

**Note:** This document demonstrating the git, maven and maven web project and tomcat deployment through Jenkins integration Manually.

**GIT Maven and Tomcat Integration with Jenkins Manully**

**Chapters:**

Install Java

Install Maven

Install Jenkins # Refer Jenkins Notes.txt in github devops docs repo

Install Tree command

Generate Maven Quick Start and Web app project using maven command

Install Git

Install Tomcat

Create Jenkins job with Maven Integration

**Install Java in Linxu:**

$ yum install java-1.8\*

$ java –version

**Install Maven in Linux:**

Please refer the below link for Maven

<https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

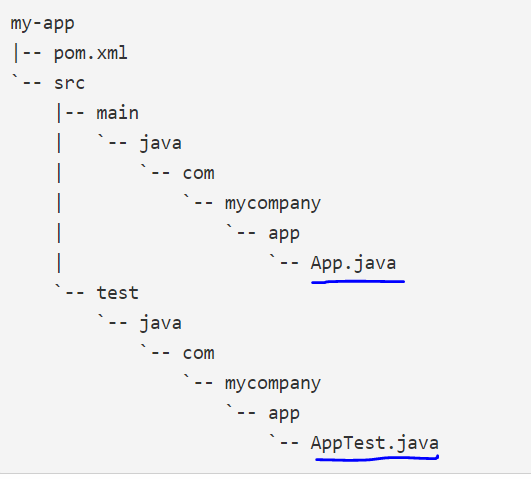
* Yum install maven -y
* mvn –version

**Generate Maven quick start project using below command:**

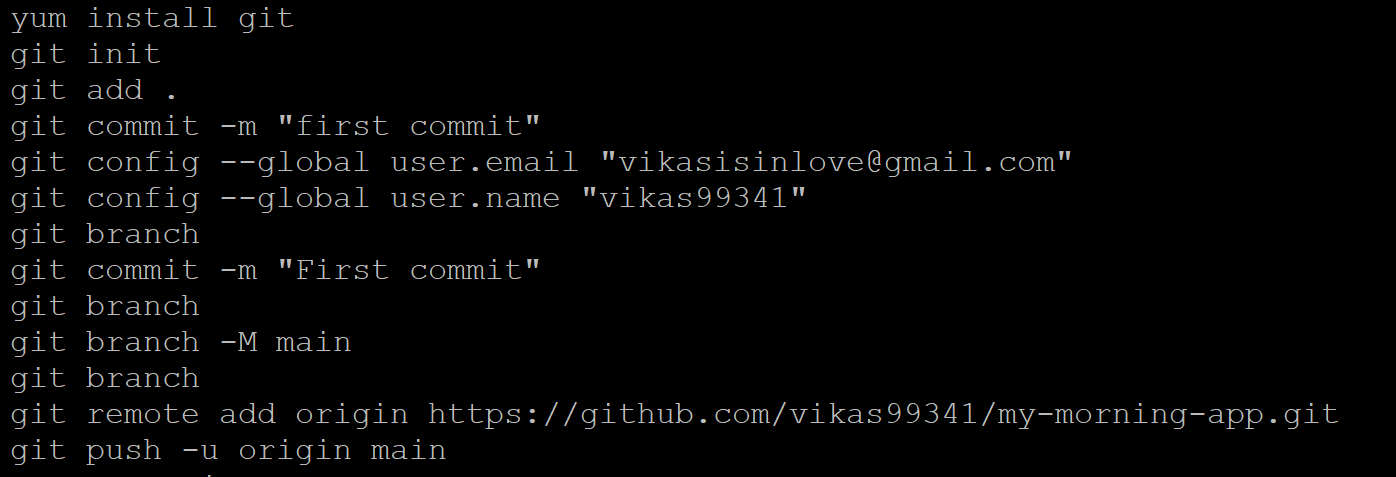
mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false

