



伯俊软件  
WWW.BURGEON.CN



伯俊软件  
WWW.BURGEON.CN

## 关于此文件

文件名称	阿里容器使用文档	文件编号	111
文件对象	伯俊全体员工		
拟文	许新帅	审核人	姚静
发文部门	平台框架部门	发文日期	2018/11/20

### 发放修改控制

版本	发布人	修改记录	发布日期
001	许新帅	新建文档	2018/11/22

# 目录

---

---

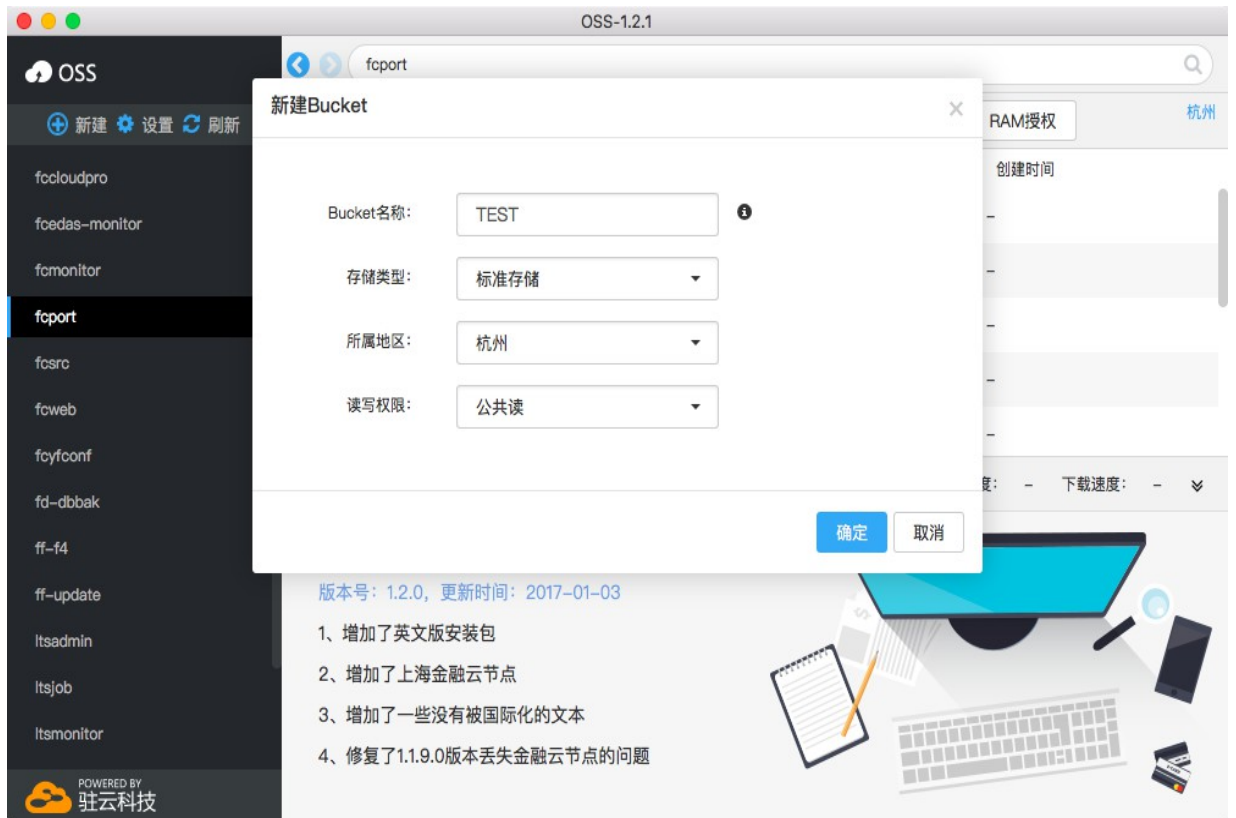
## 1. 简介

容器服务提供高性能可伸缩的容器应用管理服务，支持用 Docker 和 Kubernetes 进行容器化应用的生命周期管理，提供多种应用发布方式和持续交付能力并支持微服务架构。容器服务简化了容器管理集群的搭建工作，整合了阿里云虚拟化、存储、网络和安全能力，打造云端最佳容器运行环境。

