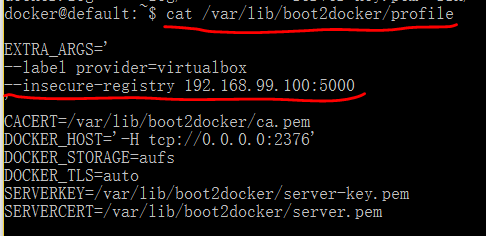
# Docker registry私有仓库：

## https ---->http:



重启容器后成功。

推送:docker push xxxxx/repo\_name

拉取:docker pull xxxxx/repo\_name

## 进入镜像：



镜像文件存放路径:/var/lib/registry/docker/registry/v2/repositories/

# Docker-maven-plugin打包上传镜像配置：

纯配置文件，不依赖dockerfile制作镜像。（默认生成dockerfile文件）

*<!-- 添加docker-maven插件 -->*<**plugin**>  
 <**groupId**>com.spotify</**groupId**>  
 <**artifactId**>docker-maven-plugin</**artifactId**>  
 <**executions**>  
 <**execution**>  
 <**id**>build-image</**id**>  
 <**phase**>package</**phase**>  
 <**goals**>  
 <**goal**>build</**goal**>  
 </**goals**>  
 </**execution**>  
 </**executions**>  
 <**configuration**>  
 <**dockerHost**>https://192.168.99.100:2376</**dockerHost**>  
 <**dockerCertPath**>C:\Users\admin01\.docker\machine\machines\default</**dockerCertPath**>  
 <**imageName**>192.168.99.100:5000/${project.artifactId}:${project.version}</**imageName**>  
 <**forceTags**>true</**forceTags**>  
 <**baseImage**>java:8</**baseImage**>  
 <**pushImage**>true</**pushImage**>  
 <**entryPoint**>["java", "-jar", "/${project.build.finalName}.jar"]</**entryPoint**>  
 <**resources**>  
 <**resource**>  
 <**targetPath**>/</**targetPath**>  
 <**directory**>${project.build.directory}</**directory**>  
 <**include**>${project.build.finalName}.jar</**include**>  
 </**resource**>  
 </**resources**>  
 </**configuration**>  
</**plugin**>