 【**总经办[2018]第111号】**



根据国际版权保护法律，文档和软件未经上海伯俊软件科技有限公司及南京希曼软件技术有限公司授权，不得整篇或部分复制、影印、复印、翻译、修改。这份资料包含商业机密，限于内部有关人员的阅读，仅仅用于沟通信息的目的，不得向外散布。

文中提到的其他产品或公司名字可能是其他相应的所有者的商标。

**阿里容器使用文档**

**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 4

2. OSS存储 4

3. 创建数据卷 5

4. 创建编排模版 7

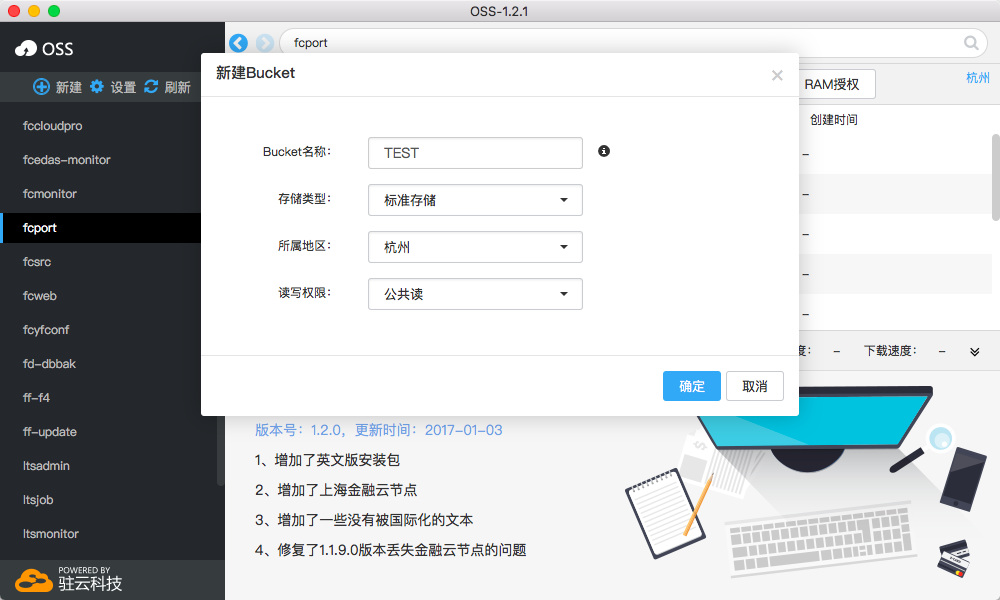
5. 创建应用 8

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。

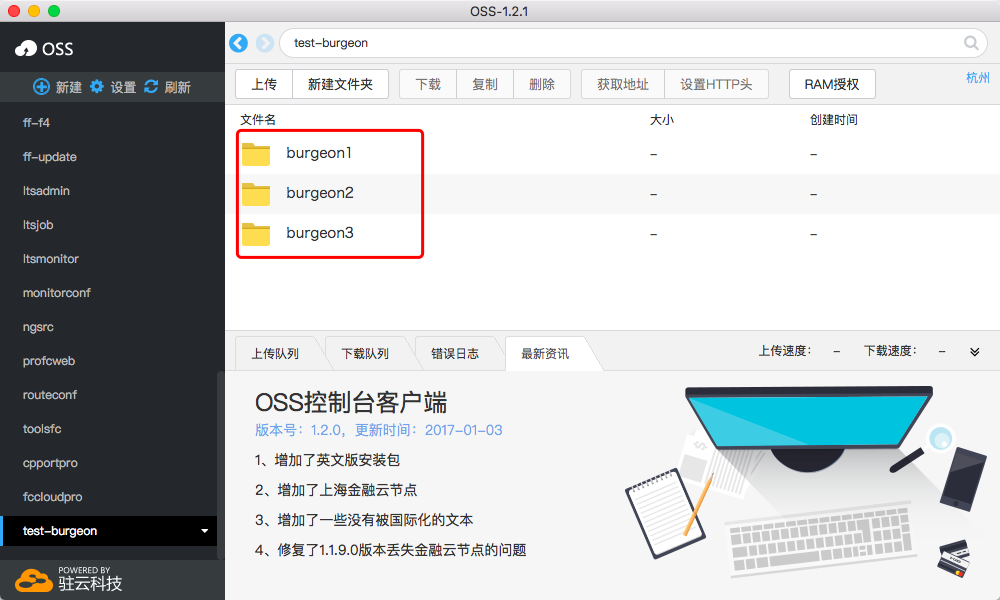
1. OSS存储
2. 新建bucket

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。