## 2. OSS 存储

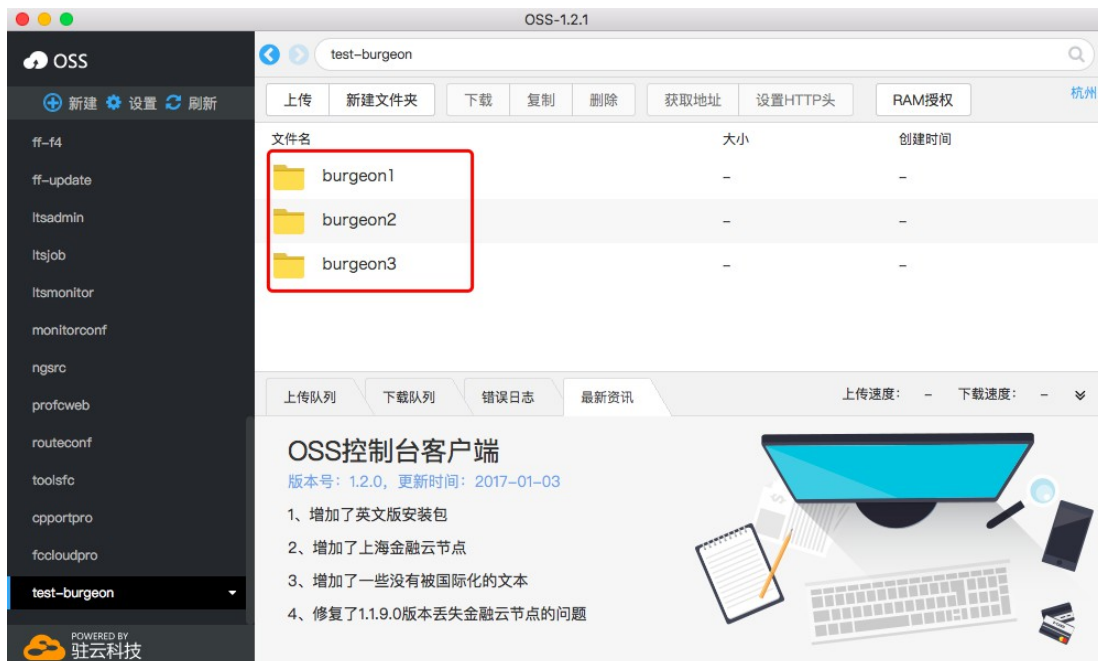
### 1. 新建 bucket

在 OSS 客户端创建 bucket，为了节省 bucket 数量（每个节点最多可创建 30 个）建议在 bucket 的二级文件夹目录中创建自己的应用。



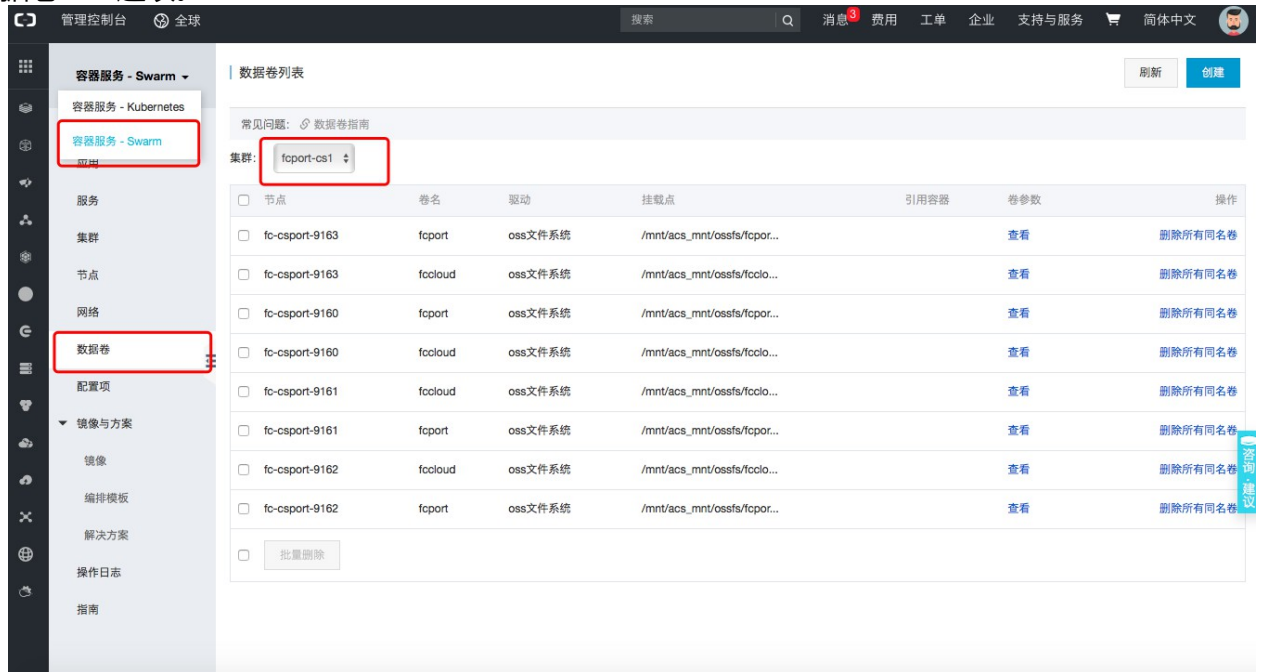
## 2. 创建应用文件夹

在新建 bucket 中新建文件夹，创建自己应用，在应用文件夹中拷贝进去自己的相应应用。



## 3. 创建数据卷

1. 打开阿里云容器服务控制台，选择 swarm 容器服务，选择应用需要部署的集群，切换到数据卷 tab 选项。



2. 选择创建数据卷，填入相应的账号信息（数据卷名建议和之前创建的 BUCKET 名称一致），选择之前创建的 bucket。

The 'Create Data Volume' (创建数据卷) dialog box is shown. It has a title bar with a close button. The form includes the following fields and options:

- 数据卷类型 (Data Volume Type): ☒ OSS, ☐ NAS, ☐ 云盘
- 数据卷名 (Data Volume Name): test-burgeon
- AccessKey ID: [Redacted]
- AccessKey Secret: [Redacted]
- 可选参数 (Optional Parameters): ☐ allow\_other, ☐ noxattr
- 更多参数 (More Parameters): [Empty text box]
- 更多参数的填写格式可以 参考该文档 (The format for filling in more parameters can be referred to this document)
- 注意: 只有 volume\_driver 在 0.7 版本及以上的集群才支持该参数, 您可以到 "应用" 列表中找到 acsvolumedriver 应用, 然后在其详情页的服务列表中查看 volumedriver 服务的镜像版本, 如果是 0.7 以下的话, 请升级系统服务 (Note: Only clusters with volume\_driver version 0.7 and above support this parameter. You can find the acsvolumedriver application in the "Applications" list, and then check the image version of the volumedriver service in its detail page. If it is below 0.7, please upgrade the system service.)
