

天翼云 · 微服务云应用平台 用户使用指南

中国电信股份有限公司云计算分公司





目 录

1	产品概述	11
1.1	产品定义	11
1.2	术语解释	11
2	快速入门	13
2.1	应用上线快速入门	13
2. 1.	1 申请免费试用 ServiceStage	
2. 1.	2 上传 AK/SK 证书	
2. 1.	3 创建虚拟私有云	
2. 1.	4 创建密钥对	
2. 1.	5 创建资源空间	
2. 1.	6 创建集群	
2. 1.	7 为集群添加节点	21
2. 1.	8 创建 Nginx 容器应用	23
2.2	微服务开发快速入门	26
2. 2.	1 快速创建并部署微服务	26
3	资源管理	31
3.1	概述	
3.2	创建集群	
3.3	添加节点	



3.4	为节点添加标签	32
3.5	删除节点	
3.6	删除集群	35
3.7	查看集群资源	36
3.8	创建 SECRET 资源	36
3.9	创建 DAEMONSET 资源	
3.10	创建 POD 资源	40
3.11	创建 LIFECYCLE 资源	41
4 应	2用开发	43
4.1	概述	43
4.2	应用场景	44
4.3	微服务工程	44
4. 3. 1	本地工程	44
4.4	微服务调试	46
4. 4. 1	云上调试	
4.5	微服务治理	47
4. 5. 1	仪表盘	47
4. 5. 2	服务目录	48
4. 5. 3	服务治理	54
4. 5. 4	全局配置	65
4. 5. 5	事务看板	66



5	应用上线	70
5.1	创建应用配置项	70
5.2	创建任务类应用	72
5. 2.	<i>1 创建定时任务</i>	72
5. 2.	2 创建普通任务	77
5.3	创建应用	81
5. <i>3</i> .	1 概述	81
5. <i>3</i> .	2 创建容器应用(无状态)	82
<i>5. 3.</i>	3 创建容器应用(有状态)	91
<i>5. 3.</i>	4 <i>创建虚机应用</i>	95
5. <i>3</i> .	5 启停应用	98
5. <i>3</i> .	6 升级容器应用	98
5. <i>3</i> .	7 升级虚机应用	98
5. <i>3</i> .	8 删除应用	99
<i>5. 3.</i>	9 将应用移至应用组	99
<i>5. 3.</i>	<i>10 查看应用实例运行情况</i>	100
<i>5. 3.</i>	<i>11 查看应用更多详情</i>	101
5.4	创建任务类应用	101
5.4.	1	101
5.4.		102
J. 4.	3 则建坪戊削准亩	103



5. 4. 4	通过模板创建堆栈	. 103
5. 4. 5	通过设计器创建堆栈	. 106
5. 4. 6	查看堆栈详情	. 117
5. 4. 7	升级堆栈	. 118
5. 4. 8	伸缩堆栈	. 119
5. 4. 9	安装堆栈	. 120
5. 4. 10	回退堆栈	. 121
5. 4. 11	卸载堆栈	. 121
5. 4. 12	删除堆栈	. 122
6 应用]运维	.123
61 5	日本な	122
0.1 1		. 120
6. <i>1. 1</i>	/用血注 概念	. 123 . 123
6. 1. 1 6. 1. 2	概念	. 123 . 123 . 123
6. 1. 1 6. 1. 2 6. 1. 3	概念	. 123 . 123 . 123 . 126
6. 1. 1 6. 1. 2 6. 1. 3 6. 1. 4	概念	. 123 . 123 . 123 . 123 . 126 . 127
6. 1. 1 6. 1. 2 6. 1. 3 6. 1. 4 6. 1. 5	概念	. 123 . 123 . 123 . 123 . 126 . 127 . 129
6. 1. 1 6. 1. 2 6. 1. 3 6. 1. 3 6. 1. 4 6. 1. 5 6. 1. 6	概念 监控应用运行状态	. 123 . 123 . 123 . 123 . 126 . 127 . 129 . 130
6. 1. 1 6. 1. 2 6. 1. 3 6. 1. 3 6. 1. 4 6. 1. 5 6. 1. 6 6. 1. 7	概念	. 123 . 123 . 123 . 123 . 123 . 126 . 127 . 129 . 130 . 131
6. 1. 1 6. 1. 2 6. 1. 2 6. 1. 3 6. 1. 4 6. 1. 5 6. 1. 6 6. 1. 7 6. 1. 8	概念	. 123 . 123 . 123 . 123 . 123 . 126 . 127 . 129 . 130 . 131 . 132
6. 1. 1 6. 1. 2 6. 1. 3 6. 1. 3 6. 1. 4 6. 1. 5 6. 1. 6 6. 1. 7 6. 1. 8 6. 1. 9	概念 监控应用运行状态	. 123 . 123 . 123 . 123 . 123 . 126 . 127 . 127 . 129 . 130 . 131 . 132 . 133



<i>6. 2</i> .	1	概念	139
6. 2. 2	2	配置采样率	140
<i>6. 2.</i> .	3	配置老化周期	140
6. 2. 4	4	查看业务关键性能指标	140
6. 2. 3	5	性能瓶颈定界	141
6. 2. 0	6	<i>故障辅助定界</i>	143
6.3	应	Z用日志	144
<i>6. 3.</i>	1	概念	144
6. 3. 2	2	查询日志	144
<i>6. 3.</i> .	3	配置日志存储时长	145
6.4	事	译件	146
6. 4.	1	查询事件	146
6.5	볃	音警	146
6. 5.	1	监控告警	147
6. 5. 2	2	查询告警	148
6.6	弹	单性伸缩	150
6. 6.	1	概念	150
6. 6. 2	2	创建弹性伸缩组	150
6. 6	3	添加伸缩策略	151
7	软件	中心	153
7.1	锾	意像仓库	153



7. 1. 1	概念	153
7. 1. 2	上传私有镜像(外网)	155
7. 1. 3	上传私有镜像(内网)	158
7. 1. 4	获取镜像仓库地址	160
7. 1. 5	更新镜像仓库	160
7.2	软件仓库	161
7. 2. 1	概念	161
7. 2. 2	创建软件仓库	161
7. 2. 3	上传软件包	162
E、软件	Þ包选择完成后,单击【开始上传】;	164
7. 2. 4	软件仓库管理	164
7.2.4 8 系	<i>软件仓库管理</i> 统管理	<i>164</i> 166
7.2.4 8 系 8.1	<i>软件仓库管理</i> 统管理 证书和秘钥管理	<i>164</i> 166 166
7. 2. 4 8 系 8.1 8. 1. 1	<i>软件仓库管理</i> 统 管理 证书和秘钥管理 <i>秘钥列表</i>	<i>164</i> 166 <i>166</i>
7. 2. 4 8 系 8.1 8. 1. 1 8. 1. 2	<i>软件仓库管理</i>	164 166 166 166
7. 2. 4 8 系 8.1 8. 1. 1 8. 1. 2 8.2	软件仓库管理 统管理 证书和秘钥管理 秘钥列表 延书列表 租户资源管控	164 166 166 166 167
7. 2. 4 8 系 8.1 8. 1. 1 8. 1. 2 8.2 8. 2. 1	<i>软件仓库管理</i>	164 166 166 166 167 167
7. 2. 4 8 系 8.1 8. 1. 1 8. 1. 2 8.2 8. 2. 1 8. 2. 2	软件仓库管理 统管理 证书和秘钥管理 秘钥列表 <i>秘钥列表 证书列表</i> 相户资源管控 <i>创建资源空间</i> 申请资源配额	164 166 166 166 167 167 167
7. 2. 4 8 系 8.1 8. 1. 1 8. 1. 2 8.2 8.2. 1 8. 2. 2 8.3	软件仓库管理 统管理 证书和秘钥管理 秘钥列表 <i>秘钥列表 证书列表</i> 相户资源管控 <i>创建资源空间</i> 申请资源配额 (可选)申请资源配额	164 166 166 166 167 167 167
7. 2. 4 8 8.1 8. 1. 1 8. 1. 2 8. 2. 1 8. 2. 2 8.3 9 x	软件仓库管理 统管理 证书和秘钥管理 秘钥列表 证书列表 超户资源管控 創建资源空间 申请资源配额 (可选)申请资源配额 战教程	164 166 166 166 167 167 167 167 167



9.2	搭建 WORDPRESS 网站	
9. 2. 1	概念	
9. 2. 2	创建 MySQL 数据库	
9. 2. 3	创建 WordPress 服务	
9.3	开发部署 HELLO WORLD 微服务	
9. 3. 1	概念	
9. 3. 2	创建微服务	
9. 3. 3	开发微服务	
9. 3. 4	部署微服务	
9. 3. 5	治理微服务	
10 常	\$见问题	178
10.1	什么是节点?	
10.1 10.2	什么是节点?	
10.1 10.2 10.3	什么是节点?	
10.1 10.2 10.3 10.4	什么是节点?	
 10.1 10.2 10.3 10.4 10.5 	什么是节点?	
 10.1 10.2 10.3 10.4 10.5 10.6 	什么是节点?	
 10.1 10.2 10.3 10.4 10.5 10.6 10.7 	什么是节点? 对于创建节点失败期间回滚的资源,我是否需要付费? 	
 10.1 10.2 10.3 10.4 10.5 10.6 10.7 10.8 	什么是节点? 对于创建节点失败期间回滚的资源,我是否需要付费? 什么是环境变量? 堆栈、编排、模板、设计包分别是什么? "有状态容器应用"和"无状态容器应用"有什么不同? 什么是调度策略? 什么是亲和与反亲和? 什么是 TOSCA 模板?	
 10.1 10.2 10.3 10.4 10.5 10.6 10.7 10.8 10.9 	什么是节点? 对于创建节点失败期间回滚的资源,我是否需要付费? 什么是环境变量? 堆栈、编排、模板、设计包分别是什么? "有状态容器应用"和"无状态容器应用"有什么不同? 什么是调度策略? 什么是亲和与反亲和? 什么是 TOSCA 模板? 如何解决堆栈无法删除的问题?	



10.11	如何收藏导航栏菜单?	183
11 🔞	▶数说明	185
11.1	如何配置容器应用卷定义?	185
11.2	如何配置自定义监控?	186
11.3	如何设置升级策略?	187
11.4	如何设置调度策略?	188
11.5	如何设置容器应用生命周期?	191
11.6	如何设置容器应用数据卷?	191
11.7	如何设置容器应用环境变量?	192
11.8	如何进行容器应用健康检查?	193
11.9	如何配置虚机应用卷定义?	194
11.10	如何设置虚机应用生命周期?	195
11.11	如何设置虚机应用数据卷?	195
11.12	如何设置虚机应用环境变量?	196
11.13	如何设置虚机应用健康检查?	197
11.14	什么是标签?	198
11.15	如何通过设计器示例模板创建堆栈?	199
11.16	如何配置服务发布参数?	201
12 参	▶考知识	203
12.1	资源描述文件配置说明	203
12. 1.	1 Json 格式资源文件配置说明	203



12. 1. 2 Ya	aml 格式资源文件配置说明	. 208
12.2 软件	包打包规范	.210
12.3 应用	监控常见问题	. 211
12. 3. 1	如何搜索指标	. 211
12. 3. 2	如何搜索阈值规则	. 212
12. 3. 3	如何修改阈值规则	. 212
12. 3. 4	如何删除阈值规则	. 213
12. 3. 5	如何删除监控视图	. 213
12. 3. 6	如何在监控视图中编辑指标图表	. 214
12. 3. 7	如何在监控视图中添加指标图表	. 214
12. 3. 8	如何在监控视图中删除指标图表	. 215
12. 3. 9	如何在监控视图中刷新指标图表	. 215
12. 3. 10	如何在监控视图中放大或移动指标图表	. 216
12.4 应用	编排设计包说明	. 216
12. 4. 1	设计包概述	. 216
12. 4. 2	设计包开发原理	. 217
12. 4. 3	设计包格式规范	. 219
12.5 应用	编排支持的对象类型与配置说明	. 222
12. 5. 1	应用组	. 222
12. 5. 2	应用	. 224
12. 5. 3	任务(Job 应用)	. 229

目 录



12. 5. 4	代理(Agent 应用)	. 231
12. 5. 5	软件	. 235
12. 5. 6	集群	. 244
12. 5. 7	生命周期脚本定义	. 245
12. 5. 8	输入	. 248
12. 5. 9	输出(高级功能)	. 249
12.6 CON	IFIGMAP 类型资源描述文件配置说明	. 250





产品概述

1.1 产品定义

微服务云应用平台(ServiceStage, CT-SS)为公有云化转型提供一站式 DevOps 平台与中间件服务, 帮助用户快速开发、构建、部署、上线和运维分布式应用,提供应用全生命周期管理体验。

微服务云应用平台提供了如下功能,帮助用户快速创建应用以及自动化的运维:

- 应用全生命周期管理:打通从应用建模、编排部署到资源调度、弹性伸缩、监控自愈的生命周期管理自动化。针对较为简单的应用,推荐用户使用创建应用功能,支持"无状态/有状态容器应用"和"虚拟机应用"的部署;针对较为复杂的应用(组件多,关系复杂),推荐用户使用编排来创建应用堆栈;
- 运维监控:资源阈值警报,应用 KPI 监控、数据库 MySQL 监控以及调用链分析帮助开发快速完成问题定位;
- 微服务运行与治理框架:为应用提供自动注册、发现、治理、隔离、调用分析等一系列分布式微服 务治理能力,屏蔽分布式系统的复杂度。

1.2 术语解释

- 1. 集群:一个集群指容器运行所需要的云资源组合,关联了若干云主机节点、负载均衡等云资源;
- 节点:每一个节点对应一台云主机,容器应用运行在节点上。节点上运行着代理程序(kubernetes), 用于管理节点上运行的容器实例。节点规格最小是CPU为1core,内存为2048MB;最大是CPU为32core, 内存为128GB;
- 3. 容器: 一个通过 Docker 镜像创建的运行实例, 一个节点可运行多个容器;
- **4.** Docker 镜像: Docker 镜像是容器应用打包的标准格式,在部署容器化应用时可以指定镜像,镜像可以来自于 ServiceStage 公有仓库,或者用户的私有镜像;
- 编排:编排模板包含了一组容器服务的定义和其相互关联,可以用于多容器应用和虚机应用的部署 和管理;



- **6. 堆栈:**由应用、服务、资源等元素组成的一个部署实例,平台将相关编排元素通过"堆栈"进行集中管理;
- **容器应用:**一个容器应用可通过单个镜像或一个编排模板创建,每个容器应用可包含1个或多个容器;
- 6库:仓库包括镜像仓库和软件仓库,镜像用于容器类应用,软件包用于虚拟机或物理机应用。用 户在创建应用前,需要将应用所需的镜像或软件包上传到仓库中;
- **9. 服务:** ServiceStage 服务市场中,展示了各种类型的中间件服务。用户可以在服务市场中订购使用 所需的服务;
- 10. 安全:以租户项目空间为隔离粒度分配工作密钥和证书,用于加密保护租户的敏感数据,建立通信安全。对运行态的容器实时监控,识别潜在的恶意容器。对租户的操作实施审计跟踪,识别异常的租户操作,阻断跨租户的数据访问、异常登录等风险;
- **11. 各模块关联关系:** 一个集群由多个节点组成,一个节点上可以部署多个应用,一个应用由一个或多 个容器部署而成,堆栈是由应用、服务、资源等元素组成的一个部署实例。





2 快速入门

2.1 应用上线快速入门

2.1.1 申请免费试用 ServiceStage

ServiceStage 目前处于公测阶段,用户可申请免费试用。申请通过后,即可登录 ServiceStage 管理 控制台。

2.1.2 上传 AK/SK 证书

使用 ServiceStage 前,您需要上传 AK/SK 证书,授权 ServiceStage 使用您的镜像和集群。

AK:访问密钥 ID,是与私有访问密钥关联的唯一标识符。访问密钥 ID 和私有访问密钥一起使用,对 请求进行加密签名。

SK: 与访问密钥 ID 结合使用的密钥,对请求进行加密签名,可标识发送方,并防止请求被修改。



- 每个用户最多可创建2个有效访问密钥(AK/SK),且一旦生成永久有效。为了账户安全性,建议定期更换访问密钥。
- 1. 注册并登录天翼云 https://www.ctyun.cn;
- 2. 切换到杭州的控制中心页面;
- 3. 点击控制中心页面右上方登录的帐户,选择【我的凭证】;

Cloud · Clip	 ♥ 收調: 	 ● 約州 	• 其他节点	ivycln@1 •	lh 🖾 ?
所有服务 我的收藏			[¹¹ 用户中心 务 我的订单	Q
计算 ~	存储 ~	网络 ~	安全 🗸	消费详单 我的凭证	
弹性云主机(1) 弹性可扩展的计算服务器	云硬魚(1) 身性可扩展的块存储	近辺 虚拟私有云(3) 安全環境的虚拟网络	A H	切损角色 n 强 提交工单	
物理机服务(0) 和户专属的物理服务器	云硬盘新份(0) 安全可靠的快行储备份	→ → → → → → → → → → → → → →		'南燕++-(> 退出	



- 4. 选择【管理访问密钥】页签;
- 5. 单击【新增访问密钥】链接, 创建新的访问密钥;

我的凭证		습 首页	器 服务列表 ▼	♡ 收藏 ▼		其他节点
用户名: 用户日: 用户日: ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	我的凭证					
用户ID: 用户ID: 税户名: 小小一〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇〇				用户名:		
・・・・・・・・・・・・・・・・・・・・・・・・・・・・・				用户ID:		
秋户ID:				帐户名:	ing in the second s	
已验证邮箱: ************************************				帐户ID:	(11) 00007(400.1004 //0.0007h.	
已验证手机: 10000000 请				已验证邮箱:	the reaction	
项目列表 管理访问密钥 请				已验证手机:	10000000	
	项目列表 管理访问密	钥				请辅
访问密朝ID * 创建时间 * 状态 *	访问密钥ID ▼			创建时间	状态	
VWUIW0SWVUZHHGXHZ1LQ 2017/07/04 14:06:27 GMT+08:00 Active	VWUIW0SWVUZHHGXHZ1LC	2		2017/07/04 14:06:27 GMT+08:00	Active	

- 6. 点击【免费获取短信验证码】按钮,查看手机收到的验证码填写在【短信验证码】文本框;
- 7. 单击【确认】按钮,下载访问密钥;

	下载确认	×	
	您的AKSK已经生成,点击确定进行下载,点击取消返回我的凭 证。		
	确定取消		
入 注意			

- 为防止访问密钥泄露,建议您将其保存到安全的位置。
- 9. 单击【应用】【资源管理】【集群】链接;



Service Stage 🏠 概览	应用 服务 编排 微服务 运维 仓库 议	2置	
概览 ivycln@126.com 租户	送资源管理 集課	合 任务管理 普通任务	
节点数	节点 其他 88 应用管理	定时任务 器 配置项	

10. 在弹出的【上传认证文件】窗口中,单击____,选择步骤 7 中下载的认证文件,单击【上传】。 认证文件格式为".csv",文件大小不能超过 512 个字节;



11. 系统提示上传成功,单击【关闭】按钮;

大算 Cloud			2 快速入门
Service Stage	一 概览 应用 服务 编排 微服务 运维 仓库 设置	♀ 鄙杭州	≗ ivycln@126.com@…
	□ 」 ○ 资源管理 > 集群		
③ 资源管理 へ	请创建集群并添加节点,用于运行您的应用。您最多可以创建2个集群。了解更多		
• 集群 • 节点	+ 创建集群 上传 ×		
 其他 命名空间 	credentials.csv ···· 上传 上传成功		
82 应用管理	「「」」「」」「」」「」」「」」「」」「」」「」」「」」「」」「」」「」」「」		
☑ 任务管理 ∨□ 配置项			

2.1.3 创建虚拟私有云

使用 ServiceStage 前,您还需要创建虚拟私有云(VPC),为 ServiceStage 服务提供一个隔离的、用 户自主配置和管理的虚拟网络环境,提升公有云中资源的安全性,简化用户的网络部署。如果用户已有 VPC, 可重复使用,不需多次创建。

- 1. 注册并登录天翼云 https://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 📴 ,选择【网络】【虚拟私有云】;
- 4. 在【总览】界面,单击【申请虚拟私有云】,创建虚拟私有云;

	颖 器服务列表▼	♥ 收藏 ▼		(◎ 杭州 •	其他节点 💮 iv	ycin@1 ▾ 🕼 🗹 ?	
「「」」の格控制台	VPC即虚拟私有云,是通过逻辑 更多 十 创建虚拟私有云	電方式进行网络隔离,提《	共安全、隔离的网络环境,	VPC为您提供与传统。	网络无差别的虚拟网络,	同时还可以为您提供弹性IP、	安全组、VPN等高级网络服务。 了解	
③ 急災	我的资源					Ð	快速指南	
 > 虚拟私有云 vpc-2444 ▼ 	「「」」 虚拟私有云	3 (4		3	创建走找私有云 创建子网 创建安全组	
子网 0 ② 安全组	(IP) 弹性IP	1 (🕥 对等连接	0	VPN VPN	0	申请弹性IP 申请VPN 申请云专线	

5. 更多虚拟私有云创建步骤,请参见《天翼云 3.0 虚拟私有云用户使用指南》;

2.1.4 创建密钥对

为安全起见, ServiceStage 登录时建议使用密钥方式进行身份验证。因此, 您需要使用已有密钥对或



新建一个密钥对,用于作为添加节点时的身份验证。如果用户已有密钥对,可重复使用,不需多次创建。

- 1. 注册并登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 , 选择【计算】【弹性云主机】;
- 4. 在【总览】界面,单击【密钥对】;
- 5. 在【密钥对】页面,单击【创建密钥对】;

	() 首页	器 服务列表 ▼	♥ 收藏 ▼		◎ 杭州	Y	其他节点	0	ivycln@1 🔻	0,	⊠	?
(=:) 云主机控制台		如果您使用的是Linux云主树 您可以新建一个密钥对,并 也可以把已有的密钥对(Key 十 创建密钥对	1,建议使用密钥对(Key Pair)进 下载私钥,私钥用于远程登录。 Pair)的公钥导入系统,使用对 导入密钥对	出行远程登录身份验证。 460认证,为保证无土机杂全,私朝归 应的私钥进行远程登录身份验证。	马能下载一次,请受善保	ē.						
圖 总览											_	
◎ 弹性云主机		~1L	1512						请输入关键字	_	Q	Ð
③ 物理机服务		治称	捐双						1382	F		
◎ 云硬盘				智无	表格数据							
◎ 云硬盘备份												
⊜ 镜像												
◎ 弾性伸缩												
◎ 弹性负载均衡(外网)												
◎ 弹性负载均衡(内网)												
⑧ 密钥对												

 输入密钥对名称,名称只能由英文字母,数字,下划线,中划线组成,且长度小于等于 64 个字符, 单击【确定】;

创建密钥对					
名称:	KeyPair-c1a	KeyPair-c1aa			
	确定	取消			

- 7. 单击【确定】按钮;
- 8. 请根据提示信息,查看并保存私钥。为保证安全,私钥只能下载一次,请妥善保管。



で で Cloud ・ 協	首页 器 服务列表 ▼	♡ 牧藤 *	◎ 杭州	• 其他节点	💮 ivycln@1 🔹 🖿	⊠?
安主机控制台 ⑤ 总览 ④ 弹性云主机 ③ 物理机服务 ③ 云硬盘 ③ 云硬盘备份 ● 镜像 ④ 弹性伸缩 ④ 弹性负载均衡(外网) ● 弹性负载均衡(外网) ● 弹性负载均衡(内网)	<u><u></u> 如果等後用的量」inux元主机 、 信可以通己年前活明対(Kep Pair <u>+</u> 创建密明対 <u>名称</u> KeyPair-claa</u>	XV4用度明約1Key Pair/进行近年登录時份後述。 IT在打开 keyPair-claa.pem 忽然導了打开: KeyPair-claa.pem 文体地理: Text Document (1.7 KB) 来源: blob: 欧那要 Firefox 知何处理此文件? ① 打开,通过(2) 记事本(像从) ④ 保存文体(5) 重 處面 ① 以后自动采用相同的动作处理此类文件,(4)	X Mit-		清給入关键字 操作 翻除	0

2.1.5 创建资源空间

使用 ServiceStage 前,用户需要创建资源空间。创建完成后,系统将会自动为用户分配一定的资源 配额。若系统分配资源不足,用户还可以申请更多资源配额。



- 若使用企业多租户登录,仅租户有权限"创建资源空间"和"申请资源配额"。
- 1. 注册并登录天翼云 https://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 , 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 点击【设置】【租户资源管控】;
- 勾选"允许平台监控,自动弹性伸缩",系统将会对您在平台上申请的资源进行监控,并根据您 资源运行状况,在资源数量允许的范围内进行自动弹性伸缩;

建 Loud						2 快速入门
Service Stage	俞 概览 应	用服务编排微服务运维合库;	2 2 1	Q	靜杭州	옫 ivycln@126.com
	设置 > 租户资源管控					
租户资源管控		资源空间配置				创建资源空间
		节点数(个):	实例数(个):			
		集群数 (个):	集群内节点数(个):			
		动态节点CPU (核):	动态节点内存 (GB):			
		资源空间能力				

 点击【设置】【租户资源管控】,系统将自动会用户分配一定的资源配额。若系统分配的资源不足时, 可申请更多资源配额,步骤请参见错误!未找到引用源。(可选)申请资源配额;

2.1.6 创建集群

集群是容器运行所需要的云资源组合,关联了若干云主机节点、负载均衡等。用户可以创建容器集群 作为私有集群,也可以将应用托管在集群上运行。请确保至少有一个可用的集群,否则无法正常使用 ServiceStage。

前提条件:已完成上传 AK/SK 证书、错误!未找到引用源。错误!未找到引用源。、创建密钥对、创 建资源空间操作。

创建集群步骤仅需在使用 ServiceStage 前操作一次,无需重复操作。具体操作步骤如下:

- 1. 进入微服务云应用平台页面;
- 2. 在主菜单栏中选择【应用】【资源管理】【集群】;



3. 点击【创建集群】按钮;

天翼云		
eCloud		2 快速入门
Service Stage	硫 概览 应用 服务 编排 微服务 运维 仓库 设置	♀ 鄙杭州
	应用 > 资源管理 > 集群	
③ 资源管理 🔷 🔨	请创建集群并添加节点,用于运行您的应用。您最多可以创建2个集群。了解更多	
• 集群 • 节点	+ 创建集群	
• 其他		
人名古法		

- 4. 设置新增集群参数,其中带"*"的参数为必填参数:
 - 集群名称:必填,新建集群的名称;
 - 服务区域:必填,默认为选择的节点,请勿修改;
 - 所在 VPC: 必填, 新建集群所在的虚拟私有云;
 - 所在子网:必填,节点虚拟机运行的子网环境;
 - 描述:新建集群的描述信息;

创建集群		×
* 集群名称:		Â
*服务区域		
cn-hzla		
* 所在VPC:	→	
vpc-2444		
* 所在子网:		E
subnet-a01f		~
描述:		
输入文本在200字以内		
		_
		-
	确定取	消

5. 配置完成后,单击【确定】;

2.1.7 为集群添加节点

每一个节点对应一台虚拟机,容器应用运行在节点上。节点上运行着代理程序(kubelet),用于管理 节点上运行的容器实例。用户需确保至少有一个运行中的节点包括在集群中,否则无法使用 ServiceStage 创建应用、订购服务等功能。

节点规格最小是 CPU 为 1core,内存为 2048MB;最大是 CPU 为 32core,内存为 128GB。一个集群最多可添加 15 个节点。

1. 在微服务页面中,选择【应用】【资源管理】【节点】;

2. 选择集群,点击【新增节点】链接;

Service :	Stage		Q	鄙杭州	& ivycln@126.com@
(日本) 应用	C	应用 > 资源管理 > 节点			
③ 资源管理	^	查看、添加、删除集群下的节点,了解更多节点的信息,请点击了解更多			
•集群		作 刊·			
 节点 		πatest V			
• 其他					
• 命名空间					
🔓 应用管理					
🖾 任务管理	\sim				
🔓 配置项		节点承载应用的计算资源单元,包括物理机、虚拟机两种。 暂无节点,您可 <mark>以 新增节点</mark>			

- 3. 在新增节点页面,设置新增节点参数,其中带 "*" 标志的参数为必填参数;
 - 节点名称:必填,节点名称不能为空,以小写字母开头,由小写字母、数字、中划线(-)组成,且不能超过 24 位;
 - 节点类型:必填,提供了通用型和内存优化型两种类型的弹性云主机。通用型:通用型实例 提供均衡的计算、存储以及网络配置,适用于大多数的使用场景。通用型实例可用于 web 服 务器、开发测试环境以及小型数据库应用等场景;内存优化型:提供内存比例更高的实例, 可以用于对内存要求较高、数据量大的应用,例如关系数据库、NoSQL等场景;
 - 弹性 EIP:可选择"暂不使用"、"现在选择"、"使用已有"。若用户需要应用可以被外 网访问,可选择"现在选择"和"使用已有"。如选择"现在选择"系统将为创建的节点分 配新的弹性 IP 地址。
 - 规格:必填,当弹性 EIP 选择"现在选择"时,此值可选,默认为"中国电信";
 - 计费模式:必填,当弹性 EIP 选择"现在选择"时,此值可选,默认为"按带宽计费";



- 带宽:必填,当弹性 EIP 选择"现在选择"时,此值可选,可拖动滑条选择或输入所需的带 宽值;
- CPU: 必填, 为节点(虚拟机)分配的 CPU;
- 内存:必填,为节点(虚拟机)分配的内存;
- 磁盘:必填,分为系统盘和数据盘。系统盘的规格为 40GB,用户不可配置;数据盘的规格为 [100,32768]GB,用户可以配置,缺省值为 100GB,您可添加多块数据盘(最多 10 块)。提供超高 10,高 10,普通 10 三种性能规格的云硬盘。超高 10:指由 SSD 存储介质构成的云硬盘。高 10:指由 SAS 存储介质构成的云硬盘。普通 10:指由 SATA 存储介质构成的云硬盘;
- 密钥对:必填,选择"2.1.4 创建密钥对"中创建的密钥对名称;
- 购买数量:必填,节点的数量;

Service Stage	俞 概览 应用	服务 编排 微服务 运维 仓库 设置
(首首,应用		
\smile		
② 姿酒管理	* 节点名称:	linatest-iqpnm
() MINER		节点名称不能为空,以小写字母开头,由小写字母、数字、中划线(-)组成,且不能超过24位。
・集群	* 节点类型:	 通用型 内存优化型
 节点 		適用型实例提供均衡的计算、存储以及网络配置,适用于大多数的使用场景,通用型实例可用于web服务器、开发测试环境以及小型数据库应用等场景。
• 其他	弹性IP:	暂不使用 现在选择 使用已有
• 命名空间		选择使用WHEIP时,将为创建的节点绑定WHEIP地址,请确码WHEIP地址電缆充足。
18 应用管理	* 规格:	 中国电信
じたを管理	* 计费模式:	 技術常计書
	*带宽:	1Mbit/s
88 配置项		1 Mbit/s
	* CPU:	1歳 2核 4核 8核 16核 32核
	* 内存:	2GB 4GB
		为了保证性能体验,Windows2008/2012系统建议选择2GB及以上内存
		选择的规格为: c2.medium 1核 2GB

- 设置完成后,单击【新增节点】按钮,跳转到节点列表页面,待节点状态为"可用",表示节点添加成功。添加节点时会自动创建云主机;
- 5. 节点创建成功后,可在节点列表中查看节点信息;

(日常) 应用	0	魚用 > 物源管理 → 市点
③ 资源管理	^	查看、添加、删除集群下的节点,了解更多节点的信息,请点击了解更多
• 集群 • 节点		集群: linatest2 v
・其他 ・命名空间		сриде 1 сого ряже 2048 мв леда 1 ба
≌ 应用管理 ☑ 任务管理	~	
88 配置项		- 1918年2月14月17日日。 十新増行点) 添加時盛 一 御除 - 1918年1月1日日 - 1918年1日日日 - 1918年1日日日 - 1918年1日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日日
		□ 市点 所有状态 √ CPU (C 内存 (弹性P 市点P 标弦 单键时P 量化 □ linatest2-l1zpx 回日 1 2048 192.168 os.architecture 2017/09/17 1 活动标签 删除

2.1.8 创建 Nginx 容器应用

Cloud

Nginx 是一款轻量级的 Web 服务器,使用 ServiceStage 平台中内置的 Nginx 镜像,可快速创建 Nginx 容器应用,搭建 Nginx Web 服务器。

前提条件: "nginx 镜像"已上传至镜像仓库;

- 1. 在微服务页面中,选择【应用】【应用管理】;
- 2. 在右侧页面中,选择应用需要部署到的集群和命名空间,单击【创建应用】链接;

(日本) 应用	0	血用 > 应用管理
 资源管理 	~	集群: linatest2 ~ 命名空间: default ~
😵 应用管理		
🖾 任务管理	~	
吕 配置项		
		应用是可認署的软件实体,包含一个或者一组容器或者进程。

3. 单击容器应用下的【创建】按钮,进入应用创建页面;

创建应用				×
	容器应用		虚机应用	
	以容器技术来运行构建应用的环境		将应用进程直接部署在虚拟机或物 理机操作系统上	
	创建]	创建	
		-		

- 4. 在应用创建页面,选择【无状态应用】,配置应用如下基本信息,其它为默认值;
 - 所在集群:选择需要部署到的集群;
 - 所属命名空间:应用所属的命名空间;
 - 应用名称:新建容器应用的名称,命名必须唯一,如nigix;



28 应用管理 ② 任务管理 28 配置项	~	ーーーー 应用基本信息	应用配置	 	 完成
		无状态应用 有状态应用			
		* 所在集群 linatest2			~
		* 所犀命名空间 default			~
		nigix			

- 5. 配置完成后,单击【下一步】;
- 6. 配置以下应用相关参数,其它参数采用默认值;
 - 在【容器】下,单击【选择镜像】

* 容器 🕖	
container1 (+) 添加容器	
基本设置	
容器 container1	
镜像	

- 在弹出的【选择镜像】窗口中,选择【公有仓库】页签,选择镜像。
- 设置内存限制: 1GB, 内存分配 0.5GB;
- 设置 CPU 限制: 1 核, CPU 分配 0.5 核;
- 单击【保持】按钮;
- 7. 配置完成后,单击【下一步】;
- 8. 配置服务发布参数:
 - 发布为服务:选择"是",将 ng inx 设为可被访问的服务;
 - 设置服务参数:服务名称——可自定义,如 nginx;服务类型——设置为 NodePort, NodePort 主要适用于内网调试,若需要外网正式访问,建议选择 LoadBalancer,服务发布的具体说明 请参见错误!未找到引用源。如何配置服务发布参数;应用端口——80(请根据 nginx 镜像 实际容器端口填写);服务端口——保持为空,系统会自动分配端口,该端口的取值范围为 "30000-32767";



- nodePort 参数值需确保"单集群"内全局唯一,否则会因为端口冲突而导致部署应用失败, nodePort 的端口范围是 30000-32767。
- 9. 单击【确认】按钮;
- 10. 配置完成后,单击【下一步】;

创建nginx应用

		○ 应用配置	● 服务发布			
发布为服务 🖗 💿 是 🔷 香 若需要创建或增加ELB实例,点击这 * 服务参数 🖗	↓里;若ELB实例未加载,可尝	试刷新				
服务名称	服务类型	应用端口	服务端口	协议	ELB实例	操作
nginx	NodePort	80		ТСР		编辑 删除
+ 添加						
					上一步	下一步

11. 确保信息配置无误后,单击【创建】按钮;

Service Stage	命 概览 应用 」	服务 编排 微服务 运维 仓库 设置			٩	靜杭州 2 ivycln@126.com@…
	✓ ┃ 应用配置 实例数量: 1 container1]				
	镜像: 100.	125.7.22:20202/test/redis		版本: 0907		
	内存限制: 0.50	βB		CPU限制: 0.5核		
	端口:					
	容器端口			协议		
			哲疗	表格数据		
	•		m			•
	生命周期	数据卷 环境变量 健康检查				
	生命周期脚本定义,	主要针对容器类应用的生命周期事件应该采取的动作				
	启动后处理	停止前处理				
	◇ 服务反仲					
	服务参数:	☆用端□	服务端口	协议	ELB实例	
	NodePort	80	AND A STORE	TCP		
	NodeFort	00		TCr		
						上一步 创建

12. 单击【应用】【应用管理】可以查看所建的应用列表,应用状态为"运行中",表示应用创建成功;



- 13. 点击应用名称,进入应用详情页面,访问地址后的"IP∶端口"是应用被访问的入口信息。若应用 需要被外网访问,必须满足以下条件:
 - 应用所在集群已有至少一个节点绑定 EIP;

$\overline{}$	镜像:	100.125./.22:20202/test/redis:0907		
	应用类型:	无状态应用	类型:	容器
	应用组:		应用版本:	1
	实例个数(正	常/全部): 1/1	实例端口:	80
	访问地址:	183.134.74.142:31918	 标签:	
	创建时间:	2017/09/18 10:16:18 GMT+08:00		
	描述:			
	启动	停止 移至应用组 更多 →		

2.2 微服务开发快速入门

2.2.1 快速创建并部署微服务

ServiceStage 提供贯穿开发、部署、运行时保障、治理的端到端的"微服务开发框架",帮助用户以 最低成本和最安全的方式进行微服务的开发和管理。

本节以 Hello World 为例,介绍如何基于 ServiceStage "微服务开发框架"快速部署一个微服务。

前提条件:已完成上传 AK/SK 证书、创建虚拟私有云、创建密钥对操作,已创建创建资源空间、集群、 为集群添加节点。

- 1. 注册并登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 , 选择 【企业中间件】-> 【微服务云应用平台】;
- 4. 选择【应用开发】->【微服务开发】,进入本地工程创建页面;
- 5. 在本地工程页面填写以下操作:
 - 基于 CSE: 选择 SDK (Software Development Kit) 版本;
 - Group: 对应源码 maven 工程的 Group ID;
 - Artifact: 对应源码 maven 工程的 Artifact ID;
 - Java 包名称:对应源码工程中的 package 名称;



- 版本号:微服务版本号;
- Rest Path: 对应发布接口的 URL, 用户可自定义填写, 如"/sayhello";
- 名称:为微服务取个名字,如创建 Hello World 的服务消费端,名称为 "provider";
- 开发风格:系统提了透明 RPC、spring-mvc、jax-rs 三种开发模式供选择,缺省配置透明 RPC;
- 6. 单击【创建并下载】,完成微服务工程的创建。微服务工程创建以后,系统会自动生成基于微服务框架的开发工程包并保存到本地;
- 7. 在本地完成 HelloWorld 的应用开发以后,将开发工程包打包成 Docker 镜像包,假设 Docker 镜像包名为 "provider.tar";
- 8. 将生成的镜像 "provider.tar" 上传到仓库,请参考 7.2.3 上传私有镜像 (内网);
- 9. 创建微服务应用:
 - 选择【应用】【应用管理】;
 - 在右侧页面中,先选择集群,再单击【创建应用】;
 - 单击容器应用下的【创建】;
 - 填写应用信息,参照下表配置应用基本信息;

参数	参数说明	值
所在集群	应用所在集群。	自动读取上一步骤选取的集群。
应用名称	新建容器应用的名称,命名必须唯一。	provider, consumer。
应用组	您可以将某类应用放到同个应用组中,实现应 用的批量管理(如批量启停 <i>、</i> 删除)。	_
	用户输入应用组名称后,会自动创建一个新的 用户组。	
描述	输入对应描述。	-

10. 定义卷信息:

- 单击【展开高级设置】,系统显示卷定义页面;
- 单击【卷定义】下的【添加】;

bud				2 快速入门
起高级设置				
起高级设置 卷定义 卷定义是创建应用时定义的卷信息。	上意:挂载到系统目录可能造成容器与节点异常! 查着详情			
 記書級發置 卷定义 卷定义是创建应用时定义的卷信息。 卷名称 	1章: 挂載到系統目录可能造成容器与节点异常! 查看详情 卷 关型	配置	操作	
 総局吸設置 後定义 様定义是创建应用时定义的卷信息。 巻名称 	1意: 挂载到系统目录可能造成容器与节点异常! 查 看详情 卷关型 	前置	操作	

- 输入卷名称,用户可自定义填写;
- 卷类型选择 HostPath。HostPath 为在容器上挂载宿主机的文件或目录。通常用于容器应用 程序生成的日志文件需要永久保存,需要访问宿主机上 Docker 引擎内部数据结构的容器应 用;
- 单击"添加配置",输入主机路径,如/var/lib/docker;

卷定义 卷定义是创建应用时定义的卷信息。*注意:挂载到系统目	录可能造成容器与节点异常! 查看详情		
卷名称	卷类型	配置	操作
datavolume	HostPath ~	添加配置	确认 修改配置 取消
(+) 添加			

添加配置		×
* 主机路径 /var/lib/docker		
	确认	取消

- 单击【确认】保存主机路径;
- 单击操作列【确认】保存卷定义;

卷定义 卷定义是创建应用时定义的卷信息。*注意:挂	载到系统目录可能造成容器与节点异常! 查 <mark>看详情</mark>		
卷名称	卷类型	配置	操作
datav	HostPath v	/var/lib/docker	<u>确认</u> 修改配置 取消
(+) 添加			

- 11. 单击【下一步】进行应用配置;
 - 在【容器】下,单击【选择镜像】,选择已准备好的镜像 "provider.tar";



- 设置内存分配为 0.5GB, 限制为 1GB;
- 设置 CPU 分配为 0.5 核, 限制为 1 核;
- 在【数据卷】页签,单击【添加】,参考下表填写配置信息;

生命周期	数据卷	环境变量	健康检查			
数据卷是指将宿	主机/网络存(儲的某个绝对路径	挂载到容器或主机的某个路径上,与Docker命令的)"-v"参数效果相同。通过	挂载可以实现宿主机和容器之间,网络存储和容器之间,网络存储和宿主	:机之间的数据共享。
卷名称			容器路径	是否只读	卷策略及日志策略	操作
datav			/var/test	否	logs(日志策略:Daily)	编辑删除
+ 添加						
删除此容器	保存					

配置项	值
卷名称	在下拉框中选择步骤 10 中创建的卷。
容器路径	数据卷挂载到容器上的路径,请不要挂载在系统目录下,如"/"、 "/var/run"等,否则会导致容器异常。
是否只读	"是":只能读容器路径中的数据卷。 "否":可修改容器路径中的数据卷,容器迁移时新写入的数据不会 随之迁移,会造成数据丢失。"卷类型"为"Secret"时,数据为只 读。
卷策略及日志策略	当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs。
操作	点击"确认",保存配置信息。 点击"取消",删除配置信息。

- 单击【确认】保存配置;
- 单击【保存】;

12. 单击【下一步】进行服务发布配置;

- 发布为服务:选择"是",将应用设为可被访问的服务;
- 其它参数:容器端口——28081,协议——TCP,节点端口——输入整数(安装 ServiceStage 时,用户可修改该端口的取值范围,请以页面实际展示为准。若安装时未做修改,则默认范围为"30000[~]32767");
- 单击【确认】按钮;

13. 单击【下一步】,进入预览页面:



- 确保信息配置无误后,单击【创建】;
- 应用状态变为"运行中",表示应用创建成功;



3 资源管理

3.1 概述

资源是应用在运行时所依赖的资源实体的抽象,代表一种或多种物理资源实体,如计算、存储、网络等。在 ServiceStage 中执行"创建应用、堆栈、微服务以及订购服务"等操作时时需确保有充足的资源 节点供应用、堆栈运行。

资源管理基本概念如下:

术语	解释
集群	一个集群指容器运行所需要的云资源组合,关联了若干云主机节点、 负载均衡等云资源。
节点	每一个节点对应一台虚拟机,容器应用运行在节点上。节点上运行着 代理程序(kubelet),用于管理节点上运行的容器实例。
其他资源	在"应用 > 资源管理 > 其他"中,用户还可以创建其他资源。
	该资源文件是 ServiceStage 中独立管理的对象,文件内容由用户决 定。资源文件创建完成后,可在容器应用中加载使用。例如,在"数 据卷"中加载资源文件,使其成为容器中的文件,或者在"环境变 量"中,使其成为容器中的环境变量。

3.2 创建集群

集群是容器运行所需要的云资源组合,关联了若干云主机节点、负载均衡等。用户可以创建容器集群 作为私有集群,也可以将应用托管在集群上运行。

目前最多可以创建 2 个集群,创建集群的详细操作请参见 2.1.6 创建集群。请确保至少有一个可用的 集群。

3.3 添加节点

节点是指接入到 ServiceStage 中的计算资源,每一个节点对应一台虚拟机或物理机。在 ServiceStage



中创建应用、堆栈时需确保有充足的资源节点供应用、堆栈运行。

添加节点的详情操作请参 2.1.7 为集群添加节点。

3.4 为节点添加标签

标签是以 key/value 键值对的形式附加节点上的。添加标签后,可通过标签对节点进行管理和选择。 您可以给多个节点打标签,也可以给指定的某个节点打标签,若需了解更多标签信息,请参见11.14 什么 是标签?

