

实验课程6-2 部署应用至容器集群之deployment的创建和部署

我们在前面的实践课程中学习了如何创建pod来在容器集群中运行应用程序，但一个pod中的应用程序是只有一个副本的，也就是说如果pod由于某些原因坏掉或宕机，那么这个应用服务的后端资源也就不存在了，整个应用服务的访问就会中断。本次课程主要介绍deployment资源的创建，一个deployment资源可以描述应用程序的副本数，做到真正的应用高可用性。

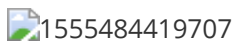
1、学习编写Deployment的yaml文件

我们新建一个deployment.yaml文件用于部署一个java-demo应用。

```
apiVersion: extensions/v1beta1 # API版本
kind: Deployment                # 资源类型, Deployment
metadata:                       # 元数据
  labels:                       # 元数据, 标签列表
    name: java-demo-deployment # 元数据, deployment的标签名称
    name: java-demo-deployment # 元数据, deployment的名字
    namespace: user2           # 元数据, deployment的命名空间
spec:                           # deployment中容器模板的详细定义
  replicas: 4                   # pod的副本数
  template:
    metadata:
      labels:
        app: java-demo-deployment
    spec:
      containers:
        - name: java-demo-deployment # 容器名称
          image: registry.cn-beijing.aliyuncs.com/demo0417/demo-test0417:latest # 容器
            使用的镜像
          imagePullPolicy: Always # 获取镜像的策略
          ports:
            - containerPort: 8080 # 容器要暴露的端口
```

2、部署deployment.yaml

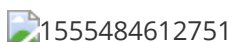
容器服务 -> 应用 -> 部署 -> 使用模板创建



依次选择集群 -> 命名空间 -> 自定义模板，并粘贴yaml文件内容：



点击 创建 后，可以在 部署 菜单查看deployment的列表：



然后可以在 容器组 菜单查看pod的列表，可以看到java-demo-deployment包含了4个以java-demo-deployment为前缀的pod：

Deployment用于在容器集群中运行应用程序并管理多个副本，解决了应用的高可用性问题，但应用程序的访问服务还需要创建Service来支持，我们会在后续课程中进行介绍。