**创建一个自带web项目的镜像**

**运行为容器后web项目自动部署在容器内的tomcat中**

要实现这个功能，主要分三步

1. 根据原始镜像配置可运行tomcat的环境
2. 让tomcat在容器启动时自动运行
3. 将web项目部署到tomcat中

**环境：**

Docker

1. **根据原始镜像配置可运行tomcat的环境**

**步骤：**

1. 拉取一个干净的原始镜像centos:7；下载jdk8和tomcat9的gz压缩包

docker pull centos:7

1. 运行该镜像并进入容器内，为jdk和tomcat创建目录

docker run -itd --name mycentos centos:7 //运行镜像

docker exec -it mycentos bash //进入容器

# mkdir /usr/java

# mkdir /usr/local/tomcat

1. 将宿主机上的jdk和tomcat压缩包拷贝到容器内的相应目录

docker cp jdk-8u171-linux-x64.tar.gz mycentos:/usr/java

docker cp apache-tomcat-9.0.8.tar.gz mycentos:/usr/local/tomcat

1. 进入容器解压jdk和tomcat，并删除压缩文件

docker exec -it mycentos bash

# cd /usr/java

# tar -zxvf jdk-8u171-linux-x64.tar.gz

# rm jdk-8u171-linux-x64.tar.gz

# cd/usr/local/tomcat

# tar -zxvf apache-tomcat-9.0.8.tar.gz

# rm apache-tomcat-9.0.8.tar.gz

# exit

1. 将容器commit为新的镜像

docker commit mycentos mycentos

1. **让tomcat在容器启动时自动运行**

**步骤：**

1. 创建dockerfile，以mycentos镜像为基础，通过ENV语句配置环境变量，通过CMD语句让容器启动时运行catalina.sh启动tomcat服务

# 以mycentos为基础

FROM mytomcat

MAINTAINER salo

# 设置工作目录，会对RUN,ADD,COPY,CMD,ENTRYPOINT等生效

WORKDIR /usr/local/tomcat/apache-tomcat-9.0.8

# 配置java与tomcat环境变量

ENV JAVA\_HOME /usr/java/jdk1.8.0\_171

ENV CLASSPATH $JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

ENV CATALINA\_HOME /usr/local/tomcat/apache-tomcat-9.0.8

ENV CATALINA\_BASE /usr/local/tomcat/apache-tomcat-9.0.8

ENV PATH $PATH:$JAVA\_HOME/bin:$CATALINA\_HOME/lib:$CATALINA\_HOME/bin

# 设置容器启动时命令

CMD ./bin/catalina.sh run

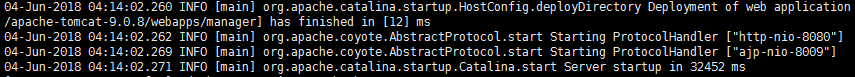
1. 根据dockerfile创建新的镜像

docker build -t test0 .

1. 运行镜像，映射容器端口8080到宿主机端口8084，并保持其容器持续运行

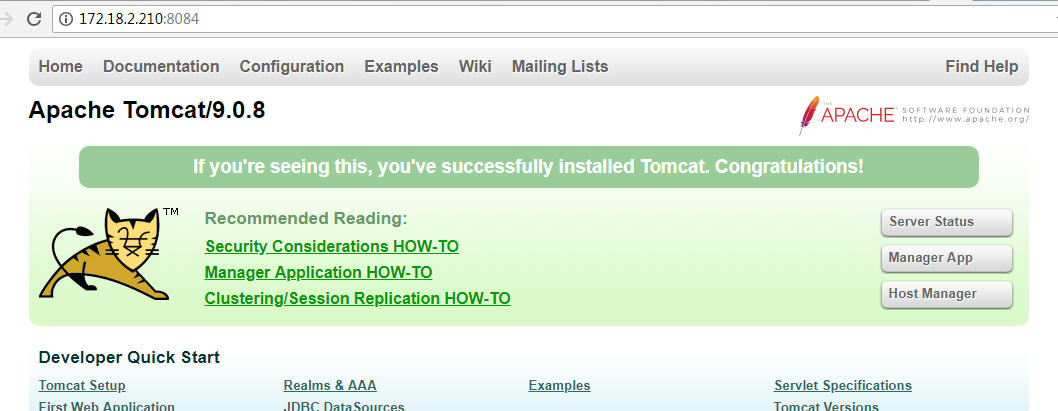
docker run -itd -p 8084:8080 --name mytest test0

// 此时可通过docker logs mytest命令查看容器内后台信息，以确认tomcat服务是否已经启动



1. 访问tomcat页面

在远程机器上通过浏览器访问172.18.2.210:8084



Tomcat服务成功在容器中启动

1. **将web项目部署到tomcat中**

**步骤：**

1. 做一个简单的web项目，并打包为war包
2. 修改dockerfile文件，添加COPY命令将本地war包拷贝到镜像系统中tomcat目录下的webapp文件夹内

# 以mycentos为基础

FROM mytomcat

MAINTAINER salo

# 设置工作目录，会对RUN,ADD,COPY,CMD,ENTRYPOINT等生效

WORKDIR /usr/local/tomcat/apache-tomcat-9.0.8

**# 拷贝本地myweb.war文件为镜像内tomcat目录下webapps文件夹下的demo.war文件**

**COPY myweb.war webapps/demo.war**

# 配置java与tomcat环境变量

ENV JAVA\_HOME /usr/java/jdk1.8.0\_171

ENV CLASSPATH $JAVA\_HOME/lib/dt.jar:$JAVA\_HOME/lib/tools.jar

ENV CATALINA\_HOME /usr/local/tomcat/apache-tomcat-9.0.8

ENV CATALINA\_BASE /usr/local/tomcat/apache-tomcat-9.0.8

ENV PATH $PATH:$JAVA\_HOME/bin:$CATALINA\_HOME/lib:$CATALINA\_HOME/bin

# 设置容器启动时命令

CMD ./bin/catalina.sh run

1. 根据dockerfile构建镜像

docker build -t test .

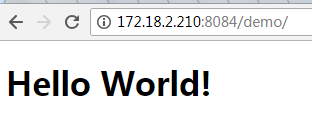
1. 运行镜像，映射端口，并保持运行

docker stop mytest // 停止之前的测试容器

docker run -itd -p 8084:8080 --name test test

1. 访问web项目页面

在远程机器上通过浏览器访问172.18.2.210:8084/demo



**附：**

dockerfile文件；简单的web项目war包

