Lab Scenario – How to run Tomcat inside Docker with the Maven war file

Steps

1. Copy the War file that is already generated by the Maven project
2. Create the Dockerfile
3. Build the Docker image
4. Create Container

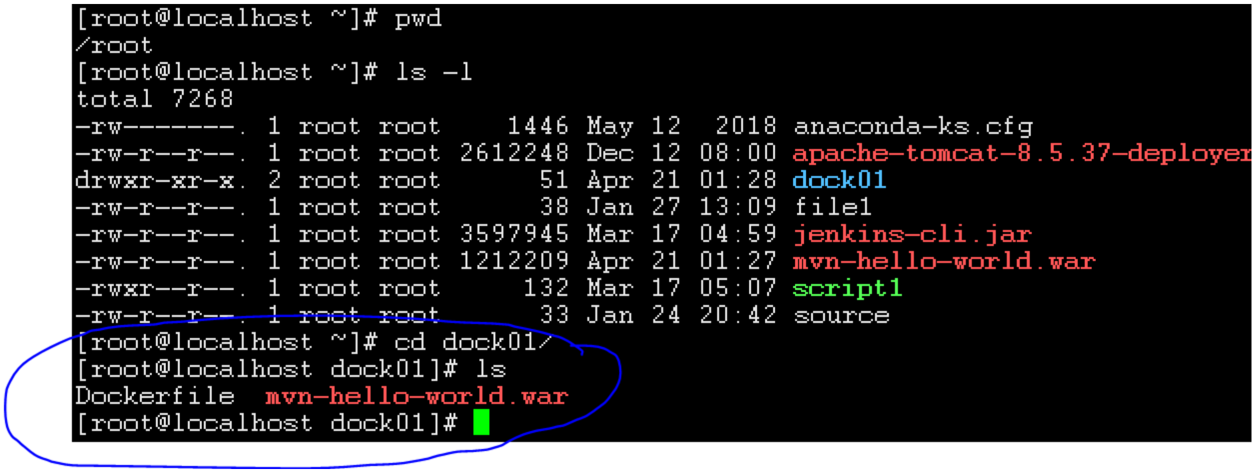
1. Copy the War file that is already generated by the Maven project

The scenario here is the war file is generated by the Maven project running in the Jenkins server.

Under the folder **/var/lib/jenkins/workspace/maven-proj1/target**

Copy the file "mvn-hello-world.war" to the docker folder

In our case we have created a docker folder called "dock01" as below



```
[root@localhost ~]# pwd
/root
[root@localhost ~]# ls -l
total 7268
-rw-------. 1 root root    1446 May 12  2018 anaconda-ks.cfg
-rw-r--r--. 1 root root 2612248 Dec 12 08:00 apache-tomcat-8.5.37-deployer
drwxr-xr-x. 2 root root      51 Apr 21 01:28 dock01
-rw-r--r--. 1 root root      38 Jan 27 13:09 file1
-rw-r--r--. 1 root root 3597945 Mar 17 04:59 jenkins-cli.jar
-rw-r--r--. 1 root root 1212209 Apr 21 01:27 mvn-hello-world.war
-rwxr--r--. 1 root root     132 Mar 17 05:07 script1
-rw-r--r--. 1 root root      33 Jan 24 20:42 source
[root@localhost ~]# cd dock01/
[root@localhost dock01]# ls
Dockerfile  mvn-hello-world.war
[root@localhost dock01]#
```

Command:

$ cp /var/lib/jenkins/workspace/maven-proj1/target/mvn-hello-world.war /root/dock01/mvn-hello-world.war

2. **Create the Dockerfile.**

Now we would need the "Tomcat" image and also the .war file to be copied to the Tomcat "webapps" folder while creating the container.

Below is the Dockerfile content.

============================

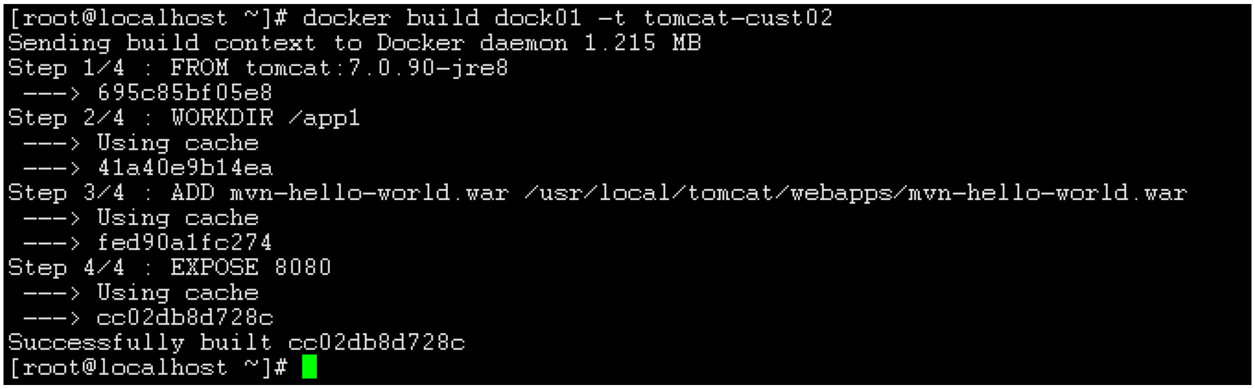
```
FROM tomcat:7.0.90-jre8
WORKDIR    /app1

ADD mvn-hello-world.war /usr/local/tomcat/webapps/mvn-hello-world.war

EXPOSE 8080
```

