**CI CD using Jenkins**

**Step 1: Create a EC2 Linux AMI t2.micro , Install tomcat. Verify if you can login. (Make sure you open port 80,443,22. For demo you can use default security group)**

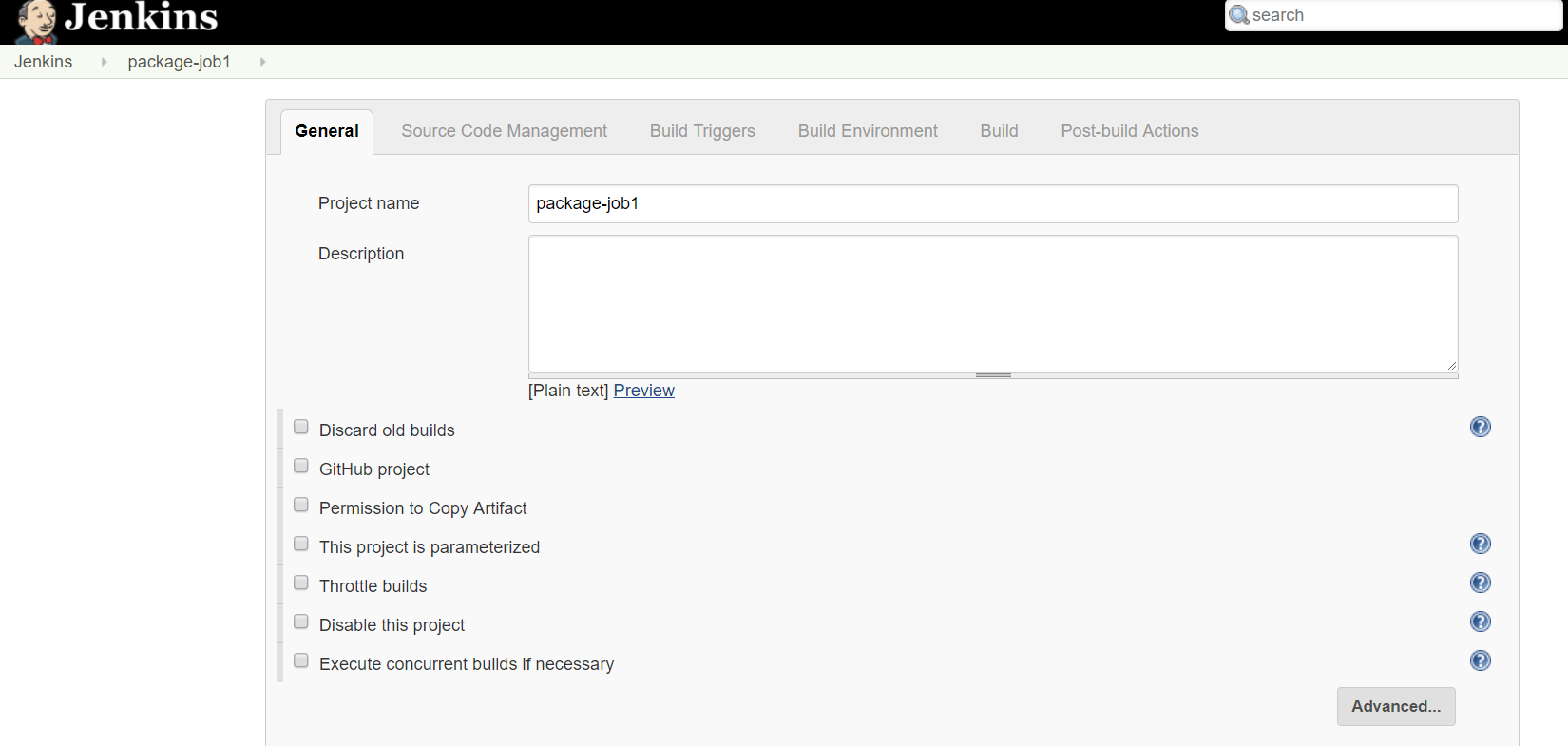
<Public IP-Tomcat Server>:8080

Then click on Manager App and provide tomcat credentials.

