6/24

一堆工具的介绍

1. 对比和搭建Docker私有镜像
2. helm工具和chart仓库
3. 一个springboot项目从本地idea开发到k8s多pod部署流程
4. rancher简介

# 1. 搭建Docker私有镜像

## 1.1 官方Registry

<https://github.com/docker/docker-registry>

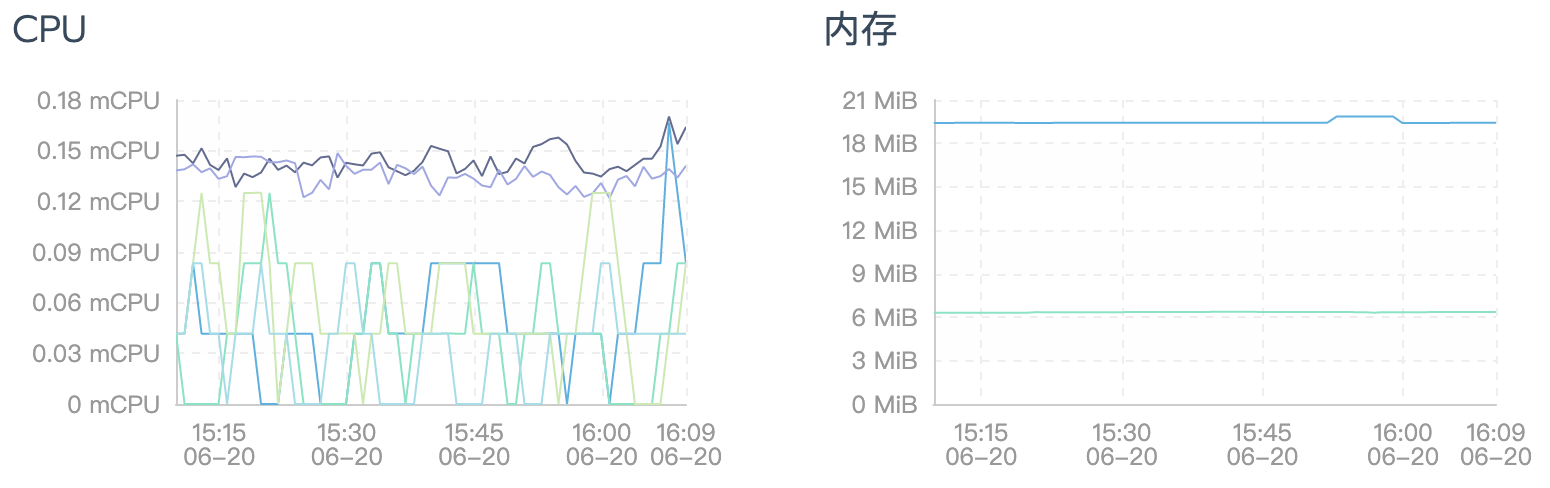
### 1.1.1 HA方案

多个pod指向同一个存储，可能存在跨机器传输的问题。定时的备份存储的folder。但我们只有local storage，不能用作共同存储。所以目前，只能单机多pod部署，指向同一个本地folder，然后定时的备份该folder。

