以参考创建集群失败常见错误进行问题排查。

2.4.4 集群参数配置说明

本文档旨在帮助您理解创建集群界面的参数含义,顺利进行参数配置,其中对一部分参数,会给出一些文档资源,供您了解更多信息。

集群名称

设置该集群的名称。

- 名称为 1-63 个字符,可包含数字、汉字、英文字符,或"-"。但不能以分隔符(-)开头。
- 您可以在创建成功后,在集群列表页面对目标集群的名称进行修改。

地域和可用区

容器服务授权创建 ECS 实例的地域和可用区,目前容器服务支持的地域和可用区属于 ECS 产品的 子集。请参见 地域和可用区。

网络类型

选择 ECS 实例的网络类型,支持 VPC 网络,阿里云 VPC 支持您创建自定义的专有网络,不同的专 有网络之间二层逻辑隔离,同时您可以灵活规划各个集群的网段,非常适合大规模容器集群的使用 场景,提供了更高的安全性和灵活性。为了更好的保障系统安全和混合云业务的支持,从2018年1 月1日起不再支持经典网络、非IO优化实例的新集群创建。

容器起始网段

该参数只在您选择 VPC 网络时,才会要求您进行配置。在规划时,需要保证容器的初始网段和 VPC 网段不能重叠。

- 每个专有网络只能指定1个网段,其中172.16.0.0/12 是专有网络的默认网段。
- 在创建 VPC 网络的容器服务集群时,需要指定对应的容器网段,目前容器服务支持的容器网段
 是 192.168.1.0/24 和 172.[16-31].1.0/24

是否新增节点

容器服务提供两种新增节点的方式:创建节点和添加已有节点。创建节点授权容器服务在创建集群时自动创建 ECS 实例,自动将新建的 ECS 实例加入到集群中。添加节点支持将已有的 ECS 实例 添加到集群中,其中,可以在集群创建页面直接进行添加,或者创建好零节点集群后,在集群页面进行添加,详情参考 添加已有节点。

节点类型

默认为按量付费。等待创建节点成功后,您可以前往 ECS 控制台转换为包年包月实例。

操作系统

设置 ECS 实例中安装的操作系统,推荐使用 Ubuntu 16.04 64位 和 CentOS 7.4 64位。

实例系列

不同系列对应不同的实例规格族。ECS 实例是以一定的规格来为您提供相应的计算能力的。根据业务场景和使用场景,ECS 实例可以分为多种规格族。具体各个实例规格族的使用场景,请参见 实例规格族。

实例规格

ECS 实例规格定义了实例的 CPU 和内存的配置这两个基本属性。但是, ECS 实例只有同时配合磁盘、镜像和网络类型,才能唯一确定一台实例的具体服务形态。

实例数量

创建 ECS 实例的数量,单个集群的 ECS 实例数量不超过 20 台机器。不建议创建单节点集群,以 提高集群的可用性。控制台默认设置为 2 个节点。

系统盘类型

选择安装系统的云盘类型,支持高效云盘和 SSD 云盘。根据您对 ECS 实例的系统性能要求,进行选择,两种云盘的性能指标对比,请参考 块存储性能。

数据盘配置

选择容器挂载的数据盘类型,需要勾选**挂载数据**盘,并选择数据盘容量,才算真正配置完毕。数据 盘会被挂载到容器的 /var/lib/docker 目录,用于存储镜像和容器数据。

登录密码和确认密码

设置和确认 ECS 节点的登录密码。8-30 个字符,且同时包含三项(大、小写字母,数字和特殊符号),您在 ECS 控制台登录或 SSH 到 ECS 实例上进行管理时需要用到。

┋ 说明:

- 集群创建过程中,容器服务将使用该密码配置 ECS 节点,容器服务不会保存该密码。
- 密码为初始化使用,请妥善保管。

EIP

EIP 即是弹性公网 IP,用于公网访问。默认保留,如果选择不保留,集群在完成实例初始化后释放 EIP 。您可用 NAT 网关或自行 Linux系统配置SNAT 实现公网访问。

负载均衡

创建集群会默认创建一个公网负载均衡实例,计费类型为按量付费,用于流量分发控制服务,实现 服务高可用。

云监控插件

勾选则在 ECS 节点上安装云监控插件,从而监控集群下 ECS 实例操作系统级别的性能指标。

RDS 白名单

您可以选择将所创建节点的 IP 添加到 RDS 实例的白名单中,方便 ECS 实例访问 RDS 实例。

• 建议您选择 创建节点 时进行该项配置。

- 如果选择在添加已有节点时进行该项配置,请在集群创建界面就添加已有 ECS 实例,该配置生效;如果在创建零节点集群后添加 ECS 实例,该配置不生效。
- 您仅能将 ECS 实例的 IP 添加到位于同一地域的 RDS 实例的白名单中。

默认安全组信息

容器服务设置了默认的安全组,容器服务只设置入方向的安全组规则,您可以在集群创建成功后,根据您的业务场景对安全组进行配置。具体请参见容器服务安全组规则

- 443端口和80端口可以根据自己的需求选择放开或者关闭。
- ICMP规则建议保留,用于节点间通信,方便排查问题。有些工具也依赖ICMP
- 务必确保放开所有您需要的端口,否则会导致服务不可访问。通过 SLB 访问的端口不需要放 开。

2.4.5 添加已有节点

您可以将已购买的云服务器添加到指定集群。

| ■ 说明:

默认情况下,每个集群中最多可添加20个节点。如果您需要添加更多节点,请提交工单申请。

您可以通过以下两种方法之一添加已有云服务器:

- 自动添加:通过此方法添加实例会重置镜像和系统盘。您可以选择一次添加一个或多个云服务器。
- 手动添加:在云服务器上执行脚本。您一次仅能选择添加一个云服务器。

前提条件

如果之前没有创建过集群,您需要先创建一个集群。有关创建集群的详细信息,参见创建集群。

使用说明

- 添加的云服务器必须与集群在同一地域并使用相同类型的网络(专有网络)。
- 添加已有云服务器时,请确保您的云服务器有 EIP (专有网络),或者相应 VPC 已经配置了
 NAT网关。总之,需要确保相应节点能正常访问公网,否则,添加云服务器会失败。
- 容器服务不支持添加不同账号下的云服务器。
- 如果您选择手动添加,请注意以下事项:

如果您的云服务器中已经安装了 Docker,手动添加的时候可能会失败。建议在添加云服务器
 之前执行清理命令。命令如下所示:

Ubuntu: apt-get remove -y docker-engine, rm -fr /etc/docker/ /var/ lib/docker /etc/default/docker

CentOS: yum remove -y docker-engine, rm -fr /etc/docker /var/lib/ docker

 一 容器服务的节点对系统有要求,推荐您使用 Ubuntu 14.04/16.04 和 CentOS 7 64位系统。我 们对这两个系统进行了非常严格的稳定性和兼容性测试。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下, 单击左侧导航栏中的集群。
- 3. 选择所需的集群,单击更多并在弹出菜单中单击添加已有节点,如下图所示。

容器服务		集群列表				您最	多可以创建 5 个领	輕群,每个绑	聽群最多可以添加 20 个节	点子账号授权	刷新创建集群 •
KWarm	Kubernetes	常见问题: Ø如何创建集群 Ø如何添	加已有云服务器	8時	可用区节点管理 🔗 集成日志服务	客 🔗 通过Do	cker客户端连接第	顒群			
应用		名称 ▼									
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本	2 操作
集群	1	routing-test-online	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●待激活❷	暂无节点状态	0	2017-11-23 18:10:29	17.06.2-ce	管理 查看日志 删除 监控 更多•
网络		test-rds no-instrumental address rice-portion	阿里云集群	华东1	虚拟专有网络 vpc-bp1gnu8br4ay7beb2wxl8	●运行中	健康 🕄	1	2017-11-23 17:30:07	17.06.2-ce	更新访问控制授权信息
 	変	swarmmode chartotocharci (2005) control Screets swarm mode	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	3	2017-11-22 10:37:07	17.06.2-ce	升级Docker 曾 升级系统服务 集群扩容
镜像编排模	扳	test-swarm chtCrebtCSclavKCMbHabStMaasfielD	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	2	2017-11-20 14:29:16	17.06.2-ce	智 添加已有节点 3 创建应用

4. 添加 ECS 实例。

页面显示的实例列表,是根据集群所定义的地域和网络类型,从您的所有云服务器列表中筛选后 同步过来的。

您可以通过以下两种方法之一添加实例。

• 自动添加



通过此方法添加实例会重置镜像和系统盘。请谨慎使用。添加之前请创建快照进行数据备份。有关如何创建快照,参见创建快照。

1. 选择要添加的实例,单击下一步。

您可以同时添加一个或多个实例。

- 2. 设置实例信息,单击下一步并在弹出的确认对话框中单击确定。
- 3. 单击完成。
- 手动添加,您需要在云服务器上执行脚本。
 - 1. 选择手动添加,选择一个 ECS 实例并单击下一步。

您一次只能添加一个实例。

- 2. 确认实例信息并单击下一步。
- 3. 页面显示专属这台云服务器的脚本命令。单击登录云服务器xxxxxxx。

	选择已有云服务器实例	>	填写实例信息	添加完成	
\bigcirc	to <u>ken牛成成功!</u> # <mark>醫症去服务器 Hop1</mark>	以下命令:		 	i.
	去执行该命令				
					完成

4. 弹出的对话框中显示远程连接密码,复制并妥善保存该密码,并单击关闭。

远程连接	密码	×
•	远程连接密码:	
	警告! 远程连接密码只出现一次,您以后每次远程连接登录时都需要输入该密码,请做好记录存档工作。	
	关闭	

5. 在弹出的对话框中,输入远程连接密码并单击确定。

输入远程连接密码	×
*请输入远程连接密码:	•••••
	确定取消

6. 输入登录云服务器的账号 (root) 和密码并回车, 登录到云服务器。



7. 单击复制命令输入,将上面的脚本命令黏贴到弹出的对话框中,单击确定并回车。

复制粘贴命令	2	×
将内容粘贴至文本框中,最大支持2000个字符,暂不支持中文等非标准键盘	且值特殊字符。	
* 文本内容: curl -Ls http://aliyuncontainerservice.oss-cn- hangzhou.aliyuncs.com/17.03.1-ce/attachNodeSc bash -s <u>4f7bc</u> instance-id i- <u>bp</u>	ript I sudo -H	
	确定取消	

系统开始执行脚本。等待脚本执行成功,显示 success。该云服务器即添加成功。

发送远程命令+	成功连接到实例i-bp	提示:如果出现持续黑屏	说明系统处于休眠状态	按任意键可以激活。	复制命令输入:	修改远程连接密码
(1998)	stbishtersenses to upgraded. I source in the installed: user in the following NEW packages will be installed: user in the installed of the installed: organaded. I source installed of the installed: organaded. I source installed of the used. Get: Intry:/introme.aligun.com/ubmetu xenial/sain end64 usely and64 6.0-20ubmetu1 Tetched to BA Bi of expression and the used of the used. Get: Intry:/introme.aligun.com/ubmetu xenial/sain end64 usely and64 6.0-20ubmetu1 Tetched to BA Bi of expression and the user of the used. Get: Intry:/introme.aligun.com/ubmetu xenial/sain end64 usely and64 6.0-20ubmetu1 Tetched Tig to unspect	揭示:如果出现9%表票屏 (150 kB)	说明系统处于休眠状态	接任會總可以撤活。	夏创命令输入:	修改运程连续运行
	<pre>inf lating: retc/docker/agent-kg, pen inf lating: retc/docker/agent-kg, pen inf lating: retc/docker/agent-kg, pen inf lating: retc/docker/agent/cs pn inf lating: retc/docker/agent/cs inf lating: retc/do</pre>	nit nay be dangerous. .9-30ba369' locally cummel-agent:0.9-30ba3 unnel-aggertes. fad' locally	99			
	Status: Rounicoded never Indeg for registry-internal.cn-haugzbou.aligancs.cov/acs/o c/Viete/140/2100301110.load/9771ba/7etc3/5405513al/s4cd5bda622b1d auccess SUCCES	igent :0.9-979af ad				

相关操作

您可以在脚本执行页面修改您的云服务器远程连接密码。单击修改远程连接密码,在弹出的对话框中输入新密码并单击确定。

修改远程连接密码							
(提示:本实例为非1/0(尤化实例,修改VNC密码后, 只有重启实例后才能生效。					
	*请输入新密码:	•••••• 限制为6位。支持数字和大小写字母。不支持特殊字 符。					
	*再次确认新密码:	•••••					
		确定取消					

2.4.6 跨可用区节点管理

为了提高应用的高可用性,在创建集群的时候,可以选择将多个节点分布在不同的可用区。

在创建集群的时候,您可以先创建一个节点的集群或者直接创建零节点的集群,待集群创建完成 后,通过集群扩容或者添加已有 ECS 实例的方式来增加不同可用区的节点。

门 说明:

- 通过集群扩容添加的节点为按量付费节点。
- 通过添加已有实例添加的节点可以是按量付费节点也可以是包年包月的节点。

通过集群扩容添加不同可用区的节点

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 单击左侧导航栏中的集群。
- 3. 选择要扩容的集群,单击更多 > 集群扩容。如下图所示。

容器服务	集群列表					您最多可以创發	建5个集群,	每个集群最多可以添加 2	0 个节点 子账号报	釵	刷新	创建集群
概览	常见问题: & 如何创建集群 & 如何添加で	日有云服务器 🔗	跨可用区节点管理	𝔗 集成日志服务	Ø 通过Docke	r客户端连接集	B¥					
应用	名称 ▼											
服务	集群名称/ID	集群类型 地	城 网络类型		集群状态	节点状态 🕖	节点个数	创建时间	Docker版本		- (2	2 操作
集群 1	routing-test-online 🖍	阿里云集群 华	东1 虚拟专有网络	surgan1fm6pubdR	●运行中	健康 ℃	2	2017-11-06 09:11:29	17.06.2-ce	管理	查看日志	
节点			vpc-op1w52	wjrgzorinnopubuo						更	mire 新访问控制授	权信息
网络										升	级Agent	
数据卷										升!	级Docker	
配置项										71	级系统服务	
▼ 镜像与方案									3	集	群扩容	
镜像										添	加已有节点	
编排模板										Û	建应用	

4. 在弹出的对话框中,设置新节点的规格。

您可以通过设置可用区创建分布在不同可用区的节点。

	集群名称:	test	
集群	地域:	华东1	
	可用区:	华东 1 可用区 E ▼	

5. 单击集群扩容将新节点添加到集群中。

6. 重复以上步骤,创建位于不同可用区的节点并添加到集群中。

通过添加已有实例添加不同可用区的节点

前提条件

使用本方法来添加节点,您需要首先通过 ECS 的售卖页面自行购买 ECS 实例。购买的过程中可以 为实例选择不同的可用区。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 单击左侧导航栏中的集群。
- 3. 选择要添加已有节点的集群,单击更多 > 添加已有节点。如下图所示。

容器服务	集群列表					您最多可以创	建 5 个集群	,每个集群最多可以添加;	20 个节点 子账号	授权	刷新	创建集群
概览	常见问题: 🖉 如何创建集群	◊ 如何添加已有云服务器	<i>8</i> 跨可	用区节点管理 🔗 集成日志服务	Ø 通过Dock	er客户端连接集	群					
应用	名称 ▼											
服务	集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕗	节点个数	创建时间	Docker版本		- 6	2 操作
集群 1	routing-test-online 🖌	阿里云集群	华东1	虚拟专有网络 was_ba1w02wizaz01fm6aubd8	●运行中	健康 ℃	2	2017-11-06 09:11:29	17.06.2-ce	管理	查看日志	删除
节点				vpc-op1w92wjig201iiiiopubus						j.		設置を
网络										Ŧ	十级Agent	
数据卷										Ŧ	十级Docker	
配置项										Ŧ	十级系统服务	
▼ 镜像与方案											載群扩容	
镜像										300	和已有节点	
编排模板										ŧ	別建应用	

4. 选择位于不同可用区的 ECS 实例并自动或手动将其添加到集群中。

有关添加已有实例的详细信息,参见添加已有节点。

选择已有云服务器 :			
实例名称/ID	IP地址	可用区	网络类型
实例名称			م
iZ2zebe6tlwmae4 i-2zebe6tlwmae4	(公) 5 (内	cn-beijing-a)	经典网络
c4bfecab22d134e i-2zed2ymaa0e3z	公) (内)	cn-beijing-c	经典网络
c806a6f7c83e449 i-2zeaaey7a7oou	公) (内)	cn-beijing-c	经典网络

5. 重复以上步骤,为集群添加位于不同可用区的节点。

2.4.7 为集群绑定和解绑负载均衡

您可以在创建集群时自动创建一个按量付费的负载均衡实例,或者在集群创建完成后为集群绑定一 个包年包月或按量付费的负载均衡实例。

容器服务集群支持绑定公网负载均衡实例和私网负载均衡实例。

使用限制

- 您只能为集群绑定位于同一地域下的负载均衡实例。
- 不支持绑定跨账号的负载均衡实例。
- VPC 集群支持绑定公网负载均衡实例和 VPC 私网负载均衡实例。
- 一个集群仅支持绑定一个负载均衡实例。
- 两个集群不能共用一个负载均衡实例。

前提条件

您已经在 负载均衡管理控制台 创建了负载均衡实例并且为负载均衡实例配置了监听后端服务器的 TCP 9080 端口。

有关如何创建,参见创建负载均衡实例。

绑定负载均衡实例

- 1. 登录 容器服务管理控制台。
- 2. 选择要配置的集群并单击管理。
| 容器服务 | | 集群列表 | | | | | 您最多可以 | 创建 5 个集 | 群,每个集群最多可以 | 添加 40 个节点 | 刷新 | 创建Swarm集群 | - |
|------------|-------|-------------------|----------|--------------|-----------------|---------|-----------|----------|------------|--------------|----|-----------|------|
| Kubernetes | Swarm | | | | | | | | | | | | |
| 概览 | | 常见问题: ②如何创建集群 ③如何 | 可添加已有云服务 | 諸 🔗 跨可月 | 目区节点管理 🔗 集成日志服务 | ; 🔗 通过D | ocker客户端述 | 接集群 | | | | | |
| 应用 | | 名称 ▼ | | | | | | | | | | | |
| 服务 | | 集群名称/ID | 集群类型 | 地域 (全部)
▼ | 网络美型 | 集群状态 | 节点状态
2 | 节点个
数 | 创建时间 | Docker版
本 | 2 | | 操作 |
| 集群 | 1 | tort-rafa | 府用于使 | | 虚拟专有网络 | | | | 2018-05-21 | | 管理 | | ₩I8÷ |
| 节点 | | | 群 | 华东1 | vpc- | + | 健康 С | 1 | 10:29:11 | 17.06.2-ce | | 监控] | 更多▼ |
| 网络 | | | | | | | | | | | | | |

3. 单击左侧导航栏中的负载均衡 > 绑定SLB。

<	集群: testsafe 刷新	
基本信息		
负载均衡	负载均衡设置 域名设置	
用户标签	您的集群目前没有绑定SLB。	
镜像列表		
容器列表		
集群事件		
节点伸缩		

4. 从下拉列表中选择您要绑定的负载均衡实例并单击确定。



如果您所选择的负载均衡实例已经绑定了后端服务器,系统会提示您"该负载均衡实例已经绑定 了后端服务器",您需要选择其它未绑定后端服务器的负载均衡实例。

重新绑定负载均衡实例

您可以根据您的需要更换集群绑定的负载均衡实例。

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的集群。
- 3. 选择要配置的集群并单击管理。

容器服务		集群列表				您最多可以创	建 5 个集群	,每个集群最多可以;	泰加 40 个节点	刷新	创建Swarm集群	-
Kubernetes	Swarm											
概览		常见问题: ③如何创建集群 ③如	阿添加已有云服务器	▶ 跨可用区节点管理 🔗 集成日日	志服务 🔗 通过Do	ocker客户端连接	逐集群					
应用		名称 ▼										
服务		集群名称/ID	地域 (: 集群类型 ▼	全部) 网络炭型	集群状态	节点状态 2	节点个 数	创建时间	Docker版 本	2		操作
集群		test-safe	阿里云集 群 群	虚拟专有网络 vpc-	●运行 中	健康 🕽	1	2018-05-21 10:29:11	17.06.2-ce	管理	查看日志 监控	删除 更多▼
网络												

4. 单击左侧导航栏中的负载均衡 > 重新绑定SLB。

<	集群:testsafe	助新	f
基本信息			
负载均衡	负载均衡设置	域名设置	
用户标签	色動物演びすい	lh bo	
镜像列表	风载时间10.		
容器列表		重新旗定SLB 解源SLB	

5. 从下拉列表中选择您要绑定的负载均衡实例并单击确定。

解绑负载均衡实例

如果您不再需要负载均衡实例,您可以通过容器服务管理控制台进行解绑。

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的集群。
- 3. 选择要配置的集群并单击管理。

容器服务		集群列表					您最多可以(创建 5 个集群	¥,每个集群最多可以;	泰加 40 个节点	刷新	创建Swarm集群	-
Kubernetes	Swarm												
概点		常见问题: ② 如何创建集群 ③ 如何	添加已有云服务	器 🔗 跨可用	区节点管理 🔗 集成日志服务		ocker客户端连	接集群					
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域 (全部) ▼	网络类型	集群状态	节点状态 2	节点个 数	创建时间	Docker版 本	2		操作
集群	1	test-safe	阿里云集 群	华东1	虚拟专有网络 vpc-	●运行 中	健康 ℃	1	2018-05-21 10:29:11	17.06.2-ce	管理	查看日志 监控	删除 更多▼
网络													

4. 单击左侧导航栏中的负载均衡 > 解绑SLB。

<	集群:testsafe		刷新
基本信息			
负载均衡	负载均衡设置	城名设置	
用户标签	奇裁物領は・	h.b.	
镜像列表	AND DIG .		
容器列表		重新绑定SLB 解绑SLB	

2.4.8 设置集群根域名

背景信息

当您通过镜像创建 Nginx 并进行简单路由配置时,您只需要填写域名的前缀 nginx,即可获得 \$ cluster_id.\$region_id.alicontainer.com 格式的域名。您可以通过设置集群根域名(本 示例使用 toolchainx.com)来替换该域名。当您重新部署服务 nginx 时,域名从 nginx. c2818a77aac20428488694c0cd1600e6e.cn-shenzhen.alicontainer.com 变为 nginx. toolchainx.com,方便您使用自己的根域名对集群应用进行访问。

- 说明:

为了保证下面的示例能够工作,请先将 Agent 升级到最新版本。

操作步骤

- 1. 绑定一个负载均衡实例。
 - a) 登录 容器服务管理控制台。
 - b) 登录容器服务管理控制台。
 - c)在Swarm菜单下,单击左侧导航栏中的集群。
 - d) 选择要配置的集群 (本示例为routing-test-online), 单击管理。

容器服务		集群列表					您最多可以创]建 5 个集團	詳,每个集群最多可以消	泰加 40 个节点	刷新	创建Swarm集群	-
Kubernetes	Swarm												
概览		常见问题: ③ 如何创建集群 ③ 如何	添加已有云服务器	6 時可用区节点管理			cker客户端连	妾集群					
应用		名称 ▼											
服务		集群名称/ID	地域 集群类型 👻	t (全部) 网络类型		集群状态	节点状态 ❷	节点个 数	创建时间	Docker版 本	2		操作
集群		and the back college	四田二曲	虚拟专有	网络				2010.05.21		-		8154
节点		routing-test-online	阿里云栗 华东 群	1 vpc-		中	健康 오	1	10:29:11	17.06.2-ce	1 BHE I		180. 18 -

e) 单击左侧导航栏中的负载均衡。

如果集群未绑定负载均衡实例,登录阿里云负载均衡管理控制台,并创建一个负载均衡实例,然后回到本页面进行绑定。

1 说明:

有关负载均衡在容器服务中的使用限制以及如何为集群绑定和解绑负载均衡,参见为集群绑 定和解绑负载均衡。

<	集群:routing-te	est-online	制新
基本信息			
负载均衡	负载均衡设置	域名设置	
用户标签	合裁り海ばす	lb.bo	
镜像列表	yan yan yan yang yang yang yang yang yan	עטיטו	
容器列表		重新绑定SLB 解散SLB	

- 2. 设置域名。
 - a) 单击 域名设置,填写您自己购买的根域名,本示例中为toolchainx.com。

<	集群 : routing-test-c	online		刷新
基本信息				
负载均衡	负载均衡设置	域名设置		
用户标签	械 空 ·		teal/hainy.com	
镜像列表				
容器列表			设置 取消	

- b) 单击设置。
- 3. 将域名解析到绑定的负载均衡实例。
 - a)在 SLB 控制台中,查找集群绑定的负载均衡实例。

b) 查看实例详情,找到绑定的负载均衡实例的 IP 地址。

<	■ slb_redis_app [t 返回负载均衡列表]	∂ 使用限制和注意事项
详情	基本信息	*
▶ 服务器	负载均衡ID: lb-bp12kb2yyydi4qev26fwz	状态: ❷运行中
监控	负载均衡名称: slb_redis_app	地域: 华东1
	地址类型: 公网	可用区: 华东1可用区 F(主)/华东1可用区 E(番)
	网络类型: 经典网络	
	付裁信息	消费明细 释放设置 个
	付歲方式: 按使用流量	创建时间: 2017-08-01 14:04:25
	服务地址:	自动释放时间: 无

c) 登录阿里云云解析产品页面,添加 A 记录,将*.toolchainx.com解析到负载均衡的 VIP 地址。

toolchainx.	com	使用时限: 正常服务期 ^①
产品信息	解析设置	
解析设置	添加解析 批量导入解析 导出解析行记录 新手引导设置	快速搜索解析记录
批量导入解析	建议您在电脑上修改公共DNS,让解析设置实时生效。 下做DNS修改工具 什么是公共DNS,如何修改?	K
负载均衡	□ 记录类型▲ 主机记录▲ 解析线路▲ 记录值 MX优先级▲	TTL 状态 操作
CDN加速 解析日志	▲ ③ * 就认 ◆ 118.178.0.249 ③ 温馨提示: 要将域名example.com解析为www.example.com , 在主机记录(RR)处填写www即可。 主机记录就是域名前暖,常见用法有:	10分钟 🕤 🦷 保存 取消 🖬 🕈
	www :将城名解析为www.example.com,填写www ; ④ :将城名解析为example.com(不带www) ,填写@成者不填写 ; mail :将城名解析为mail.example.com ,通常用于解析邮箱题务器 ; • :这解析 ,所有于城名均被解析到统一地址(除岸独设置的子城名解析) ;	
	二級城名 : 知 : mail.example.com.藏asample.com , 填写mai或abc ; 手机网站 : 知 : m.example.com , 填写m。	不再提示~

- 4. 重新部署 web 服务。
 - a) 重新部署应用wor,应用下的服务web访问端点发生了变化。

设置根域名之前的访问端点:

服务: \	wor_web									刷新	调	整容器数量
基本信	言息											
服务名	i称: web		所在应用:	wor	镜像: re	gistry.aliyuncs.com/acs-sample/wc	ordpress:4.5		容器数目: 3			储
访问端	i点: http://wo	rdpress.cf5	ib5cc0c5ccb	04214984d1	1116f3471dd2.cn-har	ngzhou.alicontainer.com						
容器	日志	EE 3	事件									
名称/ID		状态	健康	雨检测 镜	像	端口	容器IP	节点IP				操作
wor_web 2ce8241	o_1 ∅ 9fa59be79	Up 8 mir	nutes 正常	sha	gistry.aliyunc a256:08488f68c	120.26.244.234:32776->80/tcp	172.19.0.8	120.26.244.234	删除 停止 .	监控 [3志	远程终端
wor_web 33a0d6f	5_2 () 86f654a02	Up 8 mir	nutes 正常	sha	gistry.aliyunc na256:08488f68c	120.26.244.234:32775->80/tcp	172.19.0.9	120.26.244.234	删除 停止 .	监控 [3志	远程终端
wor_web afcd82fa	5_3 () 11d62121d	Up 8 mir	nutes 正常	s reg	gistry.aliyunc na256:08488f68c	120.26.244.234:32774->80/tcp	172.19.0.10	120.26.244.234	删除 停止 .	监控	3志	远程终端

设置根域名之后的访问端点:

服务: ・	wor_web								刷新	调整容器数量
基本信	言息									
服务名	称: web		所在	应用: wor	镜像	registry.aliyuncs.com/acs-sample/wo	ordpress:4.5		容器数目: 3	●更新中
访问翊	点: http://	/wordpres	s.toolchain	x.com						
容器	日志	配置	事件							
名称/ID		状态		健康检测	镜像	端口	容器IP	节点IP		操作
wor_web 2ce8241	o_1 0 9fa59be79.	Up	9 minutes	正常	registry.aliyunc sha256:08488f68c	:32776->80/tcp	172.19.0.8	121.25244.834	删除 停止 监控	日志 远程终端
wor_web 33a0d6f	_2 ∅ 86f654a02.	Up :	9 minutes	正常	registry.aliyunc sha256:08488f68c	::32775->80/tcp	172.19.0.9	0120244334	删除 停止 监控	日志 远程终端
wor_web afcd82fa	5_3 () 11d62121d.		9 minutes	正常	registry.aliyunc sha256:08488f68c	:32774->80/tcp	172.19.0.10	01.01.04.014	删除 停止 监控	日志 远程终端

b) 访问最新的访问端点http://wordpress.toolchainx.com。

→ X b wordpress.toolchainx.com/?s	pm=5176.2020520152.0.0.d9KExe		☆ ABP		0 1
hello sib ர—∱WordPressங்ள					
世界,您好! 2016年6月3日 ^{和各联论}	欢迎使用WordPress。这是您的第一篇文章。编辑或删除它,然后开始写作吧!	Search		٩	
тіяти		近期文章 世界, 忽好! 近期评论			

2.4.9 下载集群证书

背景信息

您可以使用下载的证书通过 Docker Swarm API 或 Docker Client 连接集群暴露出来的 Endpoint,参见通过 *Docker* 工具连接集群。

- 1. 获取访问地址。
 - a) 登录 容器服务管理控制台。
 - b) 登录容器服务管理控制台。

c) 在 Swarm 菜单下,单击左侧导航栏中的集群,在集群列表中选择一个集群并单击管理。

容器服务		集群列表						您最多可	以创建 5 个集群,每个集群	#最多可以添加 20 个	节点 刷象	f 创建集	群・
- Swarm Ku 概览	ubernetes	常见问题: & 如何创建集群 & 如何添加E	3有云服务器 (8 跨可用[≤节点管理 🖉 集成日志服务	Ø 通过Docker 都 □	§户端连接集群						
应用		名称 *											
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 📀	节点个数	创建时间	Docker版本			操作
集群 节点	1	test-swarmmode 🖍	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	3	2017-11-07 15:26:23	17.06.2-ce	2	查看日志 监控	
网络 数据卷		test-swarm	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	2	2017-11-01 16:11:29	17.06.2-ce	管理	查看日志 监控	删除 更多+

d) 您可以查看集群的连接信息, 如下图所示。

集群 : test-swarm				开启日志服务	登录镜像仓库 刷新
基本信息				升级Agent	升级系统服务 清理磁盘
集群ID: chc3llccf034348e4e654348e78190c1	虚拟专有网络	●运行中	地域: 华东1	节点个数 2	集群扩容 添加已有节点
安全组ID: sg-bp12pbmzw561yrmje9gi 检测安全组 重新绑定 安全组					
连接信息					
, 连接集群做管理操作需要阿里云颁发的安全证书,每个集群都有独立	的证书,如果您还没有下载过言	当前集群的证书,请点击 🎦			
集群接入点地址:					
<pre>tcp://master2g8.cs-cn-hangzhou.aliyun.com:20053</pre>					
使用指南:					
配置环境变量(linux 或 mac 操作系统):					
export DOCKER_TLS_VERIFY="1"					
export DOCKER_HOST="tcp://master2g8.cs-cn-hangzhou #设置当前路径为你保存集群证书的文件路径	.aliyun.com:20053"				
- COTT - CARGE UTS DIGGENER AND AND AN AND AND					
export DOCKER_CERT_PATH="\$PWD"					
注意事项:					
 1. 证书是安全访问容器集群的凭证,请您务必妥善保管。每个多 2. 如果下數的证书不慎泄露,可以用锎后再里新下數 	長群的证书都是不同的,需要配	置正确的证书才能利用dock	er client或docker compose访问集群	ŧ.	

2. 下载和保存证书。

要通过上面的访问地址访问 Docker 集群,您还需要配置 TLS 证书。

在集群管理页面,单击下载证书开始下载 TLS 证书。下载到的文件为 certFiles.zip 。在下面的例子中,下载的证书存放在 ~/.acs/certs/ClusterName/目录下。其中, ClusterName 是您集群的名字。您也可以使用其他目录,但是为了便于管理,推荐您将文件存 放在 ~/.acs/certs/ClusterName/ 目录下。

```
mkdir ~/.acs/certs/ClusterName/ #替换成真正的集群名字
cd ~/.acs/certs/ClusterName/
cp /path/to/certFiles.zip .
unzip certFiles.zip
```

certFiles.zip 文件包含ca.pem、cert.pem 和 key.pem。

2.4.10 扩容集群

前提条件

一个集群最多可以包含 20 个节点。

背景信息

您可以根据您的业务需求进行集群扩容。

📕 说明:

通过集群扩容添加的节点为按量付费节点。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择需要扩容的集群,单击更多并在下拉菜单中单击集群扩容。

容器服务		集群列表					您最多可以	创建 5 个集	群,每个集群最多可以	添加 40 个节点	刷新	创建Swarm集群	•
Kubernetes	Swarm												
概览		常见问题: S 如何创建集群 S 如何	添加已有云服	各器 🔗 跨可月	区节点管理 🔗 集成日志服务	 /> // //	ocker客户端道	接集群					
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域 (全部) ▼	网络类型	集群状态	节点状态 @	节点个 数	创建时间	Docker版 本			2
集群	1	test-swarm	阿里云集 群	华东1	虚拟专有网络 Vpc-	●运行 中	健康 🕽	1	2018-05-21 10:29:11	17.06.2-ce	管理	章春日志 盛控 升级Agent	<u>删除</u> 更多▼
网络												刊級Docker	
数据卷												升级系统服务	+
配置项											3	集群扩容 添加已有节点	
 ▼ 镜像与方案 镜像 												创建应用	

4. 在弹出的对话框中,设置新节点的规格。

您可以选择增加服务器节点的个数和相应的规格。

5. 单击集群扩容。

2.4.11 搜索集群

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 在搜索框中输入要搜索的集群的名称或关键字。名称中带有该关键字的集群将显示在集群列表中。如下图所示。

P	沿田·
山	- <u></u>
1文示	下区分入小马

容器服务		集群列表							您最多可以创建 5 个集群,	每个集群最多可以添加	D 20 个节点 刷穿	が 创建集	群 -
Swarm 概览	Kubernetes	常见问题: & 如何创建集群 & 如何添加已有	云服务器 🔗 跨可	「用区节点管	管理 🔗 集成!	日志服务 🔗 通	过Docker客户端道	E接集群					
应用		名称 ¥ SW											
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本			操作
集群		swarm-test	阿里云集群	华东1	经典网络	●运行中	健康 ℃	4	2017-11-28 10:04:14	17.06.2-ce	管理	查看日志 监控	删除 更多 -
节点													
网络													

2.4.12 删除集群

背景信息

您可以从容器服务中删除集群。删除集群时,会将关联的云服务器和负载均衡等云资源一起删除。 请谨慎使用该操作。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择要删除的集群并单击删除,如下图所示。

容器服务	集群列表							您最多可以创建 5 个集群,	每个集群最多可以添加 20 个节;	点 刷新	创建集群	¥ -
Swarm Kubernetes	常见问题: & 如何创建集群 & 如何添加已有	云服务器 🔗 跨可	1用区节点	管理 🔗 集成	日志服务 🔗 j	通过Docker客户端	连接集群					
成用	名称 ▼											
服务	集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本		2	攝作
集群 节点	routing-test-online 🖌	阿里云集群	华东1	经典网络	●运行中	健康 С	2	2017-11-28 10:04:14	17.06.2-ce	管理 查讀	香日志 监控	删除 更多▼

4. 在弹出的对话框中,单击确定。

2.4.13 清理集群磁盘

背景信息

清理磁盘操作会清理用户集群内每台服务器上的脏数据。脏数据限于:

- 已下载到本地但未使用的 Docker 镜像。
- 曾经挂载到容器,但容器销毁后未清理的数据卷(volumn)目录。

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择要清理的集群,单击管理,如下图所示。

容器服务 Swarm Kuberneter	集群列表							您最多可以创建5个集群,	每个集群最多可以添加 20 イ	市点 刷新	创建集	群 •
概范	* 常见问题: 8 如何创建集群 8 如何添加已有	伝服务器 🔗 跨	可用区节点	管理 🔗 集成	旧志服务 🔗	通过Docker客户望	制连接集群					
应用	名称 ▼											
服务	集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本	2		操作
集群 1	routing-test-online	阿里云集群	华东1	经典网络	●运行中	健康 ℃	2	2017-11-28 10:04:14	17.06.2-ce	管理	査看日志 监控	删除 更多▼

4. 在集群管理页面中,单击清理磁盘,如下图所示。

集群: routing-test-online				开启日志服务	登录镜像仓	库刷新
基本信息				升级Agent	升级系统服务	清理磁盘
集群ID:	经典网络	●运行中	地域: 华东1	节点个数 2	集群扩容	添加已有节点
安全组ID: sg-bp16febevhwsu2t7l84k 检测安全组 重新绑定 安全组						

2.4.14 登录镜像仓库

前提条件

- 您需要准备一个可用的镜像仓库。本例中使用阿里云容器镜像服务,在上面构建了一个可用的仓库。
- 设置仓库的独立登录密码,本示例中使用阿里云容器镜像服务,在容器镜像服务控制台上设置 与修改 Registry 的登录密码。注意首次修改 Registry 登录密码即是设置密码。

背景信息

您可以在目标集群上登录到镜像仓库,从而提供相关的集群登录信息,方便您使用集群管理工具进 行管理。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择要配置的集群并单击管理。

容器服务		集群列表				t	您最多可以爸	1]建 5 个集群,每个集群最	修可以添加 20 个节点	刷新	创建集群	ŧ -
Swarm	Kubernetes					1	⊨ a ¥					
概览		キメリージョン (19)10月1日の東京社 (19)15000	□月乙服労幣 ♥ 約	り用と口無皆理 ♥ 乗成口添飯另	0 1 <u>813</u> D00	Ker 合广 确注 按另	68 1					
应用		名称 🔻										
服务		集群名称/ID	集群类型 地域	或 网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本			操作
集群		test-swarm 📝	阿里云集群 华东	虚拟专有网络 1 vnc bp1vvct0ccb16vpc2igoc	●运行中	健康 😂	2	2017-12-04 15:26:27	17.06.2-ce	理 查	看日志	制除
节点		LEPHONE CONTRACTOR		vpc-opiivvecscsbiovaczjąoc							监控	₩9*
网络												

4. 在集群的详情页面,单击登录镜像仓库。

集郡	f ; test-swarm				开启日志服务	登录镜像行	②库 刷新
基	本信息				升级Agent	升级系统服务	清理磁盘
集	群ID: edit43disd806c144 Austal904addb2878d2	虚拟专有网络	●运行中	地域: 华东1	节点个数 2	集群扩容	添加已有节点
安安	全组ID: sg-bp12ma406e6eii7v72ty 检测安全组 重新绑定 全组						

5. 在弹出的对话框中,对相关参数进行配置。

登录镜像仓库			×
仓库域名	registry.cn-hangzhou.aliyuncs.com		
而广者: 密码 [*] :	•••••		
邮箱:			
		确定	取消

- 仓库域名:镜像仓库的 hub 域名。以一个镜像地址 registry.cn-hangzhou.aliyuncs.
 com/acs/agent:0.8 为例, registry.cn-hangzhou.aliyuncs.com 即是仓库域名。
- 用户名:您当前登录的阿里云账号全称。
- 密码:Registry 的独立登录密码。Registry 的登录密码是在容器镜像服务的控制台上设置与 修改的。
- 邮箱:可选。
- 6. 单击确定后,若无报错信息,说明已经成功登录到镜像仓库。

2.4.15 升级 Agent

背景信息

📃 说明:

升级期间用户应用不受影响,但是无法通过Web界面对集群进行管理操作,也不能用Docker client连接集群的访问端口,时间大约2分钟。

集群内的每一台服务器都会安装容器服务的 Agent,用于接收容器服务控制系统下放的指令。

容器服务会定期的增加新的功能,如果您需要最新的功能,可以升级集群的 Agent。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择要升级 Agent 的集群,单击更多 > 升级 Agent。

容器服务		集群列表					地	限最多可以创建	5 个集群,每个集群最多可以	以添加 20 个节点 子9	账号授权 易	創新 创建集群	# -
Swarm	Kubernetes												
概览		常见问题: & 如何创建集群 & 如何添加已有	云服务器 🔗 跨	可用区节点	管理 🔗 集成	战日志服务 🔗	通过Docker客户蒙	临连接集群					
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🖉	节点个数	创建时间	Docker版本		2	操作
集群		swarm-test 🖍	阿里云集群	华东1	经典网络	●运行中	健康 ℃	2	2017-11-28 10:04:14	17.06.2-ce	管理	査看日志 监控	删除 更多
节点]	更新访问控制授权(信息
网络											3 3	十级Agent	
数据卷											7	十级Docker	
配置项											7	计级系统服务	
▼ 镜像与7	方案										\$	親群扩容	
镜像											Ř	劾旧有节点	
编排模	板										ť	則建应用	

4. 在弹出的对话框中,单击确定。

2.4.16 升级 Docker Daemon

背景信息

集群内的每一台服务器都会安装标准的 Docker Daemon 用于管理容器。

📕 说明:

- 集群升级需要机器可以公网访问,以便下载升级所需的软件包。
- 集群升级 Docker 过程中,可能会有升级失败的情况,为了您的数据安全,强烈推荐您选择先打 快照然后再升级的方式。
- 集群升级 Docker 过程中,集群上部署的服务会中断,同时无法进行集群和应用的操作,请您在 升级之前安排好相关事宜。升级时间大约3~30分钟,升级完成后集群会变成运行中状态。

您可以在集群列表页面查看集群的 Docker 版本,如下图所示。

集群列表						您最多可	以创建 5 个集群,每个集群	洋最多可以添加 20) 个节点 刷新	创建集群	-
常见问题: Ø 如何创建集群 Ø 如何添加	已有云服务器	》跨 可用图	○日本語書 ◎ 集成日志服务 (§ 通过Docker客	户端连接集群						
名称 ▼											
集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本		100	l / TE
test-swarmmode 🖍 swarm mode	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	3	2017-11-07 15:26:23	17.06.2-ce	管理	查看日志 删版 监控 更多	l≩≁
routing-test-online	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	2	2017-11-01 16:11:29	17.06.2-ce	管理	查看日志 删修 监控 更多	е ₿₹

操作步骤

1. 登录容器服务管理控制台。

- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 选择您要升级 Docker Daemon 的集群,单击 Docker 版本列下的升级 或者单击更多 > 升 级Docker。

概览	
应用	小助手:如何创建集群如何添加已有云服务器跨可用区节点管理集成日志服务通过Docker客户端连接集群
服务	名称 *
集群 1	
市点	
数据卷	test 阿里云集 经典网 ●就緒 健康 C 1 2017-01-18 1.12.5 管理 重看日志 删除 c6f805575f34a440f9964652738527c71 群 络 ●就緒 健康 C 1 2017-01-18 1.12.5 管理 重雪日志 删除 c6f805575f34a440f9964652738527c71 群 站在 通知 近辺 更多 」 二 <
配置项	夏新访问控制授权信息
▼ 镜像与模板	开级Agent
镜像	升級Docker 3

4. 如果目前系统的 Agent 不是最新版本,需要先升级 Agent。单击升级Agent,根据提示进行相应的操作即可,如下图所示。

升级检测		备份快照	\rangle	升级结束
注意: 1. 集群升级需要机器可以公网访问, 2. 集群升级Docker过程中,可能会和 3. 集群升级Docker过程中,集群上記	以便下戰升级所需的软件包 新级失败的情况,为了您的数据安全,强烈 P著的服务会中断,同时无法进行集群和应F	则推荐您选择先打快照然后再升级的方式 用的操作,请您在升级之前安排好相关事	; 宜。升级时间大约3~30分钟 ,升级	吸完成后集群会变成运行中状态
检测项	检测结果			快捷操作
集群Docker版本	❷ 旧版本Docker , 需要升级			
集群Agent版本	❷ 旧版本Agent,需要先升级Agent然后	才能升级集群		升级Agent
集群状态	❷ 旧版本Agent , 需要先升级Agent然后	才能升级集群		
升级检测结果:您的集群: test 需要予	- <mark>级Agent</mark> ,请点击检测列表右侧的 ["] 升级Age	ent"按钮,升级agent完成之后再点击下-	-步	

5. 若 Agent 已是最新版本,则可以直接升级 Docker。

升级检测		备份快照	>	升级结束					
注意: 1. 集群升级需要机器可以公网访问,以便下到 2. 集群升级Docker过程中,可能会有升级失 3. 集群升级Docker过程中,集群上部署的段级	就升级所需的软件包 收的情况,为了您的数据安全,引 各会中断,同时无法进行集群和(强烈推荐您选择先打快照然后再升级的 应用的操作,请您在升级之前安排好将	9方式 目关事宜。升级时间大约3~30	分钟,升级完成后集群会变成运行中状态					
检测项	检测结果			快捷操作					
集群Docker版本	❸ 旧版本Docker,需要升级								
集群Agent版本	❷ 新版本Agent,可以继续升	₩级集群		升级Agent					
集群状态	● 集群状态正常,可以进行升	刊级							
升级检测结果:您的集群: test 处于 可升级状态,请确认是否要给等待升级的云服务器节点做快照备份(强烈推荐您做好快照,详细原因请查阅上方注意事项), 点击下方 "备份快照并升级" 按 到制作快照备份并进行Docker的升级;如果您已经手动给自己的云服务器制作了快照备份,请点击下方 "直接升级"按钮进行升级,采用该方式升级时,中途有失败的节点请自行使用快照恢复									
			备份	快照并升级 直接升级 返回集群列表					

您可以使用以下方法之一升级 Docker:

• 直接升级

单击直接升级,进入升级 Docker Engine 的流程。

• 备份快照并升级

建议您通过备份快照升级 Docker (以便于在升级过程中出问题后,可以通过快照进行恢复)。

单击备份快照并升级,此时系统会调用 ECS OpenAPI 对集群内的节点打快照。

由于备份快照需要一点时间,您需要耐心等待一会。完成快照备份后,系统自动进入升级 Docker Engine 的流程。

如果备份快照失败,继续升级和放弃升级可用。您可以单击继续升级进入升级 Docker Engine 的流程,或者单击放弃升级放弃升级 Docker Engine。

后续操作

此时,返回集群列表页面。您可以看到刚才操作的集群处于Docker-Engine升级中的状态。由于 升级 Docker Engine 会进行相应的容器数据备份等工作,所以比较耗时,请耐心等待一会。

2.4.17 升级系统服务

背景信息

集群的系统服务用来解决应用需要的通用服务,例如日志服务 acslogging,路由服务 acsrouting, volume 服务 acsvolumedriver。下面介绍这些服务的升级操作流程。

■ 说明:

集群的系统服务在升级的过程中会导致您的应用或者服务短暂不可访问或者不能正常工作。请谨慎 升级。建议选择访问低谷或者维护时间进行升级。

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 选择您要升级系统服务的集群,单击更多并在下拉菜单中单击升级系统服务。如下图所示。

容器服务		集群列表					您	最多可以创建 5	个集群,每个集群最多可以	场加 20 个节点	子账号授权	刷新	创建集群	≓ -
Swarm	Kubernetes													
概览		常见问题: ② 如何创建集群 ③ 如何添加已有	云服务器 🔗 跨	可用区节点	「管理 🔇 集日	成日志服务 🖇	通过Docker客户i	端连接集群						
应用		名称 ▼												
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本			2	操作
集群	1	test 🖍	阿里云集群	华东1	经典网络	●运行中	健康 ℃	2	2017-11-28 10:04:14	17.06.2-ce		管理 査	清日志 监控	删除 更多 •
节点												更新访	问控制授权	信息
网络												升级Ag	ent	
数据卷												升级Do	cker	
配置项											3	升级系	统服务	
▼ 镜像与方	窦											集群扩	容	
镜像												添加已	有节点	
(40+10-100-1)	-											创建应	Ħ	

4. 在弹出的对话框中,选择要升级的系统服务并单击升级,如下图所示。

例如,本示例中选择的是路由服务(对应 acsrouting,注意升级会短暂影响用户应用的访问),volume 服务(对应 acsvolumedriver,注意升级可能会短暂影响用户相关联应用的功能)。

F	级系	统服务 - test					\times				
		服务名称	当前版本	已安装	可升级	升级是否影响服务					
	•	简单路由服务	1.1.13	Ø	ø	0					
	¥	Volume服务	1.1.1	Ø	ø	0					
		日志服务	1.0.5	Ø	8	8					
		监控服务	1.0.8	Ø	8	8					
ì	说明:选择升级未安装服务将自动安装该服务的最新版本。升级系统服务可能会对部分 应用造成影响。升级Volume服务会自动重启使用OSS数据卷的容器,请根据具体的业务 情况判断是否进行升级。										
						升级 取消					

此时,单击左侧导航栏中的应用,您会发现系统服务正在升级中,如下图所示。

更新完成后,被影响的服务会恢复正常。

应用列表						刷新创建应用
小助手: 如何创	建应用 变更应用配置	简单路由蓝绿发布	策略 容器単性(申缩		
集群: test 『	□ 隐藏系统应用 □ 隐	藏离线应用 □ 隐藏	被在线应用			名称 ▼
应用名称	描述	状态	容器状态	创建时间 🔺	更新时间 🔺	<u>1</u> 500년 두
acslogging default	Logging Service	● 就绪	就绪:2 停止:0	2017-01-18 12:04:56	2017-02-20 13:37:14	停止 重新部署 事件
acsmonitoring default	Monitoring Service	●就绪	就绪:1 停止:0	2017-01-18 12:04:56	2017-02-20 13:40:16	停止 重新部署 事件
acsrouting default	Routing Service	※ 更新中	就绪:1 停止:0	2017-01-18 12:04:56	2017-01-18 12:05:14	停止 重新部署 事件
acsvolumedriver system	Data Volume Service	柴 更新中	就绪:1 停止:0	2017-01-18 12:04:56	2017-02-20 13:40:16	重新部署 事件

2.5 节点管理

2.5.1 移除节点

背景信息

您可以移除集群中的节点。移除节点可以将机器从集群中摘除。移除后将不能在节点列表内看到该 机器的信息。



- 在移除节点之前请先做好备份工作。
- 移除节点仅仅是从集群中移除 ECS,并不会释放 ECS。如果需要释放 ECS,需要到 ECS 管理 控制台自行释放。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 swarm 菜单下,单击左侧导航栏中的节点。
- 3. 选择要移除的节点所在的集群。
- 4. 选择要移除的节点,单击更多 > 移除节点。

容器服务		节点列表									刷新
Kubernetes	Swarm										
概览		常见问题: ② 按量付费转的	1年包月								
应用		集群: swarm-test ▼	2								
服务		IP地址(ID)	实例类型	实例ID/名称	状态	容器数目	配置	操作系统	Docker版本	Agent	3
集群		0 14 (eip)	阿里云ECS	10.101	●I#	16	CPU : 2核 内存 : 3.859 GB	Ubuntu 16.04.3 LTS	17.06.2-ce	0.10-98812ae	监控 更多 ▼
节点	1									4	里重 D 忌 移除节点
网络											升级Docker

5. 在弹出的确认对话框中,单击确定。

2.5.2 重置节点

背景信息

重置节点会替换该机器系统盘,替换后原机器系统盘数据会丢失,重置后的机器会重新加入到集群中。



• 被重置的 ECS 节点将会恢复到最初加入到集群的状态。

- 节点重置的过程中所有数据将被清理。
- 您在操作前做好相关备份,以免数据丢失给您造成损失。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的节点。
- 3. 选择要重置的节点所在的集群。
- 4. 选择要重置的节点,单击更多 > 重置节点。

容器服务		节点列表										刷新
Kubernetes	Swarm											
概览		常见问题: Ø 按量付费转的 集群: swarm-test ▼	2年包月									
应用			-									
服务		IP地址(ID)	实例类型	实例ID/名称	状态	容器数目	配置	操作系统	Docker版本	Agent		3
集群		0 14 (eip)	阿里云ECS	10.101	●正常	16	CPU:2核 内存:3.859 GB	Ubuntu 16.04.3 LTS	17.06.2-ce	0.10-98812ae	监控	更多•
节点	1									4	移除节点	
网络											升级Docker	

5. 在弹出的确认对话框中,填写该实例的登录密码并单击确定。



2.5.3 查看节点上运行的容器

背景信息

您可以通过节点列表页面查看运行在某个节点上的容器。

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的节点。
- 3. 选择节点所在的集群。

4. 选择所需的节点,单击节点的 ID。

容器服务		节点列表									刷新
Kubernetes Su 概览	warm	常见问题: ♂ 按量付费转 集群: swarm-test ▼	2月5章: 《按量付费转包=6月 : swam-test v 2								
服务		IP地址(ID)	实例类型	实例ID/名称	状态	容器数目	配置	操作系统	Docker版本	Agent	操作
集群		(eip)	阿里云ECS	303802	●正常	7	CPU:2核 内存:3.702 GB	CentOS Linux 7 (Core)	17.06.2-ce	0.10-98812ae	监控 更多▼