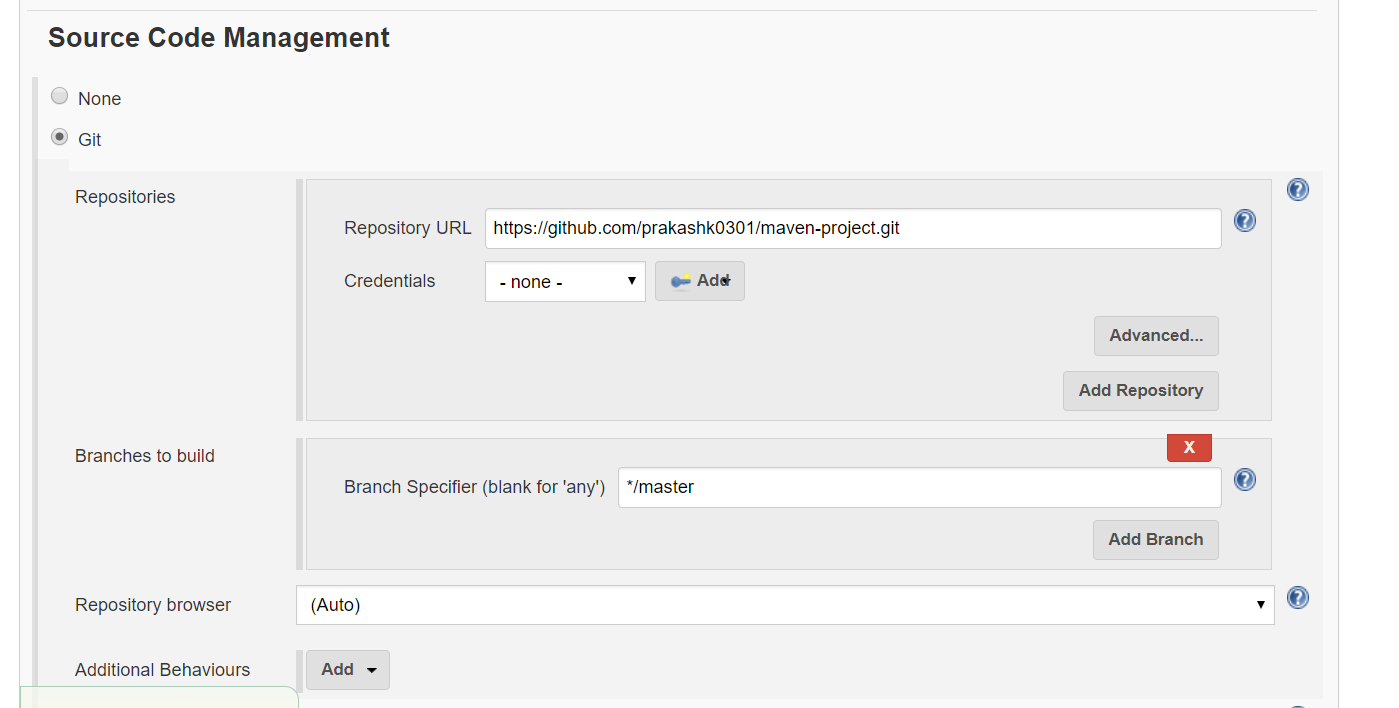
(refer tomcat-user.xml, where username="admin2" password="tomcat". You can update accordingly)