操作一:为某个节点添加标签:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【节点】;
- 6. 选择集群,可查看到当前集群下的所有节点;
- 7. 选择节点,点击【添加标签】;



8. 输入键和值,单击【保存】;



加标签			×
键	值	福作	
tag	docker1		
tay	dockeri	1米仔 - 収月	
+) 添加			
		确定	取消

9. 单击【确定】;

操作二:为多个节点添加标签:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 _____,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【节点】;
- 6. 选择集群,可查看到当前集群下的所有节点;
- 7. 选择多个节点,点击【添加标签】;
- 8. 输入键和值,单击【保存】;
- 9. 单击【确定】;

3.5 删除节点

删除该节点会将节点以及节点内运行的应用和服务都销毁,请谨慎操作。

操作一:删除单个节点:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 📴 ,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;



- 5. 单击【应用】【资源管理】【节点】;
- 6. 单击节点后的【删除】链接;
- 7. 根据提示输入节点名称,点击【确定】按钮;

删除		×
0	确定删除此节点 linatest2-l1zpx? 删除该节点会将节点以及节点内运行的应用和服务都销毁。 请输入您的节点名称linatest2-l1zpx以确认删除此节点。	
	linatest2-l1zpx	
	确定取消	

操作二:删除多个节点:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 _____,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【节点】;
- 6. 选择需要删除的多个节点,单击【删除】按钮;

当前您还可以这	b该集群新增14个节点。	余					全部状态	✓ 请输入节点名称		Q C
		所有状态 🗸	CPU (Core)	内存(MB)	弹性IP	节点IP	标签	创建时间 ~	操作	
	linatest-65058	可用	1	2048	183.134.74.142	192.168.1.245	os.architecture amd64	2017/09/18 09:11:11	. 添加标签	删除
								10 > 总条数:1	<	

7. 根据系统提示进行删除操作;





3.6 删除集群

删除该集群会将集群内的节点以及运行的应用和服务都销毁,请谨慎操作。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 3. 选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【集群】;
- 6. 单击待删除集群后的【删除】链接;

🧞 Service Stage	鼠 一 概览 应用 服务 编排 微服务 运维 仓库 设置	♀ 靜杭州 ≗ivycln@126.com@…
◎◎ 应用	▲用 > 資源管理 > 集計	
 资源管理 へ 	请创建集群并添加节点,用于运行您的应用。您最多可以创建2个集群。了解更多	
• 集群 • 节点	十份建筑群	清給入集群名称 Q ピ
• 其他	✓ 集群: linatest2 ● 可用	新增节点 删除
• 命名空间		
🔓 应用管理		1
☑ 任务管理 ∨		±=
B 配置项		

7. 根据系统提示进行删除操作;


3.7 查看集群资源

e Cloud

用户可查看集群的 CPU 总量、内存总量、节点总量。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心, 切换到杭州节点;
- 3. 单击 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【集群】;
- 6. 在右侧页面中,单击集群前的[>],查看 CPU 总量、内存总量、节点总量;

3.8 创建 Secret 资源

Secret 是 Kubernetes 提供的一种用来存储密码、Token 和秘钥等敏感数据的资源类型。

前提条件:已创建资源空间、集群、添加节点。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 3. 选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【其他】;





Cloud

Service Stage	概览 应用 服务 编排 微服务 运维 仓库	设置	으 靜抗州 음 ivych@126	5.com@
⑦ 资源管理 ~	查看、更新、创建、删除集群的其他资源信息。了解更多			
 集群 	集群: linatest v 命名空间: defa	ilt v		
 节点 	NAME			
• 其他	+ 创建资源 删除		全部 ~ 资源名称	Q 💭
•命名空间	☐ 名称	类型 > 标签	创建时间 操作	
🔓 应用管理	redis-3763983392-xfwn0	Pod app redis	… 2017/09/18 10:16:18 GMT 更新 删除	
🖸 任务管理 🗤 🗸				
18 配置项	paas.elb	Secret	2017/09/18 09:19:40 GMT 更新 删除	
	default-token-thppx	Secret	2017/09/18 09:17:18 GMT 更新 删除	
			10 v 2553: 3 🔇 🚺	>

- 7. 在集群下拉框中,选择资源所属的集群;
- ServiceStage 支持"手动输入"和"上传文件"两种方式来创建资源。若需要通过上传文件的方式 创建资源,请确保资源描述文件已创建,具体方法请参见12.1 资源描述文件配置说明;
 - 方式一:手动输入,具体步骤请参见8
 - 方式二:上传文件,具体步骤请参见9
- 9. 选择【添加方式】设【手动输入】;

集群: linatest	∽ 命名空间: default	~
添加方式 手动输入	上传文件	
类型 Secret		
Kubernetes提供的一种用来存	·储密码、Token和秘钥等敏感数据的资源类型	』。 查看更多
<mark>* 描述</mark> 必值.请输入vaml戓ison格式	的溶源描述	

- 类型:选择 "Secret";
- 在描述中,输入 yaml 或 json 格式的资源描述。yaml 格式的资源描述,请参见参考知识部分的 "Secret 资源描述文件"; json 格式的资源描述,请参见 Secret 资源描述文件;
- 配置完成,单击【添加】按钮;

10. 选择【添加方式】设【上传文件】;

	(1) 一 慨觉 应用 服务 骗排 阈服务 运驻 仓库 设直	○ 録杭州 & ivycln@126.com
o الله الله الله	応用 > 強調管理 > 创建改成	
资源管理 ^	集群: linatest v 命名空间: default v	
集群	添加方式 手动输入 上传文件	
节点	翅	
其他	Secret Kuharnata:提供的一种用业方标题起 Takan的组织服务局部局部的影响服务用 素素用文	
命名空间		
如用管理		
£务管理 ∨ _		
尼直项	上传册提出不文件	
米刑.	、选择"Secret"。	
类型:	: 选择"Secret";	
类型:	:选择"Secret"; 【上传资源描述文件】链接。	
类型: 单击	:选择"Secret"; 【上传资源描述文件】链接;	
类型: 单击	:选择"Secret"; 【上传资源描述文件】链接;	
类型: 单击 ^{##: In}	:选择"Secret"; 【上传资源描述文件】链接; watest 、 命密即: default 、	
类型: 单击 ^{集群: [in]} 添加方式	: 选择"Secret"; 【上传资源描述文件】链接; utest v 命名空间: default v	

- 选择已创建的资源类型文件后,单击【打开】按钮;
- 配置完成,单击【添加】按钮;
- 11. 资源创建完成后,您还可以执行更新资源、删除资源等操作;

3.9 创建 DaemonSet 资源

DaemonSet 能够让所有(或者一些特定)的节点运行同一个应用实例。当节点加入到 kubernetes 集群中, 应用实例会被 DaemonSet 调度到该节点上运行, 当节点从 kubernetes 集群中被移除, 被 DaemonSet 调度的应用实例会被移除, 如果删除 DaemonSet, 所有跟这个 DaemonSet 相关的应用实例都会被删除。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 📴 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;

天翼六

取消



5. 单击【应用】【资源管理】【其他】;

Cloud

- 6. 在右侧页面中,单击【创建资源】按钮;
- 7. 在集群下拉框中,选择资源所属的集群;
- ServiceStage 支持"手动输入"和"上传文件"两种方式来创建资源。手动输入参见步骤 8, 上传 文件参见步骤 9;
- 9. 选择【添加方式】设【手动输入】;

集群: linatest v 命名空间: default v	
添加方式 手动输入 上传文件	
共型 DaemonSet	~
Kubernetes提供的一种用来创建Daemon Pod的资源类型,用于在Node上运行守护进程。 查看更多	
* 捕送 业公填,请输入yamli或json格式的资源销益生	

- 类型:选择"DaemonSet";
- 在描述中,输入 yaml 或 json 格式的资源描述。yaml 格式的资源描述,请参见参考知识部分 的 "DaemonSet 资源描述文件"; json 格式的资源描述,请参见 DaemonSet 资源描述文件;
- 配置完成,单击【添加】按钮;
- 10. 选择【添加方式】设【上传文件】;

集群: linate	st v 命名空间: default v		
添加方式	手动输入 上传文件		
<u> </u>			~
Kubernetes	提供的一种用来创建Daemon Pod的资源类型,用于在Node上运行守护进程。 查看更多		
描述文件	未上停任间描述文件 正任资源描述文性 支持json或yaml格式,文件大小不得超过2MB		
		添加	取消
• 类	<型:选择"DaemonSet";		

- 单击【上传资源描述文件】链接;
- 选择已创建的资源类型文件后,单击【打开】按钮;
- 配置完成,单击【添加】按钮;



11. 资源创建完成后,您还可以执行更新资源、删除资源等操作;

3.10 创建 Pod 资源

在 Kubernetes 中, Pod 是能够创建、调度和管理的最小部署单元,是一组容器/进程的集合,而 不是单独的应用容器/进程。Pod 对应于合设在一起的一组应用,它们运行在一个共享的应用上下文中, 同一个 Pod 里的容器/进程共享同一个网络命名空间、IP 地址及端口空间和卷。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 3. 选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【其他】;
- 6. 在右侧页面中,单击【创建资源】按钮;
- 7. 在集群下拉框中,选择资源所属的集群;
- ServiceStage 支持"手动输入"和"上传文件"两种方式来创建资源。手动输入参见步骤 8, 上传 文件参见步骤 9;
- 9. 选择【添加方式】设【手动输入】;

集群: linatest v 命名空间: default v	
添加方式 手动输入 上传文件	
类型 Pod	\sim
在Kubernetes中,Pod是能够创建、调度和管理的最小都智单元,是一组容器/进程的集合,而不是单独的应用容器/进程,Pod对应于合设在一起的一组应用,它们运行在一个共享的应用上下文中,同程共享同一个网络命名空间、IP地址及端口空间和卷。重着更多]一个Pod里的容器/进
* 描述 必填,请输入yaml或json格式的资源描述	

- 类型:选择"Pod";
- 在描述中,输入 yaml 或 json 格式的资源描述。yaml 格式的资源描述,请参见参考知识部分的 "Pod 资源描述文件"; json 格式的资源描述,请参见 Pod 资源描述文件;
- 配置完成,单击【添加】按钮;

10. 选择【添加方式】设【上传文件】;



集群:	linatest	~	命名空间:	default	~											
添加方	式	手动输入 上传文	件													
类型 Pod																~
在Ku 程共享	bernetes 司一个网络	中,Pod是能够创建、认 各命名空间、IP地址及薪	周度和管理的最/ 们空间和卷。	小部署单元,是一约 适着更多	9容器/进程的集	合 , 而不是	是单独的应用	用容器/进程	,Pod对	应于合设在	一起的一组应	i用,它们运行	i在一个共享	的应用上下文	江中,同一个呼	Pod里的容器/进
描述文	件 🦻	未上传任何描述文件														



- 类型:选择"Pod";
- 单击【上传资源描述文件】链接;
- 选择已创建的资源类型文件后,单击【打开】按钮;
- 配置完成,单击【添加】按钮;

11. 资源创建完成后,您还可以执行更新资源、删除资源等操作;

3.11 创建 LifeCycle 资源

LifeCycle 主要用来定义一系列的生命周期事件。当一个生命周期事件被触发,生命周期对应的一个 或一系列生命周期脚本会被调用执行,完成应用的安装、启动、停止、删除等一系列的生命周期动作。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【资源管理】【其他】;
- 6. 在右侧页面中,单击【创建资源】按钮;
- 7. 在集群下拉框中,选择资源所属的集群;
- ServiceStage 支持"手动输入"和"上传文件"两种方式来创建资源。手动输入参见步骤 8, 上传 文件参见步骤 9;
- 9. 选择【添加方式】设【手动输入】;



集群:	linatest v 命名空间: default v	
添加方	式 手动输入 上传文件	
类型 LifeCy	ycle	\sim
LifeC	ycle用来定义一系列的生命周期事件,当一个生命周期事件被触发,生命周期对应的一个或一系列生命周期脚本会被调用执行,完成应用的安装、启动、停止、删除等一系列的生命周期动作。 宣君更多	
* 描述 必填,	请输入yamli或json格式的资源描述	

- 类型:选择"LifeCycle";
- 在描述中,输入 yaml 或 json 格式的资源描述。yaml 格式的资源描述,请参见参考知识部分的 "LifeCycle 资源描述文件"; json 格式的资源描述,请参见 LifeCycle 资源描述文件;
- 配置完成,单击【添加】按钮;

10. 选择【添加方式】设【上传文件】;

集群:	linatest v 命名空间: default v	
添加方	式 手动输入 上传文件	
类型 LifeCy	ycle	\sim
LifeC	ycle用来定义一系列的生命周期事件,当一个生命周期事件被触发,生命周期对应的一个或一系列生命周期脚本会被调用执行,完成应用的安装、启动、停止、删除等一系列的生命周期动作。查着更多	
描述文(件 未上传任何描述文件	
	支持son或yaml格式, 文件大小不得超过2MB	

- 类型:选择"LifeCycle";
- 单击【上传资源描述文件】链接;
- 选择已创建的资源类型文件后,单击【打开】按钮;
- 配置完成,单击【添加】按钮;
- 11. 资源创建完成后,您还可以执行更新资源、删除资源等操作;





4 应用开发

4.1 概述

微服务云应用平台(ServiceStage)提供贯穿开发、部署、运行时保障、治理的端到端的微服务解决 方案,帮助用户以最低成本,最安全的方式进行微服务的开发和管理。

在微服务概览页提供了对微服务开发、发布、部署、管理相关内容的入口:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 点击【微服务】【概览】,进入微服务概览页面;
- 6. 单击【微服务开发】【创建本地工程】,创建过程参见 4.3.1 本地工程的内容;
- 7. 单击【微服务开发】【契约编辑器】,进入 ServiceStage API Editor 页面;

API Editor ⊥ 1 ≓ ∂	success	预 宽	保存	取消
<pre>1 bragger: 72.0" 2 info 3 title: rest test 4 version: 10.0 5 5 6 the domain of the service 7 # host: servicestage.huclouds.com 9 # array of all schemes that your AFI supports 10 * ichnemes: 11 - https </pre>	(Base urt: /controller)			
13 # will be prefixed to all paths 14 basedwith /controller 15 - produces 16 - application/json 17 - online	Solares Sectors			
19 /add: 20 set:	e default			\sim
21 operationId: add 22 - parameters: 23 - name: a 24 in: query	act /add			
25 required true 26 type: integer 27 format: int22	POST /sayhello/{name}			
28 - name b 29 in: query	POST /saysomething			
31 type: integer 32 format: int32	сет /sayhi			
33 - responses: 34 - 2000: reption: add numer 35 - description: add numer 36 - Schema:	GET /sayhei			
38 format: int32 39	Models			\sim
40 /skyNellO(make): 41 - point 42 operation16: sayWello 43 - parameters:	Person > () 🚛			

- 8. 单击【微服务发布】【软件仓库】,进入镜像仓库页面,操作流程请参见7.2镜像仓库的内容;
- 9. 单击【微服务部署】【部署】,进入应用列表,操作流程请参考 5.3 创建应用的内容;



- 10. 单击【微服务管理】【微服务目录】,进入【微服务管理】【微服务目录】页面,查看微服务信息, 请参考 4.5.1 微服务目录的内容;
- 11. 单击【微服务管理】【微服务治理】,进入【微服务管理】【微服务治理】页面,查看微服务信息, 请参考 4.5.2 微服务治理的内容;

4.2 应用场景

• 可视化运维

用户订购了发布在 ServiceStage 上的微服务后,通过访问微服务列表,进入可视化界面,查看微服 务状态和进行微服务治理,减少企业在运维方面投入的人力和物力。

• 监控服务状态

系统运维人员可以查看微服务实时状态,包括 QPS、请求成功率等,可以根据实际情况设置告警阈值, 当指标超过所设阈值后对微服务进行治理操作。

服务治理

运维人员可以在服务库中选择需要治理的微服务,根据需要选择合适的治理策略。服务治理提供了熔 断、容错、限流、降级等高级服务治理能力,最大限度保障服务的可用性。

快速定位故障

运维人员可以在服务治理页面查看服务依赖关系,了解服务的生产者/消费者模型,能够快速精准地定 位到故障服务实例。

4.3 微服务工程

微服务工程模块下的本地工程页面可以创建和生成微服务工程。

4.3.1 本地工程

本节介绍创建微服务工程,协助用户自动生成一个基于微服务框架的开发环境。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心, 切换到杭州节点;



- 3. 单击 🔠 , 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 点击【微服务】【微服务开发】【创建本地工程】;
- 6. 在创建本地工程页面填写一下参数:

(③ 微服务 🗢	微服务 > 微服务开发 > 创	建本地工程	
8 概览	创建本地工程		
	创建本地工程:协助	协您自动生成微服务工程,快速进入业务代码开发阶段。	
 创建本地工程 	* 基于CSE	JAVA-SDK 2.1.4	
 · 开发工具 	* Group:	com.service	
	* Artifact:	小写字母和.号组成,3-24个字符	
	* Java包名称	字母开头,字母、数字组成,3-24个字符	
	* 版本号		
	* Rest Path:		
	* 名称		
	* 开发风格	透明RPC ▼	
	收起高级设置 <		
	创建并下载		

- 基于 CSE: 选择 SDK (Software Development Kit)版本, 目前支持 JAVA-SDK2.1.4;
- Group: 对应源码 maven 工程的 Group ID;
- Artifact: 对应源码 maven 工程的 Artifact ID;
- Java 包名称:对应源码工程中的 package 名称;
- 版本号:微服务版本号;
- Rest Path: 对应发布接口的 URL, 用户可自定义填写, 如"/sayhello";
- 名称:微服务名字,如 "Hello";
- 开发风格:系统提供了透明 RPC、spring-mvc、jax-rs 三种开发模式供选择,缺省配置透明 RPC;



 7. 单击【创建并下载】,完成微服务工程的创建。微服务工程创建以后,系统会自动生成基于微服务 框架的开发工程包并保存到本地;

4.4 微服务调试

微服务调试模块下的云上调试页面可以对微服务实例中的 Java 程序进行远程调试。

4.4.1 云上调试

云上调试主要用于调试人员对单个集群内微服务实例中的 Java 程序进行远程调试。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 ,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 点击【微服务】【微服务开发】;
- 6. 单击【开发工具】,进入工具和案例页面;
- 7. 在【云上调试工具】区域框,单击目标文档对应区域的 ᆇ ,可以下载;

(③ 微服务 🗢	他跟好 > 他跟好开发 > 开发工具
28 概选 四 微服务开发 ^	CSE-SDK CSE-SDK是一个支持高速传送的框架,它可以让您通过一个易用的方式来构建自己的微服务。另外,CSE-SDK提供了大量开着即用的特性,例如消费请服务发现,多种容 错机刺印编断,让您的微服务更可能更安全。服务中心地址:https://100.125.7.22.30100,配置中心地址:h <mark>ttps://100.125.7.22.30100</mark> ,仪表盘上报地址:https: //100.125.7.22.30109。
 ・ 创建本地工程 ・ 开发工具 〇〇 微服务管理 × 	Java Version: 2.1.4 • Go
	待发布 更新时间:2017-08-12 09:55 ■ cse-sdk-2.1.4.zip (29625.4.KB)
	▲ cse-sdk-pdf-2.1.4.zip [2100.4KB] 本地轻量化服务中心 用于服务元数据以及服务实例元数据的管理和处理注册、发现。

4.5 微服务治理

e Cloud

如果使用微服务框架开发应用,在应用托管后启动应用,微服务会自动注册到对应的微服务引擎,您可以 到微服务引擎控制台进行服务治理的相关操作。

目前 ServiceStage 提供了名称为"Cloud Service Engine"的 ServiceComb 微服务引擎,您可以直接使用。也可以创建专享版微服务引擎(支持 ServiceComb)

4.5.1 仪表盘

您可以通过仪表盘实时查看微服务运行相关的指标,根据丰富实时的仪表盘数据,对微服务做相应的治理 动作。

Q

框中输入微服务名称,查询

前提条件

已经进入微服务控制台。

操作步骤

步骤1 选择"仪表盘",在仪表盘页面,在 微服务,页面将展示筛选出的微服务的运行指标。

步骤 2 选择排序方式,筛选出的微服务会按照指定方式进行排序。

当前微服务支持按照吞吐量、平均时延或请求数进行排序。

*	仪表盘 您可以通过仪表盘实时查看微服务运行时的指标,以对微服务做出合适的治理动作。如何(使用仪表盘?
微服务控制台 cse-9b6umz	排序方式: 吞吐量 平均时延 请求数 查看7	示例图 全部应用 → 微服务名称 Q C
仪表盘	请求超时数 线程池拒绝数 请求失败率	×
服务目录	请求数 Amore advice #	圆圈的颜色和大小分别表示
服务治理	7418月2日前款 请求失败数 36178 19 0%	请求的健康度和流量,健康 度从绿色、黄色、橙色、红
全局配置		色递减,流量越大该实心圆 就越大。
事务看板	香む量: 54.0/S 均断状态: Closed	五分钟内的请求遗率趋势
	实例数 370 90th 10ms 中位数时延 1ms 99th 24ms 平均时延 4ms 99.5th 40ms	百分位时延,如图90%的请 求小于70ms,99%请求小于 20年 0.5%进步,下
	注: 1、以上除请求读率趋势图及健康度图以外都是过去10秒内的上报数 2、java sdk 2.3.0版本之前的都是总请求数, 2.3.0+是过去十秒内的	24ms, 99.5%前來小于 40ms 約总请求数。

4.5.2 服务目录

e Cloud

您可通过微服务目录查看微服务详细信息、搜索目标微服务等。

"服务目录"页面展示如下信息:

●应用列表:显示当前用户的所有应用,支持使用应用名称搜索目标应用。

●微服务列表:显示当前用户的所有微服务,支持使用微服务名称搜索目标服务,也可以筛选指定应用下的服务列表。

●实例列表:显示当前用户的所有实例,支持使用微服务名称搜索目标实例、筛选指定应用下的实例 列表以及通过实例诊断结果筛选实例列表。

用户可以先在服务目录创建服务,启动微服务后,根据 yaml 文件的配置,会在对应的服务下注册服 务实例。

如果没有事先创建服务或者服务已删除,启动微服务后,根据 yaml 文件的配置,在注册服务实例时 会自动创建服务。

服务在创建以后,需要注册服务实例后才能进行对应操作。

前提条件

已经进入微服务控制台

创建微服务

步骤1 选择"服务目录"进入微服务列表页。

步骤 2 单击"创建微服务",打开"创建微服务"页签,设置以下配置项。

配置项	说明
微服务名称	新建微服务的名称。
所属应用	新建微服务所属应用。
版本	新建微服务的版本号。
描述	对新建微服务的说明。

步骤3 单击"创建"完成新建微服务。

 	创建微服务 * ^{微服务名称}	数字字母开头结尾,例如:myServiceName
cse-9b6umz	* 所属应用	数字字母开头结尾,例如:myAppId
仪表盘	* 版本	1.0.0
服务目录	描述	
服务治理		
全局配置		0/25
事务看板		

查看微服务详情

天翼口 e Cloud

步骤1 选择"服务目录"进入微服务列表页。

步骤 2 单击微服务名称,在"基本信息"中查看微服务详情。

基本信息	版本	0.0.1	•			◎ 治理	<u>订</u> 删除
动态配置	创建时间	2019-06-19 11:51	:05 GMT+08:00				
灰度发布	描述	此微服务为系统自 实例有效期为7天	动生成 , 目的是 , 如你不需要可以	方便你体验CSE相关功能。此服务不 以在微服务目录界面进行删除。	会占用你的资源且不会产生任何素	贵用 ,	
	标签	标签管理					
	实例列表	被调用服务	调用服务	服务契约			
	微服务名称	称		所属应用	版本	创建时间	
	consumer	r		example	0.0.1	2019-06-19 11:51:05 GMT+08:00	

添加标签

步骤1 在微服务基本信息页,单击标签后的 "标签管理",弹出编辑页面。

步骤 2 在编辑页面,单击"添加标签",输入"键"和"值"。

步骤3 单击"确定"保存修改。



基本信息	版本创建时间	标签管理			×
灰度发布	描述	键	值	操作	,
	标签	不超过64个字符	不超过64个字符		
		⊙ 添加标签			
	实例列				
	微服务		确定取消		9
	consume	exa	ample	0.0.1	20

动态配置

步骤 1 选择"服务目录"进入微服务列表页,单击目标微服务,进入微服务详情页面,单击"动态配置",进入动态配置页签。

基本信息	 ● 创建配置 □ 导入 □ 全部写 	出	所有作用域	• 1	星项或值	Q
动态配置	作用域 🝦	配置项 ♦	值 🗧		操作	
灰度发布						
		 「 「 「 」 「 」 」 「 」 」 」 」 」 」 」 」 」 」 」 」 」				

步骤2 在动态配置页签可进行如下设置。

操作	步骤
导出配置	选择一个作用域,单击"全部导出", 导出当前作用域的所有配置 项。
导入配置	 单击"导入",选择作用域。作用域的格式如下: 微服务名称@所属应用 微服务名称@所属应用#版本号 单击"…",选择目标文件。 单击"上传文件",批量导入配置项。
创建配置	 1. 单击"创建配置",选择配置作用域。 2. 输入配置项。 3. 输入值。 4. 单击"确定"保存配置。
修改配置	 1. 单击目标配置项对应"操作"列的"编辑"。 2. 在编辑框输入"值"。 3. 单击"确定"保存配置修改。



操作	步骤	聚	
删除配置	1. 单击 2. 在"	"操作"列的"删除",弹出"确定"对话框。 删除配置"对话框中,单击"确定",删除配置。	
基本信息	 创建商品 作用域 ÷ 	创建配置	×
从度及作		 配置项不要带有敏感数据,存储都是明文保存,请知悉! 作用域 provider@example ~ 	
		* 配置项 例如:myConfigKey	
		0/256 * 值 例如:myConfigValue	
		0/256	
		确定 取消	

基本信息	⊙ 创建配置	导入配置			×
灰度发布	作用域 💠	 配置项不要帮 作用域 上传配置项文件 	带有敏感数据,存储都是明文保存,请知悉 provider@example 请选择文件,格式为Json	• 上传文件	
		_	关闭		

基本信息 动态 调置 灰度发布	 创建配 作用域 ÷ provider@ 	删除配置 确定要对以下 1 个配置进行删除操作吗 删除该配置项,服务内的配置将失效。	×	
		配置项	值	I
		myConfigValue	myConfigValue	
		确定	取消	



灰度发布

步骤1 选择"服务目录"进入微服务列表页。单击目标微服务,进入详情页面,单击"灰度发 布",进入灰度发布页面。

基本信息	十 添加发布规则				
な度治布	規则名称	作用域	灰度发布方式	规则	操作
			暂无数据		

步骤 2 单击"添加发布规则"。

- 按照权重方式添加灰度规则。
- 1、选择"权重",显示权重规则设置页签。
- 2、对权重规则进行设置,权重规则配置项如下表所示。

配置项	说明
规则名称	自定义规则名称。
作用域	选择规则作用的版本。
规则配置	配置每个分组的引流权重,流量会按照配置的百分比比率流向给 对应的服务分组。

3、单击"确定",完成权重规则配置和灰度发布。

- 按照自定义方式添加灰度规则。
- 1、选择"自定义",显示自定义规则设置页签。
- 2、对自定义规则进行设置,自定义规则配置项如下表所示。

配置项	说明
规则名称	自定义规则名称。
作用域	选择规则作用的版本。
规则配置	 参数名 根据契约的 Key,自定义参数名。要求英文字母开头,支持英文和 数字组成,最多 24 个字符。 规则 根据契约的 Key 对应的 Value,使用数字、字母和括号内的特殊字符 (@#\$%^&*+:,.?{][()\/]),长度 1-24 位。 说明

×



配置项	说明
	● 当匹配符选~时,可使用*或者?字符的模糊匹配规则自定义规则值,*表示任
	意长度字符,?表示一个字符。比如,字段 Name 的规则值配置为*1000 时,
	表示 Name 后 4 位为 1000 的都能匹配上。
	● 当匹配符没有选~时,*或者?字符只是普通字符,没有模糊匹配的作用。

3、单击"确定",完成自定义规则配置和灰度发布。

创建新规则

		-	
1	★ 规则名称	self_rule_522046	
f	乍用域		
	* 版本	☑ 0.0.1	
		2 是否添加自定义版本	
ŧ	则配置		
1	★ 权重百分比		50
		0%	100%
	• ["0.0.1"]微服务	版本将平均分配50%的流量	
	• 剩余50%的流量	按照原有的路由策略分配到其他的实例上	

取消

删除微服务

步骤1 选择"服务目录"进入微服务列表页。

选择需要删除的微服务,单击"删除",弹出"确定"对话框。

步骤 2 在对话框中输入 "DELETE", 单击 "确定"删除微服务。



服务目录	微服务和	实例的维度查看微服务详细信息。如果都	就服务较多,您可以通过搜索;	的能查找目标微服务。如何维护微服务	?			
应用列表	删除	微服务			×			
 创建微照 微肌 		确定要对以下1个微服务及其 删除后将无法恢复。如果删除的服务并 请在下方的输入框中输入"DELETE"确	t,否则服务会再次注册。 		¥	微服务名称	Q C	
_ pro		请输入DELETE确认删除				T+00.00	2019PT	
	微服	务名称	所属应用	更新时间		1+00.00	JUL9 POR	
	provi	der	example	2019-06-19 11:51:05 GMT+08:00				
	删除微服务的同时,会把微服务下的动态配置同步删除。							
			确定 取消					

🗀 说明

当微服务实例个数为0时,可直接删除微服务。

当微服务实例个数不为0时,删除微服务后过一段时间微服务将会重新注册到服务中心。

当微服务被依赖时,不可以删除微服务。

4.5.3 服务治理

您在部署完微服务后,可以根据微服务的运行情况进行微服务的治理。

前提条件

已经进入微服务控制台。

概述

步骤1 选择"服务治理",进入微服务治理页面。



*	服务治理	
\mathbf{C}	您可以在微說另部署完成后,根据微顯务的运行情况对服务进行治理。如何治理微服务?	
微服务控制台 Cloud Service Engine		全部应用(1) ~ C
仪表盘	example 下线	
服务目录	微服务数量 2 万雄 2 正常 0 异常 0	
服务治理		
全局配置		
事务看板		
	Consumer#0.0.1	

步骤 2 单击需要治理的微服务,进入微服务监控页面和治理配置页面。

在微服务治理配置页面支持负载均衡、限流、容错、降级、熔断和错误注入等策略的配置。

名称	说明
负载均衡	当出现访问量和流量较大,一台服务器无法负载的情况下,我们可以 通过设置负载均衡的方式将流量分发到多个服务器均衡处理,从而优 化响应时长,防止服务器过载。
	可以通过新增规则配置负载均衡策略,设置参数支持轮询、随机、响 应时间权值、会话粘滞等多种负载均衡策略。
限流	限流主要解决微服务之间的流量分配问题,保证微服务在自己的资源 池运行,互不影响。
	 当限流对象对当前服务实例的每秒请求数量超过设定的值,当前服务实例不再接受该对象的请求。
	• 常用的检测方法是请求超时、流量过大等。
	● 设置参数包括限流对象、QPS 阈值等。
降级	降级是容错的一种特殊形式,当出现服务吞吐量巨大,资源不够用等 情况,我们可使用降级机制关掉部分不重要、性能较差的服务,避免 占用资源,以保证主体业务功能可正常使用。
容错	容错是服务调用者访问服务实例,服务实例出现异常时的一种处理策 略,出现异常后按照定义的策略进行重试或访问新的服务实例。
	服务调用出错异常时的处理模式。一旦发生异常,服务会根据容错机 制来进行尝试重新访问或直接返回失败。
	当我们发现由于某些原因导致服务出现了过载现象,为避免造成整个 系统故障,可采用熔断来进行保护。
	熔断在服务请求处理出现异常时产生作用。进入熔断状态后,hystrix 会认为被请求的服务已经无法处理请求,在第一时间截断请求直接返



名称	说明
	回错误给调用者。
	hystrix 每隔一段时间会尝试访问后端服务,如果服务恢复正常,会退 出熔断状态,恢复正常的请求访问。
错误注入	错误注入策略用于测试微服务的容错能力,可以让用户知道,当出现 延时或错误时,系统是否能够正常运行。
	错误注入通过延时、错误等方式,供用户测试微服务的容错能力。
黑白名单	黑白名单的主要功能是为微服务设置黑白名单。
	黑白名单是为了改变网络流量所经过的途径而修改路由信息的技术。

1														
<	provider#0.0.1 example 2 监控 ③ 阈值					服务过滤	provid	ler#0.0.1	×	状	态选择	All	有实例	0
	0 0 NaN 0 0 0 石仕屋 NaN 细節状态 NaN													
	实例数 0 90th NaN 中位数时延 NaN 99th NaN 平均时延 NaN 99.5th NaN)					
	诊断结果						/	provider#0	0.1					
分 特	(戦均衡(0) 限点(0) 降级(0) 容措(0) 焰断(0) 11男注入(0) 黒白名单(0)													
	+ 新增				,	 .° .								
		まこんほう												

设置负载均衡

- 步骤1 单击需要治理的微服务,进入微服务治理配置页面。
- 步骤 2 在微服务治理配置页面,单击"负载均衡",展开负载均衡详情页。
- 步骤 3 单击"新增",进入负载均衡策略配置。先选择需要治理的微服务,再选择合理的负载 均衡策略,请参见下表。

策略名	策略说明					
轮询	支持按照服务实例的位置信息顺序路由。					
随机	提供服务实例随机路由。					
响应时 间权值	提供最小活跃数(时延)的权重路由,支持业务处理慢的服务实例 接收较少的请求,防止系统停止响应。这种负载均衡策略适合请求					



策略名	策略说明
	量少且稳定的应用。
会话粘 滞	会话粘滞是负载均衡器上的一种机制,在设定的会话保持时间内, 会保证同一用户相关联的访问请求会被分配到同一实例上。
	• 会话保持时间:会话保持的限制时间, 0-86400, 单位为秒。
	 失败次数阈值:访问失败次数,0-10。当微服务访问下属实例的失 败次数和会话保持时间超过设定的值时,微服务不再访问该实例。

步骤4 单击"确定"保存配置。

负载均衡(0) 阴	読(0) 降级(0) 容错(0)	熔断(0)
错误注入(0) 黑	(0)	
	响应时间权值	
	随机	
选择微服务	会话粘滞 轮询	
负载均衡策略	响应时间权值	*
	确定 取消	

设置限流

步骤1 单击需要治理的微服务,进入微服务治理配置页面。

步骤 2 在服务治理配置页面,单击"限流",展开限流详情页。

步骤3 单击"新增",进入限流策略配置页面,限流配置项下表所示。

配置	说明	范围
限流 对象	(微服务)访问者。	该应用依赖的微服务,下拉菜 单可直接选择。
QPS	每秒的请求数。当限流对象对当前服务实例的每秒请求数量超过设定的值,当前服务实例不再接受该对象的请求。	0-99999 的整数。

步骤4 单击"确定"保存设置。

负载均衡(0) 限調 错误注入(0) 黑日	充(0) 降级(0) 日名单(0)	容错(0)	熔断(0)		0	 	
	十 新増	1			2		
限流对象	所有微服务		•		<u>.</u>	.≏	-
* QPS	1,000			请输入0-9	。 99999之	间的整	鈫
	确定	取消			2		

设置降级

天翼云 e Cloud

- 步骤1 单击需要治理的微服务,进入微服务治理配置页面。
- 步骤 2 在服务治理配置页面,单击"降级",展开降级详情页。
- 步骤3 单击"新增",进入降级策略配置页面。
- 步骤4 在降级策略配置页面选择合理的策略,降级策略配置项如下表所示。

配置项	配置项说明
降级对象	选择需要降级的服务,可以具体到方法。
降级策略	 开启:开启降级 关闭:关闭降级



负载均衡(0) 错误注入(0)	限流(0) 降级(0) 容错(0) 熔断(0) 黑白名单(0)
	┿ 新増
降级对象	所有微服务
降级策略	所有方法 ▼ ● 开启 ○ 关闭
	确定 取消

步骤5 单击"确定"保存设置。

设置容错

步骤1 单击需要治理的微服务,进入微服务治理配置页面。

步骤 2 服务治理配置页面,单击"容错",展开容错详情页。

步骤3 单击"新增",进入容错策略配置页面。

步骤4 在容错策略配置页面选择合理的策略,容错策略配置项如下表所示。

配置项	配置项说明
容错对象	该应用依赖的应用或方法,下拉菜单可直接选择。
是否开启容错	开启:向容错对象发起请求时发生错误的处理策略,开启后, 会根据选择的处理策略处理请求。
	关闭:关闭容错策略,即使请求失败也会等到超时后,再返回 失败结果。
容错策略	Failover:在不同服务器上重新尝试建立连接。
说明 当"是否开启容错"配 置ر设置为"开启"时	Failfast:不再重新尝试建立连接,即请求失败时会立即返回失 败结果。
配置。	Failback:在同一个服务器上重新尝试建立连接。



配置项	配置项说明
	custom: 同一个服务器上尝试重新建立连接的次数,取值范围 0-9 的整数。
	• 在不同服务器上尝试建立连接的次数,取值范围 0-9 的整数。

步骤5 单击"确定"保存设置。

负载均衡(0) 限流(0) 降级(0) 容错(0) 熔断(0) 错误注入(0) 黑白名单(0)

十 新增						
容错对象	所有微服务	•				
是否开启容错	● 关闭 🛛 开启					
	确定 取消					

设置熔断

- 步骤1 单击需要治理的微服务,进入微服务治理配置页面。
- 步骤 2 在服务治理配置页面,单击"熔断",展开熔断详情页。
- 步骤3 单击"新增",进入熔断策略配置页面。

步骤4 在熔断策略配置页面选择合理的策略,熔断策略配置项如下表所示。

配置项	配置项说明	取值范围
熔断对 象	针对下属的微服务实例, 可以具体到方法。	该应用调用的服 务或方法,下拉菜 单可直接选择。
触发条 件	手动熔断:即刻触发熔断,且不再调用。	-
	取消熔断:将已经熔断的微服务实例取消	-



配置项	配置项	〔说明	取值范围	
	熔断,	继续调用。		
自动熔断		熔断时间窗:熔断的持续时 间,以毫秒为单位。通俗点来 说就是让这个实例休息一段 时间后,再重新调用。	0-99999 的整数	
		失败率:请求的失败率,以% 为单位。	0-100 的整数	
		窗口请求数:发起的请求数。 失败率和窗口请求数的条件 需同时满足才会触发熔断。	1-10000 的整数	

步骤5 单击"确定"保存设置。

负载均衡(0)	限流(0) 降级(0) 容错(0) 熔断(0)
错误注入(0)	黑白名单(0)
	十 新增
熔断对象	所有微服务
	所有方法
触发条件	 手动熔断 取消熔断 自动熔断
	确定取消

设置错误注入

- 步骤1 单击需要治理的微服务,进入微服务治理配置页面。
- 步骤 2 在服务治理配置页面,单击"错误注入",展开错误注入详情页。
- 步骤3 单击"新增",进入错误注入策略配置页面。
- 步骤4 在错误注入策略配置页面选择合理的策略,错误注入策略配置项如下表所示。



天翼 云 e Cloud
0 010010

配置项	配置项说明				
注入对象	需要测试容错能力的微服务,可以具体到微服务的方法。				
类型	测试微服务的容错能力类型:				
	• 延迟				
	• 错误				
协议	访问微服务出现延时或错误的协议:				
	• Rest				
	• Highway				
延迟时间	访问微服务出现延时的延时时长, "类型"选择为"延迟"时才需要配置。				
	单位:毫秒				
	取值范围:0-3600				
http 错误码	访问微服务出现错误的 HTTP 错误码, "类型" 选择为 "错误" 时才需要配置。该错误码为 HTTP 标准的错误码。				
	取值范围:100-999				
触发概率	访问微服务出现延时或错误的概率。				
	取值范围:0-100				



负载均衡(0)	限流(0)) 降级(0)	容错(0)	熔断(0)
错误注入(0)	黑白名	单(0)				
		+	新増			
注入对象		所有微服	务		•	
		所有方法			•	
类型	c	延迟	0	错误		
协议	c	Rest	0	Highway	r	
* 延迟时间()	ms)	10				
* 触发概率()	%)	50				
		确定]	取消		

步骤5 单击"确定"保存设置。

设置黑白名单

须知

如果要使用黑白名单,则微服务与服务中心需要满足以下流程:

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

步骤1 单击需要治理的微服务,进入微服务治理配置页面。

步骤 2 在服务治理配置页面,单击"黑白名单",展开黑白名单详情页。

步骤3 单击"新增",为应用添加黑白名单,黑白名单配置项如下表所示。



配置项	配置项说明
类型	黑名单:表示根据匹配规则匹配到的微服务都不允许访问当前服务。
	白名单:表示根据匹配规则匹配到的微服务允许访问当前服务。
匹配规	使用正则表达式表示。举例如下:
则	匹配规则:data*
	表示名称以 data 开头的服务不允许其他服务访问。

步骤4 单击"确定"保存设置。

负载均衡(0)	限流(0)	降级(0)	容错(0)	熔断(0)
错误注入(0)	黑白名单((0)		
		十 新増	1	
类型	0	黑名单	○ 白名単	
★ 匹配规则	0			
	ā	航 定	取消	

异常诊断

当微服务正常运行且与其他微服务存在调用关系时,才能进行诊断操作。

- 单个微服务诊断
 - 1、单击需要诊断的微服务,进入微服务诊断页面。
 - 2、单击基本信息区域的"诊断"。
 - 3、单击"确定"。
- 批量微服务诊断
 - 1、单击需要诊断的微服务,进入微服务诊断页面。

2、单击右侧空白处,此时在左侧展示"运行时异常列表"和"诊断异常列表"。

🗀 说明

运行时异常列表展示吞吐量、平均时延和请求失败率超过阈值的微服务。



诊断异常列表展示缓存实例与实际实例数据不一致的微服务。

- 3、单击"诊断异常列表"。
- 4、单击"全部诊断"。
- 5、单击"确定"。

4.5.4 全局配置

全局配置提供微服务间的公共配置,如日志级别、运行参数等。全局配置添加以后,如果微服务没有 自定义相同的配置项,会将全局配置作为默认配置使用。全局配置限制配置 1000 条配置项。

前提条件

已经进入微服务控制台。

操作步骤

步骤1 选择"全局配置",进入全局配置界面。

<u>_</u>	全局配置		
\sim	您可以设置微服务公共配置,如日志级别、运行	行参数等。添加全局配置后,微服务如果没有自定义相同的配置项,则会将全局配置作为默	认配置使用。 如何设置全局配置?
微服务控制台 Cloud Service Engine	 ● 创建配置 □ 号入 □ 全部号 	社 立 勝称	配置项或值QC
仪表盘	配置项 ↓	值 🗧	操作
服务目录			
服务治理			
全局配置			
事务看板		₹₽ 誓元数据	

操作	步骤
导出配置	单击"全部导出",导出所有全局配置项。
导入配置	单击"导入"。 单击"···",选择目标文件。 单击"上传文件",批量导入配置项。
添加配置	单击"创建配置",输入"配置项"和"值"。 单击"确定"保存配置。
修改配置	单击目标配置项对应"操作"列的"编辑"。 在编辑框输入"值"。 单击"确定"保存配置修改。

步骤2 在全局配置界面可进行如下操作。



操作	步骤
删除配置	选择所需删除的配置。
	单击"操作"列的"删除",弹出"删除配置"对话框。
	单击"确定",删除全局配置。
批量删除	选择所需删除的配置。
	单击配置项列表上方的"删除",批量删除全局配置项。

<u>*</u>	全局配置	服务公共配置,如日	志极别、运行参数等、添加全局配置后,微服先如果没有自定义相同的配置项,则会将会	全局配置作为默认配置	監使用。如何设置全局配置?
微服务控制台 Cloud Service Engine	⊙ 创建配量	创建配置		×	配置项或値 Q C
仪表盘	配置		要带有敏感数据,存储都是明文保存,请知悉!		操作
服务治理		* 配置项	參如:myConfigKey	 支持数字 度1-2560 	、学母和:四个字符,长 7
全局配置事务看板			0/2	256	
		* 值	部词:myConfigValue		
			0/2	256	
			職定 取消		

全局配置					
您可以设置微服务公共配置,	如日志级别、运行参数等。添加全局配置后,微服务	如果没有自定义相同的配置项,则会将全局配置作为默认配	置使用。如何设	置全局配置?	
 ● 创建配置 〇 日建配置 	☆ 全部导出 □ 删除			配置项或值	QC
☑ 配置项 💠			×	操作	
MyConfigValue1	删IV东西C直				
MyConfigValue	确定要对以下2个配置进行删除操作吗?			编辑删除	
	配置项	值			
	MyConfigValue1	MyConfigValue1			
	MyConfigValue	MyConfigValue			
		确定 取消			