### 1.1.2优缺点

优点：

简单，轻量。占用资源少。



缺点：

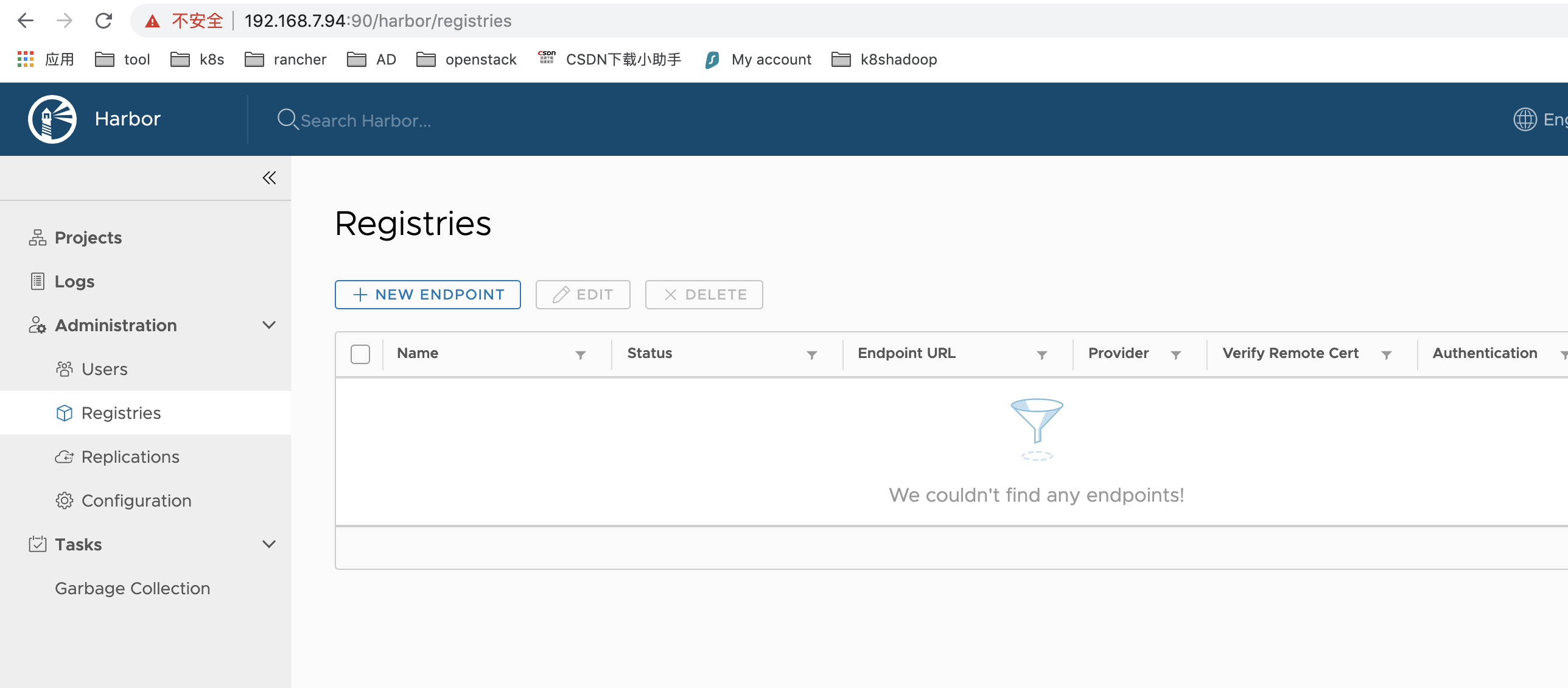
没有角色控制、用户管理等功能。

## 1.2 Vmware Harbor

[http://192.168.7.94:90/harbor/sign-in?redirect\_url=%2Fharbor%2Fprojects](http://192.168.7.94:90/harbor/sign-in?redirect_url=/harbor/projects)

<https://github.com/goharbor/harbor>

使用docker-compose发布部署。



### 1.2.1 HA 方案

[https://github.com/goharbor/harbor-helm/blob/master/docs/High%20Availability.md](https://github.com/goharbor/harbor-helm/blob/master/docs/High Availability.md)

官方HA方案：Most of Harbor's components are stateless now.

As for storage layer, it is expected that the user provide **high available PostgreSQL, Redis cluster** for application data and PVCs or object storage for storing images and charts.

简化方案：就用本地文件系统存储镜像。单机部署，harbor UI中可以配置同步image到另外一个registry。但role什么的需要额外一个定时job去备份。