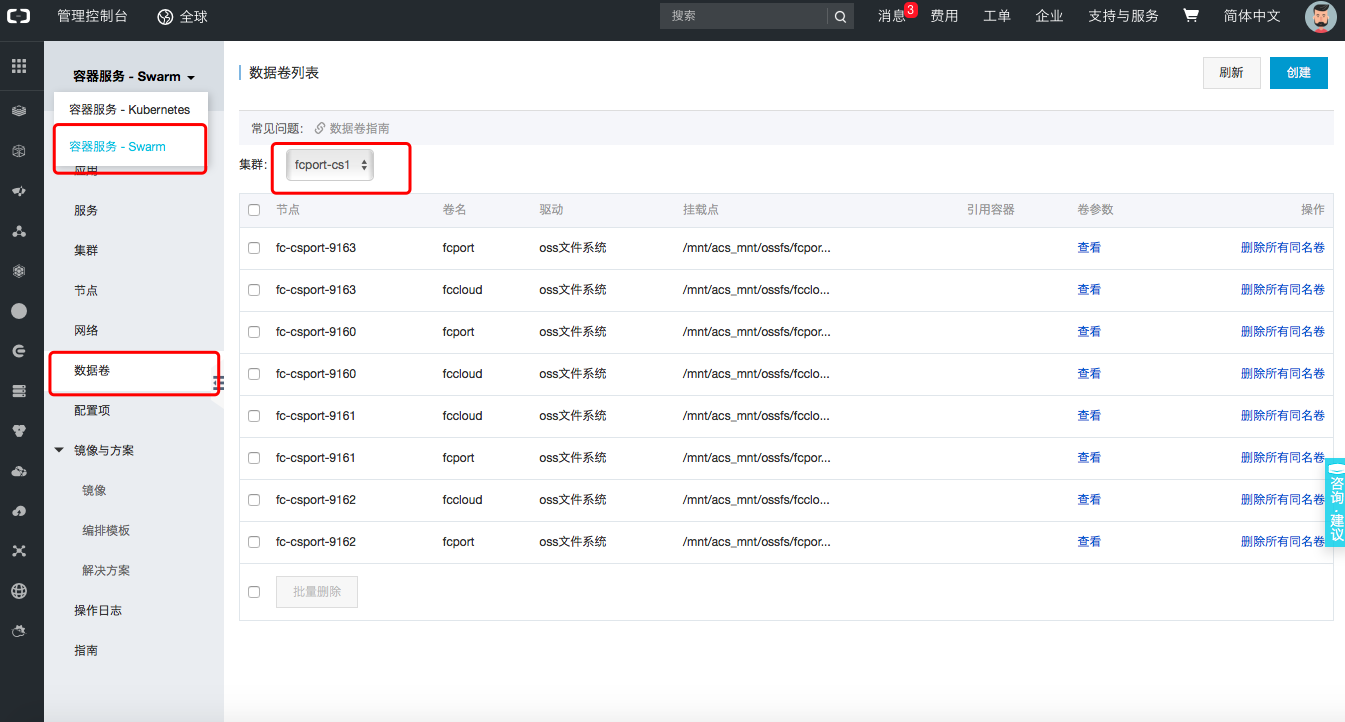
1. 创建应用文件夹

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。

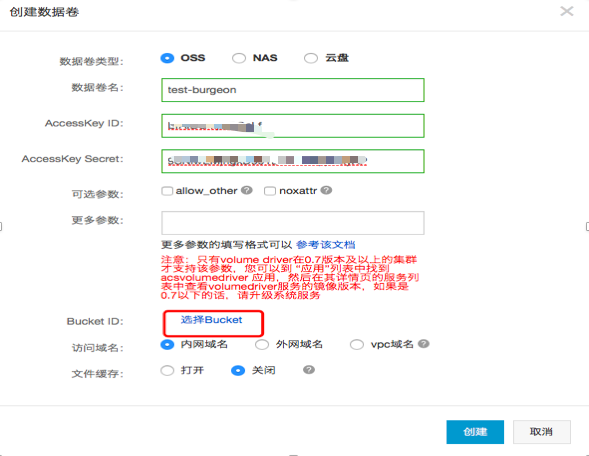


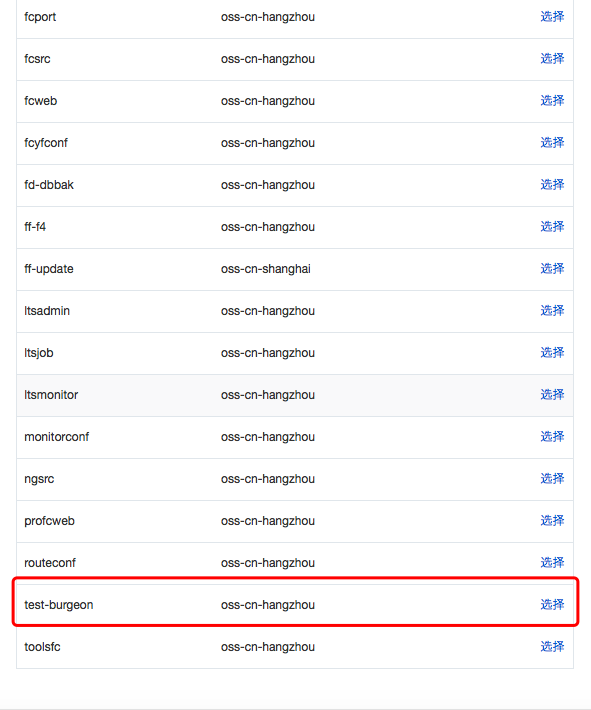
1. 创建数据卷

