**Aim of the Jenkins job setup**:

1. Checkout the code.

2. Build the code with maven on another slave.

3. Deploy the code to tomcat with the help of docker.

Dockerfile should take care of creating container.

Install tomcat on the container.

WAR package should be deployed on to the tomcat server.

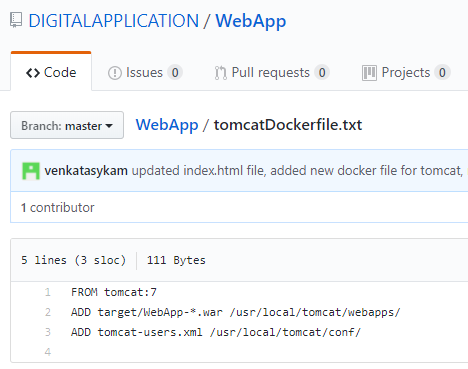
**A.1.GitHub Source Code:** Web application: <https://github.com/DIGITALAPPLICATION/WebApp.git>

**A.2. Docker file:** tomcatDockerfile.txt (keep the docker source in github)

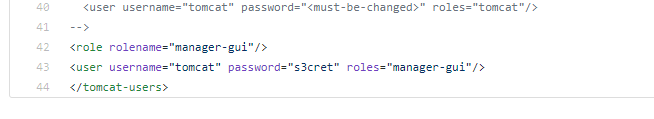
Docker will,

1. install the tomcat server.
2. Add the war package to tomcat/webapps folder.
3. Add the customized tomcat-users.xml file to tomcat/conf.

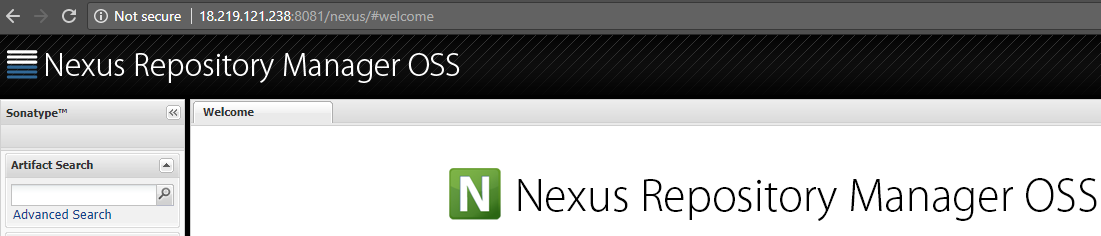
|  |
| --- |
| FROM tomcat:7  ADD target/WebApp-\*.war /usr/local/tomcat/webapps/  ADD tomcat-users.xml /usr/local/tomcat/conf/ |



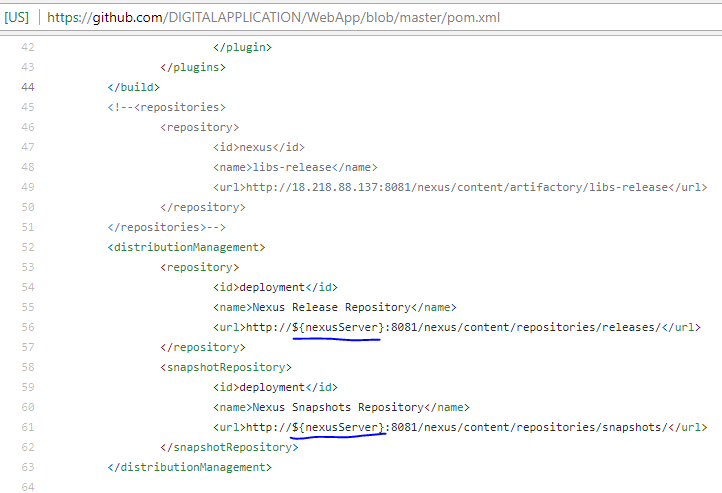
**A.3**. tomcat-users.xml:



**B.1.** My nexus running on the port: [18.219.121.238](http://18.219.121.238)



**B.2. Nexus details on pom.xml file:** Pass the nexus port thru Jenkins job. Or we can directly update the pom.xml with nexus port details.



B.3. settings.xml file (nexus server credentials details): Since the Jenkins job is going to run on the slave, we have to update the settings.xml file in the slave.

