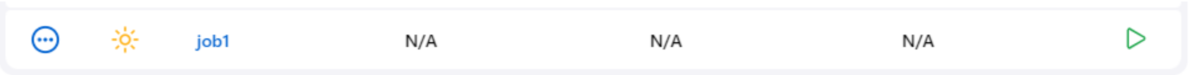
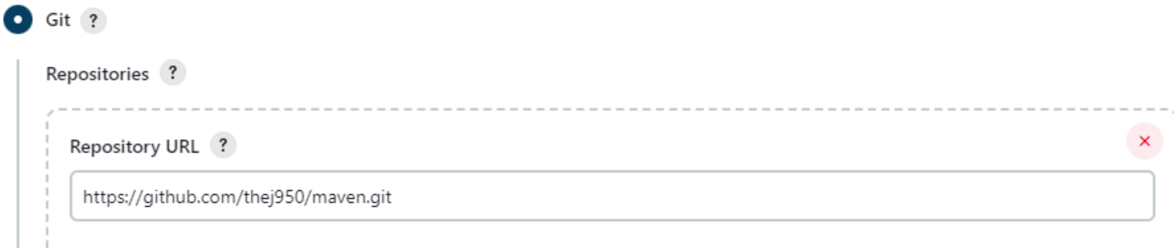
Jenkins FreeStyle Project:

Step1: open jenkins webpage create a freestyle project (job1)



job1 N/A N/A N/A

Step2: open job1 add source code from github



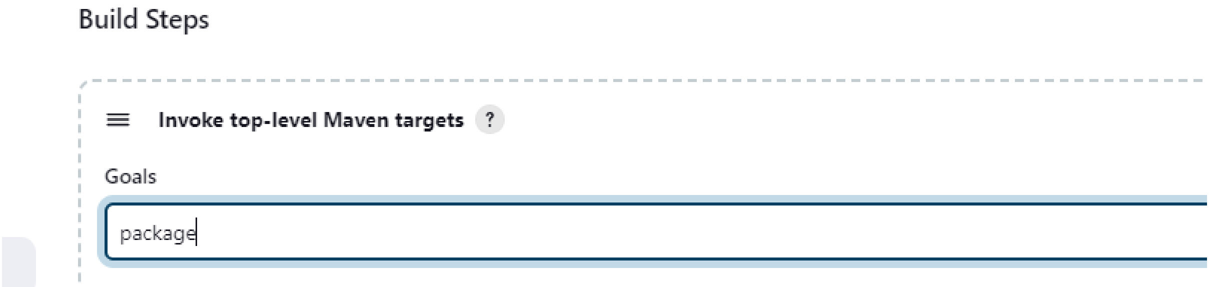
● Git ?

Repositories ?

Repository URL ?

https://github.com/thej950/maven.git

Step3: goto build section build the job with maven package command
Here you need maven plugins



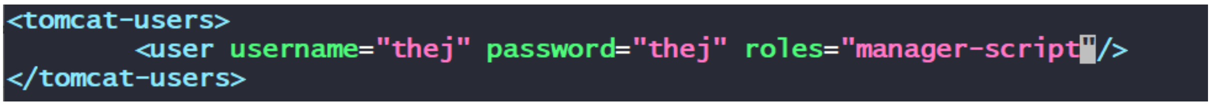
Build Steps

Invoke top-level Maven targets ?