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。

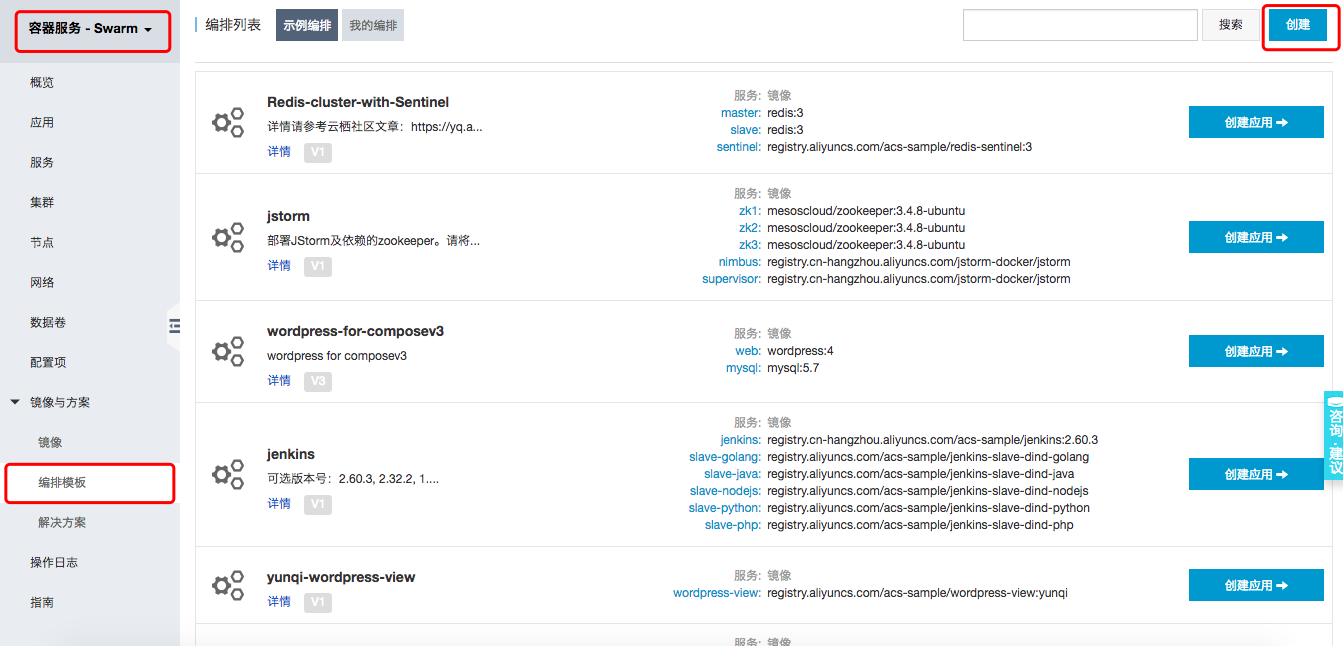


2.选择创建数据卷，填入相应的账号信息（数据卷名建议和之前创建的bucket名称一致），选择之前创建的bucket。

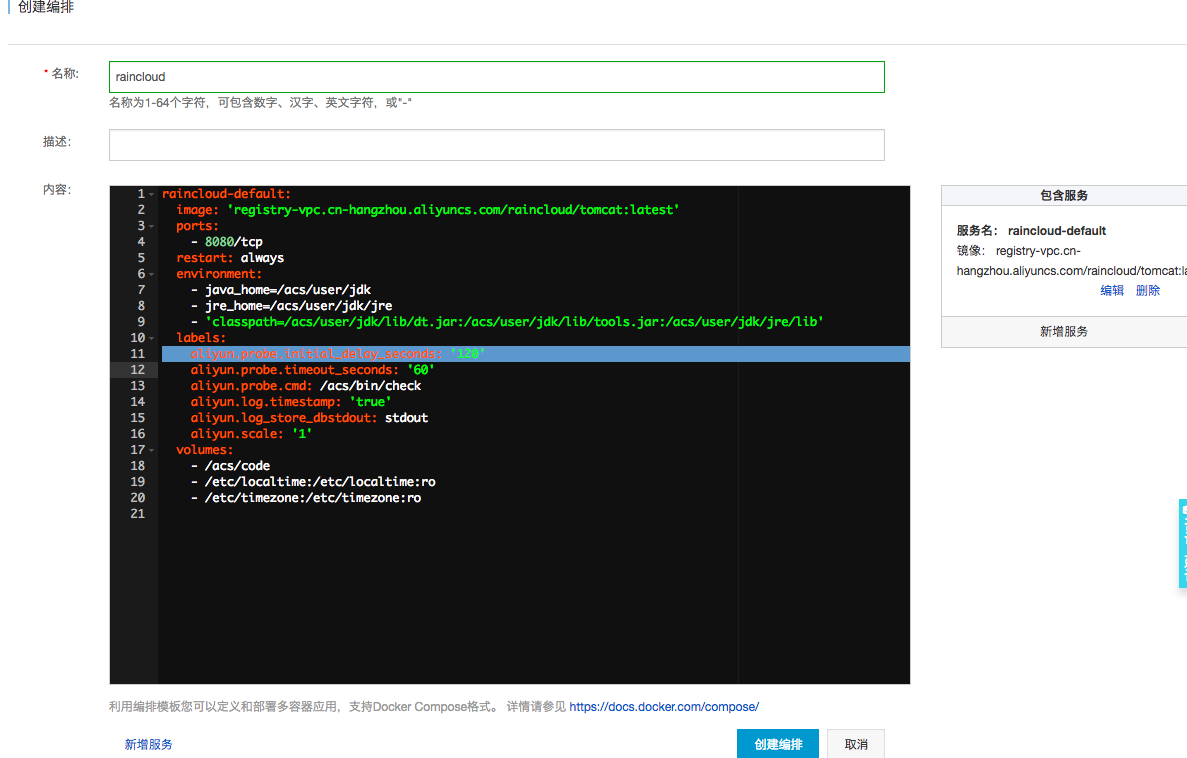




1. 创建编排模版
2. 选择容器服务-swarm,选择编排模版Tab页面，创建编排模版。