您可以看到运行在该节点上的容器的列表。

节点:40.07.1.138							
实例ID:	ASIENCINE		状态:	正常	地域: 华东1		所在集群: swarm-test
名称/ID	Labels: com.docker.swam	m.affinities:["aliyun.ser	vi	満口	容器IP	节点IP	操作
acs-agent () system	ce.id!=acsmonito nt"] aliyun.service.ver	ring_acs-monitoring-ao sion:1.0	je		47.97.8.239	47.87.8.209	监控 日志
acslogging_logsp () default	com.docker.compose.service:acs-monitor ng-agent aliyun.logs: com.docker.compose.container-number:1		ori		172.19.0.4	47.07.0.209	监控 日志 远程终端
acslogging_logta () default	com.docker.compose.container-number:1 com.docker.compose.oneoff:False com.docker.swarm.id:c35a0c9d21e403c3a 256451227c0b2582b01a56dcr0a56dc				47.97.8.239	47.87.8.239	监控 日志 远程终端
acsmonitoring_ac 0 default	c9604f7581d3 aliyun.cluster.id:c 24fba0277c	a5ce0cca9da346d68ff2	2ff		47.97.8.239	47.82.8.239	监控 日志 远程终端
acsrouting_routi default	aliyun.service.id:a oring-agent com.docker.comp	acsmonitoring_acs-mor	rin	0.0.0.0:9080->80/tcp 127.0.0.1:1936->1936/tcp	172.19.0.3	47.07.0.239	监控 日志 远程终端
acsvolumedriver 0 system	g com.docker.comp aliyun.global:true	ose.version:1.5			47.97.8.239	47.87.8.239	监控 日志
test-for-trigger () ad9809ad1a57905a	com.docker.comp e7e02a9f739021c ebff230e0d2f3be:	oose.config-hash:9de7b cd3b3ddc1ad6d879b36 19152125	od 82	0.0.0.0:32769->80/tcp	172.19.0.6	47.87.8.209	删除 停止 监控 日志 远程终端
tunnel-agent 🕖	aliyun.addon:mor running	hitoring sha256:644001fe1			47.97.8.239	47.07.0.209	监控 日志
wordpress_web_3 () 4d0e55982e5d5e63	running	registry.aliyunc sha256:592af506c		0.0.0.0:32770->80/tcp	172.19.0.12	47.57.8.239	删除 停止 监控 日志 远程终端

后续操作

通过该列表,您可以查看容器的标签(Labels),查看容器的镜像并查看镜像的 sha256,查看容器的日志和监控信息,并进行容器相关操作(包括:启停容器、删除容器、通过远程终端操作容器)。

2.6 安全组

2.6.1 容器服务安全组规则

查看安全组规则

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。

3. 选择所需集群并单击右侧的管理。

容器服务		集群列表				您最多可以	创建 5 个集	群,每个集群最多可以	添加 40 个节点	刷新	创建Swarm集群	•
Kubernetes	Swarm											
概览		常见问题: & 如何创建集群 & 如何活	如已有太服务器 🛇 跨	i可用区节点管理 🕹 集成日志服务	⊱ &' 通过D	locker器/P)漏道	接集群					
应用		名称 ▼										
服务		集群名称/ID	地域 (全部 集群 <u>类型</u> ▼	^{B)} 网络类型	集群状态	节点状态 2	节点个 数	创建时间	Docker版 本	2		攝作
集群 节点		test-safe	阿里云集 群 华东1	虚拟专有网络	● 运行 中	健康 오	1	2018-05-21 10:29:11	17.06.2-ce	管理		删除 更多▼

4. 单击安全组的 ID, 跳转到云服务器 ECS 管理控制台上该安全组的详情页面。

	<	集群 : test-safe
基本	4信息	
	-01L/AT	基本信息
贝類	死习惯	集群ID:
用戶	白标签	安全组ID: 检测安全组 重新绑定安等
镜像	刻表	

5. 单击左侧导航栏中的安全组规则。您可以查看安全组的规则。

<	alicloud-cs	-auto-crea	t 🐟 k 8s_vp	c / vpc				赦我设置 € 返回	添加安全组规则 快速创建规则
安全组规则	入方向 出方	向							▲ 导入规则 ▲ 导出全部规则
安全组内实例列表									
安全组内弹性网卡	□ 授权策略	协议类型	端口范围	授权类型	授权对象	描述	优先级	创建时间	操作
	□ 允许	全部 ICMP	-1/-1	地址段访问	0.0.0/0	-	100	2018-05-21 10:29:12	修改描述 克隆 删除
	□ 允许	全部	-1/-1	地址段访问	172.20.0.0/16	-	100	2018-05-21 10:29:12	修改描述 克隆 删除
	□ 允许	自定义 TCP	80/80	地址段访问	0.0.0/0	-	100	2018-05-21 10:29:11	修改描述 売隆 删除
	□ 允许	自定义 TCP	443/443	地址段访问	0.0.0/0	-	100	2018-05-21 10:29:11	修改描述 克隆 删除

安全组规则说明

对于 2017 年 2 月 28 日之后创建的容器服务集群,默认创建的安全组已经做了加固。阿里云容器 服务未设置出方向的安全组规则,只会设置入方向的安全组规则。2018 年 1 月 1 日以后,仅支持 新建 VPC 网络的容器集群,以 VPC 集群为例,开放的规则如下。

VPC 安全组:

<	alicloud-cs	-auto-crea	t a k8s_vp	oc / vpc				較我设置 🗘 返回	添加安全组规则	快速创建规则
安全组规则 安全组内实例列表	入方向 出方	向							▲ 导入规则	▲ 导出全部规则
安全组内弹性网卡	□ 授权策略	协议类型	端口范围	授权类型	授权对象	描述	优先级	创建时间		操作
	□ 允许	全部 ICMP	-1/-1	地址段访问	0.0.0.0/0	-	100	2018-05-21 10:29:12	修改描述	克隆 删除
	□ 允许	全部	-1/-1	地址段访问	172.20.0.0/16	-	100	2018-05-21 10:29:12	修改描述	売隆 删除
	□ 允许	自定义 TCP	80/80	地址段访问	0.0.0.0/0	-	100	2018-05-21 10:29:11	修改描述	売隆 删除
	□ 允许	自定义 TCP	443/443	地址段访问	0.0.0.0/0	-	100	2018-05-21 10:29:11	修改描述	克隆 制除



说明:

- 为方便用户业务的 Web 服务,默认开启 443 端口和 80 端口,可以根据自己的需求选择放开或者关闭。
- ICMP 规则建议保留,用于节点间通信,方便排查问题。有些工具也依赖 ICMP。
- VPC 安全组授权对象设置的是容器网段的基础地址,本示例是 172.20.0.0/16地址段,与您 在创建 VPC 集群时选择的容器初始网段有关,也可以选择其他网段,参见创建集群。保证容器 之间可以互相通信。

对于 2017 年 2 月 28 日之前创建的集群,安全组规则开的比较大。以经典网络安全组规则为例。

sg-	/ alio	10 * 安全组列表			教我设置 € 刷新	返回 添加安全组规则
内网入方向	内网出方向 公网入方向	公网出方向				
经典网络的内障	网入方向规则,推荐优先选择安全组	目授权方式;如选择IP地址方式	说授权,出于安全性的考虑,仅支	z持单IP授权,例如:10.x.y.z/3	2。 教我设置	
授权策略	协议类型	端口范围	授权类型	授权对象	优先级	操作
允许	全部	-1/-1	地址段访问	0.0.0/0 🕗	1	克隆 删除
允许	自定义 TCP	2376/2376	地址段访问	0.0.0.0/0	1	克隆 删除
允许	自定义 TCP	22/22	地址段访问	0.0.0.0/0	1	克隆 删除

sg-bp	/ ali	clo t 安全组列表			教我设置 ♀ 刷	新返回添加	安全组规则
内网入方向	内网出方向 公网入方向	公网出方向					
授权策略	协议类型	端口范围	授权类型	授权对象	优先级		操作
允许	全部	-1/-1	地址段访问	0.0.0/0	1	劳	「隆 删除
允许	自定义 TCP	2376/2376	地址段访问	0.0.0/0	1	厊	「隆 删除
允许	自定义 TCP	22/22	地址段访问	0.0.0/0	1	3	「隆 删除

如果希望收紧规则,可以参考安全组的配置进行如下修改(使用上图中的增加安全组规则和删除)。

- 在内网入方向和公网入方向添加允许 ICMP 规则。
- 如果直接访问 VM 的 80 端口和 443 端口或者其它端口,增加内网和公网规则放开此端口。

送 说明:

务必确保放开所有您需要的端口,否则会导致服务不可访问。通过负载均衡访问的端口不需要 放开。

• 删除地址段0.0.0.0 端口 -1/-1 的公网入规则和内网入规则。

安全配置原则

• 每个集群一个安全组。

容器服务每个集群都管理了一个安全组。您可以在这个安全组上配置规则。

• 最小权限原则。

为了您集群的安全性,安全组应该对外开放最小的权限。

- 容器服务创建的安全组添加了一些默认规则。
 为了方便用户操作 ECS 实例,容器服务创建的安全组添加了一些默认规则,开放了诸如 80/443
 等端口。如果不需要,您可以删除这些规则。
- 尽量使用容器内部网络进行通信,不将通信暴露到宿主机上。
- 授权其它 ECS 实例访问安全组时,授权给安全组,而非单个 IP。
 要授权其它 ECS 实例访问当前安全组,先创建一个新安全组,把要访问当前安全组的 ECS 实例加入新安全组,再授权新安全组访问当前安全组。
- VPC 内网出/入方向里要放开容器的网段。

如果不放开,会导致容器之间网络不通。

2.6.2 检测安全组

背景信息

您可以通过容器服务管理控制台检测集群的安全组规则是否安全。

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择所需的集群并单击右侧的管理。

容器服务		集群列表					您最多可以	J创建 5 个集	輕群,每个集群最多可以;	泰加 10 个节点	刷新	创建Swarm集群	-
Kubernetes	Swarm					0.000	1 ale 1.5151	- 44-WV					
概览		常见问题: & 如何创建集群 & 如何消	加己有云服势	儲 & 跨り用	△17点管理 & 集成日志服务	&' 通过Doo	Ker 各尸 調 连接	铁杆					
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域 (全部) ▼	网络类型	集群状态	节点状态 ❷	节点个 数	创建时间	Docker版 本	2		操作
集群 节点	1	test-safe	阿里云集 群	华东1	虚拟专有网络 vpc-	●运行 中	健康 ℃	1	2018-05-21 10:29:11	17.06.2-ce	管理	查看日志 监控	删除 更多▼

4. 单击检测安全组。系统会检测集群安全组的状态。

│ 集群:test-safe				开启日志服务	登录镜像仓	卑 刷新
基本信息				升级Agent	升级系统服务	清理磁盘
集群ID:co5miccaldo1+6d0470*0+6d0770;	经典网络	●运行中	地域: 华东1	节点个数 4	集群扩容	添加已有节点
安全组ID: sg-bp16febevhwsu2t7l84k 检测安全组 全组						

如果安全组规则正常,系统会显示如下提示框,单击取消即可。

确认修复安全组	\times
安全组规则正常,无需修复。	
	取消

如果您的安全组规则存在安全风险,系统会弹出提示框,列出存在安全风险的安全组规则。单击修复,系统会删除存在安全风险的安全组规则并创建新的安全组规则。

2.6.3 重新绑定安全组

背景信息

您可以通过容器服务管理控制台重新为集群绑定安全组,系统会重新创建一个安全组并将新创建的 安全组绑定到集群,同时将集群中的节点添加到新创建的安全组。

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择所需的集群并单击右侧的管理。

容器服务		集群列表					您最多可以	以创建 5 个约	東群,每个集群最多可以	添加 10 个节点	刷新	创建Swarm集群	-
Kubernetes	Swarm		同语和日本于服务			@ 通过Do	-kore chinicto	☆.dat ₩¥					
概览		HOLE REAL CONTRACTOR ON	1976KUU L/19 ZA116551	ar 🗸 1591/10		0 1012100	CKCI HY - JMLED	90909÷					
应用		名称											
服务		集群名称/ID	集群类型	地域 (全部) ▼	网络类型	集群状态	节点状态 🕜	节点个 数	创建时间	Docker版 本	2		操作
集群	1	test-safe	阿里云集		虚拟专有网络	 运行 			2018-05-21		管理	查看日志	删除
节点			— 群	华东1	vpc-	÷.20	健康 ℃	1	10:29:11	17.06.2-ce		监控	更多▼

4. 单击重新绑定安全组。

集群 : test-safe					开启日志服务	登录镜像仓师	牟 刷新
基本信息					升级Agent	升级系统服务	清理磁盘
集群ID:	经典网络	●运行中	地域: 华东1	节点个数。	ŧ	集群扩容)	泰加已有节点
安全组ID: sg-bp16febevhwsu2t7l84k 检测安全组 重新衡定安全组							

5. 在弹出的对话框中单击确定。

确认重新	绑定安全组	\times
	重新绑定安全组会为集群绑定生成新的安全组,确认重新创建并且绑定安全组?	
	确定取	肖

2.7 镜像与模板管理

2.7.1 查看镜像列表

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的镜像与方案 > 镜像。
 - 常用镜像: 容器服务推荐的一些常用镜像。
 - Docker 官方镜像: Docker Hub 提供的官方镜像。

容器服务		镜像列表 常用镜像 Dock	er官方捐參	
Kubernetes	Swarm			
概览			mongo	
应用		🖣 mongo DB	送型:公开 来源:DOCKER_HUB ★797 MongoDB document databases provide high availability and easy scalability.	创建应用 →
服务				
集群		MySQL.	mysql 类型:公开 来源:DOCKER_HUB ★1243	创建应用 🔶
节点			MySQL is a widely used, open-source relational database management system (RDBMS).	
网络			nginx	
数据卷		NGINX	类型:公开 来源:DOCKER_HUB ★1169 Official build of Nginx.	割建应用 →
配置项			php	
▼ 镜像与方案			For 英型:公开 来源:DOCKER_HUB ★381 While deciment for web development the DHB printing language also provides appared surgery use	创建应用 →
镜像	Ξ		while designed for web development, the rine scripting language also provides general-pulpose dse.	

2.7.3 查看编排模板列表

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的 镜像与方案 > 编排模板。

您可以查看模板分类,或通过模板的名称、描述、镜像信息的关键字搜索目标模板。

- 示例编排:显示容器服务推荐的常用编排模板。
- 我的编排:显示您自己创建的编排模板。

容器服务		编排列表	示例编排 我的编排	搜索 创建
Kubernetes	Swarm			
概览 应用		0 0	gitlab 洋情 11	服务: 調導 postgread: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgread-sameersbn:9.4-24 gtbb: registry.cn-hangzhou.aliyuncs.com/acs-sample/gtbb-sameersbn:latest reds: registry.cn-hangzhou.aliyuncs.com/acs-sample/redis-sameersbn:latest
服务 集群 节点 网络		0 0	jenkins 可选练本号: 2.60.3, 2.32.2, 1 洋樹 VI	服务: 機會 jenkins: registry.cn-hangzhou.aliyuncs.com/acs-sample/jenkins:2.60.3 slave-joag: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-golang slave-joag: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-adva slave-nodej: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-obdys slave-phtp: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-phtp
数据卷 配置项 ▼ 張像与方案	1	0 0	jstorm 部署JStorm及依赖的zookeeper,谢将… 详情 [17]	展等: 領參 x4: mesoscioud/zookeeper:3.4.8-ubuntu x2: mesoscioud/zookeeper:3.4.8-ubuntu x3: mesoscioud/zookeeper:3.4.8-ubuntu nimbus: registry.changzhou.aliyuncs.com/jstorm-docker/jstorm supervisor: registry.cn-hangzhou.aliyuncs.com/jstorm-docker/jstorm
镜像 编排模板] 解决方案	2	Ø00	Redis-cluster-with-Sentinel 洋情请参考云唇社区文章:https://yq.a 洋情	服务: 領缘 master: redis:3 slave: redis:3 sentinel: registry.allyuncs.com/acs-sample/redis-sentinel:3

2.7.4 创建编排模板

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的镜像与方案 > 编排模板。
- 3. 单击右上角的创建。

容器服务		编排列表	表 示例编排 我的编排	投表 创建
Kubernetes	Swarm			3
概览 应用		0 0	gītlab 洋情	服务: 镜像 postgresql: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresql-sameersbn:9,4-24 gttbb: registry.cn-hangzhou.aliyuncs.com/acs-sample/jetlab-sameersbn:latest redis: registry.cn-hangzhou.aliyuncs.com/acs-sample/redis-sameersbn:latest
服务 集群 节点 网络		0 0	jenkins 可远版本号: 2.60.3, 2.32.2, 1 洋情	服装: 環像 jenkms: registry.cn-hangzhou.allyuncs.com/acs-sample/jenkins:2.60.3 slave-golang: registry.allyuncs.com/acs-sample/jenkins-slave-dind-golang slave-poles: registry.allyuncs.com/acs-sample/jenkins-slave-dind-golang slave-poles: registry.allyuncs.com/acs-sample/jenkins-slave-dind-golang slave-pole: registry.allyuncs.com/acs-sample/jenkins-slave-dind-php
数据卷 配置项 ▼ 镜像与方案	1	0 0	jstorm 部署Jstorm及依赖的zookeeper、请将 详情 [1]	総部: 現象 えば: mesoscloud/zookeeper:3.4.8-ubuntu えば: mesoscloud/zookeeper:3.4.8-ubuntu えば: mesoscloud/zookeeper:3.4.8-ubuntu inimbus: registry-ch-hangzhou.aliyumcs.com/jstorm-docker/jstorm supervisor: registry.cn-hangzhou.aliyumcs.com/jstorm-docker/jstorm
镜像 编排模板 解决方案	2	0 0	Redis-cluster-with-Sentinel 详情请参考云振社区文章:https://yq.a 详情 21	服务: 镜像 master: redis:3 slave: redis:3 sentinel: registry.aliyuncs.com/acs-sample/redis-sentinel:3

- 4. 在创建编排页面,填写模板信息。
 - 名称:模板的名称。
 - 描述:模板的相关信息。
 - 内容: Docker Compose 的 yml 文件。有关文件的详细信息,参见 *Compose File* 的详细说明。

创建编排		
*名称:	test 名称为1-64个字符,可包含数字、汉字、英文字符,或"-"	
描述:		
内容:	<pre>1 nnn: 2 expose: 3 - 443/tcp 4 - 80/tcp 5 image: 'nginx:latest' 6 environment: 7 - 'availability:az==2' 8 labels: 9 aliyun.scale: '8' 10 restart: always 11 volumes: 12 - /var/cache/nginx</pre>	包含服务 服务名:nnn 镜像:nginx:latest 编辑删除 新增服务
	利用编排模板您可以定义和部署多容器应用,支持Docker Compose格式。详情请参见 https://docs.docker.com/compose/	
	新增服务 创建编排 取消	

编排模板中包含的服务会显示在页面的右侧。您可以单击编辑在弹出的新建服务 对话框中通过 参数修改编排模板或者单击删除删除所选服务。

此外,您还可以单击新增服务,选择所需的镜像,添加服务到编排模板中。

镜像名称: 支	持填写私有registry	选择镜像		
镜像版本:				
	j	选择镜像版本		
容器数量:	1			
端口映射: 主	机端口	容器端口	publish 协议	操作
主	机端口	容器端口	v	添加
环境变量: 变	2里名称	变量值		操作
名	称	值		添加
数据卷: 主	机路径或数据卷名	容器路径	权限	操作
±	机路径或数据卷名	容器路径	读写 ▼	添加
简单路由配置:容	「「満路」	域名:		操作
容	器端口	如 [<schema>://]<domain-name< td=""><td>>[/<context>]</context></td><td>添加</td></domain-name<></schema>	>[/ <context>]</context>	添加

5. 设置完成后,单击创建编排。

后续操作

您可以在编排列表页面中我的编排下查看所创建的编排模板。

编排列表	示例编排 我的编排		创建
		搜索	
Ø) test) 详情	服务:镜像 nnn:nginx:latest 创建应用 →	

您可以单击详情查看编排模板的详细信息或者单击创建应用使用该编排模板创建应用。

2.7.5 更新编排模板

背景信息

您只能编辑编排列表页面中我的编排下的编排模板。如果您想编辑示例编排,可以先将示例编排另存一下,从而将当前示例模板复制并保存到我的编排中,然后再编辑。

有关如何另存编排模板,参见另存编排模板。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的镜像与方案 > 编排模板。
- 3. 单击我的编排,选择一个模板并单击详情。

容器服务	编排列表示例编排 我的编排 3		搜索创建
Kubernetes Swarm			
概览	#O test	服务: 镜像	
应用	♀₀ 详情 V1	nginx: nginx:latest	
服务	4		
集群			
节点			
网络			
数据卷			
配置项			
▼ 镜像与方案 1			
镜像			
编排模板 2			

4. 单击右上角的编辑。

编排:test	编辑	删除	下载	另存为	创建应用
<pre>1 - nnn: 2 - expose: 3 - 443/tcp 4 - 80/tcp 5 image: 'nginx:latest' 6 - environment: 7 - 'availability:az==2' 8 - labels: 9 aliyun.scale: '8' 10 restart: always 11 - volumes: 12 - /var/cache/nginx</pre>					

5. 编辑模板内容。

您可以直接在模板中进行修改,或者在右侧选择所需的服务,单击编辑在弹出的对话框中通过参数进行修改或者单击删除删除所选的服务。

此外,您还可以单击新增服务,在弹出的对话框中选择所需的镜像,将服务添加到编排模板中。

编排 : test	* 返回编排列表	取消 保存
名称:	test 名称为1-64个字符,可包含数字、汉字、英文字符,或"-"	
描述:		
模板 :	<pre>1 - mnn: 2 - expose: 3 - 443/tcp 4 - 80/tcp 5 image: 'nginx:latest' 6 - environment: 7 - 'availability:az==2' 8 - restart: always 9 - labels: 10 - aliyun.scale: 'l' 11 - aliyun.scale: 'l' 11 - aliyun.scale: 'l' 12 - restart:s: always 13 - volumes: 14 - /var/cache/nginx</pre>	包含服务 服务名:nnn 镜像:nginx:latest 编辑 删除 新增服务

6. 修改完成后,单击右上角的保存保存模板。

2.7.6 下载编排模板

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的镜像与方案 > 编排模板。
- 3. 选择一个模板并单击详情。

容器服务		编排列表	天 示例编排 我的编排		搜索创建
Kubernetes	Swarm				
概览		0 0	gitlab 泽倩 VI	服务: 領導 postgresgl: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresgl-sameersbn:9.4-24 gtitab: registry.cn-hangzhou.aliyuncs.com/acs-sample/gtitab-sameersbn:latest reds: registry.cn-hangzhou.aliyuncs.com/acs-sample/redis-sameersbn:latest	创建应用 →
服务 集群 节点 网络		0 0	3 jenkins 可选版本号:2.60.3, 2.32.2, 1 洋橋 ☑	服务: 镜像 jenkins: registry.cn-hangzhou.aliyuncs.com/acs-sample/jenkins:2.60.3 slave-golang: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-golang slave-java: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-yava slave-node):: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-nodejs slave-phon: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-phon slave-php:: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-php	创建应用 →
数据卷 配置项 ▼ 镜像与方案	1	00 00	jstorm 部署)Storm及依赖的zookeeper, 请将 详情 1/1	服务: 镜像 水1: messocioud/zookeeper:3.4.8-ubuntu 水2: messocioud/zookeeper:3.4.8-ubuntu 水3: messocioud/zookeeper:3.4.8-ubuntu nimbus: registry.cn-hangzhou.aluyuncs.com/storm-docker/jstorm supervisor: registry.cn-hangzhou.aluyuncs.com/jstorm-docker/jstorm	创建应用→
镜像 编排模板 解决方案	2	00 00	Redis-cluster-with-Sentinel 详情语参考云语社区文章:https://yq.a 详情 11	服务: 領急 master: redis:3 slave: redis:3 sentinel: registry,allyuncs.com/acs-sample/redis-sentinel:3	创建应用 →

4. 单击右上角的下载,会立即下载后缀为yml格式的模板文件。

编排 : (tlab t 近回编排列表	[下载	另存为	创建应用
1 · 2	ostgresql: image: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresql-sameersbn:9.4-2	4			A
3 - 4	- DB_USER=gitlab				
5	- DB_PASS=password				
6	- DB_NAME=gitlabhq_production				
8-	- DD_EXTENSION=PB_LTBm				
9	alivun.probe.url: tcp://container:5432				
10 -	volumes:				
11	- /srv/docker/gitlab/postgresql:/var/lib/postgresql				
12 -	itlab:				
13	<pre>image: registry.cn-hangzhou.aliyuncs.com/acs-sample/gitlab-sameersbn:latest linke:</pre>				
14	- redis:redisio				
16	- postgresql:postgresql				
17 -	ports:				
18	- "10080:80"				
19	- "10022:22"				
20 -	environment:				
21	- SMTP_ENABLED=false				
23	- SMTP DOMAIN=www.example.com				
24	- SMTP_HOST=smtp.gmail.com				
25	- SMTP_PORT=587				
26	- SMTP_USER=mailer@example.com				
27	- SMTP_PASS=password				
20	- SMTP_STARTES-CORE				
30	- GITLAB TIMEZONE=Kolkata				
31	- GITLAB_HOST=localhost				
32	- GITLAB_PORT=80				
33	- GITLAB_SSH_PORT=22				
34	- GITLAB_EMAIL=admin@example.com				*

2.7.7 删除编排模板

背景信息

说明	:

您仅能删除编排列表页面中我的编排下的编排模板。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的 镜像与方案 > 编排模板。
- 3. 单击我的编排,选择一个模板并单击详情。

容器服务		编排列表示例	编排 我的编排	3			搜索	创建
Kubernetes	Swarm							
概览		<u>м</u> 0	test		服务: 镜像			
应用		¥0	详情 V:	1	nginx: nginx:latest			
服务			4					
集群								
节点								
网络								
数据卷								
配置项								
▼ 镜像与方案	1							
镜像	-							
编排模板]2							

4. 单击右上角的删除。

编排:test * 返回编排列表	编辑 删除 下载 另存为 创建应用
1 - nnn:	
2 · expose:	
3 - 443/tcp	
4 - 80/tcp	
5 image: 'nginx:latest'	
6 - environment:	
7 - 'availability:az==2'	
8 restart: always	
9- labels:	
10 aliyun.scale: '1'	
11 aliyun.routing.port_80: nginx	
12 restarts: always	
13 - volumes:	
14 - /var/cache/nginx	

5. 在弹出的确认对话框中,单击确定。

2.7.8 另存编排模板

您可以将示例编排模板或者您自己的编排模板另存为一个新的编排模板。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的镜像与方案 > 编排模板。
- 3. 选择一个模板并单击详情。

容器服务	编排列表示的编排 我的编排	搜索 的	建
Kubernetes Swarm 概览 应用	© o gitlab i≭∰ 21	服务: 镜像 postgresql: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresql-sameersbn:9.4-24 gitab: registry.cn-hangzhou.aliyuncs.com/acs-sample/gitab-sameersbn:latest redis: registry.cn-hangzhou.aliyuncs.com/acs-sample/redis-sameersbn:latest	
服务 集群 节点 网络	3 jenkins 可思様本号: 2.60.3, 2.32.2, 1 洋橋 12	服务: 镜像 jenkins: registry.cn-hangzhou.aliyuncs.com/acs-sample/jenkins:2.60.3 slave-golang: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-golang slave-poixe: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-hodejs slave-poixe: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-hodejs slave-poixe: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-python slave-php: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-php	
数据卷 配置项 ▼ 镜像与方案 1	jstorm 部署JStorm及依赖的zookeeper,请将 详情 21	服务: 镜像 241: mesoscloud/zookeeper:3.4.8-ubuntu 242: mesoscloud/zookeeper:3.4.8-ubuntu 243: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 244: mesoscloud/zookeeper:3.4.8-ubuntu 245: mesoscloud/zookeeper:3.4.8-ubun	
镜像 编排模板 2 解决方案	Redis-cluster-with-Sentinel 洋情语参考云価比区文章 : https://yq.a 洋情 11	服务: 領象 master: redis:3 slave: redis:3 sentinel: registry.allyuncs.com/acs-sample/redis-sentinel:3	

4. 单击右上角的另存为。

│编排:gitlab t 返回编排列表	下载 另存为	创建应用
<pre> ###:gttab * 送回编排列表 1 * jostgresql: 2 image: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresql-sameersbn:9.4-24 3 * environment: 4 * 0 BUSERegitlab 5 * 0B_PAMS=password 6 * 0B_WME=gitlabhg_production 7 * 0B_EXTENSION=pg_trgm 8 * labels: 9 * aliyun.probe.url: tcp://container:5432 10 * volumes: 11 * /srv/docker/gitlab/postgresql:/var/lib/postgresql 12 * gitlab: 13 * image: registry.cn-hangzhou.aliyuncs.com/acs-sample/gitlab-sameersbn:latest 14 * links: 15 * redis:redis:o 16 * postgresql:postgresql 17 * ports: 18 * * ilobels: 19 * * * ilobels: 10 * volumes: 11 * inks: 12 * index:* ilobels: 13 * image: registry.cn-hangzhou.aliyuncs.com/acs-sample/gitlab-sameersbn:latest 14 * links: 13 * inage: registry.cn-hangzhou.aliyuncs.com/acs-sample/gitlab-sameersbn:latest 14 * links: 15 * redis:redis:0 16 * postgresql:postgresql 17 * ports: 21 * Tz-Asia/Shanghai 22 * SMTP_PMOT:sample.com 23 * SMTP_PMOT:sample.com 24 * SMTP_PMOT:sample.gample.com 25 * SMTP_PMOT:sample.gample.com 26 * SMTP_USER=mail.regexample.com 27 * SMTP_USER=mail.regexample.com 27 * SMTP_USER=mail.regexample.com 27 * SMTP_USER=mail.regexample.com 27 * SMTP_PMOT:sample.gample.com 27 * SMTP_PMOT:sample.com 27 * SMTP_PMOT:sample.gample.com 27 * SMTP_PMOT:sample.com 27 * SMTP_PMOT:sample.com 27</pre>	下載 另存为	创建应用
28 - SMTP_STARTILS=Frue 29 - SMTP_AUTHENTLGATON=login 30 - GITLAB_INEZONEKOlkata 31 - GITLAB_HOST=localhost 32 - GITLAB_PORT=80 33 - GITLAB_PORT=80 34 - GITLAB_SHP_PORT=22 34 - GITLAB_MIL=admingexample.com		

5. 填写新的编排模板名称并单击确定。

模板另存为		\times
另存为名称:	gitlab-for-test	
	确定	取消

另存的编排模板会显示在编排列表页面中我的编排下。

编排列表	示例编排 我的编排		创建
	担	· 秦	
0 0	gitlab-for-test 洋情	服务: 铜像 postgresql: registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresql-sameersbn:9.4-24 gittab: registry.cn-hangzhou.aliyuncs.com/acs-sample/gittab-sameersbn:latest redis: registry.cn-hangzhou.aliyuncs.com/acs-sample/redis-sameersbn:latest	创建应用 →
00 00	self 洋情	服务:镜像 postgresql:registry.cn-hangzhou.aliyuncs.com/acs-sample/postgresql-sameersbn:9.4-24 gitlab:registry.cn-hangzhou.aliyuncs.com/acs-sample/gitlab-sameersbn:latest redis:registry.cn-hangzhou.aliyuncs.com/acs-sample/redis-sameersbn:latest	创建应用 →
00 00	test 洋情	服务: 額像 nnn : nginx:latest	创建应用 →

2.8 服务编排

2.8.1 简介

容器服务支持 Docker Compose 编排模板来描述多容器应用。

编排模板允许您描述一个完整的应用,该应用可以由许多个服务组成。例如:一个门户网站应用,由一个 Nginx 服务、一个 Web 服务和一个数据库服务组成。

一个服务可能会有多个容器实例,所有容器实例的配置保持一致。例如:上述应用中的 Web 服务,就可以根据访问量需要启动两个甚至更多的容器。

能力

容器服务支持通过编排模板文件,自动化地部署和管理一个应用。

编排模板文件使用的标签兼容大部分 Docker Compose V1 和 V2 版本实现的标签。有关具体兼容的标签,参见标签概览。

编排模板文件也支持 Compose V1 和 V2 两种版本的模板格式。更多详细信息,参见 Docker Compose V1 和 Docker Compose V2。

容器服务也在社区版本之上提供了很多扩展能力:

- 与社区的 Docker Compose 和 Swarm 不同,阿里云容器服务支持跨节点的容器连接(link),所以您可以直接将 Docker Compose 模板描述的应用部署到分布式集群上来提供高可用性和可伸缩性。
- 容器服务也在社区 Compose 模板描述的基础上提供了一系列扩展来简化 Web、微服务应用的 部署和运维。更多详细信息,参见标签概览。

示例

下面是一个 WordPress 应用,包含了由 WordPress 镜像提供的 Web 服务和 MySQL 镜像提供的 db 服务。

```
web:
  image: wordpress:4.2
 ports:
    - "80"
  environment:
    - WORDPRESS_AUTH_KEY=changeme
    - WORDPRESS_SECURE_AUTH_KEY=changeme
    - WORDPRESS_LOGGED_IN_KEY=changeme

    WORDPRESS_NONCE_KEY=changeme

    - WORDPRESS_AUTH_SALT=changeme
    - WORDPRESS_SECURE_AUTH_SALT=changeme
    - WORDPRESS_LOGGED_IN_SALT=changeme
    - WORDPRESS_NONCE_SALT=changeme
 restart: always
  links:
    - db:mysql
  labels:
    aliyun.log_store_wordpress: stdout
    aliyun.probe.url: http://container/license.txt
    aliyun.probe.initial_delay_seconds: "10"
```

```
aliyun.routing.port_80: wordpress;http://www.example.com;https://
www.nice.com
    aliyun.scale: "3"
db:
    image: mysql:5.6
    environment:
        MYSQL_ROOT_PASSWORD: password
    restart: always
    labels:
        aliyun.log_store_mysql: stdout
```

2.8.2 标签概览

容器服务编排模板文件使用的标签兼容大部分 Docker Compose V1 和 V2 版本实现的标签,并在 社区版本的基础上提供了很多扩展能力。

扩展能力的标签

容器服务扩展了编排模板的部署和生命周期管理能力,所有扩展能力都被描述在 labels 标签下面,作为子标签使用。

标签	说明
probe	设置服务的健康性检查。
rolling_updates	设置服务滚动更新。
parallelism	设置 rolling_updates 每次并行更新的容 器数量。注意:此标签必须和 rolling_up dates 配合使用,单独使用无效。
depends	设置服务的依赖关系。
scale	设置该服务的容器数量,横向扩展服务。
routing	设置该服务的访问域名。
routing.session_sticky	设置 routing 在做请求路由的时候,是否保持 session sticky,即会话保持。注意:此标签必 须和 routing 配合使用,单独使用无效。
lb	通过自定义阿里云负载均衡 nat 映射的方式来暴露服务端口到公网或者内网。
日志	和阿里云日志服务集成,采集容器日志并且发送 到阿里云日志服务。
global	设置该服务为全局服务。

功能增强的标签

容器服务提供服务部署约束#affinity:service#标签用来设置该服务的部署约束条件。

额外支持的标签

标签	说明
external	设置该服务直接链接到外部地址。
dns_options	设置 DNS 选项,和docker run 命令中的 dns-opt参数语义一致。
oom_kill_disable	设置是否禁止 OOM Killer, 和docker run 命令 中的oom-kill-disable 参数语义一致。

变量替换

容器服务支持参数化的 Docker Compose 模板。模板中可以包含环境变量作为参数,当模板部署时 会提示输入参数值,并在部署时对模板进行变量替换。

更多详细信息,参见变量替换。

容器重新调度

容器服务支持对 Docker 容器的重新调度:当一个节点失效时,容器可以被自动调度到其他可用节 点自动运行。

更多详细信息,参见容器重新调度。

高可用性调度

为了使应用有更高的可用性,容器服务支持将同一个服务的容器调度在不同的可用区(availability zone)里。当某个可用区故障时,应用依然能够提供服务。

更多详细信息,参见高可用性调度。

不支持的 Docker Compose 标签

容器服务暂不支持 Docker Compose 的部分标签。有关容器服务暂不支持的标签,参见不支持的 *Docker Compose* 标签。

2.8.4 probe

设置服务的健康性检查。

• 通过 URL 进行检查, 支持 HTTP 协议、TCP 协议。

• 通过 shell 脚本检查。

健康检查会从容器宿主机上发起,每隔一定时间(默认两秒)向容器发起请求或在容器上执行 shell 脚本命令。

检查成功的判断条件为:HTTP 请求的返回码为 2XX/3XX;TCP 端口可建立连接;shell 脚本运行 返回值为 0。

检查的字段解释:

- aliyun.probe.url:HTTP、TCP 请求的 URL。请注意您不需要填写自己的域名或者 IP 地址,只需要加上 container 这个单词,该 URL 最终会被解析成容器相应的 IP 去进行健康检查,检查结果返回 2XX 或者 3XX 才认为服务是健康的。
 - 例如,容器通过 8080 端口提供 HTTP 服务,并提供了 /ping 作为健康检查的 URL,则探测 URL 的格式为 http://container:8080/ping,容器服务会自动通过 HTTP GET 请求检查 URL 的返回结果,如果返回结果的返回码为 2XX 或 3XX,则说明健康检查成功。
 - 例如, MySQL 容器侦听 3306 端口, 探测 URL 的格式为 tcp://container:3306, 服务
 会检查容器 3306 端口是否打开, 如果打开则说明健康检查成功。
- aliyun.probe.cmd:健康检查执行的检查 Shell 命令,/check.sh;容器服务会定期在容器
 内执行该命令,当 shell 脚本返回值为 0 时表明健康检查成功。
- aliyun.probe.timeout_seconds:健康检查的超时时间。
- aliyun.probe.initial_delay_seconds:在容器启动后延迟几秒开始健康检查。

■ 说明:

- 一个服务中只能包含 aliyun.probe.url 和 aliyun.probe.cmd 其中之一。
- 如果服务不包含 aliyun.probe.url 或 aliyun.probe.cmd,则容器缺省为健康状态,且 其他 aliyun.probe.xxx标签会被忽略。

示例:

利用 URL 检测容器健康状态。

```
os:
  image: my_nginx
  labels:
    aliyun.probe.url: http://container/ping
    aliyun.probe.timeout_seconds: "10"
```

```
aliyun.probe.initial_delay_seconds: "3"
```

利用 shell 脚本检测容器健康状态。

```
os:
    image: my_app
    labels:
        aliyun.probe.cmd: health_check.sh
        aliyun.probe.initial_delay_seconds: "3"
```

2.8.5 rolling_updates

更新某个服务时,如果该服务包括超过一个以上容器(使用 scale 标签定义),在第 n 个容器更新成功后,再去做第 n+1 个容器的更新,以此来最小化停止服务时间。

示例:

部署 WordPress 服务,通过scale标签指定部署 2 个容器,使用rolling_updates标签可以使 WordPress 对外停止服务的时间最小化。

```
web:
 image: wordpress
 ports:
    - 80
 restart: always
  links:
    - 'db:mysql'
 labels:
   aliyun.logs: /var/log
    aliyun.routing.port_80: http://wordpress
    aliyun.rolling_updates: 'true'
    aliyun.scale: '2'
db:
  image: mariadb
 environment:
   MYSQL_ROOT_PASSWORD: example
 restart: always
  labels:
    aliyun.logs: /var/log/mysql
```

parallelism

您可以使用 parallelism 标签定义 rolling_updates 每次并行更新的容器数量。

说明:

此标签必须和 rolling_update 配合使用,单独使用无效。

取值:

• 默认值为1,即每次只更新一个容器。
- 当其值大于 1 的时候, rolling_updates 过程中,每次会以 parallelism 定义的值来并行 更新相应个数的容器,实现批量更新。
- 当定义值无效时,默认为1。



为了确保始终有容器在提供服务,建议 parallelism 定义的值小于服务包含的容器数。

示例:

下面的示例部署 Nginx 服务,通过 scale 标签部署 4 个容器,使用 rolling_updates 和 parallelism 标签定义每次以 2 个容器为单位来进行批量更新。

2.8.6 depends

设置服务的依赖关系。

设置之后,容器服务可以控制容器的启动顺序,一个接一个的启动容器。

示例:

```
送明:送明:多个依赖使用逗号(,)分隔。
```

```
web:
    image: wordpress:4.2
    ports:
        - 80
    links:
        - db:mysql
    labels:
        aliyun.depends: db,redis
db:
    image: mysql
    environment:
        - MYSQL_ROOT_PASSWORD=password
redis:
```

image: redis

2.8.7 scale

设置该服务的容器数量,横向扩展服务。

目前,Docker Compose 只能在每一个服务中启动一个容器,如果需要扩展容器数量,需要在启动 后手动进行设置。

现在通过scale的扩展标签,支持您在容器启动的时候进行扩展。

此外,在容器被删除之后,您可以在容器服务管理控制台对应用进行重新部署(单击左侧导航栏中的应用,选择目标应用并单击右侧的重新部署),容器服务会重启或新建容器使容器恢复到指定数量。

示例:

```
web:
    image: wordpress:4.2
    ports:
        - 80
    links:
        - db:mysql
    labels:
        aliyun.scale: "3"
db:
    image: mysql
    environment:
        - MYSQL_ROOT_PASSWORD=password
```

2.8.8 routing

设置该服务的访问域名。

格式:

```
aliyun.routing.port_$container_port: [http://]$domain|$domain_prefix[:
$context_path]
```

名词解释:

- \$container_port: 容器端口,注意该处不是主机的端口。
- \$domain: 域名, 需要用户填写自己的域名。
- \$domain_prefix: 域名前缀,如果填写域名前缀,容器服务会提供给您一个测试用的域名,域 名后缀是.<cluster_id>.<region_id>.alicontainer.com。
- \$context_path: 请求的服务路径,即可以根据请求的路径来选择区分不同的服务。

绑定域名的选择:

- 如果使用 HTTP 协议暴露服务,可以使用容器服务提供内部域名(顶级域为 alicontainer. com),供您测试使用,也可以使用您提供的域名。
- 如果使用 HTTPS 协议,那么仅支持配置您提供的域名,例如 www.example.com。您需要修改 DNS 设置将域名指定到容器集群提供的负载均衡服务上。

标签声明的格式要求:

- 容器服务为每一个集群分配了子域名,绑定内部域名只需要给出域名的前缀,域名前缀仅表示域
 名的一级,不能使用点号(.)进行分隔。
- 如果您不指定 scheme,则默认使用 HTTP 协议。
- 域名的长度不能超过 128 个字符, context root 的长度不能超过 128 个字符。
- 绑定多个域名到服务时,域名之间用分号(;)隔开。
- 一个后端服务可以有多个端口,该端口指的是容器暴露的端口,一个端口只能使用一条 label 进 行声明,带有多个端口的服务需要声明多个 label。

示例:

使用 routing 标签。

```
将容器服务提供的内部域名 wordpress.<cluster_id>.<region_id>.alicontainer.com 
绑定到 Web 服务的 80 端口,并且将您提供的自有域名 http://wp.sample.com/context 绑
定到 Web 服务的 80 端口。
```

```
web:
    image: wordpress:4.2
    links:
        - db:mysql
    labels:
        aliyun.routing.port_80: wordpress;http://wp.sample.com/context
db:
    image: mysql
    environment:
        - MYSQL_ROOT_PASSWORD=password
```

最终您得到的内部域名为 wordpress.cd3dfe269056e4543acbec5e19b01c074.cnbeijing.alicontainer.com。

Web 服务运行之后,您可以通过 http://wordpress.cd3dfe269056e4543acbec5e19b01c 074.cn-beijing.alicontainer.com 或者 http://wp.sample.com/context 访问相应的 Web 服务。 如果您需要支持 HTTPS 服务,需要自行通过阿里云官网负载均衡管理控制台上传 HTTPS 证书,并绑定相应的集群对外访问负载均衡端点。

routing.session_sticky

设置 routing 在做请求路由的时候,是否保持 session sticky,即会话保持。其效果是,在某个会话时间内,请求一直路由到同一个后端的容器,而不是每次请求都随机路由到不同的容器。

只有当您已经设置了 aliyun.routing.port_\$container_port 时,该设置才能起作用。

其设置方法如下:

• 开启会话保持

aliyun.routing.session_sticky: true

• 关闭会话保持

aliyun.routing.session_sticky: false

模板编排文件示例:

```
web:
    image: wordpress:4.2
    links:
        - db:mysql
    labels:
        aliyun.routing.port_80: wordpress;http://wp.sample.com/context
        aliyun.routing.session_sticky: true
db:
    image: mysql
    environment:
        - MYSQL_ROOT_PASSWORD=password
```

2.8.9 lb

通过自定义阿里云负载均衡 nat 映射的方式来暴露服务端口到公网或者到内网。需要升级到最新版

本的 Agent 方能支持该扩展能力标签。

标签格式如下,带\$的变量为占位符。

```
aliyun.lb.port_$container_port:$scheme://$[slb_name|slb_id]:$
slb_front_port
```

示例

web: image: wordpress:4.2

```
ports:
    - 7777:80
    - 9999:9999
    - 8080:8080
    - 53:53/udp
  links:
    - db:mysql
  labels:
    aliyun.lb.port_80: http://slb_example_name:8080
    aliyun.lb.port_9999: tcp://slb_example_name:9999
    aliyun.lb.port_8080: https://14a7ba06d3b-cn-hangzhou-dg-a01:80
    aliyun.lb.port_53: udp://14a7ba06d3b-cn-hangzhou-dg-a01:53
db:
  image: mysql
  environment:
    - MYSQL_ROOT_PASSWORD=password
```

要使用好自定义负载均衡的 1b 标签,您需要理解请求路由过程中的 3 个端口,即负载均衡的前端端口,负载均衡的后端端口(也就是 ECS vm 的端口),最后就是容器的端口。以第一个1b标签 aliyun.1b.port_80为例,从左往右看,在 key 中的 80 端口指的是容器要暴露的端口,后面的 8080 端口指的是负载均衡要暴露的前端端口。负载均衡的后端端口是 ECS 实例的端口,可从标签 ports 的主机:容器端口映射中获取,由此,您可以查到容器端口 80 对应的主机端口是 7777,因此确定了负载均衡转发的后端端口是 7777 端口。因此第一个标签说明了当向服务 Web 发起请求时,首先通过负载均衡前端的 8080 端口进入,转发到后端 ECS 实例的 7777 端口,然后再根据端口映射 ports 的声明,请求最终从容器端口 80 进入,交由容器内的 WordPress 进程提供服务。接下来的标签以此进行相同的解释。该标签配置的负载均衡均不经过集群内置的 routing 服务,请求的路由由您自己控制。

标签声明的格式要求

- 指明负载均衡实例时,可以使用负载均衡实例的名称或者负载均衡实例的 ID。
- 负载均衡实例名称的限制为 1~80 个字符,允许包含字母、数字、连字符(-)、正斜杠(/)、 点号(.)、下划线(_)。
- 容器端口限制为 1~65535。
- 负载均衡前端端口的限制为 1~65535。

带有自定义负载均衡 nat 映射的服务部署限制

您需要自己创建负载均衡实例,对负载均衡实例命名,并创建对应监听端口,然后以扩展标签的方式提供映射的容器端口 \$container_port,使用的协议 \$scheme(可能的值有 tcp、http、https、udp,负载均衡实例的名称 \$slb_name 或者 \$slb_id,以及指定负载均衡实例的前端端口 \$slb_front_port。

- 您必须指定服务要暴露端口的主机和容器端口的映射,通过 Dockerfile 标准的标签 ports 指定,注意必须指定主机端口,且与其他服务映射的主机端口不能冲突,需要主机的端口用于负载均衡绑定后端的 ECS 实例。
- 一个服务只能使用一个或者多个负载均衡实例进行服务端口的暴露,因多个服务会分布在不同的
 ECS 实例后端,多个服务不能共享使用同一个负载均衡实例。
- 通过 lb 标签来配置使用负载均衡路由时,不能配置为集群默认的负载均衡实例。
- 部署了带有负载均衡 nat 映射的服务的主机使用相同的主机:容器端口映射,因此这些服务在每 台 ECS 上只有一个实例。
- 支持的负载均衡协议 \$scheme 包括 tcp、http、https、udp 协议。
- 您需要自行在阿里云负载均衡管理控制台创建监听的端口。
- 请自行登录负载均衡管理控制台对在容器服务中使用的负载均衡实例进行具体的配置修改,例如
 带宽限制等配置。
- 1b标签的价值在于您不需要自行绑定负载均衡后端的ECS实例,只需要配置好相应的标签,就会自动帮助您完成绑定后端的操作。因此,除了绑定负载均衡后端的操作,您对负载均衡的设置和修改需要自行在阿里云负载均衡管理控制台上完成。
- 容器服务会帮助您生成一个 RAM 子账户(需要您开通 RAM),使用这个具有部分负载均衡权
 限(没有创建和删除负载均衡的权限)的账号帮助您管理在容器服务中使用的负载均衡实例,例
 如绑定集群中某些节点作为服务的后端。
- 在服务的整个生命周期内,1b标签会一直生效,除非服务被删除,或者1b标签删除之后重新部署了服务,在此期间,配置在1b标签内的 SLB 实例不能混用。

2.8.10 日志

和阿里云日志服务集成,采集容器日志并且发送到阿里云日志服务。

示例

```
mysql:
    image: mysql
    ports:
        - 80
    labels:
        aliyun.scale: "1"
    environment:
        - MYSQL_ROOT_PASSWORD=password
wordpress:
    image: registry.aliyuncs.com/jiangjizhong/wordpress
    ports:
        - 80
    labels:
        aliyun.routing.port_80: wordpress-with-log
```

```
aliyun.log_store_dbstdout: stdout #注意这里
links:
 - mysql
```

更多详细信息,参见集成日志服务。

2.8.11 global

设置该服务为全局服务。

有一些服务需要在每一个节点部署,例如监控或是日志类的服务。并且在新的节点建立的时候就对这个节点进行服务的部署。

当一个服务被设置为 global 时,该服务会在集群中的每一个节点进行部署。当集群中有新增节点时,也会自动部署一个容器实例到新节点之上。

```
monitor:
    image: sample
    labels:
        aliyun.global: true
```

2.8.12 服务部署约束 (affinity:service)

设置服务的部署约束条件。

容器服务支持 Docker Swarm 兼容的容器部署约束条件,您可以通过 Docker Swarm Filter 控制一个容器的部署。

但是在社区版 Docker Compose 中,却并没有相关的能力来控制服务直接的部署约束。

```
在容器服务中,您可以在 environment 中添加相关 affinity:service,来约束服务之间的亲 和度(Affinity),达到控制服务部署策略的功能。支持服务之间的 Soft affinity和 Hard affinity。
```

示例:

本示例中,web 服务设置了 affinity:service!=db 的部署约束。使得 web 服务一定会选择没 有部署 db 服务的节点,这样当一个节点失效时,可提高服务可用性。当您的集群只有一个节点的 时候,由于指定的是 hard anti-affinity,该部署会失败,因为部署没有办法满足所指定的强约束条 件。

```
web:
    image: registry.aliyuncs.com/acs-sample/wordpress:4.5
    ports:
        - '80'
    environment:
```

```
- affinity:service!=db
 restart: always
 links:
   - 'db:mysql'
  labels:
    aliyun.logs: /var/log
    aliyun.probe.url: http://container/license.txt
    aliyun.probe.initial_delay_seconds: '10'
    aliyun.routing.port_80: http://wordpress
    aliyun.scale: '2'
db:
  image: registry.aliyuncs.com/acs-sample/mysql:5.7
 environment:
   MYSQL_ROOT_PASSWORD: password
 restart: always
 labels:
    aliyun.logs: /var/log/mysql
```

2.8.13 external

设置该服务直接链接到外部地址。

扩展字段下有以下字段可以使用:

- host:设置链接的域名。
- ports:设置链接的端口。

示例:

不使用 external,直接启动一个 MySQL 容器。

```
web:
    image: wordpress:4.2
    ports:
        - 80
    links:
        - db:mysql
db:
    image: 10.32.161.160:5000/mysql
    environment:
        - MYSQL_ROOT_PASSWORD=password
```

通过 external, 描述一个并没有部署在集群中的 RDS 服务,并提供给部署在集群中的

WordPress 使用。

```
wordpress:
image: wordpress:4.2
ports:
    - 80
links:
    - db:mysql
environment:
    - WORDPRESS_DB_USER=cloud
    - WORDPRESS_DB_PASSWORD=MYPASSWORD
    - WORDPRESS_DB_NAME=wordpress
```

```
db:
    external:
    host: rdsxxxx.mysql.rds.aliyuncs.com
    ports:
        - 3306
```

2.8.14 dns_options

设置 DNS 选项,和docker run 命令中的 --dns-opt 参数语义一致。

```
wordpress:
  image: wordpress:4.2
  dns_options:
    - "use-vc"
```

2.8.15 oom_kill_disable

设置是否禁止 OOM Killer, 和docker run 命令中的--oom-kill-disable参数语义一致。

```
wordpress:
    image: wordpress:4.2
    oom-kill-disable: true
```

2.8.16 变量替换

容器服务支持参数化的 Docker Compose 模板。模板中可以包含环境变量作为参数,当模板部署时 会提示输入参数值,并在部署时对模板进行变量替换。

比如,您可以定义参数 POSTGRES_VERSION。

```
db:
    image: "postgres:${POSTGRES_VERSION}"
```

当部署上面的 Compose 模板的时候,容器服务会提示您输入POSTGRES_VERSION参数值,比如 9.3。容器服务会根据参数值对 Compose 模板进行变量替换。在本示例中,会部署一个 postgres :9.3 的容器。

容器服务完全兼容 Docker Compose 的语法,可以在模板中使用\$VARIABLE或者\${VARIABLE}格式的语法。

在 Compose 模板中可以使用\$\$来对需要包含\$的字符串进行转义,这样容器服务不会错误地将其 作为参数来进行处理。

关于 Compose 模板支持变量替换的详细信息,参见 Variable substitution。

2.8.17 容器重新调度

容器服务支持对 Docker 容器的重新调度:当一个节点失效时,容器可以被自动调度到其他可用节 点自动运行。

缺省情况下,容器的重新调度策略是关闭的。根据需要,您可以用如下配置来让重调度策略生效。

容器服务提供兼容 Docker Swarm 的容器重新调度策略,可以通过环境变量方式或者 label 方式启动。

环境变量:

```
redis:
    image: redis
    environment:
        - reschedule:on-node-failure
```