============================

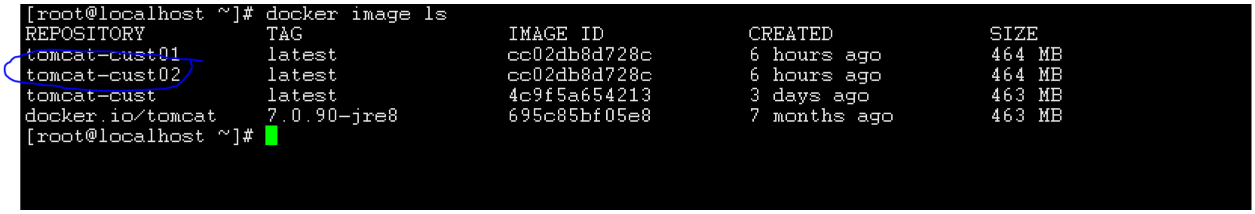
3. Build the Docker image

Run the below command

$ docker build dock01 -t tomcat-cust02



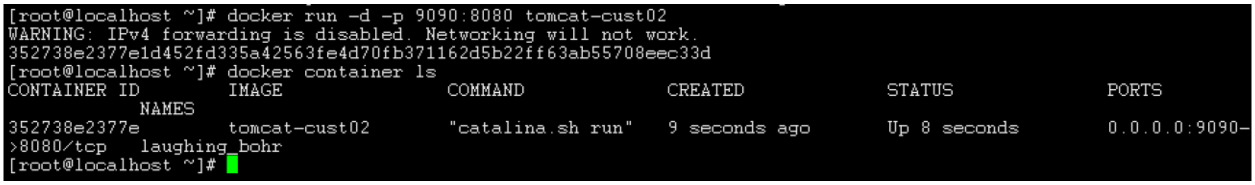
```
[root@localhost ~]# docker build dock01 -t tomcat-cust02
Sending build context to Docker daemon 1.215 MB
Step 1/4 : FROM tomcat:7.0.90-jre8
 ---> 695c85bf05e8
Step 2/4 : WORKDIR /app1
 ---> Using cache
 ---> 41a40e9b14ea
Step 3/4 : ADD mvn-hello-world.war /usr/local/tomcat/webapps/mvn-hello-world.war
 ---> Using cache
 ---> fed90a1fc274
Step 4/4 : EXPOSE 8080
 ---> Using cache
 ---> cc02db8d728c
Successfully built cc02db8d728c
[root@localhost ~]#
```



```
[root@localhost ~]# docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
tomcat-cust01       latest              cc02db8d728c        6 hours ago         464 MB
tomcat-cust02       latest              cc02db8d728c        6 hours ago         464 MB
tomcat-cust         latest              4c9f5a654213        3 days ago          463 MB
docker.io/tomcat    7.0.90-jre8         695c85bf05e8        7 months ago        463 MB
[root@localhost ~]#
```

4. Create the Container

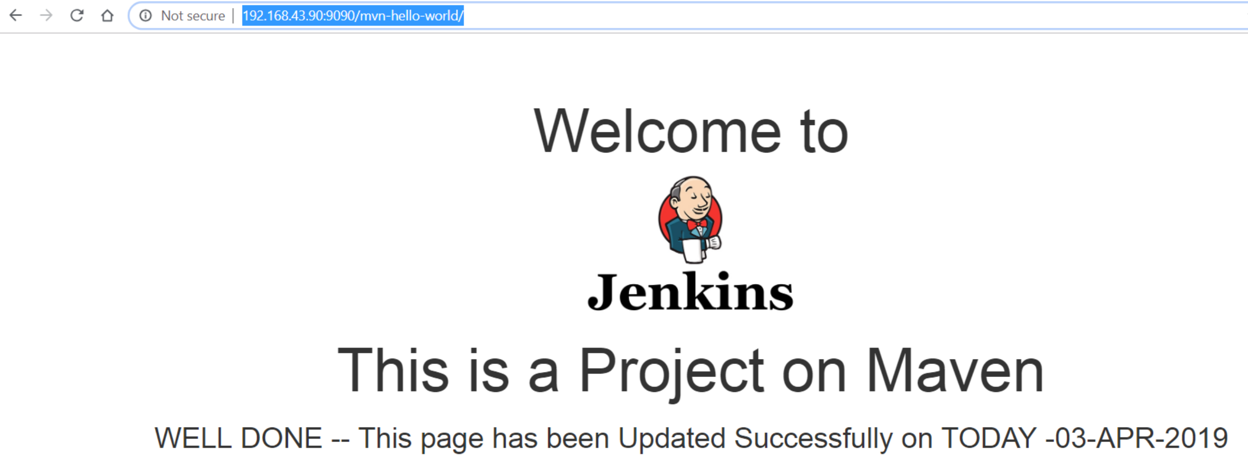
$ docker run -d -p 9090:8080 tomcat-cust02



```
[root@localhost ~]# docker run -d -p 9090:8080 tomcat-cust02
WARNING: IPv4 forwarding is disabled. Networking will not work.
352738e2377e1d452fd335a42563fe4d70fb371162d5b22ff63ab55708eec33d
[root@localhost ~]# docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
           NAMES
352738e2377e        tomcat-cust02       "catalina.sh run"   9 seconds ago       Up 8 seconds        0.0.0.0:9090-
>8080/tcp   laughing_bohr
[root@localhost ~]#
```

5. Test the tomcat output.

http://<Docker server ip>:9090/mvn-hello-world/



Not secure | 192.168.43.90:9090/mvn-hello-world/

Welcome to

Jenkins

This is a Project on Maven

WELL DONE -- This page has been Updated Successfully on TODAY -03-APR-2019

This was very easy and straight forward than setting up the Tomcat server and dependencies.