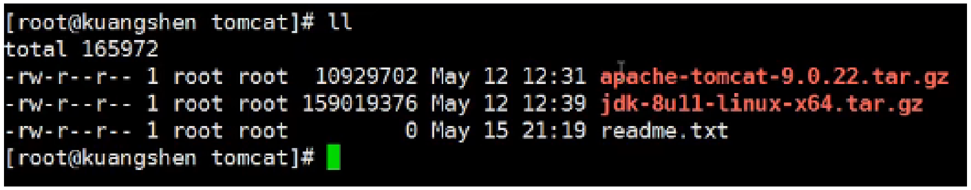
14、构建自己的docker镜像-->Tomcat镜像

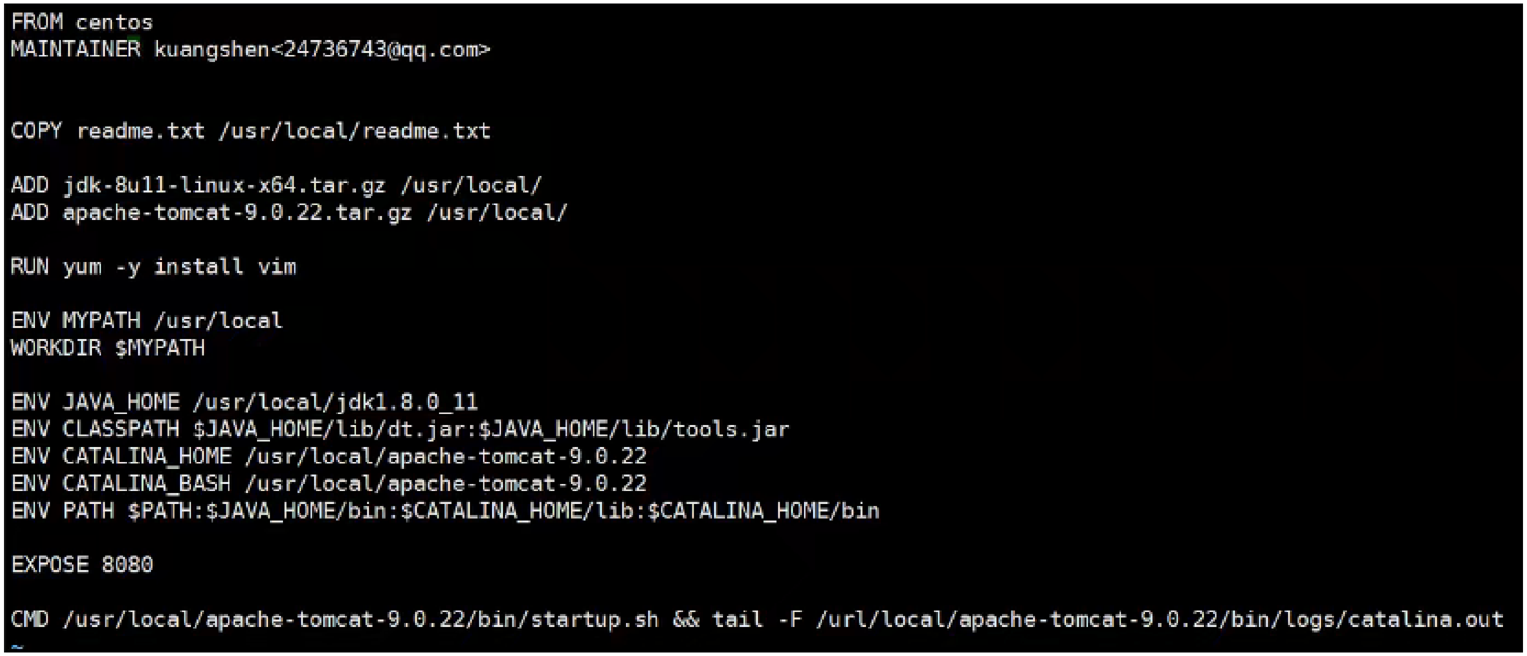
（1）准备镜像文件



```
[root@kuangshen tomcat]# ll
total 165972
-rw-r--r-- 1 root root  10929702 May 12 12:31 apache-tomcat-9.0.22.tar.gz
-rw-r--r-- 1 root root 159019376 May 12 12:39 jdk-8u11-linux-x64.tar.gz
-rw-r--r-- 1 root root         0 May 15 21:19 readme.txt
[root@kuangshen tomcat]# 
```