Label :

```
web:
    image: nginx
    restart: always
    environment:
        - aaaaa=aaaaa
    labels:
        aliyun.scale: "3"
        com.docker.swarm.reschedule-policies: "[\"on-node-failure\"]"
```

送明:

如果重新调度容器之后,需要恢复 Docker 容器所需的持久化状态,需要配合支持数据迁移或共享的 Docker 文件卷。

2.8.18 高可用性调度

为了使应用有更高的可用性,容器服务支持将同一个服务的容器调度在不同的可用区(zone)里。 当某个可用区故障时,应用依然能够提供服务。

您可以在编排文件中通过环境变量指定对可用区的选择,有以下两种格式。

availability:az==3

服务至少分布在3个可用区中;如果当前集群没有3个可用区,或机器资源不够导致无法分布 在3个可用区,容器创建会失败。

availability:az==~3

服务尽可能分布在3个可用区中;无法满足时依然可以成功创建。

在下面的示例中,服务至少要部署在两个可用区中。

```
nnn:
    expose:
        - 443/tcp
        - 80/tcp
    image: 'nginx:latest'
    environment:
        - 'availability:az==2'
    labels:
        aliyun.scale: '8'
    restart: always
    volumes:
        - /var/cache/nginx
```

2.8.19 不支持的 Docker Compose 标签

标签	说明
build	build标签用于使用当前目录中的 Dockerfile 文件和其他文档进行容器镜像构建。目前容器服 务暂不提供构建镜像功能,推荐您将构建和部署 的动作分开处理:您可以利用阿里云的容器镜像 服务直接从代码源构建镜像,或者将本地构建的 镜像推送到镜像仓库;您可以在编排模板中使用 image标签引用镜像仓库(包括私有仓库)中 的镜像。
dockerfile	同 build 标签。
env_file	容器服务暂不支持以文件方式指定环境变量,您可以通过 environment 标签添加环境变量。
mac_address	暂时不支持 Mac 地址的设置。
detach	容器服务的所有镜像都是以 detach 模式启动的,不允许您指定 attach 方式执行。
stdin_open	同 detach 标签。
tty	同 detach 标签。
extends	不支持。
networks	Compose version 2 中的网络允许服务的容器启 动在自定义的网络中,容器服务的容器都是在同 一个跨主机互通的容器网络,所以不支持您在 Compose version 2 中使用 networks 标签。

标签	说明
	关于容器服务的网络管理和服务发现,参见 跨 主机互联的容器网络。

2.9 应用管理

2.9.1 创建应用

背景信息

使用限制

swarm 集群只支持 compose V1 和 compose V2 版本的编排模板;因此,在使用编排模板创建应用时,请选择 compose V1 或 compose V2 版本的模板。如果您选择了 compose V3 版本的模板,系统会报错。



说明:

在编排模板列表页面, compose V3 版本的模板标注有 composev3 字样。

错误提示		\times
	您选择部署集群不支持v3版本模板。	
	确定取消	í

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航中的应用。
- 3. 选择目标集群,单击右上角的创建应用。如下图所示。

容器服务		应用列表								周新	创建应用
Kubernetes 概符	Swarm	常见问题: 823	建应用 🔗	变更应用配置 🔗 简单路由	3蓝绿发布策	略 🖉 容器弹性伸缩				_	3
应用	1	集群: test-swarm	• ₹ 1955	鼠系统应用 🔲 隐藏离线应用	1 🗆 降藏在	线应用			名称 ▼	Q	×
服务		应用名称	描述	状态		容器状态	创建时间 🔺	更新时间 ▲			操作
集群									共有4条,每页显示: 15 🔻 条	« «	1 > »

- 4. 设置应用的基本信息。
 - 应用名称:所创建应用的名称。名称可以包含 1~64 个字符,包括数字、英文字符和连字符(-),且不能以连字符开头。
 - 应用版本:所创建应用的版本。默认为 1.0。
 - 部署集群:所创建应用将要部署到的集群。
 - 默认更新策略:应用更新的方式,您可以选择标准发布或蓝绿发布,参见发布策略说明。
 - 应用描述:应用信息。可以不填写。该信息不能超过 1,024 个字符。该信息将显示在应用列 表页面。
 - 检查最新 Docker 镜像:选中该选项后,表示当镜像 Tag 不变的情况下,也会去仓库拉取最新的镜像。

为了提高效率,容器服务会对镜像进行缓存。部署时,如果发现镜像 Tag 与本地缓存的一 致,则会直接复用而不重新拉取。所以,如果您基于上层业务便利性等因素考虑,在做代码 和镜像变更时没有同步修改 Tag,就会导致部署时还是使用本地缓存内旧版本镜像。而勾选 该选项后,会忽略缓存,每次部署时重新拉取镜像,确保使用的始终是最新的镜像和代码。

创建应用 🔄	返回应用列表				
常见问题: 🔗	限制容器的资源 🔗 高可用性调度	♂通过镜像创建Nginx	通过编排模板创建Wordpress	S 编排模板说明 S 标签	密说明
	应用基本信息		应用配置	\rightarrow	创建完成
应用名称	wordpress-test				
	名称为1-64个字符,可包含数字、	英文字符,或"-",且不能以->	开头		
应用版本:	1.0				
部署集群:	test-swarm	¥			
默认更新策略	■ 标准发布	T			
应用描述:					
	🗷 检查最新Docker镜像 🕖				
					使用镜像创建使用编排模板创建

5. 单击使用镜像创建。

单击使用镜像创建。根据您的需要设置以下参数。

a) 基本配置。

	镜像名称:	支持填写私有registry 选择镜像	镜像版本:	选择镜像版本	
	容器数量:	1	网络模式:	默认	v
香本	Restart :				

• 设置镜像名称和镜像版本。

您可以选择容器服务提供的镜像(单击选择镜像。单击您所需的镜像并单击确定),也可 以直接填写自己的镜像地址,格式是domainname/namespace/imagename:tag。容 器服务会默认使用镜像的最新版本。如果您需要使用镜像的其它版本,单击选择镜像版 本,单击所需版本并单击确定。

- 设置容器数量。
- 设置应用的网络模式。容器服务目前支持默认和 host 网络模式。默认为 bridge 网络模式, host 网络模式允许容器使用 ECS 实例的网络堆栈信息。具体请参见 Docker container networking。
- 设置Restart。

默认为 Always,不管退出状态码是什么,始终重启容器。当指定always时,docker daemon 将无限次数地重启容器。不管容器当时的状态如何,容器也会在 daemon 启动时 尝试重启。

取消勾选 Always,重启策略是 no,表示容器退出时不自动重启。

b) 容器配置。

Command :		
Entrypoint :		
CPU現制:	内存限制:	
Capabilites: ADD DROP		

• 设置容器的启动命令(Command 和Entrypoint)。如果设置了会覆盖镜像的默认配置。

Command 用于指定容器主进程的启动命令。参见 command。

Entrypoint 用于指定容器启动程序及参数,与 command 搭配使用,可将 cmd 的内容作为 参数传递给 Entrypoint。参见 *entrypoint*。

• 设置容器的资源限制(CPU 限制 和内存限制)。

为容器将使用的 CPU 和 内存设置资源上限,有关容器资源限制的详细信息,参见限制容器资源。

• 设置Capabilities。

为容器增加或减少 Linux 的相关权限。参见 capabilities。

- 设置容器启动项。
- C) 网络配置。



• 设置端口映射。为主机和容器指定端口映射,以及网络协议(TCP/UDP)。

用于容器和主机间路由,是简单路由配置和负载均衡路由配置的前置条件,容器通过设置 好的端口映射,对外提供服务。

 简单路由配置。集群会自动创建 acsrouting 应用,包含了路由服务,即 routing,提供 简单路由的功能,每个节点都会部署该服务的一个实例。某个节点内,由 acsrouting _routing_index 容器实现集群内的路由转发,用来路由 HTTP 服务或者 HTTPS 服 务。请参见简单路由#支持 HTTP/HTTPS)。

📕 说明:

暴露 HTTP/HTTPS 服务时,您可以不配置具体的主机端口,可以使用 overlay 网络或者 VPC 网络来直接访问容器的端口。

负载均衡路由配置。预先设置好端口映射,然后再配置 container_port 和 \$scheme
 ://\$[slb_name|slb_id]:\$slb_front_port 的映射。关于负载均衡标签的具体用法,请参见/b。

配置该项参数时,由您自主控制路由的访问路径,包括 SLb 前端端口 > 后端主机端口 > 容器端口 这一整条路由映射。

d) 配置数据卷。

	数据卷:	● 如何使用第三方数据卷					
		主机路径或数据带名	寄醤踏径		权限		
物語以		e.g. /path or name	e.g. /path	1	RW	٣	•
4.9	volumes_from :						

- 选择新建一个数据卷。输入主机路径或者数据卷名,输入容器路径,设置数据卷权限 RW 或 RO。参见 *volume*。
- 或选择 volumes_from。填写其他服务或容器的名称和权限参数,如 service_name:ro
 ,如果不指定访问权限,默认是 rw 权限,具体参见 volumes_from。设置后,可成功授权
 该容器使用其他服务或容器的数据卷。
- e) 设置环境变量。

支持 array、 dictionary 和 boolean 等形式,具体参见环境变量。

f) 设置容器的labels。

有关容器服务支持的扩展标签,参见标签概览。

g) 调度配置



• 设置是否允许容器进行平滑升级。

有关平滑升级的相关信息,参见rolling_updates。

• 设置容器的可用区调度。

您可以选择跨可用区将容器部署在两个不同的可用区;如果您选择了此选项,但是当前集 群没有两个可用区或机器资源不够导致无法分布在两个可用区,容器创建会失败。您也可 以选择尽量跨可用区,容器服务会尽可能地将容器部署在两个不同的可用区中;无法满足 时依然可以成功创建。

如果您不进行此项设置,容器服务会默认将容器部署在同一个可用区。有关可用区调度的详细信息,参见高可用性调度。

• 设置容器的自动伸缩规则。

有关容器自动伸缩的详细信息,参见容器自动伸缩。

h) 设置完毕后,单击创建。

6. (可选)或者单击使用编排模板创建

应用基本信息	应用配置		创建完成	
1			新增服务	
使用已有编排模板	 	保存	模板 上一步 创建	并部署

a) 单击使用已有编排模板 ,或者自主编写一个新模板。

编排模板的内容要求符合 Docker Compose 的格式。

- b) 选择一个模板并单击选择。
- C) 编辑编排模板。

您可以根据自己的需要编辑编排模板。您可以直接在编排模板中进行修改,或者在页面右侧 选择需要修改的服务,单击编辑进行修改或者单击删除删除所选的服务。



此外,您还可以单击服务列表下面的新增服务添加服务。选择你所需要的镜像,进行参数设置并单击确定。

新建服务							>
镜像名称:	支持填写私有registry 说	择镜像					
镜像版本:	辺辺	陸陸線版本					
容器数量:							
端口映射:	主机端口	容器道	志口	publ	ish	协议	操作
	主机端口	容器道	お口			•	添加
环境变量:	变量名称	变量值					操作
	名称	值					添加
数据卷:	主机路径或数据卷名	容器	路径		权限		操作
	主机路径或数据卷名	容器	路径		读写	•	添加
简单路由配置	容器端口	域名:					操作
	容器端口	如 [<sch< td=""><td>ema>://]<domain-name></domain-name></td><td>>[/<co< td=""><td>ntext>]</td><td></td><td>添加</td></co<></td></sch<>	ema>://] <domain-name></domain-name>	>[/ <co< td=""><td>ntext>]</td><td></td><td>添加</td></co<>	ntext>]		添加
	注意:相同端口的多个域名只	能填写在同-	-个条目内。多个域名用';'	分隔			
Restart :	2						
		更 ¥	多设置				
						确定	取消

d) 设置完毕后,单击创建并部署。

2.9.2 应用参数配置说明

本文档旨在帮助您理解通过镜像创建 swarm 应用时界面的参数含义,顺利地进行参数配置,其中对 一部分参数,会给出一些文档资源,供您了解更多信息。

镜像名称

• 选择镜像列表中已有的镜像,提供全局搜索功能。

 直接输入镜像地址,以一个阿里云 wordpress 容器镜像地址为例,如 registry.cnhangzhou.aliyuncs.com/acs-sample/wordpress:4.6,由域名、命名空间、镜像名称 和标签构成一个完整的镜像地址。有关镜像地址的详细信息,参见镜像基本概念。

镜像版本

在镜像列表中选择镜像时,可以指定镜像的版本,即镜像的 tag 。如果不指定,默认会使用镜像的 最新版本。

容器数量

设置容器实例的数量。多个容器实例可有效提高应用的可用性。

网络模式

选择默认或 host。

- 默认:即 bridge 网络类型。连接到默认的网桥 docker0,此模式会为每个容器分配一个独立的 网络命名空间,使用该设置可以看到在容器中创建了eth0。
- host: 允许 container 使用 host 的网络堆栈信息。该方式创建出来的容器,可以看到主机上所 有的网络设备,容器中对这些设备有全部的访问权限。

有关 Docker 容器网络的更多信息,参见 Docker container networking。

restart

指定容器的重启策略。参见restart。

- 不勾选表示任何情况下都不会尝试重启容器。
- 勾选表示系统会一直尝试重启容器,直到指定的容器正常运行。

command

设置容器启动后默认执行的命令及其参数,推荐使用 Exec 格式。此命令会在容器启动且 docker run 没有指定其他命令时运行。参见 command。

如果 docker run 指定了其他命令, command 指定的默认命令将被忽略。

command 有三种格式:

- Exec 格式: CMD ["executable", "param1", "param2"], 这是 command 的推荐格式。
- CMD ["param1", "param2"], 与 Exec 格式的 Entrypoint 指令配合使用,为它提供额外的参数。

• Shell 格式: CMD command param1 param2。

Entrypoint

也是容器启动的执行命令, Entrypoint 指令可让容器以应用程序或者服务的形式运行。

Entrypoint 看上去与 CMD 很像,它们都可以指定要执行的命令及其参数。不同的地方在于 Entrypoint 不会被忽略,一定会被执行,即使运行 docker run 时指定了其他命令。

Entrypoint 有两种格式:

- Exec 格式: ENTRYPOINT ["executable", "param1", "param2"], 这是 Entrypoint 的 推荐格式。
- Shell 格式: ENTRYPOINT command param1 param2。

CPU 和 内存限制

CPU 以 100 代表 1 核,内存的单位为 MB,可以为单个容器设置 CPU 和 内存的资源上限,方便 您进行资源规划。对应的 compose 标签是 mem_limit 和 cpu_shares。具体请参见限制容器资源。

Capabilities

默认情况下, Docker 容器中 root 的权限是有严格限制的。借助 Linux kernel capabilities,可以为 容器赋予相关权限,关于可为容器赋权的参数,参见 *Runtime privilege and Linux capabilities*。

相关参数命令如下:

- ADD 输入框:对应 cap-add: Add Linux capabilities 参数,在此框中输入容器可以 添加的 Capability Key,表示为容器增加这一权限。
- DROP 输入框: 对应 cap-drop: Drop Linux capabilities 参数,在此框中输入容器 默认已有的 Capability key,表示为容器删除这一权限。

容器启动项

勾选 stdin 表示为该容器开启标准输入;勾选 tty 表示为该容器分配一个虚拟终端,以便于向容器发送信号。通常这两个选项是一起使用的,表示将终端(tty)绑定到容器的标准输入(stdin)上,比如一个交互式的程序从用户获取标准输入,并显示到终端中。

端口映射

为主机和容器指定端口映射,以及网络协议(TCP/UDP)。用于容器和主机间路由,实现从外部访问容器。

端口映射是简单路由配置和负载均衡路由配置的前置条件,容器通过设置好的端口映射,对外提供 服务。

简单路由配置

容器服务集群创建成功后,会自动创建 acsrouting 应用,包含了路由服务,即 routing,提供了简单路由的功能,每个节点都会部署该服务的一个实例。某个节点内,由 acsrouting_routing_index 容器实现集群内的路由转发,用来路由 HTTP 服务或者 HTTPS 服务。请参见简单路由#支持 HTTP/HTTPS)。



暴露 HTTP/HTTPS 服务时,您可以不配置具体的主机端口,可以使用 overlay 网络或者 VPC 网络 来直接访问容器的端口。

负载均衡路由

配置该项参数时,由您自主控制路由的访问路径,包括SLB 前端端口 > 后端主机端口 > 容器端口这 一整条路由映射。

预先设置好端口映射,然后再配置 container_port 和 \$scheme://\$[slb_name|slb_id]:\$ slb_front_port 的映射。关于负载均衡标签的具体用法,请参见*lb*。

数据卷

推荐使用数据卷来存储容器产生的持久化数据,更加安全,更容易管理、备份、迁移。参见 Use volumes。

- 选择新建一个数据卷。输入主机路径或者数据卷名,输入容器路径,设置数据卷权限 RW 或 RO 。
- 或选择 volumes_from。填写其他服务或容器的名称和权限参数,如 service_name:ro,如果不 指定访问权限,默认是 rw 权限。参见 volume compose。设置后,可成功授权该容器使用其他 服务或容器的数据卷。

环境变量

环境变量支持键值对的输入形式,支持 array、 dictionary 和 boolean 等格式,具体参见 *environment-variables*。

您可以为 Docker 容器设置相关的环境变量,环境变量可以作为标志,代表部署环境的一些参数。 可以用来传递配置,构建自动化部署脚本等。

labels

标签是将元数据应用到 Docker 对象的一种机制。可以使用标签来构建镜像,记录 license 信息,描述容器、数据卷和网络之间的关系等。可以实现非常强大的功能。

标签支持键值对的输入形式,以字符串形式存储。您可以为容器指定多个标签。支持 Docker 原生的 Docker labels 和标签概览。

平滑升级

选择是否允许开启。开启相当于添加 rolling_update=true 标签, 需配合 probe 标签使用, 可保证容器成功更新, 参见probe 和rolling_updates。

可用区调度

勾选跨可用区或者尽量跨可用区。

您可以选择跨可用区将容器部署在两个不同的可用区;如果您选择了此选项,但是当前集群没有两 个可用区或机器资源不够导致无法分布在两个可用区,容器创建会失败。

您也可以选择尽量跨可用区,容器服务会尽可能地将容器部署在两个不同的可用区中;无法满足时 依然可以成功创建。

如果您不进行此项设置,容器服务会默认将容器部署在同一个可用区。有关可用区调度的详细信息,参见高可用性调度。

自动伸缩

为了满足应用在不同负载下的需求,容器服务支持服务的弹性伸缩,即根据服务的容器资源占用情况自动调整容器数量。

有关容器自动伸缩的详细信息,参见容器自动伸缩。

2.9.3 限制容器资源

Docker 容器的一大优势就是可以限制资源,包括 CPU、内存、IO 等,swarm 集群下可对应用进行 资源限制。

您可以在容器服务管理控制台上通过设置界面上的参数或者通过编写编排模板来限制容器资源。

通过设置界面上的参数

您可以在创建应用 或者变更应用配置 时,设置资源限制。

在 swarm 集群下, cpu 以 100 代表 1 核; 内存的单位为 MB。

CPU規制: 50 内存限制: 512	MB

通过编排模板

在编排模板中,支持通过 mem_limit 和 cpu_shares 标签对 CPU 和内存的资源进行限制。

限制 CPU

一个 CPU 核等于 100 CPU 资源。如果机器配置是 4 核,则总共可用的 CPU 资源为 400。在编排 模板中,可以通过 cpu_shares 参数指定。cpu_shares: 50表示使用 0.5 个核。

限制内存

您可以使用 mem_limit 参数限制内存,单位为 Byte,最小内存为 4MB。如果设置了内存限制,当容器申请的内存超过限制时,容器会因为 OOM 而停止运行。

下面的编排模板演示了如何限制 CPU 和内存。

```
n1:
    expose:
        - 443/tcp
        - 80/tcp
    image: 'nginx:latest'
    cpu_shares: 50 #0.5核
    mem_limit: 536870912 #512MB
    labels:
        aliyun.scale: '1'
    restart: always
    volumes:
        - /var/cache/nginx
```

资源调度

为了保证容器能获得足量的指定资源,比如上述例子里的 0.5 核 CPU 和 512MB 内存,容器服务会 为容器预留资源。比如,一台 4 核的机器,最多会调度 8 个 cpu_shares=50 的容器。但是,如 果创建容器时未指定 cpu_shares 和 mem_limit,则默认不会为容器预留资源。

限制其他资源

其他资源限制请参考 Docker Compose 的说明。

2.9.4 高可用性调度

为了使应用有更高的可用性,容器服务支持将同一个服务的容器调度在不同的可用区(zone)里。 当某个可用区发生故障时,应用依然能够提供服务。

您可以在编排文件中通过环境变量指定对可用区的选择,有以下两种格式。

- availability:az==3:服务至少分布在三个可用区中;如果当前集群没有三个可用区,或机器资源不够导致无法分布在三个可用区,容器创建会失败。
- availability:az==~3:服务尽可能分布在三个可用区中;无法满足时依然可以成功创建。



部署约束只对新创建容器生效,对老容器变更配置时不起作用。

在下面的示例中,服务至少要部署在两个可用区中。

```
nnn:
    expose:
        - 443/tcp
        - 80/tcp
    image: 'nginx:latest'
    environment:
        - 'availability:az==2'
    labels:
        aliyun.scale: '8'
    restart: always
    volumes:
        - /var/cache/nginx
```

2.9.5 指定节点调度

如果您需要将某个服务部署在指定的节点上,可以使用 constraint 关键字来实现这个功能。



部署约束只对新创建容器生效,对老容器变更配置时不起作用。

在下面的示例中,服务指定部署在 node1 上。

```
web:
    image: 'nginx:latest'
    restart: always
    environment:
        - 'constraint:aliyun.node_index==1'
    ports:
        - 80
    labels:
        aliyun.scale: 2
```

容器服务支持以下表达式:

表达式	说明
constraint:aliyun.node_index==1	指定部署到 node1。
constraint:aliyun.node_index!=1	不部署到 node1。

constraint:aliyun.node_index==(1 2 3)	指定部署到 node1 或者 node2 或者 node3。
constraint:aliyun.node_index!=(1 2 3)	部署到除 node1、node2、node3 的其他机器 上。
affinity:image==~redis	尽量部署到有 Redis 镜像的机器上。支持填 写镜像全称,如 registry.cn-hangzhou. aliyuncs.com/xxx/xxx。
affinity:service!=~redis	尽量不部署到有 Redis 服务的机器上。参见服 务部署约束#affinity:service#。

2.9.6 指定多节点调度

如果您希望一个应用在指定的某几个节点上部署,推荐您使用用户标签和 constraint 关键字来 进行部署设置。

📃 说明:

- 部署约束只对新创建容器生效,对老容器变更配置时不起作用。
- 使用用户标签部署应用后,如果您删除了用户标签,不会影响到已经部署的应用,但是会影响 下次新的部署。请谨慎删除用户标签。

操作步骤

- 1. 为节点添加用户标签。
 - a. 登录 容器服务管理控制台。
 - b. 在 Swarm 菜单下,单击左侧导航栏中的集群。
 - C. 选择所需集群并单击右侧的管理。

容器服务		集群列表			您最多	5可以创建 5	个集群,每个集群最多可	可以添加 20 个节	5点 刷新	创建集群 🔹
Swarm	Kubernetes									
概览		常见问题: ③ 如何创建集群 ⑧ 如何添け	n已有云服务器 🔗 🛛	河用区节点管理 🔗 集成日志	服务 🔗 通过Docker客户	端连接集群				
应用		名称 ▼								
服务		集群名称/ID	集群类型 地域	网络类型	集群状态 节点状态 🖉	节点个数	创建时间	Docker版本	2	操作
集群 节点	1	test-swarm 📝 acti-staticizza (14) sociolizza di secolizza di staticizza di secolizza di secolizza di secolizza di secolizza di	阿里云集群 华东1	虚拟专有网络 vpc-bp1vvet9csb16vac2jqoc	●运行中 健康 €	2	2017-12-04 15:26:27	17.06.2-ce	管理	重看日志 删除 监控 更多▼

- d. 单击左侧导航栏中的用户标签。
- e. 勾选您要部署应用的节点并单击添加标签。

<	集評:test-swarm	周期
基本信息	 实例ID/名称 	用户标签
用户标签	Happentitute-extension The interface-extension	
镜像列表 容器列表	2 Hot Bin Mindolf R. R. (De Liter Mindolf R. A.	name:node2 group:2 size:2
集群事件 节点伸缩	 制除标签 添加标签 3 	

f. 输入您自定义的标签键和标签值并单击确定,为所选节点添加用户标签。

添加标签		\times
标签:	group:1	
	确定关	闭

2. 创建应用,选择使用编排模板创建并在编排模板中配置 constraint 关键字,如下所示。

有关如何创建应用,参见创建应用。

```
environment:
- constraint:group==1 #表示在所有带有 "group:1" 标签的节点上部署
```

删除用户标签

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择所需集群并单击右侧的管理。

容器服务	Kubernetes	集群列表				您最多	可以创建 5	个集群,每个集群最多。	可以添加 20 个节	市点 刷新	创建集群	¥ •
概览		常见问题: ③ 如何创建集群 ⑧ 如何添加	加已有云服务器 🔗	跨可用区节点管理 🔗 集成日志	服务 🔗 通	i过Docker客户i	端连接集群					
应用		名称 ▼										
服务		集群名称/ID	集群类型 地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本	2		操作
集群	1	test-swarm 📝 och-tododinasco-to-poordina-andidic carried to	阿里云集群 华东:	虚拟专有网络 l vpc-bp1vvet9csb16vac2jqoc	●运行中	健康 오	2	2017-12-04 15:26:27	17.06.2-ce	管理	查看日志 监控	删除 更多▼

- 4. 单击左侧导航栏中的用户标签。
- 5. 勾选要删除用户标签的节点,选中想要删除的标签并单击删除标签。

<) 集群:test-swarm	用证明
基本信息	2 来例ID/名称	用户标签
	Hope Bertra Heckelmanna iZbar State Sta	
镜像列表 容器列表	 i-be intervention in the second second	name:node2 group:2 size:2
集群事件	4 删除标签 添加标签	
节点伸缩		

6. 在弹出的确认对话框中,单击确定。

删除标签			×
?	您是否确认要删除所选的标签?		
		确定	关闭

2.9.7 查看应用详情

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航中的应用。
- 3. 选择所要查看的应用所在的集群。
- 4. 单击所要查看应用的名称。

容器服务		应用列表						刷新创建应用
Kubernetes	Swarm	常见问题: 8 2 主点	2月 🔗 変更应用配置 🔗 简单	路由蓝绿发布策略 🔗 容	器弹性伸缩			
概览		集群: test-swarm ▼	✔ 隐藏系统应用 📄 隐藏离线	应用 🔲 隐藏在线应用			名称 ▼	Q ¥
服务		应用名称	描述 状态	容器状态	创建时间 🔺	更新时间 🔺		操作
集群		wordpress-test	●就绪	就绪:1 停止:0	2018-06-15 14:34:17	2018-06-15 14:34:17	停止 🌾 变更配置	一 删除 単新部署 単 事件

5. 单击服务列表查看该应用的服务列表。

服务列表	容器列表	日志	事件	路由列表			
服务名称	所属应用	3	服务状态	态 容器状态	镜像		操作
db	wordpre	55	●就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	停止 重启 重	新调度 变更配置 删除 事件
web	wordpre	55	●就绪	就绪:3 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5	停止 重启 重	新调度 变更配置 删除 事件

6. 单击容器列表查看该应用的容器列表。

服务列表 容	器列表	日志	事件 路	由列表				
名称/ID		状态	健康检测	镜像	端口	容器IP	节点IP	操作
wordpress_db_1 5affa87d39d6930	0 61	running	正常	registry.aliyunc sha256:ec7e75e52	3306/tcp	172.18.3.4	172.16.33.	删除 停止 监控 日志 远程终端
wordpress_web_ 30fad224e2a221	10 51	running	正常	registry.aliyunc sha256:592af506c	172.16.33.40:32768->80/tcp	172.18.2.4	172.16.33.	删除 停止 监控 日志 远程终端
wordpress_web_ c12f609891a49c	2 () 56	running	正常	registry.aliyunc sha256:592af506c	172.16.33.41:32768->80/tcp	172.18.3.5	172.16.33.	罰除 停止 监控 日志 远程终端
wordpress_web_ b2b28590f3e1e2	_ 3 ()	running	正常	registry.aliyunc sha256:592af506c	172.16.33.40:32769->80/tcp	172.18.2.5	172.16.33.	删除 停止 监控 日志 远程终端

7. 单击日志查看该应用的日志信息。

每个容器查看条目: 100条 ▼ 按容器名称筛选: 全部 ▼ 按日志起始时间筛选: 下载日志
<pre>wordpress_web_2 2017-11-22T07:16:17.826571577Z WordPress not found in /var/www/html - copying now wordpress_web_2 2017-11-22T07:16:17.836454864Z WARNING: /var/www/html is not empty - press Ctrl+C now if this is an error! wordpress_web_2 2017-11-22T07:16:17.837658616884Z + 1s - A wordpress_web_2 2017-11-22T07:16:17.83765616853Z wp-content wordpress_web_2 2017-11-22T07:16:17.837654445Z + sleep 10 wordpress_web_2 2017-11-22T07:16:17.837654445Z + sleep 10 wordpress_web_2 2017-11-22T07:16:28.729167167Z AM90558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.3.5. Set th e 'ServerName' directive globally to suppress this message wordpress_web_2 2017-11-22T07:16:28.292167478Z AM90558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.3.5. Set th e 'ServerName' directive globally to suppress this message wordpress_web_2 2017-11-22T07:16:28.393899034Z [Wed Nov 22 15:16:28.303688 2017] [mpm_prefork:notice] [pid 6] AH00163: Apache/2.4.10 (Debian) PHP/5.6.23 configu</pre>

8. 单击事件查看该应用的事件信息。

	服务列表	容器列表	日志	事件	路由列表
	0.0017.11.0	15-16-10			
e	シZU17-11-2。 AU2書去田	2 15:10:18			
	创建应用W	orapress完成			
e	0 2017-11-22	2 15:16:17			
	创建服务w	ordpress_web	完成		
e	2017-11-22	2 15:16:17			
	启动容器w	ordpress web	2成功		
G	0 2017-11-22	2 15:16:17			
	创建交器w	ordoress web	25世155		
6	0.0017-11-00	0.15.16.15			
e	2017-11-2	2 15:16:15			
	启动容器w	ordpress_web	_1成功		
e	2017-11-22	2 15:16:15			
	启动容器w	ordpress_web	_3成功		

9. 单击路由列表查看该应用的路由地址。

服务列表 容器列表	日志	事件	路由列表		
路由地址(集群绑定SLB后					
vordpress.ce9f1a31d6a8249e7b5ad8e6d3ffcdadb.cn-hangzhou.alicontainer.com					设置服

2.9.8 启停应用

背景信息

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择目标应用所在的集群。
- 4. 根据应用当前的状态,选择启动或停止。

选择目标应用并单击启动。

容器服务		应用列表						刷新创建应用		
Kubernetes	Swarm	常见问题: & 女 2 应用 &								
应用	1	集群: test-swarm ▼ 🕑 陰調	集群: test-swarm ▼ Ø 隐藏系统应用 □ 隐藏菌线应用 □ 隐藏在线应用 名称 ▼							
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 🔺		操作		
集群		wordpress-test	●目停止	就绪:0 停止:1	2018-06-15 14:34:17	2018-06-15 14:42:22	启动 支更	配置 删除 重新部署 事件		

选择目标应用并单击停止。

容器服务		应用列表						刷新	创建应用
Kubernetes	Swarm	堂见问题 · 《 2 建应用 。	· 亦雨立田配齋 · 心 節单路由立		之题2单性(由)缩				
概览					17 Hz +	0	~		
应用 1		编辑: test-swarm * @ Pe					当 标 *	ų	^
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 🔺			操作
集群		wordpress-test	● 就绪	就绪:1 停止:0	2018-06-15 14:34:17	2018-06-15 14:34:17	停止 变更配置	制除 重	1新部署 事件
节点									

5. 在弹出的对话框中,单击确定。

2.9.9 变更应用配置

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择目标应用所在的集群。
- 4. 选择目标应用并单击变更配置。

容器服务		应用列表						刷新创建应用		
Kubernetes	Swarm	enge, 2 -								
概览										
应用		集群: test-swarm ▼		☑用 □ 隠藏住我心	2/H			Q X		
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 🔺	3	操作		
集群		test	● 就绪	就绪:0 停止:0	2018-06-15 14:45:37	2018-06-15 14:45:37	停止 变更配置	删除 重新部署 事件		
节点										

5. 在弹出的对话框中,修改配置。

送 说明:

您必须修改应用版本;否则,确定按钮不可用。

 \:默认情况下,您变更应用配置时,为了保证原服务容器本地数据卷不丢失,容器服务会 在原机器上重启或重新创建容器。如果您希望将容器调度到其它机器上,您可以选择重新调 度,容器服务会根据您模板中的调度设置将容器调度到其它机器上。更多详情参见变更配置 常见问题。 📋 说明 :

选择该选项将容器调度到其它机器上,容器之前的本地数据卷数据将会丢失。请谨慎操作。
 使用已有编排模板:您可以单击使用已有编排模板,选择所需的编排模板进行配置变更。

新的模板会覆盖当前编排模板。

:

变更配置		×
应用名称: *应用版本:	test 1.1	
应用描述:	注意:提交配查变更需要您更新应用版本号,	
使用最新镜像: 发布模式:	重新调度: 「 本 推 发 布 ▼ ⑦	
模板 :	<pre>1 - test: 2 restart: always 3 expose: 4 - 80/tcp 5 - 443/tcp 6 labels: 7 aliyun.scale: '1' 8 image: 'nginx:latest' 9</pre>	
	使用已有编排模板 标签说明	
	确定	取消

后续操作

如果您变更配置后,发现应用没有更新,可以尝试重新部署应用。有关重新部署应用的详细信息,参见重新部署应用。

2.9.10 重新部署应用

应用部署之后您可以根据您的需求对应用进行重新部署。重新部署会重新拉取应用使用的镜像,因此如果您部署应用之后更新了镜像,重新部署会使用新的镜像进行应用部署。

📋 说明:

重新部署不会更新 volume,宿主机上已有的 volume 仍会继续使用。因此,如果您挂载了 volume 并在新镜像中对 volume 设置进行了修改,重新部署后新设置不会生效。

在以下情况下,您会用到重新部署功能:

- 部署应用之后,您更新了镜像的内容,需要按照新的镜像部署应用。
- 您停止或删除了某些容器,希望可以启动或重新创建这些容器。重新部署时,容器服务会重新启动已经停止的容器并重新创建已经删除的容器。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择应用所在的集群。
- 4. 选择要进行重新部署的应用,单击右侧的重新部署。

容器服务		应用列表						刷新创建应用
Kubernetes	Swarm	2		n -l- who (-m) () -l- dologies				
概览		常儿问题:《	用 & 受更应用配直 & 间単3					
应用	1	集群: test-swarm ▼	■ 隐藏系统应用 □ 隐藏离线点	2月 🔲 隐藏在线应	用		名称 ▼	Q X
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 🔺		3 操作
集群		nginx-test	● 就绪	就绪:1 停止:0	2018-06-15 14:54:41	2018-06-15 14:54:41	停止 变更配置	删除 重新部署 事件

5. 在弹出的对话框中,单击确定。

查看重新部署是否成功

您可以通过查看镜像的 sha256 确定重新部署后容器的镜像是否为最新镜像,从而确定重新部署是 否成功。

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择应用所在的集群。
- 4. 单击应用的名称。

容器服务		应用列表						刷新	城建应用	
Kubernetes	Swarm	常见问题: 8 2 建应用								
概览		集群: test-swarm ▼ 8	2 隐藏系统应用 🗌 隐藏离线应用		名称 ▼	٩	×			
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 ▲			操作	
集群		nginx-test 3	● 就绪	就绪:1 停止:0	2018-06-15 14:54:41	2018-06-15 14:54:41	停止 🦾 変更	記蓋 删除 重新	新部署	
节点				1.0 000- 0						

5. 单击 容器列表并查看镜像的sha256。

如果容器的镜像为新镜像,则重新部署成功。

触发器 1. 年 目前没有任何編	种类型的触发 触发器,点击;	諸只能 台上角按	创建1个❷ 钮创建触发	記録					
服务列表	容器列表	日志	事件	路由列表					
名称/ID		X	怸	健康检测	镜像	端口	容器IP	节点IP	
nginx-test_ng b1778e4e5f7	ginx () Oca4f	r	unning	正常	nginx:1.12.1 sha256:0f5baf09c	sha256:0f5baf090 c1d53be8293f95e f2ea37c48a667d5	c628c0f44 æ80cd542 2.4 i35614b12	1000	删除 停止 监
						а			

2.9.11 删除应用

背景信息

您可以删除不再使用的应用。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择所要删除的应用所在的集群。
- 4. 选择所要删除的应用并单击删除。

容器服务		应用列表						刷新	创建应用
Kubernetes	Swarm								
概览		常川问题: 31 2 王公	第以问题: 8 2 建应用 8 交更应用配置 8 简单路由蓝绿发布策略 8 容器等性伸缩						
应用 1		集群: test-swarm ▼	☑ 隐藏系统应用 □ 隐藏裔线应		名称 ▼	a	ί Χ		
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 ▲		3	操作
集群		test	● 就绪	就绪:1 停止:0	2018-06-15 15:10:59	2018-06-15 15:10:59	停止 🌾 交更配置	删除	重新部署 事件
节点									

5. 在弹出的对话框中,单击确定。

如果您勾选同时移除数据卷(volume),系统将删除与应用相关的所有数据卷。但是,已命名的数据卷无法删除。



2.9.12 运行离线任务

容器服务抽象出离线计算的基本模型,推出了基于 Docker 容器的离线计算功能。

其核心功能包括:

- 作业编排
- 作业调度与生命周期管理
- 存储与日志等功能的集成

基本概念

下表中列出了离线应用与在线应用的概念对比。

概念	离线应用	在线应用
容器	任务执行单元	服务的执行单元
运行历史	任务出错重试的执行历史	无
服务(任务)	一个特定的功能,可以分割成 若干个容器来执行	一组功能相同的容器
应用(作业)	若干个任务的组合	若干个服务的组合

一个离线作业包含若干个任务,每个任务可以由若干个容器来执行,每个容器可以有多个运行历 史;而一个在线应用包含若干个服务,每个服务可以有若干个容器同时服务。

基于 Docker Compose 的作业编排

和在线应用一样,您可以使用 Docker Compose 来描述和编排作业。Docker Compose 支持 Docker 的绝大部分功能,比如:

- CPU、内存等资源限制
- 数据卷 (Volume)

- 环境变量与标签
- 网络模型、端口暴露

除此之外,阿里云容器服务还扩展了以下功能:

- 容器数量:每个任务分成多少个容器
- 重试次数:每个容器重试多少次
- 移除容器:容器运行完后是否删除,可选策略包括 remove-finished(删除完成的容器)、 remove-failed(删除失败的容器)、remove-all(删除全部容器)、remove-none(不 删除)。
- DAG 模型的任务依赖:同一个作业的任务之间可以有依赖关系,被依赖的任务会先执行。

离线作业的 Docker Compose 示例:

```
version: "2"
labels:
 aliyun.project_type: "batch"
services:
 s1:
    image: registry.aliyuncs.com/jimmycmh/testret:latest
   restart: no
    cpu_shares: 10
   mem_limit: 10000000
    labels:
      aliyun.scale: "10"
      aliyun.retry_count: "20"
      aliyun.remove_containers: "remove-all"
  s2:
    image: registry.aliyuncs.com/jimmycmh/testret:latest
    cpu_shares: 50
    mem_limit: 10000000
    labels:
      aliyun.scale: "4"
      aliyun.retry_count: "20"
      aliyun.remove_containers: "remove-finished"
      aliyun.depends: "s1"
```

注意:

- 该功能只支持 Docker Compose 2.0。
- 您需要在作业级别添加标签 aliyun.project_type: "batch"。如果您未添加该标签或标签 值不为 batch,则认为该应用为在线应用。
- 无论您将 restart 设置为什么值,都会被修改为 no。
- 您可以用 aliyun.depends 标签指定依赖关系。可以依赖多个任务,用逗号(,)分隔。
- aliyun.retry_count 的默认值为 3。
- aliyun.remove_containers 的默认值为 remove-finished。

作业生命周期管理

容器状态由容器的运行及退出状态决定;任务状态由该任务中所有容器的状态决定;作业状态由该 作业的所有任务决定。

容器状态

- 运行中(Running):容器在运行。
- 完成 (Finished):容器退出且 ExitCode==0。
- 失败(Failed):容器退出且 ExitCode!=0。

任务状态

- 运行中(Running):有容器在运行。
- 完成(Finished):所有容器都完成了。
- 失败(Failed):有容器失败次数超过给定值。

作业状态

- 运行中(Running):有任务在运行。
- 完成(Finished):所有任务都完成了。
- 失败(Failed):有任务失败了。

上述状态都可以通过 API 获取,方便您自动化运维。

共享存储

容器之间、任务之间会有数据共享和交换,共享存储可以解决这一问题。比如在 Hadoop上跑 MR 作业,是通过 HDFS 来交换数据的。在容器服务中,可以使用三类共享存储,其特性及应用场景对 比如下所示。

存储	优点	缺点	适用范围
OSSFS 数据卷	跨主机共享	读写、ls 性能低;修改 文件会导致文件重写	共享配置文件;附件上 传
阿里云 NAS 数据卷	跨主机共享;按需扩 容;高性能、高可靠 性;挂载速度高	成本略高	需要共享数据的重 IO 应用,如文件服务器 等;需要快速迁移的重 IO 应用,如数据库等

存储	优点	缺点	适用范围
您自己集成成三方存 储,如 Portworx	将集群内的云盘虚拟 成共享的大磁盘;性能 高:snapshot、多拷贝	需要一定运维能力	同 NAS

具体使用数据卷的帮助,可以参考以下文档:

- 创建 OSSFS 数据卷
- *#unique_197*
- 使用 OSSFS 数据卷实现 WordPress 附件共享

集成日志和监控服务

日志和监控是分析离线作业的重要工具。阿里云容器服务集成了阿里云日志服务与云监控功能,只 要在编排模板中添加一个标签,就可以将日志收集到日志服务,将容器的 CPU、内存等数据收集到 云监控。具体使用方便请参考下面的文档。

- 集成日志服务
- 容器监控服务

操作步骤

1. 登录 容器服务管理控制台并创建一个集群。

有关如何创建集群的详细信息,参见创建集群。

- 2. 单击左侧导航栏中的应用并单击右上角的创建应用。
- 3. 设置应用的基本信息,单击使用编排模板创建。
- 4. 填入上文中的编排模板并单击创建并部署。
- 5. 单击应用并单击创建的应用的名称,可以查看应用的运行状态,如下图所示。
| < | 应用: batch | | | | | | | |
|----|-----------|----------|---------|-------|-------------------------------------|-------|--------------------------------|------------------|
| 详情 | | | | | | | | |
| | 基本信息 | | | | | | | |
| | 应用名称: | batch | | | | 创建时间: | 2016-08-11 | 更新时间: 2016-08-11 |
| | | | | | | | | |
| | 触发器 1. | 每种类型的触发 | 器只能创建1个 | 2.关于资 | 资源伸缩类型 | 的触发器② | | |
| | 目前没有任何触 | 虫发器,点击右」 | 上角按钮创建触 | 发器 | | | | |
| | 服务列表 | 容器列表 | 路由列表 | 日志 | 事件 | | | |
| | 服务名称 | 所属应用 | 服务 | 状态 | 容器状 | 态 | 镜像 | |
| | s1 | batch | ●売 | 成 | 等待:0
运行:0
完成:1
失败:0 | D r | registry.aliyuncs.com/jimmycmł | v/testret:latest |
| | s2 | batch | ●売 | 成 | 等待:0
运行:0
完成:4
失败:0 | r | registry.aliyuncs.com/jimmycml | v/testret:latest |
| | | | | | | | | |

2.9.13 定时任务

定时任务是常见需求。普遍的做法是,选择一台或几台机器,通过 crontab 实现定时任务。但是对 于大规模或大量的定时任务,这种做法的缺点非常多,比如:

- 可靠性低,一台机器宕机,该机器上的定时任务就无法执行了。
- 没有调度功能,机器之间的负载可能不均衡。
- 没有重试机制,任务可能运行失败。
- 无法运行大规模分布式任务。

容器服务在离线任务的基础上,增加了定时任务的功能,通过简单的描述,解决了上述问题。关于 离线任务的详细信息,参见运行离线任务。



只有2016年10月25号之后升级了 Agent 版本或新创建的集群才能使用该功能。

基于 Docker Compose 的定时任务描述

同离线任务一样,定时任务也是基于 Docker Compose 的,您只需要在应用模板里添加 aliyun. schedule 标签即可实现定时功能。如下面的例子所示。



定时任务在创建和更新时不会拉取最新镜像,因为使用最新镜像会导致同一个任务在不同的时间使 用不同的镜像,会造成排查问题复杂化。建议您通过更改镜像 tag 来变更镜像。

```
version: "2"
labels:
    aliyun.project_type: "batch"
    aliyun.schedule: "0-59/30 * * * * *"
services:
    s1:
    image: registry.aliyuncs.com/jimmycmh/busybox:latest
    labels:
        aliyun.scale: "5"
        aliyun.scale: "5"
        aliyun.retry_count: "3"
        aliyun.remove_containers: "remove-all"
        command: date
```

说明:

- aliyun.schedule: "0-59/30 * * * * * "表示每 30 秒执行一次该任务; schedule 的格式和 crontab 完全相同(但要注意 schedule 的格式为秒 分 时 天 月 星期,比 Linux 上的 crontab 多了秒这一项),使用的时间为北京时间。
- 因为定时功能只适用于离线任务,所以只要您添加了 aliyun.schedule 标签,系统会自动添加 aliyun.project_type: "batch" 标签,因此上述例子中的 aliyun.project_type:
 "batch" 可以省略。
- 另外,离线任务中所有的功能,在定时任务中依然可用,比如 scale、retry_count、 remove_containers等。有关标签的具体含义,参见运行离线任务。

执行过程

定时任务被创建后,应用处于"等待"状态。当任务指定的时间到达时,任务会被启动运行,其后的 状态变化和离线应用相同;下一个执行时间到达时,应用状态会重复这一过程。

同一个定时任务同一时刻只会有一个实例在执行,如果任务的执行时间大于其执行周期(比如上述 任务的执行时间大于 30 秒),则下一次执行会进入执行队列;如果执行队列长度大于 3,则会丢弃 该次执行。

您可以在应用详情页面单击运行历史查看运行历史及结果,如下图所示。运行历史列表只保留最后 10次的运行历史。

基本信息	基本信息									
应用名称:	应用名称: test				创建时间: 2017-12-11	更新时间: 2017-12-3	11	所在集群: test		
触发器 1.每种类型的触发器只能创建1个⊘							创建	謝发器 ^		
目前没有任何	触发器,点击;	白上角按钮	日创建触发							
服务列表	容器列表	日志	事件	路由列表	运行历史					
名称	状态	:		开始时	间		结束时间			
test	●完	励		2017-1	2-11 23:43:00		2017-12-11 2	3:43:10		
test	st ●完成 2017-12		2-11 23:43:30 2017-12-11 23:43:40			3:43:40				
test	●竞	訪花		2017-1	.7-12-11 23:44:00		2017-12-11 23:44:10			
test	●竞	肠花		2017-1	12-11 23:44:30 20			2017-12-11 23:44:40		
test	●完	励花		2017-1	17-12-11 23:45:00			2017-12-11 23:45:10		
test	●完成 2017-12-1)17-12-11 23:45:30		2017-12-11 23:45:40					
test	●竞	sat		2017-1	2-11 23:46:00		2017-12-11 2	3:46:10		
test	● 完	版		2017-1	2-11 23:46:30		2017-12-11 2	3:46:40		
test	●完	戚		2017-1	2-11 23:47:00		2017-12-11 2	3:47:10		

高可用性

定时任务控制器采用主-从备模式。主控制器故障时,控制功能将切换至备用控制器。

如果任务的执行时刻正好在主从切换期间,则会延迟至切换完成后执行。如果主从切换期间同一个任务有多次执行,切换完成后只会执行一次;因此,为了保证任务不丢失,请不要设计重复周期小于一分钟的任务。

2.9.14 系统默认应用列表

应用名称	中文名称	包含的服务	简介
acsrouting	路由服务	routing	提供 7 层协议的请求路 由服务。组件包括负载 均衡和一个 HAProxy 容器。域名配置正确 后,即可将请求发送到 指定的容器内。使用方 法参见路由服务常见问 题。
acslogging	日志服务	logtail,logspout	与阿里云日志服务相 结合,将容器内应用 程序打印的日志上传到 阿里云日志服务中进行 保存,方便您查找和分

应用名称	中文名称	包含的服务	简介
			析。使用方法参见集成 日志服务。
acsmonitoring	监控服务	acs-monitoring-agent	与阿里云云监控进行 集成,并与目前比较流 行的第三方开源监控框 架做集成,方便您查询 监控信息并配置监控报 警。使用方法参见容器 监控服务。
acsvolumedriver	数据卷服务	volumedriver	与阿里云的存储服务 OSS 以及 NAS 做集 成,方便您以 volume 卷(数据卷)的方式来 使用共享式存储,告别 有状态的容器运维。使 用方法参见概述。

2.10 配置项

2.10.1 创建配置项

背景信息

容器服务支持创建配置项,并通过配置参数传递配置,方便您对多个容器环境变量进行管理。

使用限制

- 创建配置项时,您只能选择当前已有集群的地域。
- 如果所选地域下的集群中没有节点,创建配置项会失败并报错。
- 当配置项所属地域下的集群全部被删除,或者地域内没有存在节点的集群时,您在配置文件列表 页面下无法查看该地域下的配置项。一旦该地域下有包含节点的集群,您即可在配置文件列表页 面查看该地域下的配置项。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的配置项。

3. 选择需要创建配置项的地域并单击创建配置项。

容器服务		配置文件列表		周新创建配置项
Kubernetes	Swarm			3
概览		地域: 华东1 🔻 2		
1971H		配置文件名	描述	撮作
服务				冊(日本) 45-24
集群		CONCOURSE_EXTERNAL_OKE		103K7 1 79 6X
节点		提示:如果选择的地域下没有任何集群,则	该地域对应的配置文件都无法显示;一旦用户在该地域新建集群,配置文件会恢复可见。每个地域最多可以创建50个配置项。	
网络				
数据卷				
配置项	1			

- 4. 填写配置文件的信息并单击确定。
 - 配置文件名:可包含 1~32 个字符。
 - 描述:最多可包含 128 个字符。
 - 配置项:单个地域下,您最多可以设置 50 个配置项。填写变量名称和变量值后,需要单击右侧的添加。

您也可以单击编辑配置文件 在弹出的对话框里编写配置项并单击确定。

配置文件		
* 配置文件名:	test_config 名称长度最大为32个字符,最小为1个字符。	
描述:		
配置项:	/////////////////////////////////////	
HUE A	<u>安量名称</u> <u>安量值</u> 操作	
	scale_number 3 编辑 删除 parallelism_number 1 添加]
	变量名长度最大为32个字符,最小为1个字符;变量值长度最大为128个字符,最小为1个字符。 变量值不能重复,变量名和变量值不能为空。	
	确定取消	

本示例中设置了 scale_number和 parallelism_number变量,分别用于传递阿里云扩展标 签scale和rolling_updates的参数。



2.10.2 修改配置项

背景信息

您可以修改配置项的配置。

道 说明:

修改配置文件会影响使用该配置文件的应用。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的配置项。
- 3. 选择所需的地域。

4. 选择需要修改的配置项并单击右侧的修改。

容器服务		配置文件列表		刷新创建配置项
Kubernetes	Swarm			
概览		地域: 华东1 🔻 2		
应用				
服务		配置文件名	描述	操作
集群		test_config		删除 修改
节点		提示: 如果选择的地域下没有任何集群,则该地域对应的雕	置文件都无法显示;一旦用户在该地域新建集群,配置文件会恢复可见。每个地域最多可以创建50个配置	项. 3
网络				
数据卷				
配置项	1			

5. 在弹出的确认对话框中,单击确定。

修改确认			\times
	修改配置文件会影响使用该配置文件的应用,您确定要修改吗?		
		确定	取消

- 6. 修改配置项。
 - 选择需要修改的配置项并单击右侧的编辑,修改配置后单击保存。
 - 或者单击编辑配置文件,完成编辑后单击确定。

配置文件				
* 配置文件名	: test_config			
描述:				
	描述最长不能超过128	个字符。		11
配置项:	编辑配置文件			
	变量名称	变量值		操作
	scale_number	3	编辑	删除
	parallelism_number	1	编辑	删除
	名称	值		添加
	变量名长度最大为32个 变量值不能重复,变量	▷字符 , 最小为1个字符 ; 变量 名和变量值不能为空。	植长度最大为128个字符,最小为1个	字符。
		确定 取消		

7. 完成配置项修改后,单击确定。

2.10.3 使用配置项实现多套环境

应用分为代码和配置两个部分,当应用容器化之后,通常通过容器环境变量的方式传递配置,从而 实现同一个镜像使用不同的配置部署多套应用。

使用限制

- 关联配置文件时,仅能关联与应用处于同一地域中的配置文件。
- 目前,在创建应用时关联配置文件的场景仅适用于通过编排模板创建应用。

创建应用

- 1. 登录 容器服务管理控制台。
- 在 Swarm 菜单下,单击左侧导航栏中的配置项,选择需要创建配置项的地域,然后单击创建配置项。

容器服务		配置文件列表	刷新创建配置项
Swarm	Kubernetes		
概览		地域: 华东1 🔻 🔼	
应用			
服务		配置文件名 描述	操作
集群			
节点		揭示:如果选择的地域下没有任何集群,则该地域对应的配置文件都无法显示;一旦用户在该地域新建集群,配置文件会恢复可见。每个地域最多可以创建50个配置项。	
网络			
数据卷			
配置项	1		

- 3. 填写配置文件的信息并单击确定。
 - 配置文件名:可包含 1~32 个字符。
 - 描述:最多可包含 128 个字符。
 - 配置项:每个地域下,您最多可以设置 50 个配置项。

本示例中设置了 size 变量。

* 配罢立供交,	tost anoun		
10001又14名:	test-group		
	名称长度最大为32	个字符,最小为1个字符。	
描述:	test group		*
			•
	描述最长不能超过	128个字符。	
配置项:	编辑配置文件		
	变量名称	变量值	操作
	size	2	编辑 删除
	名称	值	添加
	变量名长度最大为 变量值不能重复,	32个字符,最小为1个字符;变 变量名和变量值不能为空。	量值长度最大为128个字符,最小为1个字符。
		确定取消	Ξ́.