4.5.5 事务看板

您可以通过事务看板界面上展示的事务概览、事务列表和活动事务了解事务的基本信息、运行状态、 执行日志等。

前提条件

已经进入微服务控制台。





事务概览

事务概览区域框中的参数说明如下。

参数	参数说明
成功事务实例数量	累计成功运行完成的事务数量。
回滚事务实例数量	累计回滚事务数量。
TPS	每秒事务处理数量。



事务列表

事务列表展示全部事务的基本信息,参数说明如下。

参数	参数说明
事务名称	事务的名称。
子事务数量	事务包含的子事务数量。
成功	累计成功事务实例数量。
回滚	累计回滚事务实例数量。
活动	当前活动事务实例数量。
超时时间	事务运行超时时间。
事务类型	事务类型,包含:
	• TCC
	• 2PC



心志舟							
服务目录	事务列表						
服务治理	事务名称≜	子事条数量 ≜	成功 ≜	回滚 ≜	活动 👌	招时时间 (ms) ≜	事名类型
全局配署							
事务看板							
			đ				
			- market	I J			
			智	无数据			

活动事务

在"事务列表"中,单击目标行"活动"列的活动事务实例数查看活动事务实例的详细信息。

🗀 说明

"活动事务实例数"为 0 时无法查看活动事务实例的详细信息

活动事务区域框中可查看其当前正在运行的事务及其分支事务的执行状态,参数说明如下。

参数	参数说明
事务实例 ID	字段 Map key 的值为事务实例 ID。
事务实例状态	包含:
	• TCC
	– TRYING
	对事务做检测及资源预留。
	– CONFIRMING
	对事务做确定提交。
	– CANCELLING
	在事务执行出错, 需要回滚的状态下执行的事务取消和预留 资源释放。
	• 2PC
	- Preparing
	强一致事务处于准备中状态。
	- Prepared
	强一致事务准备完成。
	– ACTIVE
	强一致事务处于活动状态,还未开始准备提交。
	– Committing
	强一致事务处于提交中状态。





参数		参数说明
		- Committed
		·····································
子 事 条	子事 务 ID	事务分支 Map 的 key 为分支 ID。
- 务	子 事 务 次	 包含: TCC TRYING 对事务做检测及资源预留。 CONFIRMING 对事务做确定提交。 CANCELLING 在事务执行出错,需要回滚的状态下执行的事务取消和预留资源释放。 2PC Preparing 强一致事务处于准备中状态。 Prepared 强一致事务准备完成。
		 ACTIVE 强一致事务处于活动状态,还未开始准备提交。 Committing
		强一致事务处于提交中状态。 - Committed 强一致事条提交完成
		压

🗀 说明

单击目标行最前面的 , 展开目标事务实例的所有子事务。



1

5.1 创建应用配置项

为了提升易用性,ServiceStage 提供了镜像和配置文件的解耦,以此来保持容器化应用程序的便携性。 若应用配置较多,建议用户在创建应用前添加配置。添加配置后,可以在创建应用时,通过添加卷的方式 进行使用。

前提条件:已创建资源,具体操作请参见错误!未找到引用源。创建资源空间、错误!未找到引用源。 REF _ZH-CN_TOPIC_0053496188-chtext \h创建集群、错误!未找到引用源。为集群添加节点。若已有集 群和节点资源,无需重复操作。

操作步骤如下:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 3. 选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【配置项】,进入配置项页面;

Service St	tage	俞 概览 应	1 服务 编排 微服务 运维 仓库	设置	Q
应用	C	应用 > 配置项]	用 资源管理 集群	日务管理 普通任务	
③ 资源管理	~	查看、添加	节点 一 其他	定时任务	
😵 应用管理		5	☆ 応用管理		
🖾 任务管理	~	集群: cln_		ult ~	
🞖 配置项					

6. 单击【添加配置】或【添加配置项】(初次创建),进入创建页面;

	应用 > 配置项				
③ 资源管理 🗸	查看、添加、删除集群的配置项 ConfigMap 资源	,了解更多配置项的信息,请点击了解更多			
🛚 应用管理					
☑ 任务管理 ∨	集群: ch1 ~ 節名空间:	default ~			
吕 配置项	+ 添加配置			请输入配置名称	Q
	□ 名称 ~	标签	创建时间 ~	操作	
	config-manul	keyl 1	2017/10/26 22:19:16 GM	T+0 更新 删除	
				10 ~ 总条数:1	1 >

- 7. 在【集群】下拉框中,选择配置项所属的集群;
- 8. ServiceStage 支持手动输入、上传文件两种方式来创建配置项。
- 9. 【添加方式】设置为【手动输入】,设置新增参数:
 - 配置名称:必填参数,新建的配置名称,命名必须唯一;
 - 标签:标签以 Key/value 键值对的形式附加到各种对象上(如应用、节点、服务等)。标签 定义了这些对象的可识别属性,用来对它们进行管理和选择;
 - 数据:应用配置的数据可以在容器中使用,或被用来存储配置数据。其中"键"代表文件名, "值"代表文件中的内容;

(日本) 应用	• 5	空用 > 配置项 > 添加配置					
③ 资源管理	~	集群: cln1	✓ 命名空间; def	fault ~			
於 应用管理		添加方式 手动输入 上作	专文件				
☑ 任务管理	~						
吕 配置项		* 配置名称 config-manul					
		标签					
		12		值		操作	
		key1		1		编辑	删除
		④ 添加					
		数据					
		键		值		操作	
		name		ivy		编辑	删除

10. 【添加方式】设置为【上传文件】,单击【上传资源描述文件】。选择已创建的 ConfigMap 类型资源 文件后,单击【打开】。若需要通过上传文件的方式创建资源,请确保资源描述文件已创建。 ServiceStage 支持 json 或 yaml 格式,详细请参见 12.6 ConfigMap 类型资源描述文件配置说明;


11. 配置完成后,单击【添加】按钮,应用配置列表中会出现新创建的应用配置;

12. 配置项创建完成后,还可以执行更新配置、删除配置操作;

5.2 创建任务类应用

e Cloud

从程序运行形态上,应用可分为长时运行服务和一次性任务。大部分应用(如 Nginx、Redis 等)都属于长时运行服务,一次性任务如执行冒烟测试、数据计算等。

ServiceStage 中的短任务指一次性任务。短任务和应用的主要区别在于,普通应用是长时间稳定运行的,而短任务是执行完就退出。

5.2.1 创建定时任务

定时任务是按照指定时间周期运行的短任务。使用场景为在某个固定时间点,为所有运行中的节点做 时间同步。

定时任务需要基于镜像创建,若选择私有镜像,用户首先需要将镜像上传至镜像仓库,具体操作请参见 7.2.3 上传私有镜像(内网)。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【任务管理】【定时任务】;

e eleuu						5 应用上线
Service Stage		应用 服务 编排	微服务运维	仓库设置		
	应用 > 任务管理 >	应用 之 资源管理			任务管理	
		集群			普通任务	
③ 资源管理 🗸 🗸	集群: cin	节点		default B8	定时任务 配置项	
₿ 应用管理		部 应用管理				
☑ 任务管理 🔷 👋						1

6. 在【集群】下拉框中,选择任务所属的集群;

天舅

7. 单击【定时任务创建】,首次创建时,此处展示的是【创建定时任务】;

(日本) 应用	0	应用 > 任务(理 > 定时任务					
\bigcirc								
③ 资源管理	\sim	集群:	cln1	~	命名空间:	default	~	
🔡 应用管理								
💟 任务管理	^							
• 普通任务								
• 定时任务								
😳 配置项								
					定时任务	是按照指定时间周期	运行的批处理日	任务,包含一组容器及其需要执行的命令,暂无任务,您可以 1996年97任务

- 8. 设置定时任务名称,命名必须唯一;
- 9. 选择模式,支持 Allow、Forbid、Replace 三种模式。
 - Allow——定时任务不断新建 Job;
 - Forbid——在前一个任务未完成时,不创建新任务;
 - Replace——已到新任务创建时间点,但前一个任务还未完成,新的任务会取代前一个任务;
- 10. 自定义时间:定时任务时间规则,指定新建定时任务在何时执行。时间各位:*****,第一个* 代表分钟(0-59),第二个*代表小时(0-23),第三个*代表日期(1-31),第四个*代表月份(1-12), 第5个*代表每周的第几天(1-7)。例如,0011*代表每年1月1日凌晨执行任务。此外,*/代表每 隔,比如*/5分钟代表每个5分钟;
- 11. 卷定义: 卷是指 Pod 中能够被多个容器访问的共享目录, 卷与 Pod 的生命周期相同:
 - 单击【添加】按钮;
 - 选择对应的卷类型,并添加配置,参数说明如下表:

卷类型 卷类型解释 步骤说明	
----------------	--



卷类型	卷类型解释	步骤说明
HostPath	在容器上挂载宿主机上的文件或目录。	4. 选择【卷类型】为"HostPath";
		5. 输入【卷名称】,用户可自定义填写;
	1、谷器应用程序生成的日志又件需要求 久保存。	6. 单击【添加配置】;
	2、需要访问宿主机上 Docker 引擎内部	7. 输入【主机路径】,如/var/lib/docker;
	数据结构的容器应用。 	8. 单击【确认】;
		9. 单击操作列的【确认】;
EmptyDir	在容器分配到节点时系统自动创建,初	1. 选择【卷类型】为"EmptyDir" ;
	始内容为空。在同一个 Pod 中所有容器 可以读写 EmptyDir 中的相同文件。当	2. 输入【卷名称】,用户可自定义填写;
	Pod 从节点上移除时, EmpryDir 中的数据也会永久删除。通常用于临时数据的	3. 单击【添加配置】;
	高速存储。	4. 选择【存储介质】;
		默认:存储在硬盘上,适用于数据量大,读写效率 要求低的场景;
		内存:存储在内存中,适用于数据量少,读写效率 要求高的场景;
		5. 单击【确认】;
		6. 单击操作列的【确认】;
Secret	用来处理敏感数据,比如密码、Token	1. 选择【卷类型】为"Secret"。
	和密钥,相比于直接将敏感数据配置在 Pod 的定义或者镜像中,Secret 提供了	2. 输入【卷名称】,用户可自定义填写。
	更加安全的机制,防止数据泄露。应用 纳定服条时,也需选择 Secret 中的"服	3. 单击【添加配置】:
	务访问凭证"进行绑定。	所有凭证:在配置文件下拉框中,选择已配置好的
	说明	Secret 又件。也可以単击【去创建】新建配直又件, 创建配置文件步骤请参见 <u>3.8 创建 Secret 资源</u> ;
	 若应用需绑定服务,需确保应用的 业务逻辑中支持绑定需要的服务。 	服务访问 凭证:为应用绑定平台中的服务。当前不 支持此功能。
	• 创建 Secret 类型资源时,	4. 单击【确认】;
	ServiceStage 会针对 Secret 信息 做加密处理,建议不要在 Secret	5. 单击操作列的【确认】;
	中输入涉及个人隐私的信息。	
ConfigMap	提供应用代码和配置文件的分离,	1. 选择【卷类型】为"ConfigMap";
	Lont Igmap 用于处理应用配直参数。用 户需要提前创建应用配置,操作步骤请	2. 输入【卷名称】,用户可自定义填写;
	参见	3. 单击【添加配置】;



卷类型	卷类型解释	步骤说明
		4. 在应用配置下拉框中选择需添加的配置文件;
		5. 输入配置文件所在的相对路径;
		6. 单击【确认】;
		7. 单击操作列的【确认】;

③ 资源管理 ~ ※ 应用管理	 任务基本信息	普通任务配置	の	् 70डी				
☑ 任务管理 ∧	* 定时任务名称 test							
• 普通任务 • 定时任务 83 配置项	• 旣式 @ ③ Allow ○ Forbid ○ Replace 自定义的问							
	* 138411584169000 0011*	務 款 ④						
	卷定义							
	卷名称	老类型	8625	操作				
	myvolume1	HostPath	/var/lib/docker	编辑修改配置册除				
	emptydirvol	EmptyDir	内存	编辑 修改配置 删除				
	secretvol	Secret	default-secret	编辑 修改配置 删除				
	configmap	ConfigMap	config-manul	编辑修改配置删除				
	④ 添加							
	描述							

12. 描述: 输入对应定时任务的描述;

會应用	C	如用 > 任务管理 > 定时任务 > 创建定时任务			
 资源管理 应用管理 	~		普通任务配置	页范	完成
 任务管理 普通任务 	^	* 定时任务名称 job1			
· 定时任务		* 模式 @ Allow			
🗠 配置项		自定义时间		_	
		* 定时任祭时间规则 0001*	帮助 😡		
		卷定义			
		卷名称 vol1	卷类型 HostPath	配置 /host	操作 编辑 修改配置 删除
		(+) 添加			

- 13. 配置完成后,单击【下一步】,进入容器配置页面;
- 14. 设置容器名称,如 container1;



- 15. 单击【选择镜像】链接,选择对应的镜像。使用公有仓库或私有仓库,需确保镜像已上传至镜像仓库,具体方法请参见 7.2 镜像仓库
 - 公有仓库:显示 ServiceStage 所有用户上传的公有镜像;
 - 私有仓库:显示用户自己上传的公有和私有镜像。若所选镜像仓库为私有类型,选择镜像后, 还需设置认证方式。此处认证方式的取值从 2.1.2 上传 AK/SK 证书处读取;
 - 第三方镜像:使用第三方镜像,如 dockerhub 提供的镜像。a、认证:用户可根据实际需求 选择是否添加认证。若选择"是",请选择已创建好的认证。若没有创建,请单击【创建认 证】创建"类型"为"Sercret"的资源,详细步骤请参见 3.8 创建 Secret 资源; b、镜像 地址:请输入第三方镜像的仓库地址,如 https://hub.docker.com/_/wordpress/;

一副应用	C	应用 > 任务管理 >	定时任务 > 创建定时任务					
			生尽培侮	_	_	_	×	
🚱 资源管理	\sim							_
88 应用管理			公有仓库私有仓库	第三方镜像				
🖸 任务管理	^	容器	租户全部	~			i库名称 Q	
• 普通任务		cont	镜像仓库名称	租户	命名空间 描述	版本	更新时间	
• 定时任务			 redis 	chensl@chi	test	0907 ~	2017-09-07 16:06:48	
🔡 配置项		容) liny	chensl@chi	jaxrsserver	tag ~	2017-08-12 19:44:16	
		cc 镜值	 hostname 	chensl@chi	hostname	1.1 ~	2017-08-09 14:36:37	
		Ð	 hostname 	chensl@chi	hostname	1.1 ~	2017-08-09 14:36:37	
			⊖ test	chensl@chi	hostname	1.0 ~	2017-08-07 17:10:49	
					5 ~ 总条数: 8	(1 2	> 1 跳转	上一步

- 16. 设置容器内存限制: 启动容器时, 限制容器最多能够使用的内存量;
- 17. 设置容器 CPU 限制: 启动容器时, 限制容器最多能够使用的 CPU 量;
- 18. 单击数据卷下方的【添加】链接,并设置数据库相关参数:
 - 容器路径:数据卷挂载到容器上的路径,请不要挂载在系统目录下,如"/"、"/var/run"
 等,会导致容器异常,建议挂载在空目录下;
 - 是否只读:配置为"是",只能读容器路径中的数据卷;配置为"否",可修改容器路径中的数据卷,容器迁移时新写入的数据不会随之迁移,会造成数据丢失;
 - 卷策略及日志策略:配置为 none,不支持卷策略;配置为 logs,当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs。当卷策略配置为 logs 时,可支持用户自定义日志收集的路径,收集的日志在监控页面可以查看。可选策略包括 none(不进行日志回滚转储)、Hour ly (每小时扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一个历史文件中,在 日志所在目录下),并将原文件清空)、Daily(每天扫描一次日志文件,如文件超过 20M,会

て で Cloud

将其转储,并将原文件清空)、Weekly(每周扫描一次日志文件,如文件超过 20M,会将其转储,并将原文件清空);

🕑 资源管理	~	?				0	0	
28 应用管理		任务基本信息		普通任务配置		预览	完成	
🖸 任务管理	^	容器						
• 普通任务		container1 🛞 添加容器						
・定时任务								
58 配置项								
		容器 container1						
		镜像 100.125.7.22:20202/test/redis	版本: 0907 🧪					
		内存限制						
			1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	24 28	0.1 GB			
			, , , , , , , , , , , , , , , , , , ,	6 7	0.3 核			
		数据卷						
		卷名称	容器路径	是否只读	卷策略及日志策略		操作	
		myvolume1	/var/docker/test	是	none		编辑 删除	
		⊕ 添加						
		删除此容器						

19. 配置完成后,单击【保存】按钮;

20. 若需要创建多个容器,请选择【添加容器】;

21. 单击【下一步】,进入预览页面;

22. 点击【创建】按钮,完成定时任务的创建;

副应用	0	应用 > 任务管理 > 定时任务							
③ 资源管理	~	集群: cin1	 命名空间: default 	~					
🛚 应用管理		+ 定时任务创建 删除						全部状态 ~ 请输入任务名称	Q
☑ 任务管理	~	□ 定时任务名称 >	状态 ~	创建时间 ~	时间规则 ~	容器个数	正在运行的任务个数	操作	
• 普通任务		test	已启动	2017/10/27 10:30:59 GMT+0	0011*	0	0	删除停止	
・定时任务									
83 配置项								10 ~ 总乐数 1	9 💶 📎

5.2.2 创建普通任务

普通任务是一次性运行的短任务,部署完成后即可执行。使用场景为在创建应用前,执行普通任务, 将镜像或软件包上传至软件仓库。

定时任务需要基于镜像创建,若选择私有镜像,用户首先需要将镜像上传至镜像仓库,具体操作请参见 7.2.3 上传私有镜像(内网)。

1. 登录天翼云 http://www.ctyun.cn;





- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 ____,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【任务管理】【普通任务】;
- 6. 在【集群】下拉框中,选择任务所属的集群;
- 7. 点击【创建任务】;

应用	(应用 > 任务	管理 > 普通任务				
③ 资源管理	~	集群:	cln1	~	命名空间:	default	•
🔓 应用管理							
🖸 任务管理	^						
• 普通任务							
· 定时任务							
🕄 配置项							
							任务是一次性运行的批处理任务,包含一组容器及其需要执行的命令,暂无任务,您可以 创建任务

- 8. 填写普通任务名称,命名必须唯一;
- 9. 配置卷信息:
 - 单击【添加】按钮;
 - 选择对应的卷类型,并添加配置,参数说明如下表:

卷类型	卷类型解释	步骤说明
HostPath	在容器上挂载宿主机上的文件或目录。	8. 选择【卷类型】为"HostPath";
		9. 输入【卷名称】,用户可自定义填写;
	久保存。	10. 单击【添加配置】;
	2、需要访问宿主机上 Docker 引擎内部	11. 输入【主机路径】,如/var/lib/docker;
	数据结构的谷岙应用。 	12. 单击【确认】;
		13. 单击操作列的【确认】;
EmptyDir	在容器分配到节点时系统自动创建,初	14. 选择【卷类型】为"EmptyDir" ;
	始內容为空。在同一个 Pod 中所有容器 可以读写 EmptyDir 中的相同文件。当 Pod 从节点上移除时, EmpryDir 中的数	15. 输入【卷名称】,用户可自定义填写;



卷类型	卷类型解释	步骤说明
	据也会永久删除。通常用于临时数据的 高速存储。	16. 单击【添加配置】;
		17. 选择【存储介质】;
		默认:存储在硬盘上,适用于数据量大,读写效率 要求低的场景;
		内存:存储在内存中,适用于数据量少,读写效率 要求高的场景;
		18. 单击【确认】;
		19. 单击操作列的【确认】;
Secret	用来处理敏感数据,比如密码、Token	20. 选择【卷类型】为"Secret"。
	和密钥,相比于直接得敏感数据即直在 Pod 的定义或者镜像中,Secret 提供了	21. 输入【卷名称】,用户可自定义填写。
	更加安全的机制,防止数据泄露。应用 绑定服务时, 也需选择 Secret 中的"服	22. 单击【添加配置】:
	务访问凭证"进行绑定。	所有凭证:在配置文件下拉框中,选择已配置好的
	说明	Secret 又件。也可以单击【去可建】新建配直又件, 创建配置文件步骤请参见 <u>3.8 创建 Secret 资源</u> ;
	 若应用需绑定服务,需确保应用的 业务逻辑中支持绑定需要的服务。 	服务访问 凭证:为应用绑定平台中的服务。当前不 支持此功能。
	• 创建 Secret 类型资源时,	23. 单击【确认】;
	ServiceStage 会针对 Secret 信息 做加密处理,建议不要在 Secret 由检入进及人路利的信息	24. 单击操作列的【确认】;
0 6 1	中输入沙及个入隐松的信息。	25 选择【类类型】为"0-nf; man"
ContigMap	提供应用代码和配直又件的分离, ConfigMap 用于处理应用配置参数。用	25. 远拜【卷尖型】 / Configmap ;
	户需要提前创建应用配置,操作步骤请 参见 5_1 创建应用配置顶	26. 输入【卷名标】,用尸可目定义填写;
	多见 <u>5.1 的建应用配量数</u> 。	27. 单击【添加配置】;
		28. 在应用配置下拉框中选择需添加的配置文件;
		29. 输入配置文件所在的相对路径;
		30. 单击【确认】;
		31. 单击操作列的【确认】;

10. 描述: 输入对应定时任务的描述;

天	異こ						
ec	loud	ł					5 应用上线
	<u>。</u> 应	用 > 任务管理 > 普通任务 > 创建任务					
⑧ 资源管理	~	o		0	0	0	
88 应用管理		任务基本信息		普通任务配置	预览	完成	ĉ
☑ 任务管理	^	* 普通任务名称 nttask					
• 普通任务		卷定义					
· 定时任务		卷名称	老类型	825		操作	
08 配置坝		hostpathvol	HostPath ~	/var/lib/doc	ker	确认修改配置取消	
		() 添加					
		描述 测试普通任务					
							6/200
							下一步

- 11. 配置完成后,单击【下一步】,进入容器配置页面;
- 12. 设置容器名称,如 container1;
- 13. 单击【选择镜像】链接,选择对应的镜像。使用公有仓库或私有仓库,需确保镜像已上传至镜像仓库,具体方法请参见 7.2 镜像仓库
 - 公有仓库:显示 ServiceStage 所有用户上传的公有镜像;
 - 私有仓库:显示用户自己上传的公有和私有镜像。若所选镜像仓库为私有类型,选择镜像后,还需设置认证方式。此处认证方式的取值从 2.1.2 上传 AK/SK 证书处读取;
 - 第三方镜像:使用第三方镜像,如 dockerhub 提供的镜像。a、认证:用户可根据实际需求选择是否添加认证。若选择"是",请选择已创建好的认证。若没有创建,请单击【创建认证】创建"类型"为"Sercret"的资源,详细步骤请参见 3.8 创建 Secret 资源; b、镜像地址:请输入第三方镜像的仓库地址,如 https://hub.docker.com/_/wordpress/;

(日本) 应用	O	应用 > 任务管理 >	定时任务 > 创建定	时任务				
 资源管理 	~	_ ž	选择镜像	_	_	_	×	
88 应用管理			公有仓库和	有仓库 第三方镜像				
☑ 任务管理	^	容器	租户全部	~		请输入镜像仓	库名称	
•普通任务		cont	镜像仓库	約約 租户	命名空间 描述	版本	更新时间	
• 定时任务			• redis	chensl@chi	test	0907 ~	2017-09-07 16:06:48	
80 配置项		容) liny	chensl@chi	jaxrsserver	tag ~	2017-08-12 19:44:16	
		结代	 hostna 	ame chensl@chi	hostname	1.1 🗸	2017-08-09 14:36:37	
		BUC IS) hostna	ame chensl@chi	hostname	1.1 ~	2017-08-09 14:36:37	
			⊖ test	chensl@chi	hostname	1.0 ~	2017-08-07 17:10:49	
					5 ~ 总条数 8	(1 2	> 1 姚转	上一步

14. 设置容器内存限制: 启动容器时, 限制容器最多能够使用的内存量;



15. 设置容器 CPU 限制: 启动容器时,限制容器最多能够使用的 CPU 量;

- 16. 单击数据卷下方的【添加】链接,并设置数据库相关参数:
 - 容器路径:数据卷挂载到容器上的路径,请不要挂载在系统目录下,如"/"、"/var/run"
 等,会导致容器异常,建议挂载在空目录下;
 - 是否只读:配置为"是",只能读容器路径中的数据卷;配置为"否",可修改容器路径中的数据卷,容器迁移时新写入的数据不会随之迁移,会造成数据丢失;
 - 卷策略及日志策略:配置为 none,不支持卷策略;配置为 logs,当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs。当卷策略配置为 logs 时,可支持用户自定义日志收集 的路径,收集的日志在监控页面可以查看。可选策略包括 none(不进行日志回滚转储)、Hour ly (每小时扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一个历史文件中,在 日志所在目录下),并将原文件清空)、Daily(每天扫描一次日志文件,如文件超过 20M,会 将其转储,并将原文件清空)、Weekly(每周扫描一次日志文件,如文件超过 20M,会将其转储,并将原文件清空);
- 17. 配置完成后,单击【保存】按钮;
- 18. 若需要创建多个容器,请选择【添加容器】;
- 19. 单击【下一步】,进入预览页面;
- 20. 点击【创建】按钮,完成定时任务的创建;

心用	0	应用 > 任务管理 > 普通任务						
 资源管理 8 应用管理 	~	集群: ch1 + 任务创建 删除	✓ 命名空间: default	~		全部状态	✓ 満線入任务名称	Q []
☑ 任务管理	^	□ 任务名称 ~	状态 ~	执行完成时间	□ 开始执行时间 ~ □ 容器 个	数 創建时间 ~	操作	
・普通任务		pttask	运行中		2017/10/27 10:50:17 GMT+0 1	2017/10/27 10:50:1	7 GMT+0 删除	
·定时任务								
😳 配置项							10 V BRR 1	

5.3 创建应用

5.3.1 概述

ServiceStage 提供三种应用的创建方式,用户可根据实际需求选择创建。

1、创建容器应用(无状态):无状态应用中各实例之间相互独立,互不依赖,任意一个 Web 请求完全 与其他请求隔离。无状态容器应用更易实现可靠性和伸缩性。

2、创建容器应用(有状态):有状态应用中应用有多个实例,每个实例承担不一样的功能,且每个实





例间有依赖和前后启动顺序关系。有状态应用通常适用于有数据持久化存储要求的应用。

3、创建虚机应用:是指将应用进程直接部署在虚拟机上。

应用相关的概念如下:

术语	解释
应用	应用是可部署的软件实体,包含一个或一组容器或进程。
容器应用	以容器技术来运行构建应用的环境。一个容器应用可通过单个镜像或一个编排 模板创建,每个容器应用可包含 1 个或多个容器。
无状态容器应用	应用实例间互相不依赖,任意一个 Web 请求完全与其他请求隔离。无状态容器 应用更易实现可靠性以及可伸缩性。
有状态容器应用	应用有多个实例,且每个实例承担不一样的功能,且每个实例间有依赖和前后 启动顺序关系。通常用于有数据持久化存储要求的应用。
应用组	用户可以将某些相关应用放到同一个应用组中,方便统一管理。
标签	标签以 Key/value 键值对形式附加到各种对象上(如应用、节点、服务等)。 标签定义了这些对象的可识别属性,用来对它们进行管理和选择。
容器	一个通过 Docker 镜像部署的实例,一个节点可运行多个容器。

5.3.2 创建容器应用(无状态)

若用户需要托管以 docker 容器打包的应用,请创建容器应用。无状态应用中各实例之间相互独立, 互不依赖,任意一个 Web 请求完全与其他请求隔离。无状态容器应用更易实现可靠性和伸缩性。

前提条件:创建应用前,请确保系统中有可用节点,若没有请参照 2.1.7 为集群添加节点中内容创建; 容器应用需要基于镜像创建,若选择私有镜像,用户首先需要将镜像上传至镜像仓库,具体操作请参见 7.2.3 上传私有镜像(内网)。

创建步骤如下:

- 1. 登录天翼云 <u>http://www.ctyun.cn</u>;
- 2. 选择控制中心,切换到杭州节点;

3. 单击 ,选择 【企业中间件】 【微服务云应用平台】;

4. 单击【申请使用】;



5. 单击【应用】【应用管理】;

Service Sta	ge 🎧 概览	应用服务编排微服务运	医维 仓库 设置	
⑧ 资源管理 ∨	-	13 资源管理	☆ 任务管理	
😵 应用管理		集群	普通任务	
	711/6	节点	定时任务	
	顶克	其他	日本 配置项	
• 普通任务	~ 基本信	高 器 应用管理	☆	
・定时任务	普通任	务名称: testtest		
😳 配置项	卷定义			

- 6. 在【集群】下拉框中,选择任务所属的集群;
- 7. 单击【创建应用】;

一副应用	0	应用 > 应用管	11				
\bigcirc		作 理·	cin1 v	命名四间·	default	~	
🛞 资源管理	~	SHEWT.	Citiz	바라고드라	delaur	-	
🛚 应用管理							
🖸 任务管理	\sim						
😳 配置项							
							应用是可部署的软件实体,包含一个或者一组容器或者进程。 暫无应用,您可以 <mark>创建应用</mark>

8. 单击容器应用下的【创建】;

创建应用			×
	容器应用	虚机应用	
		•	
	以容器技术来运行构建应用的环境	将应用进程直接部署在虚拟机或物 理机操作系统上	
	创建	创建	

9. 选择【无状态应用】;



10. 设置应用基本信息:

参数	参数说明
所在集群	新建应用所在的集群。
应用名称	新建容器应用的名称,命名必须唯一。
应用组	您可以将某类应用放到同个应用组中,实现应用的批量管理(如 批量启停、删除)。用户输入应用组名称后,会自动创建一个新 的用户组。
描述	应用描述信息。
标签	标签以 Key/value 键值对形式附加到各种对象上 (如应用、节点、服务等)。标签定义了这些对象的可识别属性,用来对它们进行管理和选择。

- 11. 点击【展开高级设置】(可选);
- 12. 定义卷信息,步骤可参见 5.2.1 创建定时任务的步骤 10;
- 13. 自定义监控:配置完成后,可以在"运维 > 应用监控 > 指标监控 > 浏览指标"中查看到监控数据,若不需要监控,请将"监控类型"设置为"null"。如需设置监控,请设置下列参数:
 - 监控类型: prometheus, Prometheus 是一个开源的服务监控系统和时间序列数据库;
 - 上报路径:指标上报的 URL 路径,如/metrics;
 - 上报端口:指标上报端口,如 9121;
 - 监控维度:需要监控的指标名称列表,如
 ["redis_used_cpu_sys", "redis_used_cpu_user", "redis_memory_used_bytes", "redis_memory_used_peak_bytes", "redis_key_size", "redis_connected_clients"];
- 14. 升级策略:提供三种升级方式,替换升级、滚动升级、原地滚动升级,每种升级策略涵义如下:
 - 替换升级:先删除旧实例,再创建新实例。升级过程中业务会中断;
 - 滚动升级:先安装新实例,再移除旧实例。升级过程中,业务会同时均衡分布到新老实例上,
 因此业务不会中断;
 - 原地滚动升级:滚动升级的一个特殊场景,即在升级过程中,将旧实例更新为新实例,新实例和老实例所在节点保持一致;

设置具体的升级策略,参数如下表:

参数	参数说明



参数	参数说明
最大不可用个数或比 例(可选)	升级过程中允许的最大失效实例数值,可选配置,值可以是绝对值(5) 或者是比例(10%),默认为1。
最大超出实例数或比 例 (可选)	升级过程中允许超过指定实例数目的最大数值,可选配置,值可以是绝 对值(5)或者是比例(10%),默认为1。
实例可用最短时间(s)	升级时新创建的实例,达到就绪状态所需的最短时间,默认为 0。
历史版本上限(个)	最多保留的历史版本数量。

- 15. 删除策略:应用删除提供一个时间窗,若超过此时间窗,应用进程仍未停止,该应用将被强制删除;
- 调度策略:你可以根据需要自由组合静态的全局调度策略或动态的运行时调度策略来实现自己的需求。可设置应用间的亲和性和应用与节点间的亲和性:
 - 应用间的亲和性:应用间的亲和性——决定应用部署在相同或不同"主机"中;应用间的反 亲和性——"不同应用"或"相同应用的多个实例"部署在不同主机中。同个应用的多个实 例反亲和部署,减少宕机影响。互相干扰的应用反亲和部署,避免干扰;
 - 应用与节点间的亲和性:应用与节点亲和——决定应用部署在某些特定的主机中;应用与节 点反亲和——决定应用不能部署在某些特定的主机中;

亲和、反亲和	步骤说明
将应用设为亲和, 部署在相同主机	说明:在设置应用亲和/反亲和性时,您需要选择对应的拓扑域,拓扑域主要用 于限定应用的亲和性或反亲和在哪个范围内生效。
中 	kubernetes.io/hostname: 指同个节点内的亲和,即应用部署在同个节点上。
	创建应用间的亲和性有如下两种方式:
	方式一:选择应用
	1、单击【应用间的亲和性】【亲和】;
	2、单击【选择应用】,进入【选择应用(亲和)】页面;
	3、选择应用对应的拓扑域;
	4、选择待亲和的应用;
	5、设置完成后,单击【确认】按钮。系统会将您选择的应用设置为亲和,部署 到相同的节点中。
	方式二: 自定义
	说明: 自定义亲和中, 若亲和了系统中没有的应用或节点, 调度会失败。

• 调度策略配置如下:



亲和、反亲和	步骤说明
	1、单击【应用间的亲和性】【亲和】;
	2、单击【自定义】,进入【添加自定义(亲和)】页面;
	3、选择应用对应的拓扑域;
	4、单击【添加】;
	5、输入待亲和应用的键、值后,单击【确认】;
	6、单击【确认】,系统会根据您所添加的标签(键/值)选择应用,自动将该类 应用设置为亲和,部署到相同的节点中;
将应用设为反亲	设置应用间的反亲和性有如下两种方式:
和, 部署在个同主 机中	方式一:选择应用
	1、单击【应用间的亲和性】【反亲和】;
	2、单击【选择应用】,进入【选择应用(反亲和)】页面;
	3、选择应用对应的拓扑域;
	4、选择待反亲和的应用;
	5、设置完成后,单击【确认】,系统会将您选择的应用设置为反亲和,部署到 不同的节点中。
	方式二: 自定义
	1、单击【应用间的亲和性】【反亲和】;
	2、单击【自定义】,进入【添加自定义(反亲和)】页面
	3、选择应用对应的拓扑域;
	4、单击【添加】,输入待反亲和应用的键、值后,单击【确认】;
	5、单击【确认】,系统会根据您所添加的标签(键/值)选择应用,自动将该类 应用设置为反亲和,部署到不同的节点中;
将节点设为亲和	设置应用与节点间的亲和性有如下两种方式:
	方式一:选择节点属性
	1、单击【应用与节点间的亲和性】【亲和】;
	2、单击【选择节点属性】,进入【选择节点属性(亲和)】页面;
	3、选择待亲和节点的属性类型:
	操作系统:勾选某个操作系统,应用部署时会优选该操作系统; 版本:勾选某个版本,应用部署时会优选该版本;



亲和、反亲和	步骤说明
	类型架构: 勾选某个类型架构, 应用部署时会优选该类型架构;
	4、单击【确认】;
	方式二: 自定义
	1、单击【应用与节点间的亲和性】【亲和】;
	2、单击【自定义】;
	3、系统显示【添加自定义(亲和)】页面;
	4、单击【添加】;
	5、在"键"对应的下拉框中选择 key 后,单击【确认】;
	说明:
	此处下拉框中的取值来源于添加节点时设置的节点标签。 若多个节点,添加的标 签"键"相同,"值"不同,此时还需在"值"的下拉框中选择对应的 value;
	6、单击【确认】
将节点设为反亲	设置应用与节点间的反亲和性有如下两种方式:
和	方式一:选择节点属性
	1、单击【应用与节点间的亲和性】【反亲和】;
	2、单击【选择节点属性】,进入【选择节点属性(反亲和)】页面;
	3、选择待反亲和节点的属性:
	操作系统:勾选某个操作系统,应用部署时不会选择该操作系统;
	版本:勾选某个版本,应用部署时不会选择该版本;
	类型架构:勾选某个类型架构,应用部署时不会选择该类型架构;
	万式二: 目疋义
	3、单击【添加】;
	4、在【键】对应的下拉框中选择 key 后,单击【确认】;
	说明:
	此处下拉框中的取值来源于添加节点时设置的节点标签。若多个节点,添加的标签"键"相同,"值"不同,此时还需在"值"的下拉框中选择对应的 value;
	5、单击【确认】;



17. 单击【下一步】,设置应用配置信息;

参数	参数说明
实例数量	应用可以有一个或多个实例,用户可以设置具体实例个数,范围是 1-1000,直 接拖动 进行设置。
	说明:每个应用实例都由相同的容器部署而成。设置多个实例主要用于实现高可靠性,当 某个实例故障时,应用还能正常运行。
容器	容器是应用运行的真实环境。一个 Docker 容器应用可以由一个或多个容器组成。
	1、单击【选择镜像】;
	2、请根据需求,选择创建应用所需的镜像;
	说明:使用公有仓库或本租户仓库,需确保镜像已上传至镜像仓库,具体方法 请参见 7.2 镜像仓库。
	公 们 仓 库 : 显示 ServiceStage 所 有 用 户 上 传 的 公 有 镜 像 ;
	私有仓库:显示用户自己上传的公有和私有镜像;
	若所选镜像仓库为私有类型,选择镜像后,还需设置认证方式。此处认证方式 的取值从 2.1.2 上传 AK/SK 证书处读取。 第三方镜像:使用第三方镜像,如 dockerhub 提供的镜像;
	认证 :用户可根据实际需求选择是否添加认证。若选择【是】,请选择已创建 好的认证。若没有创建,请单击【创建认证】创建【类型】为【Sercret】的 资源,详细步骤请参见错误!未找到引用源。错误!未找到引用源。。 镜像地址 :请输入第三方镜像的仓库地址,如
	https://hub.docker.com/_/wordpress/
	3、单击【确认】;

18. 镜像上传完成后配置;

1)设置容器端口、内存限制、CPU 限制,参数如下表:

参数	说明			
容器	容器的名称,可修改;			
镜像	导入的镜像,单击 🖍 图标可进行修改;			
端口	将容器端口映射到主机端口上:			
	1、单击【添加】;			
	2、输入待映射的容器端口和协议;			
	3、配置完成后,单击【确认】;			
内存限制	启动容器时,分配和限制容器最多能够使用的内存量:			



参数	说明		
	• 分配:给应用分配的内存量;		
	 限制:应用能使用的内存量上限; 		
CPU 限制	启动容器时,分配和限制容器最多能够使用的 CPU 量:		
	● 分配:给应用分配的 CPU 量;		
	● 限制:应用能使用的 CPU 量上限;		

2)设置容器应用生命周期: ServiceStage 提供了回调函数,在容器的生命周期的特定阶段执行 调用,比如容器在停止前希望执行某项操作,就可以注册相应的钩子函数。目前提供的生命周期 回调函数如下所示:

参数	说明
命令行方式	在容器中执行指定的命令,配置为需要执行的命令。 如需要执行的命令如下:
	exec: command: - /bin/sh - c - /er-registy.sh registry 请在执行脚本中填写: /bin/sh -c "/er-registry.sh regisrty"。这条
HttpGet 请求方式	发起一个 HTTP 调用请求。配置参数如下: 路径:请求的 URL 路径,可选项。 端口:请求的端口,必选项。 主机地址:请求的 IP 地址,可选项,默认是容器所在的节点 IP。 协议:请求的协议,可选项,默认为 HTTP。

3)设置容器应用数据卷:数据卷用于实现容器持久化数据,在 Docker 设计实现中,容器中的数据是临时的,如果需要持久化数据,需要使用 Docker 数据卷挂载宿主机上的文件或者目录到容器中。通过挂载可以实现宿主机和容器之间,网络存储和容器之间,网络存储和宿主机之间的数据共享;

分类	说明
前提条件	挂载数据卷之前,用户需要提前添加卷类型。
设置容器应用数据卷操	1、在【数据卷】页签,单击【添加】;

5 应用上线



分类	说明
作步骤	2、设置数据卷对应的参数;
	卷名称:在下拉框中选择【应用基本信息】中创建的卷;
	容器路径:数据卷挂载到容器上的路径;
	注意:请不要挂载在系统目录下,如"/"、"/var/run"等,会导致容器异常。建议挂载在空目录下,若目录不为空,请确保目录下无影响容器启动的文件,否则文件会被替换,导致容器启动异常,应用创建失败;
	3、是否只读:
	• 配置为"是":只能读容器路径中的数据卷;
	 配置为"否":可修改容器路径中的数据卷,容器迁移时新写入的数据不会随之迁移,会造成数据丢失;
	4、卷策略及日志策略:
	● 配置为 none:不支持卷策略;
	配置为 logs: 当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs, 参数说明如下。当卷策略配置为 logs 时,可支持用户自定义日志收集的路径,收集的日志在监控页面可以查看,查看日志的详细操作请参见 6.3.2 查询日志。
	- None:不进行日志回滚转储
	 Hourly:每小时扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一个历史文件中,在日志所在目录下),并将原文件清空
	 Daily:每天扫描一次日志文件,如文件超过 20M,会将其转储(即 压缩到一个历史文件中,在日志所在目录下),并将原文件清空
	 Weekly:每周扫描一次日志文件,如文件超过 20M,会将其转储 (即压缩到一个历史文件中,在日志所在目录下),并将原文件清 空
	5、单击【确认】;

3) 设置容器应用环境变量,请参见11.7 如何设置容器应用环境变量?;

4) 设置容器应用健康检查, 请参见 11.8 如何进行容器应用健康检查?;

5) 配置完成后,单击【保存】;

19. 一个应用实例包含1个或多个相关容器。若您的应用包含多个容器,请选择【添加容器】;

20. 镜像设置完成后,单击【下一步】;

21. 将应用发布为可被访问的服务:

1)选择【否】,应用不需要被其他服务所访问(包括外网或内网访问),请直接单击"下一步";

 2)选择【是】,将应用发布为可访问的服务,请参考11.16如何配置服务发布参数?,配置服务 发布参数;

- 22. 配置完成后,单击【下一步】;
- 23. 确保信息配置无误后,单击【创建】;
- 24. 单击【应用详情】,进入应用详情页面;
- 25. 待应用状态为"运行中",应用创建成功(应用状态不会实时更新,请按F5查看);
- 26. 在应用详情页面,单击访问地址后的【IP:端口】,即可通过外网访问应用;

说明: 若应用需要被外网访问, 必须满足以下条件:

- 1)应用所在集群已有至少1个节点绑定 EIP,即"添加节点"中表 2-2"是否选择 EIP"参数选择 "是"。若没有绑定 EIP,可以参照如何绑定弹性 IP 绑定;
- 2)应用需要已发布为服务;

5.3.3 创建容器应用(有状态)

有状态应用中应用有多个实例,每个实例承担不一样的功能,且每个实例间有依赖和前后启动顺序关 系。有状态应用通常适用于有数据持久化存储要求的应用。

前提条件:

- 1、创建应用前,请确保系统中有可用节点;
- 2、创建多个容器应用时,请确保容器应用使用的端口不冲突,否则部署会失败;

创建步骤如下:

- 1. 登录天翼云 <u>http://www.ctyun.cn</u>;
- 2. 选择控制中心, 切换到杭州节点;
- 3. 单击 3, 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【应用】【应用管理】;
- 6. 在【集群】下拉框中,选择任务所属的集群;
- 7. 单击【创建应用】;
- 8. 单击容器应用下的【创建】;



9. 选择【有状态应用】;

□● 应用	0	应用管理 > 应用列表 > 创建育器公用				
🕲 资源管理	~	创建容器应用				
88 应用管理		o	0	0	0	0
☑ 任务管理	~	应用基本信息	应用配置	服务发布	预览	完成
😳 配置项						
		无状态应用有状态应用				
		基本信息				
		* hr在集群 cln1				~
		- 新属会名空间 default				~
		* 应用名称 以小尼字母开始、由小尼字母、教字、由別ば())組成 2				

10. 设置应用基本信息,其中带 "*" 标志的参数为必填参数;

参数	参数说明	
*所在集群	新建应用所在的集群。	
*应用名称	新建容器应用的名称,命名必须唯一。	
应用组	您可以将某类应用放到同个应用组中,实现应用的批量管理(如批量启停、删除)。	
描述	应用描述信息。	
标签	标签以 Key/value 键值对形式附加到各种对象上(如应用、节点、服务等)。	
	标签定义了这些对象的可识别属性,用来对它们进行管理和选择。	