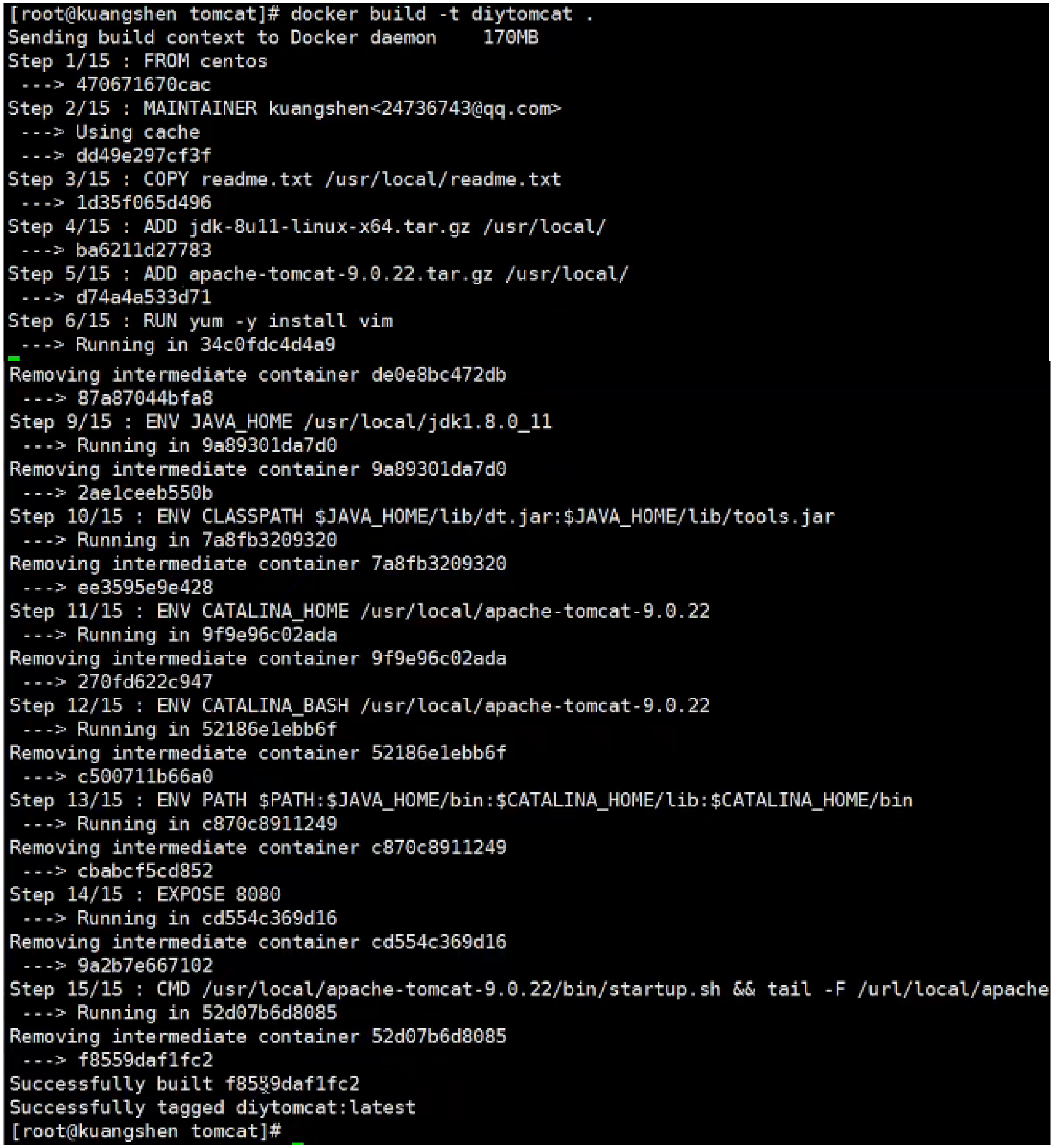
（2）编写Dockerfile文件

- 如果文件名创建为DockerFile，和官方默认的命名一样，则docker build时会自动寻找这个文件，就不用-f指定了，请看第（3）步的构建命令没有使用-f指定dockerfile文件的路径
- 注意我这里Dockerfile文件内容有写错，CMD命令后面的文件路径url写错了，应该是/usr/local ,所以在运行镜像或启动容器时，控制台如果没有显示CMD命令执行的tail打印tomcat启动日志，不要有疑惑，是因为这里有错误。



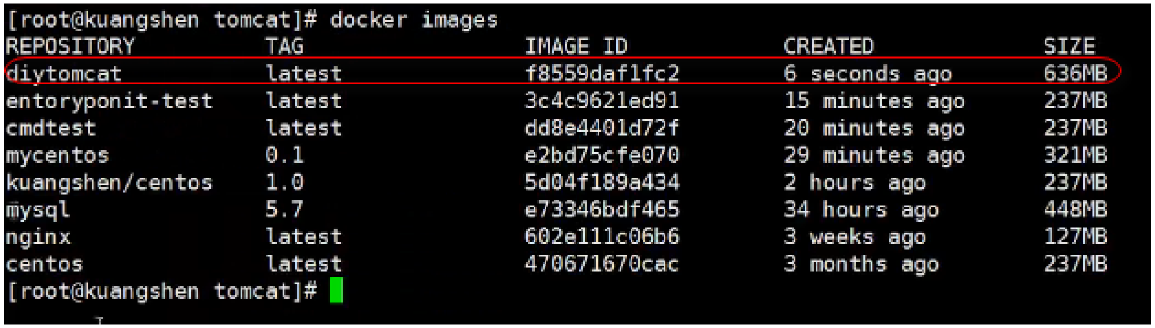
```
FROM centos
MAINTAINER kuangshen<24736743@qq.com>


COPY readme.txt /usr/local/readme.txt

ADD jdk-8u11-linux-x64.tar.gz /usr/local/
ADD apache-tomcat-9.0.22.tar.gz /usr/local/

RUN yum -y install vim

ENV MYPATH /usr/local
WORKDIR $MYPATH

ENV JAVA_HOME /usr/local/jdk1.8.0_11
ENV CLASSPATH $JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
ENV CATALINA_HOME /usr/local/apache-tomcat-9.0.22
ENV CATALINA_BASH /usr/local/apache-tomcat-9.0.22
ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/lib:$CATALINA_HOME/bin

EXPOSE 8080

CMD /usr/local/apache-tomcat-9.0.22/bin/startup.sh && tail -F /url/local/apache-tomcat-9.0.22/bin/logs/catalina.out
~
```