**Step 2: Go to Jenkins Dashboard->Manager Plugin-> Download Deploy to container plugin and copy artifact plugin.**

**Step 3: create a Jenkins Job1 (EX: package-job1):**

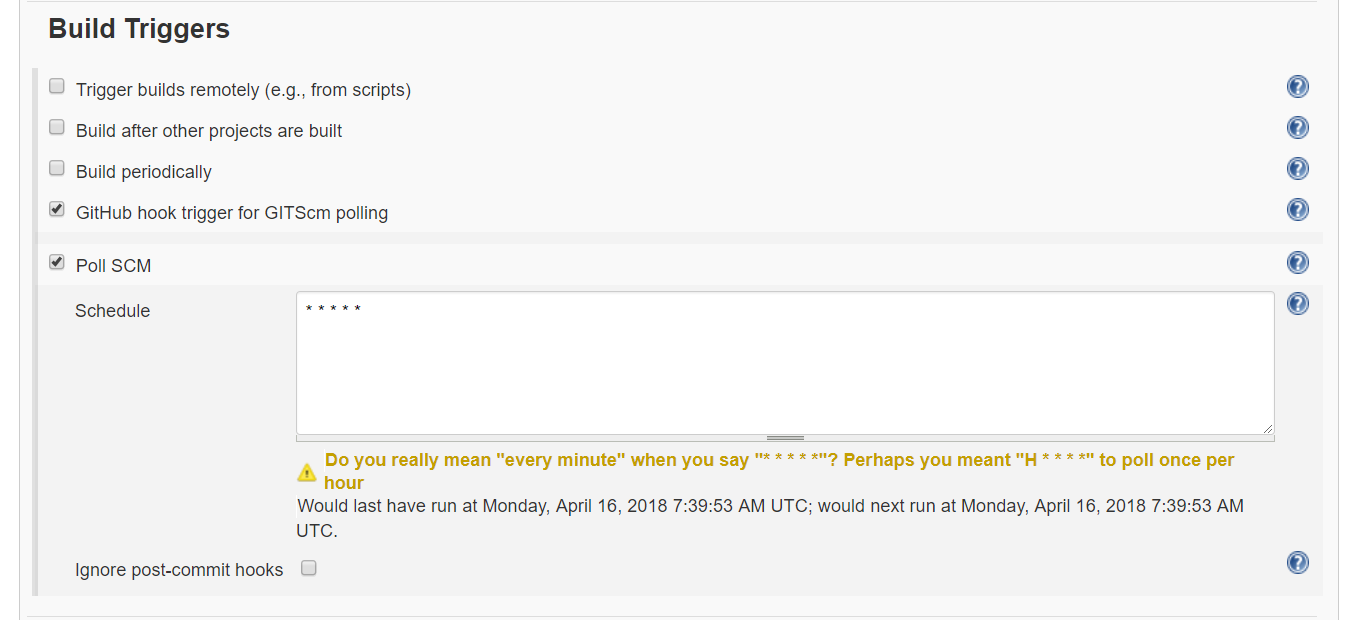
Job1: This is the job which on build- Create a build of all the files from the git repo.  
Configurations, 

Source code Management: Update according to your project. (In our case repo is public so no need to provide credentials, but if repo is private then you have to provide credentials)