- Bucket ID: 选择Bucket (Select Bucket) - This button is highlighted with a red box.
- 访问域名 (Access Domain): ☒ 内网域名, ☐ 外网域名, ☐ vpc域名
- 文件缓存 (File Cache): ☐ 打开, ☒ 关闭
- Buttons: 创建 (Create), 取消 (Cancel)

fcport	oss-cn-hangzhou	<a href="#">选择</a>
fcsrc	oss-cn-hangzhou	<a href="#">选择</a>
fcweb	oss-cn-hangzhou	<a href="#">选择</a>
fcyfconf	oss-cn-hangzhou	<a href="#">选择</a>
fd-dbbak	oss-cn-hangzhou	<a href="#">选择</a>
ff-f4	oss-cn-hangzhou	<a href="#">选择</a>
ff-update	oss-cn-shanghai	<a href="#">选择</a>
ltsadmin	oss-cn-hangzhou	<a href="#">选择</a>
ltsjob	oss-cn-hangzhou	<a href="#">选择</a>
ltsmonitor	oss-cn-hangzhou	<a href="#">选择</a>
monitorconf	oss-cn-hangzhou	<a href="#">选择</a>
ngsrc	oss-cn-hangzhou	<a href="#">选择</a>
profcweb	oss-cn-hangzhou	<a href="#">选择</a>
routeconf	oss-cn-hangzhou	<a href="#">选择</a>
test-burgeon	oss-cn-hangzhou	<a href="#">选择</a>
toolsfc	oss-cn-hangzhou	<a href="#">选择</a>

## 4. 创建编排模版

1. 选择容器服务-swarm,选择编排模版 Tab 页面,创建编排模版。

容器服务 - Swarm

编排列表

示例编排

我的编排

概览

应用

服务

集群

节点

网络

数据卷

配置项

镜像与方案

镜像

编排模板


解决方案

操作日志

指南


搜索

创建

**Redis-cluster-with-Sentinel**  
详情请参考云栖社区文章: [https://yq.a...](https://yq.aliyun.com/articles/111111)  
[详情](#) V1


服务: 镜像  
master: redis:3  
slave: redis:3  
sentinel: registry.aliyuncs.com/acs-sample/redis-sentinel:3