（3）构建自己的diy的docker镜像-->tomcat镜像



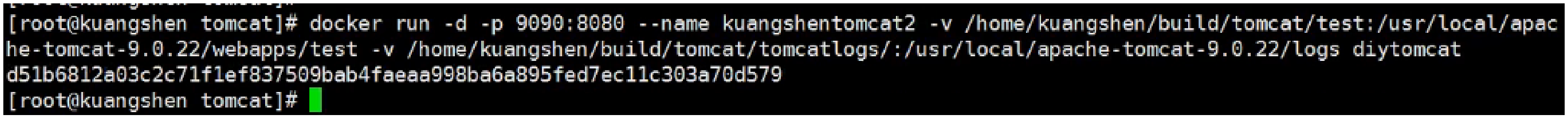
```
[root@kuangshen tomcat]# docker build -t diytomcat .
Sending build context to Docker daemon    170MB
Step 1/15 : FROM centos
 ---> 470671670cac
Step 2/15 : MAINTAINER kuangshen<24736743@qq.com>
 ---> Using cache
 ---> dd49e297cf3f
Step 3/15 : COPY readme.txt /usr/local/readme.txt
 ---> 1d35f065d496
Step 4/15 : ADD jdk-8u11-linux-x64.tar.gz /usr/local/
 ---> ba6211d27783
Step 5/15 : ADD apache-tomcat-9.0.22.tar.gz /usr/local/
 ---> d74a4a533d71
Step 6/15 : RUN yum -y install vim
 ---> Running in 34c0fdc4d4a9
Removing intermediate container de0e8bc472db
 ---> 87a87044bfa8
Step 9/15 : ENV JAVA_HOME /usr/local/jdk1.8.0_11
 ---> Running in 9a89301da7d0
Removing intermediate container 9a89301da7d0
 ---> 2ae1ceeb550b
Step 10/15 : ENV CLASSPATH $JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
 ---> Running in 7a8fb3209320
Removing intermediate container 7a8fb3209320
 ---> ee3595e9e428
Step 11/15 : ENV CATALINA_HOME /usr/local/apache-tomcat-9.0.22
 ---> Running in 9f9e96c02ada
Removing intermediate container 9f9e96c02ada
 ---> 270fd622c947
Step 12/15 : ENV CATALINA_BASH /usr/local/apache-tomcat-9.0.22
 ---> Running in 52186e1ebb6f
Removing intermediate container 52186e1ebb6f
 ---> c500711b66a0
Step 13/15 : ENV PATH $PATH:$JAVA_HOME/bin:$CATALINA_HOME/lib:$CATALINA_HOME/bin
 ---> Running in c870c8911249
Removing intermediate container c870c8911249
 ---> cbabcf5cd852
Step 14/15 : EXPOSE 8080
 ---> Running in cd554c369d16
Removing intermediate container cd554c369d16
 ---> 9a2b7e667102
Step 15/15 : CMD /usr/local/apache-tomcat-9.0.22/bin/startup.sh && tail -F /url/local/apache
 ---> Running in 52d07b6d8085
Removing intermediate container 52d07b6d8085
 ---> f8559daf1fc2
Successfully built f8559daf1fc2
Successfully tagged diytomcat:latest
[root@kuangshen tomcat]#
```

（4）查看构建好的镜像并运行镜像，运行出一个容器

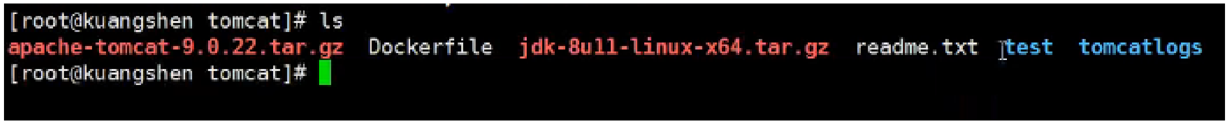


```
[root@kuangshen tomcat]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
diytomcat           latest              f8559daf1fc2        6 seconds ago       636MB
entoryponit-test    latest              3c4c9621ed91        15 minutes ago      237MB
cmdtest             latest              dd8e4401d72f        20 minutes ago      237MB
mycentos            0.1                 e2bd75cfe070        29 minutes ago      321MB
kuangshen/centos    1.0                 5d04f189a434        2 hours ago         237MB
mysql               5.7                 e73346bdf465        34 hours ago        448MB
nginx               latest              602e111c06b6        3 weeks ago         127MB
centos              latest              470671670cac        3 months ago        237MB
[root@kuangshen tomcat]# 
```

运行镜像，并且对即将运行的镜像做了挂载（挂载到宿主机的test、tomcatlogs目录）



```
[root@kuangshen tomcat]# docker run -d -p 9090:8080 --name kuangshentomcat2 -v /home/kuangshen/build/tomcat/test:/usr/local/apac
he-tomcat-9.0.22/webapps/test -v /home/kuangshen/build/tomcat/tomcatlogs/:/usr/local/apache-tomcat-9.0.22/logs diytomcat
d51b6812a03c2c71f1ef837509bab4faeaa998ba6a895fed7ec11c303a70d579
[root@kuangshen tomcat]# 
```