### 1.2.2 优缺点

优点：

基于角色的访问控制

镜像复制

Web UI管理界面

可以集成LDAP或AD用户认证系统

审计日志

提供RESTful API以提供外部客户端调用

镜像安全漏洞扫描（从v1.2版本开始集成了Clair景象扫描工具）

缺点

HA部署比较复杂。

## 1.3 其他方案

Portus，Sonatype Nexus

## 1.4方案对比

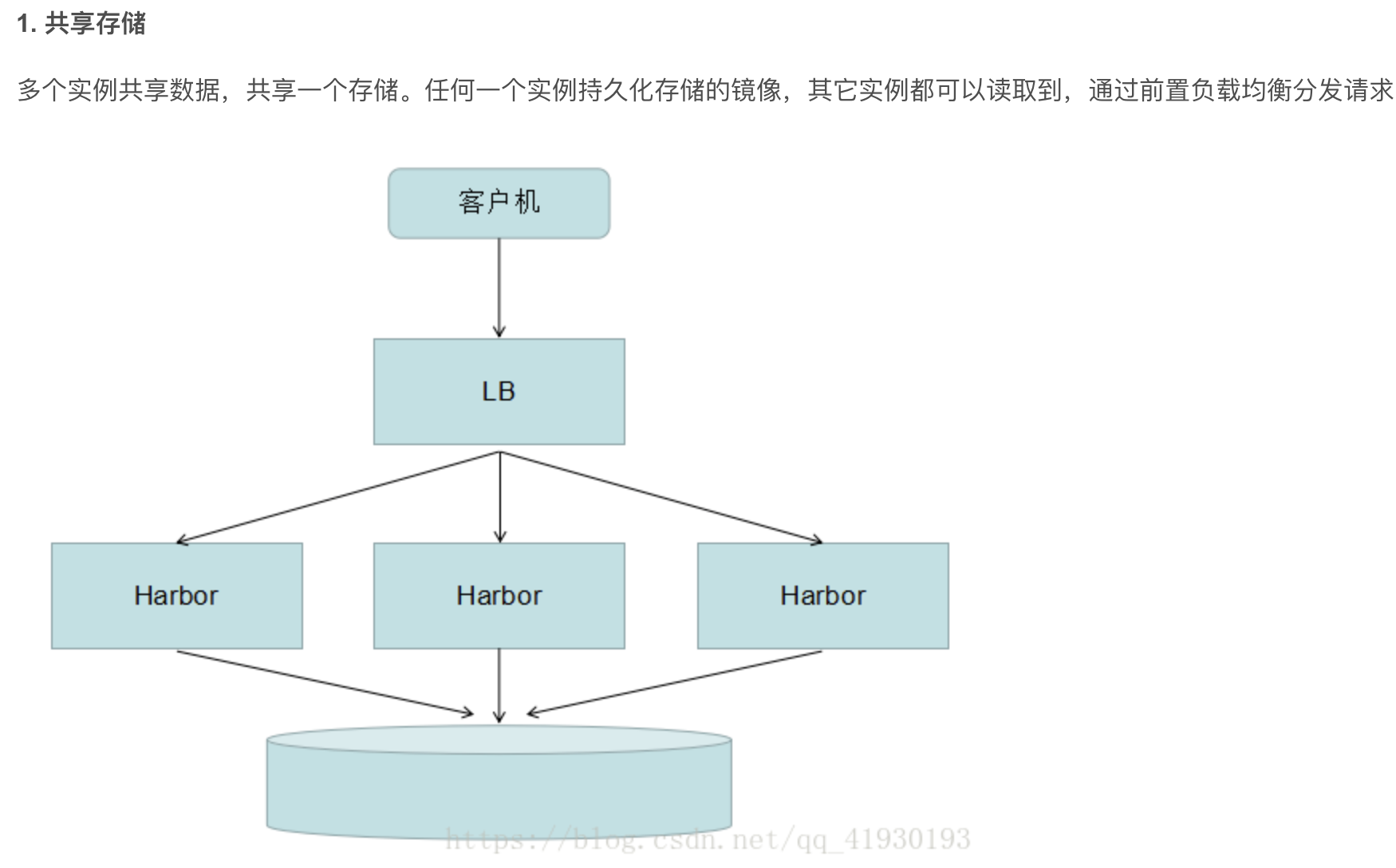


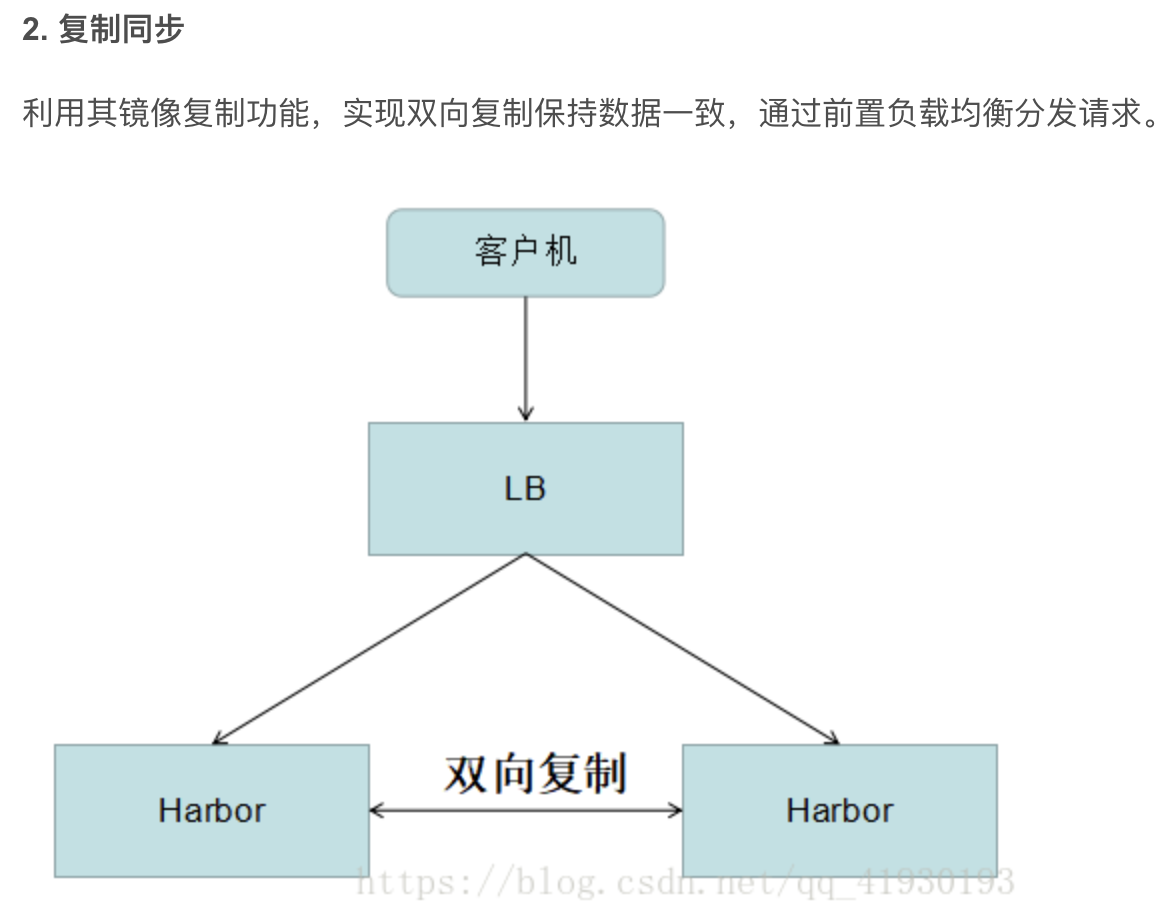
Depend on how we provide services:

Need role control?

## 1.5 HA方案解释

HA提供的方案





## 1.6 Solution

### 1.6.2 官方Registry + Harbor

两个节点分步部署一个官方Registry， 一个Harbor。 Harbor 将registry push给registry备份。

优点：registry的轻量。

缺点：Harbor的掉线，将会导致整个私有仓库的不可用。

### 1.6.3 两个Harbor

两个节点部署两个Harbor。两个Harbor互相push image。

优点：两个Harbor 形成image的HA。

缺点：Harbor本身就占用资源比较大。role由于不能同步，所以对role的操作需要手动执行两遍。

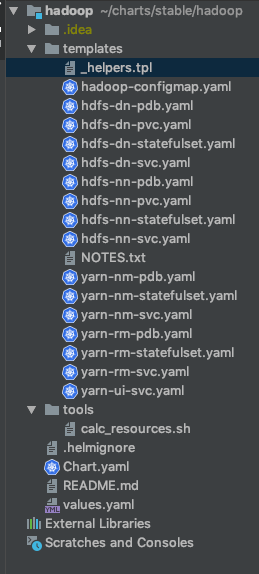
以上两个方案都不能支持role的备份，若要进行role的备份，需要再设置一个job进行备份。