Goals

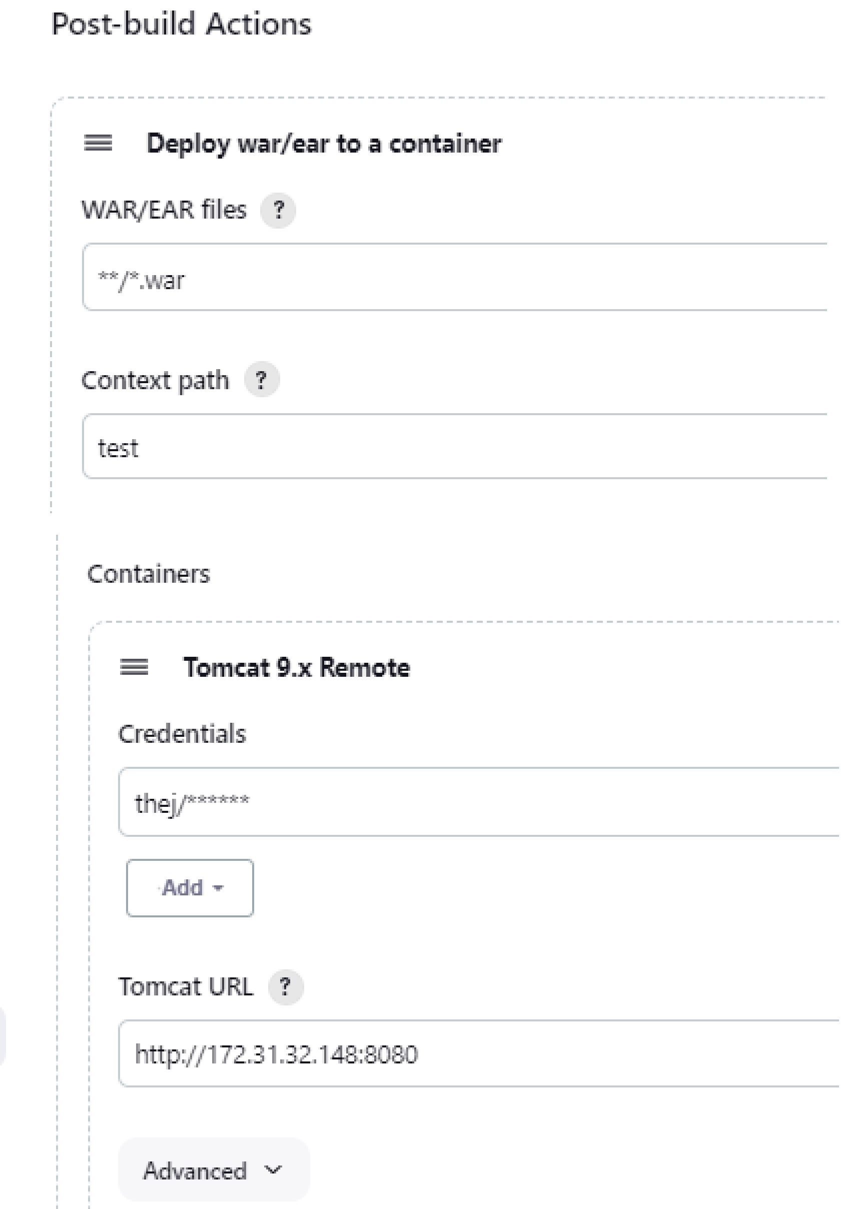
package

Step4: deploy war files into qa servers
Here you need pluggins like Deploy war or ear pluggin
Make sure provide username and password in qa server and prod server
To deploy war files inside qa ad prod server (/var/lib/tomcat9/webapps/)



```
<tomcat-users>
        <user username="thej" password="thej" roles="manager-script"/>
</tomcat-users>
```