**Install tree command**

* yum install tree -y

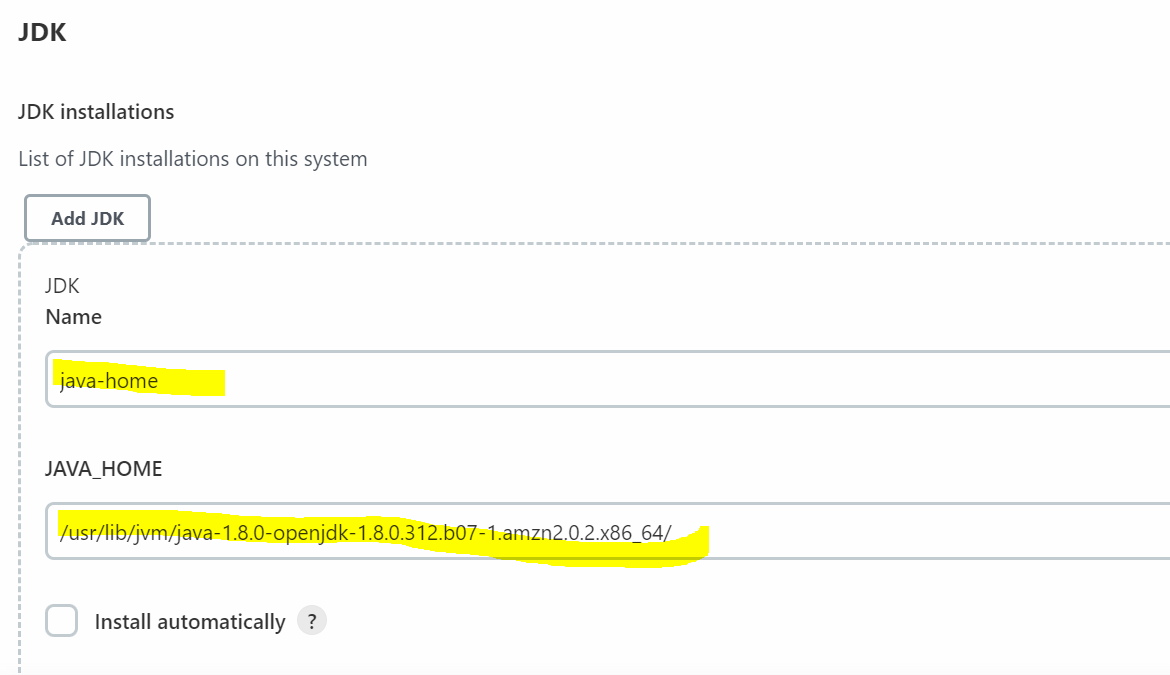


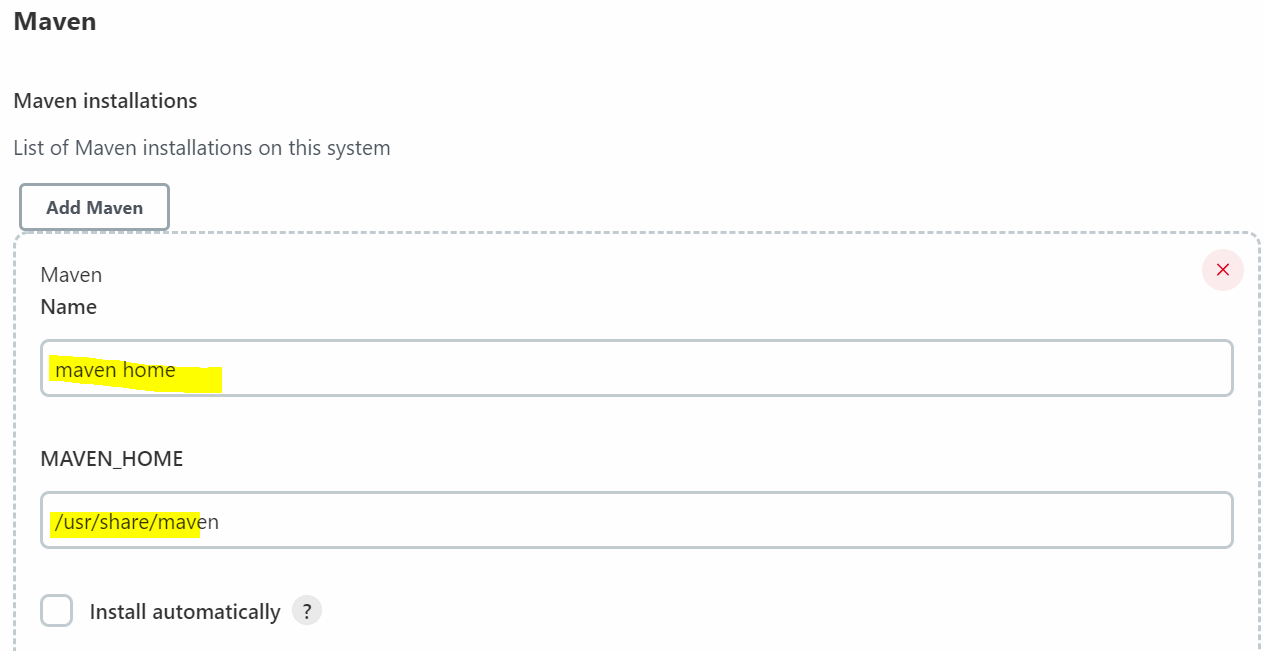
Install Git:

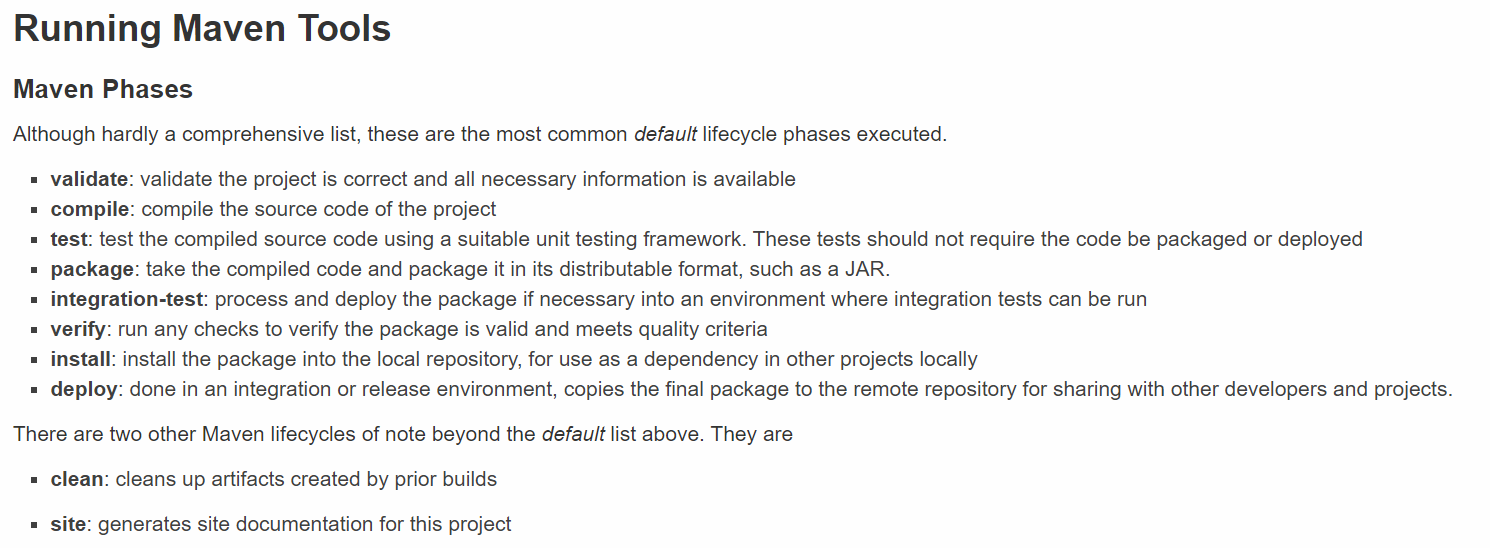


**Note:** Perquisites for Jenkins and Maven integration

First go to Jenkins Dashboard ---> Global Tool Configuration and Set Maven and Java path accordingly. See the below screenshots for the same







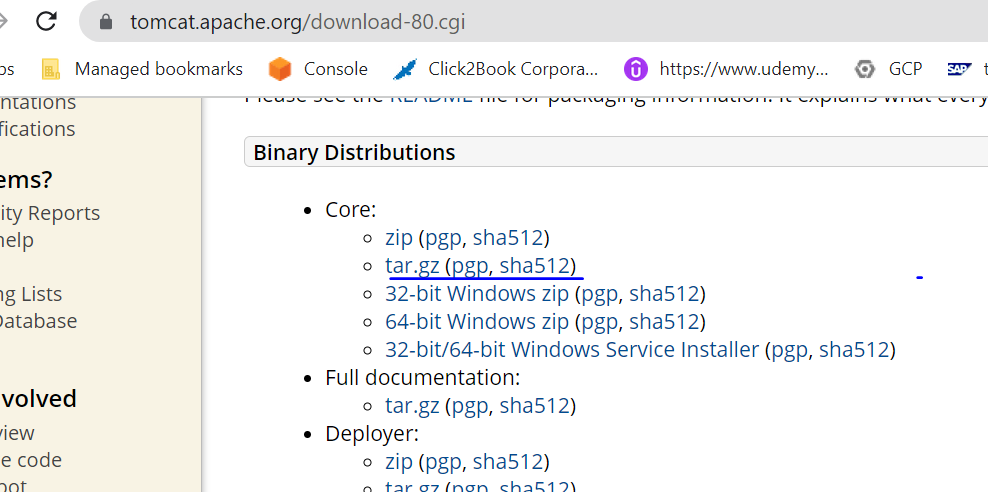
**Tomcat Installation Steps:**

Official website: https://linuxhint.com/installation-of-tomcat-ec2-amazon-linux/

<https://medium.com/@raguyazhin/step-by-step-guide-to-install-apache-tomcat-on-amazon-linux-120748a151a9>

**Tomcat Official Website**- <https://tomcat.apache.org/download-90.cgi>

Right click on highlighted one tar.gz and copy as link address



* Launch a EC2(**t2** **small** minimum **2GB RAM**) Instance and install tomcat on that.