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。



1. 创建应用

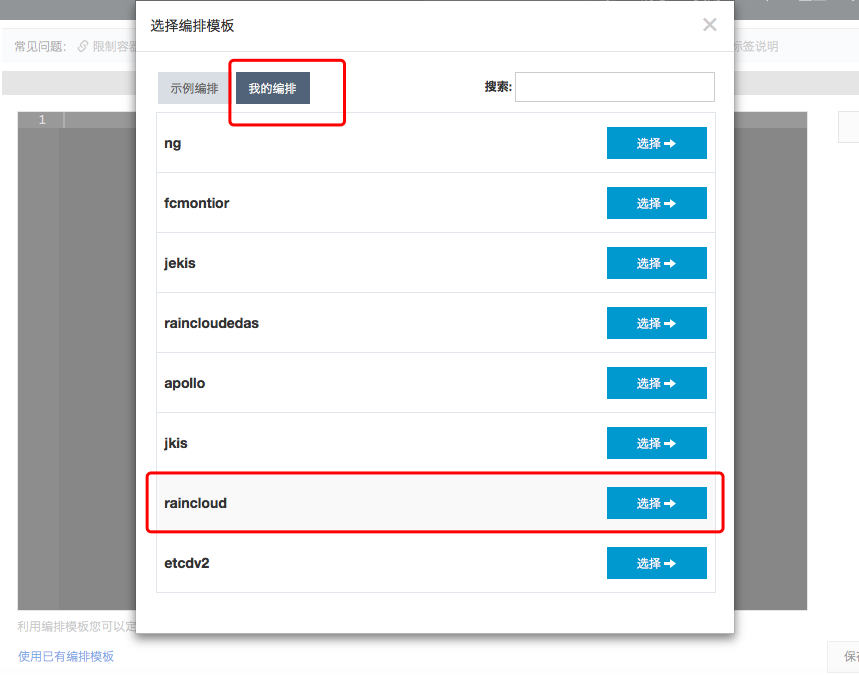
1.切换到应用tab页，选择创建应用，填入相应的应用信息。



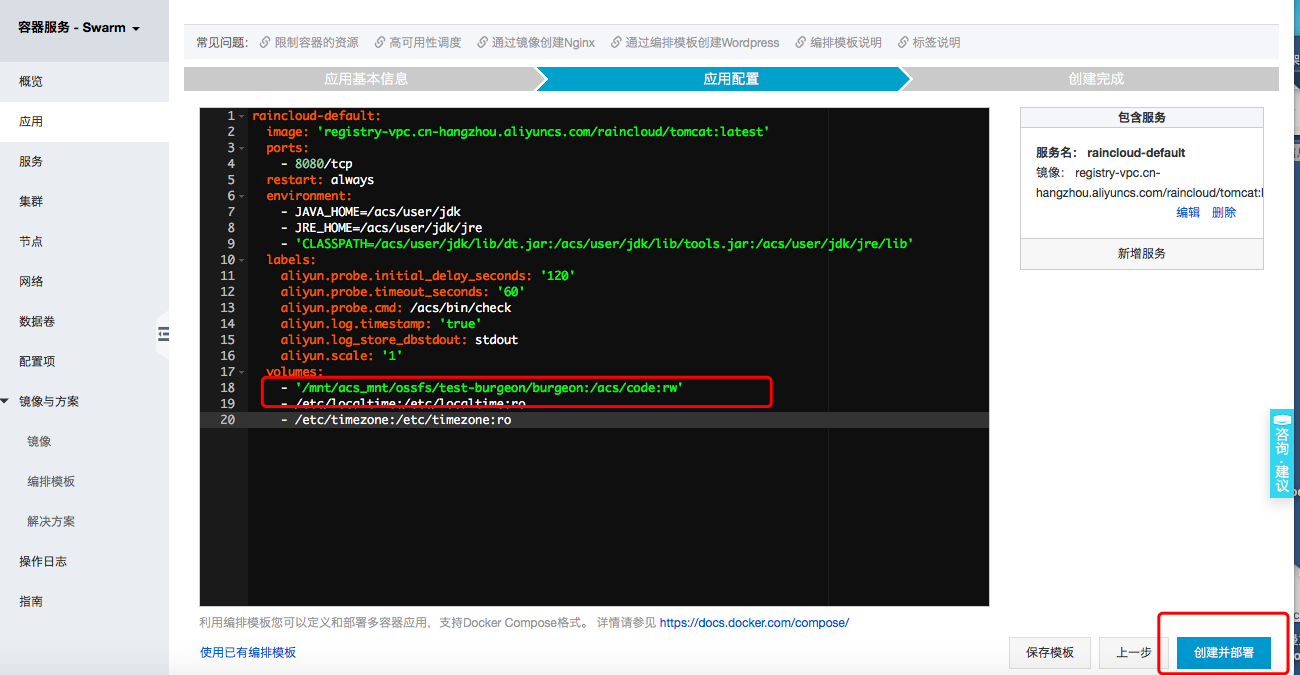
2.选择使用编排模版创建（可以直接输入模版创建并部署，模版参照文档后例子，本文档寻那呃已有的编排模版）, 选择已有编排模版