11. 设置高级参数:

1)卷定义:若需要"数据存储"或实现"容器与主机间的文件共享"则需挂载卷。详细操作请参见
 11.1 如何配置容器应用卷定义?;

2) 自定义监控: ServiceStage 支持监控数据输出到 prometheus (Prometheus 是一个开源的服务监控 系统和时间序列数据库)。配置完成后,可以在"运维 > 应用监控 > 指标监控 > 浏览指标"中查看 到监控数据,详细步骤请参见操作请参见 11.2 如何配置自定义监控?。若不需要监控,请将"监控 类型"设置为"null";

3)删除策略:应用删除提供一个时间窗,若超过此时间窗,应用进程仍未停止,该应用将被强制删除。用户可输入延迟的时间,范围为[0-120]秒;



4) 调度策略:你可以根据需要自由组合静态的全局调度策略或动态的运行时调度策略来实现自己的 需求。详细操作请参见 11.4 如何设置调度策略?;

12. 单击【下一步】,设置应用配置信息:

1) 实例数量:请输入实例数量。每个应用实例都由相同的容器部署而成。设置多个实例主要用于实现高可靠性,当某个实例故障时,应用还能正常运行;

2) 服务名称:输入应用所对应的服务名称,用于实例间互相访问。例如一个应用有五个实例,此处服务名称为 service,五个实例的名称,系统会自动排序取名为: service0、service1、service2、 service3、service4;

3) 设置实例间发现服务,用于实例间的互相访问;

参数	参数说明		
添加端口	1、单击【添加端口】;		
	2、输入【端口名称】和【容器端口】,单击【确认】。端口名称用于给容 器端口命名,通常以端口用途命名;		

4)(可选)添加预加载容器:预加载容器中通常存放应用的配置初始化脚本等,为每个应用实例执行 初始化操作。如没有初始化操作,不需要执行此操作;

步骤	步骤说明
添加预加载容器	1、单击【添加预加载容器】,系统出现名为"initcontainer1"的按钮;
	2、单击【选择镜像】;
	3、选择对应的镜像;
	4、单击【确认】;
	5、添加容器【名称】;
	6、将容器端口映射到主机端口上;
	• 单击【添加】;
	• 输入待映射的容器端口和协议;
	• 配置完成后,单击【确认】;
	7、内存限制:启动容器时,分配和限制容器最多能够使用的内存量;
	• 分配:给应用分配的内存量;



步骤	步骤说明
	• 限制:应用能使用的内存量上限;
	8、CPU 限制:启动容器时,分配和限制容器最多能够使用的 CPU 量
	• 分配:给应用分配的 CPU 量;
	• 限制:应用能使用的 CPU 量上限;
	9、更多高级设置请参见 11.6 如何设置容器应用数据卷? 、11.7 如何
	设置容器应用环境变量?;
	10、单击【保存】

5) 添加应用容器镜像;

操作	说明	
*container1	1、单击【container1】;	
	2、单击【选择镜像】;	
	3、选择对应的镜像;	
	4、单击【确认】;	
	5、添加容器【名称】;	
	6、将容器端口映射到主机端口上;	
	 单击【添加】; 输入待映射的容器端口和协议; 配置完成后,单击【确认】; 	
	7、内存限制:启动容器时,分配和限制容器最多能够使用的内存量;	
	• 分配:给应用分配的内存量;	
	 限制:应用能使用的内存量上限; 8、CPU 限制:启动容器时,分配和限制容器最多能够使用的 CPU 量 分配:给应用分配的 CPU 量; 限制:应用能使用的 CPU 量上限; 9、更多高级设置请参见 11.5 如何设置容器应用生命周期?、11.6 如何设置容器应用数据卷?、11.7 如何设置容器应用环境变量?、11.8 如何进行容器应用健康检查?; 10、单击【保存】; 	

13. (可选)一个应用实例包含1个或多个相关容器。若您的应用包含多个容器,请选择【添加容器】;14. 镜像设置完成后,单击【下一步】;

15. 将应用发布为可被访问的服务:

1)选择【否】,应用不需要被其他服务所访问(包括外网或内网访问),请直接单击"下一步";

 2)选择【是】,将应用发布为可访问的服务,请参考 11.16 如何配置服务发布参数?,配置服务 发布参数;

- 16. 配置完成后,单击【下一步】;
- 17. 确保信息配置无误后,单击【创建】;
- 18. 单击【应用详情】,进入应用详情页面;
- 19. 待应用状态为"运行中",应用创建成功(应用状态不会实时更新,请按F5查看);
- 20. 在应用详情页面,单击访问地址后的【IP:端口】,即可通过外网访问应用;

说明: 若应用需要被外网访问, 必须满足以下条件:

- 1)应用所在集群已有至少1个节点绑定 EIP,即"添加节点"中表 2-2"是否选择 EIP"参数选择 "是"。若没有绑定 EIP,可以参照如何绑定弹性 IP 绑定;
- 2) 应用需要已发布为服务;

5.3.4 创建虚机应用

虚机应用是指将应用进程直接部署在虚拟机或物理机操作系统上。创建虚机应用时需遵循以下约束:

- 1、多个软件包部署在同一个节点上,需确保目录、端口等不冲突;
- 2、虚机应用部署到同个节点时,多个应用实例间要保证能同时运行在一个虚机上且没有冲突;
- 3、若虚机应用部署时设置了声明端口, ServiceStage 平台会基于端口执行冲突检查;

4、虚机应用部署为集群时,系统默认支持软性亲和,即同一个应用的不同实例会分散部署,但 不硬性保证;

创建步骤如下:

- 创建虚机应用前,您需要先将应用软件包上传至软件仓库。单击【仓库】【软件中心】,将软件包上 传至软件仓库,具体操作请参见 7.3 软件仓库;
- 2. 单击【应用】【应用管理】;
- 3. 在【集群】下拉框中,选择任务所属的集群;
- 4. 单击【创建应用】;



- 5. 单击虚机应用下的【创建】;
- 6. 设置应用基本信息,其中带"*"标志的参数为必填参数;

参数	参数说明
*所在集群	新建应用所在的集群。
*应用名称	新建容器应用的名称,命名必须唯一。
应用组	您可以将某类应用放到同个应用组中,实现应用的批量管理(如批量启停、删除)。
描述	应用描述信息。
标签	标签以 Key/value 键值对形式附加到各种对象上(如应用、节点、服务等)。
	标签定义了这些对象的可识别属性,用来对它们进行管理和选择。

7. 设置高级参数:

1)卷定义:若提前设置应用配置则需添加卷,详细操作请参见11.9 如何配置虚机应用卷定义?;

2) 升级策略: 支持替换升级、滚动升级、原地滚动升级, 详细操作请参见 11.3 如何设置升级策略?;

- 3)删除策略:应用删除提供一个时间窗,若超过此时间窗,应用进程仍未停止,该应用将被强制删除。用户可输入延迟的时间,范围为[0-120]秒;
- 4)调度策略:你可以根据需要自由组合静态的全局调度策略或动态的运行时调度策略来实现自己的 需求。详细操作请参见 11.4 如何设置调度策略?;
- 8. 单击【下一步】,设置应用配置信息:

参数	参数说明		
实例数量	应用可以有一个或多个实例,用户可以设置具体实例个数,范围是 1-1000。		
	说明:如果应用重要不能中断,建议创建2个及以上的实例,避免单应用实例故障时 业务中断。		
进程	1、在 process1 下,单击【选择软件包】,进入【选择软件包】页面;		
	2、在对应的软件仓库下,选择对应的软件包。若所选软件仓库为私有类型,还需设置 认证方式。此处认证方式的取值从 2.1.2 上传 AK/SK 证书处读取。		
	3、单击【确定】;		

9. 软件包上传完成后配置如下参数;



A. 181		
参数	参数说明	
进程	进程名称,可修改。	
软件包	已导入的软件包,可单击 🗸 进行修改。	
端口	您设置的进程端口将映射到主机端口上:	
	1、单击【添加】;	
	2、输入待映射的进程端口、协议;	
	3、配置完成后,单击【确认】;	
内存限制	启动进程时,限制虚机最多能够使用的内存量	
CPU 限制	启动进程时,限制虚机最多能够使用的 CPU 量。	

 10. 请参见 11.10 如何设置虚机应用生命周期?、11.11 如何设置虚机应用数据卷?、11.12 如何设置 虚机应用环境变量?、11.13 如何设置虚机应用健康检查?;

- 11. (可选)一个应用实例包含1个或多个相关进程。若您的应用包含多个进程,请选择【添加进程】;
- 12. 配置完成后,单击【保存】;
- 13. 软件包设置完成后,单击【下一步】;
- 14. 将应用发布为可被访问的服务:
 - 1)选择【否】,应用不需要被其他服务所访问(包括外网或内网访问),请直接单击"下一步";

 2)选择【是】,将应用发布为可访问的服务,请参考 11.16 如何配置服务发布参数?,配置服务 发布参数;

- 15. 配置完成后,单击【下一步】;
- 16. 确保信息配置无误后,单击【创建】;
- 17. 单击【应用详情】,进入应用详情页面;
- 18. 待应用状态为"运行中",应用创建成功(应用状态不会实时更新,请按F5查看);
- 19. 在应用详情页面,单击访问地址后的【IP:端口】,即可通过外网访问应用;

说明: 若应用需要被外网访问, 必须满足以下条件:

- 1)应用所在集群已有至少1个节点绑定 EIP,即"添加节点"中表 2-2"是否选择 EIP"参数选择 "是"。若没有绑定 EIP,可以参照如何绑定弹性 IP 绑定;
- 2)应用需要已发布为服务;



5.3.5 启停应用

停止运行中的应用,应用将无法访问,状态显示为"停止"。应用停止后,可直接将其启动。

- 1. 单击【应用】【应用管理】;
- 2. 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
- 3. 单击待停止应用后的【停止】,停止运行中的应用;
- 4. 单击待启动的应用后的【启动】,启动已停止的应用;

5.3.6 升级容器应用

- 1. 单击【应用】【应用管理】;
- 2. 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
- 3. 单击待升级应用后的【更多】【升级】;
- 4. 对应用配置进行更新,容器、端口、内存和 CPU 限制均不可修改,用户可上传对应镜像进行操作;
 - 1) 实例数量: 调整实例数量;
- 更多设置请参见11.5 如何设置容器应用生命周期?、11.7 如何设置容器应用环境变量?、11.8 如 何进行容器应用健康检查?其中,数据卷不可编辑;
- 6. 更新完成后,单击【保存】;

5.3.7 升级虚机应用

- 1. 单击【应用】【应用管理】;
- 2. 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
- 3. 单击待升级应用后的【更多】【升级】;
- 对应用配置进行更新,更新软件包:进程、端口、内存和 CPU 限制均不可修改,用户可上传对应软件包进行操作;
 - 1) 实例数量: 调整实例数量;
 - 2) 更新软件包:在进程下,在待升级的软件包后,单击 </ 图标进行修改;



- 更多设置请参见 11.5 如何设置容器应用生命周期?、11.7 如何设置容器应用环境变量?、11.8 如 何进行容器应用健康检查?其中,数据卷不可编辑;
- 6. 更新完成后,单击【保存】;

5.3.8 删除应用

- 1. 单击【应用】【应用管理】;
- 2. 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
- 3. 单击待升级应用后的【更多】【删除】;
- 4. 请仔细阅读系统提示,单击【确认】;

5.3.9 将应用移至应用组

将功能相似的应用或该应用相关的应用移至应用组,方便统一管理。

将某个应用移至应用组操作步骤如下:

- 1. 单击【应用】【应用管理】;
- 2. 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
- 3. 单击待升级应用后的【更多】【移至应用组】;
- 4. 在弹出的应用组窗口中,可选择已有用户组,或输入用户组名称新建用户组;
- 5. 单击【确认】;

将多个应用批量移至应用组操作步骤如下:

- 1. 单击【应用】【应用管理】;
- 2. 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
- 3. 勾选待移至同个应用组的应用;
- 4. 单击【移至应用组】;
- 5. 在弹出的应用组窗口中,可选择已有用户组,或输入用户组名称新建用户组;
- 6. 单击【确认】;



5.3.10 查看应用实例运行情况

- 1、查看应用实例 CPU 和内存状态:
 - 1) 单击【应用】【应用管理】;
 - 2) 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
 - 3) 单击待查看的"应用名称",进入应用详情页面;
 - 4) 在【实例】页签下,单击某个实例名称前的²;
 - 5) 在【监控】页签下,可查看到该实例的 CPU 和内存情况;
- 2、查看应用实例操作事件:
 - 1) 单击【应用】【应用管理】;
 - 2) 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
 - 3) 单击待查看的"应用名称",进入应用详情页面;
 - 4) 在【实例】页签下,单击某个实例名称前的 /;
 - 5) 在【事件】页签下,可查看到该实例的操作事件;
- 3、查看应用实例中容器运行状态及所在节点:
 - 1) 单击【应用】【应用管理】;
 - 2) 在【集群】下拉框中,选择任务所属的集群,可查看到当前集群下的所有应用;
 - 3) 单击待查看的"应用名称",进入应用详情页面;
 - 4) 在【实例】页签下,单击某个实例名称前的[>];
 - 5) 在【容器】页签下,可查看到该实例下所有容器的 ID、运行状态、运行节点等;

5.3.11 查看应用更多详情

e Cloud

应用创建成功后,在"应用详情"页面,您可以对应用进行管理,如下表:

操作	操作说明	
查看卷定义	单击【更多】【查看卷定义】,查看创建应用时定义的卷信息。	
查看升级策略	单击【更多】【查看升级策略】,查看创建应用时配置的升级策略。	
查看调度策略	单击【更多】【查看调度策略】,查看创建应用时配置的调度策略。	
查看应用服务	单击【服务】,查看服务详情。	
查看应用事件	单击【事件】,查看事件详情。	

5.4 创建任务类应用

5.4.1 概述

ServiceStage 中的编排主要用于对一组应用及应用运行时相关的元素进行管理。其提供的能力主要包括拓扑描述,组网设计,自动部署、升级卸载等。当应用模型确定之后,系统自动将应用的组件放置到合适的节点上,并配置相应的网络和存储来满足应用的要求。在使用编排前,建议您先了解以下基本概念。

概念	概念说明
编排	编排是指对组成堆栈的元素(Element)按照一定的次序进行处理的过程。通过定 义应用组件、对资源的需求、依赖的服务、元素之间的关系(拓扑)等,完整地描 述应用本身以及所依赖的服务、资源,最后交由 ServiceStage 平台完成部署。
设计包	运维人员对应用的拓扑、生命周期管理计划进行设计,并输出应用模板(也可称为 应用设计包)。
设计器	ServiceStage 还提供了图形化设计器工具,用户可通过拖拽方式完成复杂应用的 编排及拓扑设计,可以保存成模板,使用模板创建堆栈,简化应用部署难度,提升 效率。
模板	模板是对堆栈的描述,包括基于应用模型的堆栈拓扑定义、堆栈生命周期描述、运 行时资源描述、软件组件描述等。模板通过设计包创建而来,本质与设计包相同。
堆栈	由应用、服务、资源等元素组成的一个部署实例,平台将相关编排元素通过"堆栈" 进行集中管理。

编排、设计包、模板、堆栈、设计器的相互关系如下图:





5.4.2 操作流程

用户可选择两种方式来创建堆栈,具体流程如下图所示。建议采用通过模板来创建堆栈。



- 1. 创建堆栈前准备:方法一和方法二的操作都是相同的;
 - 1) 上传 AK/SK 证书、创建 VPC、创建集群等;
 - 2) 上传堆栈所需的镜像或软件包到仓库;
- 2. 用户可以根据需求选择其中一个方法来创建堆栈;
 - 1) 方法一: 通过模板创建堆栈, 用户需要准备设计包来创建模板, 再创建堆栈;

 方法二:通过设计器创建堆栈,通过设计器,用户可通过拖拽方式完成应用编排拓扑设计, 简化部署难度;



5.4.3 创建堆栈前准备

- 上传应用所需的镜像到镜像仓库,单击【仓库】【镜像仓库】,将创建容器应用所需的镜像上传至镜 像仓库,具体步骤请参见 7.2 镜像仓库;
- 上传应用所需的软件包到软件仓库,单击【仓库】【软件仓库】,将软件包上传至软件仓库,具体步骤请参见 7.3 软件仓库;

5.4.4 通过模板创建堆栈

开发人员或运维人员对应用的拓扑、生命周期管理计划进行设计,并输出应用模板(也可称为应用设 计包),系统根据应用模板自动创建堆栈。

5.4.4.1 创建模板

前提条件:

创建模板所需的设计包已准备完成,且满足 12.4 应用编排设计包说明中的规范要求。

操作步骤:

- 1. 登录天翼云 <u>http://www.ctyun.cn</u>;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 📴 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【编排】【模板】;



在右侧页面中,单击【创建模板】;

天翼云					
eCloud				5 应用上线	;
Service Stage	☆ 概览 应用 服务 编排 微服务 运维 仓库 设置	ρ	鄙 杭州	≗ ivycln@126.com@…	
(品编排) •	编辑 > 模板				(
堆栈					
☐ 模板					
📄 设计器					
	模板是对堆钱的定义与描述,包括基于应用模型的堆钱石 <u>1、堆栳土金属</u> 期、运行时资源、软件组件等。 暂无堆栈模板,您可 <mark>1、创建模板</mark>				

7. 上传设计包,用于创建模板。上传设计包有两种方式,您可以根据需求选择:

1) 方式一: 单击【本地上传】, 上传创建模板所需的设计包;

2)方式二:单击【从软件仓库上传】,选择部署模板所需的设计包。若选择"方式二",用户需要提前上传设计包到软件仓库,步骤请参见 7.3 软件仓库;

8. 配置模板信息,其中带 "*" 标志的参数为必填参数:

参数	参数说明
*模板名称	新建模板的名称,命名必须唯一。
*版本	设计包的版本信息。
发布者	模板发布者。
描述	模板描述信息。

9. 配置完成后,单击【创建】。系统提示模板创建成功,且在模板列表中可以查看到已创建的模板;

后续处理:

模板创建完成后,您还可以执行下表中的操作:

操作	说明	
部署模板	单击 ອ, 使用已创建的模板部署堆栈, 详情请参见 5.4.4.2 创建 堆栈。	
更新模板	单击 个,更新模板信息,具体操作请参见 5.4.4.1 创建模板。	
查看拓扑图	单击。。 查看该模板的拓扑图及各组件间的关联关系。	
	说明:在查看拓扑图页面,单击左上角的【编辑】,可对拓扑图进 行编辑。此处编辑拓扑图的操作,其本质就是将模板导入设计器,	



操作	说明
	在设计器中保存一份。
删除模板	1、单击 , 系统提示 "确定删除该模板?"; 说明:系统默认在删除模板时直接删除设计包,若不希望删除设计包,请取消
	对"删除设计包"的选中。
	2、单击【确定】;

5.4.4.2 创建堆栈

- 1. 在 servicestage 页面, 单击【编排】【堆栈】;
- 2. 在右侧页面中,单击【创建堆栈】;
- 在【当前模板】中,选择用于创建堆栈所需的模板。此处可以选择系统中已存在的模板来部署堆栈, 也可创建新模板来部署堆栈;
- 4. 参照下表配置堆栈相关参数,其中带"*"标志的参数为必填参数:

参数	参数说明
*堆栈名称	新建堆栈的名称,命名必须唯一。
描述	堆栈描述信息。

- 5. 配置完成后,单击【下一步】;
- 6. 参照下表配置相关参数:

参数	参数说明
输入参数	此处展示的参数是由创建模板时导入设计包中的 yaml 文件定义的。不同的 设计包,定义的 yaml 文件不同,此处展示的参数也不同。
	说明
	 若此处输入的是虚机软件包,需确保该软件包已上传到软件仓库。
	 若此处输入的是容器镜像包,需确保该镜像包已上传到容器镜像仓库。
	 信息配置完成后,您可以单击"导出配置文件"导出已配置好的信息。下次创建 类似堆栈时可直接导入使用。导出文件名称为"堆栈名称 config. json"。
	● 镜像参数值一般为镜像的仓库地址:版本号,若不输入版本号,默认为 latest。
	例如参数值: "10.175.125.135:20202/cfe-test/guestbook-redis:v2"。
高级设置	



参数	参数说明
超时时间	设置任务最大执行时间,超过时间后任务会自动停止并判定为错误。
自动操作	若勾选,系统会在堆栈安装失败后自动回退到初始化状态。

- 7. 配置完成后,单击【下一步】;
- 确认信息配置无误后,单击【创建】。系统提示堆栈创建成功,且在堆栈列表中可查看到已创建的 堆栈。当堆栈状态为"运行中"时,表示堆栈已部署完成;

5.4.5 通过设计器创建堆栈

5.4.5.1 概述

图形化编排工具可以通过拖拽的方式完成复杂应用的编排及拓扑设计,并且可以保存成应用模板,使 用模板可以直接创建多个容器或虚机组成的复杂应用,大大简化应用部署难度,提升生产力。设计器仅支 持内置节点类型,其他自定义节点类型暂不支持。

平台提供符合 TOSCA 标准的图形化编排工具。使用图形化设计器前,建议您先了解 TOSCA 模板的相关 信息,请参见什么是 TOSCA 模板。

本章以 guestbook 应用为例,展示如何通过设计器创建堆栈。Guestbook 应用由 frontend、 redis-master 和 redis-slave 三个组件组成,共计有3个镜像。若采用独立部署的话,需要规划3主机; 若在1个节点部署只需规划1台主机。本例以独立部署为例进行说明。分别需要拖拽3个应用组件、软件 组件和资源组件。

5.4.5.2 设计器界面介绍

- 1. 登录天翼云 <u>http://www.ctyun.cn;</u>
- 2. 选择控制中心, 切换到杭州节点;
- 3. 单击 , 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【编排】【设计器】;





6. 设计器界面说明如下表:

编号(对 应上图)	解释
1	设计器名称,双击可重命名,也可在下拉框中选择其他模板进行操作。
2	设计器的快捷操作图标。
	● 单击 🎦 ,最大化显示设计器界面。
	● 単击 <mark>□</mark>
	- 新建空白模板:新建空白模板来创建堆栈,目前最多仅支持创建 10 个。
	 选择示例模板:使用系统内置的示例模板来创建堆栈。通过示例模板使用户能快速 了解设计器的使用方法,详情请参见 11.15 如何通过设计器示例模板创建堆栈?。
	• 单击 🖺 ,保存应用模板。
	• 单击📩,下载由设计器编排而来的设计包,用于创建模板。
	• 单击 🗹, 校验设计器编排的模板是否符合 TOSCA 标准。
	● 单击 🕝, 重命名模板名称。
	● 单击 →, 部署堆栈。
	● 单击 💼 , 删除当前的模板。
	• 单击 2 ,复制当前模板,并另存为一个新模板。
	●单击 →, 导入设计包或模板,设计包支持"tar.gz、zip、tgz"三种格式,模板支持 "yaml"文本格式。




编号(对 应上图)	解释
3	●编排拓扑:显示图形化的界面。
	 输入输出:定义拓扑的输入输出参数。
	 源代码:通过图形化拖拽,自动生成源代码。
4	各组件的图标,用于拖拽到右侧面板,并设置节点间的关联关系。
	● 应用
	 【应用】【应用组】:应用组,由一个或多个应用组成,可整体进行生命周期操作,如 部署、升级。应用组可以对应到客户产品、业务系统/子系统等发布概念。
	- 【应用】【应用】: 运行在资源上的云应用,是对最小可部署对象的一种描述。 MicroService 是一种 Application。
	 【应用】【任务】:是一种一执行完就退出的应用。任务类应用继承了普通应用的属性, 并在其基础上添加了一些属性和限制。
	- 【应用】【代理】: 代理类型应用类似于 K8S 里 deamonSet 资源应用。该应用继承了普通应用的属性,并在其基础上添加了一些限制。
	- 【软件组件】【软件】: 云应用组件的组成部分,即软件包。也可以作为应用的属性, 是可选节点。
	●资源
	- 【存储】【本地卷】: 应用可以绑定对应的卷。
	 【资源模板】【集群】:具有相同属性的节点组成的资源模板。同一个集群中的节点具有相同的类型、镜像、CPU、内存、磁盘、认证信息等配置,并共享卷,处于相同的安全组、子网等。
	●群组
	解决方案:对应于完整的解决方案。
5	设计器编排拓扑的界面,可拖拽左侧组件到此界面,并设置组件间的关联关系,进行模 板设计。

5.4.5.3 部署应用组件

前提条件:

1、已创建资源,包括集群、节点;

2、已将部署 Guestbook 应用所需的镜像上传到容器镜像仓库,并已获取镜像的仓库地址。Guestbook 主要包括3个组件的镜像,上传镜像步骤请参见7.2 镜像仓库。本例中,镜像名称分别为:frontend.tar、 redis-master.tar、redis-slave.tar;

操作步骤:



- 1. 在 servicestage 页面,单击【编排】【设计器】;
- 2. 单击,选择新建空白模板;
- 3. 双击应用模板名称,将模板命名为 guestbook-test,单击【重命名】。
- 4. 拖拽三个应用组件到设计区域;



5. 在 "application-1" 上单击右键选择【编辑】,或双击【application-1】。将应用组件名称设置为
 "frontend-app", "类型"选择 "container",其他参数均采用默认:

frontend-app
立田
童 删除
可伸缩 🕜
true
有状态 2
false
可启停 🕜
true
送型 ⑦
container ជាក
process
container
true

6. 按照上述方法,设置 application-2、application-3 属性。将 "application-2" 命名为 "redismaster-app", "application-3" 命名为 "redisslave-app", "类型"均设置为 "container";

天翼云 e Cloud

		frontend-app	
	redis	master-app	
	redisslave-a	4pp	
_			



5.4.5.4 部署软件组件

软件组件用于定义应用所需要的软件(docker 镜像/进程类型软件包)等信息。

操作步骤:

1. 拖拽3个软件组件到设计区;



- 2. 在【swcomponent-4】上单击右键选择【编辑】:
 - 1) 将软件包组件名称修改为"frontend-software";
 - 2) 单击【软件包】参数后的 🗹,设置软件包参数,设置为如下内容,其他参数均采用默认值:

A、设置 frontend 对应的镜像仓库信息。单击 image 后的参数后的 ²¹,在输入参数名的下拉 框中选择【frontend_image】,单击【确定】;

```
package :

▼ container_spec : ①

▶ command : ⊕

image : {"get_input":"frontend_imaç []]
```



B、设置 package_type 参数,将软件包类型定义为容器类型 container;

package_type :	container	
----------------	-----------	--

C、设置 frontend-app 的容器监听端口,将 containerPort 参数值设置为 80;

package :	
▷ container_spec : ①	
deploy : 🕐 true 🗸 🎽	
▶ env : ⊕ 🗊	
ingress_spec :	
package_pull_secret :	
package_type : container	
▼ ports : ⊕ 🗊	
— containerPort : 80	

D、配置 service_spec 参数,将 frontend-app 发布为可被外网访问的服务;

 servi 	ce_spec: 🗊	
▼ po	orts : 🕀 🗊	
—	name :	ß
	nodePort :	Ø
	port : 80	E
	protocol : TCP	Ø
	targetPort :	Ľ
type	: NodePort	Ø

E、单击【确定】;

- 3. 设置 "swcomponent-5" 、 "swcomponent-6" 属性:
 - 1)将"swcomponent-5"命名为"redismaster-software"、"swcomponent-6"命名为 "redisslave-software";



2) 单击【软件包】参数后的 2,设置软件包参数,设置为如下内容,其他参数均采用默认值;

A、设置"redismaster-software"和"redisslave-software"对应的镜像仓库信息。单击 image 后的参数后的 , 在输入参数名的下拉框中选择"redismaster_image"和 "redisslave_image" , 单击【确定】;

B、参照步骤 2.2.b-步骤 2.2.e,分别配置 redismaster-software 和 redisslave-software, 配置内容和 frontend-software 相同;



5.4.5.5 部署资源组件

资源组件(本版本中代表集群)是具有相同属性的节点组成的资源模板。同一个集群中的节点具有相同的类型、镜像、CPU、内存、磁盘、认证信息等配置,并共享卷,处于相同的安全组、子网等。

操作步骤:

1. 拖拽三个资源组件(Cluster)到设计区,用于运行 guestbook 的 3 个应用组件;





- 在 "cluster-26" 上单击右键选择【编辑】。本例中,将资源组件名称修改为 "frontend-host"。
 在【属性】【名称】下拉框中,选择该资源组件所属的集群,其他参数采用默认值;
- 按照上述方法,将 "cluster-27"命名为 "redismaster-host"、 "cluster-28"命名为 "redisslave-host"。 资源组件 "redismaster-host"和 "redisslave-host"可以和 "frontend-host"在同一个集群,也可以在不同集群;





5.4.5.6 创建堆栈

- 1. 创建组件间的关联关系:在控件的 图标上,按住鼠标左键,拖拽到待关联控件的 图标上,并 释放鼠标。连线关系解释如下:
 - 1) "frontend-app" 连接"frontend-software", 再连接"frontend-host";
 - 2) "redisslave-app" 连接 "redisslave-software", 再连接 "redisslave-host";
 - 3) "redismaster-app" 连接 "redismaster-software", 再连接 "redismaster-host";
 - 4) "frontend-app" 连接"redisslave-app";
 - 5) "redisslave-app" 连接 "redismaster-app";



- 2. 设置输入输出参数:
 - 1) 单击图形化设计器右上侧的【输入输出】;



2)添加输入参数;

输入参数名称	类型	描述	默认值
frontend_image	字符串	(可选)输入对 参数的描述。	镜像的仓库地址:版本号,例如: 10.175.11.161:20202/apptest/guestbook-fro ntend:v1
redismaster_image	字符串		镜像的仓库地址:版本号,例如: 10.175.11.161:20202/apptest/redismaster:v 1
redisslave_image	字符串		镜像的仓库地址:版本号,例如: 120.109.220.72:20202/aos-team/redisslave: latest

- 3)参数设置完成后,单击【保存】;
- 3. 设置完成后,单击 🖺 图标,保存应用模板;
- 4. 单击 ☑,校验 Guestbook 模板是否符合 TOSCA 规范。界面中显示"校验成功",表示模板检验成功,符合 TOSCA 规范;
- 5. 您可以选择两种方式创建堆栈:
 - 1) 先创建模板,再根据模板创建堆栈;
 - A、单击🗻,下载设计包;
 - B、创建模板后再创建堆栈,具体步骤请参见 5.4.4.2 创建堆栈;
 - 2) 单击→,直接创建堆栈;
 - A、参照下表配置堆栈信息:

参数	参数说明	
	模板信息	
说明:使用设	计器创建堆栈时,该部分参数设计器会自动配置,用户可根据需求进行修改。	
*模板名称	模板名称。	



参数	参数说明
*版本	模板版本信息。
发布者	模板创建者。
描述	模板描述信息。
	堆栈信息
*堆栈名称	堆栈的名称,命名必须唯一。配置为: guestbook
描述	堆栈描述信息。
输入参数	此处展示的输入参数为 <u>步骤 2</u> 中设置的输入参数。您可以对输入参数的取值进行修改。
超时时间	设置任务最大执行时间,超过时间后任务会自动停止并判定为错误。
自动操作	若勾选,系统会在堆栈安装失败后自动回退到初始化状态。

B、配置完成后,单击【创建】;

6. 查看堆栈是否部署完成;

大 更 Cloud

- 7. 在堆栈页面,单击已创建堆栈的名称;
- 8. 单击【执行日志】,确认创建的状态显示为【成功】;

5.4.6 查看堆栈详情

堆栈安装成功后 (状态为运行中),可以查看堆栈的组成元素、参数配置、执行日志和拓扑图

- 1. 单击【编排】【堆栈】;
- 2. 单击待查看的堆栈,进入堆栈详情页面。可查看堆栈详情,请参见下表:

堆栈详情	步骤说明
查看堆栈组成	1、在【堆栈元素】页签下,可查看到该堆栈的组成元素,如应用、资源。
元素	2、在应用下,可查看到堆栈包含的应用名称以及实例情况;
查看参数配置	在【参数配置】页签下,可查看该堆栈的输入输出参数;
查看执行日志	1、在【执行日志】页签下,可查看当前应用的生命周期、状态;
	2、单击【查看】,可查看操作日志;



堆栈详情	步骤说明
查看拓扑图	1、在【拓扑图】页签下,可查看到当前堆栈的拓扑图,可视化呈现应用关联 关系;
	2、单击【编辑】,系统进入设计器页面。在该页面您可以对模板进行编辑;
	说明
	 设计器仅支持内置节点类型,若使用自定义节点,查看拓扑图会失败。
	 通过设计器创建的堆栈,编辑拓扑图后,需要重新创建堆栈。
	 通过模板创建的堆栈,对其进行编辑拓扑图的操作,其本质就是将模板导入设计器, 在设计器中保存一份。

5.4.7 升级堆栈

堆栈安装成功后(状态为运行中),可以对堆栈进行升级。目前支持如下两种方式升级堆栈:

- 1、配置参数升级: 仅升级参数, 不升级软件包或镜像;
- 2、软件包镜像升级:升级软件包或镜像;

5.4.7.1 通过配置参数升级

- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 · / / 升级];
- 3. 【升级方式】选择【配置参数升级】;
- 4. 参数下表**错误!未找到引用源。**配置升级参数。采用配置参数升级堆栈时,不支持设置堆栈升级失 后自动进行回退的操作:

参数	参数说明
输入参数	此处展示的参数是由创建模板时导入设计包中的 yaml 文件定义的。不同的设计包,定义的 yaml 文件不同,此处展示的参数也不同。
	说明
	• 若此处输入的是虚机软件包,需确保该软件包已上传到软件仓库。
	 若此处输入的是容器镜像包,需确保该镜像包已上传到容器镜像仓库。
	 信息配置完成后,您可以单击"导出配置文件"导出已配置好的信息。下次创建 类似堆栈时可直接导入使用。导出文件名称为"<i>堆栈名称</i> config. json"。



参数	参数说明
	 镜像参数值一般为镜像的仓库地址:版本号,若不输入版本号,默认为 latest。 例如参数值: "10.175.125.135:20202/cfe-test/guestbook-redis:v2"。
高级设置	
超时时间	设置任务最大执行时间,超过时间后任务会自动停止并判定为错误。
自动操作	若勾选,系统会在堆栈安装失败后自动回退到初始化状态。

5. 配置完成后,单击【下一步】;

6. 确认参数配置无误后,单击【升级】

5.4.7.2 通过软件包镜像升级

- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 💛 >【升级】:
- 3. 【升级方式】选择【软件包镜像升级】;
- 4. 选中待升级的应用组后,单击 🕀;
- 5. 选择升级包:
 - 1) 容器应用时此处展示的是"选择镜像包";
 - 2) 虚机应用时此处展示的是"选择软件包";
- 6. 超时设置:设置升级任务的最大执行时间,超过时间后任务会自动停止并判定为错误;
- 7. 设置自动操作: 若勾选, 系统会在堆栈安装失败后自动回退到初始化状态;
- 8. 配置完成后,单击【下一步】;
- 9. 核对信息无误后,单击【下一步】;
- 10. 单击【完成】;

5.4.8 伸缩堆栈

堆栈安装成功后(状态为运行中),可以根据自身需求调整堆栈中的应用实例数。

说明:当前不支持资源的伸缩。



- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 · / (伸缩);
- 3. 选中待伸缩的应用组后,单击 🖽 图标;
- 4. 勾选需要伸缩的应用组件,调整应用需要调整的实例数;
- 5. (可选)超时设置:设置伸缩任务的最大执行时间,超过时间后任务会自动停止并判定为错误;
- 6. 配置完成后,单击【确定】,开始对堆栈进行伸缩处理;

5.4.9 安装堆栈

针对未安装或安装失败的堆栈,支持在界面中直接安装。

- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 😳 > 【安装】;
- 3. 参照下表配置安装相关参数,若安装失败再重新安装时,模板和堆栈名称不可修改;

参数	参数说明
输入参数	此处展示的参数是由创建模板时导入设计包中的 yaml 文件定义的。不同的设计包, 定义的 yaml 文件不同,此处展示的参数也不同。
	说明:
	• 若此处输入的是虚机软件包,需确保该软件包已上传到软件仓库。
	 若此处输入的是容器镜像包,需确保该镜像包已上传到容器镜像仓库。
	 信息配置完成后,您可以单击"导出配置文件"导出已配置好的信息。下次创建类似堆栈时可直接导入使用。导出文件名称为"<i>堆栈名称</i> config.json"。对于安装失败的堆栈,在界面再次执行安装时不提供"导出配置文件"功能。
	• 镜像参数值一般为镜像的仓库地址:版本号,若不输入版本号,默认为 latest。
	例如参数值: "10.175.125.135:20202/cfe-test/guestbook-redis:v2"。
高级设置	
超时时间	设置任务最大执行时间,超过时间后任务会自动停止并判定为错误。

- 4. 单击【下一步】;
- 5. 确认参数配置无误后,单击【完成】;



6. 堆栈开始安装,等待堆栈安装完成,待堆栈状态为"运行中"时表示安装成功;

5.4.10 回退堆栈

针对已升级过的堆栈,支持回退到前一个版本。回退堆栈有如下两种方式:

1、软件镜像包回退:只回退软件包或镜像包;

2、整体回退:回退所有内容,如软件包、镜像包、参数等;

5.4.10.1 软件包镜像回退

- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 🔛 > 【回退】;
- 3. 将【回退方式】设置为【软件镜像包回退】;
- 4. 超时设置:设置回退任务的最大执行时间,超过时间后任务会自动停止并判定为错误;
- 5. 单击【确认回退】;

5.4.10.2 整体回退

- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 💛 > 【回退】;
- 3. 将【回退方式】设置为【整体回退】;
- 4. 超时设置:设置回退任务的最大执行时间,超过时间后任务会自动停止并判定为错误;
- 5. 单击【确认回退】;

5.4.11 卸载堆栈

每个堆栈中会包含一个或多个应用组,每个应用组中又包含一个或多个应用,每个应用下又包含一个 或多个组件或实例,此处的卸载就是卸载应用下的组件或实例。当前只支持整体卸载堆栈。

- 1. 单击【编排】【堆栈】;
- 2. 在待升级的堆栈上,单击 > 【卸载】;



- 3. 勾选需要卸载的应用组(当前只支持整体卸载);
- 4. (可选)配置超时时间:设置任务最大执行时间,超过时间后任务会自动停止并判定为错误;
- 5. 配置完成后,单击【确定】,开始卸载;

5.4.12 删除堆栈

若执行删除堆栈,系统将执行卸载和删除,删除后堆栈不能恢复,若只是卸载并保留堆栈,请选择卸 载操作。

- 1. 单击【编排】【堆栈】;
- 在待升级的堆栈上,单击 > 【删除】。系统将执行卸载和删除,删除后堆栈不能恢复。若希望 保留堆栈,请选择卸载操作;
- 3. 单击【确定】;



6 应用运维

6.1 应用监控

6.1.1 概念

1、什么是应用监控?

应用监控(Application Monitor Service, 简称 AMS)是一项针对资源和应用的监控服务,通过应用监控 您可以及时了解应用的资源使用情况、趋势和告警,使用这些信息,您可以快速响应,保证应用流畅运行。

2、什么是命名空间?

命名空间是指标的容器,不同命名空间中的指标彼此独立,因此来自不同应用程序的指标不会被错误地聚 合到相同的统计信息中。

3、什么是维度?

维度是指标的分类。每个指标都有各自不同的特征,维度即指标特征的类别。

4、什么是指标?