Restart tomcat9 service after modify it



Post-build Actions

Deploy war/ear to a container

WAR/EAR files: **/*.war

Context path: test

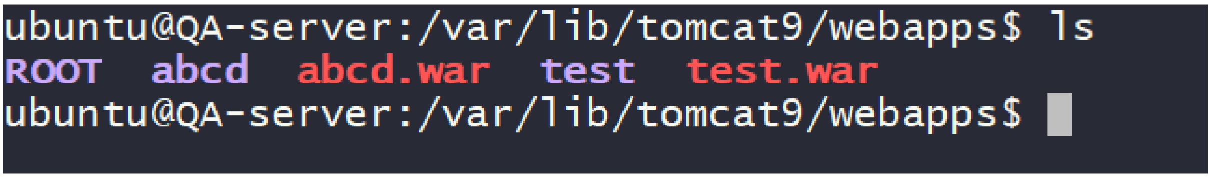
Containers

Tomcat 9.x Remote

Credentials: thej/******

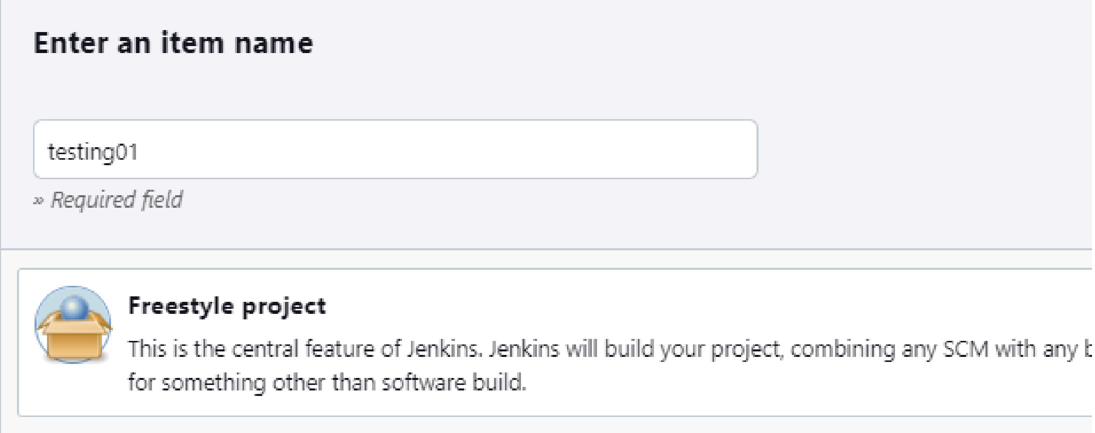
Tomcat URL: http://172.31.32.148:8080

After deploying war files inside /var/lib/tomcat9/webapps a context path will be shown as .war file by using that we can able access from server



```
ubuntu@QA-server:/var/lib/tomcat9/webapps$ ls
ROOT  abcd  abcd.war  test  test.war
ubuntu@QA-server:/var/lib/tomcat9/webapps$
```

Step5: Now create a testing job to perform testing for our code download testing scripts in jenkins



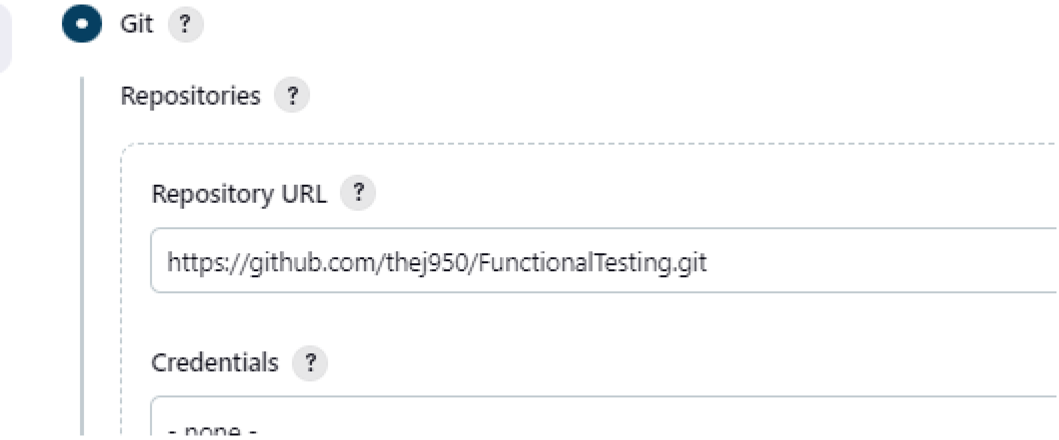
Enter an item name

testing01

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any b
for something other than software build.