创建应用 →

**jstorm**  
部署JStorm及依赖的zookeeper。请将...  
[详情](#) V1


服务: 镜像  
zk1: mesoscloud/zookeeper:3.4.8-ubuntu  
zk2: mesoscloud/zookeeper:3.4.8-ubuntu  
zk3: mesoscloud/zookeeper:3.4.8-ubuntu  
nimbus: registry.cn-hangzhou.aliyuncs.com/jstorm-docker/jstorm  
supervisor: registry.cn-hangzhou.aliyuncs.com/jstorm-docker/jstorm

创建应用 →

**wordpress-for-composev3**  
wordpress for composev3  
[详情](#) V3


服务: 镜像  
web: wordpress:4  
mysql: mysql:5.7

创建应用 →

**jenkins**  
可选版本号: 2.60.3, 2.32.2, 1...  
[详情](#) V1

服务: 镜像  
jenkins: registry.cn-hangzhou.aliyuncs.com/acs-sample/jenkins:2.60.3  
slave-golang: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-golang  
slave-java: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-java  
slave-nodejs: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-nodejs  
slave-python: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-python  
slave-php: registry.aliyuncs.com/acs-sample/jenkins-slave-dind-php

创建应用 →

**yunqi-wordpress-view**  
[详情](#) V1

服务: 镜像  
wordpress-view: registry.aliyuncs.com/acs-sample/wordpress-view:yunqi

创建应用 →

2.输入编排模版名称和内容（模版内容参考文档后面例子），点击创建编排，编排模版创建完

成。  
| 创建编排

\* 名称: raincloud  
名称为1-64个字符，可包含数字、汉字、英文字符，或“-”

描述:

内容:

```
1 raincloud-default:
2   image: 'registry-vpc.cn-hangzhou.aliyuncs.com/raincloud/tomcat:latest'
3   ports:
4     - 8080/tcp
5   restart: always
6   environment:
7     - java_home=/acs/user/jdk
8     - jre_home=/acs/user/jdk/jre
9     - 'classpath=/acs/user/jdk/lib/dt.jar:/acs/user/jdk/lib/tools.jar:/acs/user/jdk/jre/lib'
10  labels:
11    aliyun.probe.initial_delay_seconds: '1.50'
12    aliyun.probe.timeout_seconds: '60'
13    aliyun.probe.cmd: '/acs/bin/check'
14    aliyun.log.timestamp: 'true'
15    aliyun.log_store_dbstdout: stdout
16    aliyun.scale: '1'
17  volumes:
18    - /acs/code
19    - /etc/localtime:/etc/localtime:ro
20    - /etc/timezone:/etc/timezone:ro
21
```

利用编排模板您可以定义和部署多容器应用，支持Docker Compose格式。详情请参见 <https://docs.docker.com/compose/>

新增服务 创建编排 取消

包含服务
服务名: raincloud-default 镜像: registry-vpc.cn-hangzhou.aliyuncs.com/raincloud/tomcat:latest <a href="#">编辑</a> <a href="#">删除</a>
新增服务

## 5. 创建应用

1. 切换到应用 tab 页，选择创建应用，填入相应的应用信息。

容器服务 - Swarm

常见问题: [限制容器的资源](#) [高可用性调度](#) [通过镜像创建Nginx](#) [通过编排模板创建WordPress](#) [编排模板说明](#) [标签说明](#)