指标是对资源性能的数据描述或状态描述。

6.1.2 监控应用运行状态

通过应用卡片列表,对每个应用的类型、CPU占用、内存占用和告警状态进行展示,可以直观发现每 个应用的运行健康状态。打开每个应用卡片,展示了应用实例列表、告警、节点状态和接口调用情况,可 以立体监控应用的运行状态。

提供应用下钻到应用实例,并对应用实例进行监控。提供应用实例下钻到容器,并对容器进行监控。

1. 登录天翼云 http://www.ctyun.cn;

2. 选择控制中心,切换到杭州节点;



- 3. 单击 🔡 , 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【运维】【应用监控】;
- 6. 在左侧导航栏中选择【应用】;

运维 •	运维 > 应用监控 > 应用									
□ 总览 ■ 监控 ^	应用监控 全部 正常	异常						全部	~	应用名称
· Dashboard	tes	st		运行中		dsfafdsf			t	estelb
·应用	一 如	搂型:无状态。	立用 250/241)			应用类型:无状态				2月类型:无状
·节点	07	1 (19,88/±30) (1997)			1/1 (1982/±)	ap 34 199])		2	/ Z (19088/3
·指标监控	CPU占用:2	20%	内存占用:1	.9%	CPUŁ	:用:	内存占用:		CPU占用	
·阈值规则								-		
	0	0	0	0	0	0	0	0	0	0
	紧急	重要	次要	提示	紧急	重要	次要	提示	「「「」」「「」」」」	重要
	ree	dis1		运行中		server		运行中	t	est

7. 应用卡片参数说明请参见下表;

参数	说明
应用名称	系统所监控的某一个应用的名称,即应用卡片左上角内容。
应用状态	系统所监控的某一个应用的运行状态,即应用卡片右上角内容。
	应用状态包含"运行中"、"异常"、"未知"、"未就绪"、"升级/回滚中"、 "停止"和"删除中"。
应用类型	应用类型包含"有状态应用"、"无状态应用"和"守护应用"。
问题实例/全部	系统所监控的某一个应用的问题实例(问题实例指除了状态为"运行中"外的所有 实例)个数和总实例个数。
CPU 占用	系统所监控的某一个应用运行时的 CPU 占比, 单位为%
内存占用	系统所监控的某一个应用在运行时所有容器(或所有进程)占用的物理内存之和。 单位是 M、G
告警	应用指标对应的超限阈值的告警级别。告警级别包含"紧急"、"重要"、"次要" 和"提示"。
	说明: 当统计数据的值不为0时, 单击统计数据的值, 系统跳转到"告警查询"页面, 可以进一



参数	说明
	步查看、处理该应用的告警。

8. 在【应用监控】页面右上角的下拉列表框中选择应用类型后,再通过应用名称搜索要查看的应用;

9. 单击应用名称,进入【应用监控详情】页面;

说明: 在【应用监控详情】页面右上角的下拉列表框中选择不同的统计周期,可以查看该应用在不同 时间段的应用监控详情; 单击"指标趋势"右侧的【添加指标图表】,可以自定义指标图表在"指标 趋势"下的展示,方便实时监控关注的指标和查看其趋势;

1) 在【实例】列表中,查看该应用所有实例的概况,实例的参数说明请参见:

参数	说明
实例名称	该应用的某一个实例的名称。
	说明:
	 在【应用监控详情】页面中单击实例名称,可以下钻到【实例监控详情】 页面,并对实例进行监控。
	 在【实例监控详情】页面中单击容器名称,可以下钻到【容器监控详情】 页面,并对容器进行监控。
状态	该应用的某一个实例的运行状态。
IP 地址	该应用的某一个实例所在节点的 IP 地址。
	说明:
	单击某一个实例的 IP 地址,系统跳转到【节点监控详情】页面,可以进一步 查看该实例所在节点的运行状态,详细操作请参考 6.1.3 监控节点运行状态。
创建时间	该应用的某一个实例的创建时间。

2) 在【指标趋势】下面,查看该应用的指标趋势;

A、在指标图表的【更多】下拉列表框中选择【添加阈值】,可以对该应用指标创建阈值规则;

B、在指标图表的【更多】下拉列表框中选择【指标浏览】,系统跳转到【已选指标】页面,

可以通过设置统计周期、统计方式和时间范围,从不同维度查看该应用指标的指标图表。

3) 在页面右侧的【告警】区域框下面,查看该应用在不同级别下告警的统计数据;



4) 在页面右侧的【节点】区域框下面,查看该应用所有实例所在节点的状态统计数据;

5)在页面右侧的【接口调用情况统计】区域框下面,单击"较慢"、"很慢"、"停滞"或"错误",系统跳转到【调用链分析】页面,可以进一步查看该应用的调用轨迹,定位故障的接口;
10. 单击【告警】页签,可以查看或设置该应用在不同告警级别下的超限阈值;

6.1.3 监控节点运行状态

提供节点运行状态监控,包括如下方面:

- a、所有节点的资源占用情况;
- b、每个节点的资源占用情况;
- c、每个节点上的应用实例分布与健康状态;
- d、每个节点上的应用和资源告警汇聚展示;

操作步骤:

- 1. 单击【运维】【应用监控】;
- 在左侧导航栏中选择【节点】,在该页面上方查看系统所监控的所有节点的 CPU、磁盘、物理内存和 虚拟内存的资源占用情况。节点卡片参数说明请参见下表:

参数	说明
节点名称	系统所监控的某一个节点的名称,即节点卡片左上角内容。
IP 地址	系统所监控的某一个节点所在的 IP 地址。
节点状态	系统所监控的某一个节点的运行状态,即节点卡片右上角内容。
	节点状态包含"异常"、"未知"、"可用"、"不可用"、"创建中"、"删 除中"、"删除失败"和"失败"。
问题/全部实例	系统所监控的某一个节点下所有应用的问题实例(问题实例指除了状态为"运行 中"外的所有实例)个数和总实例个数。
CPU	系统所监控的某一个节点的 CPU 使用率。单位:%。
硬盘	系统所监控的某一个节点的磁盘使用率。单位:%。
物理内存	系统所监控的某一个节点的物理内存使用率。单位:%。



参数	说明
虚拟内存	系统所监控的某一个节点的虚拟内存使用率。单位:%。
告警	节点指标对应的超限阈值的告警级别。告警级别包含"紧急"、"重要"、"次 要"和"提示"。
	说明: 当统计数据的值不为 0 时, 单击统计数据的值, 系统跳转到"告警查询"页面, 可以进一步查看、处理该节点的告警。

- 3. 在【节点监控】右侧的搜索框中输入节点名称,搜索要查询的节点;
- 单击节点名称,进入【节点监控详情】页面。在【节点监控详情】页面右上角的下拉列表框中选择 不同的统计周期,可以查看该节点在不同时间段的节点监控详情;单击【指标趋势】右侧的【添加 指标图表】,可以自定义指标图表在【指标趋势】下的展示,方便实时监控关注的指标和查看其趋势;

1) 在【实例】列表中,查看该节点的所有应用实例的概况。单击需要查看的实例的名称,系统 跳转到【实例监控详情】页面,可进一步查看该实例的运行状态,详细操作请参见 6.1.2 监控应 用运行状态;

2) 在【指标趋势】下面,查看该节点的指标趋势:

A、在指标图表的【更多】下拉列表框中选择【添加阈值】,可以对该节点指标创建阈值规则;

B、在指标图表的【更多】下拉列表框中选择【指标浏览】,系统跳转到【已选指标】页面, 可以通过设置统计周期、统计方式和时间范围,从不同维度查看该节点指标的指标图表;

3)在页面右侧的【告警】区域框下面,查看该节点在不同级别下告警的统计数据。当统计数据的值不为0时,单击统计数据的值,系统跳转到"告警查询"页面,可以进一步查看、处理该节点的告警;

4) 在页面右侧的【资源占用】区域框下面,查看该节点的 CPU、内存和磁盘占用情况;

5. 单击【告警】页签,可以查看或设置该节点在不同告警级别下的超限阈值;

6.1.4 创建阈值规则

当用户需要对某些指标进行重点监控并在异常情况下及时响应时,可在日常运维中通过对这些重点指标创建阈值规则。当指标值满足已设定的阈值条件时,系统会主动产生超限阈值,用户可在第一时间发现



异常并进行处理。

默认最多可以创建 1000 个阈值规则。在"所有阈值"、"超限阈值"、"数据不足"和"正常"四 个页面的任意一个页面中,均可用同样的方法创建阈值规则。以在"所有阈值"页面中创建阈值规则为例, 操作步骤如下:

- 1. 单击【运维】【应用监控】;
- 2. 在左侧导航栏中选择【阈值规则】【所有阈值】;
- 3. (可选)在【所有阈值】页面中,确认是否选中【创建阈值规则并发送超限阈值】后的复选框;

 1)若选中"创建阈值规则并发送超限阈值"后的复选框,当系统的"应用"命名空间中存在 "instanceAvailableRate"指标,"节点"命名空间中存在"diskUsedRate"(仅限于文件系 统中的diskUsedRate指标)或"nodeStatus"指标时,系统会自动为"instanceAvailableRate"、 "diskUsedRate"或"nodeStatus"指标创建对应的默认阈值规则。默认阈值规则支持修改和删 除操作。默认阈值规则的缺省设置请参见下表:

命名 空间	指标	命名方式	阈值条件	统计周 期	连续 周期	统计方 式	告 警 级别
应用	instanceAva ilableRate	命名空间-应用名称-指 标 例如, CONTAINER-aos-apiserv er-instanceAvailableR ate	instanceA vailableR ate<100%	1 分钟	2	平均值	重要
节点	diskUsedRat e	命名空间-挂载点-节点 名称-指标 例如, NODE-/var/log/audit-G lobal-MonitorO1-diskU sedRate 说明:挂载点和节点名称之间 的先后顺序可以调换,请以实 际环境为准。	diskUsedR ate≥90%	1 分钟	2	平均值	重要
节点	nodeStatus	命名空间-节点名称-指 标 例如, NODE-Global-Monitor01	nodeStatu s≥1	1 分钟	2	平均值	重要



命名 空间	指标	命名方式	阈值条件	统计周 期	连续 周期	统计方 式	告 警 级别
		-nodeStatus					

2)若取消选中"创建阈值规则并发送超限阈值"后的复选框,系统不会实时创建默认阈值规则,
 同时不会删除已经创建的阈值规则;

- 4. 在【所有阈值】页面中,单击【添加阈值】;
- 5. 在【选择指标】页面中,选择一个关注的指标,指标的详细信息请参见 6.1.9 应用监控指标总览;
- 6. 在【视图预览】区域框下面设置已选指标的统计方式、统计周期和时间范围,也可以通过执行步骤8.2设置已选指标的统计方式和统计周期;
- 7. 单击【下一步】;
- 8. 在【定义阈值】页面中,根据需要设置阈值规则的相关参数:

 1) 在【阈值设置】区域框下面,输入【阈值名称】和【阈值描述】并设置已选择指标的【阈值 条件】、【连续周期】和【告警级别】;

- 9. 单击【确定】;
- 10. 后续处理:
 - 1) 可以搜索、修改和删除已创建的阈值规则;
 - 2) 可以通过如下操作查看超限阈值,并对该超限阈值进行分析和处理;

A、在左侧导航栏中选择【阈值规则】【超限阈值】;

B、单击超限阈值名称前的²,查看其详细信息;

C、在超限阈值名称右侧【操作】列的【更多】中选择【历史记录】, 查看其历史记录;

6.1.5 创建监控视图

用户通过创建监控视图,将需要对比的指标数据置于同一监控视图中,方便对指标数据和趋势进行对 比。默认最多可以创建 100 个监控视图。监控视图的每个指标图表中最多可以选择 4 条指标。操作步骤如

²⁾⁽可选)在【阈值预览】区域框下面,设置已选指标的【统计方式】和【统计周期】;



- 1. 单击【运维】【应用监控】;
- 2. 在左侧导航栏中选择【Dashboard】;
- 在【操作】下拉列表框中选择【创建监控视图】,或单击【操作】下拉列表框右侧的【创建监控视图】;
- 4. 输入监控视图名称后单击【确定】;
- 在【选择指标】页面中,选择一个或多个关注的指标,指标的详细信息请参见 6.1.9 应用监控指标 总览;
- 6. 单击平均值 *和5分钟 *分别设置统计方式和统计周期;
- 7. 单击【确定】;
- 8. 在页面右侧单击【保存监控视图】,或在【操作】下拉列表框中选择【保存监控视图】;
- 9. 后续处理:
 - 1) 在日常监控中,打开已创建的监控视图,可以实时观察指标数据的走向;
 - 2) 可以向已创建的监控视图中添加指标图表,将多个指标图表置于同一监控视图中;
 - 3) 可以删除不需要的监控视图;

6.1.6 上报自定义指标并进行监控

用户已上报自定义指标。可以通过如下两种方式上报自定义指标:

1、方式一:参考应用监控 API 参考,上报自定义指标;

2、方式二:在创建容器应用时,通过自定义监控功能上报自定义指标。自定义监控功能的详细设置请参见
 11.2 如何配置自定义监控?

操作步骤如下:

- 1. 单击【运维】【应用监控】;
- 2. 在左侧导航栏中选择【指标监控】;
- 3. 在【自定义指标】下拉框中选择【CUSTOMMETRICS】,查看自定义指标是否上报成功;
- 4. (可选)使用自定义指标创建阈值规则,详细操作请参见 6.1.4 创建阈值规则;
- 5. (可选)使用自定义指标创建监控视图,详细操作请参见 6.1.5 创建监控视图;



- 6. (可选)将自定义指标添加到【指标趋势】中,以应用的自定义指标为例,详细操作如下:
- 7. 单击【确定】;
- 8. 在页面右侧单击【保存监控视图】,或在【操作】下拉列表框中选择【保存监控视图】;
 - 1) 在左侧导航栏中选择【应用】;
 - 2) 在【应用监控】页面右上角的搜索框中输入关键字,搜索应用;
 - 3) 单击应用名称,进入【应用监控详情】页面;
 - 4) 在【概览】页签中单击【添加指标图表】;
 - 5) 在【添加指标图表】对话框左侧的【自定义指标】下拉列表框中选中该自定义指标;
 - 6) 在【添加指标图表】对话框右侧设置该自定义指标的【统计方式】;
 - 7) 单击【确定】;

6.1.7 在第三方应用中集成监控应用

前提条件:

1、已具备"应用监控"的操作权限;

2、已登录系统,或第三方应用已与应用监控进行了 SSO (single sign-on)对接;

操作步骤如下:

- 1. 单击【运维】【应用监控】;
- 2. 点击【指标监控】,选择一个命名空间;
- 3. 在命名空间中选择一个或多个关注的指标,指标的详细信息请参见 6.1.9 应用监控指标总览;
- 4. 在下方的图表中设置指标的统计方式、统计周期和时间范围:
 - 1) 当设置时间范围时,如果设置了相对时间范围,时间范围和统计周期有如错误!未找到引用
 - 。所示的默认对应关系,可以根据需要重新设置统计周期;

时间范围	统计周期
●最近3小时	5 分钟



时间范围	统计周期
●最近 12 小时	
•最近一天	1 小时
● 最近一周	24 小时
● 最近 15 天	

2) 当设置时间范围时,如果设置了绝对时间范围,需要单击 17 才能刷新指标图表;

- 5. 单击【复制 URL】;
- 6. 在第三方应用中调用复制的 URL 以获取数据;

6.1.8 分析指标历史数据并导出监控报告

前提条件:应用监控中已存在需要进行分析的指标。

操作步骤如下:

- 1. 单击【运维】【应用监控】;
- 2. 点击【指标监控】,选择一个命名空间;
- 3. 在命名空间中选择一个或多个关注的指标,指标的详细信息请参见 6.1.9 应用监控指标总览;
- 4. 在下方的图表中设置指标的统计方式、统计周期和时间范围:

1)当设置时间范围时,如果设置了相对时间范围,时间范围和统计周期有如错误!未找到引用 。所示的默认对应关系,可以根据需要重新设置统计周期;

时间范围	统计周期
●最近 3 小时 ●最近 12 小时	5 分钟
●最近一天	1 小时
●最近一周 ●最近 15 天	24 小时

2) 当设置时间范围时,如果设置了绝对时间范围,需要单击 7 才能刷新指标图表;

5. 单击【导出监控报告】;

6. 查看指标图表,并对指标历史数据进行分析;





6.1.9 应用监控指标总览

1. 应用指标**:**

指标	解释
cpucorelimit	CPU 内核总量
cpucoreUsed	CPU 内核占用
cpuUsage	CPU 使用率
diskReadRate	磁盘读取速率
diskWriteRate	磁盘写入速率
memUsed	内存使用量
memCapacity	内存总量
instanceAvailableRate	实例可用率
sendBytesRate	数据发送速率
recvBytesRate	数据接收速率
totalBytesRate	数据总传输速率
memUsage	物理内存使用率
totalDiskRate	总磁盘读写速率

2. 容器指标:

指标	解释
cpucorelimit	CPU 内核总量
cpucoreUsed	CPU 内核占用
cpuUsage	CPU 使用率
diskReadCount	磁盘读取次数
diskReadRate	磁盘读取速率
diskWriteCount	磁盘写入次数



指标	解释
diskWriteRate	磁盘写入速率
memUsed	内存使用量
memCapacity	内存总量
sendBytesRate	数据发送速率
recvBytesRate	数据接收速率
totalBytesRate	数据总传输速率
memUsage	物理内存使用率
totalDiskRate	总磁盘读写速率
totalDiskRead	总磁盘读取大小
totalDiskWrite	总磁盘写入大小

3. 无状态应用指标:

指标	解释
cpucorelimit	CPU 内核总量
cpucoreUsed	CPU 内核占用
cpuUsage	CPU 使用率
diskReadRate	磁盘读取速率
diskWriteRate	磁盘写入速率
memUsed	内存使用量
memCapacity	内存总量
instanceAvailableRate	实例可用率
sendBytesRate	数据发送速率
recvBytesRate	数据接收速率



指标	解释
totalBytesRate	数据总传输速率
memUsage	物理内存使用率
totalDiskRate	总磁盘读写速率

4. 无状态容器指标:

指标	解释
cpucorelimit	CPU 内核总量
cpucoreUsed	CPU 内核占用
cpuUsage	CPU 使用率
diskReadCount	磁盘读取次数
diskReadRate	磁盘读取速率
diskWriteCount	磁盘写入次数
diskWriteRate	磁盘写入速率
memUsed	内存使用量
memCapacity	内存总量
sendBytesRate	数据发送速率
recvBytesRate	数据接收速率
totalBytesRate	数据总传输速率
memUsage	物理内存使用率
totalDiskRate	总磁盘读写速率
totalDiskRead	总磁盘读取大小
totalDiskWrite	总磁盘写入大小

5. 应用实例指标:

指标	解释

()	人異し
	eciouu

指标	解释
cpucorelimit	CPU 内核总量
cpucoreUsed	CPU 内核占用
cpuUsage	CPU 使用率
diskReadRate	磁盘读取速率
diskWriteRate	磁盘写入速率
memUsed	内存使用量
memCapacity	内存总量
status	实例状态
sendBytesRate	数据发送速率
recvBytesRate	数据接收速率
totalBytesRate	数据总传输速率
memUsage	物理内存使用率
totalDiskRate	总磁盘读写速率

6. 磁盘设备指标:

指标	解释
disklOUtil	磁盘 10 操作时间占比
disklOWait	磁盘 10 等待时间
diskReadRate	磁盘读取速率
diskReadRspTime	磁盘读响应时间
diskWriteRate	磁盘写入速率
diskWriteRspTime	磁盘写响应时间

7. 文件系统指标:

指标	解释
diskCapacity	磁盘空间容量



指标	解释
diskUsedRate	磁盘使用率
diskAvailableCapacity	可用磁盘空间

8. 网卡指标:

指标解释sendBytesRate上行 BpssendPackRate上行 ppssendBandWideUsage上行带宽利用率recvBytesRate下行 BpsrecvPackRate下行 ppsrecvBandWideUsage下行带宽利用率tota1BytesRate总 Bpstota1PackRate总 ppstota1BandWideUsage总带宽利用率		
sendBytesRate上行 BpssendPackRate上行 ppssendBandWideUsage上行带宽利用率recvBytesRate下行 BpsrecvPackRate下行 ppsrecvBandWideUsage下行带宽利用率totalBytesRate总 BpstotalPackRate总 ppstotalPackRate总 mps	指标	解释
sendPackRate上行 ppssendBandWi deUsage上行带宽利用率recvBytesRate下行 BpsrecvPackRate下行 ppsrecvBandWi deUsage下行带宽利用率totalBytesRate总 BpstotalPackRate总 ppstotalBandWi deUsage总带宽利用率	sendBytesRate	上行 Bps
sendBandWideUsage上行带宽利用率recvBytesRate下行 BpsrecvPackRate下行 ppsrecvBandWideUsage下行带宽利用率totalBytesRate总 BpstotalPackRate总 ppstotalBandWideUsage总带宽利用率	sendPackRate	上行 pps
recvBytesRate下行 BpsrecvPackRate下行 ppsrecvBandWideUsage下行带宽利用率totalBytesRate总 BpstotalPackRate总 ppstotalBandWideUsage总带宽利用率	sendBandWideUsage	上行带宽利用率
recvPackRate下行 ppsrecvBandWideUsage下行带宽利用率totalBytesRate总 BpstotalPackRate总 ppstotalBandWideUsage总带宽利用率	recvBytesRate	下行 Bps
recvBandWideUsage下行带宽利用率totalBytesRate总 BpstotalPackRate总 ppstotalBandWideUsage总带宽利用率	recvPackRate	下行 pps
totalBytesRate总 BpstotalPackRate总 ppstotalBandWideUsage总带宽利用率	recvBandWideUsage	下行带宽利用率
totalPackRate总 ppstotalBandWideUsage总带宽利用率	totalBytesRate	总 Bps
totalBandWideUsage 总带宽利用率	totalPackRate	总 pps
	tota BandWideUsage	总带宽利用率

9. 节点指标:

指标	解释
cpucorelimit	CPU 内核总量
cpucoreUsed	CPU 内核占用
cpuUsage	CPU 使用率
diskUsedRate	磁盘使用率
nodeStatus	节点状态
processNum	进程数量
freeMem	可用物理内存
freeVirMem	可用虚拟内存
sendBytesRate	数据发送速率



指标	解释
recvBytesRate	数据接收速率
totalBytesRate	数据总传输速率
totalMem	物理内存容量
memUsedRate	物理内存使用率
totalVirMem	虚拟内存容量
virMemUsedRate	虚拟内存使用率

10. 接口实例:

指标	解释
tpn_0	tpn_0
tpn_1	tpn_1
tpn_2	tpn_2
tpn_3	tpn_3
tpn_4	tpn_4
successRate	成功率
avgDelay	平均时延
successCalls	调用成功次数
throughput	吞吐量
totalDelay	总时延
totalCalls	总调用次数

11. 服务实例指标:

指标	解释
successRate	成功率
insStatus	健康度状态
avgDelay	平均时延



指标	解释
tp50	时延 TP50
tp999	时延 TP999
throughput	吞吐量

12. 接口指标:

指标	解释
successRate	成功率
avgDelay	平均时延
tp50	时延 TP50
tp999	时延 TP999
throughput	吞吐量

13. 服务指标:

指标	解释
successRate	成功率
avgDelay	平均时延
instanceAvailableRate	实例可用率
tp50	时延 TP50
tp999	时延 TP999
throughput	吞吐量

6.2 应用调用分析

6.2.1 概念

应用调用分析(Application Tracing Service,简称 ATS)跟踪、记录接口的调用情况,可视化地还 原业务请求在分布式系统中的执行轨迹和状态,用于性能及故障快速定界。



6.2.2 配置采样率

调用链会产生海量数据,用户可以通过配置采样率减少数据量。对于重要的应用或接口,还可以单独 设置采样率。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 _____,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【运维】【应用调用分析】;
- 6. 在左侧导航栏中选择【调用链配置】【采样率】;
- 在【采样率】页面中按照界面提示分别设置全局采样率、应用采样率、接口采样率。因为采样率设置过高,会产生大量的数据,占用大量数据库和磁盘空间,所以请进行合理的设置;

6.2.3 配置老化周期

应用调用分析老化周期是指调用链历史数据的保存天数,默认为7天,根据需要可以进行调整。当磁 盘空间不足或不需要长期保存调用链历史数据时,可以配置数据保存天数。

- 1. 单击【运维】【应用调用分析】;
- 2. 在左侧导航栏中选择【调用链配置】【老化周期】;
- 在【老化周期】页面中根据需要设置数据保存天数。数据保存天数默认为7天,取值范围为1~7 天;
- 4. 单击【保存】;

6.2.4 查看业务关键性能指标

当需要查看某一个业务的运行情况时,可以在"业务统计"页面中查看该业务在某一段时间内的平均 响应时间、最大响应时间、最小响应时间和调用次数等关键性能指标,通过关键性能指标对该业务运行情 况进行评估。

前提条件:已部署带有调用链埋点功能的应用。推荐使用具有调用链埋点功能的 CSE(Cloud Service



Engine)开发框架开发应用,详细内容请参见该版本《开发指南》的"调用链跟踪"章节。

操作步骤:

- 1. 单击【运维】【应用调用分析】;
- 2. 在左侧导航栏中选择【业务统计】;
- 3. (可选)在【业务统计】页面中单击【堆栈】或【应用】,并在右侧的下拉列表框中进行选择;
- 4. 在【业务统计】页面右上角的下拉列表框中选择统计周期;
- 5. 在【调用链分析】页面中对异常业务进行进一步地分析和定界;

6.2.5 性能瓶颈定界

运维人员从应用监控或告警中发现应用的接口响应较慢,通过应用调用分析对该应用在一段时间内所 有调用链的耗时情况进行分析,锁定可疑的调用链,找出性能瓶颈。

前提条件:已部署带有调用链埋点功能的应用。推荐使用具有调用链埋点功能的 CSE(Cloud Service Engine)开发框架开发应用,详细内容请参见该版本*《开发指南》*的"调用链跟踪"章节。

操作流程:



操作步骤:

天翼**云** e Cloud

- 1. 单击【运维】【应用调用分析】;
- 2. 在左侧导航栏中选择【调用链分析】;
- 在【调用链分析】页面中查询相应的调用链。单击^Q,可以自定义显示列,设置筛选条件时,可
 以同时设置多个标签和属性,单击【清除条件】,可以清除已设置的筛选条件;
 - 1) 单击搜索框设置应用或业务名称等筛选条件;
 - 2) 通过搜索框右侧的下拉列表框设置时间信息;
 - 3) 单击【查询】;
- 4. 在【调用 ID】列单击相应的调用 ID, 查看业务的调用过程;
- 5. 根据"时间线"锁定耗时较长的接口,找出性能瓶颈;
- 6. (可选)查看接口调用的辅助信息,进一步定界问题产生的原因:



- 1)单击【操作】列的 3查看相应应用的日志信息;
- 2) 单击【操作】列的 🔀 查看相应调用的扩展信息;

7. 根据结果对性能瓶颈接口进行优化或对性能瓶颈应用进行升级、扩容;

6.2.6 故障辅助定界

运维人员从应用监控或告警中发现应用的接口调用失败,通过应用调用分析对该应用在一段时间内的 所有调用结果进行分析,找出故障的调用链。

前提条件:已部署带有调用链埋点功能的应用。推荐使用具有调用链埋点功能的 CSE(Cloud Service Engine)开发框架开发应用,详细内容请参见该版本*《开发指南》*的"调用链跟踪"章节。

操作流程:



操作步骤:


- 1. 单击【运维】【应用调用分析】;
- 2. 在左侧导航栏中选择【调用链分析】;
- 在【调用链分析】页面中查询相应的调用链。单击²,可以自定义显示列,设置筛选条件时,可
 以同时设置多个标签和属性,单击【清除条件】,可以清除已设置的筛选条件;
 - 1) 单击搜索框设置应用或业务名称等筛选条件;
 - 2) 通过搜索框右侧的下拉列表框设置时间信息;
 - 3) 单击【查询】;
- 4. 在【调用 ID】列单击相应的调用 ID, 查看业务的调用过程;
- 在【状态】列查看相应的状态、接口返回值和描述信息是否正常,找出故障。将鼠标移至【状态】
 列,可以通过弹出的 Tip 信息查看相应的接口返回值和描述信息;
- 6. (可选)查看接口调用的辅助信息,进一步定界问题产生的原因:
 - 1) 单击【操作】列的 2查看相应应用的日志信息;
 - 2) 单击【操作】列的 🔀 查看相应调用的扩展信息;
- 7. 根据故障信息进行深入分析或联系故障对应技术支持工程师处理故障;

6.3 应用日志

6.3.1 概念

应用日志(Application Log Service,简称 ALS)是一个集中式日志管理平台,主要提供了海量运行 日志的数据采集、检索和分析等功能,可以提升运维效率。

6.3.2 查询日志

当需要通过日志来分析和定位问题时,使用日志检索功能可以快速地在海量日志中查询到所需的日志, 还可以结合日志的来源信息和上下文原始数据一起辅助定位问题。

前提条件: 创建容器应用时已配置"卷策略"为"logs"。配置"卷策略"的详细操作请参见 11.6 如何设置容器应用数据卷?。



- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心, 切换到杭州节点;
- 3. 单击 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【运维】【应用日志】;
- 在【日志检索】页面中按照界面提示设置日志查询条件。页面右侧的搜索框中最多可以输入 255 个 字符,搜索时字母不区分大小写。单击【清除】,可以清除已设置的查询条件;
- 查看日志查询结果。符合查询条件的日志会按照时间倒序排序(例如,今天,昨天,前天的顺序排 序),并且关键词会高亮显示;

1)单击日志列表左侧的[>],可以进一步查看该条日志的节点名称、节点 IP 和文件路径等来源信息;

2)单击日志列表右侧的[>],可以进一步查看该条日志的上下文原始数据,该条日志在上下文原始 数据中高亮显示;

3)在【上下文显示行数】下拉列表框中,可以设置该条日志的上下文原始数据显示行数。例如, 设置"上下文显示行数"为"200", 若该日志之前已打印的日志条数 ≥100,该日志之后已打 印的日志条数 ≥99,则该日志之前的 100 条和之后的 99 条日志会被作为上下文显示;若该日志 之前已打印的日志条数 <100 (例如,已打印 90 条日志),该日志之后已打印的日志条数 <99 (例 如,已打印 80 条日志),则该日志之前的 90 条和之后的 80 条日志会被作为上下文显示;

4)单击【导出】,可以将已显示的日志上下文原始数据导出到本地。为了保障租户节点和应用的 正常运行,租户的节点上会运行部分系统提供的应用(如 kube-dns 应用)。查询租户日志时也会 查询到这些应用的日志;

8. (可选)单击【导出本页】,可以将当前页查询结果导出到本地;

6.3.3 配置日志存储时长

在磁盘大小已定的情况下,日志不可能无限期存储在磁盘中,当日志实际存储时长超过已设置的存储 时长时,该日志将被自动删除。用户可以根据实际情况对日志存储时长进行配置。日志存储时长默认为3 天,根据需要可以进行调整。



日志存储时长以区域为单位进行配置,不同区域间的日志存储时长彼此独立,即对不同区域可以配置 不同的日志存储时长。同一个区域下不同用户配置的日志存储时长,以最后一次配置为准。

- 1. 单击【运维】【应用日志】;
- 2. 将鼠标移至页面右上角的 部, 选择需要配置日志存储时长的区域;
- 3. 在左侧导航栏中选择【应用日志】【日志配置】;
- 4. 在【日志配置】页面中根据需要设置日志保存天数。日志保存天数默认为3天,取值范围为1~7 天;
- 5. 单击【保存】;

6.4 事件

6.4.1 查询事件

通过在"事件查询"页面中设置过滤条件,可查看需要关注的事件。

操作流程: 当修改了当前设置的过滤条件时, 系统即按照新的过滤条件进行搜索。

操作步骤:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 _____,选择【企业中间件】【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【运维】【事件】;
- 6. 在左侧导航栏中选择【事件查询】,单击℃设置过滤条件,单击【过滤】进行查询;
- 7. 单击【清空】可以清空过滤条件;

6.5 告警

告警服务提供了监控、查询告警等功能,便于更快地发现、定位并恢复网络故障。



6.5.1 监控告警

通过在"告警监控"页面中设置告警监控模板,实时监控当前需要关注和处理的告警。

操作流程: 当修改了当前设置的告警监控模板时, 系统即按照新的模板进行监控。

操作步骤:

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 ,选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【运维】【告警】;
- 6. 在左侧导航栏中选择【告警监控】,单击
 设置过滤条件,单击【另存为】创建告警监控模板;
 说明:
 - 1)设置过滤条件后,需要将过滤条件存为模板后才能按照过滤条件进行监控;
 - 2) 单击【清空】可以清空过滤条件;
 - 3) 可在右上角的【所有告警】下拉列表框中选择现有模板进行监控;
 - 4)选择现有模板后,可对其进行修改,单击【保存】确认修改;
- 7. 设置模板名称,单击【确定】。【告警监控】页面中最多显示 50 条告警。如果需要查看更多,请在 "告警查询"页面中查看;
- 8. 在【告警监控】页面中可以执行如下操作:

操作	方法	说明
锁定告警	在页面中单击【滚动锁定】,当前 告警列表中的告警处于锁定状态。 此时【滚动锁定】按钮自动更新为 【滚动解锁】按钮。	当告警处于锁定状态时需要注 意,新上报的告警不会更新到当 前告警列表中,解锁后才会更新 到当前告警列表中。
解锁告警	在页面中单击【滚动解锁】,系统 会自动上报告警到当前告警列表 中。此时【滚动解锁】按钮自动更	-



操作	方法	说明
	新为【滚动锁定】按钮。	
确认告警	 • 在页面中选中一条或多条告警, 单击【确认】。 • 在当前告警列表中单击目标告 警所在【操作】列中的^会。 	"确认"表示用户知道当前已 经发生告警。
反确认告警	 在页面中选中一条或多条告警, 单击【反确认】。 在当前告警列表中单击目标告 警所在【操作】列中的^会。 	通过反确认后,告警由已确认 状态变成未确认状态。
清除告警	 • 在页面中选中一条或多条未清除的告警,单击【清除】。 ● 在当前告警列表中单击目标告警所在【操作】列中的 	 "清除"表示告警故障已经解除。已清除的告警底色将变成绿色。 当用户对目标告警执行确认和清除操作约3秒钟后,该告警会自动从告警列表中删除。 被执行确认和清除操作后的告警将不能进行监控。
定制告警列表显示 列	单击 ^{•••} ,在打开的页面中设置告警 列表显示列,单击【确定】。	此设置即时生效,当前用户再次 登录系统时也生效,但不对其他 用户生效。
查看告警详情	单击》可查看告警详情。	-

6.5.2 查询告警

通过在"告警查询"页面中设置过滤条件,查看当前需要关注和处理的告警。

操作流程: 当修改了当前设置的过滤条件时, 系统即按照新的过滤条件进行搜索。。

操作步骤:

- 1. 单击【运维】【告警】;
- 在左侧导航栏中选择【告警查询】,单击♥设置过滤条件,单击【过滤】进行查询板;
 说明:
 - 1) 单击【清空】可以清空过滤条件;



2)设置过滤条件后,单击【另存为】,将过滤条件存为模板,下次可直接在右上角的【所有告警】
 下拉列表框中选择该模板进行查询;

- 3)【告警查询】页面中最多显示 2500 条告警;
- 3. 在【告警查询】页面中可以执行如下操作:

操作	方法	说明
确认告警	 ・在页面中选中一条或多条告警, 単击【确认】。 ・在告警列表中单击目标告警所 在【操作】列中的^会。 	"确认"表示用户知道当前已经发生告 警。
反确认告警	 在页面中选中一条或多条告警, 单击【反确认】。 在当前告警列表中单击目标告 警所在【操作】列中的^会。 	通过反确认后,告警由已确认状态变成未 确认状态。
清除告警	 •在页面中选中一条或多条未清除的告警,单击【清除】。 •在当前告警列表中单击目标告 警所在【操作】列中的 	 "清除"表示告警故障已经解除。已清除的告警底色将变成绿色。 当用户对目标告警执行确认和清除操作约3秒钟后,该告警会自动从告警列表中删除。 被执行确认和清除操作后的告警将不能进行查询。
定制告警列 表显示列	单击 ^交 ,在打开的页面中设置告警 列表显示列,单击【确定】。	此设置即时生效,当前用户再次登录系统时 也生效,但不对其他用户生效。
查看告警详 情	单击》可查看告警详情。	_
导出告警	 导出部分告警:在告警列表选中 一条或多条告警,单击【导出】 下拉列表框,选择【导出选中】。 导出全部告警:单击【导出】下 拉列表框,选择【导出选中】。 说明:如果当前按照某些过滤条件 查询告警,则"导出全部"是指导 出符合该过滤条件的全部告警。 	 最多导出 300 条告警。 告警文件保存为压缩包格式。
按照告警列 表表头字段 排序	单击告警列表表头字段,对告警进 行排序。	可以按照"名称"、"告警源"、"产 生时间"和"级别"四个字段中的任 一字段进行排序。

6.6 弹性伸缩

6.6.1 概念

e Cloud

根据用户的业务需求,通过设定策略自动调整业务使用的资源。您可以根据业务需求自行定义弹性伸 缩策略,满足业务需求的同时,减少资源投入。

6.6.2 创建弹性伸缩组

针对应用创建弹性伸缩组,通过弹性伸缩组中设置的伸缩策略,自动调整业务资源,以满足业务需求。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心,切换到杭州节点;
- 3. 单击 , 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【运维】【弹性伸缩】;
- 6. 在弹性伸缩组页面中,单击

➡ 弹性伸缩组 以创建弹性伸缩组:

参数	说明
伸缩组类型	伸缩组所针对的对象类型,一般为应用。
伸缩组名称	名称。
集群名称	伸缩组对应的集群。单击下拉选项进行选择。
应用名称	伸缩组对应的应用。单击下拉选项进行选择。
最小实例数	应用的最小实例数。
最大实例数	应用的最大实例数。
冷却时间(秒)	两次扩容/缩容之间的间隔时间。

- 7. 单击【立即创建】,并确认规格,单击【确认】完成弹性伸缩组的创建;
- 填写完成弹性伸缩组的各项参数后,也可以单击【下一步】进行伸缩策略的添加。或者参见 6.6.3 添 加伸缩策略;



9. 后续操作:

天翼 Cloud

后续操作	操作步骤
启用弹性伸缩组	方法一:
	1、在主菜单中选择【运维】【弹性伸缩】;
	2、在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【启用】;
	方法二:
	1、在主菜单中选择【运维】【弹性伸缩】;
	2、在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【伸缩策略】;
	3、单击页面右上方的【启用】;
停用弹性伸缩组	方法一:
	1、在主菜单中选择【运维】【弹性伸缩】;
	2、在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【停用】;
	方法二:
	1、在主菜单中选择【运维】【弹性伸缩】;
	2、在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【伸缩策略】;
	3、单击页面右上方的【挺用】;
删除弹性伸缩组	1、在主菜单中选择【运维】【弹性伸缩】;
	2、在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【删除】;
	3、在确认对话框中,单击【删除】;
修改弹性伸缩组	1、在主菜单中选择【运维】【弹性伸缩】;
	2、在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【伸缩策略】;
	3、单击页面右上角的【修改】,以修改伸缩组的相关信息;

6.6.3 添加伸缩策略

通过在弹性伸缩组中设置伸缩策略,能够对应用实现自动调整业务资源,以满足业务需求。

1. 单击【运维】【弹性伸缩】;



- 2. 在弹性伸缩组页面中,单击弹性伸缩组列表中操作列的【伸缩策略】;
- 3. 单击【添加伸缩策略】;
- 4. 在添加伸缩策略页面,填写或选择相关参数:

参数	说明
策略类型	告警触发。
策略名称	策略名称。
指标名称	可选择应用指标或自定义指标。
	说明:应用指标为系统内置的应用常用指标,可单击下拉选 项进行选择。包括 CPU 使用率、物理内存使用率等。
阈值条件	设置伸缩策略触发的条件及动作。
连续周期	若设置为 3,则表示若指标数据连续三个统计周期达到了设定 的阈值,则触发策略动作。
统计周期	指标统计周期。
统计方式	平均值。

参数	说明
策略类型	定时触发。
策略名称	策略名称。
触发时间	策略触发时间。
执行动作	包括"增加"、"减少"、"设置"。
实例个数	执行动作对应的实例个数。

参数	说明
策略类型	周期触发。
策略名称	策略名称。



参数	说明
开始时间	策略的开始时间。
结束时间	策略的结束时间。
重复周期	包括"每天"、"每周"、"每月"。 若重复周期为每周,还需要选择星期。 若重复周期为每月,还需要选择日期。
触发时间	策略触发时间。
执行动作	包括"增加"、"减少"、"设置"。
实例个数	执行动作对应的实例个数。

- 5. 单击【确定】完成伸缩策略的添加;
- 6. (可选)在伸缩策略列表中,可单击操作列的"删除"、"停用"、"启用"、"修改"对相应策 略进行操作;

7

软件中心

7. 添加伸缩策略也可以在创建弹性伸缩组的同时进行,可参见 6.6.2 创建弹性伸缩组;

7.1 镜像仓库

7.1.1 概念

仓库包括镜像仓库和软件仓库,镜像用于容器类应用,软件包用于虚拟机应用。用户在创建应用前, 需要将应用所需的镜像或软件包上传到仓库中。





- 软件仓库不扫描用户上传的镜像和软件,不负责对用户上传的镜像和软件内容进行安全性验证。上传的镜像和软件内容中请不要包含未加密的口令,密码等隐私信息,以避免隐私泄露。下载公有镜像和软件时,应确定数据来自于可信的仓库源,以避免下载到恶意软件。
- 如果使用自定义镜像,请确保镜像来源可信,不在容器镜像内安装不必要的软件,在升级时使用安全 补丁升级镜像。使用第三方镜像,造成的后果(例如,环境不可用)用户需自己承担。
- 磁盘满只会导致无法上传镜像或软件包到仓库,会产生异常提示信息,不会影响其他服务;为防止其 它业务(例如日志)把磁盘占满,导致仓库无法上传,建议对仓库的存储独立挂盘。

在使用之前,您需要了解以下基本概念:

术语	解释
仓库	仓库是用于存储、管理应用软件和 docker 容器镜像的场所,可以让使 用人员轻松存储、管理、部署应用软件和 docker 容器镜像。
	说明:
	 当租户名含大写字母时,软件仓库中的各项信息会将租户名转化 为小写来处理。包括但不限于:仓库地址、软件仓库中各个软件 包的"下载地址"、容器镜像下载地址等。
	 公有仓库的内容,任何用户均可以下载,包括非登录用户。
镜像仓库	提供 docker 容器镜像管理功能,用户在创建容器应用前,需要将应用 所需的镜像上传到镜像仓库。docker 镜像是一个模板,用于创建 docker 容器。docker 提供了一个简单的机制来创建新的镜像或更新已有镜像。 例如,一个 docker 镜像可以包含一个完整的 ubuntu 操作系统环境, 里面仅安装了用户需要的应用程序及其依赖文件。
软件仓库	提供软件包管理功能,用户在创建虚拟机或物理机应用前,需要将应 用所需的软件包上传到软件仓库。

镜像仓库首页中包含如下内容:

1、我的镜像:展示了用户创建的所有镜像仓库。

2、第三方公开镜像:展示了所有用户上传到平台的公开镜像,您也可以基于公开镜像创建应用。

7.1.2 上传私有镜像(外网)

"创建容器应用"或"通过编排创建堆栈(若堆栈中有容器应用)"时,均需要先上传容器镜像。上 传成功后,会在镜像仓库显示已上传成功的镜像。本章节以 ng inx:1.10 镜像为例,介绍如何上传镜像到 ServiceStage 镜像仓库。

说明: 在使用 Docker 客户端上传镜像的过程中,如果出现网络异常断开或软件仓库被异常重启的情况, 是由于 Docker 客户端,其不会继续向软件仓库发送上传镜像的请求,从而导致 Docker 客户端卡机无法退 出。在此情况下可以通过重启 docker daemon 的方式解决。

前提条件:

1、准备一台已安装 docker 客户端的虚拟机,且 docker 版本为 1.11.2。您可以在

<u>https://www.docker.com/</u>下载 docker, 安装指导请参见 <u>https://docs.docker.com/;</u>

- 2、确保虚拟机能连通外网;
- 3、已获取 2.1.2 上传 AK/SK 证书中的 AK/SK;
- 4、请确保镜像的正确性,能够成功后台启动;

操作步骤:

- 1. 执行以下操作,允许 docker 访问我的镜像仓库:
 - 1) 以 root 用户登录 docker 所在的虚拟机;
 - 2) 配置 docker 参数:

A、Ubuntu 系统下,执行 vi /etc/default/docker;

在 DOCKER_OPTS 中, 增加如下租字体所示:

DOCKER_OPTS="--insecure-registry registry.cn-north-1.hwclouds.com"

B、SuSE 操作系统下,执行 vi /etc/sysconfig/docker;

若配置文件中有 DOCKER_OPTS 字段,则在在 DOCKER_OPTS 中,增加如下租字体所示:

DOCKER_OPTS="--insecure-registry registry.cn-north-1.hwclouds.com"



若配置文件中没有 DOCKER_OPTS 字段,在 INSECURE_REGISTRY 中,增加如下粗体所示:

INSECURE_REGISTRY="--insecure-registry registry.cn-north-1.hwclouds.com"

- 3)保存文件并退出;
- 4) 重启 docker;

service docker restart

- 2. 创建镜像仓库:
 - 1) 登录 ServiceStage, 单击菜单栏的【仓库】【镜像仓库】;
 - 2) 在右侧页面中,单击【创建镜像仓库】;
 - 3) 设置如下参数:

参数	参数说明
*仓库名称	容器镜像仓库名称,命名必须唯一。本例配置为:nginx。
	说明:上传镜像时创建的镜像仓库名称应与上述第三点中打包镜像的镜像仓 库名称保持一致。
	例如,上述第三点中打包的镜像为 ng inx. tar, 则创建镜像仓库时镜像仓 库的名称应为 ng inx。
*类型	容器镜像仓库类型,默认为公有。
	• 公有:任何租户、用户和非登录用户均可以下载。
	• 私有: 仅用户本人可用。
*仓库分类	根据镜像类型选择对应仓库分类。
描述	容器镜像仓库的描述信息。

4) 配置完成后,单击【创建镜像仓库】;

5) 单击【镜像仓库详情】,进入仓库详情页面;

3. 获取仓库访问权限文件:

- 1) 在【Pull/Push 指南】页面,单击【生成 docker login 指令】;
- 2) 可查看到生成的 docker login 指令;



- 3) 单击 🕞, 复制 docker login 指令;
- 4. 授权 Docker 客户端访问仓库的权限:
 - 1) 以 root 用户登录 Docker 所在的虚拟机;
 - 2) 将步骤 3.3 中拷贝到虚拟机中,执行 docker login 命令。例如执行如下命令:

```
docker login -u cn-north-1@5HPBFFCG0AlQMEES0EVA -p
a501765e5e2edd27b86668268e82d63fdb99298c8935f34e745e6a7cfc56c926
registry.cn-north-1.hwclouds.com:443
```

回显如下表示登录成功:

Login Succeeded

- 5. 给 nginx:1.10 镜像打标签:
 - 1) 查看镜像 ID:

docker images

2) 为镜像打标签

```
sudo docker tag [IMAGE ID] [本镜像仓库地址]/[租户名]/[仓库名]:[版本号]
```

样例如下:

```
sudo docker tag 5766334bdaa0 registry.cn-north-1.hwclouds.com/aos-team/nginx:1.10
```

```
其中,registry.cn-north-1.hwclouds.com:443 为镜像仓库外网地址,aos-team 为租户名,nginx
为仓库名,1.10 为版本号。
```

6. Push 镜像至镜像仓库:

```
sudo docker push [tag]
```

样例如下:

```
sudo docker push registry.cn-north-1.hwclouds.com:443/aos-team/nginx:1.10
```

终端显示如下信息,表明 push 镜像成功:

to a repository [registry.cn-north-1.hwclouds.com:443/aos-team/nginx] 6d6b9812c8ae: Pushed 695da0025de6: Pushed fe4c16cbf7a4: Pushed



1.10: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948

返回系统,在容器镜像仓库的"镜像详情"页面,执行刷新操作后可查看到对应的镜像信息。

7.1.3 上传私有镜像(内网)

"创建容器应用"或"通过编排创建堆栈(若堆栈中有容器应用)"时,均需要先上传容器镜像。上 传成功后,会在镜像仓库显示已上传成功的镜像。本章节以 ng inx:1.10 镜像为例,介绍如何上传本地镜 像到 ServiceStage 镜像仓库。

说明: 在使用 Docker 客户端上传镜像的过程中,如果出现网络异常断开或软件仓库被异常重启的情况, 是由于 Docker 客户端,其不会继续向软件仓库发送上传镜像的请求,从而导致 Docker 客户端卡机无法退 出。在此情况下可以通过重启 docker daemon 的方式解决。

前提条件:

- 1、准备一台已安装 Docker 客户端的虚拟机,且 Docker 版本为 1.11.2;
- 2、该虚拟机需要能联通 ServiceStage 的内部网络;
- 3、已获取 2.1.2 上传 AK/SK 证书中的 AK/SK;
- 4、请确保镜像的正确性,能够成功后台启动;

操作步骤:

- 1. 创建镜像仓库:
 - 1) 登录 ServiceStage,单击菜单栏的【仓库】【镜像仓库】;
 - 2) 在右侧页面中,单击【创建镜像仓库】;
 - 3) 设置如下参数:

参数	参数说明
*仓库名称	容器镜像仓库名称,命名必须唯一。本例配置为: nginx。
	说明:上传镜像时创建的镜像仓库名称应与上述第三点中打包镜像的镜像仓 库名称保持一致。
	例如,上述第三点中打包的镜像为 nginx. tar, 则创建镜像仓库时镜像仓