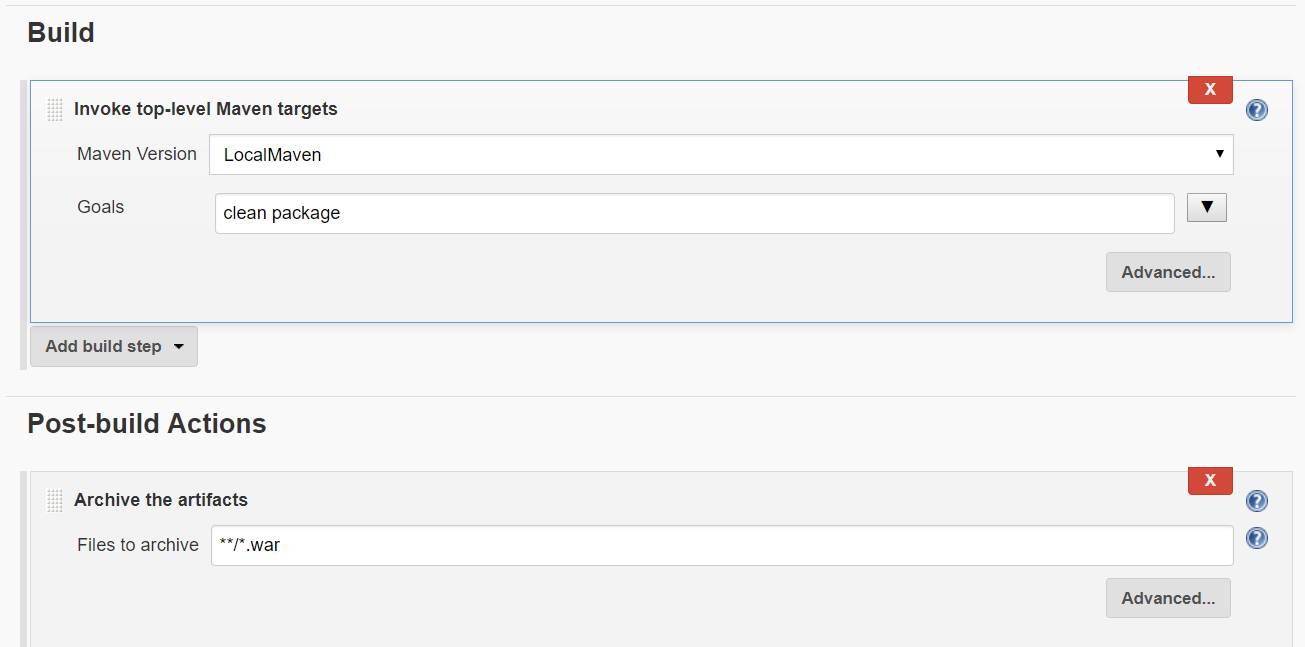
Build Trigger: This feature enables builds after [post-receive hooks in your GitHub repositories](http://help.github.com/post-receive-hooks/). This trigger only kicks git-plugin internal polling algo for every incoming event against matched repo.

Poll SCM: When you want to build your job. (Here **\* \* \* \* \*** , poll per minute.)



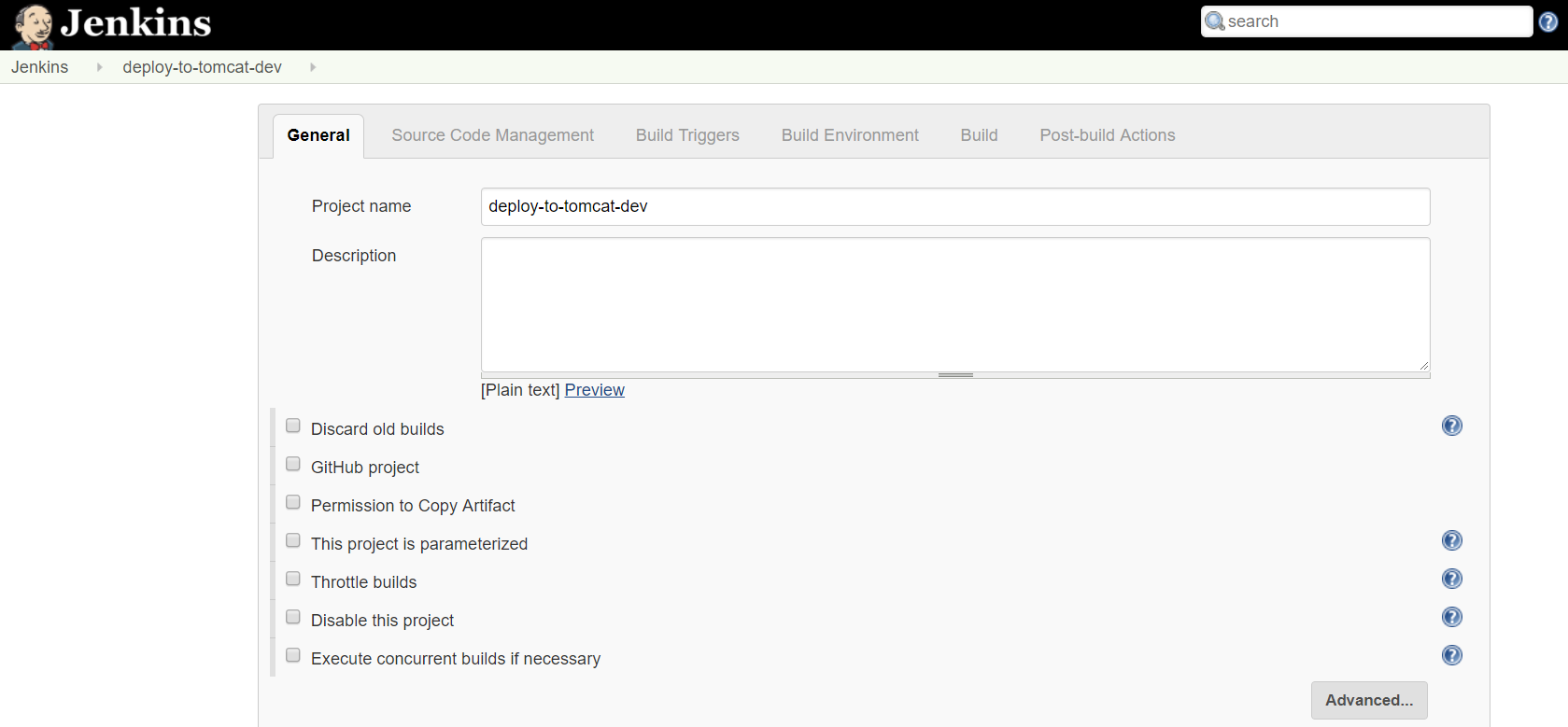
Here we will be using Maven build tool. and our goal is Package.