● Git ?

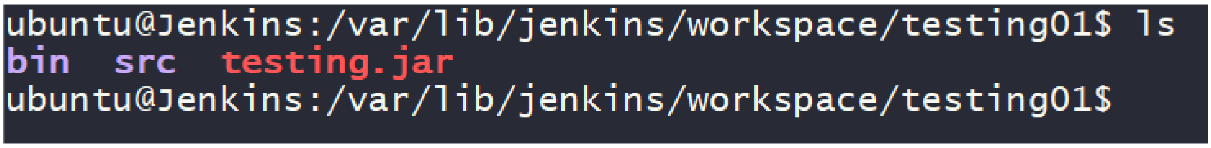
Repositories ?

Repository URL ?

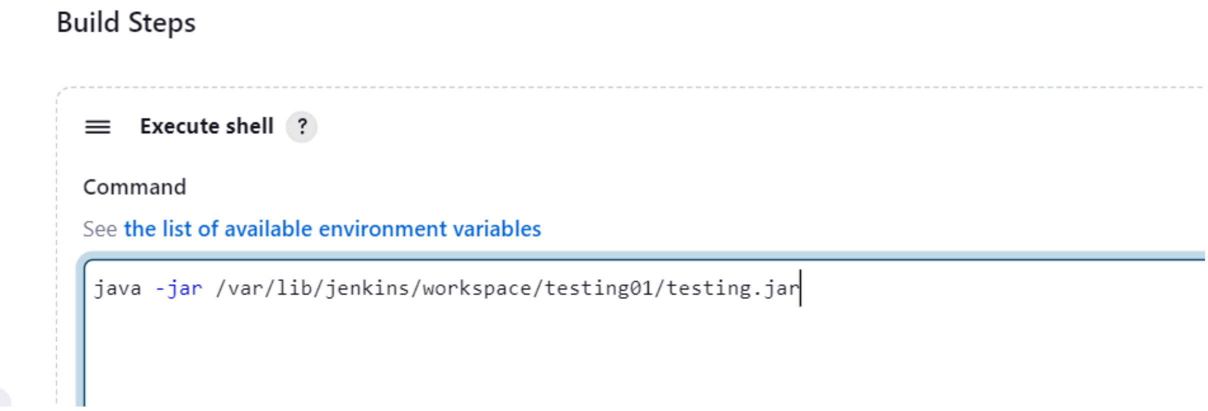
https://github.com/thej950/FunctionalTesting.git

Credentials ?

- none -



```
ubuntu@Jenkins:/var/lib/jenkins/workspace/testing01$ ls
bin  src  testing.jar
ubuntu@Jenkins:/var/lib/jenkins/workspace/testing01$
```



Build Steps

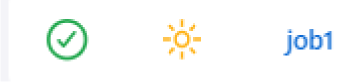
Execute shell ?

Command

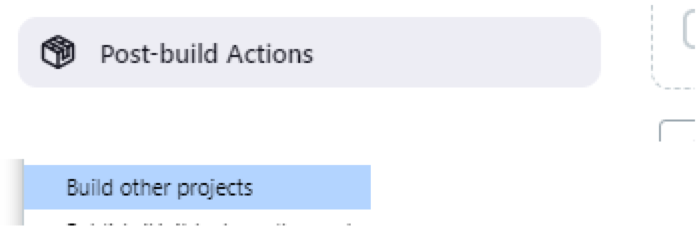
See the list of available environment variables