**Jav1.8 :** yum install java-1.8\* -y //install either of one

**Java11 :** sudo amazon-linux-extras install java-openjdk11 -y

java –version

Java installation path : /**usr/lib/jvm**

* Now change the directory to **opt** folder to get install the tomcat server

cd /opt

wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.74/bin/apache-tomcat-9.0.74.tar.gz

unzip in using below command

tar -xvzf apache-tomcat-9.0.59.tar.gz

* Once the file has been extracted, change its name for easy use using the following command:

mv apache-tomcat-9.0.59 tomcat9

* remove the tomcat tar file after unzip it

rm -rf apache-tomcat-9.0.59.tar.gz

* After that, create a new user for tomcat using the following command:

$sudo useradd –r tomcat

* Once the user is created, we want to allow this user to access our tomcat folder, for that use the following command:

**chown -R tomcat:tomcat /opt/tomcat9**

* The next step is to change the “tomcat.service” file, for that first type the following common:

sudo tee /etc/systemd/system/tomcat.service<<EOF

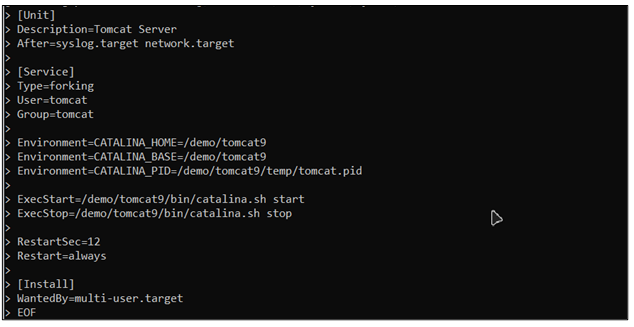
* Executing this command will allow us to type the configuration inside the terminal until we type EOF and hit enter:



Now, type the following lines:

[Unit]  
Description=Tomcat Server  
After=syslog.target network.target  
  
[Service]  
Type=forking  
User=tomcat  
Group=tomcat  
  
Environment=CATALINA\_HOME=/opt/tomcat9  
Environment=CATALINA\_BASE=/opt/tomcat9  
Environment=CATALINA\_PID=/opt/tomcat9/temp/tomcat.pid  
  
ExecStart=/opt/tomcat9/bin/catalina.sh start  
ExecStop=/opt/tomcat9/bin/catalina.sh stop  
  
RestartSec=12  
Restart=always  
  
[Install]  
WantedBy=multi-user.target  
EOF

After that, simply hit enter and the terminal will look like this:



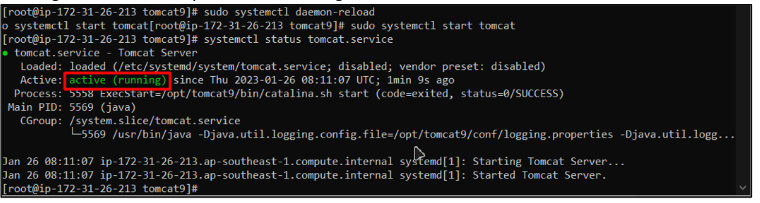
After that, restart the services of tomcat by using the following commands:

sudo systemctl daemon-reload  
sudo systemctl start tomcat

After that, verify that the Apache Tomcat is running by checking the service status of tomcat:

systemctl status tomcat.service

Running the command will produce the following results:



The status is running, which means that the apache tomcat service has been successfully installed and is running.

**Note:**If you want to access the tomcat server from outside, then make sure to allow port 8080 for the inbound rules of the EC2 instance.

**Note**: This is direct approach to run the tomcat server from installation pat. Incase above case not work then follow the below approach2.