- 在 Swarm 菜单下,单击左侧导航栏中的应用,选择和所创建配置项位于同一地域的集群,然后 单击创建应用。
- 5. 填写应用的基本信息并单击使用编排模板创建。
- 6. 填写如下所示的编排模板并单击创建并部署。

其中, size 即为动态的变量, 这个变量会被配置项中的数值覆盖。

7. 在弹出的对话框中,选择要关联的配置文件,单击使用配置文件变量,并单击确定。

模板参数		×			
关联配置文件	4: test-group ▼				
参数	值	与配置文件 对比			
size	2	相同			
与配置文件对比说明: 相同在选择的关联配置文件中,有相同的变量名,且变量值相同 不同在选择的关联配置文件中,有相同的变量名,但变量值不同 缺失在选择的关联配置文件中,没有相同的变量名					
	使用配置文件变量确定	定取消			

更新应用

如果创建应用时关联了配置文件,您可以通过修改配置项并重新部署的方式更新应用。

- 1. 登录 容器服务管理控制台。
- 在 Swarm 菜单下,单击左侧导航栏中的配置项,选择需要修改的配置项所在的地域,选择需要 修改的配置项并单击修改。

容器服务		配置文件列表		刷新	创建配置项
Swarm	Kubernetes				
概览		地域: 4东1 🔻 2			
应用		第1990 ナルク	ى.··بى		+9 //
服务		印度又件名	调述		1981 F
集群		test-group	test group		删除 修改
节点		提示: 如果选择的地域下没有任何集群,	则该地域对应的配置文件都无法显示;一旦用户在该地域新建集群,配置文件会恢复可见,每个地域最多可以创建50个配置项。		3
网络					
数据卷					
配置项					

- 3. 在弹出的确认对话框中,单击确定。
- 4. 选择要修改的变量并单击编辑(单击后变为保存)。修改变量值,单击保存并单击确定。

*配置文件名:	test- group		
描述:	test group		*
		行。	
配置项:	编辑配置文件		
	变量名称	变量值	操作
	size	3	保存 删除
	名称	值	添加
	变量名长度最大为32个字符 变量值不能重复,变量名利	夺,最小为1个字符;变量值长度最大为128个字符,最 1变量值不能为空。	最小为1个字符。
		确定取消	

 在 Swarm 菜单下,单击左侧导航栏中的应用,选择和所创建配置项位于同一地域的集群,选择 目标应用并单击重新部署。

容器服务		应用列表							刷新	创建应用
Swarm	Kubernetes									
概览		常见问题: 🔗 如	何创建应用	§ 变更应用配置 § 简单	路由蓝绿发布策略	◎ 容器単性伸縮				
应用	1	集群: test	v 🖉 🖡	隐藏系统应用 □ 隐藏离线	应用 🗆 隐藏在线。	立用		名称 ▼	0	x ×
服务		应用名称	描述	状态	容器状态	创建时间 🔺	更新时间 🔺			2 🗉
集群		test-group		●就绪	就绪:2 停止:0	2017-12-11 15:32:56	2017-12-11 15:32:58	停止 变更翻訳	11日 一 删除	重新部署 事件

更新完成后,容器数变为3个。

容器服务	应用列表				
Swarm Kubernetes					
概览	常见问题: 🔗	如何创建应用		置 🔗 简单路	由蓝
应用	集群: test	¥ Ø	隐藏系统应用	□ 隐藏离线应	用
服务	应用名称	描述	状态		容
集群	test-group		●就绪		就停
节点					

触发更新

如果创建应用时关联了配置文件,您可以通过触发器触发的方式进行重新部署。

- 1. 登录 容器服务管理控制台。
- 在 Swarm 菜单下,单击左侧导航栏中的配置项,选择需要修改的配置项所在的地域,选择需要 修改的配置项并单击修改。

容器服务		配置文件列表		刷新创建配置项
Swarm	Kubernetes			
概览		地域: 4年1 7 2		
应用				
服务		配置文件名	描述	操作
集群		test-group	test group	删除 修改
节点		提示: 如果选择的地域下没有任何集群	,则该地域对应的配置文件都无法显示;一旦用户在该地域新建集群,配置文件会恢复可见。每个地域最多可以创建50个配置项。	3
网络				
数据卷 配置项	1			

- 3. 在弹出的确认对话框中,单击确定。
- 4. 选择要修改的变量并单击编辑 (单击后变为保存)。修改变量值,单击保存并单击确定。

* 配置文件名:	test- group		
描述:	test group		•
	」 描述最长不能超过128个字	符。	
配置项:	编辑配置文件		
	变量名称	变量值	握作
	size	3	保存 删除
	名称	值	添加
	变量名长度最大为32个字符 变量值不能重复,变量名和	, 存,最小为1个字符;变量值长度最大为128个字符, 1变量值不能为空。	最小为1个字符。
		确定取消	

5. 创建重新部署触发器。

有关如何创建触发器的详细信息,参见触发器。

应用:test-group						刷新
基本信息						
应用名称: test-group	创建时间: 2017-12-11	更新时间: 2017-12-11		所在集群: test		
触发器 1.每种类型的触发器只能创建1个@					创	建缺发器 ^
触发器链接(鼠标滑过复制)			secret (🕅	乱标滑过复制)	类型	操作
https://cs.console.aliyun.com/hook/trigger?triggerUrl=Y2ViZD NhbGluZ3wxOXZwYzNmOXFINTcwfA==&secret=4	hkZTIwZGMyMjRmOTM4NDIzMTgwMzI3	NmIwM2IxfHRlc3QtZ3JvdXB8c2	46624033 96675355	1077361940523446843048018068873 608000576015686402907302	资源伸缩	删除触发器

6. 触发重新部署触发器。

```
curl "https://cs.console.aliyun.com/hook/trigger?triggerUrl=
Y2ViZDhkZTIwZGMyMjRmOTM4NDIzMTgwMzI3NmIwM2IxfHRlc3QtZ3JvdXB8
c2NhbGluZ3wxOXZwYzNmOXFiNTcwfA==&secret=46624237677565495154
6d6451656a7a66e7f5b61db6885f8d15aa64826672c2"
```

更新完成后,容器数变为3个。

容器服务		应用列	表			
Swarm	Kubernetes					
概览		常见问	题: 🖉 如	旧何创建应用	♂ 变更应用配置	⑥ 简单路由盘
应用		集群:	test	▼ 🖉	隐藏系统应用 🗆	隐藏离线应用
服务		应用名称	弥	描述	状态	容
集群		test-gro	pup		●就绪	就 : 停:
京の						

2.10.4 删除配置项

背景信息

您可以删除不再使用的配置项。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的配置项。
- 3. 选择所需的地域。
- 4. 选择需要删除的配置项并单击右侧的删除。

容器服务		配置文件列表	刷新创建配置项
Kubernetes	Swarm		
概览		地域: 4 年1 7 2	
服务		配置文件名 描述	操作
集群		test_config	删除 修改
节点		提示:如果选择的地域下没有任何集群,则该地域对应的配置文件都无法显示;一旦用户在该地域新建集群,配置文件会恢复可见。每个地域最多可以创建50个配置项。	3
网络			
数据卷	_		
配置项	1		

2.11 服务管理

2.11.1 使用说明

一个应用由一个或多个服务组成,您可以选择变更应用的配置来升级应用,也可以选择单独变更某个服务的配置来独立升级。

2.11.2 查看服务详情

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择所要查看的服务所在的集群。
- 4. 单击所要查看的服务的名称。如下图所示。

容器服务		服务列表					同新
Kubernetes	Swarm		2				
概览		常见问题: Ø	上集團服务到Inten	net 🔗 给暴露公网的	的服务添加域名 🔗	,将访问协议从http修改为https 🔗 更改应用对外端口	
应用		集群: test-si	warm 🔻 🗹 隐藏系	美術服务 🔲 隐藏商約	刬服务 □ 隐藏在线	服务	名称 Y Q X
服务 1		服务名称	所属应用	服务状态	容器状态	镜像	操作
集群		db	wordpress	● 就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	监控 停止 重新调度 变更配置 删除 重启 事件
や点 网络		web 3	wordpress	● 就绪	就绪:3 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5	监控 停止 重新调度 变更配置 删除 重合 事件

5. 您可以在服务详情页面查看该服务的所有容器。

基本信息										
服务名称: web	所	在应用: w	ordpress	镜像: registry.aliyuncs.com/acs-san	nple/wordpres	5:4.5	容器数目: 3		●运行中	Þ
访问端点: http://wo	rdpress.	n an	A DOLLAR STREET, SALES	-hangzhou.alicontainer.com						
容器 日志 配置	置 事件									
名称/ID	状态	健康检测	镜像	端口	容器IP	印点中				操作
wordpress_web_1 66d0d78b4d187c3a	running	正常	registry.aliyunc sha256:592af506c	10.000	100.00		删除 停止 监控	8;	志 辽	躍终端
wordpress_web_2 0ff8a87daae5f014	running	正常	registry.aliyunc sha256:592af506c				删除 停止 监控	8;	志 辽	躍终端
wordpress_web_3 () 3526d2930feb2e07	running	正常	registry.aliyunc sha256:592af506c		100.00		删除 停止 监控	8;	志 辽	躍终端

6. 单击日志查看服务级别的日志信息。

每个容器查看条目: 100条 ▼	按容器名称筛选: 全部 ▼ 按日志起始时间筛选: 下载日志 如何查看日志
wordpress_web_2 2018-01-02T06:25:27.5603243612 172.1 wordpress_web_2 2018-01-02T06:25:30.7212267342 172.1 wordpress_web_2 2018-01-02T06:25:33.8779936122 172.1 wordpress_web_2 2018-01-02T06:25:33.0274906472 172.1	<pre>17.0.1 - [02/Jan/2018:14:25:27 +0800] "GET /license.txt HTTP/1.1" 200 7603 "-" "Go-http-client/1.1" 17.0.1 - [02/Jan/2018:14:25:30 +0800] "GET /license.txt HTTP/1.1" 200 7603 "-" "Go-http-client/1.1" 17.0.1 - [02/Jan/2018:14:25:33 +0800] "GET /license.txt HTTP/1.1" 200 7603 "-" "Go-http-client/1.1" 17.0.1 - [02/Jan/2018:14:25:37 +0800] "GET /license.txt HTTP/1.1" 200 7603 "-" "Go-http-client/1.1" 17.0.1 - [02/Jan/2018:14:25:37 +0800] "GET /license.txt HTTP/1.1" 200 7603 "-" "Go-http-client/1.1"</pre>

7. 单击配置查看服务的配置信息。

容器 日志 配置 事件								
端口映时								
容器端口			映射端口					
80			动态					
环境变量								
变量名称		变量值						
WORDPRESS_AUTH_SALT		changeme						
WORDPRESS_LOGGED_IN_SALT		changeme						
WORDPRESS_NONCE_SALT		changeme						
WORDPRESS_SECURE_AUTH_KEY		changeme						
WORDPRESS_NONCE_KEY		changeme						
WORDPRESS_SECURE_AUTH_SALT		changeme						
WORDPRESS_NONCE_AA		changeme						
WORDPRESS_AUTH_KEY		changeme						
WORDPRESS_LOGGED_IN_KEY		changeme						
数据卷								
主机路径	容器路径		权限					
标签								
标签名		标签值						
aliyun.routing.port_80		http://wordpress						
aliyun.logs		/var/log						
aliyun.probe.initial_delay_seconds		10						
aliyun.probe.url		http://container/license.txt						
aliyun.scale		3						

8. 单击事件查看服务的事件信息。

容器日志	25 事件							
© 2018-1-2 14:23:27								
启动容器wordpress_	_web_2成功							
© 2018-1-2 14:23:27								
启动容器wordpress_	web_3成功							
② 2018-1-2 14:23:26								
创建容器wordpress_	创建容器wordpress_web_2成功							
© 2018-1-2 14:23:26								
从镜像registry.aliyur	ncs.com/acs-sample/wordpress:4.5创建并启动容器wordpress_web_2							
C 2018-1-2 14:23:26								
从镜像registry.aliyur	ncs.com/acs-sample/wordpress:4.5创建并启动容器wordpress_web_3							

2.11.3 启停服务

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择目标服务所在的集群。

4. 根据当前服务的状态,选择启动或停止。

选择目标服务并单击停止。

容器服务		服务列表						刷新
Kubernetes	Swarm		2					
概览		常见问题: &	全人基本服务到Intern	et 🔗 给暴露公网的	的服务添加域名 🔗	将访问协议从http修改为https 🔗 更改应用对外端口		
应用		集群: test-sv	集群: test-swarm ▼ ■ 隐蔽高线服务					
服务 1		服务名称	所厘应用	服务状态	容器状态	镜像		操作
集群		db	wordpress	● 就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	監控 停止 重新御 変更配置 量 3 重启	腹 事件
「口点」		web	wordpress	● 就绪	就绪:3 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5		酸 事件

选择目标服务并单击启动。

容器服务		服务列表					周新	
Kubernetes	Swarm	600 ED (2018)	2					
概览		帛见问题: ⑦		er () 35 # 20 / 20 / 20 / 20 / 20 / 20 / 20 / 20	183572014864 0	HADIOWACANADIS & EBUTHOMBO		
应用		集群: test-sv	集群: test-swarm ▼ ◎ 障蔵系統服务 □ 障蔵高线服务 □ 障蔵在线服务 □ 障蔵在线服务 □ 障蔵在线服务 □ 障					
服务		服务名称	所屬应用	服务状态	容器状态	镜像	操作	
集群		db	wordpress	● 就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	监控 停止 重新调度 变更配置 删 3 重启 事件	
网络		web	wordpress	●已停止	就绪:0 停止:3	registry.aliyuncs.com/acs-sample/wordpress:4.5	些控 居动 - 重新调度 交更配置 删除 重启 事件	

5. 在弹出的对话框中,单击确定。

2.11.4 变更服务配置

背景信息

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择目标服务所在的集群。
- 4. 选择目标服务并单击变更配置。如下图所示。

容器服务	服务列表					同则例行			
Kubernetes Swar	m 常见问题: 🖉	常见问题: 22 Jaan Alexa の 社員書公网的服务添加結名 の 将访问协议从http/修改为https の 更改应用对外端口							
概 应用	集群: test-s	集群: test-swarm ▼ ◎ 除蔵系統服务 ◎ 除蔵在线服务 ◎ 除廠在线服务							
服务 1	服务名称	所雇应用	服务状态	容器状态	鏡像	操作			
集群	db	wordpress	● 就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	<u> 监控</u> 停止 重新调度 変更配置 制除 重启 事件			
节点	web	wordpress	● 就绪	就绪:3 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5	监控 停止 重新调度 夏更配置 删除 重向 専件			

- 5. 在弹出的对话框中,修改服务的配置。
- 6. 单击更新。

2.11.5 重新调度服务

背景信息

您可以重新平衡各个节点运行容器数量,将负载较重节点的容器迁移到新加入的节点和负载较轻的 节点上,实现集群负载重新的平衡。

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择目标服务所在的集群。
- 4. 选择目标服务并单击重新调度。如下图所示。

容器服务		服务列表					刷新
Kubernetes	Swarm		2				
概览		常见问题: 0	人 展翻服务到Interr	iet 🔮 增暴露公网B	9服务添加城省 🕹	"将访问协议从http修改为https 🔗 更改应用对外端口	
应用		集群: test-sv	varm 🔻 🗹 隐藏系	统服务 🔲 隐藏离线	名称 🔻 🔍 🗶		
服务 1		服务名称	所属应用	服务状态	容器状态	镜像	操作
集群		db	wordpress	● 就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	监控 停止 重新调度 变更配置 删除 重同 3 ³ 件
や点		web	wordpress	● 就结	就绪:3 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5	

 在弹出的对话框中,根据服务的情况,选择是否勾选忽略本地数据卷和强制重新调度,最后单 击确定。

提示		×
0	您所选的服务将执行 <mark>重新调度</mark> 操作,您是否确认操作? ■ 忽略本地数据卷 ■ 强制重新调度	
	确定	取消

- 忽略本地数据卷:对于有本地数据卷的容器,重新调度可能会将容器迁移到其他机器造成数据丢失。如果您需要忽略本地数据卷,请选择此参数;否则,对于有本地数据卷的容器将不进行重新调度。
- 强制重新调度:目前为了保证线上服务的稳定性,默认只有在机器内存使用率超过
 60%,CPU使用率超过40%时才会进行重新调度。如果不想受限于这个限制,请选择此参数,容器服务将忽略使用率限制强制重新调度。

道 说明:

已使用内存和 CPU 数值以容器配置为准,因此不一定是机器的实际使用情况。

2.11.6 删除服务

背景信息

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择所要删除的服务所在的集群。
- 4. 选择所要删除的服务并单击删除。

容器服务	服务列表	服务列表							
Kubernetes Swarm	常见问题: 6	1)问题: 2 1 素服得分到Internet & 绘展墨公网的服务添加域名 & 将访问协议从http修改为https 》 更改应用对外端口							
成用	集群: test-sv	warm 🔻 🗹 隐藏著	(统服务 🗌 隠蔵商約	名称 Y Q X					
服务 1	服务名称	所属应用	服务状态	容器状态	镜像	操作			
集群	db	wordpress	● 就绪	就绪:1 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7	监控 停止 重新调度 変更配置 删除 重启 事件			
节点 网络	web	wordpress	● 就緒	就绪:3 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5				

5. 在弹出的对话框中,单击确定。

提示		\times
•	您所选的 服务:web 将执行 删除 操作,您是否确认操 作? ■ 同时移除数据卷(volume) ■ 强制删除	
	确定取	消

2.12 网络管理

2.12.1 跨主机互联的容器网络

容器服务为容器创建全局的网络,集群中的容器都可以通过容器的 eth0 的网络接口访问其它容器。

编排示例

例如:分别在两台机器上创建容器,并记录它们的 IP 地址、容器名、hostname,登录容器远程终端,通过 ping 命令测试这两个容器跨节点网络通信情况。

```
network-test1:
    image: busybox
    hostname: server1
    command: sh -c 'ifconfig eth0; sleep 100000'
    tty: true
    environment:
        - 'constraint:aliyun.node_index==1'
network-test2:
    image: busybox
    hostname: server2
    command: sh -c 'ifconfig eth0; sleep 100000'
    tty: true
    environment:
        - 'constraint:aliyun.node_index==2'
```

容器跨主机互联测试方法

可以看到这两个服务的容器分布在不同的节点上,如下图所示。

服务列表	容器列表	日志	事件路由	列表				
名称/ID		状态	健康检测	镜像	端口	容器IP	印点中	操作
test_network- c4890487b6e	-tes 🕧 6436d	running	正常	busybox:latest sha256:807fd4df4		172.18.0.4	.181.138	删除 停止 监控 日志 远程终端
test_network- f7ded18d8af8	-tes ① 81d26	running	正常	busybox:latest sha256:807fd4df4		172.18.6.4	.181.139	删除 停止 监控 日志 近檀终端

通过容器 IP 进行测试

通过容器服务管理控制台或者通过 test_network-test1_1 容器输出的ifconfig eth0日志,您可以看到 test_network-test1_1 容器的 IP 地址为172.18.0.4。您可以通过连接远程终端在 test_network-test2_1 容器内访问 test_network-test1_1 容器的 IP 地址,来测试网络是否通信。 在 shell 输入栏执行 sh 命令,然后输入 ping 172.18.0.4 命令。如下图所示。

shell	sh	执行
•		
/ # pilly 1/2.18.0.4		
PING 1/2.18.0.4 (1/2.1	18.0.4): 56 data bytes	
64 bytes from 172.18.0).4: seq=0 ttl=62 time=0.297 ms	
64 bytes from 172.18.0).4: seq=1 ttl=62 time=0.295 ms	
64 bytes from 172.18.0).4: seq=2 ttl=62 time=0.282 ms	
64 bytes from 172.18.0).4: seq=3 ttl=62 time=0.309 ms	
64 bytes from 172.18.0).4: seq=4 ttl=62 time=0.271 ms	
64 bytes from 172.18.0).4: seq=5 ttl=62 time=0.264 ms	
64 bytes from 172.18.0).4: seq=6 ttl=62 time=0.258 ms	
64 bytes from 172.18.0).4: seq=7 ttl=62 time=0.276 ms	
64 bytes from 172.18.0).4: seq=8 ttl=62 time=0.263 ms	
64 bytes from 172.18.0).4: seq=9 ttl=62 time=0.255 ms	
64 bytes from 172.18.0).4: seq=10 ttl=62 time=0.268 ms	
64 bytes from 172.18.0).4: seq=11 ttl=62 time=0.279 ms	

通过容器名进行测试

您可以在对应容器的详情页面查看容器名,默认的容器名为 {project-name}_{service-name}

}_{container-index}。本示例的容器名如下图所示。

在 shell 输入栏执行 sh 命令, 然后输入 ping test_network_test1_1, 通过容器名进行网 络连通性测试。如下图所示。

shell	sh	执行
•		
<pre>/ # ping test networ</pre>	k-testl l	
PING test_network-test	:1_1 (172.18.0.4): 56 data bytes	
64 bytes from 172.18.0).4: seq=0 ttl=62 time=0.245 ms	
64 bytes from 172.18.6).4: seq=1 ttl=62 time=0.276 ms	
64 bytes from 172.18.6).4: seq=2 ttl=62 time=0.263 ms	
64 bytes from 172.18.6	0.4: seq=3 ttl=62 time=0.304 ms	
64 bytes from 172.18.6	0.4: seq=4 ttl=62 time=0.254 ms	
64 bytes from 172.18.6).4: seq=5 ttl=62 time=0.281 ms	
64 bytes from 172.18.6).4: seq=6 ttl=62 time=0.270 ms	
64 bytes from 172.18.6).4: seq=7 ttl=62 time=0.278 ms	
64 bytes from 172.18.6).4: seq=8 ttl=62 time=0.308 ms	
64 bytes from 172.18.0).4: seq=9 ttl=62 time=0.244 ms	

通过 hostname 进行测试

在本示例中,在编排模板中指定了 hostname,因此您可以也通过 hostname 进行网络连通性测试。如下图所示。

shell	sh		执行
<pre>• / # ping server1</pre>			
PING server1 (172.18.6	.4): 56 data bytes		
64 bytes from 172.18.0	.4: seq=0 ttl=64 time=0.0	48 ms	
64 bytes from 172.18.0	.4: seq=1 ttl=64 time=0.0	65 ms	
64 bytes from 172.18.0	.4: seq=2 ttl=64 time=0.0	73 ms	
64 bytes from 172.18.0	.4: seq=3 ttl=64 time=0.0	61 ms	
64 bytes from 172.18.0	.4: seq=4 ttl=64 time=0.0	64 ms	
64 bytes from 172.18.0	.4: seq=5 ttl=64 time=0.0	64 ms	
64 bytes from 172.18.0	.4: seq=6 ttl=64 time=0.0	72 ms	
64 bytes from 172.18.0	.4: seq=7 ttl=64 time=0.0	62 ms	
64 bytes from 172.18.6	.4: seq=8 ttl=64 time=0.0	60 ms	
64 bytes from 172.18.0	.4: seq=9 ttl=64 time=0.0	59 ms	

2.12.2 容器服务 VPC 网络使用指南

本文档从容器服务使用 VPC 网络的角度来介绍如何在容器服务中正确地使用 VPC 网络以及相应的 注意事项。

VPC 网段

- 为了可以顺利地在容器服务中创建 VPC 的容器集群,首先需要根据实际情况规划网络。创建
 VPC 网络的时候,您必须指定对应的 CIDR 来划分对应的子网。
- 每个专有网络只能指定1个网段,网段范围如下,其中172.16.0.0/12 是专有网络的默认网段。
 - **__** 10.0.0/8
 - **—** 172.16.0.0/12
 - **__** 192.168.0.0/16

容器网段

在创建 VPC 网络的容器服务集群时,需要指定对应的容器网段,目前容器服务支持的容器网段如下:

- 192.168.1.0/24
- 172.[16-31].1.0/24

网络规划

为了保证容器间的网络互通,需要把每个容器网段都添加到路由表中;因此,为了避免网段冲突,需要提前根据 VPC 和容器网段对自身的应用服务进行相应的网络规划。

网段规划

在 VPC 网段和容器网段中,172 网段是重合的;因此,VPC 网段和容器网段选择 172 网段时,需 要特别注意。下面举例来说明一下。

假设 VPC 网段选择为 172.16.0.0/12,交换机网段设定为 172.18.1.0/24。那么根据交换机的网段定 义,归属于该交换机的 ECS 的 IP 地址应该是 172.18.1.1-172.18.1.252。如果此时,容器网段也定 义为 172.18.1.0/24,那么就会出现容器的 IP 和 ECS 的 IP 重复的情况,在这种情况下会导致容器 间的网络通信出现异常。因此,在使用 VPC 网络时,必须注意网络的规划。

路由表规划

目前,同一个 VPC 的最大路由表条目是 48 条。

示例

下面以一个完整的示例流程,来演示整个容器服务 VPC 网络集群的创建流程。

步骤1创建 VPC 网络

- 1. 登录到 VPC 管理控制台。
- 2. 单击左侧导航栏中的专有网络。
- 3. 选择所需的地域 (本示例中选择华东 1) 并单击创建专有网络。
- 4. 填写专有网络的信息并单击确定。

在本示例中,为了避免和容器网段可能的冲突,选择创建 10.0.0.0/8 网段的 VPC 网络。

创建专有网络		×
*专有网络名称:	VPC网络规划	
	名称为2-128个字符,以大小字母或 中文开头,可包含数字,"_"或"-"	
描述:	VPC网络规划	
	描述可以为空 ; 或填写2-256个中英 文字符 , 不能以http://和https://开 头	
*网段:	10.0.0/8 •	
	① 一旦创建成功,网段不能修改	
		确定取消

步骤 2 创建交换机

VPC 创建好之后,在该 VPC 下创建相应的交换机。

1. 在专有网络列表中,选择刚创建的专有网络。

专有网络	华东1 亚太东南1(新加坡) 华北1 亚太东)	南 2 (悉尼) 中东东部 1 (迪拜)	香港 华北	3 美国东部 1 (弗吉尼亚) 欧洲中部 1 (法兰克福)	华东 2	€刷新	创建专有网络
概览	亚太东北 1 (东京) 华北 2 美国西部 1 (建谷)	华南 1						
专有网络	VPC ID/名称	网段	状态	描述	创建时间	默认专有网络		操作
▼ 网关 NAT网关	vpc-bp1 VPC网络规划	10.0.0/8	可用	VPC网络规划	2017-04-03 14:34:22	Ϋ́.	管理	编辑 删除
▼ 快捷链接	vpc-bp my	10.0.0/8	可用	rds实例	2017-03-17 19:50:41	答	管理	编辑 删除
高速通道 弹性公网IP管理	vpc-bp test	192.168.0.0/16	可用		2017-03-15 10:03:55	否	管理	编辑 删除
产品帮助文档	vpc-br	172.16.0.0/16	可用		2016-11-19 00:24:01	是	管理	编辑 删除
E						共 有 4条 ,每页显示:10	× <	1 > »

- 2. 单击右侧的管理。
- 3. 单击左侧导航栏中的交换机并单击右上角的创建交换机。

<	交换机列表						—个考	7有网络最多只能创建24个交	时换机 2刷新	创建交换机
专有网络洋情路由器	交换机ID ▼ 请输入交换机	D进行精确查询	搜索						2	
交换机 1	交换机 ID/名称	ECS实例数	网段	状态	可用区	可用私有IP数	创建时间	默认交换机	描述	攝作

4. 填写交换机的信息并单击确定。

本示例中,创建华东1地域10.1.1.0/24 网段的交换机。创建 ECS 时,如果选择该交换机,最终 ECS 的 IP 地址为10.1.1.1-10.1.1.252,合计 252 个 IP 地址(也就是说该网段的交换机下可以购买 252 台 ECS)。

创建交换机		×
*名称:	交换机	
	名称为2-128个字符,以大小字母或 中文开头,可包含数字,"_"或"-"	
*专有网络:	vpc-bp1asti7kduqzxxexipc6	
专有网络网段:	10.0.0/8	
*可用区:	华东 1 可用区 E 🔹	
	① 创建后无法修改	
*网段:	10.1.1.0/24	
	① 创建后无法修改 必须等于或属于该专有网络的网 段,网段掩码必须在16和29之间。 例如:192.168.1.0/24	
可用IP数:	252 个	
描述:		
	描述可以为空;或填写2-256个中英 文字符,不能以http://和https://开 头	
	确定	取消

步骤3查看路由表

VPC 和交换机创建好之后,可以查看路由表的信息。

单击左侧导航栏中的路由器。可以看到,交换机的网络会被作为默认的系统路由表。

<	虚拟路由列表				每个路由表最多只能创建48个自定义路由条	この新	添加路由
专有网络详情	路由鸓基本信息					编	辑へ
路由器	名称: -	ID : vrt-bp	and the second		创建时间: 2017-04-03 14:34:22		
~~~	备注: -						
	路由条目列表						
	路由表ID	状态	目标网段	下一跳	下一跳英型 英國	2	操作
	vtb-bp	可用	10.1.1.0/24	-		t	-
	vtb-bp	可用	100.	-		Ť	-

#### 步骤 4 创建容器集群

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的集群。
- 3. 单击右上角的创建集群。

容器服务	集計列表	制新创建集群
概览应用	小助手: 如何创建集群 如何质如已有云服务器 跨可用区节点管理 集成日志服务 通过Docker客户确连接集群	2
服务	名称 •	
集群	集群名称/ID 集群共型 地域 网络英型 集群状态 节点状态 ◎ 节点个数 创建时间 Docker版本	操作
节点	routing-test-online cd3161d06bd7646e0a5a4280d5a7cab79 阿里云集群 华东1 成版专有网络 vpc-bp1gmu8br4ay7beb2wxl8 ●就總 健康 C 1 2017-03-28 14:58:33 1.12.7 升级 管理 3	直看日志   删除 监控   更多→
数据卷		
配置项		
▶ 镜像与模板		
操作日志		

4. 填写集群信息并单击创建集群。

* 集群名称:	容器服务VPC网络					
	名称为1-64个字符,可	包含数字、汉字、英文	(字符,或"-"			
地域:	华北 1	华北 2	华东 1	华东 2	华南 1	亚太东北 1 (东京)
	美国西部1(硅谷)					
可用区:	华东 1 可用区 E	-				
网络类型:	经典网络	专有网络				
	vpc-bp1asti7kduqzxx	æxipc6 (V▼ 交换	机	<b>.</b>		
容器起始网段:	172.18.0.0/24	查看E	有网段 ②			

本示例中,容器起始网段选择 172.18.0.0/24。那么,该集群内的节点上的容器网段为 172.18. [1-254].0/24,每个节点上容器的 IP 地址为 172.18.x.[1-255]。

步骤 5 验证节点 IP

容器集群创建成功后,可以通过校验集群节点 IP 地址,路由表,查看应用容器的 IP 地址等方式来 验证之前提到的网络规划的理论。

通过查看容器集群的节点列表中 ECS 的 IP 地址,即可验证交换的网段。

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的节点。
- 3. 选择节点所在的集群。

从下图可以看到新购的 ECS 的 IP 归属于 10.1.1.0/24 这个交换机网段。

容器服务	节点列表									[	刷新
概览 应用	集群: 容器服务VPC网络 🔻	2									
服务	IP地址(ID)	实例类型	实例ID/名称	状态	容器数目	配置	操作系统	Docker版本	Agent		操作
集群	10.1.1.224 () c4fd34a2761da47fc8679b32	阿里云ECS	<b>i-bp</b> c4fc	●正常	7	CPU : 1核 内存 : 1.954 GB	Ubuntu 14.04.5 LTS	17.03.1-ce	0.9-093100b	监控	更多▼
节点 1											
配置项											
▶ 镜像与模板											
操作日志											

步骤6验证路由表

验证完节点的 IP 地址,继续校验路由表。

- 1. 登录到 VPC 管理控制台。
- 2. 单击左侧导航栏中的专有网络。
- 3. 在专有网络列表中,选择刚创建的专有网络。
- 4. 单击右侧的管理。

专有网络	<b>华东 1</b> 亚太东南 1 (新加坡) 华北 1 亚太东南	12 (悉尼) 中东东部 1 (迪拜)	香港 华北	3 美国东部 1 (弗吉尼亚)	) 欧洲中部 1 (法兰克福)	华东 2	2 刷新	创建专有网络
概览	业太东北1(东京) 华北2 奥迪四部1(姓合)	华南 1						
专有网络	VPC ID/名称	网段	状态	描述	创建时间	默认专有网络		操作
▼ 网关 NAT网关	vpc-bp1 VPC网络规划	10.0.0/8	可用	VPC网络规划	2017-04-03 14:34:22	否	管理	编辑   删除
▼ 快捷链接	vpc-bp my,	10.0.0/8	可用	rds实例	2017-03-17 19:50:41	否	管理	编辑   删除
高速通道 弹性公网IP管理	vpc-bp test	192.168.0.0/16	可用		2017-03-15 10:03:55	Æ	管理	编辑   删除
产品帮助文档	vpc-bp	172.16.0.0/16	可用		2016-11-19 00:24:01	是	管理	编辑   删除
						共有4条 ,每页显示:10	₩ « <	1 > »

5. 单击左侧导航栏中的路由器。

可以看到系统向该 VPC 的路由表里新增网段为 172.18.x.0/24 的路由信息,下一跳为对应的 ECS 实例的 ID。

<	虚拟路由列表				 每个路由表最多只能创建48个自	定义路由条目	C刷新	添加路	铀
专有网络详情	路由器基本信息						编	辑	^
交换机	名称: -		ID : vrt-bp	and an and an a	创建时间: 2017-04-03 14:3	4:22			
	备注: -								
	路由条目列表								
	路由表ID	状态	目标网段	绳一才	下一跳类型	类型		1	操作
	vtb-bp	可用	172.18.1.0/24	i-bp134	ECS 实例	自定义		<del>10</del>	除
Ξ	vtb-bp	可用	10.		-	系统			-
	vtb-bp	可用	100.			系统			-

步骤7验证容器IP

最终,需要验证容器的 IP 是否正确。

本示例中,通过编排模板从容器服务管理控制台部署一个 WordPress 应用,然后通过查看某节点上的容器列表,来验证一下容器的 IP。

有关如何创建 WordPress 应用,参见通过编排模板创建应用。

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的应用。
- 3. 选择应用所在的集群并单击应用的名称。
- 4. 单击容器列表。

<	应用:wordpress								刷新			
详情												
	基本信息.	基本信息										
	应用名称: wordpress 创建时间: 2017-04-03 更新时间: 2017-04-03 所在集群: 容職服务VPC网络											
	● 約次器 1.每种块型的触发器只能创建1个④ 创理被发展											
	目前没有任何触发器,点击右。	上角按钮创建制	±发器									
	服务列表 容器列表	路由列表	日志 専	¥(4								
	名称/ID	状态	健康检测	镜像	満口	容器IP	节点IP		操作			
E	wordpress_db_1 () d204e2b6b60fdd04	running	正常	registry.aliyunc sha256:ec7e75e52	3306/	172.18.1.3	10.	删除   停止   监控   日志	远程终端			
	wordpress_web_1 2cb73324391c367f	running	正常	registry.aliyunc sha256:592af506c	10.1.1	172.18.1.6	10.	删除   停止   监控   日志	远程终端			
	wordpress_web_2  670611fbb978df0c	running	正常	registry.aliyunc sha256:592af506c	10.1.1.	172.18.1.4	10.	删除   停止   监控   日志	远程终端			
	wordpress_web_3 () 4b1bc79655198405	running	正常	registry.aliyunc sha256:592af506c	10.1.1.	172.18.1.5	10.	删除   停止   监控   日志	远程终端			

以上的校验显示我们成功地创建了一个 VPC 网络的容器集群。

# 2.13 数据卷管理

## 2.13.1 概述

Docker 的特性决定了容器本身是非持久化的,容器被删除后其中的数据也会被一并删除。虽然 Docker 提供的数据卷(Volume)可以通过挂载宿主机上的目录来实现持久存储,但在集群环境 中,宿主机上的数据卷有很大的局限性。

- 容器在机器间迁移时,数据无法迁移
- 不同机器之间不能共享数据卷

为了解决这些问题,阿里云容器服务提供第三方数据卷,将各种云存储包装成数据卷,可以直接挂载在容器上,并在容器重启、迁移时自动重新挂载。目前支持云盘、OSSFS 这 2 种存储。

# 2.13.2 创建 OSSFS 数据卷

OSSFS 是阿里云官方提供的基于 FUSE 的文件系统 (项目主页见https://github.com/aliyun/ossfs

)。OSSFS 数据卷可以将 OSS 的 Bucket 包装成数据卷。

由于数据需要经过网络同步到云端,OSSFS 在性能和功能上与本地文件系统有差距。请不要把数据库等重 IO 应用、日志等需要不断改写文件的应用运行在 OSSFS 上。OSSFS 比较适合多容器之间共享配置文件,或者附件上传等没有改写操作的场景。

OSSFS 和本地文件系统具体差异如下所示:

- 随机或者追加写文件会导致整个文件的重写。
- 因为需要远程访问 OSS 服务器,元数据操作(例如 list directory)性能较差。
- 文件/文件夹的重命名操作不是原子的。
- 多个客户端挂载同一个 OSS Bucket 时,需要您自行协调各个客户端的行为。例如,避免多个客户端写同一个文件等等。
- 不支持硬链接(hard link)。

### 前提条件

您的集群必须满足以下两个条件,才可以开通数据卷功能:

• 集群 Agent 的版本为 0.6 或更高。

您可以在集群列表页面查看您的 Agent 的版本。选择所需的集群,单击右侧的更多 > 升 级Agent。

升级Agent - tes	stnassys	$\times$
	<ul> <li>您所选的 集群: testnassys</li> <li>1. 当前的Agent版本为: 0.8-6fa1e84 (最新版本为: 0.8-6fa1e84)</li> <li>2. 升级期间用户应用不受影响,但是无法通过Web界面对集群进行管理操作,也不能用Docker client连接集群的访问端口,时间大约2分钟。</li> <li>3. 点击确认后,您可以关闭对话框继续进行其他集群操作。</li> </ul>	
	确定	取消

如果您的 Agent 版本低于 0.6,请先升级您的 Agent。有关如何升级 Agent,参见升级 Agent。

• 集群里部署了 acsvolumedriver 应用。建议您将 acsvolumedriver 升级到最新版本。

您可以通过升级系统服务部署和升级 acsvolumedriver 应用。详细操作参见升级系统服务。



升级或重启 acsvolumedriver 时,使用 OSSFS 数据卷的容器会重启,您的服务也会重启。

#### 步骤 1 创建 OSS bucket

- 1. 登录 对象存储管理控制台
- **2.** 创建一个 bucket。

本示例创建了一个位于华东 1 地域的 bucket。

f testvolume	<b>¢</b> 💼	
地域:华东 1 创建时间:2017-03-23 1 存储类型:标准存储	.6:20:29	
储存量 0.0Byte	当月公网流出 0.0Byte	当月请求次数 0

## 步骤 2 创建 OSSFS 数据卷

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的数据卷。
- 3. 选择需要创建数据卷的集群并单击页面右上角的创建。

容器服务 - Swarm ▼	数据卷列表						刷新创建
概览	常见问题: 🔗 数据卷	指南					3
应用	集群: test-swarm	· 2					
服务	□ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
集群							
节点				① 沒有旦闻	刊4.03%1HENC%		
网络							
数据卷 1							

 在弹出的对话框中,选择数据卷类型为OSS,设置数据卷参数并单击创建。容器服务会在集群的 所有节点上创建名称相同的数据卷。

创建数据卷	$\times$
数据卷类型:	● OSS ◎ NAS ◎ 云盘
数据卷名:	test
AccessKey ID :	- Andrew State Sta
AccessKey Secret :	The respective sector product of
可选参数:	🗷 allow_other 🕜 🗆 noxattr 🚱
更多参数:	
	更多参数的填写格式可以参考该文档 注意:只有volume driver在0.7版本及以 上的集群才支持该参数,您可以到"应 用"列表中找到 acsvolumedriver 应用, 然后在其详情页的服务列表中查看 volumedriver服务的镜像版本,如果是 0.7以下的话,请升级系统服务
Bucket ID :	testvolume 选择Bucket
访问域名:	◎ 内网域名 ◎ 外网域名 ◎ vpc域名 🖗
文件缓存:	◎ 打开 ⑧ 关闭 🛛 🖗
	创建取消

- 数据卷名:数据卷的名称。数据卷名在集群内必须唯一。
- AccessKey ID、AccessKey Secret:访问 OSS 所需的 AccessKey。您可以从 阿里云账号 AccessKey 控制台 获取。
- Bucket ID: 您要使用的 OSS bucket 的名称。单击选择Bucket,在弹出的对话框中选择所 需的 bucket 并单击选择。
- 访问域名:如果 Bucket 和 ECS 实例位于不同地域(Region),请选择外网域名;如果位于相同地域,需要根据集群网络类型进行选择,若是 VPC 网络,请选择VPC域名,若是经典网络,请选择内网域名。
- 文件缓存:如果需要在不同机器间同步同一个文件的修改(比如在机器 A 中修改文件,在机器 B 中读取修改后的内容),请关闭文件缓存。

**送** 说明:

关闭文件缓存将导致 1s 文件夹变得很缓慢,尤其是同一个文件夹下文件比较多时。因此,没有上述需求时,请打开文件缓存,提高 1s 命令的速度。

您可以在数据卷列表页面查看创建的 OSSFS 数据卷。

数据卷列表						刷新创建
小助手: 数据卷指南						
集群: test v						
□ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
C6f805575f34a440f9964652	0b1e6b89c2185b768c4d8657	本地磁盘	/var/lib/docker/volumes/	redis-demo_redis-demo_1		删除所有同名卷
C6f805575f34a440f9964652	test	OSS文件系统	/mnt/acs_mnt/ossfs/testv		查看	删除所有同名卷
■ 批量删除						

#### 后续操作

创建数据卷之后,您可以在您的应用中使用创建的数据卷。有关如何在应用中使用数据卷,参见使用第三方数据卷。

## 2.13.3 创建云盘类型的数据卷

云盘是阿里云官方提供的块存储系统,是阿里云为云服务器 ECS 提供的一种分布式存储架构的弹性块存储产品。提供数据块级别的随机存储,具有低时延、持久性、高可靠等性能,采用三副本的分布式机制。

云盘可以用于关系型数据库应用或开发测试应用等场景中,具体请参考弹性块存储。

#### 使用限制

- 云盘与集群下的 ECS 实例需要处于同一地域和可用区下。
- 云盘数据卷只支持单个机器挂载,不支持共享模式。
- 一个云盘数据卷只能同时被一个容器使用。

### 前提条件

- 使用云盘数据卷需要先在 ECS 控制台手动创建一个云盘。
- 请将您的 Agent 升级到最新版本。有关如何升级 Agent,参见升级 Agent。
- 集群里部署了 acsvolumedriver 应用。建议您将 acsvolumedriver 升级到最新版本。

您可以通过升级系统服务部署和升级 acsvolumedriver 应用。详细操作参见升级系统服务。

#### 操作步骤

步骤1 创建云盘

本例创建一个与集群处于同一地域和可用区下的云盘。

- 1. 登录 ECS 管理控制台。
- 2. 在左侧导航栏中,单击磁盘。
- 3. 在磁盘列表页,单击右上角的创建云盘进入创建磁盘页面。
- 4. 配置云盘参数。注意选择地域和可用区,根据页面引导完成创建。



购买的云盘必须与服务器在同一个可用区内才能挂载。云盘不能跨可用区或跨地域挂载。

)地域	华北 1	华北 2	华北 3	华北 5	华东 1	华东 2
	华北1可用区C~	华北 2 可用区 E ~	随机分配 ~	随机分配 🗸	华东 1 可用区 F ~	华东 2 可用区 D、
	华南 1	香港	亚太东南 1 (新加坡)	美国西部1(硅谷)	美国东部 1 (弗吉尼亚)	亚太东北 1 (东京)
	华南 1 可用区 B ~	香港可用区 C ~	亚太东南1 可用区B	美国西部1 可用区B/	美国东部1可用	随机分配
	欧洲中部1(法兰克福)	中东东部 1 (迪拜)	亚太东南 2 (悉尼)	亚太东南 3 (吉隆坡)		
	随机分配 ~	随机分配 ~	随机分配 🗸	随机分配 ~		
		20 22760 60 19		用他现在这中		
云盘	SSD云盘 v 如何选择 SSD云盘 / 高效	20 - 32768 GB 18 云盘 / 普通云盘 , 请看 3	<b>300</b> IOPS 目 加密 ⑦ 洋细说明>	用快照创建磁盘		
云盘	SSD云盘        如何选择 SSD云盘 / 高效       1     块       よ       最多可开通 250 块 云盘	20 - 32768 GB <b>18</b> (云盘 / 普通云盘 , 请君 ) , 已开通 5 块	300 IOPS ■ 加密 ⑦ 洋细说明>	用快照创建或盘		
云盘 购买量 ○ 当前配置	<u>SSD云盘</u> ✓ 如何选择 SSD云盘 / 高効 <u>1</u> 块 ★ 最多可开通 250 块 云盘	20 - 32768 GB 18 云盘 / 普通云盘 , 请看 } , 已开通 5 块	300 IOPS ■ 加密 ⑦ 并细说明>	用快照创建磁盘		

### 步骤2 使用云盘创建数据卷

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的数据卷。
- 3. 选择需要创建数据卷的集群并单击页面右上角的创建。

容器服务 - Swarm ▼	数据卷列表						刷新创建
概览	常见问题: 🔗 数据卷	指南					3
应用	集群: test ▼	2					
服务	□ 节点	卷名	驱动	挂载点	引用容器	卷参数	撬作
集群							
节点				① 没有登词	到符合条件的记录		
网络							
数据卷 1							

 在弹出的对话框中,选择数据卷类型为云盘,设置数据卷参数并单击创建。容器服务会在集群的 所有节点上创建名称相同的数据卷。

创建数据卷		$\times$
数据卷类型: 数据卷名:	◎ OSS ◎ NAS ◎ 云盘	
云盘ID:	d-bp1e 选择云盘	
AccessKey ID :	LTA	
AccessKey Secret :	LURSO	
文件系统类型:	ext4	
	设置云盘快照(设置云盘快照,提高数据安 全)	
	注意:请升级最新volume driver后使用该功 能。您可以在集群列表-更多操作-升级系统服 务对volume driver进行管理。	
	创建	取消

- 数据卷名:数据卷的名称。数据卷名在集群内必须唯一。
- 云盘 ID: 您可以选择与集群属于相同地域和可用区下处于待挂载状态的云盘。本例中选择在步骤 1 中创建的云盘的 ID。
- AccessKey ID 和 AccessKey Secret: 您所使用账号的 AccessKey。

容器服务于 2017 年 12 月 5 日开始支持 STS Token 功能。如果您的集群是在此日期后创建的,您在该集群中创建云盘数据卷时,需输入 AccessKey。

• 文件系统类型:您可以选择以什么数据类型将数据存储到云盘上,支持的类型包括 ext4、 ext3、xfs、vfat。

创建成功后,您可以在数据卷列表页面查看云盘类型的数据卷。

数据卷列表						刷新创建
常见问题: 🖉 数据卷指南						
集群: swarm-test ▼						
■ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
<ul> <li>stigogenere/halpiorkanust</li> </ul>	test-clouddisk	云盘	/mnt/acs_mnt/acd/test-cl		查看	删除所有同名卷
<ul> <li>otigoetto-watereraraz</li> </ul>	test-clouddisk	云盘	/mnt/acs_mnt/acd/test-cl		查看	删除所有同名卷
■ 批量删除						

#### 后续步骤

您可以对云盘类型的数据卷进行管理,包括删除所有同名数据卷和查看卷参数。

## 2.13.4 查看或删除数据卷

背景信息

您可以查看已经创建的数据卷的参数或删除已经创建的数据卷。

#### 操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的数据卷并选择所需的集群。

数据卷列表页面列出了当前集群中所有的数据卷,包括本地数据卷和第三方数据卷。

此外,您可以在该页面查看引用了该数据卷的容器。

数据卷列表						刷新创建
小助手: 数据卷指南						
集群: testnassys 🔻						
■ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
C99da0beb55874578a936cf3	d66543348147a92d004c75d7	本地磁盘	/var/lib/docker/volumes/	elk_elk_1		删除所有同名卷
c99da0beb55874578a936cf3	testNAS		/mnt/acs_mnt/nas/testNAS		查看	删除所有同名卷
c99da0beb55874578a936cf3	数据卷		/mnt/acs_mnt/nas/数据卷		查看	删除所有同名卷
1 批量删除						

对于本地数据卷,数据卷名称的格式为node_name/volume_name。

对于第三方数据卷,单击查看可以查看数据卷的参数。

创建第三方数据卷时,容器服务会在集群的所有节点上以同一数据卷名称创建该数据卷,便于容器在不同节点间迁移。您可以单击删除所有同名卷删除所有使用该名称的数据卷。

→ 说明:

当您的数据卷被容器引用时,您无法删除数据卷。数据卷列表页面显示了引用该数据卷的容器。您需要先删除引用了数据卷的容器才可以删除数据卷。

## 2.13.5 使用第三方数据卷

第三方数据卷使用方法跟本地数据卷相同。

您可以在创建应用时设置数据卷的相关信息,或者通过变更已有应用的配置来添加数据卷的设置。

#### 前提条件

您已经在容器服务管理控制台创建了数据卷,参见创建 OSSFS 数据卷。

#### 操作步骤

本示例以 test 集群下的 OSSFS 数据卷test为例进行说明。

数据卷列表						刷新创建
小助手: 数据卷指南						
集群: test v						
□ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
C6f805575f34a440f9964652	0b1e6b89c2185b768c4d8657	本地磁盘	/var/lib/docker/volumes/	redis-demo_redis-demo_1		删除所有同名卷
C6f805575f34a440f9964652	test	oss文件系统	/mnt/acs_mnt/ossfs/testv		查看	删除所有同名卷
1 北量删除						

#### 通过镜像创建应用

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 单击页面右上角的创建应用。
- 填写应用的基本信息(本示例创建一个名为volume的应用)并单击使用镜像创建。本示例 中,部署集群选择test。


应用部署的集群必须和所要使用的 OSSFS 数据卷为同一集群。

1 创建应用 1 1 近	回应用列表							
小助手: 限制容器	韵的资源 高可用性调度	通过镜像创建Nginx	通过编排模板创建Wordpress	编排模板说明 标签说明	拉取镜像失败			
	应用基本信息			应用配置		X	创建完成	
应用名称	volume 名称为1-64个字符,可包含	*数字、英文字符 , 或"-"	,且不能以-开头的提示					
应用版本:	1.0							
部署集群:	test	,						
默认更新策略:	标准发布	•						
应用描述:			<i>k</i>					
	✓ 检查最新Docker镜像							
							使用镜像创建使用编档	1模板创建

5. 选择所需使用的镜像并进行其它参数配置。

<b>道</b> 说明:	
有关如何使用镜像创建应用的详细信息,参见创建应用。	

 单击数据卷中的加号图标,在主机路径或数据卷名处填写数据卷名称,填写容器路径并设置数据 卷的权限。

	数据卷:	● 如何使用第三方数据卷					
		主机路径或数据卷名	容器路径		权限		
数据卷		test	/testvolume	1	RW	Ŧ	•
							_
	volumes_from :						

7. 完成应用配置后,单击创建。

您可以在数据卷列表页面看到 OSSFS 数据卷 test 已经被 volume 应用的容器引用。

数据卷列表						刷新创建
小助手: 数据卷指南						
集群: test •						
□ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
c6f805575f34a440f9964652	0b1e6b89c2185b768c4d8657	本地磁盘	/var/lib/docker/volumes/	redis-demo_redis-demo_1		删除所有同名卷
c6f805575f34a440f9964652	test	oss文件系统	/mnt/acs_mnt/ossfs/testv	volume_volume_1	查看	删除所有同名卷
111日期除						

### 通过编排模板创建应用

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 单击页面右上角的创建应用。
- 填写应用的基本信息(本示例创建一个名为volume的应用)并单击使用编排模板创建。本示例中,部署集群选择test。

224 00 +
- �� �� •

应用部署的集群必须和所要使用的 OSSFS 数据卷为同一集群。

刘建应用	返回应用列表							
常见问题: 🔗 限	制容器的资源		Ø 通过镜像创建Nginx	♂通过编排模板创建Wo	rdpress 🔗 编排模板说明	月 🔗 标签说明		
	应用基本	信息		应用配置		>	创建完成	
应用名称	volume							
	名称为1-64个字	\$符,可包含数字、	英文字符,或"-",且不	能以-开头				
应用版本:	1.0							
部署集群:	test		T					
默认更新策略:	标准发布		Ŧ					
应用描述:								
	■ 检查最新Doc	ker镜像 🕖						
							使用镜像创建	使用编排模板创建

5. 选择使用已有编排模板或使用您自己的编排模板。



创建应用	€ 返回应用列表	ž						
小助手:	限制容器的资源	高可用性调度	通过镜像创建Nginx	通过编排模板创建Wordpress	编排模板说明	标签说明	拉取镜像失败	
	ł	应用基本信息			应用配	<u> </u>		
1 2 3 4 5 6 7 8 9 9 10 11	<pre>test:     restart: alway:     expose:         - 443/tcp         - 80/tcp     labels:         aliyun.scale     image: 'nginx:     volumes:         - 'test:/tes</pre>	s : 'l' latest' volume:rw'						

7. 完成应用配置后,单击创建并部署。

您可以在数据卷列表页面看到 OSSFS 数据卷test已经被volume应用的容器引用。

数据卷列表						刷新创建
小助手: 数据包括用						
集群: test v						
□ 节点	卷名	驱动	挂載点	引用容器	卷参数	操作
C6f805575f34a440f9964652	0b1e6b89c2185b768c4d8657	本地磁盘	/var/lib/docker/volumes/	redis-demo_redis-demo_1		删除所有同名卷
C6f805575f34a440f9964652	test	oss文件系统	/mnt/acs_mnt/ossfs/testv	volume_volume_1	查看	删除所有同名卷
日 批量删除						

### 变更已有应用的配置

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 选择应用所在的集群(本示例中选择test集群),选择需要更新的应用并单击右侧的变更配置。
   有关变更应用配置的详细信息,参见变更应用配置。



所选应用必须和所要使用的 OSSFS 数据卷位于同一集群。

4. 在模板的 volumes 一节中,填写数据卷名称,容器路径和权限。

变更配置		$\times$
应用名称: *应用版本:	volume 1.1 注意:提交配置变更需要您更新应用版本号,否则确定按钮无法点击	
应用描述:		
使用最新镜像:	●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●	
发布模式:	标准发布 🔹 📀	
模板 :	<pre> in volume: image: 'nginx:latest' restart: always i expose:     - 443/tcp     - 80/tcp i labels:     alivun.scale: 'l'     volumes:     - 'test:/testvolume:rw' i  de用已有编排模板 标签说明 </pre>	
	确定	取消

5. 修改完应用配置后,单击确定。

您可以在数据卷列表页面看到 OSSFS 数据卷test已经被volume应用的容器引用。

数据卷列表						刷新创建
小助手:数据卷指南						
集群: test •						
□ 节点	卷名	驱动	挂载点	引用容器	卷参数	操作
c6f805575f34a440f9964652	0b1e6b89c2185b768c4d8657	本地磁盘	/var/lib/docker/volumes/	redis-demo_redis-demo_1		删除所有同名卷
c6f805575f34a440f9964652	test	oss文件系统	/mnt/acs_mnt/ossfs/testv	volume_volume_1	查看	删除所有同名卷
□ 批量删除						

## 2.13.6 常见问题

如果用数据卷名:镜像中已有目录的方式使用第三方数据卷(如ol:/data,而镜像中有/data 目录),启动容器会失败,系统会返回类似于 chown /mnt/acs_mnt/ossfs/XXXX: input/ output error 的错误。

产生这个错误的原因是,对于命名数据卷,Docker 会把镜像中已有的文件复制到数据卷中,并用 chown 设置相应的用户权限,而 Linux 禁止对挂载点使用 chown。

您可以通过以下两种方法之一解决该问题:

• 升级 Docker 到 1.11 或以上版本,升级 Agent 到最新版本并在编排模板中指定 nocopy 选项。Docker 会跳过复制数据的过程,因此不会产生 chown 错误。