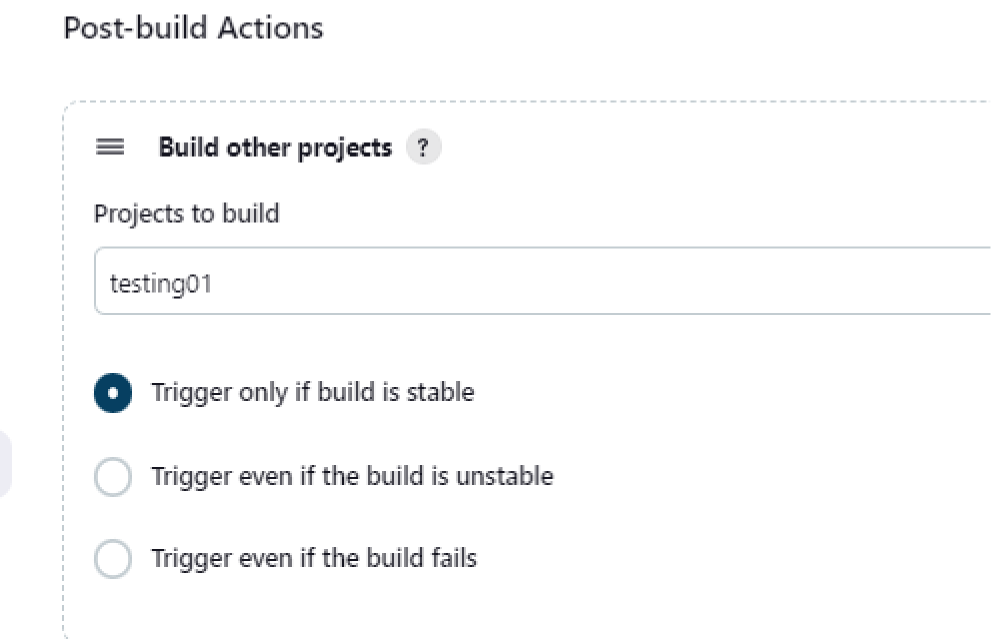
```
java -jar /var/lib/jenkins/workspace/testing01/testing.jar
```

step6: Now build other jobs from Developement job like (job1)



✓ ☼ job1





Post-build Actions

Build other projects ?

Projects to build

testing01

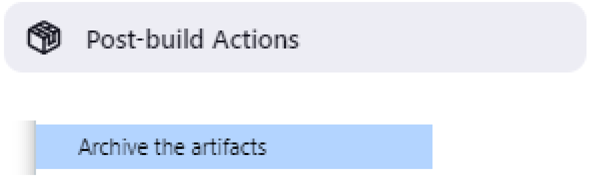
● Trigger only if build is stable

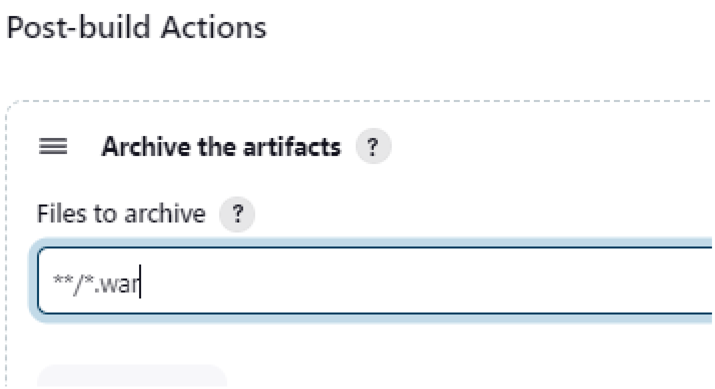
○ Trigger even if the build is unstable

○ Trigger even if the build fails

From here when you build job1 automatically after completing job1 it will build testing job also

Step7: Now goto job1 and make archive war file in jenkins to send to another server