**Approach2**:

* cd tomcat9/ #this is not user this is tomcat9 installation path
* ls –ltr
* if we want to start tomcat server go to bin folder and run the ./startup.sh file

cd bin/

./startup.sh

**How do I know which port Tomcat is running?**

where 8080 is the Tomcat port specified in conf**/server. Xml**

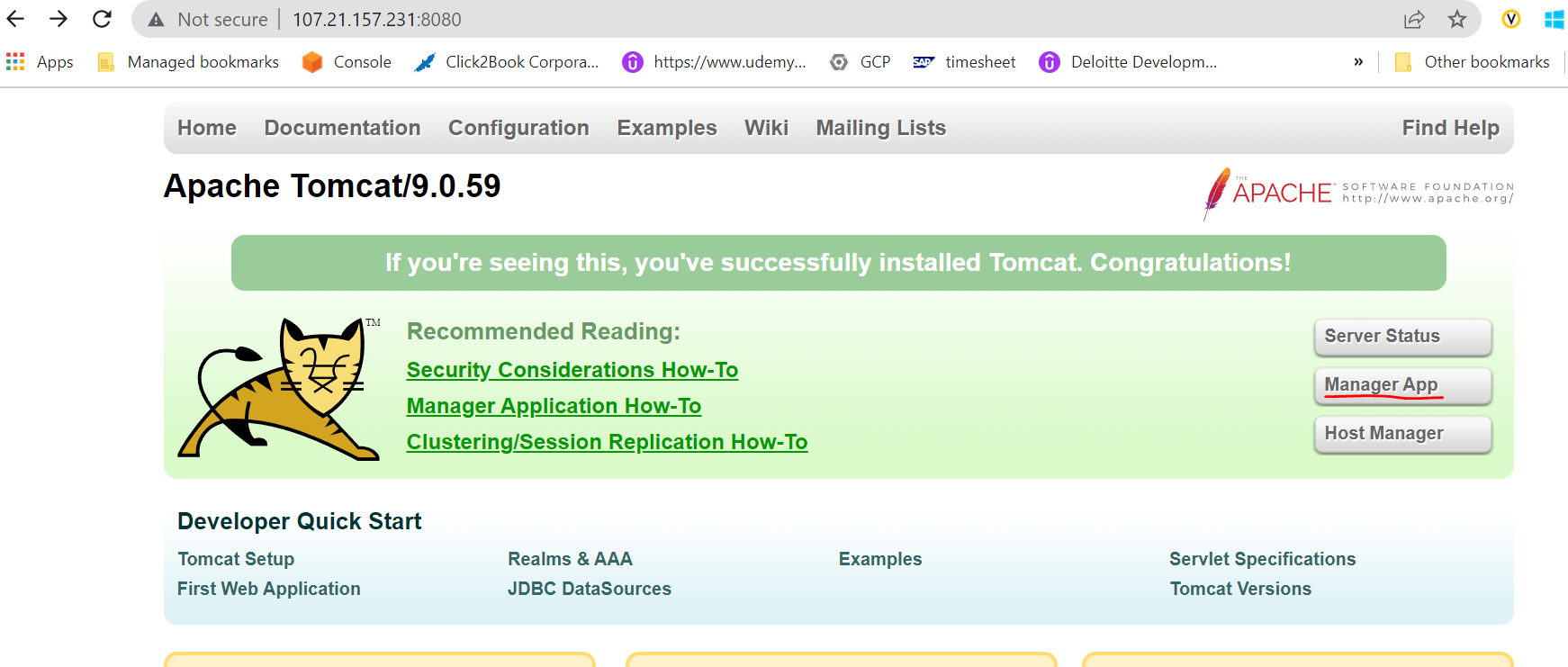
Now go to browser and copy the ec2 server public ipv4 add suffix :port