应用基本信息 应用配置 创建完成

应用名称: burgeon1  
名称为1-64个字符，可包含数字、英文字符，或“-”，且不能以“-”开头

应用版本: 1.0

部署集群: fcpport-cs1

默认更新策略: 标准发布

副本数量: 1

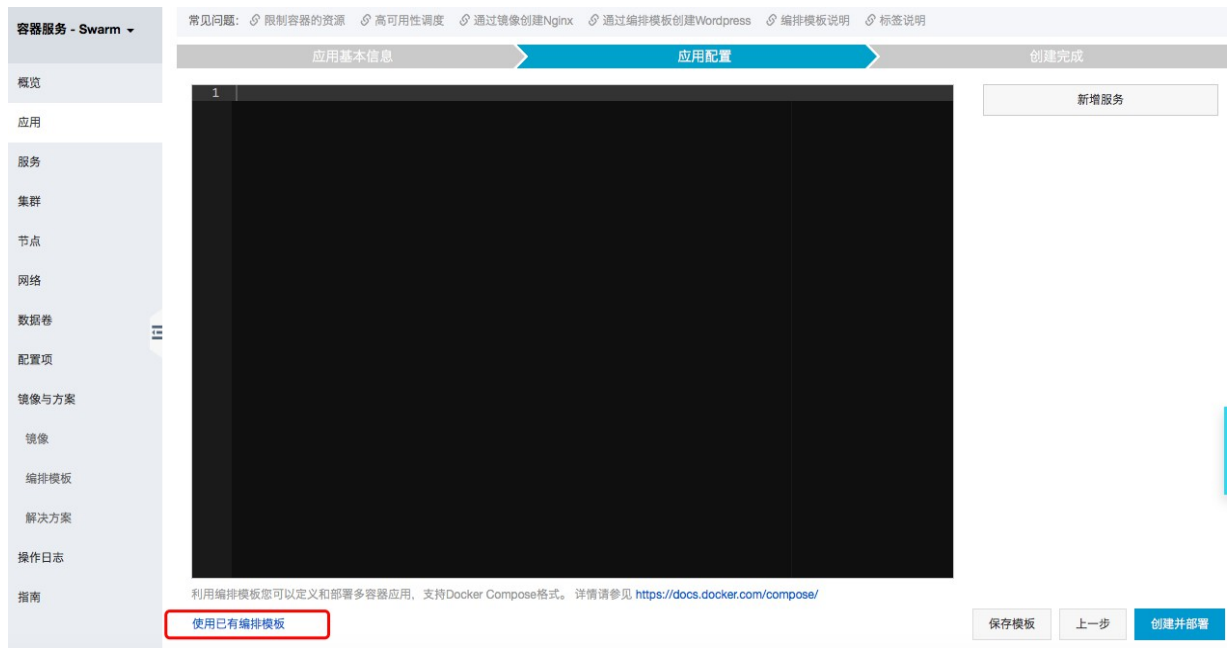
应用描述: 测试应用

☐ 检查最新Docker镜像

使用镜像创建 使用编排模板创建

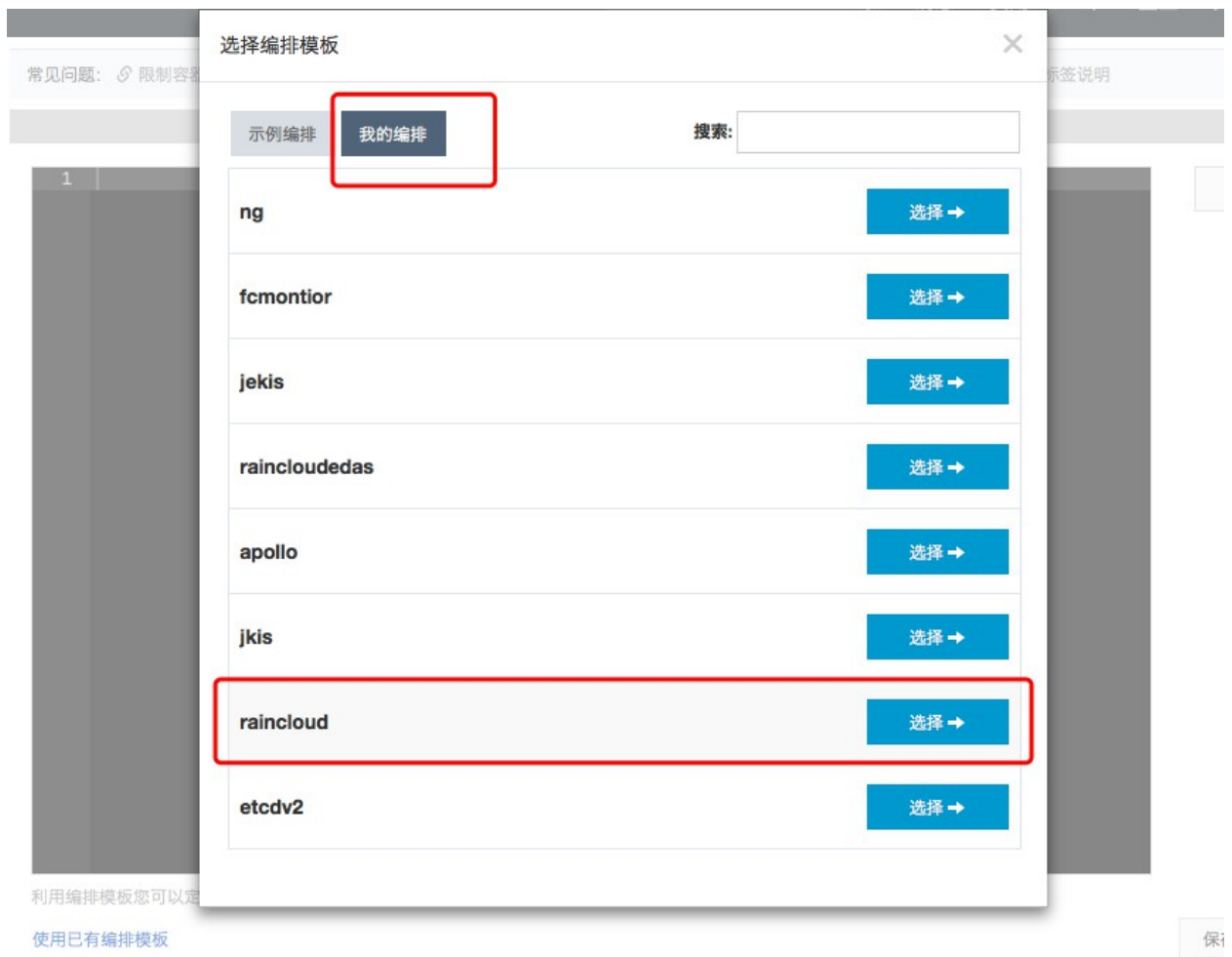
2. 选择使用编排模板创建（可以直接输入模板创建并部署，模板参照文档后例子，本文档寻