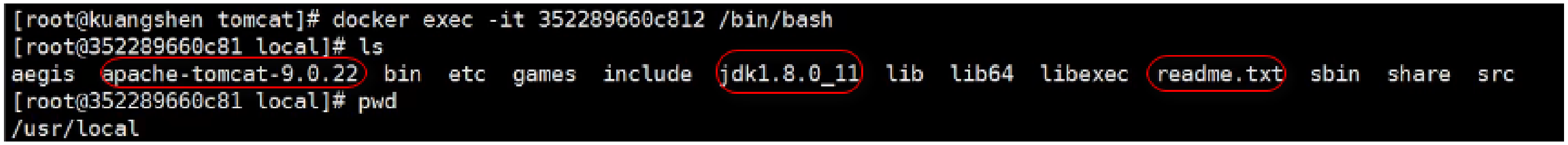
```
[root@kuangshen tomcat]# ls
apache-tomcat-9.0.22.tar.gz  Dockerfile  jdk-8u11-linux-x64.tar.gz  readme.txt  test  tomcatlogs
[root@kuangshen tomcat]# 
```

进入运行中容器的内部（这里进入的运行容器是352289，是因为第一次运行镜像后的容器352289被删除了，重新运行了镜像，容器id是d51868，所以这里进入容器时，容器id和运行镜像时生成的容器ID不一致，不要有疑惑）

(1)可以看出默认进入的是dockerfile文件在构建镜像时的WORKDIR命令指定的目录

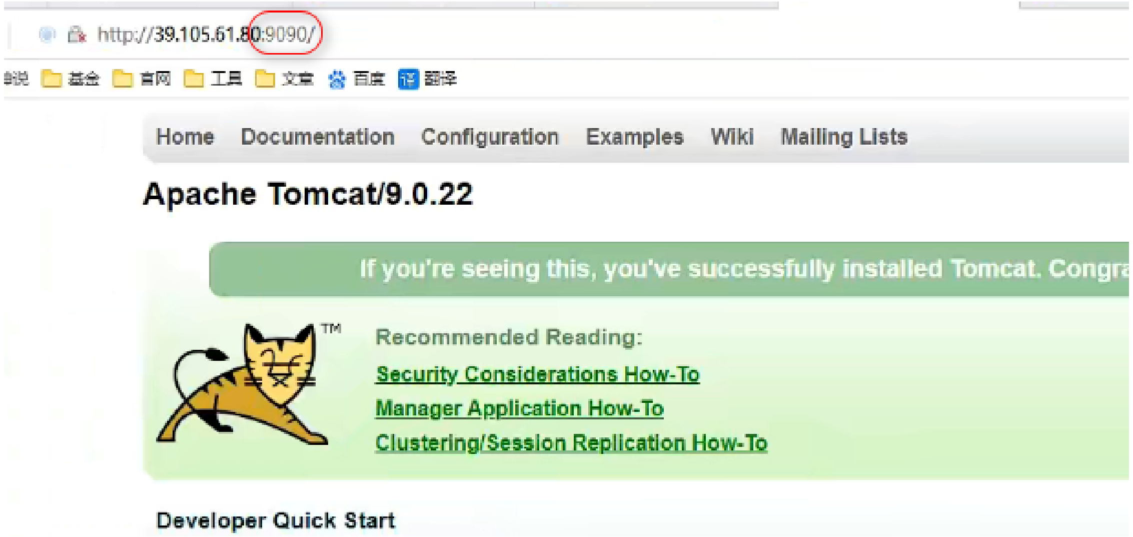
(2)readme.txt、apache-tomcat-9.0.22、jdk1.8.0_11分别在dockerfile文件构建镜像时通过COPY、ADD、ADD命令将文件复制到指定的路径下了，并且apache-tomcat-9.0.22和jdk1.8.0_11压缩包会被自动解压。