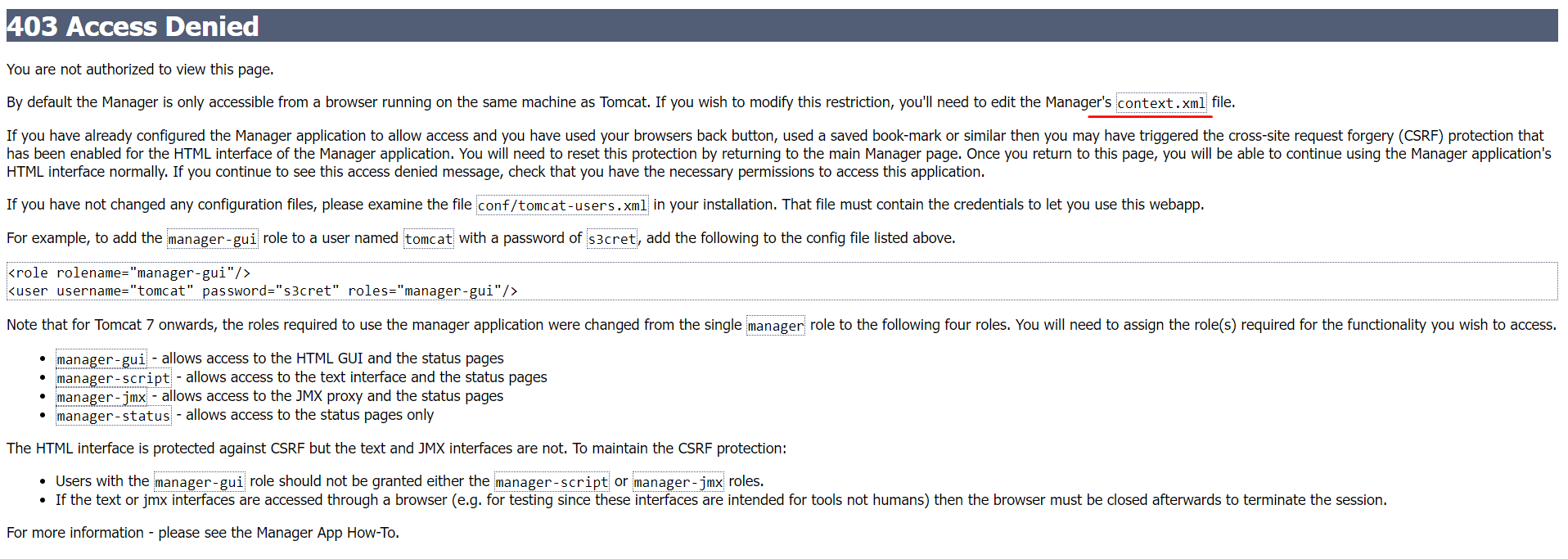
**https:// 43.205.192.163:8080**

**If you’r seeing the site can’t be reached**

Then go to security group🡪 in-bound rules and add new rule for 8080 using customer tcp

default username: admin and pwd: admin

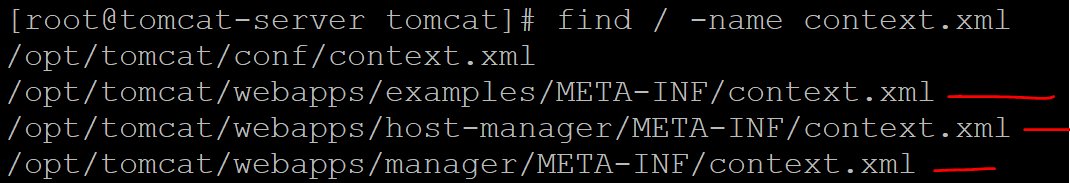
* 



If you’re seeing above error after clicking on any Manager App anywhere then issue the following command and do the necessary below changes in the below files.

cd opt/tomcat

$ find –name context.xml



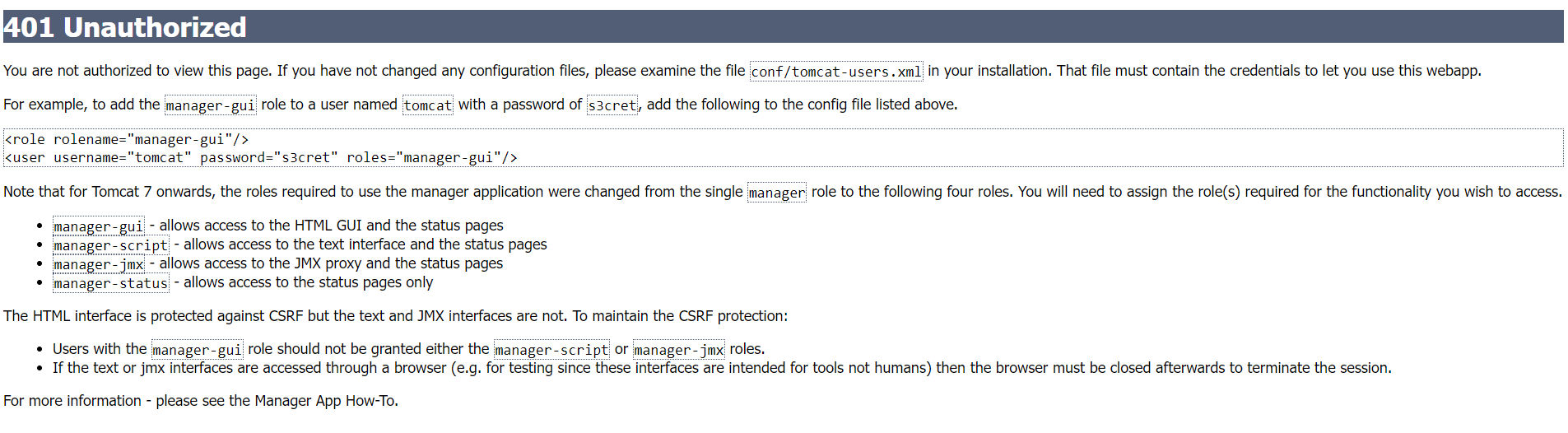
nano /opt/tomcat/webapps/examples/META-INF/context.xml

nano /opt/tomcat/webapps/host-manager/META-INF/context.xml

nano /opt/tomcat/webapps/manager/META-INF/context.xml

#do the comment below section in the above three files

* 



cd /opt/tomcat/conf

cat tomcat-users.xml

#Add below lines between <tomcat-users> tag

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<role rolename="manager-jmx"/>