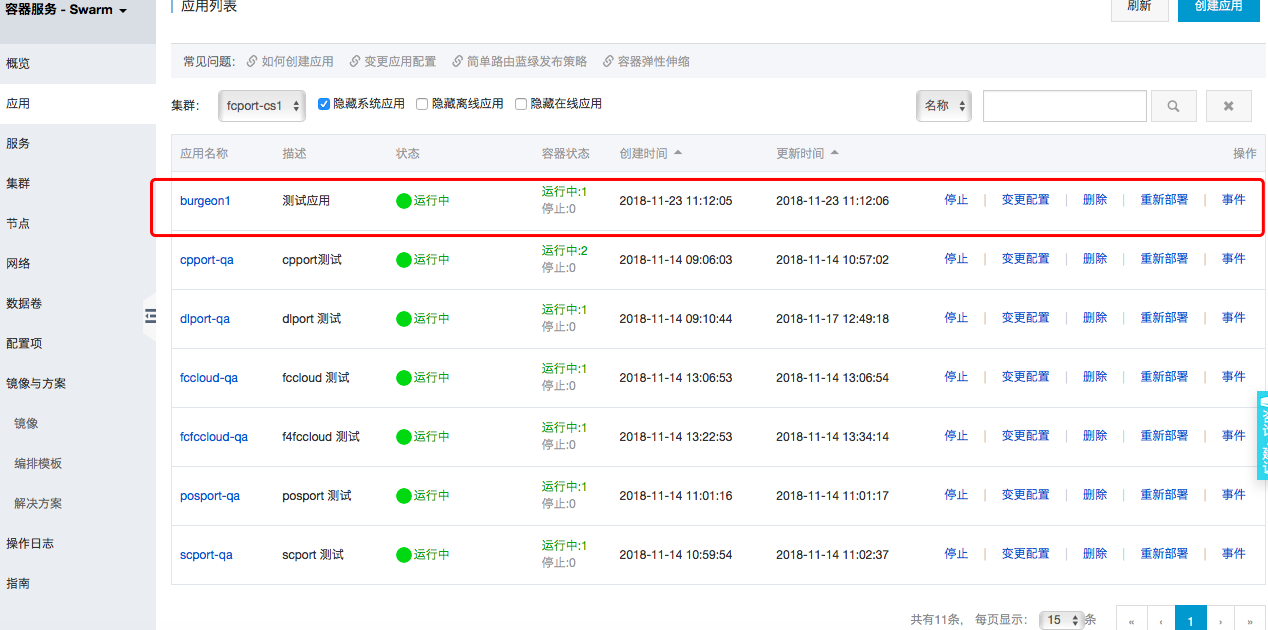
3.切换到我的编排tab，选择raincloud



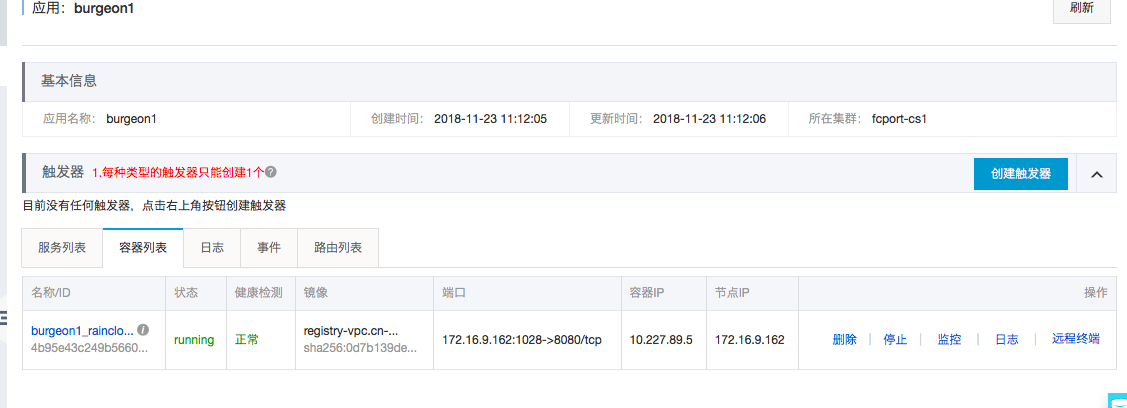
4.创建并部署，volumes标签下写入 ： -'/mnt/acs\_mnt/ossfs/test-burgeon/burgeon1:/acs/code:rw'，（或将本地磁盘下的应用拷贝到容器目录下）将数据卷指定的应用拷贝到对应的容器目录下，创建并部署。



5.应用创建完成



6.对应应用详细信息



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