(3)注意：RUN、ADD、COPY、CMD、ENTRYPOINT这些指令的默认的目的地目录就是WORKDIR指定的目录，可以手动指定目的目录。



```
[root@kuangshen tomcat]# docker exec -it 352289660c812 /bin/bash
[root@352289660c81 local]# ls
aegis  apache-tomcat-9.0.22  bin  etc  games  include  jdk1.8.0_11  lib  lib64  libexec  readme.txt  sbin  share  src
[root@352289660c81 local]# pwd
/usr/local
```

（5）外部浏览器访问tomcat



http://39.105.61.80:9090/

Apache Tomcat/9.0.22

If you're seeing this, you've successfully installed Tomcat. Congra

Recommended Reading:

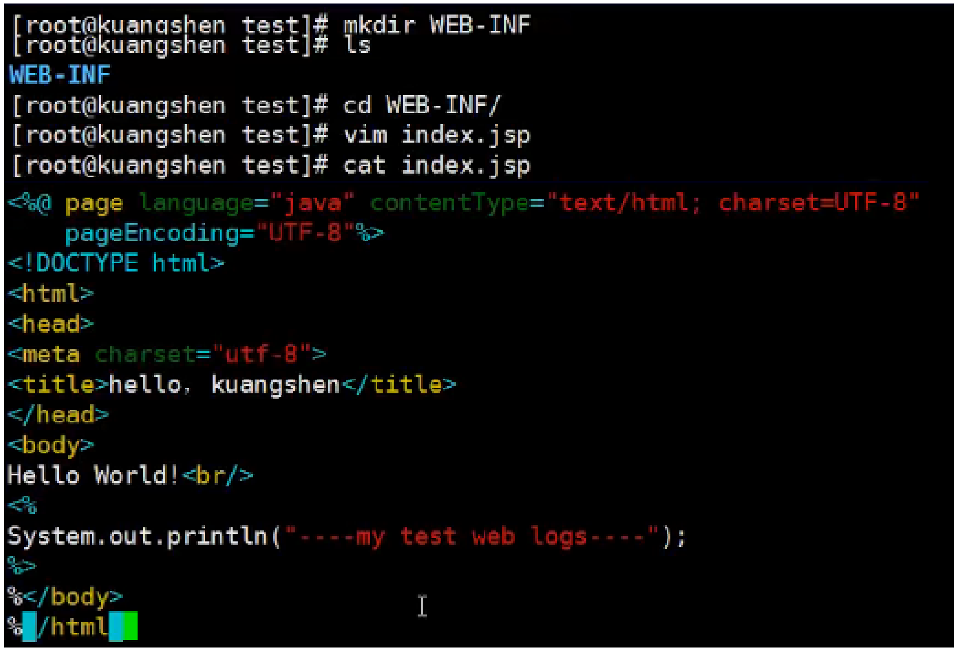
Security Considerations How-To

Manager Application How-To

Clustering/Session Replication How-To

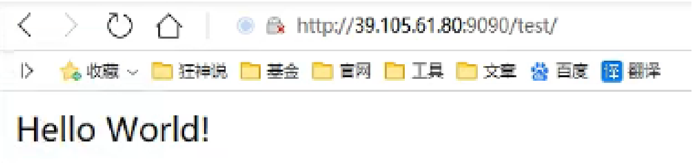
Developer Quick Start

（6）进入宿主机的挂载目录test，添加tomcat所需的目录结构WEB-INF和jsp欢迎页，容器内会同步共享



```
[root@kuangshen test]# mkdir WEB-INF
[root@kuangshen test]# ls
WEB-INF
[root@kuangshen test]# cd WEB-INF/
[root@kuangshen test]# vim index.jsp
[root@kuangshen test]# cat index.jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>hello, kuangshen</title>
</head>
<body>
Hello World!<br/>
<%
System.out.println("----my test web logs----");
%>
%</body>
%</html>
```

（7）浏览器重新访问tomcat容器的test目录（因为此时容器中的test目录已经共享宿主机的挂载目录test的内容，所以满足tomcat默认结构的目录，访问的是jsp页）



Hello World!

（8）以后开发步骤，必须掌握Dockerfile文件的编写，因为我们所有的操作都是基于Dockerfile文件所构建的docker镜像来运行的！