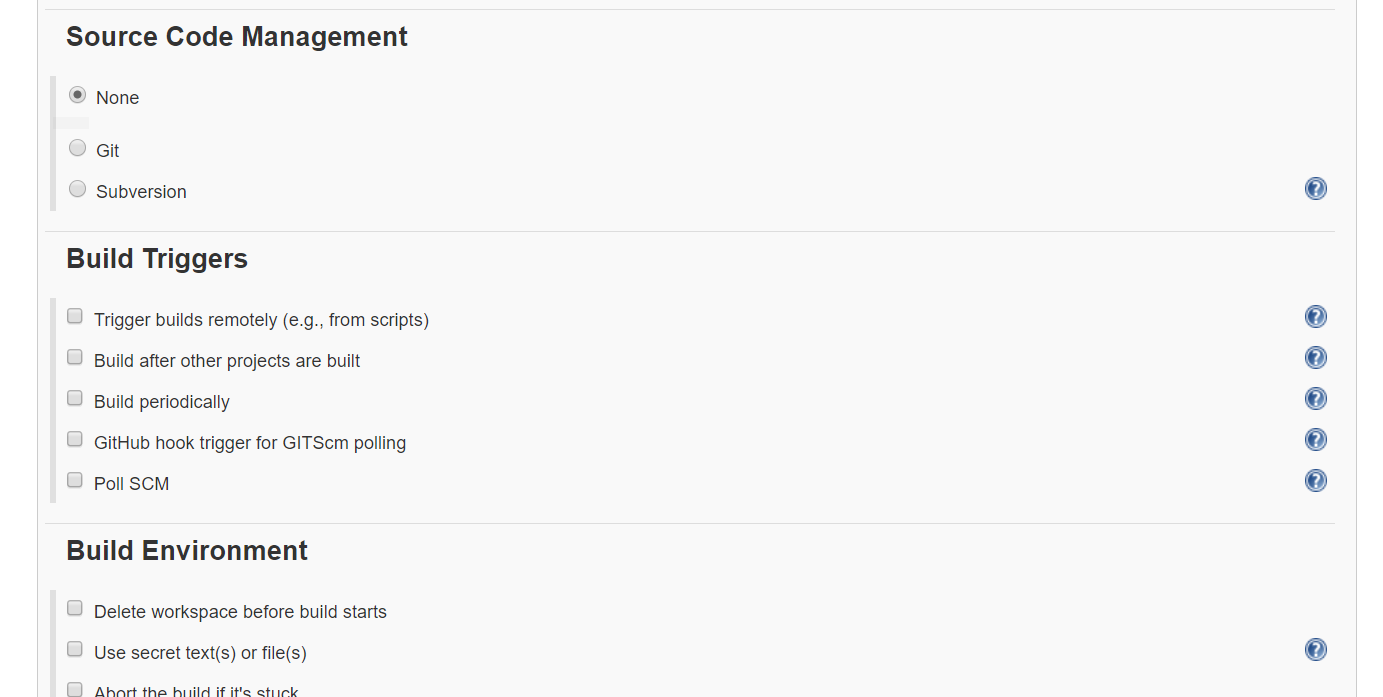
Artifact- War file.