参数	参数说明
	库的名称应为 nginx。
*类型	容器镜像仓库类型,默认为公有。
	• 公有:任何租户、用户和非登录用户均可以下载。• 私有:仅用户本人可用。
*仓库分类	根据镜像类型选择对应仓库分类。
描述	容器镜像仓库的描述信息。

- 4) 配置完成后,单击【创建镜像仓库】;
- 5) 单击【镜像仓库详情】,进入仓库详情页面;
- 6) 在【Pull/Push 指南】页面,可查看本镜像仓库的地址,云主机可使用内网地址;

2. 获取仓库访问权限文件:

- 1) 在【Pull/Push 指南】页面,单击【生成 docker login 指令】;
- 2) 可查看到生成的 docker log in 指令;
- 3) 单击 ^(一),复制 docker login 指令;
- 3. 授权 Docker 客户端访问仓库的权限:
 - 1) 以 root 用户登录 Docker 所在的虚拟机;
 - 2) 将步骤 2.3 中拷贝到虚拟机中,执行 docker login 命令。例如执行如下命令:

```
docker login -u cn-north-1@5HPBFFCGOAIQMEES0EVA -p
a501765e5e2edd27b86668268e82d63fdb99298c8935f34e745e6a7cfc56c926 10.175.11.161:20202
```

其中 10.175.11.161:20202 为镜像仓库内网地址源

- 4. 给 ng inx:1.10 镜像打标签:
 - 1) 查看镜像 ID:

docker images

2) 为镜像打标签

sudo docker tag [IMAGE ID] [本镜像仓库地址]/[租户名]/[仓库名]:[版本号]



样例如下:

sudo docker tag nginx:1.10 10.175.11.161:20202/aos-team/nginx:1.10

其中,10.175.11.161:20202 为镜像仓库内网地址,aos-team 为租户名,nginx 为仓库名,1.10

为版本号。

5. Push 镜像至镜像仓库:

sudo docker push [tag]

样例如下:

sudo docker push 10.175.11.161:20202/aos-team/nginx:1.10

终端显示如下信息,表明 push 镜像成功:

docker push 10.175.11.161:20202/aos-team/nginx:1.10The push refers to a repository [10.175.11.161:20202/aos-team/nginx]

6d6b9812c8ae: Pushed

695da0025de6: Pushed

fe4c16cbf7a4: Pushed

1.10: digest: sha256:eb7e3bbd8e3040efa71d9c2cacfa12a8e39c6b2ccd15eac12bdc49e0b66cee63 size: 948

返回系统,在容器镜像仓库的"镜像详情"页面,执行刷新操作后可查看到对应的镜像信息。

7.1.4 获取镜像仓库地址

1. 在 servicestage 页面, 点击【仓库】【镜像仓库】:

2. 单击待查看镜像的"仓库名称";

3. 在版本页签下,在待获取镜像地址的版本后面单击 ⊡,复制镜像仓库地址;

1) 内网镜像 Pull 地址: 同个 vpc 内能访问的镜像地址;

2) 外网镜像 Pull 地址: 互联网能访问的镜像地址;

7.1.5 更新镜像仓库

1. 在 servicestage 页面, 点击【仓库】【镜像仓库】:



- 2. 单击待查看镜像的"仓库名称";
- 3. 在【镜像详情】页面您还可以执行中操作:

操作	操作说明
编辑镜像仓库	1、单击页面右上角的【编辑】;
	2、修改镜像仓库的 "类型"、"仓库分类"、"描述"。
	3、单击【确定】;
删除镜像仓库	单击页面右上角的【删除】,删除已创建的镜像仓库;
删除镜像	单击待删除镜像后的【删除】,删除单个镜像。您还可以选择多 个镜像,单击"批量删除"一次性删除多个镜像;

7.2 软件仓库

软件仓库包括:

1、我的软件:本租户创建的软件仓库,包含公有和私有软件仓库;

2、公共软件:所有本租户和其他租户的公有软件仓库;

7.2.1 概念

软件仓库提供软件包管理功能,用户在创建或物理机应用前,需要将应用所需的软件包上传到软件仓 库。

7.2.2 创建软件仓库

- 1. 在 servicestage 页面,点击【仓库】【软件中心】;
- 2. 在右侧页面中,单击【创建软件仓库】;
- 3. 配置软件仓库相关信息,其中带 "*" 标志的参数为必填参数:

参数	参数说明
*仓库名称	软件仓库名称,命名必须唯一。
*类型	软件仓库所属类型,默认为公有。
	• 公有:任何租户、用户和非登录用户均可以下载。



参数	参数说明
	• 私有: 仅当前租户或租户下的用户可用。
描述	软件仓库的描述信息。
版本	软件仓库支持添加多个版本。
	1、单击【添加版本】;
	2、输入【版本号】和【版本描述】;

4. 配置完成后,单击【创建】;

5. 单击【软件仓库详情】,可查看到新建的软件仓库

7.2.3 上传软件包

上传的软件包必须满足如下要求,否则上传软件包会失败:

1、软件包的命名必须符合如下要求:字符串,长度1[~]255,由大小写字母、数字、点、下划线和中划线 "-"组成,且必须以大小写字母或数字开头。

2、对压缩后的总文件大小和压缩包内文件数量不做限制, 解压后单个文件大小不能超过 3G;

3、5.3.4 创建虚机应用章节中使用的软件包,需要满足如下格式:软件包必须为压缩包,且后缀名称为 tgz,tar.gz 或 zip;软件包中必须存在可执行的脚本或命令行工具,在定义应用的生命周期时需要使用;

前提条件:

已获取 2.1.2 上传 AK/SK 证书中的 AK/SK。

操作步骤:

- 1. 登录 ServiceStage, 单击菜单栏的【仓库】【软件仓库】;
- 2. 在【我的软件】页签下,单击待上传的软件仓库,进入软件仓库详情页面;
- 3. (可选)若创建软件仓库时未添加版本,或需要增加其他版本,需要先添加版本再上传文件;

1) 单击右上方【添加版本】;



2) 输入"版本号"以及"描述", 单击【创建】:

4. 在对应的版本后,单击【上传文件】。上传软件包有如下两种方式;

说明:软件包支持格式:图片格式 (png、jpg、jpeg、gif、bmp)、压缩包格式 (gz、tgz、zip、 rar、tar、rpm)、文件格式 (rtf、doc、docx、txt、md、sh、ps1)、网页格式 (htm、html、mht) 以及"不带扩展名的二进制文件"。

1) 方式一:

- A、单击【添加文件】;
- B、选中本地待上传的软件包后,单击【打开】;
- C、设置软件包的参数,参数均为可选项;

参数	参数说明
覆盖现有文件	若选择"是",将会覆盖该版本下的原有的相同名称且 相同路径的软件包。
解压文件	若选择"是",将会将压缩包进行解压。
	说明:解压文件会同时保存压缩包和解压文件在同一路 径下。
文件路径	输入该软件包需要存放的路径,该路径为软件仓库的虚 拟路径,默认为根目录。设置文件路径便于用户查看对 应的软件包,便于管理。

D、配置完成后,单击【创建镜像仓库】。目前仅支持一次性上传10个软件包;

E、软件包选择完成后,单击【开始上传】;

F、输入 AccessKey 和 SecretKey,单击【确定】。进度显示为"上传完成",表明软件包上传成

功;

2) 方式二:



A、拖拽待上传的软件包到页面的^{将文件拖到这里}处;



B、设置软件包的参数;

参数	参数说明
覆盖现有文件	若选择"是",将会覆盖该版本下的原有的相同名称且 相同路径的软件包。
解压文件	若选择"是",将会将压缩包进行解压。
	说明: 解压又件会同时保存压缩包和解压又件在同一路 径下。
文件路径	输入该软件包需要存放的路径,该路径为软件仓库的虚 拟路径,默认为根目录。设置文件路径便于用户查看对 应的软件包,便于管理。

C、重复上述操作上传其他软件包;

D、配置完成后,单击【创建镜像仓库】。目前仅支持一次性上传10个软件包;

E、软件包选择完成后,单击【开始上传】;

F、输入 AccessKey 和 SecretKey,单击【确定】。进度显示为"上传完成",表明软件包上传成功;

7.2.4 软件仓库管理

软件仓库创建完成后,在"软件仓库详情"页面您还可以执行下表中的操作:

操作	说明		
编辑软件仓库	单击【编辑】,可修改软件仓库的类型(公开或私有)和 描述。		
添加版本	1、单击【添加版本】,可为软件仓库添加新的版本。		
	2、输入"版本号"和"描述"。		
	3、单击【创建】。		
删除软件仓库	单击【删除】,删除已创建的软件仓库。		
	说明:删除软件仓库前请先删除所有的版本。		
删除软件包	单击待删除软件包后的【删除】,删除某个软件包。您还 可以选择多个软件包单击【批量删除】,一次性删除多个		



操作	说明
	软件包。

1、软件包的命名必须符合如下要求:字符串,长度 1[~]255,由大小写字母、数字、点、下划线和中划 线 "-"组成,且必须以大小写字母或数字开头。

2、对压缩后的总文件大小和压缩包内文件数量不做限制, 解压后单个文件大小不能超过 3G;

3、5.3.4 创建虚机应用章节中使用的软件包,需要满足如下格式:软件包必须为压缩包,且后缀名称为 tgz,tar.gz 或 zip;软件包中必须存在可执行的脚本或命令行工具,在定义应用的生命周期时需要使用;

前提条件:

已获取 2.1.2 上传 AK/SK 证书中的 AK/SK。

操作步骤:

- 1. 登录 ServiceStage, 单击菜单栏的【仓库】【软件仓库】;
- 2. 在【我的软件】页签下,单击待上传的软件仓库,进入软件仓库详情页面;
- 3. (可选)若创建软件仓库时未添加版本,或需要增加其他版本,需要先添加版本再上传文件;
 - 1) 单击右上方【添加版本】;
 - 2) 输入"版本号"以及"描述",单击【创建】:
- 4. 在对应的版本后,单击【上传文件】。上传软件包有如下两种方式;



8 系统管理

8.1 证书和秘钥管理

证书和秘钥管理服务提供了证书申请、删除和秘钥查看功能。

8.1.1 秘钥列表

秘钥主要用来加解密客户的敏感数据。您可以通过以下操作查看秘钥。

- 1. 单击菜单栏的【设置】【证书和秘钥管理】;
- 2. 在秘钥列表中即可查看;

8.1.2 证书列表

为保障用户更高安全级别,建议提前申请证书,用于应用挂载 Sercret 信息

- 1. 单击【申请证书】;
- 2. 输入申请证书相关参数,其中带 "*" 的为必填;

参数	参数说明
CA 名称(根证书名称)	保持默认,不可修改。
域名	输入域名,例如 www. name. com。
IP 地址	应用所在节点的浮动 IP。
*证书 Secret 名称	为证书 Secret 取个名字,建议能展示该证书的使用用途。
角色	高阶参数,普遍情况下仅校验域名和 IP 地址。为实现更安全 操作,可设置为某个角色使用,例如角色为 te_opr。

3. 单击【确定】;



8.2 租户资源管控

8.2.1 创建资源空间

使用 ServiceStage 前,用户需要创建资源空间。创建完成后,系统将会自动为用户分配一定的资源 配额。若系统分配资源不足,用户还可以申请更多资源配额。具体步骤请参见 2.1.5 创建资源空间。

8.2.2 申请资源配额

当申请的资源配额不足时,可通过申请资源配额操作,申请更多配额。具体步骤请参见 8.3 (可选) 申请资源配额。

8.3 (可选)申请资源配额

当 2.1.6 创建资源空间申请的资源配额不足时,可通过申请资源配额操作,申请更多配额。

1. 在主菜单栏中选择【设置】【租户资源管控】;

2. 在全局配置下,单击【申请资源配额】;

实战教程

9.1 概述

本章节通过实战教程,教您使用 ServiceStage 的相关功能。

1、以"搭建 WordPress 网站"为例,介绍如何使用"创建容器应用"功能。"创建容器应用"通常适用于比较简单的应用,此类应用组件较少,且关系简单;

2、以"通过编排创建 GuestBook 应用"为例,介绍如何使用"编排"功能创建堆栈。使用"编排" 功能创建应用,通常适用于较为复杂的应用,组件较多,且关系复杂。

3、以开发部署 helloworld 微服务为例,介绍如何通过 ServiceStage 微服务框架开发部署已拆分好的微服务,并对微服务进行运维治理操作。ServiceStage 提供贯穿开发、部署、运行时保障、治理的



天翼 e Cloud

端到端微服务解决方案,帮助用户以最低成本、最安全的方式进行微服务的开发和治理。

9.2 搭建 WordPress 网站

9.2.1 概念

本章以 WordPress 个人博客网站为例,介绍如何通过 ServiceStage 的"创建容器应用"搭建 WordPress 网站。创建容器应用适用于比较简单的应用,这类应用组件较少,且关系简单。

例如 WordPress 由 "MySQL 数据库、WordPress 应用层"两个组件组成,这两个组件运行在两个 docker 容器中:

1、MySQL 数据库:使用平台内置的 MySQL 镜像创建一个数据库服务,用来存放 WordPress 服务的数据。

2、WordPress 服务: 使用平台内置的 WordPress 镜像创建一个 WordPress 服务, 用来组织内容。

基本流程:



9.2.2 创建 MySQL 数据库

首先,用户需要基于平台内置的 MySQL 镜像来创建一个 MySQL 数据库容器应用,用来存放 WordPress 服务的数据。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心, 切换到杭州节点;
- 3. 单击 选择 (企业中间件) (微服务云应用平台);



4. 单击【申请使用】;

e Cloud

- 5. 单击【应用】【应用管理】;
- 6. 在右侧页面中,选择应用需要部署到的集群;
- 7. 在右侧页面中,单击【创建应用】;
- 8. 单击容器应用下的【创建】;
- 9. 选择【无状态应用】;
- 10. 配置应用基本信息,其它参数采用默认值:

参数	参数说明	
所在集群	选择该应用所部署的集群。	
应用名称	新建容器应用的名称,命名必须唯一。配置为:mysql	

- 11. 配置完成后,单击【下一步】;
- 12. 配置以下应用相关参数,其它参数采用默认值;
 - 1) 在【容器】下, 单击【选择镜像】, 选择 【mysq1】;
 - 2) 设置内存分配: 0.5G, 限制: 1GB;
 - 3) 设置 CPU 分配: 0.5 核, 限制: 2 核;
 - 4) 通过环境变量配置数据库信息:
 - A、单击【添加】;
 - B、输入环境变量名称和变量值。每次添加完成,均需要单击【确认】;

变量名称	解释	变量值
MYSQL_USER	数据库用户名	mysql
MYSQL_PASSWORD	数据库密码	mysql
MYSQL_DATABASE	指定镜像启动时创建的数据库名称	wordpress
MYSQL_ALLOW_EMPT Y_PASSWORD	 yes:允许 root 用户以空白密码启动容器。 no:不允许 root 用户以空白密码启动容器。 说明:不建议设置为 yes,因为这将使您的 MySQL 实例完 	no



变量名称	解释	变量值
	全无保护,从而允许任何人获得完整的超级用户访问权限。	

- 5) 单击【保存】;
- 13. 配置完成后,单击【下一步】;
- 14. 配置服务发布参数,将 MySQL 设置为可被 WordPress 访问的服务:
 - 1)发布为服务:选择"是";
 - 2) 设置服务参数:
 - A、服务名称:用户可自行定义,如 mysql;

B、服务类型: 配置为 NodePort。NodePort 主要适用于内网调试, 若需要外网正式访问, 建议选择 LoadBalancer, 服务发布的具体说明请参见 11.16 如何配置服务发布参数?

- C、应用端口: 3306(本例中 MySQL 镜像的容器监听端口为 3306);
- D、服务端口:保持为空,系统会自动分配端口。默认范围为"30000-32767";

E、协议: TCP;

- 3) 单击【确认】;
- 15. 配置完成后,单击【下一步】;
- 16. 确保信息配置无误后,单击【创建】;
- 17. 单击【应用详情】,进入应用详情页面。待应用状态为"运行中",表示应用创建成功(应用状态 不会实时更新,请按 F5 刷新);

9.2.3 创建 WordPress 服务

WordPress 服务创建完成后,用户即可通过浏览器访问 WordPress 博客网站,网站数据将存放在 MySQL 数据库中。

前提条件:

1、MySQL 数据库已创建成功;



2、 "WordPress 服务镜像"已上传至镜像仓库(本例中镜像已上传);

操作步骤:

- 1. 单击【应用】【应用管理】;
- 2. 在右侧页面中,选择应用需要部署到的集群。需要和 MySQL 数据库部署在一个集群中;
- 3. 在右侧页面中,单击【创建应用】;
- 4. 单击容器应用下的【创建】;
- 5. 选择【无状态应用】;
- 6. 配置应用基本信息,其它参数采用默认值:

参数	参数说明
所在集群	选择该应用所部署的集群。
应用名称	新建容器应用的名称,命名必须唯一。配置为:wordpress

- 7. 配置完成后,单击【下一步】;
- 8. 配置以下应用相关参数,其它参数采用默认值;
 - 1) 在【容器】下,单击【选择镜像】,选择【wordpress】;
 - 2) 设置内存分配: 0.5G, 限制: 1GB;
 - 3) 设置 CPU 分配: 1 核, 限制: 2 核;
 - 4) 通过环境变量配置数据库信息:
 - A、单击【添加】;
 - B、输入环境变量名称和变量值。每次添加完成,均需要单击【确认】;

变量名称	解释	变量值
WORDPRESS_DB _HOST	9.2.2 创建 MySQL 数据库中创建的 mysql 应用名称 及容器端口	mysql:330 6
WORDPRESS_DB _USER	MySQL 容器环境变量中用户名的值	mysql
WORDPRESS_DB	MySQL 容器环境变量中密码的值	mysql





变量名称	解释	变量值
_PASSWORD		

- 5) 单击【保存】;
- 9. 配置完成后,单击【下一步】;
- 10. 配置服务发布参数,将 wordpress 设为外网可访问的服务:
 - 1)发布为服务:选择"是";
 - 2) 设置服务参数:
 - A、服务名称:用户可自行定义,如wordpress;

B、服务类型: 配置为 NodePort。NodePort 主要适用于内网调试, 若需要外网正式访问, 建议选择 LoadBalancer, 服务发布的具体说明请参见 11.16 如何配置服务发布参数?

- C、应用端口: 80(本例中 WordPress 镜像的容器监听端口为 80);
- D、服务端口:保持为空,系统会自动分配端口。默认范围为"30000-32767";

E、协议: TCP;

- 3) 单击【确认】;
- 11. 配置完成后,单击【下一步】;
- 12. 确保信息配置无误后,单击【创建】;
- 13. 单击【应用详情】,进入应用详情页面。待应用状态为"运行中",表示应用创建成功(应用状态 不会实时更新,请按 F5 刷新);
- 14. 在应用详情页面中,访问地址后的【IP:端口】是应用被访问的入口信息;

9.3 开发部署 Hello World 微服务

9.3.1 概念

当传统应用微服务化后,通常会带来一定的管理复杂性,比如微服务之间的调用发现、访问效率、运 维容错等。ServiceStage 提供贯穿开发、部署、运行时保障、治理的端到端的微服务解决方案,帮助用户 以最低成本,最安全的方式进行微服务的开发和管理。





本节以 hello world 为例,介绍微服务项目创建、开发、部署、治理的全流程操作指南。本例中, hello world 已拆分为两个微服务: "服务端"和"消费端"。

1、服务端 (provider): 提供了 say hello 的接口;

2、消费端 (consumer): 需要调用"服务端"提供的 say hello 的接口;

9.3.2 创建微服务

本节介绍创建微服务工程,协助用户自动生成一个基于微服务框架的开发环境。

- 1. 登录天翼云 http://www.ctyun.cn;
- 2. 选择控制中心, 切换到杭州节点;
- 3. 单击 选择 【企业中间件】 【微服务云应用平台】;
- 4. 单击【申请使用】;
- 5. 单击【微服务】【微服务开发】,进入创建本地工程页面;
- 6. 在创建本地工程页面进行以下操作:
 - 1)选择 SDK (Software Development Kit) 版本;
 - 2) 填写微服务基本信息, 配置项说明如下表所示:

参数	参数说明
Group	对应源码 maven 工程的 Group ID。
Artifact	对应源码 maven 工程的 Artifact ID。
Java 包名称	对应源码工程中的 package 名称。
版本号	微服务版本号。
Rest Path	对应发布接口的 URL,用户可自定义填写,如"/sayhello"。
名称	为微服务取个名字,如创建 Hello World 的服务消费端,名称为"consumer"。
开发风格	系统提供了透明 RPC、spring-mvc、jax-rs 三种开发模式供选择, 缺省配置透明 RPC。



 3)单击【创建并下载】,完成微服务工程的创建。微服务工程创建以后,系统会自动生成基于微服务 框架的开发工程包并保存到本地;

7. 按照上述步骤, 创建 HelloWorld 的服务端, 名称为 "provider";

9.3.3 开发微服务

本章介绍如何迅速开发消费端代码以及服务端代码。provider和 consumer两个微服务工程创建以后, 分别生成基于微服务框架的开发工程包,将源码下载至本地,进行微服务应用开发。

1. 在本地搭建微服务开发环境,导入9.3.2 创建微服务中生成的微服务开发工程包;

- 2. 进行服务端开发;
- 3. 进行消费端开发;

说明: provider 服务端样例发布了/helloworld/sayhello 接口, consumer 样例作为服务消费者来调用该接口。

9.3.4 部署微服务

将 provider 和 consumer 两个微服务打包成镜像文件,打包过程参考 Docker 官方文档。制作镜像的时候,不要遗漏微服务依赖的 jar 包文件。可以使用 maven 插件,将依赖的 jar 包文件放在独立的 lib 目录,也可以参考 spring boot 插件,将依赖的 jar 包文件制作为一个可执行包。

- 1. 上传镜像到仓库,请参考 7.2.3 上传私有镜像(内网);
- 2. 创建微服务应用:
 - 1) 单击【应用】【应用管理】;
 - 2) 在右侧页面中,先选择集群;
 - 3) 在右侧页面中,单击【创建应用】;
 - 4) 单击容器应用下的【创建】;
 - 5)选择【无状态应用】;
 - 6) 配置应用基本信息,其它参数采用默认值:



参数	参数说明	值
所在集群	应用所在集群。	在下拉框中选择。
应用名称	新建容器应用的名称,命名必须唯一。	provider、consumer。
应用组	您可以将某类应用放到同个应用组中,实现应用的批 量管理(如批量启停、删除)。 如果应用组名称不存在,用户输入应用组名称后,会 自动创建一个新的用户组。	-
	如果应用组名称已创建,用户在下拉框中选择。	
描述	输入对应描述。	_
标签	标签以 Key/value 键值对形式附加到各种对象上(如 应用、节点、服务等)。 标签定义了这些对象的可识别属性,用来对它们进行 管理和选择。	_

- 3. 定义卷信息:
 - 1) 单击【展开高级设置】,系统显示卷定义页面;
 - 2) 单击【卷定义】下的【添加】;
 - 3) 输入卷名称, 用户可自定义填写;
 - 4) 卷类型选择 HostPath;

说明: HostPath 为在容器上挂载宿主机的文件或目录。通常用于容器应用程序生成的日志文件需要永久保存,或需要访问宿主机上 Docker 引擎内部数据结构的容器应用。

- 5) 单击【添加配置】, 输入主机路径, 如/var/lib/docker;
- 6) 单击【确认】保存主机路径;
- 7) 单击操作列【确认】保存卷定义;
- 4. 单击【下一步】进行应用配置:
 - 1) 在【容器】下,单击【选择镜像】,选择已准备好的镜像 "provider.tar";
 - 2) 设置内存分配为 0.5GB, 限制为 1GB;



- 3) 设置 CPU 分配为 0.5 核, 限制为 0.5 核;
- 4) 在【数据卷】页签,单击【添加】,参考下表填写配置信息;

配置项	值
卷名称	在下拉框中选择 <u>步骤3</u> 中创建的卷。
容器路径	数据卷挂载到容器上的路径,请不要挂载在系统目录下,如 "/"、"/var/run"等,否则会导致容器异常。
是否只读	"是":只能读容器路径中的数据卷。
	"否":可修改容器路径中的数据卷,容器迁移时新写入的数据 不会随之迁移,会造成数据丢失。"卷类型"为"Secret"时, 数据为只读。
卷策略及日志策略	当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs。请 参见 11. 6 如何设置容器应用数据卷? 的内容。
操作	点击【确认】,保存配置信息。
	点击【取消】,删除配置信息。

- 5) 单击【确认】保存配置;
- 6) 单击【保存】;
- 5. 单击【下一步】进行服务发布配置;
 - 1)发布为服务:选择"是",将应用设为可被访问的服务;
 - 2) 设置服务参数:请参见 11.16 如何配置服务发布参数? 配置服务发布参数。

- 6. 配置完成后,单击【下一步】;
- 7. 确保信息配置无误后,单击【创建】;
- 单击【应用详情】,进入应用详情页面。待应用状态为"运行中",表示应用创建成功(应用状态 不会实时更新,请按 F5 刷新);

9.3.5 治理微服务

微服务部署完成后,运维人员可以在服务库中选择需要治理的微服务,根据需要选择合适的治理策略。

³⁾ 单击【确认】;



服务治理提供了熔断、容错、限流、降级等高级服务治理能力,最大限度保障服务的可用性。

前提条件:微服务应用已部署到平台中且为运行状态。

操作步骤:以下以"降级"为例,介绍如何进行微服务的治理。降级是容错的一种特殊形式,通过降级限制非核心服务的可用性,保证核心服务的正常运行。

- 1. 登录 ServiceStage, 单击【微服务】【微服务管理】;
- 2. 在微服务管理页面,单击【微服务治理】,进入微服务治理页面;
- 3. 单击需要治理的微服务,展示监控面板和治理面板;
- 4. 在微服务治理面板,对该微服务进行降级治理:
 - 1) 单击【降级】,展开降级详情页;
 - 2) 单击【新增】,进入降级策略配置页面;
 - 3) 在降级策略配置页面选择合理的策略,降级策略配置项如下表所示:

配置项	说明
降级对象	选择需要降级的微服务。
降级方式	手动降级。
	容错降级。
降级策略	throwexception: 降级时直接抛出异常。
	returnnull: 降级时返回 null 值。

5. 单击【确定】保存设置;



10 常见问题

10.1 什么是节点?

每一个节点对应一台虚拟机,容器应用运行在节点上。节点上运行着代理程序(kubelet),用于管理 节点上运行的容器实例。节点规格最小是 CPU 为 1core,内存为 2048MB;最大是 CPU 为 32core,内存为 128GB。

10.2 对于创建节点失败期间回滚的资源,我是否需要付费?

是。无论节点是否创建成功,创建节点期间使用的云主机资源均需收费,该过程会根据公有云主机的 统一收费策略来收取费用。

10.3 什么是环境变量?

环境变量是指容器运行环境中设定的一个变量,您可以在创建容器模板时设定不超过 30 个的环境变量;环境变量可以在应用部署后修改,为应用提供极大的灵活性。

10.4 堆栈、编排、模板、设计包分别是什么?

概念	概念说明
编排	编排是指对组成堆栈的元素(Element)按照一定的次序进行处理的过程。通 过定义应用组件、对资源的需求、依赖的服务、元素之间的关系(拓扑)等, 完整地描述应用本身以及所依赖的服务、资源,最后交由 ServiceStage 平台 完成部署。
设计包	运维人员对应用的拓扑、生命周期管理计划进行设计,并输出应用模板(也 可称为应用设计包)。
设计器	ServiceStage 还提供了图形化设计器工具,用户可通过拖拽方式完成复杂应用的编排及拓扑设计,可以保存成模板,使用模板创建堆栈,简化应用部署 难度,提升效率。



概念	概念说明
模板	模板是对堆栈的描述,包括基于应用模型的堆栈拓扑定义、堆栈生命周期描述、运行时资源描述、软件组件描述等。模板通过设计包创建而来,本质与 设计包相同。
堆栈	由应用、服务、资源等元素组成的一个部署实例,平台将相关编排元素通过 "堆栈"进行集中管理。

编排、设计包、模板、堆栈、设计器的相互关系可参考下图:



10.5 "有状态容器应用"和"无状态容器应用"有什么 不同?

1、有状态容器应用:应用有多个实例,且每个实例承担不一样的功能,且每个实例间有依赖和前后 启动顺序关系。通常用于有数据持久化存储要求的应用。

2、无状态容器应用:应用实例间互相不依赖,任意一个 Web 请求完全与其他请求隔离,当请求端提 出请求时,请求本身包含了相应端为相应这一请求所需的全部信息。无状态容器应用更易实现可靠性以及 可伸缩性。

10.6 什么是调度策略?

根据应用的部署特性,平台把应用拆解成最小的部署实例,应用调度器实时监控应用的实例信息,当 发现有新的 pod 产生需要调度的时候就会对集群中所有的剩余资源(计算、存储、网络)进行计算,最后 得出最适合的调度目标节点。

平台支持多种调度算法,包括:应用和节点之间的亲和性调度、应用间的亲和/反亲和调度。

10.7 什么是亲和与反亲和?

在应用没有容器化之前,原先一个虚机上会装多个组件,进程间会有通信。但在做容器化拆分的时候, 往往直接按进程拆分容器,比如业务进程一个容器,监控日志处理或者本地数据放在另一个容器,并且有 独立的生命周期。这时如果他们分布在网络中两个较远的点,请求经过多次转发,性能会很差。

1、亲和性可以实现就近部署,增强网络能力实现通信上的就近路由,减少网络的损耗。
2、反亲和性主要是出于高可靠性考虑,尽量分散实例,某个节点故障的时候,对应用的影响只是 N 分 之一或者只是一个实例。

10.8 什么是 TOSCA 模板?

e Cloud

TOSCA(Topology and Orchestration Specification for Cloud Application)是开放标准联盟 OASIS 管理的独立技术委员会之一,其发布的云应用拓扑及编排描述规范(简称 TOSCA 规范)目标是规范多厂商 云应用生命周期管理流程。ServiceStage 应用拓扑模型如下图所示。缓存实例的数据被删除之后,能否找 回?



元素	说明
Resource	资源类节点,如 VM、物理机、容器等。
AppGroup	云应用组,由一个或多个云应用组成,可整体进行生命周期操作,如部 署 <i>、</i> 升级。也可对应到客户产品、业务系统/子系统等。
Application	运行在资源上的云应用,是对最小可部署对象的一种描述。其中, MicroService 也是一种 Application。
SoftwareComponent	软件组件, 云应用组件的组成部分, 即软件包。也可以作为 Application 的属性, 是可选节点。
Service	应用所依赖的服务。服务是对按需取用的功能对象的一种描述。
Depends0n	节点间的依赖关系,决定了创建顺序,为基础关系。
HostedOn	只能用于 Application 与 Resource 之间,表示应用运行在资源上。
ConsistsOf	表示组合关系。例如, AppGroup 由 Application 组成。
ConnectsTo	表示调用或连接关系。例如, Application 和 Application 之间, 资源与 资源之间。
PackageConsistsOf	应用和软件组件之间的组合关系 。





元素	说明
BindsTo	应用绑定服务。
LinksTo	网卡(Port)连接到网络 。

10.9 如何解决堆栈无法删除的问题?

问题描述:删除堆栈时,堆栈状态一直显示为"删除中",最后提示超时。堆栈状态显示"异常",操作状态为"卸载失败",再次删除依然失败。查看堆栈执行日志,出现"time out"相关字段。

解决方法:

1、在堆栈详情页面,单击【堆栈元素】【应用】;

2、在【应用】中,单击 > 展开应用组;

Service Stage	一 概览 应用 服务 编排 微服务 运维 仓库 设置	
(品编排)。	编辑 > 堆线 > 堆线洋槽	
 ◎ 堆栈 □ 模板 □ 设计器 	liny1 環想 模板: hostname-4suk2h0k4p8 描述: 価縮 升级 更多・	创建时间:2017-09-07 16:13:53
	堆栈元素 参数配置 执行日志 拓扑图	
	efault-appGroup	

3、单击应用名称,系统跳转到应用详情页面;

4、单击【更多】【删除】,删除应用。若堆栈包含多个应用,需将应用均删除成功,才能确保堆栈成功删除;

天 巽 e Cloud				10 常见间	可题
□◎ 应用 • 应用	> 应用管理 > 应用详情 > 实例				
	镜像:				
88 应用管理	应用类型 :			类型: 容器	
☑ 任务管理 ∨	应用组:			应用版本:	
88 配置项	实例个数(正常/全部):			实例端口:	
	访问地址:			标签:	
	创建时间:				
	描述:				
	启动 停止 移	至应用组 更多 - 查看卷定义			
	实例 升级 服务 事件				
	実例名称 状态 ~	田除 CPU(申请量) 内存(申请量)	所在节点	运行时长	启动时间

5、(可选)若步骤 4 中删除应用失败,请根据以下步骤删除应用对应的 pod;

1)选择菜单栏的【应用】【资源管理】【其他】;

2)选择类型为 Pod,在资源名称中输入待删除的应用名称,单击 Q 搜索;

应用	0	立用 > 按读管理 > 其他					
③ 资源管理	^	宣看、更新、创建、删除集群的其他资源信息。 了解 更	\$				
 ・集群 ・ 节点 		集群: test ~ 命名空间: d	efault	~			
・其他		+ 创建资源 删除			全部	✓ 资源名称	9 <i>C</i>
• 命名空间			类型 ~	标签	全部	操作	
88 应用管理		export2-3943164700-bn34l	Pod	app export2	Secret	.9 GMT 更新 删除	
 任务管理 記 配置项 	~	export1-4021885511-dx71j	Pod	app export1	Pod 2017/11/03 00:0	/Z:28 GMT 更新 删除	
		redis1-1574368828-jqqr3	Pod	app redis1	2017/10/31 19:1	.2:24 GMT 更新 删除	
		redis-4138923531-wn1lb	Pod	app redis	2017/10/31 19:0	/7:34 GMT 更新 删除	

3) 删除应用对应的 pod。由于一个应用通常由一个或多个 Pod 组成,应用对应的 Pod 名称为:应 用名称-动态生成的字母;

5、单击【编排】【堆栈】;

6、在待删除堆栈后,单击【更多】【删除】,删除堆栈;



10.10 如何解决上传软件包失败的问题?

问题描述:在上传软件包时,在软件包满足系统要求的情况下,界面依然出现如下提示。上传软件包 后,系统提示"无权限访问,请联系管理员"。

解决方法:

1、在谷歌浏览器中,单击【更多工具】【清除浏览数据】;



2、在弹出的清除浏览数据窗口中,保持默认勾选,单击【清除浏览数据】;

10.11 如何收藏导航栏菜单?

若经常使用某个功能,可参照本章节收藏该功能的菜单。本章节以收藏【应用】【资源管理】为例进 行说明。系统最多可以显示最近的 29 个收藏。

1、鼠标移动到菜单栏的【应用】【资源管理】;



2、资源管理后面会出现 😭, 点击收藏;

Service Stage	ហ	概览	应用	服务编排	微服务	运维	仓库	留				
	应用	1 > 资源管理	- 19	资源管理 集群			☆	88	任务管理 普通 任务			
				节点					定时任务			
③ 资源管理 ~		查看、潏	5 D. 8	其他				88	配置项			
😵 应用管理			88	应用管理								
☑ 任务管理 ∨		集群:			÷					 _		
82 配置项												



11 参数说明

11.1 如何配置容器应用卷定义?

在 Docker 的设计实现中,容器中的数据是临时的,当容器被销毁时,其中的数据也会丢失。如果需要持久化数据,可以通过使用 Docker 数据卷将持久化目录挂载到容器目录中。

ServiceStage 数据卷基于 Docker 进行了扩展,支持更加丰富的功能。帮助用户实现容器持久化数据,和 Pod 中容器的文件共享。

1、单击【展开高级设置】,系统显示卷定义页面;

2、单击【添加】;

3、选择对应的卷类型,并添加配置:

卷类型	卷类型解释	步骤说明
HostPath	在容器上挂载宿主机上的文件或目 录。通常用于: •容器应用程序生成的日志文件 需要永久保存。 •需要访问宿主机上 Docker 引擎 内部数据结构的容器应用。	 选择卷类型为"HostPath"。 、输入卷名称,用户可自定义填写。 、单击"添加配置"。 、输入主机路径,如/var/lib/docker。 、单击"确认"。 、单击操作列的"确认"。
EmptyDir	在容器分配到节点时系统自动创 建,初始内容为空。在同一个 Pod 中所有容器可以读写 EmptyDir 中 的相同文件。当 Pod 从节点上移除 时,EmpryDir 中的数据也会永久删 除。通常用于临时数据的高速存储。	 1、选择卷类型为"EmptyDir"。 2、输入"卷名称",用户可自定义填写。 3、单击"添加配置"。 4、选择"存储介质"。 默认:存储在硬盘上,适用于数据量大,读 写效率要求低的场景。 内存:存储在内存中,适用于数据量少,读 写效率要求高的场景。



卷类型	卷类型解释	步骤说明
		5、单击"确认"。
		6、单击操作列的"确认"。
Secret	用来处理敏感数据,比如密码、 Token 和密钥,相比于直接将敏感 数据配置在 Pod 的定义或者镜像	1、选择卷类型为"Secret"。 2、输入"卷名称",用户可自定义填写。
	市, secret 提供了更加安主的机制, 防止数据泄露。	3、单击"添加配置"。
	应用绑定服务时,也需选择 Secret 中的"服务访问凭证"进行绑定。	 所有凭证:在配置文件下拉框中,选择 已配置好的 Secret 文件。您也可以单击 "去创建"新建配置文件、创建配置文
	说明	件步骤请参见 3.8 创建 Secret 资源。
	 若应用需绑定服务,需确保应 用的业务逻辑中支持绑定需要 	 服务访问凭证:为应用绑定平台中的服务。当前不支持此功能。
	的服务。	4、单击"确认"。
	 创建 Secret 类型资源时, ServiceStage 会针对 Secret 信息做加密处理,建议不要在 Secret 中输入涉及个人隐私 的信息。 	5、单击操作列的"确认"。
ConfigMap	提供应用代码和配置文件的分离,	1、选择卷类型为"ConfigMap"。
	ConfigMap 用于处理应用配置参数。 用户需要提前创建应用配置,操作	2、输入"卷名称",用户可自定义填写。
	步骤请参见 5.1 创建应用配置项。	3、单击"添加配置"。
		4、在应用配置下拉框中选择需添加的配置文 件。
		5、输入配置文件所在的相对路径。
		6、单击"确认"。
		7、单击操作列的"确认"。

11.2 如何配置自定义监控?

当前仅支持监控数据输出到 prometheus(Prometheus 是一个开源的服务监控系统和时间序列数据库)。 配置完成后,可以在"运维 > 应用监控 > 指标监控 > 浏览指标"中查看到监控数据。

1、选择监控类型为 "prometheus";



- 2、输入以下数据:
- 1)上报路径:指标上报的 URL 路径,如/metrics。
- 2) 上报端口:指标上报端口,如9121。

3) 监控维度:需要监控的指标名称列表,如:

["redis_used_cpu_sys", "redis_used_cpu_user", "redis_memory_used_bytes", "redis_memory_use
d_peak_bytes", "redis_key_size", "redis_connected_clients"].

11.3 如何设置升级策略?

ServiceStage 提供了三种升级方式:

1、替换升级:先删除旧实例,再创建新实例。升级过程中业务会中断。

2 滚动升级:先安装新实例,再移除旧实例。升级过程中,业务会同时均衡分布到新老实例上,因此 业务不会中断。

3、原地滚动升级:滚动升级的一个特殊场景,即在升级过程中,将旧实例更新为新实例,新实例和 老实例所在节点保持一致。

如未配置升级策略,系统默认使用滚动升级的方式。若选择原地滚动升级,应用不支持添加弹性伸缩 策略。

参数	参数说明
最大不可用个数或比例 (可选)	升级过程中允许的最大失效实例数值,可选配置,值可以是绝 对值(5)或者是比例(10%),默认为1。
最大超出实例数或比例 (可选)	升级过程中允许超过指定实例数目的最大数值,可选配置,值 可以是绝对值(5)或者是比例(10%),默认为1。
实例可用最短时间(s)	升级时新创建的实例,达到就绪状态所需的最短时间,默认为0。
历史版本上限(个)	最多保留的历史版本数量。



ServiceStage 提供了丰富的调度策略,包括静态的全局调度策略,以及动态的运行时调度策略,用户可以根据需要自由组合使用这些策略来实现自己的需求。平台支持以下两种亲和性的调度方式。

1、应用间的亲和性:

Cloud

1) 应用间的亲和性:决定应用部署在相同或不同"主机"中。

2)应用间的反亲和性: "不同应用"或"相同应用的多个实例"部署在不同主机中。同个应用的 多个实例反亲和部署,减少宕机影响;互相干扰的应用反亲和部署,避免干扰;

2、应用与节点间的亲和性:

1) 应用与节点亲和:决定应用部署在某些特定的主机中。

2) 应用与节点反亲和:决定应用不能部署在某些特定的主机中。

亲和、反亲和	步骤说明
将应用设为亲和, 部署在相同主机	说明:在设置应用亲和/反亲和性时,您需要选择对应的拓扑域,拓扑域主要用 于限定应用的亲和性或反亲和在哪个范围内生效。
中 	kubernetes.io/hostname: 指同个节点内的亲和,即应用部署在同个节点上。
	创建应用间的亲和性有如下两种方式:
	方式一:选择应用
	1、单击【应用间的亲和性】【亲和】;
	2、单击【选择应用】,进入【选择应用(亲和)】页面;
	3、选择应用对应的拓扑域;
	4、选择待亲和的应用;
	5、设置完成后,单击【确认】按钮。系统会将您选择的应用设置为亲和,部署 到相同的节点中。
	方式二: 自定义
	说明:自定义亲和中,若亲和了系统中没有的应用或节点,调度会失败。
	1、单击【应用间的亲和性】【亲和】;
	2、单击【自定义】,进入【添加自定义(亲和)】页面;



亲和、反亲和	步骤说明
	3、选择应用对应的拓扑域;
	4、单击【添加】;
	5、输入待亲和应用的键、值后,单击【确认】;
	6、单击【确认】,系统会根据您所添加的标签(键/值)选择应用,自动将该类 应用设置为亲和,部署到相同的节点中;
将应用设为反亲	设置应用间的反亲和性有如下两种方式:
和,部署在不同主 机中	方式一:选择应用
	1、单击【应用间的亲和性】【反亲和】;
	2、单击【选择应用】,进入【选择应用(反亲和)】页面;
	3、选择应用对应的拓扑域;
	4、选择待反亲和的应用;
	5、设置完成后,单击【确认】,系统会将您选择的应用设置为反亲和,部署到 不同的节点中。
	方式二: 自定义
	1、单击【应用间的亲和性】【反亲和】;
	2、单击【自定义】,进入【添加自定义(反亲和)】页面
	3、选择应用对应的拓扑域;
	4、单击【添加】,输入待反亲和应用的键、值后,单击【确认】;
	5、单击【确认】,系统会根据您所添加的标签(键/值)选择应用,自动将该类 应用设置为反亲和,部署到不同的节点中;
将节点设为亲和	设置应用与节点间的亲和性有如下两种方式:
	方式一:选择节点属性
	1、单击【应用与节点间的亲和性】【亲和】;
	2、单击【选择节点属性】,进入【选择节点属性(亲和)】页面;
	3、选择待亲和节点的属性类型:
	操作系统:勾选某个操作系统,应用部署时会优选该操作系统;
	版本:
	4、单击【确认】;



亲和、反亲和	步骤说明
	方式二:自定义 1、单击【应用与节点间的亲和性】【亲和】;
	2、单击【自定义】;
	3、系统显示【添加自定义(亲和)】页面;
	4、单击【添加】;
	5、在"键"对应的下拉框中选择 key 后,单击【确认】;
	说明:
	此处下拉框中的取值来源于添加节点时设置的节点标签。 若多个节点,添加的标 签"键"相同,"值"不同,此时还需在"值"的下拉框中选择对应的 value;
	6、单击【确认】
将节点设为反亲	设置应用与节点间的反亲和性有如下两种方式:
和	方式一:选择节点属性
	1、单击【应用与节点间的亲和性】【反亲和】;
	2、单击【选择节点属性】,进入【选择节点属性(反亲和)】页面;
	3、选择待反亲和节点的属性:
	操作系统:勾选某个操作系统,应用部署时不会选择该操作系统;
	版本:勾选某个版本,应用部署时不会选择该版本;
	尖型采构: 勾选呆个尖型采构,应用部者时不会选择该尖型采构; 4. 单击【确认】
	1、单击【应用与节点间的亲和性】【反亲和】;
	2、单击【自定义】,进入【添加自定义(反亲和)】页面;
	3、单击【添加】;
	4、在【键】对应的下拉框中选择 key 后,单击【确认】;
	说明:
	此处下拉框中的取值来源于添加节点时设置的节点标签。 若多个节点,添加的标 签"键"相同,"值"不同,此时还需在"值"的下拉框中选择对应的 value;
	5、单击【确认】;



11.5 如何设置容器应用生命周期?