```
volumes:
- o1:/data:nocopy
- /tmp:/bbb
```

 如果必须复制数据,您可以不使用数据卷名称,而使用挂载点路径进行设置,比如用/mnt/ acs_mnt/ossfs/XXXX:/data。但这种方式绕开了 volume driver,在机器重启时,无法保证 在 OSSFS 挂载成功之后再启动容器,可能会导致容器挂载了一个本地数据卷。为了避免这种情况,您需要同时使用两个数据卷,其中一个数据卷使用数据卷名称进行设置,另外一个数据卷 使用挂载点路径进行设置。使用数据卷名称进行设置的数据卷只起到和 volume driver 同步的功能,并不用于存储。

volumes:

```
- ol:/nouse
```

- /mnt/acs_mnt/ossfs/XXXX:/data

- /tmp:/bbb

## 2.14 日志管理

# 2.14.1 查看日志

查看应用级别的日志

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择要查看的应用所在的集群。
- 4. 单击要查看的应用的名称。如下图所示。

容器服务	应用列表					刷新 创建应用
概览	常见问题: ②如(2)立用。	S 变更应用配置 S 简单;	路由蓝绿发布策略	略 🔗 容器弹性伸缩		
应用 1	集群: test-swarm ▼	☑ 隐藏系统应用 🔲 隐藏語	鶴銭应用 🔲 隠嘉	藏在线应用	名称 🔻	Q <b>X</b>
服务	应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 ▲	操作
集群	wordpress-test 3	●运行中	运行中:1 停止:0	2017-12-13 11:34:29	2017-12-13 11:34:29	停止   变更配置   删除   重新部署   事件

5. 单击日志查看该应用的日志信息。

您可以选择显示的日志条目,也可以选择下载所有日志到本地。

基本信息			
应用名称: wordpress-test	创建时间: 2017-12-13	更新时间: 2017-12-13	所在集群: test-swarm
触发器 1.每种类型的触发器只能创建1个②			创建触发器 ^
, 目前没有任何触发器,点击右上角按钮创建触发器			
服务列表 容器列表 日志 事件 路由列表			
每个容器查看条目: 100条 ▼		按日元	志起始时间筛选: 下载日志
<pre>wordpress-test_wordpress-test_1   2017-12-13T0 wordpress-test_wordpress-test_1   2017-12-13T0 wordpress-test_wordpress-test_1   2017-12-13T0 using 172.18.2.4. Set the 'ServerName' directi wordpress-test_wordpress-test_1   2017-12-13T0 using 172.18.2.4. Set the 'ServerName' directi wordpress-test_wordpress-test_1   2017-12-13T0 ebian) PHP/7.2.0 configured resuming normal wordpress-test_wordpress-test_1   2017-12-13T0 -D FOREGROUND'</pre>	3:34:47.278521974Z WordPress nc 3:34:47.370655108Z Complete! Wo 3:34:47.428021471Z AH00558: app ve globally to suppress this me 3:34:47.445922829Z AH00558: app ve globally to suppress this me 3:34:47.458021711Z [Wed Dec 13 operations 3:34:47.458038187Z [Wed Dec 13	t found in /var/www/html - copy rdPress has been successfully of che2: Could not reliably detern ssage 03:34:47.457394 2017] [mpm_press 03:34:47.457435 2017] [core:not	ying now copied to /var/www/html mine the server's fully qualified domain name, mine the server's fully qualified domain name, fork:notice] [pid 1] AH00163: Apache/2.4.25 (D tice] [pid 1] AH00094: Command line: 'apache2

### 查看服务级别的日志

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择要查看的服务所在的集群。
- 4. 单击要查看的服务的名称。如下图所示。

容器服务		服务列表					刷新				
Swarm	Kubernetes										
概览		常见问题: & 如何、 务	到Internet 🔗 给暴露公网的	的服务添加域名 🔗 将访问	可协议从http修改为I	https 🔗 更改应用对外端口					
应用		集群: test-swarm ▼	test-swarm         ▼         ● 隐藏紊统服务         ● 隐藏在线服务         ○ 隐藏在线服务         Q         ¥								
服务	1	服务名称	所属应用	服务状态	容器状态	镜像	操作				
集群 节点		wordpress-test 3	wordpress-test	●就绪	<b>就绪:1</b> 停止:0	wordpress:latest	监控   停止   重新调度   变更配置   删除   重启   事件				

5. 单击日志查看该服务的日志信息。

您可以选择显示的日志条目,也可以选择下载所有日志到本地。

基本信息											
服务名称: wordpress-test	所在应用: wordpress-test	镜像: wordpress:latest	容器数目: 1	●就绪							
访问端点: http://wordpress.cd. Charter and Ch											
容器 日志 配置 事件											
每个容器查看条目: 100条 ▼		按日志起始时间筛选:	下载日志	如何查看日志							
wordpress-test_wordpress-test_1   2017- wordpress-test_wordpress-test_1   2017- wordpress-test_wordpress-test_1   2017-	wordpress-test_wordpress-test_1   2017-12-13T05:34:32.457403403Z WordPress not found in /var/www/html - copying now wordpress-test_wordpress-test_1   2017-12-13T05:34:32.555443070Z Complete! WordPress has been successfully copied to /var/www/html wordpress-test wordpress-test 1   2017-12-13T05:34:32.609265931Z AM00558: apache2: Could not reliably determine the server's fully qualified domain name. u										
sing 172.18.1.4. Set the 'ServerName' directive globally to suppress this message wordpress-test_wordpress-test_1   2017-12-13T05:34:32.628428513Z AH00558: apache2: Could not reliably determine the server's fully qualified domain name, u sing 172.18.1.4. Set the 'ServerName' directive globally to suppress this message											
wordpress-test_wordpress-test_1   2017-12-13705:34:32.6407713642 [Wed Dec 13 05:34:32.640585 2017] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.25 (Deb ian) PHP/7.2.0 configured resuming normal operations											

### 查看容器级别的日志

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择要查看的服务所在的集群。
- 4. 单击要查看的服务的名称。如下图所示。

容器服务		服务列表					刷新				
Swarm	Kubernetes	常见问题: & 如何。2 务	到Internet 🔗 给暴露公网的服	务添加域名 🔗 将访问	协议从http修改为http	s 🔗 更改应用对外端口					
成用		集群: test-swarm ▼	: test-swarm • ※ 隐藏系统服务 ◎ 隐藏离线服务 ◎ 隐藏在线服务 名称 •								
服务	1	服务名称	所属应用	服务状态	容器状态	镜像	提出				
集群 节点		wordpress-test 3	wordpress-test	●就绪	<b>就绪:1</b> 停止:0	wordpress:latest	监控   停止   重新调度   变更配置   删除   重启   事件				

5. 选择要查看的容器并单击日志。如下图所示。

基本信	基本信息											
服务名	马称: wor	dpress-te	st	ĘJ	征应用: wordpress-test		镜像	: wordpress:latest		容器数目:1	•	就绪
访问题	访问端点: http://wordpress.cf.) Characteria (Lange Container Cont											
容器	日志	配置	事件									
名称/IC	)		状态	健康检测	镜像	端口	容器IP	节点IP				操作
wordpro afbaae	wordpress-test_w  running		正常	wordpress:latest sha256:f34533655	80/tcp	020014	201106-121120	anga	余   停止   监控	日志	远程终端	

## 您可以查看该容器的日志信息。

容器	日志	配置	事件					
容器 : /w	ordpress-	est_word	dpress-test_	1 1 2 近回列表	监控	日志	远程终端	详情
wordpre wordpre name, u wordpre name, u wordpre 25 (Det wordpre che2 -[	ess-test ess-test using 17 ess-test using 17 ess-test pian) PH ess-test p FOREGR	_wordpre _wordpre 2.18.1.4 _wordpre 2.18.1.4 _wordpre P/7.2.0 _wordpre OUND'	ess-test ess-test 4. Set the ess-test 4. Set the ess-test configure ess-test	<pre>2017-12-13T05:34:32.457403403Z WordPress not found in /var/www/html - copying now 2017-12-13T05:34:32.555443070Z Complete! WordPress has been successfully copied to /var 2017-12-13T05:34:32.609765951Z AH00558: apache2: Could not reliably determine the serve 2017-12-13T05:34:32.628428513Z AH00558: apache2: Could not reliably determine the serve 2014 / 2017-12-13T05:34:32.628428513Z AH00558: apache2: Could not reliably determine the serve 2015 / 2017-12-13T05:34:32.640771364Z [Wed Dec 13 05:34:32.640585 2017] [mpm_prefork:notice] [ 2017-12-13T05:34:32.640792272Z [Wed Dec 13 05:34:32.640635 2017] [core:notice] [pid 1] 2017-12-13T05:34:32.640792272Z [Wed Dec 13 05:34:32.640635 2017] [core:notice] [pid 1]</pre>	/www/ht r's fu r's fu pid 1] AH0009	tml lly qua lly qua AH001 4: Com	alified do alified do 63: Apache mand line:	omain omain 2/2.4. : 'apa

## 2.14.2 集成日志服务

日志服务(Log Service,简称Log)是针对日志场景的平台化服务。无需开发就可以快速完成日志 收集、分发、投递与查询,适用于日志中转、监控、性能诊断、日志分析、审计等场景。容器服务 提供了集成日志服务的能力,可以方便地将应用日志发送到日志服务里。

📕 说明:

在集群管理页,只要单击开启日志服务 > 确定,日志服务成功开启之后,通过内置的 RAM 账 户,会为每个自动创建的 Logstore 创建日志索引,由于阿里云日志服务已经开始收费,因此启用 本功能之后,按照下面的方式进行配置,将会产生计费,收费标准参见计费方式。请务必了解您的 日志量,以免产生大量非预期的费用。

开启日志服务

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 选择目标集群并单击管理。

容器服务		集群列表						您最多可	「以创建 5 个集群,每个集群	轉最多可以添加 20 个	市点 刷新	创建集	群 •
Swarm 概览	Kubernetes	常见问题: ③如何创建集群 ③如何添加引	3有云服务器 🔇	> 跨可用Ⅰ	≤节点管理 🔗 集成日志服务 🖇	,通过Docker客,	户端连接集群						
应用		名称 ▼											
服务		集群名称/ID	集群类型	地域	网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本			操作
集群	1	routing-test-online	阿里云集群	华东1	虚拟专有网络 vpc-bp15k6sx6fhdz2jw4daz0	●运行中	健康 ℃	2	2017-11-30 15:31:48	17.06.2-ce	管理	查看日志   <u>监控</u>	删除   更多+
网络		swarm-test	阿里云集群	华东1	经典网络	●运行中	健康 ℃	4	2017-11-28 10:04:14	17.06.2-ce	管理	宣看日志   监控	删除   更多 +
数据卷											2		

4. 单击页面右上角的开启日志服务。

集群 : swarm-test				开启日志服务	登录镜像仓库	刷新
基本信息				升级Agent	升级系统服务	清理磁盘
集群ID:	经典网络	●运行中	地域: 华东1	节点个数 4	集群扩容 添加	加已有节点
安全组ID: sg-bp16febevhwsu2t7l84k 检测安全组 重新绑定安 全组						

5. 在弹出的确认对话框中,单击确定。

开通容器服务的日志服务之前,您需要先开通阿里云访问控制(RAM)和阿里云日志服务。如果您还未开通,请单击去开通开通访问控制(RAM)和阿里云日志服务。



### 查看 acslogging 服务安装结果

第一次启用日志服务时,容器服务会在您的机器安装日志服务所需的 Agent。您可以在应用列表中 找到该应用。安装成功后,您就可以使用日志服务了。

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的应用。
- 3. 选择目标集群并取消勾选隐藏系统应用。

可以看到 acslogging 应用已安装成功。

容器服务		应用列表						刷新创建应用				
Swarm 概点	Kubernetes	常见问题: 🔗 🙎	■: 8 2 <u>ま</u> 应用 8 交更 3 <u>当</u> 8 简单路由蓝得发布策略 8 容器等性伸缩									
应用	1	集群: routing-tes	t-online 🔻 🗏 隐藏系统应用	名称 ▼	Q <b>X</b>							
服务	-	应用名称	描述	状态	容器状态	创建时间 🔺	更新时间 🔺	操作				
集群		acslogging default	Logging Service	●运行中	运行中:4 停止:0	2017-11-30 15:37:48	2017-11-30 15:38:04	停止   重新部署   事件				
<b>节点</b> 网络		acsmonitoring default	Monitoring Service	●运行中	运行中:2 停止:0	2017-11-30 15:37:49	2017-11-30 15:38:01	停止   重新部署   事件				

同时,系统会在阿里云日志服务上创建一个对应的 project,您可以在日志服务管理控制台上进行查看。project 的名字里包含了容器服务集群的 ID。

acslog-project-ca5ce0cca9-apgob	ca5ce0cca9da346d68ff2ff	华东 1	2017-12-01 16:30:03	修改注释	删除
deslog project descences appear	cabeooccabaab loadonenni		2017 12 01 10/00/00	15-BAUTTLE	againas

### 在编排文件里使用日志服务

大多数的 Docker 应用会直接将日志写到 Stdout,现在您依然可以这样做(对于日志写到文件的场景,可以参考下边的使用文件日志)。在开通日志管理功能后,Stdout 的日志可以自动收集并且发送到阿里云日志服务。

下面的例子创建了一个 WordPress 应用。该应用包含 WordPress 和 MySQL 两个服务,日志会收 集到阿里云日志服务。

MySQL

```
image: mysql
ports:
    - 80
labels:
    aliyun.scale: "1"
environment:
    - MYSQL_ROOT_PASSWORD=password
```

#### WordPress

```
image: registry.aliyuncs.com/jiangjizhong/wordpress
ports:
    - 80
labels:
    aliyun.routing.port_80: wordpress-with-log
    aliyun.log_store_dbstdout: stdout # 采集stdout日志到dbstdout日志库中
    aliyun.log_ttl_dbstdout: 30 # 设置dbstdout日志库日志数据保存30天
links:
    - mysql
```

#### 在上边的编排文件中

- aliyun.log_store_dbstdout: stdout 表示将容器的标准写入 logstore acslogwordpress-dbstdout里。这个标签的格式为 aliyun.log_store_{name}: {logpath }。其中:
  - name 为阿里云日志服务 logstore 的名字,实际创建的 logstore 的名字为acslog-\${app}-\${name}。
  - app 为应用名称。
  - logpath为容器中日志的路径。
  - stdout 是一个特殊的 logpath, 表示标准输出。
- aliyun.log_ttl_<logstore_name> 标签用来设置日志库初始日志保存时间,单位为 天,有效值为 1~365 天,不配置则默认初始化为2 天。



这里设置的是初始配置值,如果后期您需要修改日志保存时间,需要到日志服务控制台进行设置。

用上面的编排文件,您可以在容器服务管理控制台上创建一个名为 wordpress 的应用。在应用启动完成后,可以在阿里云日志管理控制台上找到 logstore acslog-wordpress-dbstdout,其中存储了 wordpress 的日志。

### 在日志服务管理控制台上查看日志

使用上面的编排文件部署应用之后,您可以在阿里云日志服务控制台查看收集到的日志。登录日 志服务管理控制台,找到集群对应的日志服务 project,单击进入。您可以看到编排文件里使用的 logstore acs-wordpress-dbstdout。

@ acslog-project-ca5ce0cc ▲返回Project列表     地									
Logstore列表					查看Endpoint	创建			
请输入Logstore名进行模糊置 搜索									
Lagebous (73)m Ill	1402	口士式连续一		日志消费模式		150.64-			
LOGSTORA标 盖	if:	口心水来民马	日志消费	日志投递	查询分析	17FF			
acslog-wordpress- dbstdout	~	Logtail配置(管理)  诊断   更多	预览	MaxCompute   OSS	查询	修改 删除			

## 在日志索引列中单击查询查看日志。

🗟 acslog-v	acslog-wordpress-dbstdout (雇于 acslog-project-ca5ce0cca9-apqob ) 返回日版 分享 查询分析属性 另存为快速查询 另存为快速查询 另存为投资													
20														
29分16秒	3	0승15탄 31승15탄	32分15秒 33分15秒	34分15秒	35分15秒	36分15秒	37分15秒	38分15秒	39分15秒	40分15秒	41分15秒	42分15秒	43分15秒	44分01秒
					日志	总条数:102 查	阎状态: <b>结果精</b> 确	ð						
原始日志	统	图表												
		时间小	内容 ▼											⊎ ©
1	Q	12-01 16:43:57	source: 10.81. tag_:hostname tag_:path: topic: content: 2017-12-0	89.196 :: ca5ce0cca /acs/log/wordpre 1T08:43:56.992	9da346d68ff2fl ss/dbstdout/wo	(24fba0277c-nc ordpress_web_ 18.0.1 [01/[	ode3 2/stdout Dec/2017:16:43	3:56 +08001 "G	GET /license.b	t HTTP/1.1" ;	200 7603 "-" '	"Go-http-client	/1.1"	
2	Q	12-01 16:43:55	source: 10.81. tag_:hostname tag_:path: topic: content: 2017-12-0	88.133 e: ca5ce0cca /acs/log/wordpre 1T08:43:55.615	9da346d68ff2fl ss/dbstdout/wo	f24fba0277c-nc ordpress_web_ 18.0.1 [01/E	ode1 1/stdout Dec/2017:16:43	3:55 +0800] "G	GET /license.b	t HTTP/1.1" ;	200 7603 "-" '	'Go-http-client	/1.1"	
3	Q	12-01 16:43:53	source: 10.81. tag_:hostname tag_:path: topic: content: 2017-12-0	89.196 e: ca5ce0cca /acs/log/wordpre 1T08:43:53.785	9da346d68ff2fl ss/dbstdout/wo	f24fba0277c-nc prdpress_web_ 18.0.1 [01/E	ode3 2/stdout Dec/2017:16:43	3:53 +0800] "G	GET /license.b	t HTTP/1.1" 2	200 7603 "-" '	"Go-http-client	/1.1"	

### 使用文件日志

如果您不希望日志直接写到 stdout 中,而需要将日志直接写到文件中,比如/var/log/app.log

,可以进行如下配置。

aliyun.log_store_name: /var/log/app.log

其中name为 logstore 的名字, /var/log/app.log为容器内日志的路径。

如果您需要输出多个日志文件到日志服务,可以进行如下配置将文件放在多个目录下。

```
aliyun.log_store_s1: /data/logs/access/access.log
aliyun.log_store_s2: /data/logs/error/error.log
aliyun.log_store_s3: /data/logs/exception/*.log #支持通配符
```

# 📕 说明:

暂不支持多个logstore对应同一个日志目录。上面的例子中有 3 个 logstore s1、s2 和 s3,对应的日志文件必须在 3 个目录下。

### 开启 timestamp

Docker 在收集日志的时候可以选择是否添加 timestamp。您可以在容器服务中通过aliyun.log. timestamp 标签进行配置。默认会添加 timestamp。

• 添加 timestamp

aliyun.log.timestamp: "true"

• 去除 timestamp

aliyun.log.timestamp: "false"

## 2.15 监控

## 2.15.1 容器监控服务

背景信息

容器监控服务依托于阿里云云监控服务,为容器运维用户提供默认监控、报警规则配置等服务。容器服务的监控服务提供容器维度的监控数据展示以及报警功能。此外,容器服务还提供了与第三方 开源监控方案集成的能力(详细信息参见第三方监控方案集成)。

设置报警规则。

在某些关键服务中,您可以根据自己业务的实际情况添加报警规则。容器监控服务会在监控指标达 到告警阈值后短信通知云账号联系人。

操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 通过以下三种方法之一进入容器列表页面。
  - 通过节点。
    - 单击左侧导航栏中的节点。
    - 单击节点的 IP 地址。
  - 通过应用。
    - 单击左侧导航栏中的应用。
    - 选择所要查看的应用所在的集群。
    - 单击所要查看的应用的名称,再单击容器列表。
  - 通过服务。
    - 单击左侧导航栏中的服务。
    - 选择所要查看的服务所在的集群。
    - 单击所要查看的服务的名称。
- 3. 在容器列表中,选择所要查看的容器并单击监控。

服务列表	容器列表	日志	事件	路由列表					
名称/ID		状	态	健康检测	镜像	端口	容器IP	节点IP	操作
wordpress_wordpr 2b506b2e9d24ba0e		ru	nning	正常	wordpress:latest sha256:c0905f52a	80/tcp			删除   停止   监控   日志   远程终端

您可以查看该容器的实时监控信息。



 单击查看历史监控数据/设置报警规则,跳转到阿里云云监控管理控制台,查看该容器对应的历 史监控数据。



### 5. 设置报警规则。

在某些关键服务中,您可以根据自己业务的实际情况添加报警规则。容器监控服务会在监控指标 达到告警阈值后短信通知云账号联系人。

- a) 单击页面右上角的创建报警规则。
- b) 设置关联资源,然后基于您的实际业务需求设置报警规则。

设置报警规则		
报警类型:	间值报答 事件报答	70.00
规则名称:	CPU 探醫	60.00
规则描述:	CPU使用率         ▼         5分钟         平均值         ▼         >=         ▼         70         9	6 40.00
十添加报警期	Rg	0.00
通道沉默时 间:	24JST •	15:02:00 15:46:40 16:20:00 17:08:30
连续几次超过 阈值后报警:	3 • 🛛	時(信: 70) 1995年~17月~11月~11月~11月~11月~11日~11日~11日~11日~11日~11
生效时间:	00:00 👻 至 23:59 👻	

- c) 设置报警的通知方式,包括通知对象,通知方式等。
- d) 按照页面引导,完成对该容器实例的报警规则设置。最后单击确认。
- e) 单击报警规则,查看创建的报警规则以及其相关记录。

监控图表	报警规则			
规则名称	称 状态	(全部) 🔻	启用	监控项 (全部) ▼
CPU报警	嗒 <b>◇</b> 正	常状态	已启用	CPU使用率
	OI	常状态	已启用	ClusterCPUUtilization

# 2.15.2 查看监控信息

查看服务器监控信息

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 单击目标集群的名称。

容器服务		集群列表			您最多可	以创建 5 イ	〉集群,每个集群最多可	以添加 20 个节;	点 刷新	创建集群	¥ -
Swarm	Kubernetes				8)-+nl,n						
概览		第20回题: び如何団建来研 び如何あ	加口有乙酸芬酯 ♥ 時可用区で。 	宗宮理 े 来成日志版务 🍼 1	回 <u>可</u> Docker客户y	和主按朱矸					
应用		名称 ▼									
服务		集群名称/ID	集群类型 地域 网络类型	集群状态	节点状态 🕜	节点个数	创建时间	Docker版本			操作
集群	1	test2	阿里云集群 华东1 经典网络	●运行中	健康 ℃	2	2017-12-11 09:22:39	17.06.2-ce	管理	査看日志   监控	删除 更多▼
口点网络		routing-test-online	阿里云集群 华东1 虚拟专有 vpc-bp1v	网络 vet9csb16vac2jqoc ●运行中	健康 🕄	2	2017-12-04 15:26:27	17.06.2-ce	管理	查看日志   监控	删除 更多▼

### 4. 选择所要查看的节点并单击监控。

集群:	routing-test-online	9 🔻								
IP地址(	(ID)	实例类型	实例ID/名称	状态	容器数目	配置	操作系统	Docker版本	Agent	操作
192.16	8 0	阿里云ECS		●正常	8	CPU:2核 内存:3.859 GB	Ubuntu 16.04.2 LTS	17.06.2-ce	0.10-f574e56	监控│更多▼

## 您可以查看该节点的监控信息。



## 查看容器监控信息

- 1. 登录 容器服务管理控制台。
- 2. 通过以下三种方法之一进入容器列表页面。
  - 通过节点。
    - 单击左侧导航栏中的节点。
    - 单击节点的 IP 地址。
  - 通过应用。

- 单击左侧导航栏中的应用。
- 选择所要查看的应用所在的集群。
- 单击所要查看的应用的名称,再单击容器列表。
- 通过服务。
  - 单击左侧导航栏中的服务。
  - 选择所要查看的服务所在的集群。
  - 单击所要查看的服务的名称。
- 3. 在容器列表中,选择所要查看的容器并单击监控。

容器列表						
						■ 障藏系统服务容器
名称/ID	状态	镜像	山	容器IP	节点IP	操作
wordpress-test_d  a75c181ce85b5703	running	registry.aliyunc sha256:ec7e75e52	3306/tcp	172.19.0.4	121.42.208.7	删除   停止   监控   日志   远程终端
wordpress-test_w ① cf9b9f6a78aceb48	running	registry.aliyunc sha256:592af506c	0.0.0.0:32773 -> 80/tcp	172.19.0.7	121.42.208.7	删除   停止   监控   日志   远程终端
wordpress-test_w <b>@</b> 892540d858a70d5d	running	registry.aliyunc sha256:592af506c	0.0.0.0:32771 -> 80/tcp	172.19.0.5	121.42.208.7	删除   停止   监控   日志   远程终端
wordpress-test_w  ad07d951cc949d49	running	registry.aliyunc sha256:592af506c	0.0.0.0:32772 -> 80/tcp	172.19.0.6	121.42.208.7	删除   停止   监控   日志   远程终端

## 您可以查看该容器的监控信息。



## 2.15.3 自定义监控

容器监控服务集成了阿里云云监控服务,为您提供涵盖容器、应用、集群、节点的监控与告警服务,满足了容器监控的基本需求。但是在很多业务场景中,您可能需要自定义监控以满足您系统和应用的监控需求,所以容器监控服务在基础监控能力以外,还提供了两种自定义监控方式来允许您通过自己编写数据采集脚本或者自己暴露 HTTP 监控数据接口来上报自定义的监控数据。容器服务监控框架会按照每分钟采集一次的频率来执行脚本或者调用 HTTP 接口来采集数据。

前提条件

使用自定义监控功能之前,您必须将容器监控服务与第三方监控方案进行集成(详细信息参见第三 方监控方案集成)。



目前,容器服务监控集成默认只支持 InfluxDB 和 Prometheus。

您的自定义监控数据会上报给您的 InfluxDB 或 Prometheus, 然后再对接您的数据展示和分析服务。

### 使用自定义监控脚本上报监控数据

1. 构建 Docker 镜像,在镜像中添加自定义的数据采集脚本。

采集脚本的输出数据必须遵守 InfluxDB 的数据格式协议,如下所示。

weather,location=us-midwest temperature=82 146583983	0100400200
<pre> measurement ,tag_set   field_set   timestamp  ++</pre>	

更详细的数据格式协议,参见 InfluxDB 数据协议 Line Protocol。

 2. 登录容器服务管理控制台,使用编排模板创建应用,使用aliyun.monitoring.script标签 声明监控服务用于采集数据的脚本。

示例模板如下所示:

```
custom-script:
image: '您自己的镜像仓库地址'
labels:
```

aliyun.monitoring.script: "sh gather_mem.sh"

aliyun.monitoring.script 定义了监控服务执行应用容器内什么命令来收集监控数据。对 应的 label 配置方式如下所示。

```
labels:
aliyun.monitoring.script: "执行脚本的命令"
```

3. 打开 InfluxDB 的 web 端管理界面,查看以数据指标名称为表名的数据库表。

有关如何查看数据库表,参见第三方监控方案集成中的相关信息。

使用自定义 HTTP 监控数据接口采集数据

1. 构建 Docker 镜像,在应用中对外暴露 HTTP 接口。

该接口输出监控数据。您可以自定义监控数据的格式,只需符合 JSON 语法即可。此外,由于 系统不能分辨自定义的 HTTP 接口返回的 JSON 数据中哪些是数据指标字段,哪些是数据指标 的元数据标签,所以您还需要另外一个配置项来指明 JSON 数据中哪些数据的属性是 tag。具体 可参考 *Telegraf JSON* 数据格式。

 登录容器服务管理控制台,使用编排模板创建应用。您需要在模板中填加 aliyun. monitoring.http标签来声明采集数据的接口,使用 aliyun.monitoring.tags: "您自己的 tag 属性名 1,您自己的 tag 属性名 2,……"声明 HTTP 数据接口返回的数据字段中哪些的属性是 tag。

参考模板 :

```
nodejsapp:
command: "bash /run.sh"
ports:
    - "3000:3000"
image: '您自己的镜像仓库地址'
labels:
    aliyun.monitoring.http: "http://container:3000/metrics/data"
    aliyun.monitoring.tags: "tagl,tag2"
```

nodejsapp应用对外暴露的数据接口 http://container:3000/metrics/data 返回的数据 如下所示:

```
{
    "tag1": "tag1value",
    "tag2": "tag2value",
    "field1": 1,
    "field2": 2,
    "field3": true,
    "field4": 1.5
```

}

使用 aliyun.monitoring.tags: "tag1,tag2" 定义上报的 JSON 数据中, tag1 属性和 tag2 属性为上报数据的 tag。

3. 打开 InfluxDB 的 web 端管理界面,查看以 httpjson_ 前缀加容器名称为表名的数据库表。

例如,容器名称为 nodejsapp_nodejsapp_1,则 InfluxDB 中数据库表的表名为 httpjson_nodejsapp_nodejsapp_1。

有关如何查看数据库表,参见第三方监控方案集成中的相关信息。

## 2.15.4 第三方监控方案集成

前提条件

创建一个被监控应用,本例中创建一个 nginx 应用,参见通过镜像创建 Nginx。

背景信息

容器服务提供了与第三方开源监控方案集成的能力。



目前,容器服务监控集成默认只支持 InfluxDB 和 Prometheus。

下面的示例以 InfluxDB 为例介绍如何进行容器服务的第三方监控方案集成。

#### 操作步骤

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择目标集群,单击页面右上角的创建应用。

容器服务		应用列表						刷新	创建应用
Kubernetes 概符	Swarm	常见问题: 82建应用 (	⑦ 变更应用配置  ◎ 简单路由蓝绿发布第1	略 🖉 容器弹性伸缩					3
应用	1	集群: test-swarm ▼ 🗹 障	藏系统应用 🔲 隐藏离线应用 🔲 隐藏在	线应用			名称 ▼	٩	×
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 🔺			操作
集群							共有4条,每页显示: 15▼条	« < 1	> >>

4. 填写应用的基本信息并单击使用编排模板创建。

本示例创建名为influxdb的应用。

创建应用	* 返回应用列表									
常见问题: 《	》限制容器的资源		♂ 通过镜像创發	書Nginx 🔗 通过编排模板	创建Wordpress		♂ 标签说明			
	应	用基本信息				应用配置		>	创建完成	
应用名称	influxdb									
	名称为1-64个字	符,可包含数字、芽	乾文字符,或"-"	,且不能以-开头						
应用版本:	1.0									
部署集群:	test-swarm		٣							
默认更新知	離:标准发布		T							
应用描述:										
	□ 检查最新Dock	ker镜像 🕖								
									使用镜像创建	使用编排模板创建

5. 填写下面的编排模板并单击创建并部署。

# 道 说明:

在实际生产环境中,本示例中的模板需要做一些修改,其中 influxdb 的服务定义部分不要对 宿主机暴露端口。

6. 应用创建成功后,在应用列表页面,单击本示例所创建应用的名称influxdb,查看应用详情。单击容器列表,查看当前应用对外暴露的节点 IP 和端口号,并复制该值(本示例中,复制 8086端口对应的节点 IP 和端口号;该信息为influxdb对外暴露的数据上报地址),如下图所示。

应用 : influxdb									5	刮新
基本信息										
应用名称: influxdb				创建时间: 20	017-12-01	更新时间	同: 2017-12-01		所在集群: test-swarm	
触发器 1.每种类型的制 目前没有任何触发器,点	雙器只能创建1 击右上角按钮创	▲ 小口 全触发器							的建始发露	^
服务列表 容器列表	日志 🧃	事件 路由列	表							
名称/ID	状态	健康检测	镜像				容器IP	节点IP		攝作
influxdb_influxd  191bda6c5f1c285a	running	正常	tutum/influxdb sha256:7aa2a3	<b>0:0</b> 38f2	:8086->8086/tcp :8083->8083/tcp	2	172.18.6.4	172.16.33.49	删除   停止   监控   日志   远程	终端

7. 单击左侧导航栏中的应用,返回应用列表页面。选择准备好的 nginx 应用,单击更新配置,在现 有的模板中,添加以下内容来声明 InfluxDB 和容器监控服务的集成并单击确定。

labels:

aliyun.monitoring.addon.influxdb: "http://刚才复制的节点 IP:端口 号" aliyun.monitoring.addon.influxdb_retention_policy: "default"

🗐 说明:

第三方开源监控集成目前只支持 InfluxDB 和 Prometheus, label 分别为 aliyun. monitoring.addon.influxdb 和aliyun.monitoring.addon.prometheus。标签取值 的格式必须为 schema:hostIp:port。

由于容器监控服务的 Agent 采用了 host 网络模式,容器服务无法使用 link 来识别 InfluxDB,所 以您需要先创建influxdb,再将influxdb对外暴露的数据上报地址添加到应用 labels 中,来通知数 据采集客户端。完成以上步骤以后,监控服务会将 nginx 应用采集到的容器运行状态数据自动写 入 influxdb 中。

8. 在应用列表页面,单击本示例所创建应用的名称influxdb并单击容器列表。复制influxdb容器对 外暴露的端口。

服务列表	容器列表	日志	事件	路由列表				
名称/ID		状态	健康检测	镜像	二世	容器IP	节点IP	操作
influxdb_influ 666d2a745a	u <b>xd ()</b> 929434	running	正常	tutum/influxdb:0 sha256:7aa2a38f2	100-000-001-007:8086->8086/tcp	172.18.2.4	381406131437	删除   停止   监控   日志   远程终端

9. 推荐您通过 Web 代理的方式进行访问。

如果您想通过公网进行访问,首先需要进行公网认证,然后为 influxDB 应用所在的节点配置 8083 和 8086 的入方向安全组规则,VPC 网络的集群通过 http://EIP:8083端口进行访问。

10.在浏览器中访问 InfluxDB 管理页面,查看容器监控服务写入的各项指标数据。

- a) 选择**telegraf**,若未出现 **telegraf**,请先运行 CREATE DATABASE telegraf,并重新部署 需要监控的应用。
- b) 单击Query Templates并在下拉菜单中单击Show Measurements。
- c) 按 Enter 键。

您可以查看数据库表,如下图所示。

	Database: telegraf 🗸
Query: SHOW MEASUREMENTS	
	Query Templates
measurements	Show Databases Create Database Drop Database
name	Show Measurements
docker_container_blkio docker_container_cpu	Show Tag Keys Show Tag Values
docker_container_mem	Show Retention Policies
docker_container_net	Create Retention Policy Drop Retention Policy
	Show Users
	Create User Create Admin User
	Drop User

查看某个表的数据详情,如下图所示。

§≋InfluxD	B Write Data Documentation				Database: telegraf <del>-</del>	
Query: sele	ect * from docker_container_cpu					
					Query Templates	•
docke	r container cpu	1				
docke	r_container_cpu aliyun_cluster_id	aliyun_instance_id	aliyun_project_id	aliyun_service_id	container_id	

### 后续操作

容器服务与 InfluxDB 集成之后,您可以根据自己的情况选择其他数据展示图表框架,比如 Grafana 等。

# 2.15.5 容器自动伸缩

为了满足应用在不同负载下的需求,容器服务支持服务的弹性伸缩,即根据服务的容器资源占用情况自动调整容器数量。

您可以在创建应用时为应用设置容器自动伸缩规则,或者通过变更应用配置为已创建的应用添加容器自动伸缩规则。

弹性伸缩策略:

- 当监测指标值超过所设定的上限,以用户设定的步长增加容器数量。
- 当监测指标值低于所设定的下限,以用户设定的步长减少容器数量。

### 服务监测指标:

- CPU 平均使用量
- 内存平均使用量
- 容器的入网速率(目前仅支持通过编排模板进行设置)
- 容器的出网速率(目前仅支持通过编排模板进行设置)

### 前提条件

- 需要将集群的 Agent 升级到最新版本。有关如何升级 Agent 的详细信息,参见升级 Agent。
- 需要将集群的监控服务(acsmonitoring)升级到最新版本。有关如何升级监控服务的详细信息,参见升级系统服务。
- 开通 RAM 并在集群中更新访问授权信息(操作路径:在容器服务管理控制台,单击左侧导航栏 中的集群,选择所需的集群,单击右侧的更多,然后单击更新访问控制授权信息)。

### 使用说明

- 容器服务在判断监测指标是否超出所设定的上下限时,使用的是采集周期(一分钟)内监测指标的平均值(即 CPU 平均使用量和内存平均使用量),而且只有当连续三个采集周期内的监测指标平均值均超出所设定的上下限时,容器服务才会触发扩容或伸缩操作,以避免因为监控数据抖动而引起频繁的扩容或缩容操作。
- 容器缩容的时候,系统会删除集群里的容器,您需要提前做好数据备份。

### 设置方法

您可以通过以下方法创建容器弹性伸缩:

- 使用镜像创建应用
- 使用编排模板创建应用
- 变更应用配置



如果您设置的容器弹性伸缩不起作用,可以参考弹性伸缩不起作用怎么办来进行错误的排查。

### 使用镜像创建应用

1. 在创建应用时,选择使用镜像创建。

有关如何创建应用的详细信息,参见创建应用。

	应用基本信息.	应用配置	 创建完成
应用名称	nginx		
	名称为1-64个字符,可包含数字、英文字符,或"-",且不能	削以-开头	
应用版本:	1.0		
部署集群:	swarm-test •		
默认更新策略	标准发布  ▼		
应用描述:			
	□ 检查最新Docker镜像 <b>①</b>		
			使用镜像创建使用编排模板创建

2. 在页面最下边的调度配置中,勾选开启自动伸缩并设置自动伸缩参数。

约束规则:

- 扩容条件的可选范围是 50%~100%, 缩容条件的可选范围是 0%~50%。
- 扩容条件和缩容条件的差值不能小于30%。
- 步长的可选范围为 1~5, 默认为 1。
- ・ 设置最小容器数量 和最大容器数量。缩容时,如果容器数 ≤ 最小容器数量,不会进行缩容 操作;扩容时,如果容器数 ≥ 最大容器数量,不会进行扩容操作。

1 说明:

- 建议不要同时设置基于 CPU 使用量和内存使用量的复合伸缩规则。
- 请谨慎设置伸缩策略。如果在您设置伸缩规则的时候,应用就满足所设置的伸缩条件而且伸 缩后应用仍然满足伸缩条件,那么监控将会不断地触发伸缩。

	平滑升级:	■开启 🕜 注意:需要配合健康检查[probe标签]使用才有效果。
	可用区调度:	◎ 跨可用区 ◎ 尽量跨可用区
нц	自动伸缩:	◎开启 注意:如需使用自动伸缩,请先升级集群Agent到最新版本。 指标: CPU使用量
调度配		缩容条件: 使用量< 0 % 扩容条件: 使用量> 70 %
		步长:     1     可选范围:1-5       最大容器数量:     10     可选范围:2-100
		最小容器数量: 1 可选范围:1-100

### 使用编排模板创建应用

1. 在创建应用时,选择使用编排模板创建。

有关如何创建应用的详细信息,参见创建应用。

	应用基本信息		应用配置	>	创建完成
应用名称	nginx 名称为1-64个字符,可包含数字、英文字符	夺,或"-",且不能以-开头			
应用版本:	1.0				
部署集群:	swarm-test	¥			
默认更新策略	标准发布	T			
应用描述:		J.			
	□ 检查最新Docker镜像 <b>0</b>				
				传	明镜像创建使用编排模板创建

- 2. 选择使用已有编排模板 或者使用您自己的编排模板。
- 3. 添加容器自动伸缩的配置信息。
  - 通过新增服务。

在弹出的对话框中,选择 nginx 镜像,配置需要的参数,然后单击更多设置,勾选开启自动伸缩并设置自动伸缩参数。

应用基本信息	$\geq$	应用配置		创建完成	
1				新增服务	
利用编排模板您可以定义和部署多容器应用,支持Docke	er Compose格式。详情	青请参见 https://docs.docker.com/comp	oose/		
使用已有编排模板				保存模板 上一步	创建并部署

• 手动在模板中设置。

在编排模板的labels配置中,添加相应的标签:

- 指定步长(默认值为1):aliyun.auto_scaling.step

- 最小容器数量(默认值为1):aliyun.auto_scaling.min_instances
- 最大容器数量(默认值为10):aliyun.auto_scaling.max_instances
- 以 CPU 使用量为指标
  - 指定上限: aliyun.auto_scaling.max_cpu
  - 指定下限:aliyun.auto_scaling.min_cpu
- 以内存使用量为指标
  - 指定上限:aliyun.auto_scaling.max_memory
  - 指定下限:aliyun.auto_scaling.min_memory
- 以出网速率 (bytes/s) 为指标
  - 指定上限:aliyun.auto_scaling.max_internetOutRate
  - 指定下限:aliyun.auto.scaling.min_internetOutRate
- 以入网速率 (bytes/s) 为指标
  - 指定上限:aliyun.auto_scaling.max_internetInRate
  - 指定下限:aliyun.auto_scaling.min_internetInRate

示例:

应用基本信息	应用配置	创建完成
1 · nginx: 2 restart: always 3 image: 'nginx:latest' 4 · expose: 5 · 80/tcp 6 menswap_limit: 0		包含服务 服务名:nginx 镜像:nginx:latest 编辑 删除
<ul> <li>Iabels:</li> <li>aliyun.rcuting.port_80: nginx</li> <li>aliyun.scale: '1'</li> <li>aliyun.auto_scaling.max_instances: '10</li> <li>aliyun.auto_scaling.max_epu: '70'</li> <li>aliyun.auto_scaling.max_epu: '70'</li> <li>aliyun.auto_scaling.min_epu: '30'</li> <li>aliyun.auto_scaling.min_instances: '1'</li> <li>fs.mm_size: 0</li> <li>menswap_reservation: 0</li> <li>menswap_reservation: 0</li> </ul>		新進設务
18 men_limit: 0		
利用编排模版您可以定义和部署多容器应用,支持Docker Con 使用已有编排模板	apose格式。 详情请参见 https://docs.docker.com/compose/	保存模板 上一步 创建并部署

#### 变更应用配置

您可以通过变更已有应用的配置为容器添加自动伸缩设置。

1. 在应用列表页面,选择需要设置的应用并单击右侧的变更配置。

有关变更配置的操作信息,参见变更应用配置。

集群: sw	varm-test 🔹	☑ 隐藏系统应用 □	隐藏离线应用 🗆 隐;	藏在线应用	名称 ▼	Q <b>X</b>
应用名称	描述	状态	容器状态	创建时间 🔺	更新时间 ▲	操作
nginx		●就绪	<b>就绪:1</b> 停止:0	2018-01-02 11:51:34	2018-01-02 11:51:37	停止   变更配置   删除   重新部署   事件
wordpress		●就绪	<b>就绪:2</b> 停止:0	2018-01-02 11:51:14	2018-01-02 11:52:48	停止   <mark>变更配置</mark>   删除   重新部署   事件

2. 在模板的labels配置中,添加容器自动伸缩相应的标签。

变更配置		$\times$
应用名称: <b>*</b> 应用版本:	wordpress 1.2 注意:提交配置变更需要您更新应用版本号,否则确定按钮无法点击	
应用描述:		
使用最新镜像:	●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●	
发布模式:	标准发布 🔻 🖉	
模板:	<pre>15 restart: always 16 links: 17 - 'db:mysql' 18 labels: 19 aliyun.logs: /var/log 20 aliyun.probe.url: http://container/license.txt 21 aliyun.probe.initial_delay_seconds: '10' 22 aliyun.routing.port_80: http://wordpress 23 aliyun.scale: '1' 24 aliyun.auto_scaling.max_instances: '10' 25 aliyun.auto_scaling.max_cpu: '70' 27 aliyun.auto_scaling.min_cpu: '30' 28 aliyun.auto_scaling.min_instances: '1' 29 db: 30 image: registry.aliyuncs.com/acs-sample/mysql:5.7 31 environment:</pre>	
	确定	取消

### 查看已创建的容器伸缩规则

您可以查看已创建的容器伸缩规则。

- 1. 登录 容器服务管理控制台。
- 2. 单击左侧导航栏中的应用。
- 3. 在应用列表页面,选择所需的应用,单击变更应用配置。

变更配置		×
应用名称: *应用版本:	nginx 1.0 注意:提交配置变更需要您更新应用版本号,否则确定按钮无法点击	
应用描述:		
使用最新镜像:	重新调度:	
发布模式:	标准发布 🔻 🕖	
模板 :	<pre>1 - hginx: 2     restart: always 3     image: 'nginx:latest' 4     expose: 5        - 80/tcp 6     memswap_limit: 0 7     labels: 8        aliyun.routing.port_80: nginx 9        aliyun.scale: '1' 10        aliyun.auto_scaling.max_instances: '10' 11        aliyun.auto_scaling.max_cpu: '70' 13        aliyun.auto_scaling.min_cpu: '30' 14        aliyun.auto_scaling.min_instances: '1' 15        shm_size: 0 16        memswap_reservation: 0 17        kernel_memory: 0 </pre>	
	确定	取消

您可以看到已创建的容器伸缩规则。您可以在模板中修改容器伸缩规则。

# 2.15.6 节点自动伸缩

为了满足应用在不同负载下的需求,容器服务不仅提供了容器级别的弹性伸缩,还提供了节点级别的自动伸缩,即通过监测节点的资源占用情况自动调整节点数量。

节点伸缩采取的策略:

- 当监测指标值超过所设定的扩容条件,以用户设定的扩容步长增加节点数量。
- 当监测指标值低于所设定的缩容条件,以系统默认步长1减少节点数量。

自动伸缩的监测指标:

• 集群 CPU 平均使用量。

• 集群内存平均使用量。

#### 前提条件

- 需要将集群的 Agent 升级到最新版本。有关如何升级 Agent 的详细信息,参见升级 Agent。
- 需要将集群的监控服务(acsmonitoring)升级到最新版本。有关如何升级监控服务的详细信息,参见升级系统服务。
- 开通 RAM 并在集群中更新访问授权信息(操作路径:在容器服务管理控制台,单击左侧导航栏中的集群,选择所需的集群,单击右侧的更多,单击更新访问控制授权信息)。

#### 使用说明

- 容器服务在判断监测指标是否超出所设定的上下限时,使用的是采集周期(一分钟)内监测指标的平均值(即 CPU 平均使用量和内存平均使用量),而且只有当连续三个采集周期内的监测指标平均值均超出所设定的上下限时,容器服务才会触发扩容或伸缩操作,以避免因为监控数据抖动而引起频繁的扩容或缩容操作。
- 节点缩容只会对通过节点扩容创建出来的节点进行,您手动创建或者添加的节点不受影响。如果
   您希望这些手动添加的节点也可以自动缩容,需要为这些节点添加以下标签。

aliyun.reschedule==true

- 节点缩容的时候,系统会删除集群里的 ECS 实例,您需要提前做好数据备份。
- 不要调度有状态服务到可缩容的节点上。
- 扩容出来的 ECS 实例不会影响已经部署的容器。新部署的容器会根据容器的部署规则进行部署。
- 节点缩容时,容器服务会把删除掉的 ECS 实例上的容器迁移到其它 ECS 实例上。

#### 创建节点伸缩规则

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 在集群列表页面,选择要设置的集群,单击管理。

容器服务		集群列表					您最多	可以创建 5	个集群,每个集群最多	可以添加 20 个节	点刷新	创建集群	-
Swarm	Kubernetes	常见问题: 8 如何创建集群 8 如何添	动已有云服务器	8	第可用区节点管理 🔗 集成日:	志服务 🔗 ;	通过Docker客户	『端连接集群	ŧ				
<b>職</b> 逸 应用		名称 🔻											
服务		集群名称/ID	集群类型	也域	网络类型	集群状态	节点状态 2	节点个 数	创建时间	Docker版本	2		操作
集群 节点	1	routing-test-online	阿里云集 <u>4</u> 群 1	¥东 L	虚拟专有网络 vpc- bp1vvet9csb16vac2jqoc	●运行 中	健康 ℃	2	2017-12-04 15:26:27	17.06.2-ce	管理	查看日志 │ 监控 │ ↓	删除 更多∓

4. 单击左侧导航栏中的节点伸缩,单击请新建自动伸缩规则。

<	集群 : routing-test-online											
基本信息 负载均衡	集群ID: cd142dad866c	集開D: cd142dad866c1414aacd994eddb2878d3 ●运行中 地域:华东1										
用户标签	伸缩规则:											
镜像列表	<ul> <li>请新建伸缩规则</li> <li>注意:1.节点缩容只会对通</li> </ul>	过节点扩容创建	出来的节点进行,用户手工创。	建或者添加的节	点不受影响。	2. 节点缩容的时候,	系统会删除集群里的ECS,)	电户需要提前做好	数据备份。			
容器列表 集群事件	节点列表											
节点伸缩 1	IP地址(ID)	实例类型	实例ID/名称	状态	容器数目	配置	操作系统	Docker版本	Agent	撮作		
	0.00.00.00	阿里云ECS	of Columbia (10.4)	●正常	8	CPU:2核 内存:3.859 GB	Ubuntu 16.04.2 LTS	17.06.2-ce	0.10-f574e56	监控   更多▼		
	10, 14, 21, 29, 8 11, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20	阿里云ECS	10000-04000-0400-0400-0400-0400-0400-04	●正常	8	CPU : 2核 内存 : 3.859 GB	Ubuntu 16.04.2 LTS	17.06.2-ce	0.10-f574e56	监控   更多+		

5. 配置伸缩规则,并单击下一步。

约束规则:

- 扩容条件的可选范围是 50%~100%, 缩容条件的可选范围是 0%~50%。
- 扩容条件和缩容条件的差值不能小于30%。
- 扩容步长的可选范围是 1~5, 缩容步长目前默认是 1, 不支持配置。
- 设置集群最小节点数和集群最大节点数。缩容时,如果节点数 ≤ 集群最小节点数,不会进行 缩容操作;扩容时,如果节点数 ≥ 集群最大节点数,不会进行扩容操作。



请谨慎设置伸缩策略。如果在您设置伸缩规则的时候,集群就满足所设置的伸缩条件而且伸缩 后集群仍然满足伸缩条件,那么监控将会不断地触发伸缩。

配置节点伸缩规则   +返回节点伸缩	
伸缩监测指标	选择实例规格
监控指标 (CPU或内存只能选择一种作为伸缩监控指标) ≥ 根据监控指标自动伸缩	
CPU使用量	内存使用量
<ul> <li>☞ 扩容条件: CPU使用量 &gt; 70</li> <li>% (可选范围: 50-100, 缩容和扩容的差值不能小于30%)</li> <li>■ 缩容条件: 未设置</li> </ul>	<ul> <li>「扩容条件:未设置</li> <li>/ 缩容条件:未设置</li> </ul>
扩容步长:	
步长: 1 (可选范围:1-5)	
限定条件	
集群最小节点数: 2 (可选范围:1-10) 集群最大节点数: 10 (可选范围:1-20,最大节点数必须大于最小节点数)	
	返回 下一步

6. 配置实例规格,并单击确认配置。

当您设置了扩容条件时,您需要设置扩容出来的节点的规格。有关实例规格配置的信息,参见创 建集群。 此外,您还可以设置是否将扩容出来的节点的 IP 添加到 RDS 实例的白名单中,方便 ECS 实例 访问 RDS 实例。

📕 说明:

您仅能将 ECS 实例的 IP 添加到位于同一地域的 RDS 实例的白名单中。

单击请选择您想要添加白名单的RDS实例,在弹出的对话框中选择所需的 RDS 实例并单击确定。

IIII IIII IIII RDS白名单: 请选择你想要添加白名单的RDS实例

### 查看已创建的节点伸缩规则

您可以查看已创建的节点伸缩规则。

- 1. 登录 容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 3. 在集群列表页面,选择要设置的集群,单击管理。
- 4. 单击左侧导航栏中的节点伸缩,即可以看到已创建的节点伸缩规则。

集群: test							刷新		
集群ID:c7f 地域:44东1									
自动伸缩:	自动伸缩:								
监控指标	伸缩步长	伸缩范围		实例规格		操作			
CPU使用量: 下限: 10 % 内存使用量: 未设置	扩容步长: 1	集群最小节点数 集群最大节点数	: 1 : 10	1 核 1GB ( ecs.n1.tiny )		修改	删除		

您可以单击修改修改节点伸缩规则或者单击删除删除节点伸缩规则。

### 查看监控指标

- 1. 在 Swarm 菜单下,单击左侧导航栏中的集群。
- 2. 在集群列表页面,选择要设置的集群,单击监控。

容器服务		集群列表				您最多	≶可以创建!	5 个集群,每个集群最	多可以添加 20 个节	点 刷新	创建集群	¥ -
Swarm	Kubernetes	常见问题: ②如何创建集群 ③如	叩何添加已有云服务器	跨可用区节点管理		)通过Docker客/	□	詳				
应用		名称 ▼										
服务		集群名称/ID	集群类型 地域	或 网络类型	集群状态	节点状态 2	节点个 数	创建时间	Docker版本		2	操作
集群 节点	1	routing-test-online	阿里云集 华线 群 1		●运行 :2jqoc 中	健康 오	2	2017-12-04 15:26:27	17.06.2-ce	管理	查看日志 监控	删除 更多→

进入云监控管理控制台,您可以看到集群的监控信息。

📋 说明 :

如果没有监控数据,查看监控服务(acsmonitoring)是否正确安装,如果异常,请重新启动或 部署。查看集群 Agent 是否为最新版本,监控服务(acsmonitoring)是否为最新版本;如果不 是,请进行升级。



3. 单击左侧导航栏中的容器服务,查看集群列表,单击查看全部规则。

云监控	容	器服务集群列表									€ 刷新
▼ 云服务监控	请辅	认进行查询			搜索		应用分组				
云服务器ECS		集群名称	运行状态	网络	(	区域	监控				操作
云数据库RDS版		k8s-test	运行中	VPC	4	华东 1	节点监控	服务监控	容器监控	监控图表	报警规则
负载均衡		routing-test-online	运行中	VPC	đ	华东 1	节点监控	服务监控	容器监控	监控图表	报警规则
对象存储OSS CDN		批量设置报警	查看全部规则	2							
弹性公网IP											
云数据库Redis版											
流计算											
容器服务 1											

4. 在报警规则列表页面,可以看到弹性伸缩自动设置的报警规则。

#	容器服务 报警规则列表	t 返回实例列表							ł
E	所屬实例	监控项 (全部) 👻	规则描述	统计周期	通知对象	状态 (全部) 👻	启用	摄	PTE
e	c4a766825b2574e eab85d0361c2153 00f	公网网络入流量	如果公闲网络入泥里 干均	5分钟	云账号报警联系人 查響	告양	已启用	报警历: 修改   萘用   删	史除
e	c4a766825b2574e eab85d0361c2153 00f	CPU使用率	如果CPU使用至 干均酱	5分钟	云账号报警联系人 宣書	正常	已启用	报替历: 修改   萘用   删	史除
	启用 智停	删除					共2条 10 ▼ «	<1 $>$ $>$	

如果没有监控报警规则,请在集群中更新访问授权信息(在集群列表页面,选择所需的集群,单 击更多>更新访问控制授权信息)。更新前需要先开通 RAM;否则会报错。

错误提示		×
8	Aliyun API Error: RequestId: 51C6F096-5A42-41D2-9FC3-16472F34A45A Status Code: 4 04 Code: Inactive Message: Account is inactive to this service	
	确定	

 选中一个报警规则,您可以修改报警条件及通知人(支持短信,邮件等通知方式),也可以禁用 报警规则。

容(	醫服务 报警规则列表	t 返回实例列表						€ 問新
	所屬实例	监控项 (全部) 👻	规则描述	统计周期	通知对象	状态 (全部) 👻	启用	操作
	c4a766825b2574e eab85d0361c2153 00f	公网网络入流量	如果公司网络入流量 干约	5分钟	云账号报警联系人 查看	# <b>W</b>	日日	报警历史 修改   禁用   删除
۲	c4a766825b2574e eab85d0361c2153 00f	CPU使用率	如果GPU晚用些 平均值	5分钟	云账号报警联系人 重着	正常	已启 用	报警历史 修改   萘用   删除
J	自用 暫停	删除					共2条 10 • «	< 1 > >

### 错误排查

如果您设置的节点弹性伸缩不起作用,可以参考弹性伸缩不起作用怎么办来进行错误的排查。

# 2.15.7 监控指标

容器服务提供多维度的监控服务。您可以通过容器服务管理控制台或阿里云云监控管理控制台查看监控信息。

容器服务的监控指标分为自动监控指标和云监控插件指标两部分。通过监控容器服务的 CPU 使用 率、内存、网络、I/O、异常等指标,帮助您了解容器服务的使用情况。您可以登录云监控控制台并 进入容器服务页面查看监控详情,还可对监控项设置报警规则,以便指标异常时收到报警通知。 容器服务提供以下维度的监控信息:

- 集群
- 节点
- 服务
- 容器

〕 说明:

文档版本: 20180808

您需要在创建集群时安装云监控插件,才能获取集群和节点的内存指标,参见创建集群。如果 想查看 ECS 实例更详细的指标,前往云监控控制台查看云服务器 ECS 的监控指标,参见监控 项说明。

### 监控指标说明

### 集群监控项

监控项	维度	单位	最小监控粒度	聚合方式
CPU 使用率	集群维度	百分比	60秒	Average
网络入流量	集群维度	bit/s	60秒	Average
网络出流量	集群维度	bit/s	60秒	Average
内存使用率	集群维度	百分比	60秒	Average
GPU 内存使用量	集群维度	bytes	60秒	Average
GPU 使用率	集群维度	百分比	60秒	Average
GPU 温度	集群维度	摄氏度	60秒	Average

### 节点监控项

监控项	维度	单位	最小监控粒度	聚合方式
CPU 使用率	节点维度	百分比	60秒	Average
网络入流量	节点维度	bit/s	60秒	Average
网络出流量	节点维度	bit/s	60秒	Average
内存使用率	节点维度	百分比	60秒	Average
GPU 内存使用量	节点维度	bytes	60秒	Average
GPU 使用率	节点维度	百分比	60秒	Average
GPU 温度	节点维度	摄氏度	60秒	Average

### 服务监控项

监控项	维度	单位	最小监控粒度	聚合方式
CPU使用率	服务维度	百分比	60秒	Average
网络流入速率	服务维度	bytes/s	60秒	Average
网络流出速率	服务维度	bytes/s	60秒	Average

监控项	维度	单位	最小监控粒度	聚合方式
服务 IO 读	服务维度	bytes	60秒	Average
服务 IO 写	服务维度	bytes	60秒	Average
内存使用量	服务维度	byte	60秒	Average
内存使用率	服务维度	百分比	60秒	Average
网络流入流量	服务维度	bytes	60秒	Average
网络流出流量	服务维度	bytes	60秒	Average
容器 IO 读速率	容器维度	bytes/s	60秒	Average
容器 IO 写速率	容器维度	bytes/s	60秒	Average
服务 GPU 内存使 用量	服务维度	bytes	60秒	Average

### 容器监控项

监控	维度	单位	最小监控粒度	聚合方式
CPU 使用率	容器维度	百分比	60秒	Average
网络流入速率	容器维度	bytes/s	60秒	Average
网络流出速率	容器维度	bytes/s	60秒	Average
10 读	容器维度	bytes	60秒	Average
10 写	容器维度	bytes	60秒	Average
内存使用率	容器维度	百分比	60秒	Average
内存使用量	容器维度	byte	60秒	Average
网络流入流量	容器维度	bytes	60秒	Average
网络流出流量	容器维度	bytes	60秒	Average
容器 IO 读速率	容器维度	bytes/s	60秒	Average
容器 IO 写速率	容器维度	bytes/s	60秒	Average
GPU 内存使用量	容器维度	bytes	60秒	Average

## 注意事项

你需要前往云监控控制台查看历史监控数据或设置监控报警规则。
- 监控数据最多保存31天。
- 用户最多可连续查看14天的监控数据。
- 支持设置批量报警规则。

# 2.15.8 弹性伸缩不起作用怎么办

如果您创建的容器弹性伸缩或节点弹性伸缩不起作用,您可以按照以下内容来进行问题排查。 查看监控指标,确认监控指标有数据并达到阈值一段时间。

- 查看容器弹性伸缩的监控指标
  - 1. 单击左侧导航栏中的服务。
  - 2. 选择服务所在的集群。
  - 3. 选择所需的服务并单击监控。

页面跳转到云监控管理控制台,您可以查看容器的监控信息。你需要确认监控指标有数据并 达到阈值一段时间。

# 📋 说明:

容器服务在判断监测指标是否超出所设定的上下限时,使用的是采集周期(一分钟)内监测指标的平均值(即 CPU 平均使用量和内存平均使用量),而且只有当连续三个采集周期内的监测指标平均值均超出所设定的上下限时,容器服务才会触发扩容或伸缩操作,以避免因为监控数据抖动而引起频繁的扩容或缩容操作。



• 查看节点弹性伸缩的监控指标

1. 单击左侧导航栏中的集群。

2. 选择所需的集群,单击监控。

进入云监控管理控制台,您可以看到集群的监控信息。你需要确认监控指标有数据并达到 阈 值一段时间。



容器服务在判断监测指标是否超出所设定的上下限时,使用的是采集周期(一分钟)内监测指标的平均值(即 CPU 平均使用量和内存平均使用量),而且只有当连续三个采集周期内的监测指标平均值均超出所设定的上下限时,容器服务才会触发扩容或伸缩操作,以避免因为监控数据抖动而引起频繁的扩容或缩容操作。



如果没有监控数据

您需要查看监控服务(acsmonitoring)是否正确安装(容器状态及数量是否正确),查看集群 Agent 是否为最新版本,监控服务(acsmonitoring)是否为最新版本;如果不是,请进行升级。参 见升级系统服务 和升级 Agent。

查看监控报警规则及状态,确认规则生成且状态正确

1. 在云监控管理控制台,单击左侧导航栏中的云服务监控>容器服务。

您可以看到集群列表。

容器服务集群列表								℃刷新
请输入进行查询		搜索	应用分组					
□ 集群名称	运行状态 网络	区域	浙拉					操作
swarm-test	运行中 经典网	络 华东1	节点监控	服务监控	容器监控	监控图表	Ħ	<b>長警规则</b>
i route	运行中 VPC	华东 1	节点监控	服务监控	容器监控	监控图表	H	反警规则
routing-test-online	运行中 VPC	华东 1	节点监控	服务监控	容器监控	监控图表	Ħ	侵警规则
	晉全部规则							

2. 单击查看全部规则,可以看到弹性伸缩自动设置的报警规则。

报警	规则列表								€ 刷新	创建报	警規則
请输	入进行查询			搜索							
	规则名称	状态 (全部) 🔻	启用	监控项 (全部) ▼	維度	报警规则	产品名称 (容器服务) 🔻	通知对象			操作
		♥正常状态	已启用	CPU使用率	serviceId:jenkins_jenkins, clust erId:c27f25f087e764fefa780835 b58a8cbea	1分钟 CPU使用率 平均值>70 % 连续 3 次 则报警	容器服务	查看	修改	て  禁用	查看 删除
		●正常状态	已启用	ClusterCPUUtilization	cluster1d:c27f25f087e764fefa78 0835b58a8cbea	1分钟 平均值>70 连续 3 次 则报警	容器服务	云账号报警联系人 查看	修己	な  禁用	查看   删除
		●报警状态	已启用	ClusterCPUUtilization	clusterId:c27f25f087e764fefa78 0835b58a8cbea	1分钟 平均值<30 连续 3 次 则报警	容器服务	云账号报警联系人 查看	修改	な  禁用	查看   删除
	启用	<b>禁用 删除</b>						共 3条 10	▼ « <		>

### 如果没有监控报警规则

您需要在集群中更新访问授权信息(在集群列表页面,选择所需的集群,单击更多>更新访问控制授权信息)。更新前需要先开通 RAM;否则会报如下所示的错误。

11 庆远小		
	Aliyun API Error: Requestid: 51CbF095-5A42-41D2-9FC3-16472F34A45A Status Code: 4 04 Code: Inactive Message: Account is inactive to this service	
	确定	

您需要查看监控服务(acsmonitoring)是否正确安装(容器状态及数量是否正确),查看集群
 Agent 是否为最新版本,监控服务(acsmonitoring)是否为最新版本;如果不是,请进行升级。
 参见升级系统服务和升级 Agent。

查看报警历史

如果报警规则列表里显示某个报警规则的状态不正常(为报警状态),您可以查看该规则的报警历 史以便定位问题。

选择所要查看的报警规则,并单击右侧的查看。

报警	规则列表								₿別新	创建报	警規則
请输	入进行查询			搜索							
	规则名称	状态 (全部) 🔻	启用	监控项 (全部) 👻	维度	报警规则	产品名称 (容器服务) ▼	通知对象			操作
		♥正常状态	已启用	CPU使用率	serviceId:jenkins_jenkins, clust erId:c27f25f087e764fefa780835 b58a8cbea	1分钟 CPU使用率 平均值>70 % 连续 3 次 则报警	容器服务	查看	修己	女  禁用	查看   删除
		♥正常状态	已启用	ClusterCPUUtilization	clusterId:c27f25f087e764fefa78 0835b58a8cbea	1分钟 平均值>70 连续 3 次 则报警	容器服务	云账号报警联系人 查看	修改	女  禁用	查看   删除
		●报警状态	已启用	ClusterCPUUtilization	clusterId:c27f25f087e764fefa78 0835b58a8cbea	1分钟 平均值<30 连续 3 次 则报警	容器服务	云账号报警联系人 查看	修改	女  蔡用	查看   删除
	启用 募	制除						共 3条 10	▼ ≪ <	1 >	>

单击报警历史。您可以选择时间来查看指定时间段内发生的报警。

基本信息	报警历史						禁用	修改 删除
				11/141 21/141 41/141 61/141	12小时 1天	沃 仄	2017-08-23 18:37:21 - 2017-0	8-30 18:37:2
产品类型	故障资源	发生时间	持续时间	规则名称	通知方式	状态	通知对象	操作
容器服务	实例: clusterId:c27f25f087e764f efa780835b58a8cbea	2017-08-30 18:35:12	12分钟12秒	alert_c27f25f087e764fefa780835b58a8cbeaClusterCPUUtilization_scaleIn	邮箱,旺旺	通道沉默		展开
容器服务	实例: clusterId:c27f25f087e764f efa780835b58a8cbea	2017-08-30 18:34:12	11分钟12秒	alert_c27f25f087e764fefa780835b58a8cbeaClusterCPUUtilization_scaleIn	邮箱,旺旺	报警发生	云账号报警联系人	展开
容器服务	实例: clusterId:c27f25f087e764f efa780835b58a8cbea	2017-08-30 18:26:12	3分钟12秒	alert_c27f25f087e764fefa780835b58a8cbeaClusterCPUUtilization_scaleIn	邮箱,旺旺	通道沉默		展开
容器服务	实例: clusterId:c27f25f087e764f efa780835b58a8cbea	2017-08-30 18:24:12	1分钟12秒	alert_c27f25f087e764fefa780835b58a8cbeaClusterCPUUtilization_scaleIn	邮箱,旺旺	报警发生	云账号报警联系人	展开

# 2.17 DevOps

# 2.17.1 基于 Jenkins 的持续交付

持续集成作为敏捷开发重要的一步,其目的在于让产品快速迭代的同时,尽可能保持高质量。每一次代码更新,都要通过自动化测试来检测代码和功能的正确性,只有通过自动测试的代码才能进行 后续的交付和部署。本文主要介绍如何将时下最流行的持续集成工具之一的 Jenkins 结合阿里云容 器服务,实现自动测试和镜像构建推送。

以下内容演示如何通过阿里云容器服务 Jenkins 实现自动测试和 Docker 镜像构建,实现高质量的 持续集成。

# 背景信息

每次代码提交到 GitHub 上的 nodejs 的项目后,阿里云容器服务 Jenkins 都会自动触发单元测试,测试通过则继续镜像构建及推送到目标镜像仓库中,最后邮件通知结果。

大致流程如下图所示:



slave-nodejs 用于进行单元测试,构建镜像和推送镜像的 slave 节点。

### Jenkins 相关介绍

Jenkins 是基于 Java 开发的一种开源持续集成工具,监控并触发持续重复的工作,具有开源,支持 多平台和插件扩展,安装简单,界面化管理等特点。Jenkins 使用 job 来描述每一步工作。节点是 用来执行项目的环境。Master 节点是 Jenkins job 的默认执行环境,也是 Jenkins 应用本身的安装 环境。

#### master/slave

master/slave 相当于 server 和 agent 的概念。master 提供 web 接口让用户来管理 job 和 slave, job 可以运行在 master 本机或者被分配到 slave 上运行。一个 master 可以关联多个 slave 用来为 不同的 job 或相同的 job 的不同配置来服务。

可以通过配置多个 slave 为不同项目准备单独的测试和构建环境。

说明:

本文中提到的 Jenkins job 和项目均指的是 Jenkins 一个构建单元,执行单元。

#### 步骤 1 使用容器服务部署 Jenkins 应用和 slave 节点

不同应用构建、测试所需要的依赖不同,最佳实践是使用包含相应的运行时依赖和工具的不同 Slave 容器执行测试和构建。通过阿里云容器服务提供的针对 Python、Node.js、Go 等不同环境的 Slave 镜像以及示例模板,用户可以简单快速地生成 Jenkins 应用以及各种 slave 节点,在 Jenkins 应用中配置节点信息,然后在构建项目中指定执行节点,从而实现整个持续集成流程。 送明:

有关阿里云容器服务提供的开发 slave 节点镜像,参见 https://github.com/AliyunContainerService /jenkins-slaves。

# 1.1 创建 Jenkins 编排模版

新建模版,以如下内容为基础,创建编排。

阿里云容器服务 Jenkins master 支持标签 1.651.3、2.19.2、2.32.2。

📋 说明:

有关如何创建编排模板,参见创建编排模板。