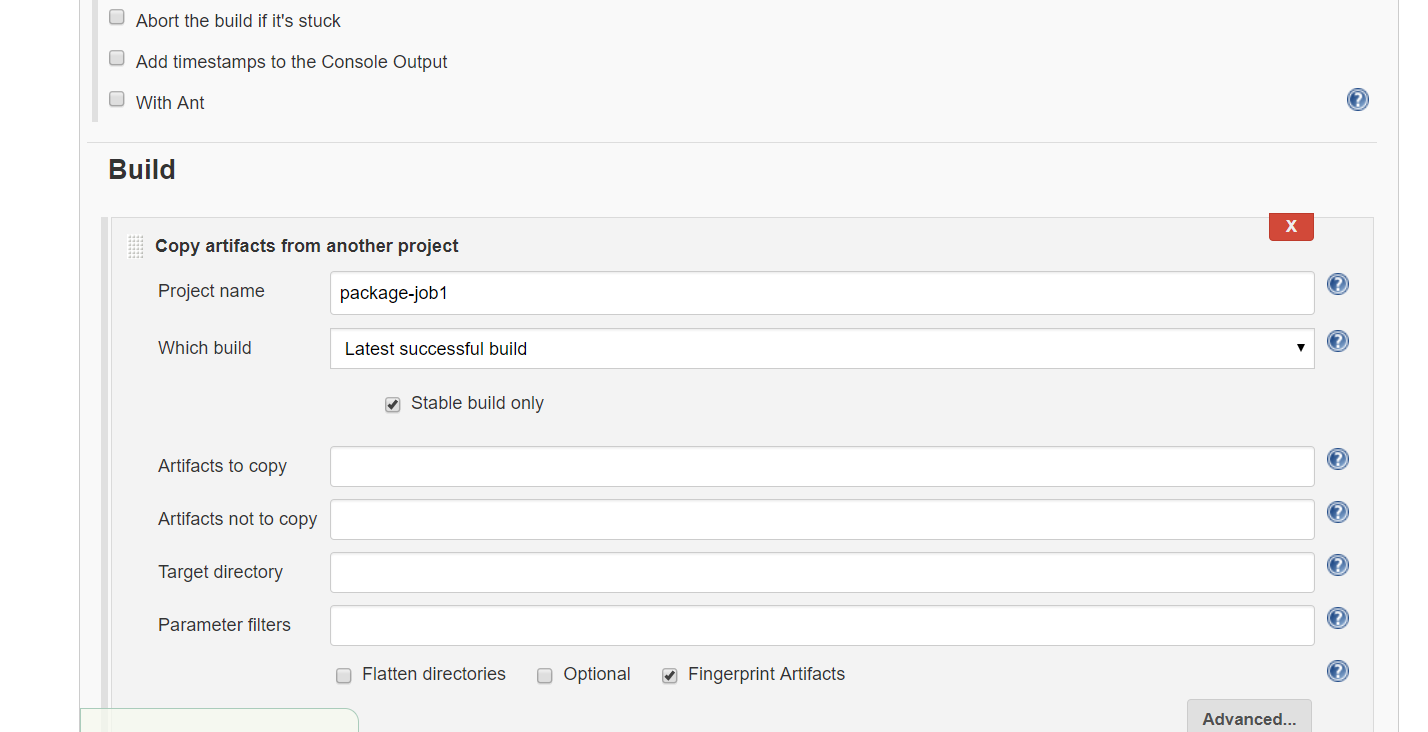
And Save your job 1.