那呢已有的编排模版)，选择已有编排模版

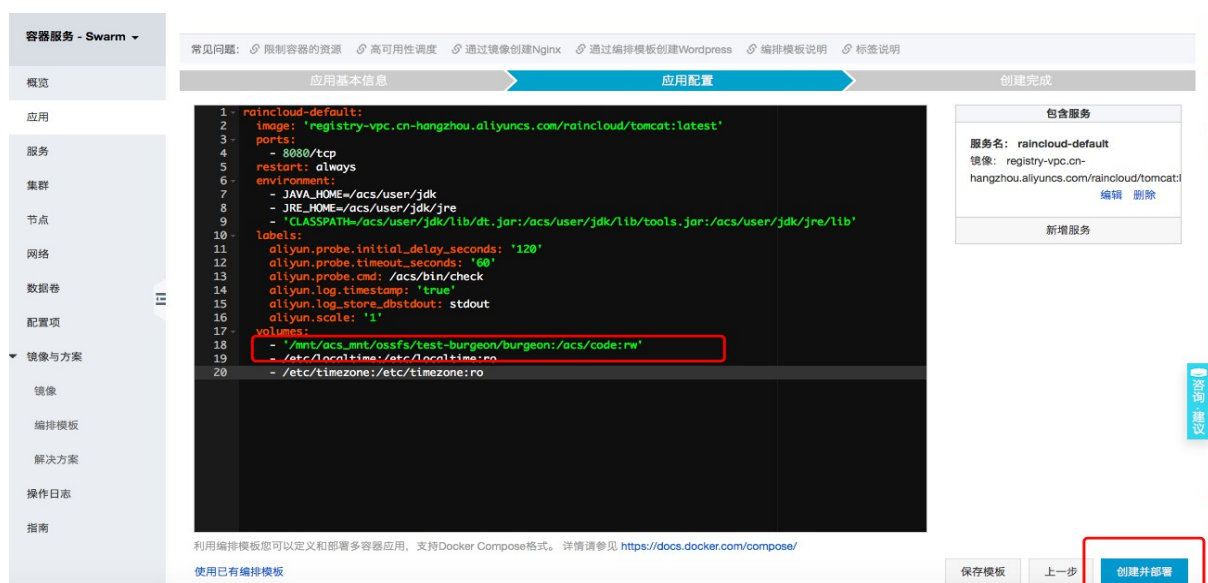


3.切换到我的编排 tab，选择 raincloud





4. 创建并部署，volumes 标签下写入：`-'/mnt/acs_mnt/ossfs/test-burgeon/burgeon1:/acs/code:rw'`，（或将本地磁盘下的应用拷贝到容器目录下）将数据卷指定的应用拷贝到对应的容器目录下，创建并部署。