<role rolename="manager-status"/>

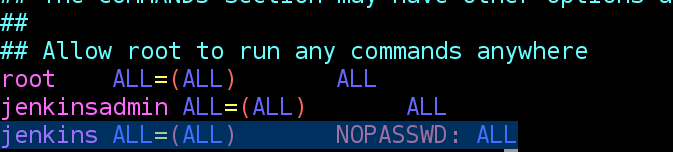
<user username="admin" password="admin" roles="manager-gui,manager-script,manager-jmx,manager-status"/>

<user username="tomcat" password="pass123" roles="manager-gui"/>

1. Jenkins machine : run visudo command

$visudo 🡪 esc + i => esc :wq!

1. If we don’t know this where to find, pls refer **Jenkins note.txt** file in github



1. Change the tomcat root user to ec2-user using below command
2. On tomcat Machine:

chown -R ec2-user:ec2-user /opt/tomcat/webapps

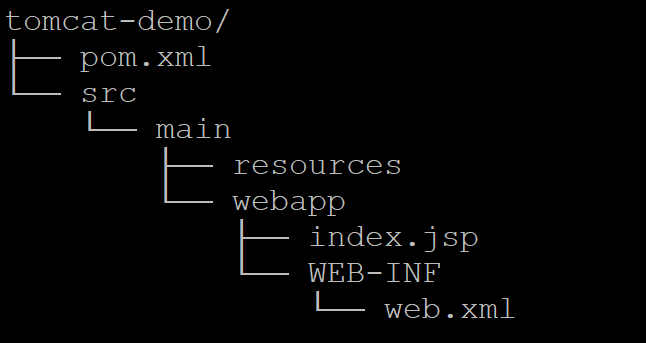
**Create Maven Web Project:**

Go to below official website of maven and follow the steps If required

<https://maven.apache.org/archetypes/maven-archetype-webapp/>

1. mvn archetype:generate -DgroupId=com.sample.webproject -DartifactId=tomcat-demo -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false

run tree command it will show the tree format for better visibility.



Run tree command



Now let’s create Jenkins job for maven web project integration and deploy the war file into tomcat server.