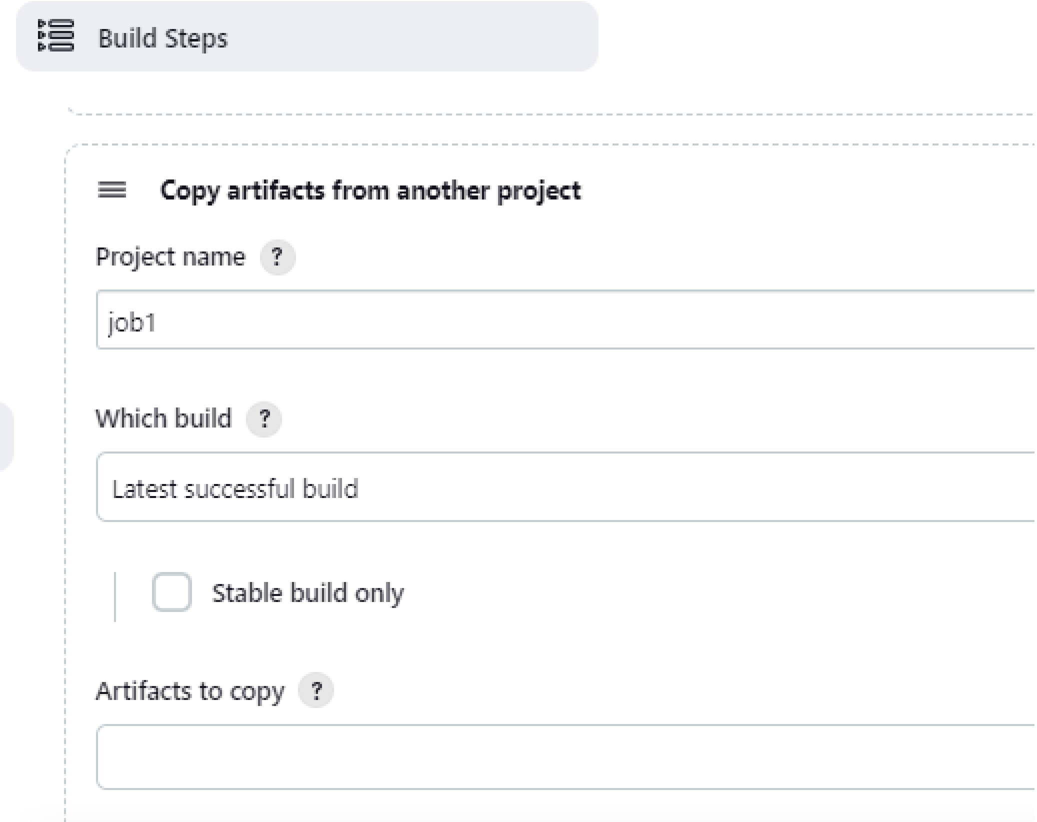
Post-build Actions

Archive the artifacts ?

Files to archive ?

**/*.war

Step8: Now goto testing job setup copy files from another machine
For this we need a pluggin called copy artifacts plugin



Build Steps

Copy artifacts from another project

Project name ?

job1

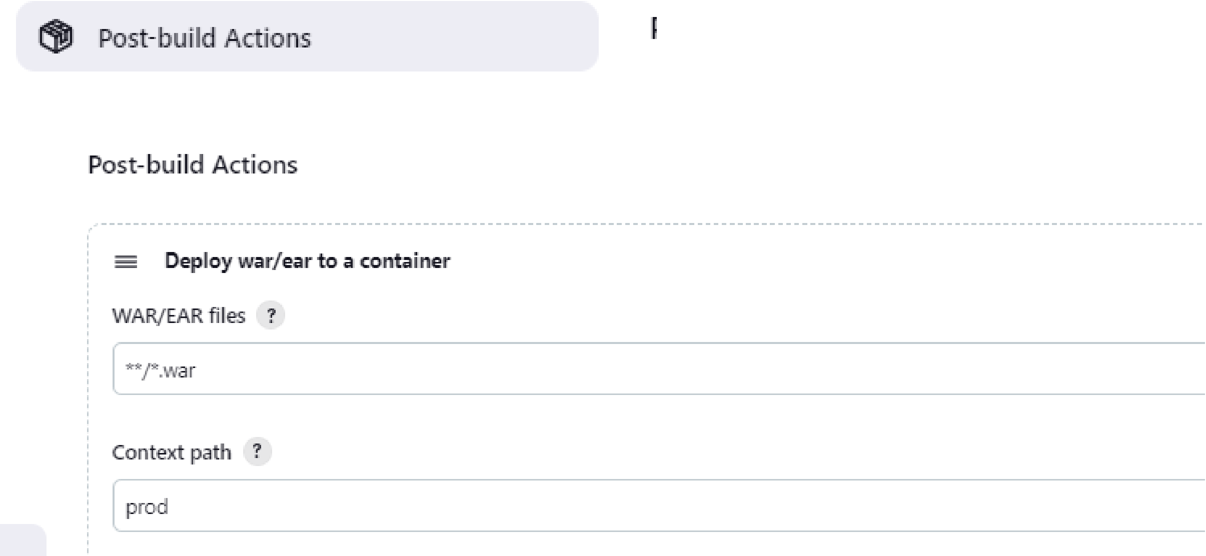
Which build ?