5. 应用创建完成

容器服务 - Swarm	应用列表	刷新	创建应用
概览	常见问题: 如何创建应用 变更应用配置 简单路由蓝绿发布策略 容器弹性伸缩		
应用	集群: fcport-cs1 <input checked="" type="checkbox"/> 隐藏系统应用 <input type="checkbox"/> 隐藏离线应用 <input type="checkbox"/> 隐藏在线应用	名称: <input type="text"/>	<input type="text"/>
服务			
集群			
节点			
网络			
数据卷			
配置项			
镜像与方案			
镜像			
编排模板			
解决方案			
操作日志			
指南			

应用名称	描述	状态	容器状态	创建时间	更新时间	操作
burgeon1	测试应用	运行中	运行中:1 停止:0	2018-11-23 11:12:05	2018-11-23 11:12:06	停止   变更配置   删除   重新部署   事件
cpport-qa	cpport测试	运行中	运行中:2 停止:0	2018-11-14 09:06:03	2018-11-14 10:57:02	停止   变更配置   删除   重新部署   事件
dlport-qa	dlport 测试	运行中	运行中:1 停止:0	2018-11-14 09:10:44	2018-11-17 12:49:18	停止   变更配置   删除   重新部署   事件
fccloud-qa	fccloud 测试	运行中	运行中:1 停止:0	2018-11-14 13:06:53	2018-11-14 13:06:54	停止   变更配置   删除   重新部署   事件
fccloud-qa	fccloud 测试	运行中	运行中:1 停止:0	2018-11-14 13:22:53	2018-11-14 13:34:14	停止   变更配置   删除   重新部署   事件
posport-qa	posport 测试	运行中	运行中:1 停止:0	2018-11-14 11:01:16	2018-11-14 11:01:17	停止   变更配置   删除   重新部署   事件
scport-qa	scport 测试	运行中	运行中:1 停止:0	2018-11-14 10:59:54	2018-11-14 11:02:37	停止   变更配置   删除   重新部署   事件

共有 11 条, 每页显示: 15 条

## 6. 对应应用详细信息

应用: burgeon1

刷新

基本信息

应用名称: burgeon1

创建时间: 2018-11-23 11:12:05

更新时间: 2018-11-23 11:12:06

所在集群: fcport-cs1

触发器 1.每种类型的触发器只能创建1个

创建触发器

目前没有任何触发器, 点击右上角按钮创建触发器

服务列表

容器列表

日志

事件

路由列表

名称/ID	状态	健康检测	镜像	端口	容器IP	节点IP	操作
burgeon1_rainclo... 4b95e43c249b5660...	running	正常	registry-vpc-cn-... sha256:0d7b139de...	172.16.9.162:1028->8080/tcp	10.227.89.5	172.16.9.162	删除   停止   监控   日志   远程终端

## 7. 编排模版示例

后端:

raincloud-default:

image: 'registry-vpc.cn-hangzhou.aliyuncs.com/raincloud/tomcat:latest'

ports:

- 8080/tcp

restart: always

environment:

- java\_home=/acs/user/jdk

- jre\_home=/acs/user/jdk/jre

- 'classpath=/acs/user/jdk/lib/dt.jar:/acs/user/jdk/lib/tools.jar:/acs/user/jdk/jre/lib'

labels:

aliyun.probe.initial\_delay\_seconds: '120'

aliyun.probe.timeout\_seconds: '60'

aliyun.probe.cmd: /acs/bin/check

```
aliyun.log.timestamp: 'true'  
aliyun.log_store_dbstdout: stdout  
aliyun.scale: '1'
```

volumes:

- /acs/code
- /etc/localtime:/etc/localtime:ro
- /etc/timezone:/etc/timezone:ro

前端:

ngcs:

```
image: 'registry-vpc.cn-hangzhou.aliyuncs.com/dtstack/tengine:2.1.1'
```

ports:

- 80/tcp

restart: always

expose:

- 443/tcp

labels:

```
aliyun.scale: '1'
```

volumes:

- /opt/tengine/logs
- 'b2b:/opt/tengine/www:rw'
- '/opt/tengine/conf/conf.d:/opt/tengine/conf/conf.d:rw'
- /opt/tengine/conf/certs