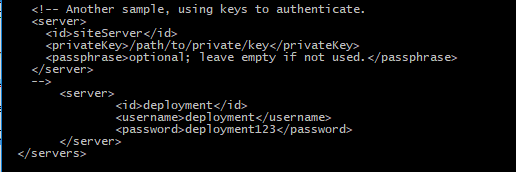
Instead of updating the global settings file which is there in maven home directory, updating the local settings.xml file(i.e., ubuntu user level settings.xml file). For this, copying the settings.xml file from global to .m2 path. (in future if maven is uninstalled from slave, and restrict to use the maven home from master Jenkins, then we can just update the .m2 settings file).

cp /usr/share/maven/conf/settings.xml /home/ubuntu/.m2/settings.xml

vi /home/ubuntu/.m2/settings.xml

add the below snippet b/w <servers>...</servers> as showing in below image.

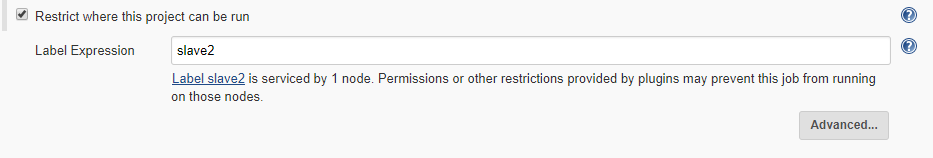
|  |
| --- |
| <server>  <id>deployment</id>  <username>deployment</username>  <password>deployment123</password>  </server> |



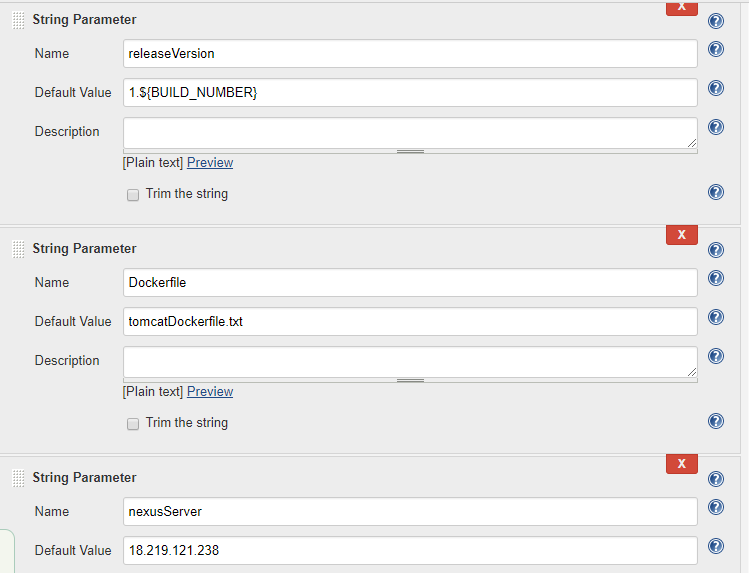
As part of mvn depoy🡪 maven will read the nexus target location details from <distributedManagement> & credentials in settings.xml file <server> etc. Maven will get the server details based on the <id> in both settings.xml & pom.xml file if the settings.xml file contains more than one server details. Please see **D.3. Maven deploy** for more clarity.

**C. Jenkins Job Configuration**:

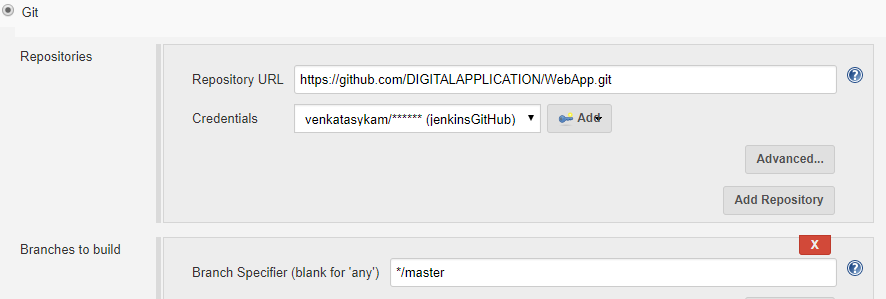
**C.0.** **Slave**:



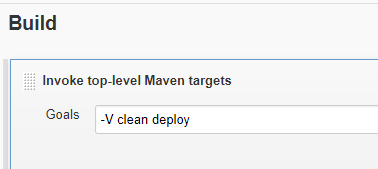
**C.1.Parameters**:



**C.2**. **Source Code management**:



**C.3. Maven goals**:

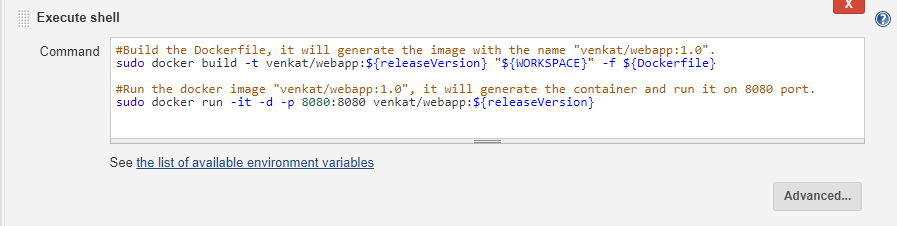


**C.4.** **Docker commands** (Execute Shell):

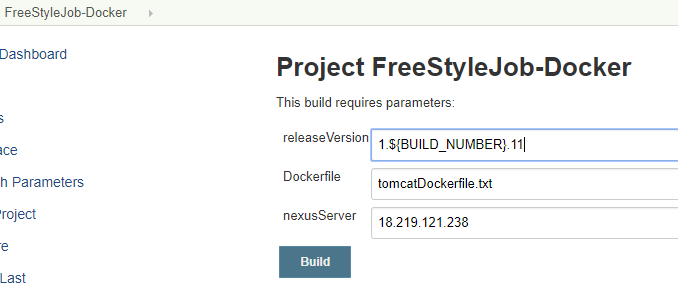
C.4.1. Passing the parameter image tag version i.e., releaseVersion.

C.4.2. Once the code checkout, the source code & dockerfile everything will be placed under the Jenkins workspace. (This workspace created on the slave machine).

C.4.3. There are two Docker files in the GitHub source code for different purpose. Passing the docker file name thru Jenkins parameter.



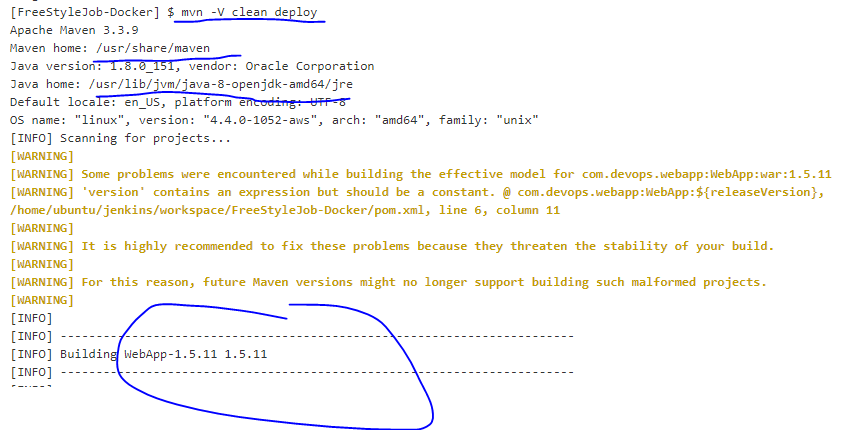
D.1. Trigger the job:



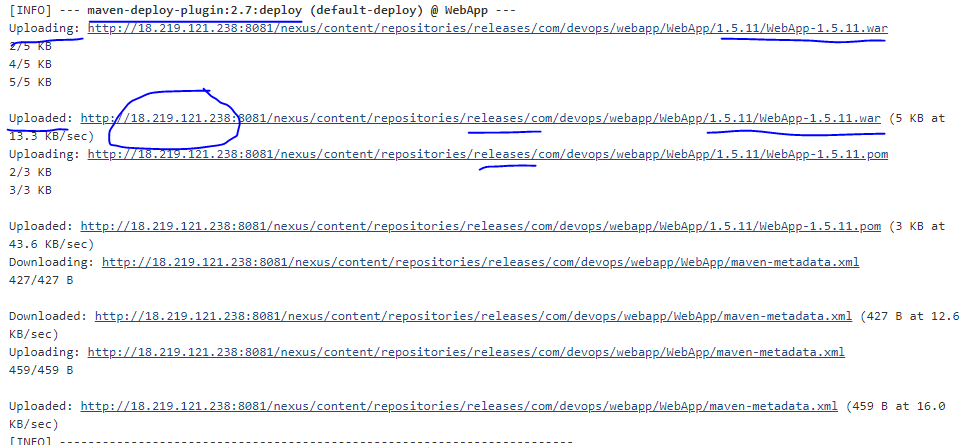
D.2. Build Console: Github code has been checkout into the slave machine.