Latest successful build

□ Stable build only

Artifacts to copy ?

Step9: deploy into prod server



Post-build Actions

Post-build Actions

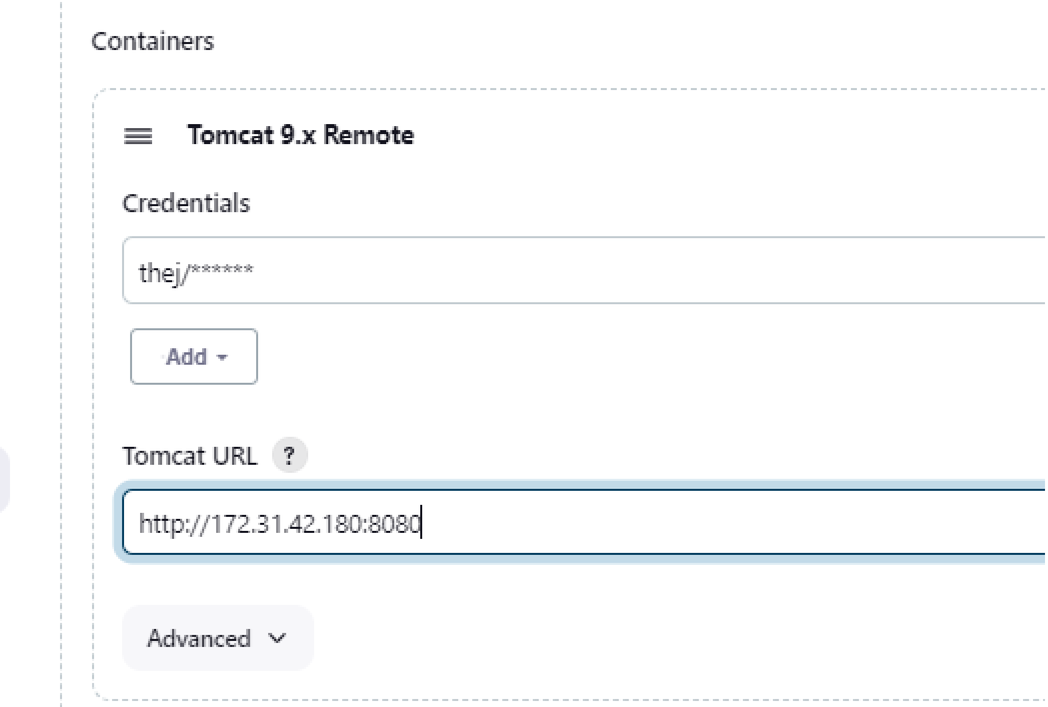
Deploy war/ear to a container

WAR/EAR files ?

**/*.war

Context path ?

prod



Containers

Tomcat 9.x Remote

Credentials

thej/******

Tomcat URL

http://172.31.42.180:8080