```
jenkins:
    image: 'registry.aliyuncs.com/acs-sample/jenkins:1.651.3'
    volumes:
        - /var/lib/docker/jenkins:/var/jenkins_home
   restart: always
    labels:
        aliyun.scale: '1'
        aliyun.probe.url: 'tcp://container:8080'
        aliyun.probe.initial_delay_seconds: '10'
        aliyun.routing.port_8080: jenkins
    links:
        - slave-nodejs
slave-nodejs:
    image: 'registry.aliyuncs.com/acs-sample/jenkins-slave-dind-nodejs
    volumes:
        - /var/run/docker.sock:/var/run/docker.sock
    restart: always
    labels:
        aliyun.scale: '1'
```

#### 1.2 使用模板创建 Jenkins 应用和 slave 节点

使用上边新建的编排模板或者直接使用阿里云容器服务提供的 Jenkins 示例模版创建 Jenkins 应用和 slave 节点。



有关如何使用编排模板创建应用,参见创建应用。

编排列表	示例编排 我的编排			创建
		搜索		
<b>0</b> 0	<b>gitlab</b> 详情	說規 postgresq gitlat redit	: 镜像 : registry.aliyuncs.com/acs-sample/sameersbn-postgresql:9.4-3 : registry.aliyuncs.com/acs-sample/sameersbn-gitlab:7.14.1 : registry.aliyuncs.com/acs-sample/sameersbn-redis:latest	创建应用→
<b>0</b> 0	jenkins 详情	服费 jenkins	: 镜像 : registry.allyuncs.com/acs-sample/jenkins:latest	创建应用→

# 创建成功后, Jenkins 应用和 slave 节点显示在服务列表中。

应用: jenki	ins											刷新
基本信息												
应用名称:	应用名称: jenkins					2016-06-30	更新时间: 2016-06-30	所在集群: test				
触发器 1.	每种类型的触发	<b> 这器只能创</b>	建1个 2.关于	于资源伸缩类型	≥的触发器②					ê	別建触发器	^
服务列表	容器列表	日志	事件									
服务名称	所	属应用	服务状	态容	器状态	镜像						操作
jenkins	jenkins ●就绪		<b>运</b> 停	行:1 止:0	registry.aliyuncs.com/acs-sample/jenkins:latest			停止	变更配置	删除	事件	
slave-nodejs	jen	kins	- 就绪	<b>运</b> 停	;行:1 ;止:0	registry.aliyuncs.com/acs-s	sample/jenkins-slave-d		停止	変更配置	删除	事件

# 打开容器服务提供的访问端点,就可以使用刚刚部署的 Jenkins 应用。

服务: j€	š: jenkins_jenkins									
基本信	息									
服务名	称: jenkin	าร	所	在应用: jenkins	s 镜像:	registry.aliyuncs.com/acs-sample/jenl	容器数目: 1	●就绪		
访问端。	点: http://	/jenkin	ns.cc7d3829f003f	lea78046a4f279	9b59c03.cn-beijing.alicon	tainer.com				
访问端」 容器	点: http:// 日志	/jenkin 配3	ns.cc7d3829f003f 置  事件	lea78046a4f27	9b59c03.cn-beijing.alicon	tainer.com				
访问端, 容器 名称VID	点: http:// 日志	/jenkin 配3	ns.cc7d3829f003f 置 事件 状态	<b>lea78046a4f27</b> 健康检测	9b59c03.cn-beijing.alicon 镜象	talner.com	容器IP	节点IP		操

## 步骤2实现自动测试及自动构建推送镜像

2.1 将 slave 容器配置成 Jenkins 应用的 slave 节点。

打开 Jenkins 应用,进入系统设置界面,选择管理节点,新建节点,配置相应参数。如下图所示。

	slave-nodejs-ut	
描述	slave-nodejs-ut	
# of executors	s 1	(
远程工作目录	/home/jenkins	
标签	slave-nodejs-ut	
用法	尽可能的使用这个节点	
启动方法	Launch slave agents on Unix machines via SSH	(
	Host 172.19.0.4	
	Credentials jenkins/***** 🗘 🖝 Add 🗸	
		高级
Availability	Keep this slave on-line as much as possible	
Node Properti	ies	
Environme	nt variables	
Tool Location	ons	



- 标签是 slave 的唯一标识。
- slave 容器和 Jenkins 容器同时运行在阿里云平台上,因此需要填写外网访问不到的容器节点 IP ,隔离测试环境。
- 添加 Credential 的时候,使用创建 slave-nodejs 镜像的 Dockerfile 里的 jenkins 用户(初始密码为 jenkins)。镜像 Dockerfile 的地址为 https://github.com/AliyunContainerService/jenkins-slave-dind-nodejs。

# 2.2 创建项目实现自动化测试

- 1. 新建 Item,选择构建一个自由风格的软件项目。
- 2. 填写项目名称,并选择项目运行节点。此示例中填写上述准备的 slave-nodejs-ut 节点。

项目名称	Ŷ	nodejs-ut	
□ 丢弃	旧的构建	[Plain text] <u>我竟</u>	
GitH Project	ub project url	https://github.com/qinyujia/containerops/	<b>@</b>
<ul> <li>参数</li> <li>关闭</li> <li>在必</li> </ul>	化构建过程 构建 (重新开启 要的时候并发	自构建前不允许进行新的构建) 构建	0 0 0
Rest Label E	rict where this xpression	s project can be run slave-nodejs-ut Littett is surviced by 1 node	0

3. 配置源码管理和代码分支。此示例中源代码使用 GitHub 管理。

源码管理		
<ul> <li>None</li> <li>CVS</li> <li>CVS Projectset</li> </ul>		
Git Repositories	Repository URL https://github.com/qinyujia/containerops.git Credentials 756	
Branches to build	高级 Add Repository Delete Repository Branch Specifier (blank for any') ・/jenkins-test Add Branch Delete Branch	]0

4. 配置构建触发器,此示例采用结合 GitHub Webhooks & services 实现自动触发项目执行。



5. 在 GitHub 中添加 Jenkins 的 service hook,完成自动触发实现。

在 GitHub 项目主页单击Settings,单击左侧菜单栏中的Webhooks & services,单击Add Service,在下拉框中选择Jenkins(Git plugin)。在Jenkins hook url 对话框中填写\${ Jenkins IP}/github-webhook/,例如:

http://jenl -webhook/	<pre><ins.cd****************.cn-beijing.alicontainer.com github<="" pre=""></ins.cd****************.cn-beijing.alicontainer.com></pre>
♀     qinyujia / containerops forked from ringtal/containerops       ⇔     Code     ↓     Pull requests     0	O Unwatch ▼     1     ★ Star     0     ¥ Fork     1       Image: Wiki → Pulse International State     Image: Settings     Image: Settings     Image: Settings
Options Collaborators Branches Webhooks & services Deploy keys	Services / Add Jenkins (GitHub plugin)         Jenkins is a popular continuous integration server.         Using the Jenkins GitHub Plugin you can automatically trigger build jobs when pushes are made to GitHub.         Install Notes         1. *Jenkins Hook Ud* is the URL of your Jenkins server's webhook endpoint. For example: <a href="http://ci.jenkins-ci.org/github-webhook/">http://ci.jenkins-ci.org/github-webhook/</a> .         For more information see <a href="http://wiki.jenkins-cl.org/display/JENKINS/GitHub+plugin">http://ci.jenkins-ci.org/display/JENKINS/GitHub+plugin</a> .         Jenkins hook unt         http://jenkins.cd1411         Mattive         We will run this service when an event is triggered.         Add service

6. 增加 Execute shell 类型的构建步骤,编写 shell 脚本执行测试。

构建		
Execute	shell	(
Command	pwd ls cd chapter2 npm test	
	See the list of available environment variables	
		删除

本示例中的命令如下所示:

pwd	
ls	
cd (	chapter2
npm	test

#### SVN 代码源示例:

在源码管理模块选择**Subversion**。**Repository URL** 中填入 SVN repository 的地址。(如果 Jenkins master 时区和 SVN 服务器时区不一致,请在 repository 地址末尾添加 @HEAD) **Credentials** 中添加 SVN 服务器的用户名和密码。

源码管理		
CVS CVS		
<ul> <li>Git</li> </ul>		
Repositories	Repository URL https://github.com/qinyujia/containerops.git	
	Credentials 756	
	高毀	
	Add Repository Delete Repository	
Branches to build	Branch Specifier (blank for any') •/jenkins-test	0
	Add Branch Delete Branch	

配置构建触发器,此示例采用 Post-commit hook 实现自动触发项目执行。在身份验证令牌中填写

# 您设置的 token。

构	建触发器		
-	Build after other pr	ojects are built	0
F	Projects to watch	nodejs-ut	
		O Trigger only if build is stable	
		Trigger even if the build is unstable	
		<ul> <li>Trigger even if the build fails</li> </ul>	
	Build periodically		C
	Build when a chan	ge is pushed to GitHub	0
	Poll SCM		C

登录 SVN 服务器,在代码 repository (svn-java-demo)的 hooks 文件目录下创建 post-

### commit 文件。

```
cd /home/svn/svn-java-demo/hooks
cp post-commit.tmpl post-commit
chmod 755 post-commit
```

在 post-commit 文件中添加

命令。例如:

```
curl -u test:test
http://127.0.0.1:8080/jenkins/job/svn/build?token=qinyujia
```

0

#### 2.3 创建项目实现自动构建,推送镜像。

- 1. 新建 Item,选择构建一个自由风格的软件项目。
- 2. 填写项目名称,并选择项目运行节点。此示例中填写上述准备的 slave-nodejs-ut 节点。
- 3. 配置源码管理和代码分支。此示例中源代码使用 GitHub 管理。

4. 添加如下触发器,设置只有在单元测试成功之后才执行自动构建镜像。

构建触发器		
<ul> <li>Build after other p</li> </ul>	rojects are built	0
Projects to watch	nodejs-ut	
	Trigger only if build is stable	
	O Trigger even if the build is unstable	
	O Trigger even if the build fails	
<ul> <li>Build periodically</li> </ul>		0
<ul> <li>Build when a cha</li> </ul>	nge is pushed to GitHub	•
Poll SCM		•

5. 编写构建镜像和推送镜像的 shell 脚本。

构建		
Execute	shell	?
Command	cd chapter2 docker build -t registry.aliyuncs.com/ginyujia-test/nodejs-demo . docker login -u disersitivani cossilization -p (disersiti registry.aliyuncs.com docker push registry.aliyuncs.com/ginyujia-test/nodejs-demo	
	See the list of available environment variables	
	<b>翻除</b>	

本示例的命令如下所示:

```
cd chapter2
sudo docker build -t registry.aliyuncs.com/qinyujia-test/nodejs-
demo .
sudo docker login -u ${yourAccount} -p ${yourPassword} registry.
aliyuncs.com
sudo docker push registry.aliyuncs.com/qinyujia-test/nodejs-demo
```

#### 步骤3自动重新部署应用

3.1 首次部署应用

使用编排模板,将 2.3 中创建的镜像部署到容器服务中,创建 nodejs-demo应用。

示例如下所示:

#### 3.2 自动重新部署

1. 选择刚刚创建的应用 nodejs-demo, 创建触发器。

📋 说明:

有关如何创建触发器,参见触发器。

触发器				^
触发器链接(鼠标滑过复制)	secret(鼠标滑过复制)	类型	操作	
https://cs.console.aliyun.com/hook/trigger?triggerUt=Y2Y5ZTA3YmZhMDFINTFIIZDhIZGI5NzZiMWU2YTU3ZJY1fG5vZ G 36ec4139bb78907	4138683246566e4d474c4a4d336a4 844 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	重新部署	删除触发器	

2. 在2.3 中的 shell 脚本中添加一行,地址即为上文创建的触发器给出的触发器链接。

```
curl 'https://cs.console.aliyun.com/hook/trigger?triggerUrl=***==&
secret=***'
```

3. 把 2.3 示例中的命令改为:

```
cd chapter2
sudo docker build -t registry.aliyuncs.com/qinyujia-test/nodejs-
demo .
sudo docker login -u ${yourAccount} -p ${yourPassword} registry.
aliyuncs.com
sudo docker push registry.aliyuncs.com/qinyujia-test/nodejs-demo
curl 'https://cs.console.aliyun.com/hook/trigger?triggerUrl=***==&
secret=***'
```

到此,镜像推送之后, Jenkins 会自动触发重新部署 nodejs-demo 应用。

#### 步骤4配置邮件推送结果

如果希望单元测试或者镜像构建的结果能够通过邮件推送给相关开发人员或者项目执行发起者。可以通过如下配置来实现。

1. 在 Jenkins 主页,选择系统管理,系统设置,配置 Jenkins 系统管理员邮箱。

# Jenkins Location

Jenkins URL

http://jenkins.cs.aliyun-

系统管理员邮件地址

jenkins-cs@alibaba-in

**2.** 安装 Extended Email Notification plugin, 配置 SMTP server 等相关信息,并设置默认邮件接收 人列表。如下图所示。

Extended E-mail Notification		
SMTP server	smtp.alibaba-inc.com	
Default user E-mail suffix		
Use SMTP Authentication		(
User Name	jenkins-cs@alibaba-inc.com	
Password		
Use SSL	۵	(
SMTP port	465	(
Charset	UTF-8	
Default Content Type	Plain Text (text/plain)	\$
Use List-ID Email Header		(
Add 'Precedence: bulk' Email Header		(
Default Recipients	with alibaba-inc.com	

以上是 Jenkins 应用系统参数设置,下面是为需要用邮件推送结果的 Jenkins 项目进行相关配置。

**3.** 在 Jenkins 项目中添加构建后操作步骤。选择 Editable Email Notification 类型,填写邮件接收人 列表。

增加构建步骤 ▼		
构建后操作		
Editable Email Notification Disable Extended Email Publisher		•
Project Recipient List	Allows the user to disable the publisher, while maintaining the settings	
	y alibaba-inc.com	0

4. 添加邮件发送触发器。

Triggers	llways	0	)
Ser	nd To	② 删除	
	Requestor	<b>③</b> 納除	
	Developers	劉隆	0
	Add 👻	高级 Remove Trigger	

# 2.18 服务发现和负载均衡

# 2.18.1 概述

服务发现和负载均衡主要解决通信的可靠性问题。为了达到可靠性,容器服务引入了负载均衡机制。通信又可以分为对外暴露服务的通信和内部服务之间的通信。下面根据场景引导您使用不同的解决方案。

场景一

普通且简单的 7 层协议负载均衡, Web 服务的反向代理,推荐使用简单路由服务。更多详细信息,参见简单路由#支持 HTTP/HTTPS),简单路由-域名配置,简单路由-HTTP 协议变为 HTTPS 协议。

场景二

4 层协议的负载均衡,负载均衡直接负载均衡到多个相同功能的容器,在将传统架构迁移到容器架构过程中非容器集群的服务访问容器集群中容器的服务,推荐使用负载均衡路由。

场景三

同一个集群内,服务间需要相互发现和相互进行通信,且需要负载均衡的能力,推荐使用<u>集群内服</u>务间路由和负载均衡。

#### 场景四

同一个集群内,服务间需要相互发现和相互进行通信,但是不需要负载均衡的能力,推荐使用容器 间的互相发现。

场景五

对负载均衡和服务发现有较高的定制需求,例如需要支持泛域名,自定义错误页面,支持记录访问 日志,URL 参数值选择后端服务,自定义 HAProxy 配置文件等等,推荐使用自定义路由-使用手 册。更多详细信息,参见自定义路由-简单示例。

# 2.18.2 简单路由 (支持 HTTP/HTTPS)

适用场景

普通且简单的 7 层协议负载均衡, Web 路由服务, 容器集群内服务之间 7 层协议互相访问的通信 代理和负载均衡。

#### 原理

如下图所示,当您新建一个集群的时候,会默认给这个集群分配一个负载均衡实例。该负载均 衡实例会将集群中的所有节点加入作为后端,同时前端会暴露 80 端口,后端所有节点的机器 会暴露 9080 端口。容器服务会启动一个路由应用 acsrouting,即阿里云容器服务路由应 用(Alibaba Cloud Container Service Routing)。该路由应用只有一个服务,即路由服务。该服 务是全局(global)的,即每个节点(下面说到的主机和节点都是同一个意思,即 ECS 的 vm 实 例)都部署了该服务(或者说镜像)的一个拷贝,也就是容器。每个节点都由这个容器用来路由 HTTP 服务或者 HTTPS 服务。



如上图所示,HTTP 服务,负载均衡实例的前后端端口的映射为 80:9080,主机与路由容器之间的端口映射为 9080:80,即路由的容器暴露 80 端口,其它用作 Web 服务的容器可以暴露任意的端口。只要在容器启动的时候设置主机和容器端口的映射,routing 服务就能获取到相应的端口进行请求的路由。有关暴露 HTTP 服务的完整示例,参见通过镜像创建 Nginx。

## 设置方式

📕 说明:

请务必确保容器服务相关节点的内核参数 (/etc/sysctl.conf#中#如下配置值为 0#否则可能会

导致无法访问#

```
net.ipv4.conf.default.rp_filter = 0
net.ipv4.conf.all.rp_filter = 0
net.ipv4.conf.eth0.rp_filter = 0
```

通过容器服务管理控制台进行设置

通过 服务 > 变更配置 进行设置

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 单击左侧导航栏中的服务。
- 4. 选择所要暴露的服务所在的集群。
- 5. 选择所要暴露的服务(本示例中为 wordpress-web)并单击变更配置。

容器服务	服务列表					刷新
概览	常见问题: 🖉 如何暴露!	服务到Internet 🔗 给	暴雷公网的服务添加	加域名 🔗 将访问	协议从http修改为https 🔗 更改应用对外端口	
应用	集群: swarm-test	▼ 🗷 隐藏系统服	务 🛛 隐藏离线服务	序 □ 隐藏在线服务	ž	服务名 *
服务 1	服务名称 2	所属应用	服务状态	容器状态	镜像	操作
集群 节点	logtail	wordpress	●就绪	<b>就绪:1</b> 停止:0	registry.aliyuncs.com/acs-sample/logtail:yunqi	<u> 広</u> 控   停止   重新调度   交更配置   删除   重启   事件
四時 数据卷 配置项	mysql	wordpress	●就绪	<b>就绪:1</b> 停止:0	mysql:latest	<u> 监控</u>   停止   重新调度   交更配置   删除   重启   事件
<ul> <li>镜像与方案     镜像     </li> </ul>	wordpress-web	wordpress	●就绪	<b>就绪:1</b> 停止:0	registry.aliyuncs.com/acs-sample/wordpress:yunqi	3 <u>出控</u>   停止   重新调度   支更配置   删除   重启   事件

6. 在变更配置页面,配置主机和容器端口的映射,如下图所示。

主机端口为空,表示随机暴露一个主机的端口(暴露 HTTP/HTTPS 服务时,您可以不需要 知道主机暴露的具体端口是什么,可以使用 overlay 网络或者 VPC 网络来直接访问容器的端 口),容器端口为 80。您使用 wordpress-web 服务的 80 端口来提供 HTTP 服务,使用的协议 是 TCP 协议。

	主机端口		容器端口	协议
	e.g. 8080	>	80	/ тср 🔹 👄
	注意:主机端口不能设置为9080.2376.3376			
简单路由配置:	注意: 主机端口不能设置为9080,2376,3376 ● ① 如何暴露 HTTP 服务 容器端口	域名		

路由配置通过域名来暴露服务,须标明要暴露的端口,此处为 Web 服务 80 端口。域名字段 只填写了域名前缀,如果域名前缀为 XXX,获得的域名为 XXX.\$cluster_id.\$region_id .alicontainer.com 供测试使用。此处您获得的域名为 wordpress.cb668bde43 f054cd7bd515c8739f38310.cn-hangzhou.alicontainer.com。您也可以填写自己的 域名,需要添加解析到相应的负载均衡实例 IP。关于配置路由的容器端口和 HTTP 服务的域名 的详细信息,参见*routing*。

7. 配置完毕后,单击更新,然后返回服务列表,进入服务详情页,单击访问端点。

服务:wordpress_wordpres	s	服务:wordpress_wordpress 刷新 调整容器数量								
基本信息										
服务名称: wordpress		所在应	用: wordpress		镜像: wordpress:latest 容器数目				●就绪	
访问端点: http://wordpresscn-hangzhou.alicontainer.com										
容器 日志 配置	事件									
名称/ID 状态 健康检测		健康检测	镜像		容器IP 节点IP				操作	
wordpress_wordpr		正常	wordpress:latest sha256:dccaeccfb	80/tcp				删除   停止   监控	日志   远程终端	

8. 您可以成功访问 wordpress 欢迎页。



通过应用的模板编辑器进行设置

- 1. 登录容器服务管理控制台。
- 2. 登录容器服务管理控制台。
- 3. 单击左侧导航栏中的应用。
- 4. 选择目标应用所在的集群。
- 5. 选择目标应用(本示例中为 wordpress)并单击变更配置。

容器服务	应用列表					刷新 创建应用
概览	常见问题: 🔗 如何创建应用	⑦ 变更应用配置     ⑦ 简单器     ◎	的古靈婦发布策略			
应用 1	集群: swarm-test	▼ 🗷 隐藏系统应用 🗆 隐藏	离线应用 □ 隠藏石	王线应用		名称 ▼
服务	应用名称 2 描述	状态	容器状态	创建时间 🔺	更新时间 ▲	操作
集群	wordpress	●就绪	就绪:3	2017-09-01 10:23:22	2017-09-01 10:53:36	停止   交更配置   删除   重新部署
节点			停止:0			3
网络						

 在模板编辑器中,添加routing标签,定义相应的域名或者域名前缀。注意升级应用的版本,以 及确定是否拉取最新的 Docker 镜像,最后单击确定更新域名,如下图所示。

变更配置		$\times$
应用名称: <b>*</b> 应用版本:	wordpress 1.2 注意:提交配置变更需要您更新应用版本号,否则确定按钮无法点击	
应用描述:		
使用最新镜像:	●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●	
发布模式:	标准发布 🔻 🖉	
模板:	1 · wordpress-web:       ▲         2       image: registry.aliyuncs.com/acs-sample/wordpress         3 · ports:       ●         4       - 80         5 · volumes:       ●         6       - /acs/log/wordpress/:/var/log/         7 · labels:       ●         8       ●         9       ●         10       aliyun.routing.port_80: wordpress         9       ●         11       aliyun.probe.url: 'http://container/license.txt'         10       aliyun.scale: 'l'         11       ●         12       ●         13       ●         14       restart: always         15 · links:       ●         16       - mysql	
	确定	取消

7. 您也可以在路由列表下访问路由地址,进入 wordpress 应用欢迎页。

务列表 容器列表 日志 事件	件 路由列表
由地址(集群绑定SLB后路由地址才能被访	近))
rdpress.c	and a local

# 通过客户端工具进行设置

- docker help run:查看使用的 "-p" 选项,路由配置在容器服务管理控制台进行。
- docker-compose: 查看支持的 "ports" 选项,路由配置规则详情见 routing。

# 2.18.3 简单路由-域名配置

# 背景信息

#### 操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的服务。
- 3. 选择要添加域名的服务所在的集群。
- 4. 选择要添加域名的服务(本示例中要添加域名的服务为 web, 所属的应用为 wordpress)并单击 变更配置。如下图所示。

容器服务		服务列表						刷新
Kubernetes	Swarm		2					
概览		常见问题: 5	。如小 服务到Inter	net 🕜 给暴露公网	的服务添加域名 🔗	,将访问协议从http修改为https 🔗 更改应用对外端口		
应用		集群: swarr	m-test 🔻 🗹 開	■藏系统服务 🔲 障疑	藏憲线服务 💷 隐藏な	王线服务	名称 ▼	Q X
服务	1	服务名称	所属应用	服务状态	容器状态	镜像		操作
集群 节点		db	wordpress	●就绪	<b>就绪:1</b> 停止:0	registry.aliyuncs.com/acs-sample/mysql:5.7		监控   停止   重新调度   变更配置   删除   重启   事件
网络 数据卷		web	wordpress	●就绪	<b>就绪:3</b> 停止:0	registry.aliyuncs.com/acs-sample/wordpress:4.5	3	<u> 監控</u>   停止   重新调度   変更配置   删除   重启   事件

5. 单击简单路由配置右侧的加号图标,输入要添加的域名(本示例中要添加的域名为www. aliyuntest.club)并单击确定更新配置。如下图所示。

	说明:
同一个	服务同一个端口的多个域名只能写在同一个条目内,并且域名和域名之间用分号(;)分
隔。	

简单路由配置:	● ^企 如何暴靈 HTTP 服务		
	容器端口	域名	
	80	http://wordpress;www.aliyuntest.club	•
	注意:相同端口的多个域名只能填写在同一个条目内。多个域名用)'分隔		

此时,服务处于更新中。更新完毕变成就绪状态后,路由服务 acsrouting_routing 就 已经将该域名配置好了。当有请求以域名 www.aliyuntest.club 访问服务 wordpressrds_web 时,就能正确的解析并转发到相应的服务了。

- 将域名解析到容器服务的集群上。容器服务在创建集群的时候,会给每一个集群分配一个负载均 衡实例,该负载均衡实例是属于您自己的。
  - a) 登录容器服务管理控制台。
  - b) 在 Swarm 菜单下,单击左侧导航栏中的集群。
  - C)选择相应的集群,本示例为 swarm-test 并单击管理。

容器服务		集群列表					(B)	最多可以创建	5 个集群,每个集群最多词	]以添加 20 个节点	剧新	创建Swarm集	群 -
Kubernetes	Swarm												
概览		常见问题: S 如何创建集群 S 如何添加E	清云服务器 🔗 🗄	可用区节点管	■ 🖉 集成日志服务		·靖连接集群						
应用		名称 🔻											
服务	-	集群名称/ID	集群类型 1	地域 网络类	型	集群状态	节点状态 🖉	节点个数	创建时间	Docker版本	2		操作
集群	1	swarm-test	阿里云集群 4	峰东1 唐拟专	有网络	●运行中	健康 ℃	1	2018-03-13 16:30:55	17.06.2-ce	管理	査看日志   监控	删除 更多▼

## d) 单击负载均衡 , 并查看负载均衡 ID。

<	集群 : swarm-test
基本信息	
负载均衡	贝敷均衡设置 网络发育
用户标签	负载均衡Id:
镜像列表	
容器列表	重新绑定SLB 解绑SLB
集群事件	
节点伸缩	

7. 登录负载均衡控制台,在实例列表中找到目标实例 ID,进入实例详情页面,您可以查看负载均 衡实例的服务地址。

🔥 acs-slb				
基本信息	~			
负载均衡ID:	状态: ❷ 运行中			
负载均衡名称: acs-slb-c01eb01f92	地域: 华东 1			
地址类型: 公网	可用区: 华东1可用区 G(主)/华东1可用区 B(音)			
网络类型: 经典网络				
付费信息	消费明细 释放设置 へ			
付费方式: 按使用流量	创建时间: 2018-03-13 16:31:23			
服务地址:	自动释放时间: 无			
	<ul> <li>▲ acs-slb-</li> <li>… ● 返回负载均衡列表</li> <li>基本信息</li> <li>负载均衡D:</li> <li>负载均衡石称: acs-slb-c01eb01f92</li> <li>地址关型: 公网</li> <li>网络类型: 经典网络</li> <li>付费信息</li> <li>付费信息</li> <li>耐病量</li> <li>服务地址:</li> </ul>			

- 8. 登录阿里云云解析 DNS 服务管理控制台,添加域名解析(本示例中为 www.aliyuntest.club)。
  - a) 添加域名。若已有域名,跳过此步。
  - b) 添加域名解析。
    - 记录类型为 A
    - 主机记录为 www。主机记录即是域名前缀,您也可以选择其他前缀。
    - 解析线路为默认。
    - 输入绑定的负载均衡实例的服务地址。
    - 设置 TTL 值。

修改解析		×
记录类型:	A - 将域名指向—个IPV4地址 ~	
主机记录:	www .aliyuntest.club (?)	
解析线路:	默认 - 必填 ! 未匹配到智能解析线路时 , 返回 [默认] 线路 🗸 ?	
记录值:		
TTL值:	10 分钟 ~	
	<b>确认</b> 取注	肖

9. 访问页面 www.aliyuntest.club。

() www.aliyuntest.club/wp-admin/install.php	
English (Uni Afrikaans الحرية المعربية الحرية المعربية खात्रमीस خَوْتَنُ لَتَبَابِدَان Аzərbaycan Былгарски बारला बत्तील Bosanski Català Cebuano Čeština Cymraeg Dansk Doutsch (Se	ted States)

# 2.18.4 简单路由-HTTP 协议变为 HTTPS 协议

# 前提条件

如果您还没有配置成功 HTTP 协议的域名访问,请先了解配置 HTTP 的域名访问。更多详细信

息,参见简单路由-域名配置。

# 操作步骤

1. HTTPS 协议是在负载均衡这一层进行支持的。为了支持 HTTPS 协议,您需要创建负载均衡证

书。

- a) 登录 负载均衡管理控制台。
- b) 单击左侧导航栏中的证书管理并单击页面右上角的创建证书。

负载均衡 实例管理	证书管理 亚太东南1(新加坡) 欧洲中部1(法兰 美国西部1(社谷) 华东1	売福) 华北1 美国东部1(弗吉尼亚) 华北2 华	东2 亚大东北1(日本) 华瑞1 香港	当前区域已创建14	C 心正书。 仅HTTPS协议需要证书,每个用户最	· 周新 2 建100个证书
征书管理 标签管理	证书名称	i£+81D	证书颁放	地域	证书类型	操作
操作日志		and constraints of the second	cd:90:1b:7b:49:4d:1d:90:f6:01:de:9a:81:7d:31:a7:38:1d:84:8d	豆太东南 1 (新加坡)	服务器证书	修改名称   删除

c) 输入证书的相关信息。

更多详细信息,参见证书要求和生成证书相关的文档,如下图所示。

표密 2010	创建证书 。迈回证书列表	
Kallenging 20-00-0+本存、月26本包含年、数本、ス・ア、ス・ビ Withergin ( 新加坡) = 40:0 年4:1 ■ 美国伝藤町 ( 集吉尼豆) ■ 4:2 = 4:5; 2 ■ 4:前 ■ 音書 ■ 美国西部 1 (後合) # 4:5; 1 Hathback/TEX.5; 前 ( 新加坡)) Witter 20:0 年4:1 ■ 美国伝藤町 ( 東古尼豆) ■ 4:2 = 4:5; 2 ■ 4:前 ■ 音書 ■ 美国西部 1 (後合) # 4:5; 1 Hathback/TEX.5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 = 4:5; 1 =	证书名称:	www.example.com
*征书region:       * 型大东南 1 (新加線) [©] 华北 1 [©] 無面分部 1 (弗吉尼亚) [©] 华北 2 [©] 华东 2 [©] 华南 1 [©] 書港 [©] 美丽田田田 1(社会) [©] 华东 1 田田田田田 1(社会) [©] 华东 1         *征书供题:       * 医安静正 4 [©] CA正书         *征书供题:       * 医安静石 4 [©] CA正书         *征书供题:       * 医安静石 4 [©] CA正书         *征书供题:       * 医安静石 4 [©] CA正书         *征书内容:       * 1         *征书内容:       * 1         *征书内容:       * 1         * 1       * 1         * 4       ● 日本(1)         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 1       * 1         * 2       * 1         * 2       * 1         * 2       * 1         * 4       * 1         * 4       * 1         * 4       * 1         * 4       * 1         * 4       * 1         * 4       * 1         * 4       * 1         * 5       2         * 2       2 <t< td=""><td></td><td>长鹰限制为1-80个李符,只允许包含字母、数字、"、"、"、"。</td></t<>		长鹰限制为1-80个李符,只允许包含字母、数字、"、"、"、"。
• '证书短照 : • · · · · · · · · · · · · · · · · · ·	*证书region:	⁸ 亚大东南 1 (新加坡) ◎ 华北 1 ◎ 萬国东部 1 (弗吉尼亚) ◎ 华北 2 ◎ 华东 2 ◎ 华南 1 ◎ 香港 ◎ 萬国西部 1 (硅谷) ◎ 华东 1 当前地域力(亚太东南 1 (新加坡))
· 正书均容:         1 ************************************	*证书类型:	● 服务器证书 ○ CA证书
*证书內容:         1        BECMI CERTIFICATE		证书编辑器里面的格式校验功能只起到揭醒的功能,格式校验通过不能代表证书格式一定能被识别
(pem编码) 員入注例 *私胡: 1BEGIN RSA PRIVATE KEY	*证书内容:	1      BEGNL CENTEFICATE
	*私钥:	(pemi88))       単人社務         1
<b>教认</b> 取消		<b>論认</b> 取0消

2. 证书创建成功后,找到创建集群时分配的负载均衡实例。

容器服务在创建集群的时候,给每一个集群分配了一个负载均衡实例,该负载均衡实例是属于您自己的。

- a) 登录容器服务管理控制台。
- b) 单击左侧导航栏中的集群,选择相应的集群,本示例中为routing-test-online,单击管理。

容韻服务	集群列表						40 40	8最多可以创建 5 个集群,每个集群最多可以添加 20 个节点	子账号授权	刷新	01325484
概范	小助手: 创建素群 如何添加已有云服负器 跨可用区节点管理	集成日志服务	通过Docker客户	洗涤按集群							
成用服务	名称 *										
## <b>1</b>	集群名称/ID	地域	网络类型	集群状态	节点状态 🔘	节点个数	创建时间	Docker版本			操作
节点	test	华北1	经共同络	<ul> <li>81/8</li> </ul>	健康 C	0	2016-10-13 13:29:11	1.11.2.1	管理	查看日志 监报	删除 2   更多▼
> 損懲−模板	routing-test-online	华北1	经典网络	🔴 Ətila	健康 C	1	2016-10-13 11:29:18	1.11.2.1	2 📰	堂看日志 监持	肥除 2   更多+
捐電											

C) 单击负载均衡,并查看负载均衡 ID。

<	集群: routing-test-online			
基本信息				
负载均衡	负载均衡设置 城名设置			
用户标签	免载的街Id: Ib-ho			
镜像列表				
容器列表	重新绑定SLB 解绑SLB			

前往产品与服务中查找,并进入负载均衡管理控制台。根据 ID 查看对应负载均衡实例的服务

地址。

🛃 lbm5ea6s1nvifz9897	@ 使用限制和注意事项
基本信息	^
负载均衡ID:	状态: 🔮 运行中
负载均衡名称: (未设置)	地域: 华北 1
地址类型: 公网	可用区: 华北 1 可用区 A(圭)/华北 1 可用区 B(甾)
网络类型: 经典网络	
付费信息	消费明细 释放设置 ヘ
付魏方式: 按圓定帶宽	带宽值: 1M
创建时间: 2016-10-14 12:23:50	服务地址:
自动厚放时间: 无	

3. 单击左侧导航栏中的监听并单击添加监听。在添加监听页面,填写端口信息,如下所示。

+ I	-+-	   tホiン	- + - 	 端口	-+ 
 +		+	 	->nu ⊨i + – – – – -	- – +
前端协议(端口)		HTTPS		443	Ι.
+		HTTP		9080 +	+

- a) 前端协议,选择 HTTPS。
- b) 端口使用 443 端口,后端端口使用 9080 端口(该端口为路由服务acsrouting_routing 在每 一台 ECS 主机上暴露的端口,所有的 HTTP 请求会在路由服务 acsrouting_routing上根据 HTTP 协议的 HOST header 转发到相应的提供各种服务的容器内)。
- C) 选择前面步骤创建的证书 www.example.com。
- d) 根据需要设置其它选项。
- e) 单击下一步。

添加监听	>
1.基本配置	2.健康检查配置 3.配置成功
前端协议 [端口]:★	HTTPS ▼ : 443 端口输入范围为1-65535。
后端协议 [端口]: <b>*</b>	HTTP : 9080 端口輸入范围为1-65535。负载均衡协议为HTTPS时,后端协议为HTTP
带宽峰值:*	1 M 可用: 1M (已用0M,共1M) 固定带宽计费方式的实例,不同监听分配的带宽峰值总和不能超出在创建负载均衡实 例时设定的带宽总值
调度算法:	加权轮询
使用虚拟服务器组:	
双向认证:	关闭
服务器证书:🔺	www.example.com/     ▼ 新建证书
创建完毕自动启动监听	
▼ 展开高级配置	
	下一步取消

4. 完成健康检查配置标签页中的配置,如下所示。单击确认。

您可以选择开启或关闭健康检查。如果您选择开启健康检查,您需要在域名中填写您自己的域名 或者在检查路径中填写 /haproxy-monitor。否则,健康检查会报异常。

添加	监听		$\times$
	1.基本配置	2.健康检查配置 3.配置成功	
	是否开启健康检 查: 🛛		
	域名:	长度1-80个字符	
		只能使用字母、数字、`-'、`-',默认使用各后端服务器的内网IP为域名	
	检查端口:	端口輸入范围为1-65535。	
		默认使用后端服务器的端口进行健康检查	
	检查路径:	/haproxy-monitor	
		用于健康检查页面文件的URI,建议对静态页面进行检查。长度限制为 1-80个字符,只能使用字母。数	
		字、`-'、`/'、`.'、`%'、`?'、`#'、`&'、`='这些字符。	
	响应超时时间:*	5 秒	
		每次健康检查响应的最大超时时间;输入范围1-300秒,默认为5秒	
	健康检查间隔:*	2 秒	
		进行健康检查的时间间隔; 输入范围1-50秒,默认为2秒	
	不健康阈值:*		
		2 3 4 5 6 7 8 9 10 表示云服务器从成功到失败的连续健康检查失败次数。	
	健康阈值:*		
		2 3 4 5 6 7 8 9 10 表示云服务器从失败到成功的连续健康检查成功次数。	
	止常状态码:	● http_2xx ● http_3xx ● http_4xx ● http_5xx 健康检查正常的http状态码	
		上一步 确认 取消	肖 (

5. 配置成功后,单击确认。

添加监听						$\times$		
	1.基本配置	>	2.健康检查配置		3.配置成功			
<ul> <li>✓ 恭喜,监听配置成功!</li> <li>◇ 监听已创建成功。</li> <li>◇ 监听启动成功。</li> <li>请关闭窗口查看新添加的监听。</li> </ul>								
						确认		

6. 访问页面https://www.example.com。

	S https://www.example.com	<ul> <li>۵ ۵ ۵</li> </ul>
开发者控制台	阿里云-帐号登录	hello wordpress-rds - 又一个WordPress站点
hello wordpress _{又─} ৵₩ordPressk点 世界,您好!	-rds	Search Q
2016年4月11日 有1条评论	欢迎使用WordPress。这是您的第一篇文章。编辑或删除它,然后开始写作吧!	近期文章 • 世界、愈好!
		近期评论 <ul> <li>WordPress先生发表在《世界,您好!》</li> </ul>

## 后续操作

完成以上配置后,如果您需要设置访问 http://www.example.com,直接跳转到https://www.example.com,请参考简单路由-HTTP 强制跳转到 HTTPS 进行设置。

# 2.18.5 简单路由-HTTP 强制跳转到 HTTPS

#### 步骤1 实现 https 协议访问 helloworld 应用

1. 您可以使用编排模板创建 hello world 应用。

应用模板模板示例如下。

2. 配置好负载均衡之后,参见简单路由-HTTP协议变为HTTPS协议,访问HTTPS协议的网站如

下所示。