**Step 4: Create another Jenkins job 2 (EX: deploy-to-container-dev).**



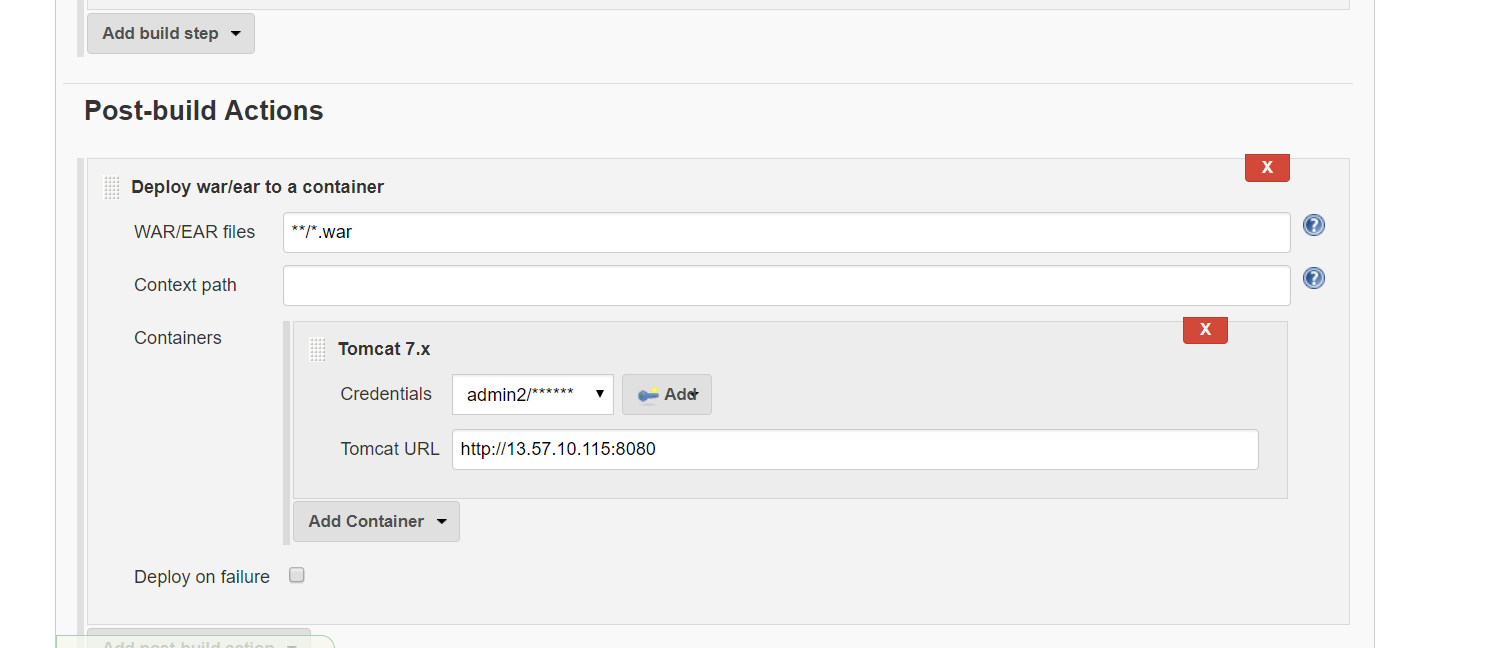


In Build section- Copy artifact from your previous job



In post build section-Where you want to deploy your artifact. In our case we want to deploy artifact to tomcat server (considering as Test or Dev or UAT or Prod environment server)