ServiceStage 提供了回调函数,在容器的生命周期的特定阶段执行调用,比如容器在停止前希望执行 某项操作,就可以注册相应的钩子函数。目前提供的生命周期回调函数如下所示:

1、启动后处理:在容器创建成功后调用该回调函数。

2、停止前处理:在容器被终止前调用该回调函数。

参数	说明
启动后处理	在容器中执行指定的命令, 配置为需要执行的命令。如需要执行的命令 如下:
	exec: command: - /bin/sh - c - /er-registy.sh registry 请在执行脚本中填写: /bin/sh -c "/er-registry.sh regisrty" 这条命令表示容器创建成功后将服务路由注册到 ER。
停止前处理	发起一个 HTTP 调用请求。配置参数如下: • 路径:请求的 URL 路径,可选项。 • 端口:请求的端口,必选项。 • 主机地址:请求的 IP 地址,可选项,默认是容器所在的节点 IP。 • 协议:请求的协议,可选项,默认为 HTTP。

11.6 如何设置容器应用数据卷?

数据卷用于实现容器持久化数据,在 Docker 设计实现中,容器中的数据是临时的,如果需要持久化数据,需要使用 Docker 数据卷挂载宿主机上的文件或者目录到容器中。通过挂载可以实现宿主机和容器之间,网络存储和宿主机之间的数据共享。

前提条件: 挂载数据卷之前, 用户需要提前添加卷类型, 详细请参见 11.1 如何配置容器应用卷定义?。 操作步骤:

1、在"数据卷"页签,单击"添加";



2、设置数据卷对应的参数:

1) 卷名称:在下拉框中选择"应用基本信息"中创建的卷。;

2) 容器路径:数据卷挂载到容器上的路径。



请不要挂载在系统目录下,如"/"、"/var/run"等,会导致容器异常。建议挂载在空目录下,若目录不为 空,请确保目录下无影响容器启动的文件,否则文件会被替换,导致容器启动异常,应用创建失败。

3)是否只读:配置为"是",只能读容器路径中的数据卷。配置为"否",可修改容器路径中的数据卷,容器迁移时新写入的数据不会随之迁移,会造成数据丢失。

4) 卷策略及日志策略:

A、配置为 none: 不支持卷策略。

B、 - 配置为 logs: 当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs。当卷策略配置为 logs 时,可支持用户自定义日志收集的路径,收集的日志在监控页面可以查看,查看日志的详细操作 请参见 6.3.2 查询日志。

参数	参数说明
none	不进行日志回滚转储。
Hour I y	每小时扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到 一个历史文件中,在日志所在目录下),并将原文件清空。
Daily	每天扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一 个历史文件中,在日志所在目录下),并将原文件清空。
Weekly	每周扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一 个历史文件中,在日志所在目录下),并将原文件清空。

3、单击"确认"。

11.7 如何设置容器应用环境变量?

环境变量是容器运行环境中设定的一个变量。可以在应用部署后修改,为应用提供极大的灵活性。



1、在"环境变量"页签,单击"添加"。

2、输入新增环境变量的名称和值后,单击"确认"。

11.8 如何进行容器应用健康检查?

健康检查是指容器运行过程中,根据用户需要,定时检查容器健康状况或是容器中应用的健康状况。 平台提供了两种健康检查的方式:

1、应用存活探针,探测应用是否已经启动:用于容器的自定义监控检查,如果检查失败,平台将删 除该应用实例,然后根据应用的重启策略来决定是否重启容器。

2、应用业务探针,探测应用业务是否已经就绪:用于监控应用是否启动完成,且准备接受请求,如 果检测到失败,则应用可用状态将被修改。此处 InitialDelaySeconds、TimeoutSeconds、ExecAction、 TCPSocketAction、HttpGetAction参数与"Liveness Probe"中相同。

参数	参数说明
应用存活探针,探测应 除该应用实例,然后相	Z用是否已经启动: 用于容器的自定义监控检查,如果检查失败,平台将删 根据应用的重启策略来决定是否重启容器。
延迟检查时间	延迟检查时间,单位为秒。例如,设置为 10,表明 10 秒后开始执行健康 检查。
超时时间	超时时间,单位为秒。例如,设置为 10,表明执行健康检查的超时等待 时间为 10 秒。若设置为 0 或不设置,默认超时等待时间为 1 秒。
命令行方式	在容器内部执行一个命令,如果该命令的退出状态码为 0,表示容器健康。 例如,用户若希望配置检测应用是否成功创建目录 foo 的探针,那么需要 在界面上做如下配置:
	if [!-d "foo"]; then exit 1 else exit 0 fi
TCP 建链方式	通过容器的 IP 地址和端口号执行 TCP 检查,如果端口能被访问,表示容器健康。
HttpGet 请求方式	通过容器 IP 地址和端口号以及路径调用 HTTP Get 方法,如果响应状态码 大于等于 200 且小于等于 400,认为容器健康。 1、单击"添加"。



参数	参数说明
	2、输入以下信息。
	路径:请求的 URL 路径,可选项。
	端口:请求的端口,必选项。
	主机地址:请求的主机 IP,可选项,默认是 Pod 的 IP。
	协议:请求的协议,可选项,默认 HTTP。
	3、单击"确认"。

应用业务探针,探测应用业务是否已经就绪:用于监控应用是否启动完成,且准备接受请求,如 果检测到失败,则应用可用状态将被修改。此处展示的参数与"**应用存活探针,探测应用是否已** 经启动"中相同。

11.9 如何配置虚机应用卷定义?

- 1、单击"展开高级设置",系统显示卷定义页面;
- 2、单击"添加";
- 3、选择对应的卷类型,并添加配置:

卷类型	卷类型定义	步骤说明
HostPath	在虚机上挂载宿主机上的文	1、选择卷类型为"HostPath"。
	仟或日求。	2、输入卷名称,用户可自定义填写。
		3、单击"添加配置"。
		4、输入主机路径,如/var/lib/docker。
		5、单击"确认"。
		6、单击操作列的"确认"。
ConfigMap	提供应用代码和配置文件的	1、选择卷类型为"ConfigMap"。
	分离, ConfigMap 用于处理应 用配置参数。用户需要提前创 建应用配置,操作步骤请参见 5.1 创建应用配置项。	2、输入"卷名称",用户可自定义填写。
		3、单击"添加配置"。
		4、在应用配置下拉框中选择需添加的配置文件。
		5、输入卷路径。
		6、单击"确认"。





卷类型	卷类型定义	步骤说明
		7、单击操作列的"确认"。
Secret	用来处理敏感数据,比如密 码、Token 和密钥,相比于直 接将敏感数据配置在进程中, Secret 提供了更加安全的机 制,防止数据泄露。 应用绑定服务时,也需选择 Secret 中的"服务访问凭 证"进行绑定。 说明 • 若应用需绑定服务,需确 保应用的业务逻辑中支 持绑定需要的服务。 • 创建 Secret 类型资源 时,ServiceStage 会针 对 Secret 信息做加密处 理,建议不要在 Secret 中输入涉及个人隐私的 信息。	 选择卷类型为"Secret"。 、输入"卷名称",用户可自定义填写。 、单击"添加配置"。 所有凭证:在配置文件下拉框中,选择已配置好的Secret文件。您也可以单击"去创建"新建 配置文件,创建配置文件步骤请参见3.8 创建Secret 资源。 服务访问凭证:为应用绑定平台中的服务。当前版本不支持。 、输入卷路径。 、单击"确认"。 、单击操作列的"确认"。

11.10 如何设置虚机应用生命周期?

生命周期脚本定义,主要针对的是进程类应用的脚本定义。进程类应用的软件组件,软件包中存在安装脚本,平台对软件包的脚本存放有建议的目录,详情请参见 12.2 软件包打包规范。

1、选中"生命周期"下相应的生命周期动作,如安装。

2. 在输入框中输入对应的脚本,如 sh scripts/install.sh。此处输入的脚本,必须和软件包中的脚本一致。

11.11 如何设置虚机应用数据卷?

数据卷是指将应用进程的某个绝对路径挂载到主机的某个路径上。通过挂载可以实现宿主机和应用进 程之间的数据共享,实现数据的持久化存储。

前提条件: 挂载数据卷之前, 用户需要提前添加卷类型, 详细请参见 11.9 如何配置虚机应用卷定义?。



操作步骤:

1、在"数据卷"页签,单击"添加"。

2、设置数据卷对应的参数。

1) 卷名称:在下拉框中选择"应用基本信息"中创建的卷。

2) 挂载路径:数据卷挂载到应用进程上的路径。请不要挂载在系统目录下,如"/"、"/var/run"等,否则会导致虚机异常。

3) 是否只读:否。

4) 卷策略及日志策略:

A、配置为 none: 不支持卷策略。

B、配置为 logs: 当选择卷名称的类型为 HostPath 时,可配置卷策略为 logs。当卷策略配置为 logs 时,可支持用户自定义日志收集的路径,收集的日志在监控页面可以查看,查看日志的详细操作请参见 6.3.2 查询日志。

参数	参数说明
none	不进行日志回滚转储。
Hourly	每小时扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到 一个历史文件中,在日志所在目录下),并将原文件清空。
Daily	每天扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一 个历史文件中,在日志所在目录下),并将原文件清空。
Weekly	每周扫描一次日志文件,如文件超过 20M,会将其转储(即压缩到一 个历史文件中,在日志所在目录下),并将原文件清空。

3、单击"确定"。

11.12 如何设置虚机应用环境变量?

环境变量是指虚机应用运行环境中设定的一个环境变量。设置环境变量有两种方式,一种是手工添加, 一种是从配置文件导入。

1、若采用从配置文件导入的方式,请事先创建好资源文件,具体方法请参见 5.1 创建应用配置项。



您也可以单击"去创建"新建资源。

- 2、手工添加方式如下:
 - A、在"环境变量"页签,单击"添加"。

B、根据实际需求增加相应的环境变量后,单击"确认"。

11.13 如何设置虚机应用健康检查?

健康检查是指应用运行过程中根据用户需要定时检查应用健康状况。平台提供了两种健康检查的方式:

1、应用存活探针,探测应用是否已经启动:用于应用的自定义监控检查,如果检查失败,平台将删除该应用实例,然后根据应用的重启策略来决定是否重启容器。

2、应用业务探针,探测应用业务是否已经就绪:用于监控应用是否启动完成,且准备接受请求,如
 果检测到失败,则应用可用状态将被修改。

参数	参数说明
应用存活探针,探测应 删除该应用实例,然危	Σ用是否已经启动: 用于应用的自定义监控检查,如果检查失败,平台将 后根据应用的重启策略来决定是否重启应用进程。
延迟检查时间	延迟检查时间,单位为秒。例如,设置为10,表明10秒后开始执行健康检查。
超时时间	表示探测超时时间,单位为秒。例如,设置为10,表明执行健康检查的 超时等待时间为10秒。若设置为0或不设置,默认超时等待时间为1秒。
命令行方式	该配置能够保证在应用所在的虚机或节点上定时执行用户配置的命令。 命令执行的结果成功与否,作为健康检查探针的最终结果返回。
	例如,用户若希望配置检测应用是否成功创建目录 foo 的探针,那么需 要在界面上做如下配置:
	if [!-d "foo"]; then exit 1 else exit 0 fi
HttpGet 请求方式	通过容器 IP 地址和端口号以及路径调用 HTTP Get 方法,如果响应状态 码大于等于 200 且小于等于 400,认为容器健康。
	1、单击"添加"。



参数 参数说明 2、输入以下信息。
2、输入以下信息。
路径:请求的 URL 路径,可选项。
端口:请求的端口,必选项。
主机地址:请求的主机 IP,可选项,默认是 Pod 的 IP。
协议:请求的协议,可选项,默认 HTTP。
3、单击"确认"。
应用业务探针,探测应用业务是否已经就绪: 用于监控应用是否启动完成,且准备接受请求, 如果检测到失败_则应用可用状态将被修改_此处_InitialDelaySeconds_TimeoutSeconds

如果检测到失败,则应用可用状态将被修改。此处 InitialDelaySeconds、TimeoutSeconds、 ExecAction、HttpGetAction 延迟检查时间、超时时间、命令行方式、HttpGet 请求方式的参数 与"Liveness Probe"中相同。

11.14 什么是标签?

新建的标签不能和系统标签重复,目前系统标签有 os. architecture、os. name、os. version、 supportContainer。

标签以 key/value 键值对的形式附加到各种对象中,如应用、服务、节点等。标签定义了这些对象的可识别属性,用来对它们进行管理和选择。

如下图,假设为应用(APP1、APP2、APP3)定义了3个标签:release、env、role。不同应用定义了不同的取值。

在使用应用调度或其他功能时,选择"key/value"值分别为"role/frontend"的应用,则会选择到 "APP1 和 APP2"。



标签值格式要求:其中,"键"为字符串,长度为1[~]63,最多只能包含一个"/"。除此之外,其取值 范围还需满足如下要求:

1、不存在"/":由大小写字母、数字、中划线(一)、下划线(_)和英文句号(.)组成,且必须以



大小写字母或数字开头结尾。

2、存在"/",分如下三种情况:

1)"/"前取值包含".":"."前后分别由小写字母、数字和中划线(-)组成,且首尾字母必须 为小写字母或数字。

 2) "/"前取值不包含".":由小写字母、数字和中划线(-)组成,且首尾字母必须为小写字母 或数字。

3)"/"后:由大小写字母、数字、中划线(-)、下划线(_)和英文句号(.)组成,且必须以大小写字母或数字开头结尾。

11.15 如何通过设计器示例模板创建堆栈?

平台提供符合 TOSCA 标准的图形化编排工具。图形化编排工具可以通过拖拽的方式完成复杂应用的编 排及拓扑设计,并且可以保存成应用模板,使用模板可以直接创建多个容器或虚机组成的复杂应用,大大 简化应用部署难度,提升生产力。

为用户更好的理解如何使用设计器创建堆栈, ServiceStage 提供了示例模板供用户使用。

本章以混编应用为例(混编应用指容器应用和虚机应用的混编),介绍如何使用图形化设计器。

混编应用示例介绍:这是一个混合编排的典型示例,包括3个组件:

1、1个前端组件,运行在容器中。

2、1个后端组件,运行在虚拟机中。

3、1个 MySQL 数据库,运行在容器中,前后端应用都可以从后端 MYSQL 应用中获取数据。

前提条件:

1、已创建资源,具体操作请参见 2.1.5 创建资源空间、2.1.6 创建集群、2.1.7 为集群添加节点。
 若已有集群和节点资源,无需重复操作。

2、已上传混编应用所需的镜像和软件包,并记录仓库地址。

操作步骤:





1、单击菜单栏的"编排 > 设计器",进入设计器界面。设计器界面介绍请参见 5.4.5.1 概述。

2、单击"选择示例模板"。

×	D -	B	*	Ø	>	圃	ආ	ļ
新建空白樓	韨							
选择示例模	詼	-						

3、选择"混编应用典型示例",单击"确定"。

4、拖拽对象。

1) - 2个应用:前端虚机应用 fe-vm、后端容器应用 backend-container。

2) 2个软件:分别存放前面 2个应用的镜像或软件包(fe-vm-soft、backend-soft)。

3)2个集群:将应用部署到对应的集群,可以选择同个集群(集群名称分别为:fe-vm-host、 backend-host)。

5、拖拽关联关系。

- 1) 将应用与对应的软件相连接。
- 2) 将应用与部署的集群相连接。

6、设置各属性的参数。用户可右键单击各组件,查看参数设置数据。

7、单击"输入输出",设置输入输出参数。在示例模板中,由于没有放置软件包和镜像,所以不涉 及输入输出参数,用户创建堆栈时需要根据实际情况配置。

1) 单击参数后的"编辑",分别编辑3个参数;

参数	参数说明
access_path	域名的路径信息
container_image	容器的镜像地址



参数	参数说明
container_port	容器暴露的端口号
vm_package	虚机应用的软件包地址
vm_port	虚机应用对外暴露的监听端口

2) 配置完成后,单击"保存"。

8、单击 🗹 ,校验设计包,若成功代表设计包已部署成功。

9、单击 📥 , 浏览器左下角会自动下载该示例模板的设计包。

11.16 如何配置服务发布参数?

配置服务发布参数。当前支持三种服务类型,使应用可以被访问(无论内部还是外部)。您可以根据 应用需求选择对应的服务发布类型。当前支持如下服务类型:

1、InternalService: 若希望应用可以被系统内部访问,请选择InternalService。使用此类型,将 使用集群内的私有 IP 来暴露应用。操作步骤请参见服务发布类型为InternalService。

2、NodePort: 若应用是内部调试,但希望可以被内网或外网访问,建议选择 NodePort。用户可以通过"节点 IP:节点端口"访问到内部容器应用,但要求用户必须可以访问到该节点。如果该节点是外网 IP,外部可直接访问。操作步骤请参见服务发布类型为 NodePort。

3、LoadBalancer:若希望应用可以被外部访问,建议选择 LoadBalancer 。操作步骤请参见服务发布 类型为 LoadBalancer 。

服务发布类型为 InternalService 操作步骤:

1、选择服务类型为: InternalService。

2、配置服务参数:

参数	参数说明
服务名称	输入应用发布的可被外部访问的名称。
应用端口	容器中应用启动监听的端口。



参数	参数说明
服务端口	Cluster IP 的服务端口,映射到容器的应用端口。
协议	TCP/UDP。

3、单击"确认"。

服务发布类型为 NodePort 操作步骤:

- 1、择服务类型为: NodePort。
- 2、配置服务发布参数

参数	参数说明
服务名称	输入应用发布的可被外部访问的名称。
应用端口	容器或进程中应用启动监听的端口。
服务端口	容器或进程映射到节点上的端口。建议为空,系统会自动分配端口 号。
	若用户有特定需求,可手动输入节点端口。端口范围由安装 FusionStage 的运维人员配置,一般情况下默认为 30000−32767。
	配置完成后,系统会在用户所在项目的所有节点上打开一个真实的 端口号。
	说明:创建拥有多个 Docker 容器的应用时,请确保容器应用使用 的端口不冲突 , 否则部署会失败。
	nodePort 参数值需确保全局唯一,否则会因为端口冲突而导致部署应用失败。若选择手动输入节点端口号,建议提前登录到任意一个纳管节点上(需确保已获取节点登录账号和密码),执行 netstat -apn grep LISTEN 命令,查询端口是否已被占用。若已被占用,需填写其他参数值(nodePort 的端口范围是 30000-32767)。
协议	TCP/UDP.

3、单击"确认"。

服务发布类型为 LoadBalancer 操作步骤:

1、选择服务类型为: LoadBalancer。

2、(可选)若需要创建或增加 ELB 实例,请单击"这里",创建 ELB 实例。若不创建 ELB 实例,系统 会提供平台默认的 ELB,该 ELB 和其他租户共享。



3、配置服务发布参数:

参数	参数说明
服务名称	输入应用发布的可被外部访问的名称。
应用端口	容器中应用启动监听的端口。
服务端口	代表 ELB 上注册的对外端口,外部访问使用 ELB 的 VIP+服务端口。 若对外协议或后端协议类型选择 HTTP/HTTPS,服务端口不可自定义。
协议	仅支持 TCP。
ELB 实例	若未创建 ELB 实例,则无需选择。 系统会提供平台默认的 ELB,该 ELB 和其他租户共享。

4、单击"确认"。

12 参考知识

12.1 资源描述文件配置说明

12.1.1 Json 格式资源文件配置说明

背景信息

本章节主要介绍 Json 类型的资源描述文件的配置示例。

Secret 资源描述文件

文件名称为 secret. json, 配置示例如下:

```
{
  "kind": "Secret",
  "apiVersion": "v1",
  "metadata": {
    "name": "lcq-test-secret",
    "namespace": "lcqtest",
    "enable": true
},
  "data": {
```

```
"token": "xxxxxx"
},
"type": "Opaque"
}
```

大異 Cloud

DaemonSet 资源描述文件

{

文件名称为 daemonset. json, 配置示例如下:

```
"kind": "DaemonSet",
 "apiVersion": "extensions/v1beta1",
 "metadata": {
   "name": "icagent",
   "namespace": "cse-system",
   "labels": {
    "name": "icagent"
  },
   "enable": true
 },
 "spec": {
   "selector": {
     "matchLabels": {
      "app": "icagent"
    }
   },
   "template": {
    "metadata": {
      "creationTimestamp": null,
     "labels": {
      "app": "icagent"
     },
      "enable": true
   },
     "spec": {
      "restartPolicy": "Always",
     "terminationGracePeriodSeconds": 30,
      "dnsPolicy": "ClusterFirst",
      "securityContext": {},
      "imagePullSecrets": [
       {
          "name": "cse-system-secret"
       }
      ],
      "processes": [
       {
                   "name": "foundation",
                                                  "package":
"http://10.175.10.205:20202/app/v1/icagent-monitor/icagent-repo/undefined/undefined/ICProbeAgent-
4.0.RC17.zip/4.0.RC17",
          "resources": {
           "limits": {
            "cpu": "100m",
             "memory": "100Mi"
          }
         },
         "lifecyclePlan": {
```



```
"kind": "ProcessLifecycle",
    "namespace": "cse-system",
    "name": "icagent-plc"
    }
    }
}
```

Pod 资源描述文件

}

文件名称为 pod. json, 配置示例如下:

```
{
 "kind": "Pod",
 "apiVersion": "v1",
 "metadata": {
   "name": "icagent-af6hc",
   "namespace": "lcqtest",
   "labels": {
    "app": "icagent"
   },
   "annotations": {
    "kubernetes.io/created-by":
"{\"kind\":\"SerializedReference\",\"apiVersion\":\"v1\",\"reference\":{\"kind\":\"DaemonSet\",\"
namespace\":\"cse-system\",\"name\":\"icagent\",\"uid\":\"aaf82d69-8e98-11e6-a651-0242ac102a14\",
\"apiVersion\":\"extensions\",\"resourceVersion\":\"24290369\"}}\n",
    "scheduler.alpha.kubernetes.io/isExclusive": "Ignore"
   },
   "enable": true
 },
 "spec": {
   "restartPolicy": "Always",
   "terminationGracePeriodSeconds": 30,
   "dnsPolicy": "ClusterFirst",
   "serviceAccountName": "default",
   "serviceAccount": "default",
   "nodeName": "10.175.11.240",
   "securityContext": {},
   "imagePullSecrets": [
    {
      "name": "lcqtest-secret"
    }
   ],
   "processes": [
    { "name": "foundation",
                                      "package":
"http://10.175.10.205:20202/app/v1/icagent-monitor/icagent-repo/undefined/undefined/ICProbeAgent-
4.0.RC17.zip/4.0.RC17",
      "env": [
        {
         "name": "KUBERNETES SERVICE TOKEN DIR",
          "value": "/var/run/secrets/kubernetes.io/serviceaccount/cse-system/icagent-af6hc"
        }
```

```
],
      "resources": {
        "limits": {
         "cpu": "100m",
         "memory": "100Mi"
        },
        "requests": {
         "cpu": "100m",
         "memory": "100Mi"
        }
      },
      "volumeMounts": [
       {
         "name": "default-token-o4uqj",
         "readOnly": true,
         "mountPath": "/var/run/secrets/kubernetes.io/serviceaccount/cse-system/icagent-af6hc"
        }
      ],
      "lifecyclePlan": {
        "kind": "ProcessLifecycle",
        "namespace": "cse-system",
        "name": "icagent-plc",
        "apiVersion": "v1alpha1"
      }
     }
   ]
 }
}
```

LifeCycle 资源描述文件

大翼口 e Cloud

文件名称为 lifecycle. json, 配置示例如下:

```
{
 "kind": "ProcessLifecycle",
 "apiVersion": "paas/vlalphal",
 "metadata": {
  "name": "icagent-plc",
   "namespace": "lcqtest",
  "enable": true
 },
 "spec": {
   "lifecycle": {
    "install": {
     "actionFlow": [
       "install"
      ]
    },
    "uninstall": {
     "actionFlow": [
       "uninstall"
      ]
    },
     "start": {
      "actionFlow": [
```

12 参考知识

```
"start"
 ]
 },
 "stop": {
   "actionFlow": [
     "stop"
 ]
 },
 "update": {
  "actionFlow": [
   "update"
  ]
 }
},
"actions": [
 {
   "name": "install",
   "commands": [
   "/bin/bash",
    "bin/manual/setup.sh",
    "-ip",
    "10.175.10.219:31943",
    "-localip",
    "127.0.0.1",
    "-user",
    "paas"
   ]
},
{
   "name": "start",
   "commands": [
    "/bin/bash",
     "-c",
     "cd /opt/oss/servicemgr/ICAgent/bin/manual/;su paas -c '/bin/bash mstart.sh'"
   ]
 },
 {
   "name": "stop",
  "commands": [
   "/bin/bash",
    "-c",
    "cd /opt/oss/servicemgr/ICAgent/bin/manual/;su paas -c '/bin/bash mstop.sh'"
  ]
},
{
   "name": "uninstall",
   "commands": [
    "/bin/bash",
    "-c",
    "cd /opt/oss/servicemgr/ICAgent/bin/manual/;/bin/bash uninstall.sh"
   ]
 },
{
   "name": "update",
   "commands": [
```

大翼口 e Cloud



```
""
}
}
```

12.1.2 Yaml 格式资源文件配置说明

Secret 资源描述文件

文件名称为 secret. yaml, 配置示例如下:

```
apiVersion: v1
kind: Secret
metadata:
   name: lcq-test-secret1
   namespace: cse-system
data:
   data1: OEdGTFFVUFZUS1BXWTdPUEFBRks=
   data2: VFMOMOVZU1JPTzFLWkJDVUhBWk90Vk5LTVVMR0s0TVpIU0ZUREVWSw==
```

DaemonSet 资源描述文件

```
文件名称为 daemonset. yaml, 配置示例如下:
```

```
apiVersion: "extensions/v1beta1"
kind: DaemonSet
metadata:
 name: testprocess1
 labels:
   name: liveness
spec:
 selector:
   matchLabels:
    node: 127.0.0.1
 template:
   metadata:
    labels:
      node: 127.0.0.1
    enable: true
  spec:
    restartPolicy: Always
     processes:
     - name: test1
       package: http://10.162.197.181/Test process.tar.gz
        lifecyclePlan:
        kind: ProcessLifecycle
         name: universe
        readinessProbe:
         exec:
           command:
           - /bin/bash
```



```
- -c
- "/bin/cat /home/hello.sh && /bin/bash /home/hello.sh"
initialDelaySeconds: 1
timeoutSeconds: 1
failureThreshold: 5
```

Pod 资源描述文件

文件名称为 pod. yaml, 配置示例如下:

```
apiVersion: v1
kind: Pod
metadata:
 name: busybox0
 labels:
   app: busybox0
   status: replaced
spec:
 containers:
 - image: busybox
  command:
    - sleep
    - "3600"
   imagePullPolicy: IfNotPresent
   name: busybox
   resources:
    limits:
      cpu: 0.5k
      memory: 200Mi
    requests:
      cpu: 200m
      memory: 100Mi
 restartPolicy: Always
```

LifeCycle 资源描述文件

文件名称为 lifecycle. yaml, 配置示例如下:

```
kind: ProcessLifecycle
apiVersion: paas/vlalphal
metadata:
 name: test
 enable: true
spec:
 lifecycle:
   install:
    actionFlow:
     - install
   postStart:
    actionFlow:
     - poststart
   configure:
    actionFlow:
    - configure
```

update:	
actionFlow:	
- update	
rollback:	
actionFlow:	
- rollback	
actions:	
- name: fail	
commands: ["echo",	"Fail >> /home/res.log"]
- name: poststart	
commands: ["echo",	"poststart >> /home/res.log"
- name: configure	
commands: ["echo",	"configure >> /home/res.log"]
- name: install	
commands: ["echo",	"install >> /home/res.log"]
- name: update	
commands: ["echo",	"update >> /home/res.log"]
- name: rollback	
commands: ["echo",	"rollback >> /home/res.log"]

12.2 软件包打包规范

命名规范

天翼**云** e Cloud

软件包名称,需要符合"软件名+后缀"的方式。其中后缀必须为:tar.gz、tar或zip。

```
{软件名}.{后缀}
```

```
🛄 说明
```

后缀必须和软件包压缩方式一致,否则无法正常解压软件包。

目录结构

软件包需要确保在解压缩后能够正常执行生命周期指定的命令脚本。

建议软件包目录结构如下:

🛄 说明

目前不建议解压缩软件包后存在顶层目录的目录结构。这种目录结构下,需要修改生命周期执行命令,需要带上顶层 目录名称,才能找到相应脚本。

```
|- bin
   |- xxx.tar.gz
   |- xxx.bin
   |- scripts
   |- install.sh
   |- start.sh
```

目录说明如下:



目录	说明
bin	该目录下存放的是用户软件包的可执行信息,比如可执行的 bin 文件,依赖 的压缩包文件等。
scripts	该目录下存放的是生命周期脚本。
	在创建应用时,可以根据生命周期脚本的位置指定执行命令。比如 install 阶段,指定"bash scripts/install.sh",执行安装脚本。
	软件包应用支持的生命周期如下:
	 安装(Install):软件安装命令。 启动后处理(PostStart):软件启动后操作。 启动(Start):软件启动命令。 重启(Restart):执行软件重启命令,用于应用健康检查失败恢复使用。 停止前处理(PreStop):软件停止前操作。 停止(Stop):软件停止命令。 更新(Update):软件升级命令。
	●卸载(Uninstall):软件卸载命令。

12.3 应用监控常见问题

12.3.1 如何搜索指标

1、在主菜单中选择"运维 > 应用监控"。

2、在左侧导航栏中选择"指标监控 > 浏览指标"。

3、通过如下两种方式中的任意一种方式选择需要搜索的指标所在的命名空间。

方式一:在页面左侧导航栏中,选择"指标监控 > 应用"、"指标监控 > 节点"或"指标监控 > SLA"。

方式二:在页面右侧单击"应用"、"节点"或"SLA"。

4、在页面右侧选择需要搜索指标的类型。例如,"应用"命名空间下指标的类型有"无状态容器指标"、"无状态应用指标"、"应用实例指标"、"应用实例状态指标"、"容器指标"和 "应用指标"。

5、在右上角的搜索框中输入关键字。

说明:



1、"指标名称(metricName)"字段只支持搜索括号里的内容。例如,当指标名称为"CPU 使用率 (cpuUsage)"时,则需要输入"cpuUsage"里的内容进行搜索。

2、支持"应用名称(appName)"、"指标名称(metricName)"等多个字段联合搜索,多个字段间 用空格隔开。例如,当搜索应用名称为"cfe-addons"的应用的 CPU 使用率指标时,则需要输入: cfe-addons cpuUsage。

3、支持模糊搜索。例如,当搜索应用名称中包含"cfe-"的应用的 CPU 使用率指标时,则需要 输入: cfe- cpuUsage。

6、单击⁹,或在键盘上按"Enter"键。

12.3.2 如何搜索阈值规则

1、在主菜单中选择"运维 > 应用监控"。

2、在左侧导航栏中选择"阈值规则 > 所有阈值"。

3、(可选)在"所有阈值"页面右上角的"状态"下拉列表框中选择需要搜索的阈值规则的状态。

4、在"所有阈值"页面右上角的搜索框中输入关键字。

说明:1)支持"名称"、"指标名称"和"描述"等字段搜索,但不支持多个字段联合搜索;2)"指标名称"字段只支持搜索括号里的内容。例如,当指标名称为"CPU使用率(cpuUsage)"时,则需要输入"cpuUsage"里的内容进行搜索。

5、单击⁹,或在键盘上按"Enter"键。

6、(可选)在页面右侧单击🔍,可以自定义显示列。

12.3.3 如何修改阈值规则

1、在主菜单中选择"运维 > 应用监控"。

2、在左侧导航栏中选择"阈值规则 > 所有阈值"。

3、在"所有阈值"页面右上角的搜索框中输入关键字,搜索需要修改的阈值规则,详细操作请参见



12.3.2 如何搜索阈值规则。

4、单击需要修改的阈值规则右侧"操作"列的"修改"。

5、在"修改阈值"对话框中,根据需要修改阈值规则的相关参数。

6、单击"确定"。

12.3.4 如何删除阈值规则

1、在主菜单中选择"运维 > 应用监控"。

2、在左侧导航栏中选择"阈值规则 > 所有阈值"。

3、在"所有阈值"页面右上角的搜索框中输入关键字,搜索需要删除的阈值规则,详细操作请参见

12.3.2 如何搜索阈值规则。

4、选中一个或多个需要删除的阈值规则前的复选框,单击页面上方的"删除"。

说明:在需要删除的阈值规则右侧"操作"列的"更多"中选择"删除",可以删除该条阈值规则。 默认阈值规则被删除后,系统将不会重新创建。

5、在"警告"对话框中单击"确定"。

12.3.5 如何删除监控视图

1、在主菜单中选择"运维 > 应用监控"。

2、在左侧导航栏中选择"Dashboard"。

3、在"操作"下拉列表框的左侧选择需要删除的监控视图。

4、在该监控视图的"操作"下拉列表框中选择"删除监控视图"。

说明:通过"操作"下拉列表框中的选项还可以对该监控视图进行另存为、重命名和导出监控报告等 操作。

5、在"警告"对话框中单击"确定"。



- 1、在主菜单中选择"运维 > 应用监控"。
- 2、在左侧导航栏中选择"Dashboard"。

3、在"操作"下拉列表框的左侧选择需要编辑指标图表的监控视图。

4、在该监控视图中,单击需要编辑的指标图表右上角的下拉箭头并选择"编辑"。

5、在"选择指标"页面中,根据需要编辑指标图表的相关参数。可以通过如下操作编辑指标图表的标题:

- 1) 单击"标题"旁边的
- 2) 在文本框中输入新的指标图表标题。
- 3) 单击"保存"。
- 6、单击"确定"。

7、在页面右侧单击"保存监控视图",或在"操作"下拉列表框中选择"保存监控视图"。

12.3.7 如何在监控视图中添加指标图表

可以通过如下两种方式在监控视图中添加指标图表。

1、方式一:通过添加指标图表功能将指标图表添加到监控视图中。

- 1) 在主菜单中选择"运维 > 应用监控"。
- 2) 在左侧导航栏中选择"Dashboard"。

3) 在"操作"下拉列表框的左侧选择需要添加指标图表的监控视图。

5) 在 "选择指标"页面中,选择一个或多个关注的指标,指标的详细信息请参见 6.1.9 应用监控



指标总览。

说明:监控视图的每个指标图表中最多可以选择4条指标。可以通过指标搜索功能,快速查找到 需要的指标,详细操作请参见12.3.1 如何搜索指标。

- 6)单击平均值 *和5分钟 *分别设置统计方式和统计周期。
- 7)单击"确定"。
- 8) 在页面右侧单击"保存监控视图", 或在"操作"下拉列表框中选择"保存监控视图"。
- 2、方式二:通过复制指标图表功能将指标图表添加到监控视图中。
 - 1) 在主菜单中选择"运维 > 应用监控"。
 - 2) 在左侧导航栏中选择"Dashboard"。
 - 3) 在"操作"下拉列表框的左侧选择需要添加指标图表的监控视图。
 - 4) 在该监控视图中,单击需要复制的指标图表右上角的下拉箭头、并选择"复制"。
 - 5)(可选)编辑已复制的指标图表,详细操作请参见12.3.6 如何在监控视图中编辑指标图表。
 - 6) 在页面右侧单击"保存监控视图", 或在"操作"下拉列表框中选择"保存监控视图"。

12.3.8 如何在监控视图中删除指标图表

- 1、在主菜单中选择"运维 > 应用监控"。
- 2、在左侧导航栏中选择"Dashboard"。
- 3、在"操作"下拉列表框的左侧选择需要删除指标图表的监控视图。
- 4、在该监控视图中,单击需要删除的指标图表右上角的下拉箭头并选择"删除"。
- 5、在页面右侧单击"保存监控视图",或在"操作"下拉列表框中选择"保存监控视图"。

12.3.9 如何在监控视图中刷新指标图表

指标图表的刷新周期为1分钟,如果用户需要实时查看指标图表的详细信息,可以通过如下操作刷新


指标图表:

- 1、在主菜单中选择"应用运维 > 应用监控"。
- 2、在左侧导航栏中选择"Dashboard"。
- 3、在"操作"下拉列表框的左侧选择需要刷新指标图表的监控视图。

4、在该监控视图中,单击需要刷新的指标图表右上角的下拉箭头并选择"刷新"。

12.3.10 如何在监控视图中放大或移动指标图表

1、在主菜单中选择"运维 > 应用监控"。

2、在左侧导航栏中选择"Dashboard"。

3、在"操作"下拉列表框的左侧选择需要放大或移动指标图表的监控视图。

4、在该监控视图中,双击需要放大的指标图表,或单击需要放大的指标图表右上角的下拉箭头、并选择"放大",均可暂时放大该指标图表。

5、在其中一个指标图表的任意位置按住鼠标左键并拖动至另一个指标图表所在的位置上,松开鼠标 后系统将自动交换两个指标图表的位置。

6、在页面右侧单击"保存监控视图",或在"操作"下拉列表框中选择"保存监控视图"。

12.4 应用编排设计包说明

12.4.1 设计包概述

设计包是用来描述应用拓扑、编排模板及相关文件的压缩包。压缩包中至少要包含一个文件: TOSCA 模板 (Topology and Orchestration Specification for Cloud Application)。

TOSCA 模板是由开放标准联盟 OASIS 管理的独立技术委员会发布的规范,目标是拉通多厂商云应用生命周期管理流程。在 TOSCA 中,有以下一些重要基本概念:

A、Node: 节点, TOSCA 的基本元素单元, 可表示资源/软件/应用等对象, 包含了属性/关系/依赖/生命周期等信息。



B、Relationship:关联关系,定义了Node之间的关系,包含了属性/生命周期等信息。

C、Capability: Node 提供的开放能力, 主要用于和 requirement 匹配, 定义在关联关系的接收端。

D、Requirement: Node 需要的需求,需要与被依赖的 node 匹配 capability,定义在关联关系的发起端。

E、Interface: 定义生命周期事件,作用在节点和关联关系上。

F、Group:组类型,将一组特定的节点组成一个 Group,可以使用相同的策略、执行同一生命周期事件等。

12.4.2 设计包开发原理

1、概述

应用拓扑模型遵循 TOSCA 规范,由节点(Node)+关系(Relationship)组成。一个基础的应用拓扑 模型如下图。下图中,方块代表节点,连线代表关系,虚线框的方块代表可选元素。



名词	解释
Resource	资源类节点,比如 VM、物理机、容器等。
Application	云应用,由一个或多个云应用组件组成。
Application	运行在资源上的云应用组件,是对最小可部署对象的一种描述。 MicroService 是一种 Application。
SoftwareComponen t	软件组件,云应用组件的组成部分,即软件包。也可以作为 Application 的属性,是可选节点。
Service	应用所依赖的服务,服务是对按需取用的功能对象的一种描述。
HostedOn	只能用于 App 与 Resource 间,表示应用运行在资源上。
ConsistsOf	表示组合关系,比如表示 App 由 Application 组成,Application 由 SoftwareComponent 组成。



名词	解释
ConnectsTo	表示调用或连接关系,比如用于 Application/MicroService 之间, 以及资源与资源之间。
BindsTo	只能用于 Application 与 Service 间,表示服务实例绑定关系。

2、设计包样例

如上图所示,本例是一个基于 TOSCA 模板的设计包示例。本例中只包含两个 Node:MySQL 和 db_server, 它们分别是一个 DBMS. MYSQL 和 Compute 类型的 Node,通过一个 HostedOn 关联关系相连接形成拓扑, 组成一个简单的 MySQL 云应用。

关联关系的两端,db_server的Container能力描述需要与MySQL对宿主环境的需求匹配,这样此 HostedOn关系才能够成立。MySQL节点中定义了密码、端口等属性。



3、TOSCA 函数

TOSCA 提供内置函数,用于方便书写 TOSCA 模板。下表列出了一些常用内置函数。

函数名称 解释 示例	
--------------------------	--



函数名称	解释	示例
get_input	节点/关联关系 的属性值,从 TOSCA的 inputs 中获取	inputs: hello-package: default: "10.136.100.100:20202/domain/backend:v2"
		node templates:
		hello-package:
		type: hwpaas.nodes.SoftwareComponent
		properties:
		package:
		package_type: container
		process_spec:
		name: {get_input: hello-package}
get_attrib ute	TOSCA 的输出, 需要从指定的节 点获取指定的运 行态属性信息	<pre>node_templates: hello-app: type: hwpaas.nodes.Application properties: type: container outputs:</pre>
		hostip:
		value: {get_attribute:[hello-app,ips]}
get_proper ty	获取节点静态属 性	<pre>node_templates: web_server: type: hwpaas.nodes.Compute properties: port: 80 outputs: web_server_id: description: Web server port value: { get property: [web_server_port] }</pre>

12.4.3 设计包格式规范

1、设计包命名规范

目前支持如下方式命名的设计包:

{模板名}. {后缀}

其中,后缀必须为tgz或tar.gz或zip。

2、目录结构



设计包的目录结构如下:

- |- blueprint.yaml (设计包描述文件, 必选)
- |- extend (设计包描述的补充文件,可选)
 - |- app name.yaml
- • •
- |- resources
 - |- i18n-zh-cn.json
 - |- i18n-en-us.json

其中:

A、blueprint.yaml: TOSCA 描述文件, 应用的 TOSCA 描述入口文件, 必选。

B、extend: 该目录存放应用的 TOSCA 描述的补充信息,包括子应用的描述、节点类型定义、策略定 义、生命周期等,都可以定义在该目录下,也可以直接定义在应用的整体 TOSCA 描述中。文件名可自 定义,只要在 blueprint.yaml 中的 imports 引用正确即可。比如:

```
tosca definitions version: hwpaas tosca version 1 4
description: Template for deploying aspecific application.
imports:
    - folder/hello.yaml
```

C、resources:存放资源。主要存放输入输出参数的显示语言资源,用以支持应用的多语言显示,目前支持简体中文和美式英文。可选。

D、i18n-zh-cn.json: 简体中文显示资源文件。可选。

E、i18n-en-us.json: 美式英文显示资源文件。可选。

3、blueprint.yaml 配置示例

blueprint.yaml 文件配置示例如下:

```
tosca definitions version: hwpaas tosca version 1 4
description: Template for deploying aspecific application.
imports:
    - extend/hello.yaml
inputs:
    hello-package:
    default: "10.136.100.100:20202/domain/backend:v2"
    instances:
    type: integer
    description: Number hello app instances.
    constrains:
        - valid_values: [ 1, 2, 3]
node_types:
    hello_type:
    derived_from:hwpaas.nodes.Application
```



```
node_templates:
 hello-app:
   type: hello_type
  properties:
    type: container
   requirements:
    - package:
      node: hello-package
      relationship: hwpaas.relationships.PackageConsistsOf
 hello-package:
  type: hwpaas.nodes.SoftwareComponent
   properties:
    package:
     package type: container
      process spec:
       name: {get input: hello-package}
outputs:
```

hostip:

```
value: {get attribute:[hello-app,ips]}
description: ip list of hello-app
```

各字段说明请见下表:

字段	字段说明
tosca_definitions_versio	应用模板所基于的类型定义版本。
n	若平台使用的版本为 hwpaas_tosca_version_1_4,在设计包模板 文件中此参数必选。
description	应用模板的描述信息,可选配置项。
imports	引用的类型定义/模板,可选配置项。
	当全部使用平台内置类型时,将不需要引用额外的 imports 文件。
inputs	定义输入参数。用户可以在 inputs 中定义参数的类型、描述信息、 默认值等,默认值信息可在创建堆栈时由外部输入替换。对于类 型为 password、secret 的输入参数,不允许输入默认值。
node_types	自定义类型文件,用于基于平台内置的 TOSCA 节点类型,扩展属性、生命周期等用户自己定义的类型。
	说明:图形化编排设计器中暂不支持此功能。
node_templates	定义应用拓扑描述,所有的 TOSCA 拓扑节点,都在 node_templates 中定义。节点是基于某个 node_type 类型的实例化,类型中定义 的属性值、关联关系、生命周期等,都将在节点中体现。在节点 的实例化中也可以更新属性值、关联关系的赋值,但更新的范围





字段	字段说明
	都限制在了类型定义的范围内。
outputs	定义拓扑的输出,通过一定的组织形式得到输出信息,如应用的 IP 运行环境地址等。

4、i18n-zh-cn. json 配置示例

i18n-zh-cn. json 文件配置示例如下:

```
{
 "inputs" : {
  "inputparam1" : {
    "name" : "输入参数1",
    "description" : "输入参数描述 1"
   },
  "inputparam2" : {
   "name": "输入参数 2",
   "description": "输入参数描述 2"
   }
 },
 "outputs" : {
  "outputparam1" : {
    "name" : "输出参数1",
    "description" : "输出参数描述 1"
  },
  "outputparam2" : {
   "name" : "输出参数 2",
   "description" : "输出参数描述 2"
  }
 }
```