# helm工具和chart仓库

## 2.1 chart

chart是描述相关的一组k8s资源的文件集合。单个chart可能用于部署简单的东西，比如比如memcached pod，或者一些复杂的东西，比如完整的具有HTTP服务，数据库，缓存等的Web应用程序堆栈。

chart通过创建为特定目录树的文件，将它们打包到版本化的压缩包，然后进行部署。



## 2.2 helm

Helm是一个kubernetes应用的包管理工具，用来管理charts——预先配置好的安装包资源，有点类似于Ubuntu的APT和CentOS中的yum。

chart就是包，Helm就是包管理工具，类似于APT。可以自定义Helm的包源。helm的主要任务就是在repository中查找需要的chart，然后将chart以release的形式安装到k8s集群中。

## 2.3 chart库

### 2.3.1 chart库的选择

由于 chart 库可以是任何可以提供 YAML 和 tar 文件并可以回答 GET 请求的 HTTP 服务器，因此当托管自己的 chart 库时，很多选择。例如，可以使用 Google 云端存储（GCS）存储桶，Amazon S3 存储桶，Github Pages，甚至可以创建自己的 Web 服务器。

### 2.3.2 Harbor as chart repository

sudo ./install.sh --with-clair --with-chartmuseum



# 示例简单项目开发部署

Springboot in k8s

## 3.1 代码

@RestController  
public class DemoResource {  
  
 @RequestMapping("/")  
 public String index() {  
 return "Hello Docker!";  
 }  
}

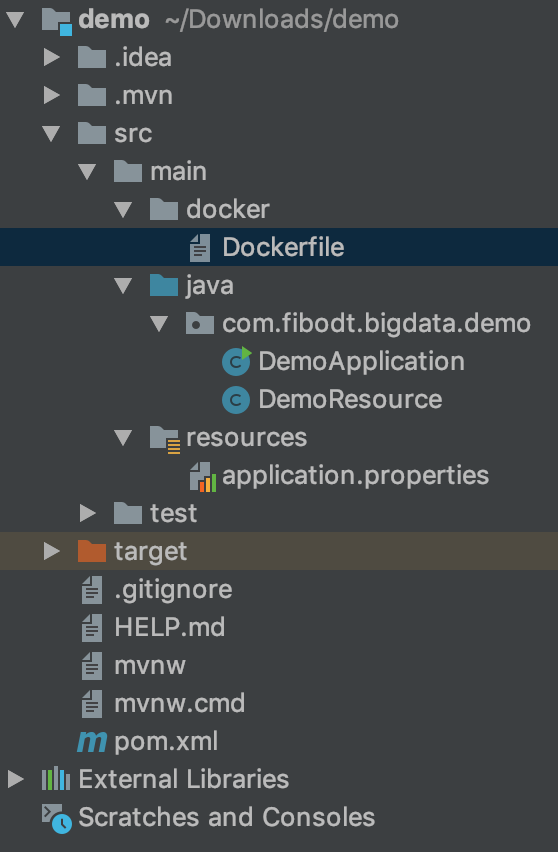
插件

<plugin>  
 <groupId>com.spotify</groupId>  
 <artifactId>docker-maven-plugin</artifactId>  
 <configuration>  
 <imageName>${docker.image.prefix}/${project.artifactId}</imageName>  
 <dockerDirectory>src/main/docker</dockerDirectory>  
 <resources>  
 <resource>  
 <targetPath>/</targetPath>  
 <directory>${project.build.directory}</directory>  
 <include>${project.build.finalName}.jar</include>  
 </resource>  
 </resources>  
 </configuration>  
</plugin>

## 3.2 Dockerfile

FROM openjdk  
VOLUME /tmp  
ADD demo-0.0.1-SNAPSHOT.jar app.jar  
ENTRYPOINT ["java","-Djava.security.egd=file:/dev/./urandom","-jar","/app.jar"]

目录结构



## 3.3 Build and push to remote repo

打image命令

mvn package docker:build

push命令

1. Tag:

docker tag springboottest/demo:latest 192.168.7.94:90/testlibrary/springboottest/demo

1. Push(maybe you need log in docker first)

docker push 192.168.7.94:90/testlibrary/springboottest/demo:latest

<http://192.168.7.94:31212/>

# rancher介绍

[https://192.168.7.95](https://192.168.7.95/)

Doing：redeploy harbor and add chartmuseum, then test add Harbor chart repo to Rancher Catalogs.

researching

1. helm模板语法
2. yaml配置文件
3. 详细看stable的hadoop chart的内容。