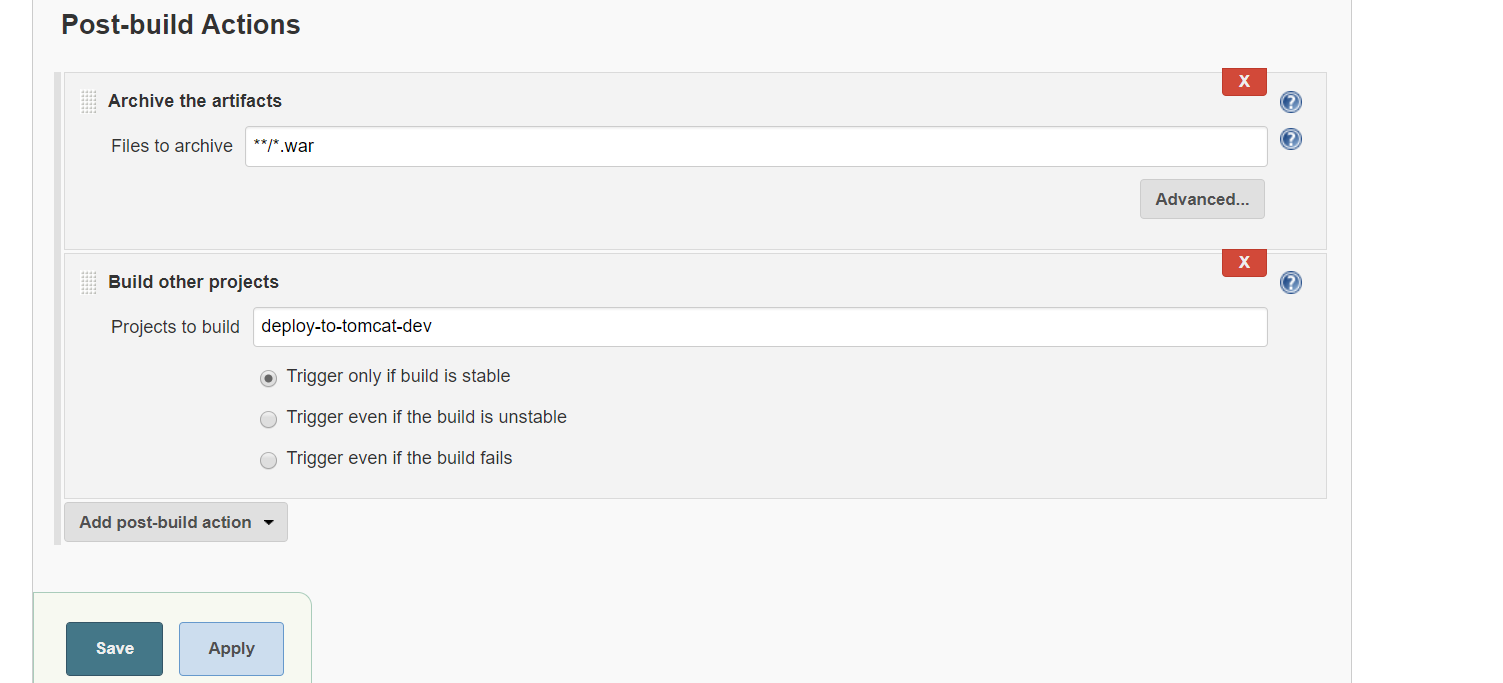
You need to click on Add -> provide your tomcat server Credentials, refer step 1



And save Your Job 2.

**Step 5: Go back to Jenkins Job 1-> Configure**

In post build section-> project to build -> select your job 2. And Save



Step 6:

Build Job1 first.  
  
Job1 should trigger Job2.  
  
Job2 should display webpage on tomcat server.  
  
Verification: Open browser: <public IP-tomcat server>:8080/webapp

**Optional**

Step 7: Install git msi on your laptop

Open command prompt (cmd)

Create new directory

mkdir march\_devops18

cd march\_devops18

git init

Clone your github project

git clone https://github.com/prakashk0301/maven-project.git

(update your github repository url, not this one)

now go to your laptop and check for folder maven-project

ex: C:\Users\Prakash\_pc\march\_devops18\maven-project\webapp\src\main\webapp

Now open index.jsp in notepad ++

Edit index.jsp and save

git add index.jsp

git commit -m “changes ”

git push <https://github.com/prakashk0301/maven-project.git>

and wait for 1 minute, refresh tomcat server url)