文件分成两段, "inputs"和 "outputs",分别对应 blueprint.yaml 中的 inputs 和 outputs。每一 个大段中,按参数逐个展开,例如 inputparam1 来自 blueprint.yaml 中的某个输入参数,其 name 表示页 面显示的参数名称, description 表示页面显示的参数描述,这两个字段按不同的语言填写,从而实现国 际化显示功能。outputs 的内容也是如此的。

12.5 应用编排支持的对象类型与配置说明

12.5.1 应用组

1、模型定义

应用组是一种逻辑节点,主要用于体现多个应用之间的组织关系。应用组的节点类型定义如下:

12 参考知识

hwpaas.nodes.AppGroup: derived_from: hwpaas.nodes.Root properties: deploy: type: boolean default: true immutable: true custom_spec: default: {} requirements: - member: relationship: hwpaas.relationships.ConsistsOf node: hwpaas.nodes.Application

2、属性配置

e Cloud

属性	说明
是否部署 (dep I oy)	应用组是否部署。因应用组为逻辑节点,当前该属性并不会影响到应用 组所包含的应用是否部署。当应用组的 deploy 为 false 时,应用将属于 defaultGroup 的应用组。该属性在设置后不允许修改。
	●true: 部署 ●false: 不部署
自定义属性 (custom_spec)	应用组的用户自定义配置信息,可以允许用户配置任意自定义属性(属 性信息为 json 结构),但必须保证输入的 json 对象是合法的。
	此处设置的自定义信息在创建应用时,会传递到应用的配置信息中。

3、关联关系

关联关系	说明
member	应用组的关联关系组成中,需要通过 member 关系定义应用组包含的应用。若不定义 member,应用组没有实际意义。
	其中,relationship 固定为 hwpaas.relationships.ConsistsOf(组 成关系),对应的 node 必须是应用类型。
	目前应用的数量暂时没有限制,建议不要超过5个。

4、配置示例

```
node templates:
  use-app: //应用组名称
  type: hwpaas.nodes.AppGroup
  properties:
    custom spec: //应用组自定义属性, 配置了 tag 的数组
    tags:
        - key: netype
        value: "com.huawei.Universe.Application"
        globalConfigurations:
```





```
requirements:
- member:
node: use-appcomponent //应用组依赖应用
```

12.5.2 应用

1、模型定义

应用是对最小可部署对象的一种描述,代表一个软件应用的逻辑实体,表示一个有业务功能呈现给用 户的计算机软件应用。应用的节点类型定义如下:

```
hwpaas.nodes.Application:
   derived from: hwpaas.nodes.Root
   properties:
     type:
      type: string
      description: process/container, application type, now for process(directly on \ensuremath{vm/pm}) and
container
      default: process
      immutable: true
     instances:
      type: integer
      default: 1
     deploy:
      type: boolean
      description: deploy this application or not
      default: true
      immutable: true
     stateful:
      type: boolean
      description: useful on container mode, false --> ReplicaSet, true --> StatefulSet;
      default: false
      immutable: true
    packages:
      default: []
     custom_spec:
      default: {}
     exclusive:
      type: string
      description: this application need exclusive resources or not, default: (container-->false,
process-->true)
      default: "default"
     resource selector:
      description: defined for select resource
      type: hwpaas.datatypes.resource selector
      default: {}
    bind services:
      description: defined for services which applications binds to
      default: []
      type: hwpaas.datatypes.service bind
      islist: true
    name:
```



```
type: string
      default: ''
      constrains:
        max length: 24
        regex: (^$) | (^[a-z] ([-a-z0-9]*[a-z0-9])?$)
    immutable: true
    certification:
      type: hwpaas.datatypes.Certification
      default: {} annotations: type: hwpaas.datatypes.Annotations islist: true default: []
upgrade_strategy: type: hwpaas.datatypes.upgrade_strategy default: {} restart_policy: type: string
default: 'Always' description: the way to restart an application instance when it is failed
   capabilities:
    endpoint:
      type: hwpaas.capabilities.Endpoint
   requirements:
     - storage:
        capability: hwpaas.capabilities.Attachment
        relationship: hwpaas.relationships.HostedOn
        node: hwpaas.nodes.Resource
     - volume:
        capability: hwpaas.capabilities.Attachment
        relationship: hwpaas.relationships.AttachesTo
        node: hwpaas.nodes.Volume
     - host:
        capability: hwpaas.capabilities.Container
        relationship: hwpaas.relationships.HostedOn
        node: hwpaas.nodes.Resource
     - service:
        capability: hwpaas.capabilities.Endpoint.Service
        relationship: hwpaas.relationships.BindsTo
       node: hwpaas.nodes.Service
     - package:
        relationship: hwpaas.relationships.PackageConsistsOf
        node: hwpaas.nodes.SoftwareComponent
     - dependency:
       relationship: hwpaas.relationships.ConnectsTo
   interfaces:
     Standard:
      type: hwpaas.interfaces.AppLifecycle
```

2、属性配置

属性	说明
名称(name)	应用的名称,默认为空。名称格式非空时,需要满足 ^[a-z]([-a-z0-9]*[a-z0-9])?\$的正则条件和最大 24 的长度限制。该属性在 设置后不允许修改。
	 老定义名称,应用相关资源的名称都将使用这个名称。 若使用默认值,应用相关资源的名称将采用"节点名称+随机后缀"的格式。
实例数 (instances)	应用的实例数。取值范围为[0, 100], 其中"0"表示没有实例。



属性	说明
类型(type)	应用类型,包括进程类型应用(process)和容器类型(container)应用,默 认为 process。该属性在设置后不允许修改。应用所包含的软件组件需要与应 用的类型一致。
是否部署(deploy)	应用是否部署,默认为 true。该属性在设置后不允许修改。若 deploy 设置为 false,则应用将不部署(包括应用所包含的软件组件、依赖的主机资源等)。
有状态(stateful)	标识应用是有状态还是无状态,默认为无状态应用。该属性 immutable 为 true, 设置后不允许修改。
	● false: 无状态。无状态应用对应 k8s 的 replicaset,允许伸缩、升级等生 命周期操作。
	● true: 有状态。有状态应用对应 k8s 的 statefulset, 需要挂载卷, 目前不 支持伸缩、升级等生命周期操作。
包信息列表 (packages)	定义应用的软件包,目前暂未实现。
自定义属性 (custom_spec)	应用组的用户自定义配置信息,允许用户配置任意自定义属性(属性信息为 json 结构)。在创建应用时,该参数将传递到应用的配置信息中。
是否独占	应用的独占资源标记,默认值为"default"。
(exclusive)	• "True":资源独占,配置需要加引号(""),作为字符串处理。
	• "False":资源不独占,配置需要加引号(""),作为字符串处理。
	● "default": 在进程奕应用场景下为独占,在容器场景下为共享。
资源过滤 (resource_select	资源调度,可以在此处配置亲和性/反亲和性,用于指定应用部署的资源,与 应用之间的亲和性关系。
017	亲和性部署的结构如下,目前支持 labels(对应 k8s 的 node selector)和 affinities(目前已支持 node affinity,以及 pod 自身的反亲和; 尚不支持 配置 k8s 的 pod 级别的 affinity)。当 label 与 affinities 同时配置时,两 种调度策略均生效。
	resource_selector: labels: #数组结构,默认为空 - key: xx #标签名称 value: #标签值 affinities:
	antiself: true #反亲和性,默认为 true hosts: #主机亲和性,数组结构,默认为空 - key: aa #主机标签名称 values: [aa, bb, cc] #主机标签值范围 op: In #主机标签匹配策略
绑定服务 (bind_services)	绑定服务,可以在应用中绑定已经创建的服务实例。需要定义绑定服务的关联 关系,并指定服务信息的挂载文件路径。应用可以直接读取挂载路径的文件,



属性	说明
	以获取服务的信息。
	bind_services: - bind_path: string #服务实例信息挂载路径 service_instance_name: string #服务实例名称
证书 (certification)	应用可以通过 certification 来挂载证书和 secret。
	certification: path: string #证书挂载路径 secret: string #证书引用的 secret 名称
注解 (annotations)	用来添加注入自定义的元数据信息,数组形式。每个元素的内容包括下面的内 容
	name: string #自定义元数据名称 value: string #自定义元数据内容
升级策略 (upgrade_strateg y)	应用升级时使用的策略,默认值"RollingUpdate"
	upgrade_strategy: strategy_type: "RollingUpdate" #升级策略,可选范围为 RollingUpdate/Recreate/InplaceRollingUpdate max_unavailable: "1" #最大不可用数,字符串类型,滚动升级时生效 max_surge: "1" #最大超出范围,字符串类型,滚动升级时生效
重启策略 (restart policy)	应用出现异常后,重启应用时使用的策略,默认值"Always"。也可配置为 "OnFailure"或"Never"。
	说明: 设计器中暂时只支持"Always"策略。

3、关联关系

关系	说明
storage	应用使用的存储,目前暂未实现。
volume	应用挂载的卷资源,目前仅支持挂载本地卷。
host	定义应用的运行主机,通过 hwpaas.relationships.HostedOn 的关系关联到主 机(hwpaas.nodes.Compute/hwpaas.nodes.ResourceTemplate)上。
service	定义应用使用的服务(实例),通过 hwpaas.relationships.BindsTo 的关系绑 定服务,对象类型是 hwpaas.nodes.Service。
package	定义应用的软件组件,通过 hwpaas.relationships.PackageConsistsOf 关系指 定软件组件(hwpaas.nodes.SoftwareComponent)。
dependency	定义应用的基本依赖关系,对象可以是任意类型的节点,主要用于指定先后顺 序。关联关系为 hwpaas.relationships.ConnectsTo。



4、配置示例

1)无状态容器应用

```
node templates:
redis-app:
type: hwpaas.nodes.Application
properties:
type: container //定义类型为容器,默认即无状态
requirements:
- package:
    node: redis-soft //定义软件节点,具体容器相关定义都在软件组件上
- host:
    node: redis-cluster //定义主机,在资源池情况下可不定义主机,直接使用资源池配置
```

2) 无状态进程应用

```
node templates:
 redis-app:
  type: hwpaas.nodes.Application //默认即进程类应用
  properties:
  resource selector: //定义标签方式的调度,可选
   labels:
     - key: application
      value: redis
  custom spec: //自定义属性部分,可选
   tags:
     - key: aaa
      value: bbb
  configurations:
   testa: ok
  requirements:
    - package:
      node: redis-soft //定义软件节点,具体容器相关定义都在软件组件上
    - host:
   node: redis-cluster //定义主机,在资源池情况下可不定义主机,直接使用资源池配置
```

3) 有状态容器应用

```
node_templates:
ec:
type: hwpaas.nodes.Application
properties:
type: container //容器应用
name: ec //显示定义 name 字段,应用创建出来的 service/configmap 等资源名称都将使用这个名称
instances : 3 //实例数
stateful: true //有状态应用
resource_selector: //可选
affinities:
antiself: true //配置反亲和性,可选
hosts: //配置主机节点的亲和性,可选
- key: use
values:
```

- ec
- etcd
op: in
requirements:
- package:
node: etcd-unit //软件组件
- host:
node: etcd-cluster //运行主机
- volume:
node: etcd-datadir //挂载本地卷
- volume:
node: etcd-logfile //挂载本地卷
etcd-datadir: //本地卷的定义
type: hwpaas.nodes.LocalVolume //本地卷类型
properties:
name: datadir 名称
type: local_storage //类型: 本地存储
host path: /var/paas/sys/run/ec //定义本地的挂载文件路径

12.5.3 任务(Job 应用)

1、模型定义

天翼**云** e Cloud

```
hwpaas.nodes.Job:
derived from: hwpaas.nodes.Application
properties:
   completions:
    type: integer
    default: 1
    description: the number of repitions pod(s) need(s) to complete
   restart policy:
    type: string
    default: 'OnFailure'
    constrains:
      valid values: ['Never', 'OnFailure']
   interfaces:
      Standard:
      type: hwpaas.interfaces.JobLifecycle
```

2、属性配置

属性	说明
名称(name)	应用的名称,默认为空。名称格式非空时,需要满足 ^[a-z]([-a-z0-9]*[a-z0-9])?\$的正则条件和最大 24 的长度限制。该属性 在设置后不允许修改。
	 若定义名称,应用相关资源的名称都将使用这个名称。 若使用默认值,应用相关资源的名称将采用"节点名称+随机后缀"的格式。
实例数	同一时间执行任务应用的实例数,默认值为1。实际值应该不大于任务数。



属性	说明
(instances)	
类型(type)	应用类型,包括进程类型应用(process)和容器类型(container)应用, 默认为 process。该属性 immutable 为 true,设置后不允许修改。
	应用所包含的软件组件需要与应用的类型一致。
成功执行数 (completions)	重复完成同一任务的次数,默认值为 1。
包信息列表 (packages)	定义应用的软件包,目前暂未实现。
自定义属性 (custom_spec)	应用组的用户自定义配置信息,允许用户配置任意自定义属性(属性信息为 json 结构)。在创建应用时,该参数将传递到应用的配置信息中。
是否独占	应用的独占资源标记,默认值为"default"。对于任务类型
(exclusive)	● "True":资源独占,配置需要加引号(""),作为字符串处理。
	• "False": 资源不独占,配置需要加引号(""),作为字符串处理。
	• "default":
资源过滤 (resource_select	资源调度,可以在此处配置亲和性/反亲和性,用于指定应用部署的资源, 与应用之间的亲和性关系。
or)	亲和性部署的结构如下,目前支持 labels(对应 k8s 的 node selector) 和 affinities(目前已支持 node affinity,以及 pod 自身的反亲和;尚 不支持配置 k8s 的 pod 级别的 affinity)。当 label 与 affinities 同时配 置时,两种调度策略均生效。
	resource_selector: lables: #数组结构,默认为空 - key: xx #标签名称 value: #标签值 affinities:
	antiself: true #反亲和性,默认为 true host: #主机亲和性,数组结构,默认为空 - key: aa #主机标签名称 values: [aa, bb, cc] #主机标签值范围 op: In #主机标签匹配策略
证书	应用可以通过 certification 来挂载证书和 secret。
(certification)	certification: path: string #证书挂载路径 secret: string #证书引用的 secret 名称
注解	用来添加注入自定义的元数据信息,数组形式。每个元素的内容包括下面的



属性	说明
(annotations)	内容
	name: string #自定义元数据名称 value: string #自定义元数据内容
重启策略 (restart_policy)	当应用出现异常后,重启应用时使用的策略,默认值"OnFailure"。也可 配置为"Never"。
是否部署(dep l oy)	应用是否部署,默认为 true。该属性在设置后不允许修改。若 deploy 设置 为 false,则应用将不部署(包括应用所包含的软件组件、依赖的主机资源 等)。

3、关联关系

关系	说明
storage	应用使用的存储,目前暂未实现。
volume	应用挂载的卷资源,目前仅支持挂载本地卷。
host	定义应用的运行主机,通过 hwpaas.relationships.HostedOn 的关系关联到主 机(hwpaas.nodes.Compute/hwpaas.nodes.ResourceTemplate)上。
service	定义应用使用的服务(实例),通过 hwpaas.relationships.BindsTo 的关系 绑定服务,对象类型是 hwpaas.nodes.Service。
package	定义应用的软件组件,通过 hwpaas.relationships.PackageConsistsOf 关系 指定软件组件(hwpaas.nodes.SoftwareComponent)。
dependency	定义应用的基本依赖关系,对象可以是任意类型的节点,主要用于指定先后顺序。关联关系为 hwpaas.relationships.ConnectsTo。

3、配置示例

```
node templates:
job-app:
type: hwpaas.nodes.Job
properties:
type: container
completions: 3
requirements:
- package:
node: job-soft
relationship: hwpaas.relationships.PackageConsistsOf
```

12.5.4 代理(Agent 应用)

Agent 应用对应于类似于 K8S 里 deamonSet 资源应用。该应用继承了普通应用的属性,并在其基础上



添加了一些限制。

1、模型定义

```
hwpaas.nodes.Agent:
 derived from: hwpaas.nodes.Application
properties:
    type:
      type: string
      description: process/container, application type, now for process(directly on vm/pm) and
container
      default: process
      immutable: true
    deploy:
      type: boolean
      description: deploy this application or not
      default: true
      immutable: true
    packages:
      default: []
    custom spec:
      default: {}
    exclusive:
      type: string
      description: this application need exclusive resources or not, default:(container-->false,
process-->true)
      default: "default"
     resource selector:
      description: defined for select resource
      type: hwpaas.datatypes.resource selector
      default: {}
    bind services:
      description: defined for services which applications binds to
      default: []
      type: hwpaas.datatypes.service_bind
      islist: true
    name:
      type: string
      default: ''
      constrains:
       max length: 24
       regex: (^$) | (^[a-z]([-a-z0-9]*[a-z0-9])?$)
       immutable: true
     certification:
      type: hwpaas.datatypes.Certification
                     annotations:
      default: {}
      type: hwpaas.datatypes.Annotations
      islist: true
      default: []
     upgrade strategy:
      type: hwpaas.datatypes.upgrade strategy
      default: {}
```

2、属性配置



属性	说明
名称(name)	应用的名称,默认为空。名称格式非空时,需要满足 ^[a-z] ([-a-z0-9]*[a-z0-9])?\$的正则条件和最大 24 的长度限制。该属性在 设置后不允许修改。
	 若定义名称,应用相关资源的名称都将使用这个名称。 若使用默认值,应用相关资源的名称将采用"节点名称+随机后缀"的格式。
类型(type)	应用类型,包括进程类型应用(process)和容器类型(container)应用, 默认为 process。应用所包含的软件组件需要与应用的类型一致。
是否部署(dep l oy)	应用是否部署,默认为 true。如果 immutable 设置为 true,该属性 immutable 为 true,设置后不允许修改。若 deploy 设置为 false,则应用将不部署(包 括应用所包含的软件组件、依赖的主机资源等)。
包信息列表 (packages)	定义应用的软件包,目前暂未实现。
自定义属性 (custom_spec)	应用组的用户自定义配置信息,允许用户配置任意自定义属性(属性信息为 json 结构)。在创建应用时,该参数将传递到应用的配置信息中。
资源过滤 (resource_select	资源调度,可以在此处配置亲和性/反亲和性,用于指定应用部署的资源,与 应用之间的亲和性关系。
or)	亲和性部署的结构如下,目前支持 labels(对应 k8s 的 node selector)和 affinities(目前已支持 node affinity,以及 pod 自身的反亲和;尚不支 持配置 k8s 的 pod 级别的 affinity)。当 label 与 affinities 同时配置时, 两种调度策略均生效。
	resource_selector: lables: #数组结构, 默认为空 - key: xx #标签名称 value: #标签值 affinities:
	antiself: true #反亲和性, 默认为 true host: #主机亲和性, 数组结构, 默认为空 - key: aa #主机标签名称 values: [aa, bb, cc] #主机标签值范围 op: In #主机标签匹配策略
绑定服务 (bind_services)	绑定服务,可以在应用中绑定已经创建的服务实例。需要定义绑定服务的关 联关系,并指定服务信息的挂载文件路径。应用可以直接读取挂载路径的文 件,以获取服务的信息。
	bind_services: - bind_path: string #服务实例信息挂载路径



属性	说明
证书	应用可以通过 certification 来挂载证书和 secret。
(certification)	certification: path: string #证书挂载路径 secret: string #证书引用的 secret 名称
注解 (annotations)	用来添加注入自定义的元数据信息,数组形式。每个元素的内容包括下面的 内容
	name: string #自定义元数据名称 value: string #自定义元数据内容
重启策略 (restart_policy)	当应用出现异常后,重启应用时使用的策略,默认值"Always"。也可配置 为"OnFailure"或"Never"。

3、关联关系

关系	说明
storage	应用使用的存储,目前暂未实现。
volume	应用挂载的卷资源,目前仅支持挂载本地卷。
host	定义应用的运行主机,通过 hwpaas.relationships.HostedOn 的关系关联到主 机(hwpaas.nodes.Compute/hwpaas.nodes.ResourceTemplate)上。
service	定义应用使用的服务(实例),通过 hwpaas.relationships.BindsTo 的关系 绑定服务,对象类型是 hwpaas.nodes.Service。
package	定义应用的软件组件,通过 hwpaas.relationships.PackageConsistsOf 关系 指定软件组件(hwpaas.nodes.SoftwareComponent)。
dependency	定义应用的基本依赖关系,对象可以是任意类型的节点,主要用于指定先后顺序。关联关系为 hwpaas.relationships.ConnectsTo。

4、配置示例

```
node templates:
frontend-app:
type: hwpaas.nodes.Agent
properties:
type: container
requirements:
- package:
node: frontend-soft
relationship: hwpaas.relationships.PackageConsistsOf
- host:
node: frontend-cluster
relationship: hwpaas.relationships.HostedOn
```



12.5.5 软件

软件组件用于定义应用所需要的软件,如 docker 镜像、进程类型软件包等信息。软件组件的模型定 义如下:

1、模型定义

```
hwpaas.nodes.SoftwareComponent:
   derived_from: hwpaas.nodes.Root
   properties:
    deploy:
      description: deploy this application or not
      default: true
    package:
      package_type:
        default: process
      deploy:
        description: the package deploy or not
        type: boolean
        default: true
      process spec:
        name:
         default: ''
         description: package info, url or full name
        version:
         default: ''
         description: package version
        url:
         default: ''
        script path:
         default: 'scripts/'
      container spec:
        image:
         default: ''
        command:
         default: []
      probe spec:
        livenessProbe:
         default: {}
         description: probe the liveness, some command or scripts
        readinessProbe:
         default: {}
         description: probe the readiness, some command or scripts
      ingress spec:
        rules:
         default: []
        tls:
         default: []
        tls backend:
         default: false
      resource spec:
         deafult: {}
```

service_spec:
<pre>default: {}</pre>
env:
default: []
secrets:
default: []
ports:
default: []
volumes:
default: []
<pre>package_pull_secret:</pre>
type: string
default: ''
custom spec:
<pre>default: {}</pre>
requirements:
- dependency:
relationship: hwpaas.relationships.DependsOn
node: hwpaas.nodes.SoftwareComponent
interfaces:
Standard:
type: hwpaas.interfaces.AppLifecycle

2、属性配置

天翼**云** e Cloud

1) 属性配置说明

属性	说明
名称(name)	软件组件的名称,默认为空。名称格式非空时,需要满足 ^[a-z]([-a-z0-9]*[a-z0-9])?\$的正则条件和最大 24 的长度限制。该属性在 设置后不允许修改。
	● 若定义名称,应用相关资源的名称都将使用这个名称。
	● 若使用默认值, 应用相关资源的名称将采用"节点名称+随机后缀"的格式。
是否部署	软件组件是否部署。
(deploy)	●false:不部署。
	●true: 部署。
	当应用存在 PackageConsists0f 的关联关系,且依赖这个软件组件时,软件 组件属性设置为 false 会导致拓扑检查失败。此时需要执行强制创建,从应 用中删除这个应用组件的关联关系,才能够正常的创建堆栈。
软件包(package)	用于定义软件组件的主要属性信息,详细请参见错误!未找到引用源。。
自定义属性 (custom_spec)	应用组的用户自定义配置信息,允许用户配置任意自定义属性(属性信息为 json 结构)。在创建应用时,该参数将传递到应用的配置信息中。

2) 软件包参数说明



参数	说明
package_type	软件组件类型,默认为 process。
	●process:进程类
	●container: 容器类
是否部署 (dep l oy)	预留字段,目前暂未实现。
process_spec	进程类属性信息,对应软件包应用特性。包含以下几个字段:
	process_spec: name: 字符串 version: 字符串 url: 字符串 script_path: 字符串
	● name:软件包名称,在 url为空时使用。
	• Version: 私什也版本, 在 uri 为空时使用。
	● scrint nath. 预留字段 目前暂未实现。
container_spec	容器类属性信息,对应容器应用特性。包含以下几个字段:
	container_spec: image: 字符串, 默认为空 command: 字符串数组, 默认为空 securityContext: 数组, 可选 imagePullPolicy: 字符串
	●image: 镜像地址
	● command:容器启动命令,默认为空,使用镜像中自带的启动命令
	● securityContext: 定义了容器的安全选项
	● imagePullPolicy: 拉取镜像策略, 可为 Always, IfNotPresent 或 Never
probe_spec	健康检查项配置,k8s 的探针机制。包含以下几个字段:
	probe_spec: livenessProbe: 存活探针, 默认为空 readinessProbe: 就绪探针, 默认为空
	● livenessProbe: 存活探针
	l i venessProbe 中的参数配置请参见错误! 未找到引用源。。
	●readinessProbe: 就绪探针
	readinessProbe 的参数配置同"livenessProbe",请参见错误!未 到引用源。。
ingress_spec	访问路由配置。包含以下几个字段:
	ingress_spec:

12 参考知识



参数	说明
	rules: 规则数组 tls: 安全配置数组 tls_backend: 是否启用安全传输
	•rules:访问路由的规则配置,每个 rule 结构包含以下几个字段:
	host:字符串 http:结构 paths: http 路径信息数组 - path:路径,字符串格式 backend:路径的后端配置 serviceName:服务的名称 servicePort:服务的端口
	- host: 配置域名,填写 autohost 将自动生成 *.vip.io 域名。
	- http: 配置该域名下具体路由与后端服务的映射。
	- paths: 配置该域名的路径信息,数组格式。
	– path: 域名的路径信息。
	 backend: 域名的后端配置, 对应软件的 service。其中 serviceName 可以不填, 默认即为当前软件所在应用的 service; servicePort 在 nodePort 端口随机分配的情况下, 需要配置成 0。
	●tls:访问路由的tls安全配置。每个结构包含以下几个字段:
	hosts: 字符串数组 secretName: 授权秘钥的名称
	– hosts: 配置域名数组。
	- secretName: 配置这些域名使用的授权秘钥。
	●tls_backend:后端是否开启tls 传输,默认为否。
	(只有在 service 用的是 loadbalancer)
	● ingress_type:可分为 internal 和 external 两种类型, 用于内部或 外部访问。
	• role: 当 type 类型为 external 时有效,标记注册到哪个 ELB 上,可 用取值为: region, tenant 和 data
	● Ioadba1ancer_ip:指定注册到哪个 ELB 实例 IP 地址上。
service_spec	服务配置。包含以下几个字段:
	service_spec: type: 字符串 ports: 端口数组
	●type: 服务类型, 主要包括 NodePort、ClusterIP 和 LoadBalance, 默认为 ClusterIP。
	说明
	若定义了 ingress_spec 字段,不支持 Cluster IP 服务类型,请选择 NodePort 或 LoadBalance。
	●ports:端口数组。包含以下几个字段:
	port: 数字





参数	说明
	name: 字符串 targetPort: 数字 nodePort: 数字 protocol: 协议, TCP/UDP
	- port: 对外暴露的应用监听端口。
	- name: 端口名称, 可选。
	- targetPort: 应用程序内部的监听端口。
	- protocol: 服务的传输协议, 默认 TCP。
	 nodePort: nodePort 类型时有效,端口取值范围可在安装平台时指定,若安装时未修改默认为 30000[~]32767。配置端口时,需要保证 nodePort 参数值全局唯一,否则会因为端口冲突而导致部署应用失败。若不配置,系统使用随机分配的端口。
resource_spec	对资源请求和限制。包含以下几个字段:
	resource_spec: requrest: cpu: 字符串 memory: 字符串 limit: cpu: 字符串 memory: 字符串
	● request: 所申请的资源
	cpu: 单位为 cores. (500m = .5 cores)
	memroy: Memory, 单位为 bytes.(500Gi = 500GiB = 500 * 1024 * 1024 * 1024)
	●limit: 对资源限制
	cpu: CPU, 单位为 cores.(500m = .5 cores)
	memroy: 单位为 bytes.(500Gi = 500GiB = 500 * 1024 * 1024 * 1024)
env	环境变量配置,数组格式。每一条环境变量,都需要配置 name 和 value。
	● name:环境变量名称。
	●value:环境变量值。
secrets	用户自己用到的认证信息,如用户名/密码,AK/SK 等。数组格式,形式 与"env"相同。
	● name: 隐私信息名称。名称字段仅支持小写字母与数字的组合,否则 在安装过程中会失败。
	● value:隐私信息值。将加密保存。
ports	开放端口配置,进程和容器场景配置信息不同。
	●对于进程:
	- ports: 配置端口列表。



参数	说明
	port:进程使用的端口号。
	protocol:必填项,请根据应用的使用场景填写,如 TCP 或 UDP。
	● 对于容器
	– ports: 配置端口列表。
	- containerPort: 容器使用的端口号。
volumes	挂载卷配置。目前仅支持在应用中挂载卷,不支持在软件组件定义中挂 载卷。特别注意的是,如果应用是 process 型应用,则 mountPath 目录 必须与主机目录一样,而且该目录必须在应用中自行创建。结构字段如 下:
	●name: 挂载名称,必填。
	●mountPath: 挂载目标目录,必填。
	● read0n1y: true 表示只读模式; false 表示可读可写模式, 默认为读 写模式。可选。
	●policy: 挂载策略。目前仅做了日志策略,其值为 logs,用于日志收 集与防爆。可选。
	日志策略书写格式如下:
	policy: logs: rotate: Daily
	其中,rotate 可以配置为 Hourly、Daily、Weekly、""。
	 Hourly:每小时扫描一次日志文件,如文件超过 20M,会将其转储 (即压缩到一个历史文件中,在日志所在目录下),并将原文件清空。
	 Daily:每天扫描一次日志文件,如文件超过20M,会将其转储(即 压缩到一个历史文件中,在日志所在目录下),并将原文件清空。
	 Weekly:每周扫描一次日志文件,如文件超过 20M,会将其转储(即 压缩到一个历史文件中,在日志所在目录下),并将原文件清空。
	- "":只做日志收集,不会做日志防爆。
package_pull_se cret	授权秘钥,用于安装软件组件时,从软件/镜像仓库下载获取指定的包/ 镜像。
	● 若为空,系统会根据用户的项目信息自动配置成名称为 project-secret 的秘钥。
	● 若为非空,系统使用用户指定的秘钥。

3) livenessProbe 参数说明

参数	说明
initialDelaySecon ds	探针在定义时长后开始检测



参数	说明
timeoutSeconds	检测超时
periodSeconds	探针检测间隔
successThreshold	检测成功阀值
failureThreshold	检测失败阀值。比如当前 pod 状态为健康,但是健康检查失败,不会 立即刷新 pod 健康状态。当连续失败次数达到 failure 阈值(比如阈 值是 5),要连续 5 次健康检查失败才会把 pod 状态置为不健康。
exec	
command	探针执行的健康检查命令,字符串数组格式。
httpGet	
path	探针发送 http 请求的路径。
port	探针发送 http 请求的端口。
host	探针发送 http 请求的主机。

3、关联关系

关系	说明
dependency	定义软件组件之间的依赖关系,主要用于控制安装的先后顺序。

4、配置示例

1) 容器类软件组件

```
node templates:
 redis-app:
   type: hwpaas.nodes.Application
   properties:
    type: container
   requirements:
     - package:
       node: redis-soft
       relationship: hwpaas.relationships.PackageConsistsOf
     - host:
       node: redis-cluster
       relationship: hwpaas.relationships.HostedOn
 redis-soft:
   type: hwpaas.nodes.SoftwareComponent
   properties:
    package:
```



```
package_type: container
container_spec:
  image: {get_input: image_id}
```

```
redis-cluster:
  type: hwpaas.nodes.Cluster
  properties:
    host_type: container
```

2) 进程类应用软件组件

```
node templates:
 hello-app:
   type: hwpaas.nodes.AppGroup
   requirements:
     - member:
        node: hello-component
        relationship: hwpaas.relationships.ConsistsOf
 hello-component:
   type: hwpaas.nodes.Application
   properties:
    type: process
   requirements:
     - host:
       node: hello-cluster
       relationship: hwpaas.relationships.HostedOn
     - package:
        node: hello-package
        relationship: hwpaas.relationships.PackageConsistsOf
 hello-package:
   type: hwpaas.nodes.SoftwareComponent
   properties:
    package:
      process spec:
        url: hello-V1-Linux-x64.tar.gz
        version: V1
      env:
        - name: BIND SERVICES
        value: default
 hello-cluster:
   type: hwpaas.nodes.Cluster
   properties:
```

```
host_type: vm
```

3) 带有健康检查的软件组件

```
node templates:
nginx-soft:
type: hwpaas.nodes.SoftwareComponent
properties:
package:
package_type: container
```

```
container_spec:
  image: {get_input: image_id}
probe_spec:
  livenessProbe:
   httpGet:
    path: /
    port: 80
ports:
   - containerPort: 80
```

4) 带有服务开放的软件组件

天翼 Cloud

```
node_templates:
frontend-soft:
type: hwpaas.nodes.SoftwareComponent
properties:
package:
package type: container
container spec:
image: {get input: image id}
service spec:
type: NodePort
ports:
- port: 80
nodePort: 30002
```

5) 带有访问路由的软件组件

```
node templates:
hello-package:
   type: hwpaas.nodes.SoftwareComponent
   properties:
    package:
      process spec:
       name: {get input: package}
       version: V1
      env:
        - name: BIND SERVICES
         value: default
      ports:
       - port: 9876
         protocol: TCP
      service spec:
       type: NodePort
       ports:
         - port: 9876
      ingress spec:
        tls:
        - hosts:
         - autohost
         secretName: default-ingress-ssl
        rules:
        - host: autohost
         http:
           paths:
```

- path: /hello backend: servicePort: 0 tls backend: true

6) 带有环境变量的软件组件

```
node_templates:
frontend-soft:
type: hwpaas.nodes.SoftwareComponent
properties:
package:
package_type: container
env:
- name: BACKEND
value: redis
container spec:
image: {get_input: image_id}
```

12.5.6 集群

大異 Cloud

1、模型定义

集群的定义如下:

```
hwpaas.nodes.Cluster:
   derived from: hwpaas.nodes.ResourceTemplate
   properties:
    name:
      type: string
      description: cluster name in k8s
      required: true
    namespace:
      type: string
      description: namespace for deploy in k8s
      required: true
      immutable: true
   capabilities:
    host:
     type:hwpaas.capabilities.Container
    os:
     type:hwpaas.capabilities.OperatingSystem
    storage:
      type:hwpaas.capabilities.Attachment
   requirements:
     - port:
      capability: hwpaas.capabilities.PortBindable
      relationship:hwpaas.relationships.BindsToPort
      node: hwpaas.nodes.Port
   interfaces:
    Standard:
    type:hwpaas.interfaces.ResourceLifecycle
```





2、属性配置

属性	说明
名称(name)	集群名称。
命名空间(namespace)	集群的命名空间。

3、关联关系

关联关系	说明
port	可以用来将 cluster 和一个 port 进行绑定,从而获取 IP。

4、配置示例

```
frontend-host:
   type: hwpaas.nodes.Cluster
   properties:
    name: cluster-1 //使用已有的集群
   namespace: default
frontend-app: //应用 hostedOn 集群
   type: hwpaas.nodes.Application
   properties:
    type: container
   requirements:
        - host:
            node: frontend-host
```

12.5.7 生命周期脚本定义

1、脚本定义

平台应用支持生命周期的定义。容器中命令本身已制作在容器中,是不需要额外定义的。此处介绍的 生命周期脚本定义,主要针对的是进程类应用的脚本定义。进程类应用的软件组件,软件包中存在安装脚 本,平台对软件包的脚本存放有建议的目录。

应用生命周期事件,将触发执行应用生命周期脚本。生命周期脚本定义在两个层次,即应用和软件包上,在相应的生命周期执行对应的脚本。脚本可支持 shell/bat 等格式,后续也将扩展支持 salt/chef 等 配置工具脚本。

在模板中定义的属性等信息,通过环境变量的形式导入到应用中,在生命周期脚本中进行获取得到。 环境变量的名称,一般为{Node}_{Property},即定义在 node 下的属性信息,跨 node 的环境变量,也通 过此种方式得到。

```
hwpaas.interfaces.AppLifecycle:
 derived from: hwpaas.interfaces.NodeLifecycle
 actions:
   create:
    command: '[ ! -s scripts/create.sh ] || sh scripts/create.sh'
    fail-command: ''
   configure:
    command: '[ ! -s scripts/configure.sh ] || sh scripts/configure.sh'
    fail-command: "
   start:
    command: '[ ! -s scripts/start.sh ] || sh scripts/start.sh'
    fail-command: ''
   poststart:
    command: '[ ! -s scripts/poststart.sh ] || sh scripts/poststart.sh'
    fail-command: ''
   prestop:
    command: '[ ! -s scripts/prestop.sh ] || sh scripts/prestop.sh'
    fail-command: ''
   stop:
    command: '[ ! -s scripts/stop.sh ] || sh scripts/stop.sh'
    fail-command: ''
   delete:
    command: '[ ! -s scripts/delete.sh ] || sh scripts/delete.sh'
    fail-command: ''
   restart:
    command: '[ ! -s scripts/restart.sh ] || sh scripts/restart.sh'
    fail-command: ''
   upgrade:
    command: '[ ! -s scripts/upgrade.sh ] || sh scripts/upgrade.sh'
    fail-command: "
   rollback:
    command: '[ ! -s scripts/rollback.sh ] || sh scripts/rollback.sh'
    fail-command: ''
   reconfigure:
    command: '[ ! -s scripts/reconfigure.sh ] || sh scripts/reconfigure.sh'
    fail-command: ''
   active:
     command: '[ ! -s scripts/active.sh ] || sh scripts/active.sh'
    fail-command: ''
   standby:
    command: '[ ! -s scripts/standby.sh ] || sh scripts/standby.sh'
    fail-command: "
   scale:
    command: '[ ! -s scripts/scale.sh ] || sh scripts/scale.sh'
    fail-command: ''
   update:
     command: '[ ! -s scripts/update.sh ] || sh scripts/update.sh'
    fail-command: ''
 plans:
   install:
    actions: [create]
    fail-actions: []
   configure:
    actions: [configure]
```

Cloud

interfaces:

fail-actions: [] start: actions: [start] fail-actions: [] poststart: actions: [poststart] fail-actions: [] prestop: actions: [prestop] fail-actions: [] stop: actions: [stop] fail-actions: [] uninstall: actions: [delete] fail-actions: [] reconfigure: actions: [reconfigure] fail-actions: [] upgrade: actions: [backup, upgrade, start, poststart] fail-actions: [] update: actions: [update] fail-actions: [] rollback: actions: [rollback] fail-actions: []

2、配置示例

大翼口 e Cloud

```
#定义生命周期
lifecycles:
 my backend lifecycle:
   derived from: hwpaas.interfaces.AppLifecycle
   actions: #定义动作,与实际的脚本相对应
    install-this:
     command: 'install.sh' #脚本执行命令
                        #脚本失败执行命令
     fail-command: ''
    config:
     command: 'config.sh'
     fail-command: ''
    start app:
      command: 'start app.sh'
      fail-command: ''
    stop:
     command: 'stop app.sh'
     fail-command: ''
    uninstall:
      command: 'delete app.sh'
      fail-command: ''
    rollback status:
     command: 'rollback.sh'
      fail-command: ''
   plans: #生命周期计划,引用上述定义的 action
```

create: actions: [install-this, config] #action 数组, 串行执行 fail-actions: [] start: actions: [start_app] fail-actions: [] stop: actions: [stop] fail-actions: [] delete: actions: [uninstall] fail-actions: [] #使用生命周期 node templates: backend: type: hwpaas.nodes.Application properties: template files: - resolv.conf: path: resolv.conf # 模板文件在相对路径(packages/app_name/resources/templates/resole.conf) target_path: /etc/resolv.conf #目标文件路径 values: #模板待替换项 nameserver: 10.10.10.10 lifecycles: Standard: #覆盖平台默认提供的生命周期 type: my backend lifecycle

12.5.8 输入

大異口 e Cloud

1、概念

输入是指在实例化模板时输入的参数,为可选项。在部署时用户可根据实际需求修改输入参数的默认

值。

2、属性配置

属性	说明
名称	输入参数的名称。取值范围:字符串,长度1 [~] 64,由英文字母、数字、中划 线(-)和下划线(_)组成,且必须以字母开头。
类型(type)	目前支持字符串、整数、浮点数,布尔、密码、密钥、IP 和枚举。
描述 (description)	输入参数描述信息。取值范围:字符串,长度1 [~] 255,由英文字母、数字和 英文符号组成。其中,英文符号包括- [~] !@#\$% [^] &*() _+={}<>?,.;:`'"\/[]和 空格。
默认值(default)	输入参数默认值。取值范围:字符串,长度1 [~] 8192,由英文字母、数字和英 文符号组成。其中,英文符号包括- [~] !@#\$% [^] &*() _+={}<>?,.;:`'"\/[]和空



属性	说明
	格。
标签(Iabel)	输入参数的分类标记。取值范围:字符串,长度 1~255
约束(constrains)	输入参数的有效值条件,总共下面几种有效值范围
	 equal:等于,支持数字、字符串类型的比较 in_range:数字的范围,支持整数的有效性比较 valid_values:有效值集合,数组格式,样例为[a, b, ccc] length:字符串长度 regex:正则表达式
高级参数 (advanced)	输入参数是否为高级参数。取值范围: true/false
隐藏字段 (invisible)	输入参数的值是否隐藏不可见。取值范围: true/false

3、源代码

```
inputs:
    user:
    default: root
    type: string
    password:
    default: xxxxxx //用户输入的密码
    type: string
    user_password:
    type: secret //隐私类型字段,界面上不回显。后端自动加密,在获取时解密
```

12.5.9 输出(高级功能)

1、概念

输出是指模板生成堆栈实际部署完成后的输出参数,为可选项。

2、属性配置

属性	说明
名称	输出参数的名称。取值范围:字符串,长度1 [~] 64,由英文字母、 数字、中划线(-)和下划线(_)组成,且必须以字母开头。
描述	输出参数描述信息。取值范围:字符串,长度1 [~] 255,由英文字 母、数字和英文符号组成。其中,英文符号包括 - [~] !@#\$% [^] &*() _+={}<>?,.;:`'"\/[]和空格。



属性	说明
默认值	输出参数默认值。取值范围:字符串,长度1 [~] 8192,由英文字 母、数字和英文符号组成。其中,英文符号包括 - [~] !@#\$% [^] &*() _+={}<>?,.;:`'"\/[]和空格。

3、已支持的输出参数(动态 attribute)

以下的动态 attribute 属性,只有编排到了相关的资源才能获取得到,否则得到的信息为空。

1) 应用:

A、应用的 IP 地址: get_attribute: [{app-name}, ips], 获取 pod 运行的主机地址

B、应用的服务端口信息(nodePort): get_attribute: [{app-name}, Service, ports, 0, nodePort]

C、应用的访问路由(http): get_attribute: [{app-name},

Ingress, annotations, ingress. kubernetes. io/external-address]

D、应用的访问路由(https): get_attribute: [{app-name},

Ingress, annotations, ingress. kubernetes. io/external-ssl-address]

2)资源:

A、端口的 IP 地址: get_attribute: [{port-name}, Port, fixed_ips, 0, ip_address]

B、端口的 IP 对地址:get_attribute: [{port-name}, Port, allowed_address_pairs, 0, ip_address]

C、Cluster 地址: get_attribute: [{cluster-name}, cluster, {vmindex}, server, addresses, {network}, {netindex}, addr]。"vmindex"是多实例情况下虚拟机的序号,从0开始计数;"netindex" 是 cluster 所在网络的网络名称, " netindex"是多网络平面情况下的网络序号,从0开始计数。 用户也可以直接使用 get_attribute: [{cluster-name}, cluster]获取 Cluster 的所有信息,其中就 包括 Cluster 所有实例的 IP 表。

12.6 ConfigMap 类型资源描述文件配置说明

ConfigMap 资源文件支持 json 和 yaml 两种格式,且文件大小不得超过 2MB。

1、json 格式





文件名称为 configmap. json, 配置示例如下:

```
{
 "kind": "ConfigMap",
 "apiVersion": "v1",
 "metadata": {
  "name": "paas-broker-app-017",
  "namespace": "lcqtest",
  "enable": true
 },
 "data": {
   "context":
"{\"applicationComponent\":{\"properties\":{\"custom spec\":{}},\"node name\":\"paas-broker-app\"
,\"stack id\":\"0177eae1-89d3-cb8a-1f94-c0feb7e91d7b\"},\"softwareComponents\":[{\"properties\":{
\"custom spec\":{}},\"node name\":\"paas-broker\",\"stack id\":\"0177eae1-89d3-cb8a-1f94-c0feb7e9
1d7b\"}]}"
}
}
```

2、yaml 格式

文件名称为 configmap. yaml, 配置示例如下:

apiVersion: v1
kind: ConfigMap
metadata:
 name: test-configmap
data:
 data-1: value-1
 data-2: value-2