```
← → C ▲ https://www.example.com
Hello world!
My hostname is e1f197f06080-http-to-https-app-1
```

#### 步骤2 配置 nginx 容器实现强制跳转到 HTTPS

1. 您可以配置使 HTTP 协议请求强制跳转到 HTTPS 协议。

本例中,创建一个 nginx 容器,监听 http 请求,将所有的 http 请求通过 rewrite 重写到

https上,从而将请求重定向至 https 协议,实现基于 www.example.com 域名的强制跳转。

下面的示例配置了 nginx 容器并将 rewrite 规则写到配置文件中,即如果收到请求 http://www .example.com,则返回 301 且自动跳转到 https://www.example.com。

• 登录集群中的每台机器,创建 nginx.conf 配置文件,该配置文件将以 volume 形式挂载到容器 nginx 中。

cd /	##	返回根目录
mkdir ngx	##	创建 ngx 目录
vim nginx.conf	##	创建 nginx.conf 配置文件

• 在/ngx/nginx.conf 配置文件中输入如下的配置代码。

```
user nginx;
error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;
```

```
events {
  worker connections 65535;
http {
                /etc/nginx/mime.types;
  include
  default_type application/octet-stream;
  log_format main '$remote_addr - $remote_user [$time_local] "$
request" '
                     '$status $body_bytes_sent "$http_referer" '
  '"$http_user_agent" "$http_x_forwarded_for"';
access_log /var/log/nginx/access.log main;
  keepalive_timeout 65;
  gzip on;
  server {
                    80;
      listen
      server_name localhost;
      return 301 https://$host$request_uri;
  }
}
```

2. 使用编排模板创建 nginx 应用。

nginx 应用的编排模板如下所示。

```
nginx:

ports:

- 80:80/tcp # 映射到主机的 80 端口

image: 'nginx:latest'

labels:

aliyun.global: true # 每台机器均部署一个 nginx 容器,达到高可用目的

volumes:

- /ngx/nginx.conf:/etc/nginx/nginx.conf

restart: always
```

3. 配置集群负载均衡的监听规则。

如下图所示(其中,前端 80 端口 > 后端 80 端口,即负载均衡的前端端口 > 后端 ECS 实例的端口 80)。

1 #	缸								添加监听  刷新
	) 前端协议/端口	后端协议/端口	状态	转发规则	会话保持	健康检查	带宽峰值	服务器组	操作
	TCP: 80	TCP: 80 正常	❷ 运行中	加权轮询	关闭	已开启 🕖	不限制	无	配置   详情   更多-
	HTTPS: 443	HTTP: 9080 正常	◙ 运行中	加权轮询	关闭	已开启 🕖	不限制	无	配置   详情   添加转发策略   更多-
	启动停止	删除							

4. 验证 HTTP 强制跳转到 HTTPS

当您访问 http://www.example.com 时,会自动跳转到 https://www.example.com。返回的 HTTP 协议内容如下图所示,即完成了正确跳转到 https://www.example.com。

R	6	Ele	ement	s	Cons	ole	Sour	ces	Netw	ork	Tin	nelin	e	Profi	les	Applic	ation	Securit	y Au	dits	Adblock F	lus	
٠	$\otimes$		Y	V	/iew:		N		Preser	ve lo	g	0 Di	isabl	e cac	he	□ 0	ffline N	o thrott	ling		•		
Filter					] 🗆 F	legex		Hide	data U	RLs	All	Xł	ΗR	JS	CSS	Img	Media	Font	Doc	WS	Manifest	Other	
_	10	00ms			200ms			300m	s		400m	15		5	00ms		600m	5	70	Oms	800	Dms	90(
Name Path	me th							× Headers Preview Response Cookies Timing															
	www.example.com						Request URL: http://www.example.com/ Request Method: GET Status Code: 301 Moved Permanently Remote Address: 120.76.69.214:80																
	www.example.com																						
CSS	css?family=Open+Sans:400,700 fonts.googleapis.com					Response Headers view source     Connection: keep-alive																	
	content.min.css mgijmajocgfcbeboacabfgobmjgjcoja k3k702ZOKiLJc3WVjupIzBampu5_7CjHW5spxoeN3Vs.w fonts.gstatic.com/s/opensans/v13						Content-Length: 185 Content-Type: text/html Date: Thu, 29 Sep 2016 11:57:50 GMT Location: https://www.example.com/ Server: nginx/1.11.4																

# 2.18.6 集群内服务间路由和负载均衡

在容器服务上可以通过简单路由#支持 HTTP/HTTPS)将基于域名的 HTTP 服务暴露出去,而且能够配合健康检查自动的负载均衡和服务发现,当其中一个容器出现问题之后,routing 会自动将健康检查失败的容器从后端摘除,所以能做到自动的服务发现。然而这个是将服务暴露到外网环境。

那么集群内服务间如何通过这种方式做到自动的服务发现和负载均衡呢?阿里云容器服务中的 routing 容器具备负载均衡的功能,您只需要使用以.local 结尾的域名,让容器仅能被集群内的其 他容器所访问,然后配合 external_links 标签,从而实现集群内服务间发现和负载均衡。

#### 实现原理

- 利用了 Docker 1.10 之后支持在容器中做别名的方式,在依赖负载于 restserver.local 的 restservice 容器中, restserver.local 域名实际解析的是 routing 容器的地址, restclient 服 务发起请求时,首先将 HTTP 请求转发到 routing 容器,并带上 HOST 为 restserver.local 的请求头。
- routing 容器会对配置 aliyun.routing_port_xxx: restserver.local 标签的容器健康 状态进行监听,并将其挂载到 HAProxy 的后端, HAProxy 接收到带有 restserver.local HOST 头的 HTTP 请求后,就能转发到对应的容器。



优势

- 相对于使用 link 或者 hostname 的基于 DNS 的方式,首先不同客户端对 DNS 缓存的处理不一 致会导致服务发现的延迟性,其次 DNS 的方案也只有 round robin,不能满足微服务的场景需求。
- 而相对于其他的微服务服务发现的解决方案,本方案提供了一个实现无关的服务发现和负载均衡 机制,无需 server 端和 client 应用做任何修改即可使用。
- 服务生命周期是解耦的,每个微服务可以采用一个 docker-compose 模板独立部署,更新。相互 之间只是通过一个虚拟域名实现动态绑定即可。

## 编排示例

在下面的编排示例中,为 restserver 服务增加 aligun.routing.port_80: restserver. local标签,确保只有集群内的容器可以访问这个域名。然后为 restclient 服务配置 external_1 inks,指向 restserver.local 这个域名。reclient 服务便可以这个域名访问到 restserver 服务,并且 能够配合健康检查做到自动的服务发现。

```
restserver: # 模拟 rest 服务
image: nginx
labels:
```

aliyun.routing.port_80: restserver.local # 使用 local 的域名,只有集 群内的容器可以访问这个域名 aliyun.scale: "2" # 扩展出两个实例,模拟负载均衡 aliyun.probe.url: "http://container:80" # 定义容器的健康检查策略是 http,端口是 80 aliyun.probe.initial_delay_seconds: "2" # 健康检查在容器起来之后两秒之 后再检查 aliyun.probe.timeout_seconds: "2" # 健康检查超时时间,如果两秒还没返回认 为不健康 restclient: # 模拟 rest 服务消费者 image: registry.aliyuncs.com/acs-sample/alpine:3.3 command: "sh -c 'apk update; apk add curl; while true; do curl -head restserver.local; sleep 1; done'" #访问 rest 服务 测试负载均衡 tty: true external_links: - "restserver.local" #指定 link 的服务的域名。请确保您设置了 external_1 inks,否则访问会失败。

然后,通过如下的 restclient 服务的日志,您可以看到 restclient 的 curl 的 http 请求被路由到不同 rest 服务的容器上,容器 ID 分别为 053cb232fdfbcb5405ff791650a0746ab77f26cc e74fea2320075c2af55c975f 和 b8c36abca525ac7fb02d2a9fcaba8d36641447a7 74ea956cd93068419f17ee3f。

internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066803626Z Server: nginx/1.11.1 internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066814507Z Date: Fri, 01 Jul 2016 06:43:49 GMT internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066821392Z Content-Type: text/html internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066829291Z Content-Length: 612 internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066835259Z Last-Modified: Tue, 31 May 2016 14:40:22 GMT internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066841201Z ETaq: "574da256-264" internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066847245Z Accept-Ranges: bytes internal-loadbalance_restclient_1 | 2016-07-01T06:43:49.066853137Z Set-Cookie: CONTAINERID=053cb232fdfbcb5405ff791650a0746ab77f26cc e74fea2320075c2af55c975f; path=/ internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.080502413Z HTTP/1.1 200 OK internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082548154Z Server: nginx/1.11.1 internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082559109Z Date: Fri, 01 Jul 2016 06:43:50 GMT internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082589299Z Content-Type: text/html internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082596541Z Content-Length: 612 internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082602580Z Last-Modified: Tue, 31 May 2016 14:40:22 GMT internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082608807Z ETaq: "574da256-264" internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082614780Z Accept-Ranges: bytes

```
internal-loadbalance_restclient_1 | 2016-07-01T06:43:50.082621152Z
Set-Cookie: CONTAINERID=b8c36abca525ac7fb02d2a9fcaba8d36641447a7
74ea956cd93068419f17ee3f; path=/
```

# 2.18.7 负载均衡路由

### 暴露 HTTP 协议或者 HTTPS 协议的服务

推荐使用简单路由服务(即 routing)的方式来暴露 HTTP 服务或者 HTTPS 协议的服务,如果您希望搭建自己的路由链路,可以开通新的内网或者公网负载均衡实例路由到 VM 的端口(通过阿里云扩展标签/b 来实现),并设置主机和容器的映射关系来进行请求的路由。

适用场景:

7 层协议负载均衡,自定义各服务的路由,在将传统架构迁移到容器架构过程中非容器集群的服务 访问容器集群中的服务。

### 暴露 TCP 协议或者 UDP 协议的服务

目前如果要暴露 TCP 协议的服务,需要您自行设置负载均衡实例或者公网 IP,并配置好主机端口 与容器端口的映射(通过阿里云扩展标签/b 来实现)。

📕 说明:

如果要使用负载均衡实例,您需要购买一个新的负载均衡实例。多个服务不能共享使用同一个负载均衡实例,同时不能共享使用集群默认负载均衡实例。

适用场景:

4 层协议的负载均衡,自定义各服务的路由,在将传统架构迁移到容器架构过程中非容器集群的服务访问容器集群中的服务。

示例:

通过自定义负载均衡的方式来将容器集群内的 Redis 服务暴露给容器集群外的 Python 应用。

首先在负载均衡管理控制台负载均衡管理控制台(单击页面右上角的创建负载均衡)购买创建
 一个用于路由的负载均衡实例。

本示例中选择的是公网实例,您可以根据自己的需要选择公网或者私网。



域。														
	地域	华北 1(青岛)	华北 2(北京)	华北 3(张家口)	华北 5(呼和浩特)	华东 1(杭州)	华东 2(上海)							
					亚太东南 1(新加		亚太东南 3 (吉隆							
		华南 1(深圳)	香港	亚太东北 1(东京)	坡)	亚太东南 2(悉尼)	坡)							
					欧洲中部 1(法兰									
		美东 1(弗吉尼亚)	美西 1(硅谷)	中东东部 1(迪拜)	克福)									
		不同地域之间的产品内网不互通;订购后不支持更换地域,请谨慎选择 <b>教我选择&gt;&gt; 查看我的产品地域&gt;&gt; 各区</b> 域黑洞触发阈值>>												
本記書	可用区类型	多可用区												
파		单可用区指实例只 复服务。 <b>详情参考</b> 》	在一个可用区存在 >>	E;多可用区指实例Q	在两个可用区存在,	当主可用区不可用的	村会在备可用区恢							
	主可用区	华东 1 可用区 F ▼												
		主可用区是当前承	載流量的可用区,	备可用区默认不承望	載流量,主可用区2	不可用时才承载流量	▌教我选择>>							
	备可用区	华东1可用区E	•											
	实例规格	请选择规格	•											
	实例类型	公网	私网	实例类型详解>>(	3									
副		负载均衡实例仅提供公网IP,可以通过Internet访问的负载均衡服务												
	计费方式	按使用流量计费	按固定带宽计费											
調図		开通即按使用流量; 进行变配操作时,	十费,停止或释放 告选择的计费方式	(实例才不会产生流量 (与当前计费方式不同	量费用 司,则代表变更计絜	豊方式,变更计费方	式将在第二天0点							
		生效 阿用 <del>テ展</del> 宮提供50	hns的亚奇法县功	本院拍 了解面名。	。。坦升防拉能力。。									
		阿里云现已开通共 >>	算流量包,可同时	抵扣 ECS、EIP、S	SLB、NAT 产生的》	充量。点击购买特惠	試享流量包套餐							
民	购买数量	1												
2025		您当前已经拥有6个	实例,您还可以创	健54个实例										

由于负载均衡不支持跨地域(Region)部署,因此应选择与您所使用容器服务集群相同的地

 返回负载均衡管理控制台,将购买创建的负载均衡实例命名为 slb_redis_app。容器服务能通过 该名称来引用这个负载均衡实例。您也可以通过负载均衡实例 ID 来引用。

单击左侧导航栏中的实例管理,选择实例所在的地域>选择所需实例>编辑实例的名称并单 击确定。

负载均衡	次例管理         亚太东湾 1 (新加坡)         欧洲中部 1 (決主売福)         华北1         美国东部 1 (清吉尼亚)         华北2         华北2	天 2 亚太东北1(东京) 华南1 香港 亚太东南2(恶尼)
实例管理 1		C 刷新 的建负载均衡
证书管理	○ 気気均衡10 ▼ 诺输入实例ID进行微频面词。多个校,分档 方式 ●标签	۷ ۵
操作日志	□ 负载均衡ID/名称 可用区 服务地址(全部) ▼ 状态 网络类型(全部) ▼	第□/健康检查 后端服务器 标签 付费方式(全部) ▼ 操作
产品文档	Ib-bp1nvemd0         华东1可用区 E(主)           (未设置)         华东1可用区 D(釜)公网	未配置(配置) 未配置(配置) 未绑定 按使用流量 管理   更多 ↓ 2017-02-23 16:18:45 创建
	编辑负载均衡名称: (bb_redis_app 长旗限制力1-80个字符, 允许包含字母、数字、 ¹¹ 、 ¹¹ 、 ¹¹ 、 ¹¹ 、 ¹¹ 、 ¹² 、 ¹² 、 ¹²	

3. 创建监听端口。

单击实例右侧的管理,单击左侧导航栏中的监听 > 添加监听 > 设置监听配置。创建协议为 TCP,端口映射为 6379:6379。如下图所示。

添加监听					×
1.基本配	E	2.健康检查配置	>	3.配置成功	
前端协议 [i 口]:*	端 TCP 端口输 四层监则 查看详情	▼ : 6379 \范围为1-65535。 f请选择TCP、UDP;	] 七层监听请选择	HTTP、HTTPS;	
后端协议 [i 口]:*	端 TCP 端口输 <i>)</i>	: 6379 \范围为1-65535。	]		
带宽峰值:	不限制 使用流量 5000	配置 量计费方式的实例默认	人不限制带宽峰值	訂峰值輸入范围1-	
调度算法:	加权轮	涧 ▼			
使用服务器 <b>②</b>	組: 〇	◙什么是服务器组?	,		
创建完毕自 动监听:	动启	已开启			
● 展开高	汲配置				
				下一步即	训
登录容器服务管理控制台,选择一个已有集群,创建一个名称为 redis-demo 的应用,单击使用 镜像创建。

有关如何创建应用,参见创建应用。



由于负载均衡不支持跨地域(Region)部署,因此您所使用的容器服务集群需要和上边创建的 负载均衡实例处于相同的地域。

	应用基本信息		应用配置	$\rightarrow$	创建完成
应用名称	redis-demo				
	名称为1-64个字符,可包含数	字、英文字符,或"-",且	不能以-开头		
应用版本 :	1.0				
部署集群:	swarm-test	¥			
默认更新策略:	标准发布	v			
应用描述 :					
	🗆 检查最新Docker镜像 🕧				
					使用镜像创建 使用编排模板创建

5. 选择 Redis 镜像并设置端口映射。

# - 说明:

此处 Redis 镜像只是开通了容器的 6379 端口,为了使创建的负载均衡路由到这个容器端口,您必须指定 Redis 镜像的主机端口:容器端口的映射。

在端口映射中,指定主机端口为 6379,主机端口 6379 即为负载均衡实例绑定的后端主机端口,选择使用的协议为 TCP.

		应用基本信息		应用配置
至本配置	镜像名称: 容器数量:	redis 违话的像 1	镜像版本: 网络模式:	: latest 选择镜像版本 : 默认、 ·
***	Restart :	Ø Always		
	Command :			
配置	Entrypoint :			
響	CPU限制:		内存限制:	: MB
	Capabilites:	ADD DROP		
	端口映射:	● 如何给公网的服务添加域名		
		主机端口	容器端口	协议
E		6379	> 6379	/ тор 🔹 🗢
网络		注意:主机端口不能设置为9080,2376,3376		

- 为了配置自定义负载均衡,需要让 Redis 服务知道使用的负载均衡实例的信息。您可以通过向服务注入一个标签来实现或者通过设置负载均衡路由配置。
  - 向服务注入一个标签。本示例中,标签为aliyun.lb.port_6379: tcp://slb_redis_ app:6379。

labels :	● 『☞ 阿里云扩展能力的标签		
	标签名	标签值	
	aliyun.lb.port_6379	tcp://slb_redis_app:6379	•

标签格式如下,带\$的变量为占位符。

```
aliyun.lb.port_$container_port:$scheme://$[slb_name|slb_id]:$
front_port
```

- \$container_port 表示容器要暴露的端口。
- \$scheme表示负载均衡实例监听端口支持的协议,可能的值为tcp、http、https、udp。
- \$[slb_name|slb_id] 表示可以填写负载均衡实例的名称或者 ID。
- \$front_port 表示负载均衡实例要暴露的前端端口。

更多详细信息,参见阿里云扩展标签*lb*。

在创建应用页面,单击负载均衡路由配置右侧的加号图标,设置要配置的负载均衡实例的信息,如下图所示。

该设置对应的标签内容为 aliyun.lb.port_6379: tcp://slb_redis_app:6379。

负载均衡路由配置	① 心 如何使用自定义负载均衡方式暴露服务		
	容器端口	自定义负载均衡	
	aliyun.lb.port_6379	tcp://slb_redis_app:6379	•
	注意:不同服务不能共享使用slb		

本示例中,路由到的容器端口为 6379,引用前面创建的负载均衡实例名称 slb_redis_app ,与上面主机端口:容器端口的映射设置的 TCP 协议相呼应,本示例设置监听端口的协议为 TCP 协议,同时设置负载均衡的前端端口为 6379。

🗾 说明:

本示例中,同时将负载均衡实例的前端端口、后端端口(即主机的端口)和容器端口均设置为 6379,您可以根据自己的需要设置不同的前端端口和主机端口。

- P. 单击创建, Redis 应用即开始创建了。Redis 应用在创建的过程中会自动将名称为 slb_redis_app 的负载均衡实例绑定到部署了 redis 镜像的后端主机。
- 8. 当应用处于就绪状态后,登录负载均衡管理控制台,查看名为 slb_redis_app 的负载均衡实 例的状态。

单击左侧导航栏中的实例管理,选择实例所在的地域>选择所需实例>单击实例右侧的管理> 单击左侧导航栏中的服务器>后端服务器。

由健康状态可见,负载均衡已经正确地绑定到了 Redis 的后端。

<	👗 slb_redis_app t 返回负氟均衡列表	用限制和注意事项
洋情	① 负载均衡服务器池 所在地域:华东1 可用区 E (主)/华东1 可用区 D (音) ●	
监听 ▼ 服务器	已添加的服务器 未添加的服务器	
后端服务器	云銀号器D ▼	€ 刷新
虚拟服务器组	□ 云般先職ID/名称         可用区         公网/内阿P地址         状态(全部) ▼         网络关型(全部) ▼         健康检查状态         权重	操作
主音服务器组	i-bp1bms413p14fnokchh         4/5 1 可用区 E         公)         ● 运行中         经典网络         正常         100	移除

- 9. 您可以在负载均衡管理控制台的实例管理页面查看负载均衡实例的 IP 地址,并使用命令行工具 telnet \$Server_Load_Balancer_IP_address 6379 来检查端口的可访问性。
- **10**.为了测试以上配置,在本地运行一个简单的 Python 应用来通过 slb_redis_app 负载均衡实例访问容器集群内的 Redis。

🗾 说明 :

Redis 主机地址是负载均衡的 IP 地址。

#### app.py

```
from flask import Flask
from redis import Redis
app = Flask(__name__)
redis = Redis(host='$Server_Load_Balancer_IP_address', port=6379)
@app.route('/')
def hello():
redis.incr('hits')
return 'Hello World! I have been seen %s times.' % redis.get('hits')
if __name__ == "__main__":
app.run(host="0.0.0.0", debug=True)
```

### requirements.txt

### flask redis

### shell

```
$ pip install -r requirements.txt
$ python app.py
Running on http://0.0.0.0:5000/ ## Press CTRL+C to quit
Restarting with stat
Debugger is active!
Debugger pin code: 243-626-653
```

访问结果如下图所示。



# 2.18.8 容器间的互相发现

容器服务为集群内的服务和容器提供多种服务发现方式,可以通过容器名、link、hostname 等进行发现。

# 通过容器名

容器服务不仅可以通过容器的 IP 进行访问,还可以通过网络中其他容器的容器名进行访问,通过跨 主机互联的容器网络中的例子,您可以在 test_network-test2_1 的容器中通过 test_netwo rk-test1_1 的容器名进行访问。

如果在编排文件中不指定 container_name 的话,默认的容器名为 {project-name}_{ service-name}_{ container-index}。在连接管理终端后,您可以通过另外一个服务的容器 名进行访问,来测试网络的连通性。如下图所示。

shell	sh		执行
•			
/ # ping test networ	k-testl l		
PING test_network-test	:1_1 (172.18.0.4): 56 da	ita bytes	
64 bytes from 172.18.0	).4: seq=0 ttl=62 time=0	.245 ms	
64 bytes from 172.18.0	).4: seq=1 ttl=62 time=0	.276 ms	
64 bytes from 172.18.0	.4: seq=2 ttl=62 time=0	.263 ms	
64 bytes from 172.18.0	).4: seq=3 ttl=62 time=0	0.304 ms	
64 bytes from 172.18.0	).4: seq=4 ttl=62 time=6	.254 ms	
64 bytes from 172.18.0	).4: seq=5 ttl=62 time=6	.281 ms	
64 bytes from 172.18.0	).4: seq=6 ttl=62 time=6	.270 ms	
64 bytes from 172.18.0	).4: seq=7 ttl=62 time=6	.278 ms	
64 bytes from 172.18.0	).4: seq=8 ttl=62 time=0	.308 ms	
64 bytes from 172.18.0	).4: seg=9 ttl=62 time=6	).244 ms	

## 通过 link

容器服务支持编排模板服务间的 link,服务间的 link 可以将一个服务的容器 link 到另外一个服务的容器中,而容器中可以通过 link 进来的服务别名访问到依赖的容器,并且在依赖的容器的 IP 变化时可以动态的更新别名解析的 IP。具体的例子可以参考容器服务示例编排中的 WordPress 编排,其中 WordPress 中 Web 服务 link db:mysql 的服务到容器内,如下所示。容器内部就可以通过 MySQL 的域名访问到 db 服务的容器。

```
links:
- 'db:mysql'
```

#### 通过 hostname

如果在编排模板的服务中定义了 *hostname* 的配置,则在集群中便可以通过这个 hostname 访问到 这个容器。

例如:

```
testhostname:
image: busybox
hostname: xxserver
command: sleep 100000
tty: true
```

那么,集群中就可以通过 xxserver 解析并访问到这个服务的容器,具体请参见跨主机互联的容器 网络 的编排示例。当这个服务在有多个容器时,通过这个域名访问还可以做到一定的负载均衡的作 用。

另外,如果服务没有配置 hostname 的话,容器服务会把容器的容器名作为容器内部的 hostname;如果有应用需要在容器内知道自己的容器名,用于服务的注册,比如 Eureka Client,需要注册一

个可被访问的地址到 Eureka Server,容器内的进程可以获取到容器名用于服务注册,并让其他的服务调用者通过容器名互相访问。

# 2.18.9 自定义路由-使用手册

#### acs/proxy

自定义代理镜像,通过 FROM dockercloud/haproxy 的方式继承自镜像 dockercloud/ haproxy,动态感知容器的状态,做到后端容器负载均衡代理和服务发现。特点是将 HAProxy 负 载均衡软件的所有配置都参数化了,方便您自定义自己的需求和配置。

该镜像主要用于 Alibaba Cloud 容器服务的默认路由服务不能满足您需求的场景,方便您对 HAProxy 进行自定义配置。

文档中会提到 acs/proxy 和 HAProxy,均指代该镜像或者镜像中的软件 HAProxy。

### 动态负载均衡代理和服务发现的原理

- 镜像 acs/proxy 通过容器自身环境变量确定负载均衡的全局(GLOBAL)和默认(DEFAULT)配置。
- 镜像 acs/proxy 侦听集群中的事件,例如容器状态的变化,发生变化后重新获取集群中相关容器的信息,确定最新的负载均衡配置。
- 镜像 acs/proxy 根据最新的负载均衡配置去重新加载 (reload) 该配置,使得该配置生效。

# 如何确定负载均衡的后端容器

- 根据 acs/proxy 的环境变量 ADDITIONAL_SERVICES 来确定范围。
  - ADDITIONAL_SERVICES: "*" 表示范围为整个集群。
  - ADDITIONAL_SERVICES: "project_name1:service_name1,project_name2: service_name2"表示范围为当前应用和指定应用的指定服务。
  - ADDITIONAL_SERVICES 不设置或者为空表示范围为当前应用的容器。
- 根据每个容器的标签来确定是否加入 acs/proxy 的后端。
  - aliyun.proxy.VIRTUAL_HOST: "www.toolchainx.com" 表示加入后端,且域名为 www.toolchainx.com。
  - aliyun.proxy.required: "true" 表示加入后端,且作为默认的后端。

#### 如何在前端绑定负载均衡

使用自定义负载均衡标签,例如 aliyun.lb.port_80: 'tcp://proxy:80'。

```
送明:
```

- 请不要将集群的负载均衡实例(即简单路由的负载均衡)用来做自定义负载均衡。
- 任何两个不同的服务均不能共享使用同一个负载均衡,否则会导致负载均衡后端机器被删除,服务不可用。

关于自定义负载均衡标签的使用方法,参见标签。

```
示例模板
```

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   ports:
            '80:80'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
       aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global: "true"
       # 使用自定义负载均衡,前端绑定负载均衡
aliyun.lb.port_80: tcp://proxy_test:80
   environment:
          支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       #
       ADDITIONAL_SERVICES:
                           " * "
appone:
   expose: # 被代理的服务一定要使用 expose 或者 ports 告诉 proxy 容器暴露哪
个端口
          80/tcp
       _
   image: 'nginx:latest'
   labels:
       # 此处支持 http/https/ws/wss 协议
       # 必须使用您自己的域名而不是容器服务提供的测试域名
       aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
   restart: always
```

# 配置说明

通过 acs/proxy 镜像的环境变量设置全局(GLOBAL)和默认(DEFAULT)配置

您需要重新部署 HAProxy 服务才能使此处的配置变更生效。该部分的配置针对的是镜像 acs/proxy 所在服务的环境变量配置。

环境变量	默认值	描述
ADDITIONAL_SERVICES		需要进行负载均衡的附加服 务列表 (比如:prj1:web,

环境变量	默认值	描述
		prj2:sql)。服务发现将基于 com.docker.compose.[project  service] 容器标签。该环境变量 仅适用于 compose V2,且必须 确保容器可以解析和访问依赖 的服务所在的网络。
BALANCE	roundrobin	要使用的负载均衡算法。可 能的取值包括 roundrobin 、static-rr、source 和 leastconn。更多详细信 息,参见 <i>HAProxy#balance</i> 。
CA_CERT_FILE		CA 证书文件的路径。通过使 用该环境变量,您可以直接从 数据卷挂载 CA 证书文件,而 不需要通过环境变量传递证书 文件内容。如果您设置了该变 量,系统会忽略 CA_CERT 环 境变量。可设置为 / cacerts / cert0.pem。
CA_CERT		HAProxy 用于验证客户端的 CA 证书。该环境变量的格式与 DEFAULT_SSL_CERT 相同。
CERT_FOLDER		证书的路径。通过使用该变 量,您可以直接从数据卷挂 载 CA 证书文件,而不需要 通过环境变量传递证书文件 的内容。如果您设置了该变 量,系统会忽略所依赖服务 的 DEFAULT_SSL_CERT 和 SSL_CERT 环境变量。可设置 为 /certs/。
DEFAULT_SSL_CERT		默认的 SSL 证书。该 pem 文 件包含私钥和公钥(其中,私 钥在前,公钥在后),使用 \n (两个字符)作为行分隔符。 内容应该写在同一行,参见 SSL Termination。

环境变量	默认值	描述
EXTRA_BIND_SETTINGS		额外设置,由逗号进行分隔的 字符串( <port>:<setting &gt;)。每一部分会被附加到配置 文件中对应的端口绑定节点。 如果需要使用逗号,需要使用 进行转义。可设置为443: accept-proxy, 80:name http。</setting </port>
EXTRA_DEFAULT_SETTIN GS		额外设置,由逗号进行分隔的 字符串。每一部分会被附加到 配置文件中的 DEFAULT 节 点。如果需要使用逗号,需要 使用、,进行转义。
EXTRAFRONTEND_S ETTINGS\ <port></port>		额外设置,由逗号进行分隔 的字符串。每一部分会按照 环境变量名称中指定的端 口号被附加到前端节点。如 果需要使用逗号,需要使 用进行转义。可设置为 EXTRA_FRONTEND_SETTI NGS_80=balance source, maxconn 2000。
EXTRA_GLOBAL_SETTINGS		额外设置,由逗号进行分隔的 字符串。每一部分会被附加到 配置文件中的 GLOBAL 节点。 如果需要使用逗号,需要使用 进行转义。可设置为 tune .ssl.cachesize 20000 , tune.ssl.default-dh- param 2048。
EXTRA_ROUTE_SETTINGS		该字符串会在健康检查结束后 被附加到每一个后端路由上。 依赖的服务中的设置会覆盖该 设置。可设置为 send-proxy 。
EXTRA_SSL_CERTS		多余证书的名称列表,由逗号 分隔,例如 CERT1, CERT2,

环境变量	默认值	描述
		CERT3。您还需要将每一个证 书指定为单独的环境变量,例 如 CERT1=" <cert-body1>", CERT2="<cert-body2>", CERT3="<cert-body3>"。</cert-body3></cert-body2></cert-body1>
HEALTH_CHECK	check	在每一个路由上设置健康检 查。可设置为 "check inter 2000 rise 2 fall 3"。更多详细信 息,参见 <i>HAProxy#check</i> 。
HTTP_BASIC_AUTH		HTTP 基本认证的身份信息列 表,格式为 <user>: <pass &gt;,由逗号分隔。该身份信息适 用于所有后端路由。如果需要 使用逗号,需要使用进行转 义。注意:生产环境中,请不 要使用该环境变量进行认证。</pass </user>
MAXCONN	4096	设置每个进程的最大同时连接 数。
MODE	http	HAProxy 的负载均衡模式。 可能的值包括 http、tcp 和 health。
MONITOR_PORT		要添加 MONTIOR_URI 的端口 号。该变量同 MONTIOR_URI 配合使用。可设置为 80。
MONITOR_URI		为获取 HAProxy 的健康状态所 需要拦截的具体 URI。更多详 细信息,参见 <i>Monitor URI</i> 。可 设置为 /ping。
OPTION	redispatch	default 节点中的 HAProxy option 条目列表,由逗号分 隔。
RSYSLOG_DESTINATION	127.0.0.1	HAProxy 日志将要发送到的 rsyslog 目的地址。
SKIP_FORWARDED_PROTO		如果设置了该环境变量, HAProxy 将不会添加 X-

环境变量	默认值	描述
		Forwarded-For http请求头。 当 HAProxy 与其它负载均衡一 同使用时,可以使用该环境变 量。
SSL_BIND_CIPHERS		设置 SSL 服务器所要使用的 SSL 密钥套件。该环境变量设 置了 HAProxy ssl-default- bind-ciphers 的配置。
SSL_BIND_OPTIONS	no-sslv3	设置 SSL 服务器所使用的 SSL 绑定选项。该环境变量设置了 HAProxy ssl-default-bind -options 的配置。设置为默 认值则只允许在 SSL 服务器上 使用 TLSv1.0+。
STATS_AUTH	stats:stats	访问 Haproxy stats 统计页面所 需要的用户名和密码。
STATS_PORT	1936	HAProxy stats 统计页面暴露 的端口号。如果开放了该端 口,则可以通过 http://< host-ip>: <stats_port &gt;/访问 stats统计页面。</stats_port 
TIMEOUT	connect 5000, client 50000, server 50000	default 节点中 HAProxy timeout 条目列表,由逗号分 隔。

被代理的后端服务通过相应服务镜像的标签进行某一后端服务的配置

即通过将标签写到后端服务的镜像上来进行配置。该部分的配置写在被代理的服务的模版部分 此处的设置可以覆盖 HAProxy 的设置。HAProxy 的设置仅适用于依赖的服务。如果在阿里云容器 服务上运行,当服务重新部署,加入或者退出 HAProxy 服务时,HAProxy 服务会自动进行更新来 应用这些变更。

标签	描述
aliyun.proxy.APPSESSION	会话保持选项。可设置为 JSESSIONID len 52 timeout 3h。更多详细信息,参见
	HAProxy:appsession。

标签	描述
aliyun.proxy.BALANCE	要使用的负载均衡算法。可设置为 roundrobin、static-rr、source 和 leastconn。更多详细信息,参见 <i>HAProxy:</i> <i>balance</i> 。
aliyun.proxy.COOKIE	会话保持选项。可设置为 SRV insert indirect nocache。更多详细信息,参见 <i>HAProxy:cookie</i> 。
aliyun.proxy.DEFAULT_SSL_CERT	与 SSL_CERT 类似。但是该变量将 pem 文件保存在 /certs/cert0.pem 下作为默认的 SSL证书。如果在依赖的服务或者HAProxy 中设置了多个 DEFAULT_SSL_CERT, 会导致未定义的行为。
aliyun.proxy.EXCLUDE_PORTS	端口号,由逗号分隔(例如:3306,3307)。默 认情况下,HAProxy将应用服务暴露的所有端 口添加到后端路由。您可以指定您不希望进行路 由的端口,比如数据库端口。
aliyun.proxy.EXTRA_ROUTE_SETTINGS	该字符串会在健康检查结束后被附加到每一个后端路由上。可设置为"send-proxy"。
aliyun.proxy.EXTRA_SETTINGS	额外设置,由逗号分隔。每一部分会被附加到 配置文件中相关的后端节点或监听会话。如果 需要使用逗号,使用进行转义。可设置为 balance source。
aliyun.proxy.FORCE_SSL	如果设置了该环境变量且启用了 SSL terminatio n,HAProxy 会将 HTTP 请求重定向为 HTTPS 请求。
aliyun.proxy.GZIP_COMPRESSION_TYPE	启用 gzip 压缩。该环境变量的值为将要压缩的 MIME 类型列表。可设置为 text/html text /plain text/css。
aliyun.proxy.HEALTH_CHECK	在每一个后端路由上设置健康检查。可设置为 "check inter 2000 rise 2 fall 3"。更多详细信 息,参见 <i>HAProxy:check</i> 。
aliyun.proxy.HSTS_MAX_AGE	启用 HSTS。该环境变量的值为一个整数,代 表 HSTS 的有效期,单位为秒。可设置为 31536000。

标签	描述
aliyun.proxy.HTTP_CHECK	启用 HTTP 协议来检查服务器的健康情况。 可设置为 "OPTIONS * HTTP/1.1\r\nHost:\ www"。更多详细信息,参见 <i>HAProxy:httpchk</i> 。
aliyun.proxy.OPTION	HAProxy option 条目列表,由逗号分隔。此 处设置的 option 会被添加到相关的后端节 点或监听节点,并会覆盖 HAProxy 容器中的 OPTION 设置。
aliyun.proxy.SSL_CERT	SSL 证书。该 pem 文件包含私钥和公钥(其 中,私钥在前,公钥在后),使用 \n (两个字 符)作为行分隔符。
aliyun.proxy.TCP_PORTS	由逗号分隔的端口(比如:9000,9001,2222/ ssl)。TCP_PORTS 中列出的端口将在 TCP 模 式下被负载均衡。以 /ssl 结尾的端口表示该端 口需要 SSL termination。
aliyun.proxy.VIRTUAL_HOST_WEIGHT	虚拟主机的权重,同aliyun.proxy. VIRTUAL_HOST 配合使用。该值为一个整 数,默认值为0。该设置会影响虚拟主机的ACL 规则的顺序。虚拟主机的权重越高,ACL规则 的优先级越高,即值越大,越先被路由到。
aliyun.proxy.VIRTUAL_HOST	指定虚拟主机和虚拟路径。格式为[scheme ://]domain[:port][/path],。可以 在 domain 和 path 部分使用通配符 *。

有关以上内容的更多信息,参见HAProxy 配置手册。

# 虚拟主机和虚拟路径

您可以在 VIRTUAL_HOST 环境变量中同时指定虚拟主机和虚拟路径。该变量的值由 URL 组成,多 个 URL 之前用逗号分隔,格式为 [scheme://]domain[:port][/path]。

条目	默认值	描述
scheme	http	可能的值包括 http、https 和 wss。
domain		虚拟主机。可以使用通配符 *。

条目	默认值	描述
port	80/433	虚拟主机的端口号。当 scheme 设置为 https 或 wss 时,默认的端口为 443。
/path		虚拟路径。以 / 开头,可以使 用通配符 *。

# 匹配示例

虚拟主机	匹配	不匹配
http://domain.com	domain.com	www.domain.com
domain.com	domain.com	www.domain.com
domain.com:90	domain.com:90	domain.com
https://domain.com	https://domain.com	domain.com
https://domain.com:444	https://domain.com:444	https://domain.com
*.domain.com	www.domain.com	domain.com
*domain.com	www.domain.com、domain. com、anotherdomain.com	www.abc.com
www.e*e.com	www.domain.com、www. exxe.com	www.axxa.com
www.domain.*	www.domain.com、www. domain.org	domain.com
*	any website with HTTP	
https://*	any website with HTTPS	
*/path	<pre>domain.com/path、domain .org/path?u=user</pre>	domain.com/path/
*/path/	<pre>domain.com/path/、 domain.org/path/?u=user</pre>	domain.com/path、domain .com/path/abc
*/path/*	domain.com/path/、 domain.org/path/abc	domain.com/abc/path/
*/*/path/*	<pre>domain.com/path/、 domain.org/abc/path/、 domain.net/abc/path/123</pre>	domain.com/path

虚拟主机	匹配	不匹配
*/*.js	domain.com/abc.js、 domain.org/path/abc.js	domain.com/abc.css
*/*.do/	domain.com/abc.do/、 domain.org/path/abc.do/	domain.com/abc.do
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	domain.com/path/abc.php	domain/abc.php、domain. com/root/abc.php
*.domain.com/*.jpg	www.domain.com/abc.jpg 、abc.domain.com/123. jpg	domain.com/abc.jpg
<pre>*/path, */path/</pre>	domain.com/path、domain .org/path/	
domain.com:90、https:// domain.com	domain.com:90、https:// domain.com	



说明:

- 基于 VIRTUAL_HOST 所生成的 ACL 规则的顺序是随机的。在 HAProxy 中,当范围较窄的规则(比如 web.domain.com)被放置在范围较宽的规则(比如 *.domain.com)之后时,系统永远不会达到范围较窄的规则。因此,如果您所设置的虚拟主机包含相互重叠的范围时,您需要使用 VIRTUAL_HOST_WEIGHT 来手动设置 ACL 规则的顺序,为范围较窄的虚拟主机设置高于范围较宽的虚拟主机的权重。
- 所有带有相同 VIRTUAL_HOST 环境变量设置的服务将被看做并合并为一个服务。这对于一些测 试场景比较适用。

# **SSL** termination

acs/proxy 支持证书的 SSL ternimation。您只需要为每一个需要使用 SSL ternimation 的应用 设置 SSL_CERT 和 VIRTUAL_HOST 即可。然后, HAProxy 会读取依赖环境中的证书并开启 SSL ternimation。



当环境变量的值中包含等号(=)时(这种情况在 SSL_CERT中很普遍), Docker 会跳过该环境变量。因此,您只能在 Docker 1.7.0 或更高版本上使用多 SSL ternimation。

在以下情况下会启用 SSL ternimation:

- 至少设置了一个 SSL 证书。
- 没有设置 VIRTUAL_HOST 或者设置为使用 HTTPS 协议。

您可以通过以下方法设置 SSL 证书:

- 在 acs/proxy 中设置 DEFAULT_SSL_CERT。
- 在依赖于 acs/proxy 的应用服务中设置 aliyun.proxy.SSL_CERT 和/或 DEFAULT_SS L_CERT。

aliyun.proxy.SSL_CERT 和 DEFAULT_SSL_CERT 的区别在于 SSL_CERT 指定的多个证书保存为 cert1.pem, cert2.pem, ..., 而 DEFAULT_SSL_CERT 指定的证书通常保存为 cert0.pem。在这种情况下,当没有设置 SNI 匹配时, HAProxy 会将 cert0.pem 用作默认证书。然而,当提供了多个 DEFAULT_SSL_CERTIFICATE 时,仅有其中一个证书可以保存为 cert0.pem,其它的证书将被弃用。

#### PEM 文件

acs/proxy 或依赖的应用服务所指定的证书为一个 pem 文件。该文件包含私钥和公钥(其中,私 钥必须放在公钥和其它许可证书的前面)。您可以运行下面的脚本生成一个自签名证书。

openssl req -x509 -newkey rsa:2048 -keyout key.pem -out ca.pem -days 1080 -nodes -subj '/CN=*/O=My Company Name LTD./C=US' cp key.pem cert.pem cat ca.pem >> cert.pem

您获取 pem 文件之后,可以运行下面的命令将文件转化为一行内容。

awk 1 ORS='\\n' cert.pem

拷贝转换后的内容并将其设置为 aliyun.proxy.SSL_CERT 或 DEFAULT_SSL_CERT 的值。

### 亲和性和会话保持

您可以通过以下任一方法设置亲和性和会话保持。

- 在您的应用服务中设置 aliyun.proxy.BALANCE=source。当设置了 source 负载均衡方 法时, HAProxy 会对客户端 IP 地址进行哈希并确保同一个 IP 总是连接到同一个后端服务容器 上。
- 设置 aliyun.proxy.APPSESSION=<value>。使用应用会话来确定客户端应该连接到哪一个 服务容器上。<value> 可以设置为 JSESSIONID len 52 timeout 3h。
- 设置 aliyun.proxy.COOKIE=<value>。使用应用 cookie 来确定客户端应该连接到哪一个后端服务容器上。<value>可以设置为 SRV insert indirect nocache。

更多详细信息,参见 HAProxy:appsession 和 HAProxy:cookie。

#### TCP 负载均衡

默认情况下,acs/proxy运行在http模式下。如果您希望依赖的服务运行在tcp模式下,您可以指定TCP_PORTS环境变量。该变量的值为以逗号分隔的端口号(9000,9001)。

例如,如果您运行以下命令:

```
docker --name app-1 --expose 9000 --expose 9001 -e TCP_PORTS="9000,
9001" your_app
docker --name app-2 --expose 9000 --expose 9001 -e TCP_PORTS="9000,
9001" your_app
docker run --link app-1:app-1 --link app-2:app-2 -p 9000:9000, 9001:
9001 acs/proxy
```

HAProxy 将分别在 9000 端口和 9001 端口上对 app-1 和 app-2 进行负载均衡。

此外,如果您所暴露的端口多于 TCP_PORTS 中所设置的端口,剩余的端口将会使用 http 模式进行负载均衡。

例如,如果您运行以下命令:

```
docker --name app-1 --expose 9000 --expose 9001 -e TCP_PORTS="9000,
9001" your_app
docker --name app-2 --expose 9000 --expose 9001 -e TCP_PORTS="9000,
9001" your_app
docker run --link app-1:app-1 --link app-2:app-2 -p 9000:9000, 9001:
9001 acs/proxy
```

HAProxy 会在 80 端口使用 http 模式进行负载均衡,在 22 端口使用 tcp 模式进行负载均衡。

通过这种方法,您可以同时使用 tcp 模式和 http 模式进行负载均衡。

如果您在 TCP_PORTS 中设置了以 /ssl 结尾的端口,比如 2222/ssl, HAProxy 会在 2222 端口 上设置 SSL termination。

- 说明:

- 您可以同时设置 VIRTUAL_HOST 和 TCP_PORTS,以便为 http 模式提供更多的管控。
- tcp 端口上的负载均衡适用于所有的服务。因此,如果您依赖了两个或多个使用同一 TCP_PORTS 设置的不同服务, acs/proxy 会将这些服务看做同一个服务。

#### WebSocket 支持

您可以通过以下任一方法启用 websocket 支持:

- 鉴于 Websocket 使用 HTTP 协议进行启动,您可以使用虚拟主机来指定使用 ws 或 wss 协议。
   比如, -e VIRTUAL_HOST="ws://ws.domain.com, wss://wss.domain.com"。
- 鉴于 Websocket 本身为 TCP 连接,因此您可以尝试使用本文档前面所介绍的 TCP 负载均衡。

使用场景示例

📕 说明:

一下示例中的acs/proxy镜像均是registry.aliyuncs.com/acs/proxy:0.5的简写形

式,请做相应替换。

我的 webapp 容器暴露了 8080 端口(或任意其它端口),我希望 proxy 监听 80 端口

使用下面的命令:

docker run -d --expose 80 --name webapp dockercloud/hello-world docker run -d --link webapp:webapp -p 80:80 acs/proxy

我的 webapp 容器暴露了 80 端口和 8083/8086 数据库端口,我希望 proxy 监听 80 端口并且不希

#### 望将数据库端口加入 acs/proxy

```
docker run -d -e EXCLUDE_PORTS=8803,8806 --expose 80 --expose 8033 --
expose 8086 --name webapp dockercloud/hello-world
docker run -d --link webapp:webapp -p 80:80 acs/proxy
```

我的容器暴露了 8080 端口(或任意其它端口),我希望 acs/proxy 监听 8080 端口

使用下面的命令:

docker run -d --expose 8080 --name webapp your_app
docker run -d --link webapp:webapp -p 8080:80 acs/proxy

我希望 acs/proxy 处理 SSL 连接并将我的普通 HTTP 请求转发到端口 8080(或任意其它端口)

使用下面的命令:

docker run -d -e SSL_CERT="YOUR_CERT_TEXT" --name webapp dockercloud/ hello-world docker run -d --link webapp:webapp -p 443:443 -p 80:80 acs/proxy

### 或者

docker run -d --link webapp:webapp -p 443:443 -p 80:80 -e DEFAULT_SS L_CERT="YOUR_CERT_TEXT" acs/proxy

YOUR_CERT_TEXT 中的证书包含私钥和公钥(其中,私钥在前,公钥在后)。您需要在证书的行 与行之间加上 \n。假设您的证书保存在 ~/cert.pem 中,您可以运行以下命令。

docker run -d --link webapp:webapp -p 443:443 -p 80:80 -e DEFAULT_SS
L_CERT="\$(awk 1 ORS='\\n' ~/cert.pem)" acs/proxy

#### 我希望 acs/proxy 终结 SSL 连接并将 HTTP 请求重定向为 HTTPS 请求

使用下面的命令:

docker run -d -e FORCE_SSL=yes -e SSL_CERT="YOUR_CERT_TEXT" --name
webapp dockercloud/hello-world
docker run -d --link webapp:webapp -p 443:443 acs/proxy

我希望从数据卷中加载我的 SSL 证书,而不是通过环境变量来传递证书内容

您可以使用 CERT_FOLDER 环境变量来指定证书挂载在容器的哪个文件夹中。使用以下命令。

docker run -d --name webapp dockercloud/hello-world docker run -d --link webapp:webapp -e CERT_FOLDER="/certs/" -v \$(pwd)/ cert1.pem:/certs/cert1.pem -p 443:443 acs/proxy

我希望通过域名设置虚拟主机路由

可以通过 proxy 读取依赖的容器的环境变量(VIRTUAL_HOST)来配置虚拟主机。

示例:

docker run -d -e VIRTUAL_HOST="www.webapp1.com, www.webapp1.org" -name webapp1 dockercloud/hello-world
docker run -d -e VIRTUAL_HOST=www.webapp2.com --name webapp2 your/
webapp2
docker run -d --link webapp1:webapp1 --link webapp2:webapp2 -p 80:80
acs/proxy

在上面的例子中,当您访问 http://www.webapp1.com 或 http://www.webapp1.org 时,将显示运行于容器 webapp1 上的服务,http://www.webapp2.com 会连接到容器 webapp2。

如果您使用下面的命令:

```
docker run -d -e VIRTUAL_HOST=www.webappl.com --name webappl
dockercloud/hello-world
docker run -d -e VIRTUAL_HOST=www.webapp2.com --name webapp2-1
dockercloud/hello-world
```

```
docker run -d -e VIRTUAL_HOST=www.webapp2.com --name webapp2-2
dockercloud/hello-world
docker run -d --link webapp1:webapp1 --link webapp2-1:webapp2-1 --link
webapp2-2:webapp2-2 -p 80:80 acs/proxy
```