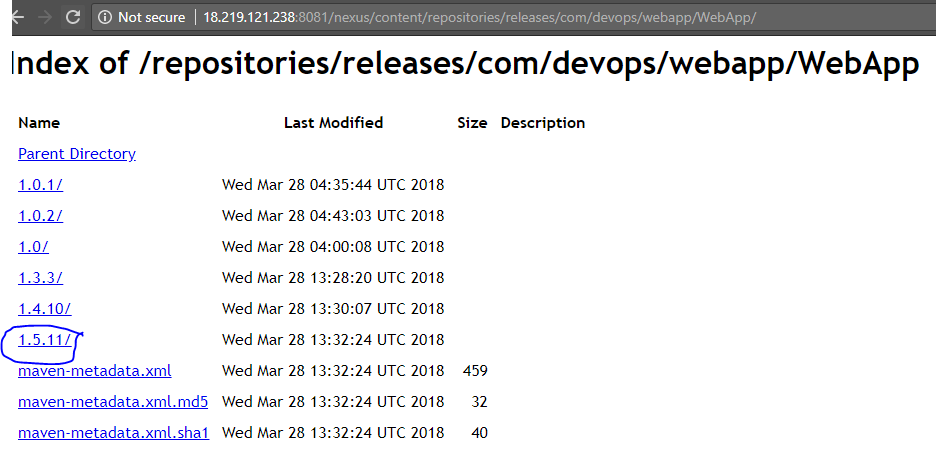
**D.3. Maven build**: Maven home, java home paths are there in slave machine. i.e., maven, java installed on slave.

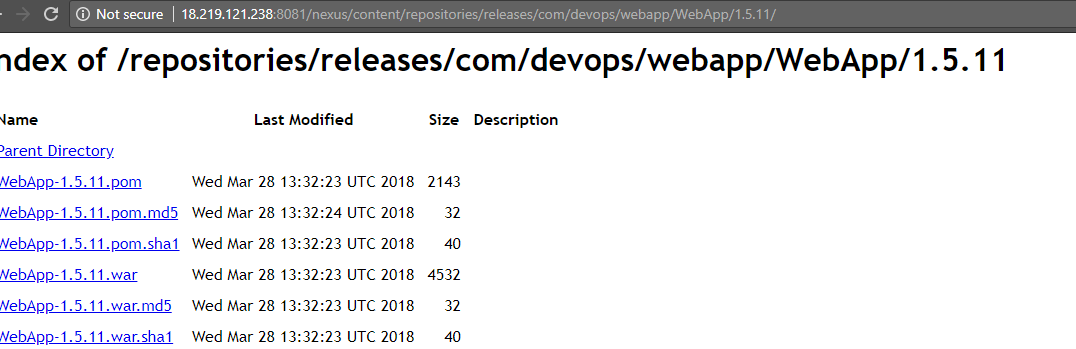


**D.3. Maven deploy**: Once the war package is created, as part of mvn deploy goal 🡪 maven will read the nexus target location details from <distributedManagement> & credentials in settings.xml file <server> etc. Maven will get the server details based on the <id> in both settings.xml & pom.xml file if the settings.xml file contains more than one server details. See **D.4. Nexus**



**D.4. Nexus (artefact search in nexus):**





**D.5. Docker commands**:



**D.6. Run the application from tomcat**:

D.6.1. First go to <http://52.14.5.172:8080/>

D.6.2. Click on Manager Webapp & then enter the credentials as configured in tomcat-users.xml file.

D.6.3. Once you logged in, you will see the apps with different version, click on the app that we recently deployed.

<http://52.14.5.172:8080/WebApp-1.5.11/>