当您访问 http://www.webapp1.com 时,将显示运行于容器 webapp1 上的服务,http:// www.webapp2.com 将基于轮询调度算法(或者任何其它 BALANCE 指定的算法)连接到容器 webapp2-1 和 webapp2-2 上。

### 我希望我所有的 *.node.io 域名均指向我的服务

```
docker run -d -e VIRTUAL_HOST="*.node.io" --name webapp dockercloud/
hello-world
docker run -d --link webapp:webapp -p 80:80 acs/proxy
```

我希望 web.domain.com 连接到一个服务 , *.domain.com 连接到另外一个服务

```
docker run -d -e VIRTUAL_HOST="web.domain.com" -e VIRTUAL_HOST_WEIGHT=
1 --name webapp dockercloud/hello-world
docker run -d -e VIRTUAL_HOST="*.domain.com" -e VIRTUAL_HOST_WEIGHT=0
--name app dockercloud/hello-world
docker run -d --link webapp:webapp --link app:app -p 80:80 acs/proxy
```

#### 我希望所有到 /path 路径的请求均指向我的服务

```
docker run -d -e VIRTUAL_HOST="*/path, */path/*" --name webapp
dockercloud/hello-world
docker run -d --link webapp:webapp -p 80:80 acs/proxy
```

#### 我希望所有的静态 html 请求均指向我的服务

```
docker run -d -e VIRTUAL_HOST="*/*.htm, */*.html" --name webapp
dockercloud/hello-world docker run -d --link webapp:webapp -p 80:80
acs/proxy
```

#### 我希望查看 HAProxy stats

docker run -d --link webapp:webapp -e STATS_AUTH="auth:auth" -e STATS_PORT=1936 -p 80:80 -p 1936:1936 acs/proxy

#### 我希望将我的日志发送到 papertrailapp

用与您的 papertrailapp 账号相匹配的值替换 <subdomain> 和 <port>。

docker run -d --name webl dockercloud/hello-world docker run -d --name web2 dockercloud/hello-world

```
docker run -it --env RSYSLOG_DESTINATION='<subdomain>.papertrailapp.
com:<port>' -p 80:80 --link web1:web1 --link web2:web2 acs/proxy
```

### 使用虚拟主机的拓扑图

# 图 2-2: 阿里云容器服务代理拓扑图

```
|---- container_a1
|---- service_a ----- |---- container_a2
| (virtual host a) |---- container_a3
internet --- Server Load Balancer -- acs/proxy ----- |
| |---- container_b1
|---- service_b ----- |---- container_b2
(virtual host b) |---- container_b3
```

## 手动重新加载 HAProxy

在大多数情况下,当依赖的服务发生变化时,acs/proxy 会自动进行配置。但是如果需要的

话,您也可以按照下面的方法手动重新加载 acs/proxy。

docker exec <acs/proxy_container_id> /reload.sh

# 2.18.10 自定义路由-简单示例

此示例会部署一个acs/proxy 容器,对外通过负载均衡实例(使用/b标签)暴露服务,同时在后端 挂载一个 nginx,本示例只展示 nginx 的首页,但是会在基本示例的基础上,逐步增加其他功能。

📕 说明:

任何两个不同的服务均不能共享使用同一个负载均衡,否则会导致负载均衡后端机器被删除,服务不可用。

## 基本示例

compose 模板如下所示。

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   ports:
            '80:80'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
      aliyun.custom addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
      aliyun.global: "true"
       #
         前端绑定负载均衡
      aliyun.lb.port_80: tcp://proxy_test:80
   environment:
         支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       #
```

```
ADDITIONAL_SERVICES: "*"

appone:

expose: # 被代理的服务一定要使用 expose 或者 ports 告诉 proxy 容器暴露哪

个端口

- 80/tcp

image: 'nginx:latest'

labels:

# 此处支持 http/https/ws/wss 协议,必须使用用户自己的域名而不是容器服

务提供的测试域名

aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"

restart: always
```

服务启动成功后,页面如下图所示。

C (i) appone.example.com

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to <u>nginx.org</u>. Commercial support is available at <u>nginx.com</u>.

Thank you for using nginx.

开启会话保持

```
lb:
    image: registry.aliyuncs.com/acs/proxy:0.5
    ports:
              '80:80'
    restart: always
    labels:
        # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
        aliyun.custom_addon: "proxy"
        # 每台 vm 部署一个该镜像的容器
aliyun.global: "true"
        #
          前端绑定负载均衡
        aliyun.lb.port_80: tcp://proxy_test:80
    environment:
           支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
        #
        ADDITIONAL SERVICES: "*"
appone:
   ports:
        - 80/tcp
        _
          443/tcp
    image:
           'nginx:latest'
    labels:
        # 此处支持 http/https/ws/wss 协议
        aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
        # 此处支持会话保持,使用 cookie 的方式,键为 CONTAINERID
aliyun.proxy.COOKIE: "CONTAINERID insert indirect"
```

restart: always

#### 自定义 503 页面

当直接输入负载均衡的 VIP 地址而不是域名时,访问的页面如下图所示,返回 503 错误页面。

C () 118.178.32.24/?spm=5176.2020520152.221.10.AmcosM

# 503 Service Unavailable

No server is available to handle this request.

如果希望在 503 页面增加一些提示信息,可以在容器所在的 vm 中增加文件夹 / errors,同时增加文件 / errors / 503.http,文件内容如下:

HTTP/1.0 503 Service Unavailable Cache-Control: no-cache Connection: close Content-Type: text/html;charset=UTF-8 <html><body><h1>503 Service Unavailable</h1> <h3>No server is available to handle this request.</h3> <h3>如果您看到这个页面,说明您访问服务出现了问题,请按如下步骤排查解决问题:</h3> <1i>如果您是应用的访问者,请寻求应用的维护者解决问题。</1i> <1i>如果您是应用的维护者,继续查看下面的信息。</1i> <1i>您现在使用的是简单路由服务,整个请求会从SLB -> acsrouting应用容器 -> 您的应 用容器,请按下列步骤排查。</1i> <1i>清登录容器服务控制台,在左侧边栏选择"服务",在"服务列表"选择相应的"集群",单击 暴露到公网的"服务",查看服务的"访问端点",仔细检查您的访问域名与在相应的服务中配置的 域名是否一致。 <//ml>

</ ">简单路由服务链路排查</a>进行问题定位和排查。 <1i>查看<a href="https://help.aliyun.com/knowledge_detail/42658.html"</li> ">路由常见问题文档</a>。 <1i>如果上面的方式还没有解决您的问题,请<a href="https://workorder.console")</li> .aliyun.com/#/ticket/add?productId=1254">提交工单</a>. 联系技术人员协助解 决,我们会竭诚为您服务。 </body></html>

您可以修改为您需要展示的错误页面。修改 compose 模板如下:

```
aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global:
                      "true"
          前端绑定负载均衡
       #
       aliyun.lb.port_80: tcp://proxy_test:80
   environment:
       #
          支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       ADDITIONAL_SERVICES:
       EXTRA_FRONTEND_SETTINGS_80: "errorfile 503 /usr/local/etc/
haproxy/errors/503.http"
   volumes:
       - /errors/:/usr/local/etc/haproxy/errors/
appone:
   ports:
          80/tcp
       -
          443/tcp
   image: 'nginx:latest'
   labels:
       # 配置 URL,支持指定 path,此处支持 http/https/ws/wss 协议
       aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
   restart: always
```

输入负载均衡的 VIP 地址之后,显示的 503 页面如下图所示。

C 0 118.178.32.24/?spm=5176.2020520152.221.10.AmcosM	🖈 🚇 📕 😂 🖆 🗄
503 Service Unavailable	
No server is available to handle this request.	
如果您看到这个页面,说明您访问服务出现了问题,请按如下步骤排查解决问题:	
如果您是应用的访问者,请寻求应用的维护者解决问题。 • 如果您是应用的维护者,继续查看下面的信息。 • 您现在使用的是简单路由服务,整个请求会从SLB > acsrouting应用容器 > 您的应用容器,请按下列步骤排查。 • 请登陆容器服务控制台,在左侧边栏选择"服务",在"服务列表"选择相应的"集群",点击暴露到公网的"服务",查看服务的"访问端点",仔细检查您的访问域名与在相应的 或。 • 按照 <u>简单路由服务链路推查进</u> 行问题定位和排查。 • 查看 <u>路由常见问题文档</u> 。 • 如果上面的方式还没有解决您的问题,请 <u>提交工单</u> ,联系技术人员协助解决,我们会竭诚为您服务。	的服务中配置的域名是否一

## 支持泛域名

如果希望 nginx 的后端支持泛域名,即 appone.example.com 能访问到 nginx 首页,*.

example.com 也能访问到 nginx 的首页,修改配置如下即可。

```
lb:
          registry.aliyuncs.com/acs/proxy:0.5
   image:
   ports:
            '80:80'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
       aliyun.custom_addon:
                          "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global: "true"
          前端绑定负载均衡
       #
       aliyun.lb.port_80: tcp://proxy_test:80
   environment:
```

```
支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       #
                           "*"
       ADDITIONAL_SERVICES:
       EXTRA_FRONTEND_SETTINGS_80: "errorfile 503 /usr/local/etc/
haproxy/errors/503.http"
   volumes:
       - /errors/:/usr/local/etc/haproxy/errors/
appone:
   ports:
       - 80/tcp
       - 443/tcp
   image: 'nginx:latest'
   labels:
       # 配置 URL,支持指定 path,此处支持 http/https/ws/wss 协议
       aliyun.proxy.VIRTUAL_HOST: "http://*.example.com"
   restart: always
```

绑定 host, 输入域名 www.example.com 之后,显示 nginx 首页如下图所示。

$\leftrightarrow \ \ni \ G$	www.example.com/?backend=appone	
		Welcome to nginx!
		If you see this page, the nginx web server is successfully installed and working. Further configuration is required.
		For online documentation and support please refer to <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .
		Thank you for using nginx.

配置默认的后端

如果希望直接通过 IP 也能访问到后端的 nginx,把 URL 配置去掉,修改配置如下即可。

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   ports:
            '80:80'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
       aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global: "true"
         前端绑定负载均衡
       #
       aliyun.lb.port_80: tcp://proxy_test:80
   environment:
          支持加载路由的后端容器的范围,"*" 表示整个集群,默认为应用内的服务
       #
       ADDITIONAL_SERVICES:
       # 返回 503 时,指定错误页面
       EXTRA_FRONTEND_SETTINGS_80: "errorfile 503 /usr/local/etc/
haproxy/errors/503.http"
   volumes:
       # 从主机挂载错误页面到容器内部
       - /errors/:/usr/local/etc/haproxy/errors/
appone:
```

```
ports:
    - 80/tcp
    - 443/tcp
image: 'nginx:latest'
labels:
    # 表明该服务需要通过 proxy 来代理
    aliyun.proxy.required: "true"
restart: always
```

输入负载均衡的 VIP 之后,显示 nginx 首页如下图所示。

```
C ● 118.178.32.24/?spm=5176.2020520152.221.10.AmcosM
Welcome to nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.
Thank you for using nginx.
```

根据 URL 参数值选择后端

您也可以根据 URL 参数值的不同来代理不同的后端。

以下示例将通过 http://www.example.com?backend=appone 访问服务 appone,即 nginx 首

- 页,通过http://www.example.com?backend=apptwo访问服务 apptwo,即 hello world 首
- 页。应用模板代码如下所示。

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   ports:
          - '80:80'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力,动态加载服务的路由
       aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global: "true"
         前端绑定负载均衡
       #
       aliyun.lb.port_80: tcp://proxy_test:80
   environment:
       #
         支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       ADDITIONAL_SERVICES:
                          " * "
       # 获取 URL 中参数名称为 backend 的参数值,同时将 HOST header 修改为
要匹配的后端的域名
       EXTRA_FRONTEND_SETTINGS_80: " http-request set-header HOST
%[urlp(backend)].example.com"
appone:
   ports:
         80/tcp
         443/tcp
```

```
'nginx:latest'
   image:
   labels:
           配置 URL,支持指定 path,此处支持 http/https/ws/wss 协议
       #
       aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
   restart: always
apptwo:
   ports:
       - 80/tcp
   image: 'registry.cn-hangzhou.aliyuncs.com/linhuatest/hello-world:
latest'
   labels:
       #
           配置 URL,支持指定 path,此处支持 http/https/ws/wss 协议
       aliyun.proxy.VIRTUAL_HOST: "http://apptwo.example.com"
   restart: always
```

绑定 host,输入链接 http://www.example.com?backend=appone,显示服务 appone 的 nginx 首页,如下所示。

```
← → C www.example.com
Welcome to nginx web server is successfully installed and working. Further configuration is required.
For online documentation and support please refer to nginx.org. Commercial support is available at nginx.com.
Thank you for using nginx.
```

绑定 host,输入链接 http://www.example.com?backend=apptwo,显示服务 apptwo 的 hello world 首页,如下所示。

←→℃ () www.example.com/?backend=apptwo
Hello world!
My hostname is 11614a37b5a3-custom-routing-apptwo-1
记录访问日志

```
labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
       aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器 aliyun.global: "true"
          前端绑定负载均衡
       #
       aliyun.lb.port_80: tcp://proxy_test:80
    environment:
          支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       #
       ADDITIONAL_SERVICES:
       EXTRA_DEFAULT_SETTINGS: "log rsyslog local0,log global,option
httplog"
   links:
        - rsyslog:rsyslog
rsyslog:
   image: registry.cn-hangzhou.aliyuncs.com/linhuatest/rsyslog:latest
appone:
   ports:
       - 80/tcp
       - 443/tcp
   image: 'nginx:latest'
   labels:
       # 此处支持 http/https/ws/wss 协议
       aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
   restart: always
```

日志会直接打到 rsyslog 容器的标准输出中,通过 docker logs \$rsyslog_container_name 即能看到自定义路由的访问日志。

服务间的负载均衡

以下模板创建一个负载均衡服务 1b 和一个应用服务 appone,整体对外提供域名为 appone. example.com 的服务。

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   hostname: proxy # 指定该服务的域名为 proxy,即 proxy 会解析到所有部署了
该镜像的容器
   ports:
             '80:80'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力,动态加载服务的路由
       aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global: "true"
       #
         前端绑定负载均衡
       aliyun.lb.port_80: tcp://proxy_test:80
   environment:
         支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       ADDITIONAL_SERVICES:
appone:
   ports:
         80/tcp
         443/tcp
   image: 'nginx:latest'
```

```
labels:
    # 此处支持 http/https/ws/wss 协议
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
restart: always
```

以下模板作为一个客户端,访问应用服务 appone,但是它的访问路径是请求访问负载均衡服务 1b

,然后再反向代理到应用服务 appone 的。

```
restclient: # 模拟 rest 服务消费者
    image: registry.aliyuncs.com/acs-sample/alpine:3.3
    command: "sh -c 'apk update; apk add curl; while true; do curl --
head http://appone.example.com; sleep 1; done'" #访问 rest 服务,测试负载
均衡
    tty: true
    external_links:
        - "proxy:appone.example.com" #指定 link 的服务的域名,以及该域名的别名
```

在服务 restclient 的容器中,您会发现域名 appone.example.com 是解析到负载均衡服务 lb 的所有容器 IP 的。

```
/ # drill appone.example.com
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 60917
;; flags: qr rd ra ; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; appone.example.com.
                        IN
                                Α
;; ANSWER SECTION:
appone.example.com.
                        600
                                IN
                                        А
                                                 172.18.3.4
                                                 172.18.2.5
appone.example.com.
                        600
                                IN
                                        А
appone.example.com.
                                                 172.18.1.5
                        600
                                IN
                                        Α
;; AUTHORITY SECTION:
;; ADDITIONAL SECTION:
;; Query time: 0 msec
;; SERVER: 127.0.0.11
;; WHEN: Mon Sep 26 07:09:40 2016
;; MSG SIZE rcvd: 138
```

```
配置监控页面示例
```

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   ports:
             '80:80'
            '127.0.0.1:1935:1935' # 监控页面对公网暴露的端口,有安全风
险,请谨慎配置
   restart: always
   labels:
       aliyun.custom_addon:
                            "proxy"
       aliyun.global: "true"
       aliyun.lb.port_80: tcp://proxy_test:80
   environment:
       ADDITIONAL_SERVICES:
                           " * "
       STATS_AUTH: "admin:admin" # 监控时用于登录的账号和密码,可以自定义
       STATS_PORT: "1935" # 监控使用的端口,可以自定义
appone:
   expose:
```

```
- 80/tcp
image: 'nginx:latest'
labels:
    aliyun.proxy.VIRTUAL_HOST: "http://appone.example.com"
restart: always
```

登录到自定义路由镜像所在的每一台机器(每一台机器都可能接收请求,不管应用容器在哪台机器上),请求 acs/proxy健康检查页面。



按照应用模版的 STATS_AUTH 环境变量配置正确的用户名和密码。

```
root@c68a460635b8c405e83c052b7c2057c7b-node2:~# curl -Ss -u admin:
admin 'http://127.0.0.1:1935/' &> test.html
```

将页面 test.html 拷贝到有浏览器的机器,用浏览器打开本地文件 test.html,查看 stats 监控统计页面,显示为绿色,表示 acs/proxy 容器到后端容器的网络是连通的,在正常工作;显示为其他颜色则为异常。

# 2.18.11 自定义路由-支持 TCP 协议

阿里云容器服务在使用的过程中,针对 TCP 负载均衡的场景,会遇到这样的问题:如果一个应用的客户端镜像和服务端镜像均部署在同一个节点(ECS)上面,由于受负载均衡的限制,该应用的客户端不能通过负载均衡访问本机的服务端。本文档以常用的基于 TCP 协议的 redis 为例,通过自定义路由acs/proxy 来解决这一问题。

任何两个不同的服务均不能共享使用同一个负载均衡,否则会导致负载均衡后端机器被删除,服务不可用。

解法一:通过调度容器,避免客户端和服务端容器部署在同一个节点

示例应用模板 (使用了lb 标签和 swarm filter 功能):

```
redis-master:
    ports:
        - 6379:6379/tcp
    image: 'redis:alpine'
    labels:
        aliyun.lb.port_6379: tcp://proxy_test:6379
redis-client:
    image: 'redis:alpine'
    links:
        - redis-master
    environment:
        - 'affinity:aliyun.lb.port_6379!=tcp://proxy_test:6379'
```

```
command: redis-cli -h 120.25.131.64
stdin_open: true
tty: true
```

■ 说明:

如果发现调度不生效,在容器服务管理控制台,单击左侧导航栏的服务进入服务列表页面,选择您需要调度的服务,单击重新调度>在弹出的对话框中勾选强制重新调度,单击确定。

• 强制重新调度会丢弃已有容器的 volume,请做好相应的备份迁移工作。

解法二:容器集群内部客户端使用 link 访问服务端,集群外部使用负载均衡

示例应用模板 (使用了*lb*标签):

```
redis-master:
    ports:
        - 6379:6379/tcp
    image: 'redis:alpine'
    labels:
        aliyun.lb.port_6379: tcp://proxy_test:6379
redis-client:
        image: 'redis:alpine'
        links:
            - redis-master
        command: redis-cli -h redis-master
        stdin_open: true
        tty: true
```

解法三:容器集群内部客户端使用自定义路由(基于 HAProxy)作为代理访问服务端,集群外部使用负载均衡

示例应用模板(使用了*lb*标签和自定义路由-简单示例):

```
lb:
   image: registry.aliyuncs.com/acs/proxy:0.5
   ports:
             '6379:6379/tcp'
   restart: always
   labels:
       # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由
       aliyun.custom_addon: "proxy"
       # 每台 vm 部署一个该镜像的容器
       aliyun.global: "true"
       # 前端绑定负载均衡,使用 lb 标签
aliyun.lb.port_6379: tcp://proxy_test:6379
       # 告诉系统,自定义路由需要等待 master 和 slave 启动之后再启动,并且对
master 和 slave 有依赖
       aliyun.depends: redis-master, redis-slave
   environment:
          支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务
       #
       ADDITIONAL_SERVICES:
       EXTRA_DEFAULT_SETTINGS: "log rsyslog local0,log global,option
httplog"
```

```
# 配置 HAProxy 工作于 tcp 模式
       MODE: "tcp"
   links:
       - rsyslog:rsyslog
rsyslog:
   image: registry.cn-hangzhou.aliyuncs.com/linhuatest/rsyslog:latest
redis-master:
   ports:
     - 6379/tcp
   image: 'redis:alpine'
   labels:
       # 告诉自定义路由需要暴露 6379 端口
       aliyun.proxy.TCP_PORTS: "6379"
       # 告诉系统,该服务的路由需要添加到自定义路由服务中
       aliyun.proxy.required: "true"
redis-slave:
   ports:
     - 6379/tcp
   image: 'redis:alpine'
   links:

    redis-master

   labels:
     # 告诉自定义路由需要暴露 6379 端口
     aliyun.proxy.TCP_PORTS: "6379"
     # 告诉系统,该服务的路由需要添加到自定义路由服务中
     aliyun.proxy.required: "true"
     # 告诉系统, slave 需要等待 master 启动之后再启动, 并且对 master 有依赖
     aliyun.depends: redis-master
   command: redis-server --slaveof redis-master 6379
redis-client:
   image:
          'redis:alpine'
   links:
     - lb:www.example.com
   labels:
     aliyun.depends: lb
   command: redis-cli -h www.example.com
   stdin open: true
   tty: true
```

该解决方案,做到了 redis 的主从架构,同时经过服务间的负载均衡做负载均衡,做到了一定程度的高可用。

# 2.18.12 自定义路由-支持多 HTTPS 证书

本示例使用的镜像是acs/proxy 镜像。

▋ 说明:

任何两个不同的服务均不能共享使用同一个负载均衡,否则会导致负载均衡后端机器被删除,服务不可用。

- '443:443' # https 必须暴露这个端口 restart: always labels: # addon 使得 proxy 镜像有订阅注册中心的能力, 动态加载服务的路由 aliyun.custom_addon: "proxy" # 每台 vm 部署一个该镜像的容器 "true" aliyun.global: 前端绑定负载均衡 # aliyun.lb.port_80: tcp://proxy_test:80 aliyun.lb.port_443:tcp://proxy_test:443 environment: 支持加载路由的后端容器的范围,"*"表示整个集群,默认为应用内的服务 # ADDITIONAL_SERVICES: " * " appone: # 被代理的服务一定要使用 expose 或者 ports 告诉 proxy 容器暴露哪 expose: 个端口 80/tcp image: 'nginx:latest' labels: 配置 URL,支持指定 path,此处支持 http/https/ws/wss 协议 # aliyun.proxy.VIRTUAL_HOST: "https://appone.example.com" 配置 appone 的证书 # aliyun.proxy.SSL CERT: "----BEGIN RSA PRIVATE KEY----\ nMIIEpQIBAAKCAQEAvqnKhephWHKWYDEiBiSjzst7nRP0DJxZ5cIOxyXmncd2kslr \nkUIB5qT/MSiJGBL3Lr4advs6kI/JFmxloFrPtwEe2FGkLBfCDXXDrWgxyFhbuPQY \nBLNueUu94sffIxg+4u5Mriui7ftindOAf0d21PSM9gb/ZUypxIgAd3RHCe/gtT0h \nVCn6FikXynXLDTODYWCthOHBwSZS88HNU+B0T9Y165Ji00mV+YF+h3D/c232E6Gp \nzK+8ehVB13s5hecUx3dvdUQPBUhJYvzsPjChqsXSMDRexiN66kbhH6dJArsrYb8t \nEBWXfCZaTcF82wkAsUe/fhlGhh97h+661h6000IDA0ABA0IBA0C4d8ifNWRI9vIB \nbbAZRne7xMm5MCU2GI8q97Rqm+nAPl5bHinMVsaBnKqaj76EH+TQ+re1xyiSKwCH \nQ7FidsQqYGwQjy9NncJATpAjQ4EPeLWQU2D9Ly+NjnhEKr/u0Ro6LhdA+hqt59dS \nXHvfEP/It5odN62yJzikDWBmk/hhK0tu28dPYUuPoWswXWFMkaNttmfLgZlagiqr \nYp7rxAFqQurzctQ2VNwezekDHQoh8ounHGEniZ+fA6sFtYi83KTKWkvFom1chZQr \nxxPbbgANJJJlNgtkl6JZNxj6SYimmWvzmrrU25khKg/klP5EtQzIx6UFhURnuTKu \nzNgqcIABAoGBAOqUOerveEUePvsAlta8CV/p2KKwenv+kUofQ4UpKFXfnHbQHQfr \nZHS290QiPxqjVXYLu8gNfLRfKtUNyqV+TDrzJ1elW2RKc00GHAwPbXxijPhmJ2fW \neskn8tlDcyXpvoqWJG34896vo4IbcL0H/eUs0jJo60JlCQBKXik+t3gxAoGBAM9k \nVOTV2caKyrZ4ta0Q1LKqKfOkt0j+vKz167J5pSLjVKQSUxGMyLnGwiQdDtB4iy6L \nFcCB/S0HM0UWkJWhNYAL8kHry53bVdHtQG0tuYFYvBJo7A+Nppsn9MtlVh8KbVu4 \nhOz/3MWwbQNnvIVCGK/fSltS1GhTk4rKL7PjNwMRAoGBALK0n3bqXj6Rrzs7FK6c \na6vlE4PFXFpv8jF8pcyhMThSdPlSzHsHCe2cn+3YZSie+/FFORZLqBAlXBUZP6Na \nFyrlqLgtofVCfppUKDPL4QXccjaeZDDIBZyPUYPQzb05WE5t2WzqNqcUOUVaMEXh \n+7uGrM94espWXEgbX6aeP91RAoGAR1JQ7t8MXuQE5GZ9w9cnKAXG/9RkSZ4Gv+cL \nKpNQyUmoE51bFKJWFZgtkC1CLrIRD5EdqQ7q1/APFGgYUoQ9LdPfKzcW7cnHic0W \nwW51rkQ2UU++a2+uhIHB4Y3U6+WP00CP4gTICUhPTo5IQC8vS8M85UZqu41LRA5W \nqnpq1uECgYEAq+6KpHhlR+5h3Y/m0n84yJ0YuCmr17HFRzBMdOcaW3oaYL83rAaq \n6dJqpAVgeu3HP8AtiGVZRe78J+n4d2JGYSqgtP2lFFTdF9HfhcR2P9bUBNYtWols \nEs3iw53t8a4BndLGBwLPA31klf7J5stYanRv6NqaRaLq4FQMxsW1A0Q=\n----END RSA PRIVATE KEY----\n---BEGIN CERTIFICATE----\nMIIDvDCCA qSgAwIBAgIBATANBgkqhkiG9w0BAQUFADBgMQswCQYDVQQGEwJDTjER\nMA8GA1UEC BMIWmhlamlhbmcxETAPBgNVBAcTCEhhbmd6aG91MRQwEgYDVQQKEwth\nbGliYWJhL  $mNvbTeVMBMGA1UeCxMMd3d3LnJvb3QuY29tMB4XDTe1MDIwOTA1MzQx \ nOFoXDTe2MDIwOTA1MzQx \ nOFoXDIWOTA1MzW \ nOFOXDIWOTAM$ DIwOTA1MzQxOFowZjELMAkGA1UEBhMCQ04xETAPBgNVBAgTCFpoZWpp\nYW5nMRQwE gYDVQQKEwthbGliYWJhLmNvbTEVMBMGA1UECxMMd3d3LnJvb3QuY29t\nMRcwFQYDV QQDEw53d3cubGluaHVhLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEP\nADCCAQoCg gEBAL4JyoXqYVhylmAxIgYko87Le50T9AycWeXCDscl5p3HdpLJa5FC\nAeak/ zEoiRgS9y6+Gnb70pCPyRZsZaBaz7cBHthRpCwXwg11w61oMchYW7j0GASz\nbnlLveLH3 yMYPuLuTK4rou37Yp3TgH9HdtT0jPYG/2VMqcSIAHd0Rwnv4LU9IVQp\n+hYpF8p1yw0 zg2FgrYUBwcEmUvPBzVPgdE/WJeuSYkNJlfmBfodw/3Nt9hOhqcyv\nvHoVQdd70 YXnFMd3b3VEDwVISWL87D4woYLF0jA0XsYjeupG4R+nSQK7K2G/LRAV\nl3wmWk3Bf

NsJALFHv34ZRoYfe4fuupYejkECAwEAAaN7MHkwCQYDVR0TBAIwADAs\nBglghkgBh vhCAQ0EHxYdT3BlblNTTCBHZW51cmF0ZWQgQ2VydGlmaWNhdGUwHQYD\nVR0OBBYEF M6ESmkDKrqnqMwBawkjeONKrRMQMB8GA1UdIwQYMBaAFFUrhN9ro+Nm\nrZn14WQzD pgTbCBhMA0GCSqGSIb3DQEBBQUAA4IBAQCQ2D9CRiv8brx3fnr/RZG6\nFYPEdxjY/ CyfJrAbij0PdKjzZKk1067chM10xs2JhJ6tMqg2sv50bGx4XmbSPmEe\nYTJjIXMY+jCoJ /Zmk3Xgu4K1y1LvD25PahDVhRrPN8H4WjsYu51pQNshil5E/3iQ\n2JoV0r8QiAsPiiY5 +mNCD1fm+QN1tyUabczi/DHafgWJxf2B3M66e3oUdtbzA2pf\nYHR8RVeSFrjaBqud08i r+uYcRbRkroYmY5Vm+4Yp64oetrPpKUPWSYaAZ0uRtpeL\nB5DpqXz9GEBb5m2Q4dK js5Hm6vyFUORCzZcO4XexDhcgdLOH5qznmh9oMCk9QvZf\n----END CERTIFICATE ----\n"

restart: always

apptwo:

expose: # 被代理的服务一定要使用 expose 或者 ports 告诉 proxy 容器暴露哪 个端口

- 80/tcp

image: 'registry.cn-hangzhou.aliyuncs.com/linhuatest/hello-world: latest'

labels:

# 配置 URL,支持指定 path,此处支持 http/https/ws/wss 协议

aliyun.proxy.VIRTUAL_HOST: "https://apptwo.example.com"

# 配置 apptwo 的证书

aliyun.proxy.SSL_CERT: "----BEGIN RSA PRIVATE KEY----\ nMIIEpQIBAAKCAQEAvgnKhephWHKWYDEiBiSjzst7nRP0DJxZ5cIOxyXmncd2kslr \nkUIB5qT/MSiJGBL3Lr4advs6kI/JFmxloFrPtwEe2FGkLBfCDXXDrWgxyFhbuPQY \nBLNueUu94sffIxg+4u5Mriui7ftindOAf0d21PSM9gb/ZUypxIgAd3RHCe/gtT0h \nVCn6FikXynXLDTODYWCthQHBwSZS88HNU+B0T9Y165JiQ0mV+YF+h3D/c232E6Gp \nzK+8ehVB13s5hecUx3dvdUQPBUhJYvzsPjChgsXSMDRexiN66kbhH6dJArsrYb8t \nEBWXfCZaTcF82wkAsUe/fhlGhh97h+661h6000IDA0ABA0IBA0C4d8ifNWRI9vIB \nbbAZRne7xMm5MCU2GI8q97Rqm+nAPl5bHinMVsaBnKqaj76EH+TQ+re1xyiSKwCH \nQ7FidsQqYGwQjy9NncJATpAjQ4EPeLWQU2D9Ly+NjnhEKr/u0Ro6LhdA+hqt59dS \nXHvfEP/It5odN62yJzikDWBmk/hhK0tu28dPYUuPoWswXWFMkaNttmfLgZlagiqr \nYp7rxAFqQurzctQ2VNwezekDHQoh8ounHGEniZ+fA6sFtYi83KTKWkvFom1chZQr \nxxPbbgANJJJlNgtkl6JZNxj6SYimmWvzmrrU25khKg/klP5EtQzIx6UFhURnuTKu \nzNgqcIABAoGBAOqUOerveEUePvsAlta8CV/p2KKwenv+kUofQ4UpKFXfnHbQHQfr \nZHS290QiPxqjVXYLu8gNfLRfKtUNyqV+TDrzJ1e1W2RKc00GHAwPbXxijPhmJ2fW \neskn8tlDcyXpvoqWJG34896vo4IbcL0H/eUs0jJo6OJlCQBKXik+t3gxAoGBAM9k \nVOTV2caKyrZ4ta0Q1LKqKfOkt0j+vKz167J5pSLjVKQSUxGMyLnGwiQdDtB4iy6L \nFcCB/S0HM0UWkJWhNYAL8kHry53bVdHtQG0tuYFYvBJo7A+Nppsn9MtlVh8KbVu4 \nhOz/3MWwbQNnvIVCGK/fSltSlGhTk4rKL7PjNwMRAoGBALK0n3bqXj6Rrzs7FK6c \na6vlE4PFXFpv8jF8pcyhMThSdPlSzHsHCe2cn+3YZSie+/FF0RZLqBAlXBUZP6Na \nFyrlqLgtofVCfppUKDPL4QXccjaeZDDIBZyPUYPQzb05WE5t2WzqNqcUOUVaMEXh \n+7uGrM94espWXEqbX6aeP91RAoGAR1JQ7t8MXuQE5GZ9w9cnKAXG/9RkSZ4Gv+cL \nKpNQyUmoE51bFKJWFZgtkC1CLrIRD5EdqQ7q1/APFGgYUoQ9LdPfKzcW7cnHic0W \nwW51rkQ2UU++a2+uhIHB4Y3U6+WP00CP4gTICUhPTo5IQC8vS8M85UZqu41LRA5W \nqnpq1uECgYEAq+6KpHhlR+5h3Y/m0n84yJ0YuCmrl7HFRzBMdOcaW3oaYL83rAaq \n6dJqpAVqeu3HP8AtiGVZRe78J+n4d2JGYSqqtP21FFTdF9HfhcR2P9bUBNYtWols nEs3iw53t8a4BndLGBwLPA31klf7J5stYanRv6NqaRaLq4FQMxsW1A0Q=<math>n---END RSA PRIVATE KEY----\n---BEGIN CERTIFICATE----\nMIIDvDCCA qSqAwIBAqIBATANBgkqhkiG9w0BAQUFADBgMQswCQYDVQQGEwJDTjER\nMA8GA1UEC BMIWmhlamlhbmcxETAPBgNVBAcTCEhhbmd6aG91MRQwEgYDVQQKEwth\nbGliYWJhL mNvbTeVMBMGA1UECxMMd3d3LnJvb3QuY29tMB4XDTE1MDIwOTA1MzQx\nOFoXDTE2M DIwOTA1MzQxOFowZjELMAkGA1UEBhMCQ04xETAPBgNVBAgTCFpoZWpp\nYW5nMRQwE gYDVQQKEwthbGliYWJhLmNvbTEVMBMGA1UECxMMd3d3LnJvb3QuY29t\nMRcwFQYDV QQDEw53d3cubGluaHVhLmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEP\nADCCAQoCg gEBAL4JyoXqYVhylmAxIgYko87Le50T9AycWeXCDscl5p3HdpLJa5FC\nAeak/ zEoiRgS9y6+Gnb70pCPyRZsZaBaz7cBHthRpCwXwg11w61oMchYW7j0GASz\nbnlLveLH3 yMYPuLuTK4rou37Yp3TgH9HdtT0jPYG/2VMqcSIAHd0Rwnv4LU9IVQp\n+hYpF8p1yw0 zg2FgrYUBwcEmUvPBzVPgdE/WJeuSYkNJlfmBfodw/3Nt9hOhqcyv\nvHoVQdd70 YXnFMd3b3VEDwVISWL87D4woYLF0jA0XsYjeupG4R+nSQK7K2G/LRAV\nl3wmWk3Bf NsJALFHv34ZRoYfe4fuupYejkECAwEAAaN7MHkwCQYDVR0TBAIwADAs\nBglghkgBh vhCAQ0EHxYdT3BlblNTTCBHZW51cmF0ZWQgQ2VydGlmaWNhdGUwHQYD\nVR00BBYEF M6ESmkDKrqnqMwBawkjeONKrRMQMB8GA1UdIwQYMBaAFFUrhN9ro+Nm\nrZn14WQzD

pgTbCBhMA0GCSqGSIb3DQEBBQUAA4IBAQCQ2D9CRiv8brx3fnr/RZG6\nFYPEdxjY/ CyfJrAbij0PdKjzZKk1067chM10xs2JhJ6tMqg2sv50bGx4XmbSPmEe\nYTJjIXMY+jCoJ /Zmk3Xgu4K1y1LvD25PahDVhRrPN8H4WjsYu51pQNshi15E/3iQ\n2JoV0r8QiAsPiiY5 +mNCD1fm+QN1tyUabczi/DHafgWJxf2B3M66e3oUdtbzA2pf\nYHR8RVeSFrjaBqud08i r+uYcRbRkroYmY5Vm+4Yp64oetrPpKUPWSYaAZ0uRtpeL\nB5DpqXz9GEBb5m2Q4dK js5Hm6vyFUORCzZcO4XexDhcgdLOH5qznmh9oMCk9QvZf\n----END CERTIFICATE ----\n" restart: always

如上所示,有两个服务 appone 和 apptwo 分别通过 aliyun.proxy.VIRTUAL_HOST 来指定二者 的域名,注意如果需要配置证书,一定要注明https协议。然后通过 aliyun.proxy.SSL_CERT 来 指定各自的证书内容,证书内容配置的方法如下。

假设 key.pem 是私钥文件,ca.pem是公钥文件,在 bash 中执行如下命令(当前目录包含公钥文件和私钥文件)。

```
$ cp key.pem cert.pem
$ cat ca.pem >> cert.pem
$ awk 1 ORS='\\n' cert.pem
```

最后将 awk 指令的输出作为 label aliyun.proxy.SSL_CERT 值填入,注意用英文双引

号("")分隔。其他内容,例如lb标签等,参考上述模版和相应自定义路由-简单示例。

# 2.19 发布策略

# 2.19.1 发布策略说明

容器服务提供两种发布策略:蓝绿发布和标准发布。两种发布策略的差异在于:

发布策略	区别
标准发布	在部署新版本应用时删除旧版本应用。发布过程 中,您的服务会出现短暂中断。
蓝绿发布	应用更新时生成蓝绿两个版本,两个版本互相热 备,通过切换路由权重的方式实现不同版本应用 的上下线。该发布策略具有以下特点:-服务不 会中断-应用零宕机-发布过程用户无感知重启- 升级失败自动回滚-同一资源栈可以多次升级迭 代
	使用镜像创建的应用,在变更应用配置时无法选择蓝绿发布。
#### 标准发布使用场景

标准发布是传统的应用发布策略,在现有环境中,部署新版本应用。除了有特殊需求以外,一般采 用此类发布策略,使用场景包括:

- 新版本应用已经过充分测试,无需线上试运行。
- 应用包含数据库服务,且发生过如数据表结构变更等不可逆转的升级情况,标准发布可以免除类 似数据迁移、回滚等业务改造的工作。

#### 蓝绿发布使用场景

蓝绿发布适用的应用形态包括前端类服务和后端类服务,常用于应用的增量升级中,面向客户的具体业务场景包括:

保证业务连续性

对业务变化迅速的互联网企业,应用版本迭代频繁是一个现实的挑战,保证线上业务的连续性是必要的。蓝绿发布策略主要的价值是零宕机应用更新,保证服务不中断,能满足云环境下持续发布的 应用更新需求。

新版本线上评测

蓝绿发布策略可以让您在线上环境进行版本测试,通过保持一段时间内新旧版本并存,来充分测试 新版本服务的功能、性能和安全性。待新版本稳定后,再将旧版本应用下线。

#### 2.19.2 简单路由蓝绿发布策略

背景信息

蓝绿发布是一种零宕机的应用更新策略。进行蓝绿发布时,应用的旧版本服务与新版本服务会同时 并存,同一个应用不同版本的服务之间共享路由,通过调节路由权重的方式,可以实现不同版本服 务之间的流量切换。验证无误后,可以通过发布确认的方式将应用的旧版本的服务删除;如果验证 不通过,则进行发布回滚,应用的新版本会进行删除。

前置条件

支持蓝绿发布需要将路由服务升级到最新的版本。详细信息参见升级系统服务。

场景介绍

假设您要进行蓝绿发布的应用是一个 Nginx 的静态页面,初始的应用模板如下。

```
nginx-v1:
  image: 'registry.aliyuncs.com/ringtail/nginx:1.0'
```

```
labels:
   aliyun.routing.port_80: nginx
restart: always
```

#### 部署后页面的效果如下。



## Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to <u>nginx.org</u>. Commercial support is available at <u>nginx.com</u>.

Thank you for using nginx.

操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择目标应用所在的集群。
- 4. 选择目标应用并单击变更配置。

容器服务		应用列表							刷新	创建应用	
Swarm	Kubernetes										
概览		常见问题: ③ 如	1何创建应用 🔗 变更应用	目配置 🔗 简单路由蓝绿发布策略 🔗	容器弹性伸缩						
应用	1	集群: EGS-clus	群: EGS-duster ▼ ■ 隐藏案统应用 □ 隐藏离线应用 □ 隐藏在线应用 名称 ▼								
服务		应用名称	描述	状态	容器状态	创建时间 🔺	更新时间 🔺			操作	
集群		nginx		● 就绪	<b>就绪:1</b> 停止:0	2018-01-19 14:08:38	2018-01-19 14:22:11	停止   支更配置	删除	重新部署   事件	
节点								- 4			

- 5. 选择变更的发布模式与新版本服务的配置。
  - 在蓝绿发布中,新版本与旧版本不能共用同一个名字。

在蓝绿发布的场景中,为了保证应用的零宕机切换,新版本的服务的路由权重默认为0,需
 要通过路由管理页面进行调整,方可进行流量切换。

变更配置	$\times$
应用名称: nginx	7
*应用版本: 1.1	
注意:提交配置变更需要您更新应用版本号,否则确定按钮无法点击 	_
应用描述:	
使用最新镜像:	
发布模式: 蓝绿发布 🔹 🖉	
模板: 1 nginx-v2: 2 image: 'registry.aliyuncs.com/ringtail/nginx:2.0' 3 labels: 4 aliyun.routing.port_80: nginx 5 restart: always	
使用已有编排模板  标签说明	
确定	取消

模板样例如下所示:

6. 单击确定,发布变更。

在发布的过程中,会经历两个状态:

• 蓝绿发布中:表示新版本的服务尚未启动完成。

 蓝绿发布待确认:表示新版本的服务已经启动完成,此时需要进行发布确认或者发布回滚方 可进行下一次发布。

进入应用的详情页面,可以看到新版本的应用和旧版本的应用并存。

基本信息	基本信息												
应用名称:	应用名称: nginx					: 2018-01-19	更新时间: 2018-01-19	所	所在集群: EGS-cluster				
触发器 1.4	專种类型的触发	諸只能创發	1个@							创建触发器			
目前没有任何触	她发器,点击右	上角按钮包	別建触发器										
服务列表	容器列表	日志	事件	路由列表									
服务名称	所属应	Ħ	服务状	态	容器状态	镜像				120			
nginx-v1	nginx		● 就為	Ě	<b>就绪:1</b> 停止:0	registry.aliyuncs.com			停止   重启	重新调度   变更配置 删除   事件			
nginx-v2	nginx		● 就銷	E .	就绪:1 停止:0	registry.aliyuncs.com			停止   重启	重新调度   变更配置 删除   事件			

7. 单击路由列表并单击设置服务权重。

如下图所示,旧版本服务的权重为100,新版本服务的权重为0。

路由地址:	nginx.cd4f22d7b5a6248	3f09d27add5b3328723	.cn-hangzhou.al
路由权重:	服务名称	权重	百分比
	nginx_nginx-v1	100 ~	100%
	nginx_nginx-v2	0	0%

要做到零宕机升级,您需要先把新版本的服务的权重值调整为100,此时新旧服务的权重各占 50%,测试新旧版本的服务都有稳定的流量。

### 📕 说明:

同时调整两个服务的权重,可能会导致部分请求失败,所以权重需要分为两步进行调整,一步 只调整一个服务的权重。例如,先将新服务权重值从0调整为100,待流量稳定后,再将旧服 务权重值从100设置为0。 将旧版本服务的权重调整为0,新版本服务的权重调整为100。

路由地址: nginx.cd4f22d7b5a6248f09d27add5b3328723.cn-hangzhou.a 路由权重: 服务名称 权重 百分比 nginx_nginx-v1 ■ 0 ◇ 0% nginx_nginx-v2 ■ 100 ◇ 100%	141	陷由权重设置			>
路由权重:     服务名称     权重     百分比       nginx_nginx-v1     0 ♀     0%       nginx_nginx-v2     100 ♀     100%		路由地址:	nginx.cd4f22d7b5a6248	f09d27add5b3328723.	cn-hangzhou.al
nginx_nginx-v1 ■ 0    0% nginx_nginx-v2 ■ 100    100%		路由权重:	服务名称	权重	百分比
nginx_nginx-v2 ■ 100 🔶 100%			nginx_nginx-v1	0	0%
			nginx_nginx-v2	100 🗘	100%
				ł	<mark>确定</mark> 取消

由于默认路由服务是进行会话保持的,您可以打开一个新的浏览器窗口,访问新的版本,结果如下所示。

nginx.c89d1a3987e0a4ad1a30a728e67b60fa5.cn-beijing 🔿	( Å ) ( ā	+

### Welcome to nginx blue-green!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to <u>nginx.org</u>. Commercial support is available at <u>nginx.com</u>.

Thank you for using nginx.

当整个发布流程验证完毕后,在应用列表页面,单击确认发布完成,弹出确认对话框。可以选择是否勾选是否自动平滑更新。最后单击确认进行发布确认,方可进行下一次发布。

- 本田利主	发布确认		×		周室の日本日
NDH9124	•				
常见问题: ③如何创建应用 ③ 变更应用配置	<ul> <li>◎ 總備定要完成这此发布吗?</li> <li>◎ 是否自动平滑更新 ◎</li> </ul>				
集群: EGS-cluster 🔻 🗷 隐藏系统应用 🛛				名称 ▼	Q <b>X</b>
应用名称 描述			确定取消	包新时间 🔺	描述 作
nginx	●蓝绿发布待确认	航绪:2 停止:0	2018-01-19 14:08:38	2018-01-19 14:12:54	停止   变更配置   删除   重新部署   事件 确认发布完成   回滚

您可以看到应用的服务列表已经更新了,旧的服务已经完全下线删除了。

触发器 1.每种类型的触发器只能创建1个 2.关于资源伸缩类型的触发器											^	
目前没有任何触发器,点击												
服务列表	容器列表	路由列表	日志	事件								
服务名称  月	所属应用 服务	状态 容器状系	5 镜像									操作
nginx-v2 r	ıginx 🔵 हे	送行:1 优绪  停止:0	registr	ry.aliyuncs.	n/ringtail/nginx:2.0 🐐	亭止 🛛	重新调度		变更配置	删除	1	事件

### 2.19.3 负载均衡路由蓝绿发布策略

蓝绿发布是一种零宕机的应用更新策略。进行蓝绿发布时,应用的旧版本服务与新版本服务会同时 并存,同一个应用不同版本的服务之间共享负载均衡,通过调节负载均衡权重的方式,可以实现 不同版本服务之间的流量切换。验证无误后,可以通过发布确认的方式将应用的旧版本的服务删 除;如果验证不通过,则进行发布回滚,应用的新版本会进行删除。

场景介绍

假设您要进行蓝绿发布的应用是一个 Nginx 的静态页面,初始的应用模板如下。

```
nginx-v1:
  image: 'registry.aliyuncs.com/ringtail/nginx:1.0'
  ports:
    - 80:80/tcp
  labels:
    aliyun.lb.port_80: tcp://proxy_test:80
  restart: always
```

部署后页面的效果如下。



使用说明

由于每个容器需要暴露主机的端口,所以执行负载均衡路由蓝绿发布过程中,需要确保某个服务容器的个数少于或者等于集群中机器数的1/2;否则,会导致端口冲突。

进行负载均衡蓝绿发布时,您可以考虑先缩容容器到集群机器数量的一半,然后进行蓝绿发布,发布结束后,再扩容到原来的规模。

#### 操作步骤

- 1. 登录容器服务管理控制台。
- 2. 在 Swarm 菜单下,单击左侧导航栏中的应用。
- 3. 选择目标应用所在的集群。
- 4. 选择目标应用并单击变更配置。

容器服务		应用列表						刷新	创建应用				
Swarm	Kubernetes	常见问题: 🔗 如何创建应)	<b>20问题:</b> 6 如何创建应用 8 变更应用配置 8 简单路由蓝爆发布策略 8 容器器性伸缩										
应用	1	集群: test-swarm 🔻 🖲	◎ 隐藏系统应用 💷 隐藏离线应	用 🗆 隐藏在线应用		名称	T	Q	×				
服务		应用名称 描述	状态	容器状态	创建时间 🔺	更新时间 ▲	2		操作				
集群		nginx	●就绪	<b>就绪:1</b> 停止:0	2018-01-29 13:33:32	2018-01-29 13:54:46	停止   变更配置	删除	重新部署   事件				

5. 选择变更的发布模式与新版本服务的配置。



- 在蓝绿发布中,新版本与旧版本不能共用同一个名字。
- 在蓝绿发布的场景中,为了保证应用的零宕机切换,新版本的服务的路由权重默认为0,需
   要通过路由管理页面进行调整,方可进行流量切换。

变更配置	$\times$
应用名称:     nginx       *应用版本:     1.2       注意:提交配置变更需要您更新应用版本号,否则确定按钮无法点击	
应用描述:	
使用最新镜像:	
发布模式: 蓝绿发布 🔻 🖉	
模板: 1 <u>nginx-v2:</u> 2 jmage: 'registry.aljyuncs.com/ringtail_nginx:2.0' 3 · ports: 4 - 80:80/tcp 5 · labels: 6 aljyun.bb.port_80: tcp://proxy_test:80 7 restart: always 使用已有编排模板 标签说明	
确定	取消

模板样例如下:

```
nginx-v2:
    image: 'registry.aliyuncs.com/ringtail/nginx:2.0'
    ports:
        - 80:80/tcp
    labels:
        aliyun.lb.port_80: tcp://proxy_test:80
    restart: always
```

6. 单击确定,发布变更。

在发布的过程中,会经历两个状态:

- 蓝绿发布中:表示新版本的服务尚未启动完成。
- 蓝绿发布待确认:表示新版本的服务已经启动完成,此时需要进行发布确认或者发布回滚方 可进行下一次发布。

进入应用的详情页面,可以看到新版本的应用和旧版本的应用并存。

基本信息											
应用名称:	应用名称: nginx				创建时间	: 2018-01-19	更新时间: 2018-01-1	19 所在	所在集群: EGS-cluster		
触发器 1.4	專种类型的触发	器只能创新	110								创建触发器 ^
目前没有任何第	触发器 , 点击右	上角按钮的	別建触发器	ł							
服务列表	容器列表	日志	事件	路由列表							
服务名称	所属应	用	服务状	态	容器状态	镜像					提作
nginx-v1	nginx		nginx ●就绪		<b>就绪:1</b> 停止:0	registry.aliyuncs.com			停止	重启   重新	「調度 │ 変更配置 │ 別除 │ 事件
nginx-v2	nginx		● 就約	ă	就绪:1 停止:0	registry.aliyuncs.com			停止	重启   重新	「調度 │ 変更配置 │ ──────────────────────────────────

7. 单击路由列表并单击设置服务权重。

如下图所示,旧版本服务的负载均衡权重为100,新版本服务负载均衡的权重为0。

路由权重设置					$\times$
路由地址:					
股山坝去。				7011	
)	服务名称			自分比	
	nginx_nginx-v1		100 ~	100%	-
	nginx_nginx-v2	•	0	0%	
				确定取消	<b></b> ≝

要做到零宕机升级,您需要先把新版本的服务的权重值调整为100,此时新旧服务的权重各占 50%,测试新旧版本的服务都有稳定的流量。

📕 说明:

同时调整两个服务的权重,可能会导致部分请求失败,所以权重需要分为两步进行调整,一步 只调整一个服务的权重。例如,先将新服务权重值从0调整为100,待流量稳定后,再将旧服 务权重值从100设置为0。

下一步,将旧版本服务的权重调整为0,新版本服务的权重调整为100。

路由权重设置				$\times$
路由地址:				
路由权重:	服务名称	权重	百分比	
	nginx_nginx-v1	0 ^	0%	
	nginx_nginx-v2	100 ^	100%	
			确定取	肖

8. 您可以打开一个新的浏览器窗口,访问新的版本,结果如下。

Welcome to nginx blue-green!	
new	
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.	
For online documentation and support please refer to <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .	
Thank you for using nginx.	

当整个发布流程验证完毕后,在应用列表页面,单击确认发布完成,弹出确认对话框。可以选择是否勾选是否自动平滑更新。最后单击确认进行发布确认,方可进行下一次发布。

「	发布确认			同新创建立田
1 722752 3-64	<ul> <li>● 您确定要完成这些发布吗?</li> <li>※ 是否自动平涛更新 @</li> </ul>			A63471 COMMISSION
常见问题: & 如何创建应用 & 变更应用配置 & 简单路由蓝				
集群: test-swarm ▼ ⊠隐藏系统应用 ■隐藏离线应用 ■			名称 ▼	Q 🗶
应用名称 描述 状态		确定取消	更新时间 ▲	操作
nginx ● 蓝绿发布待佛认	机绪:2 2018-01-29 停止:0	9 13:33:32	2018-01-29 13:57:22	停止   交更配置   删除   里新部署   事件 确认发布完成   回滾

您可以看到应用的服务列表已经更新了,旧的服务已经完全下线删除了。

服务列表	容器列表	路由列表	日志	事件		
服务名称	所属应用 服务	务状态 容器状	态镜像			操作
nginx-v2	nginx 🔴	就绪  运行:1 停止:0	registi	ry.aliyuncs.	.com/ringtail/nginx:2.0 停止 重新调度 变更配置 删除	事件

# 3 Serverless Kubernetes 集群

### 3.2 应用管理