**Step1**: Now create an Jenkins job name: tomcat-web-app

Dashboard 🡪New Item -🡪 Name : tomcat-web-app

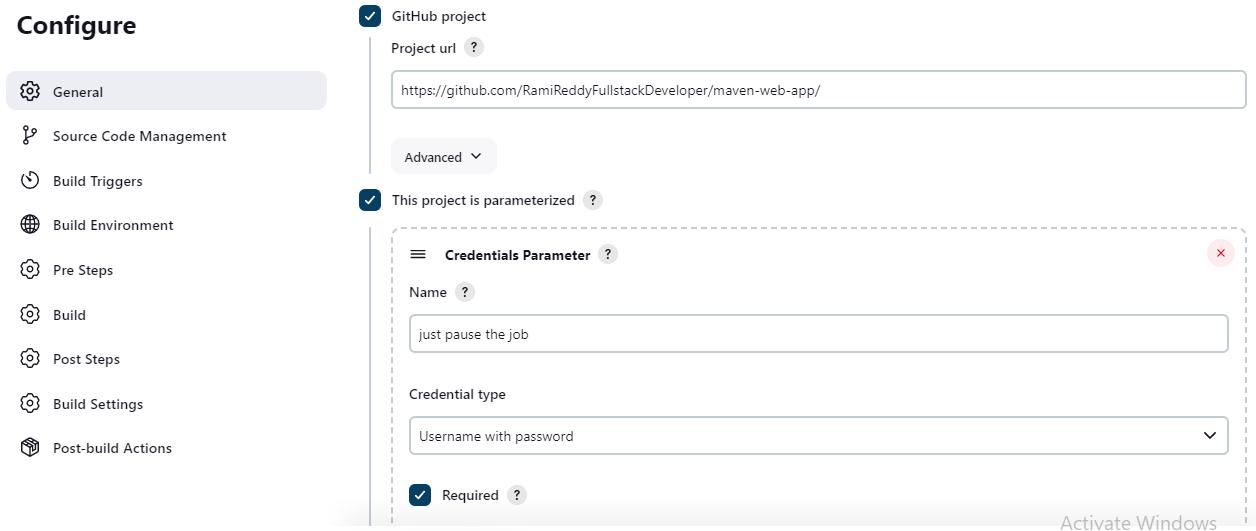
Select : Maven

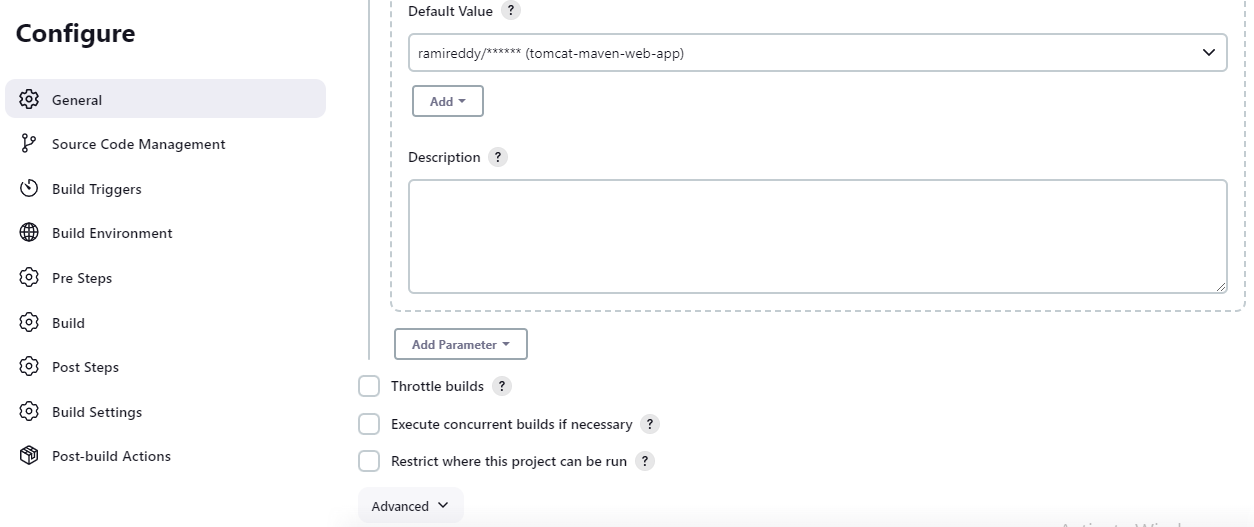
Click **OK**

**Step2**: Go back to dashboard 🡪 click tomcat-web-app job name 🡪configure 🡪

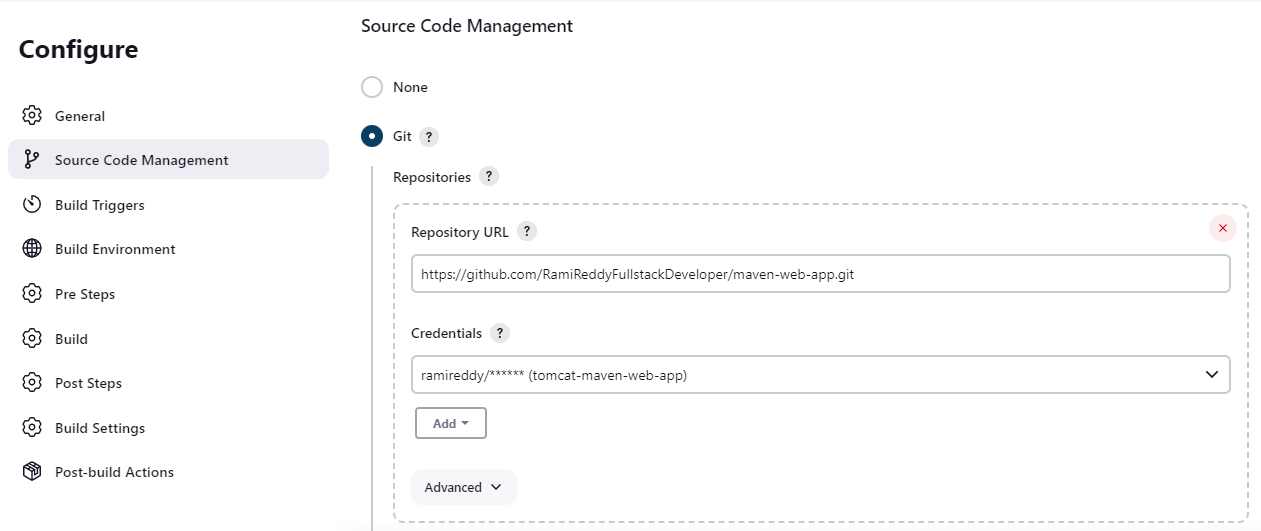
Description : Any

**Step3**: Check Github project

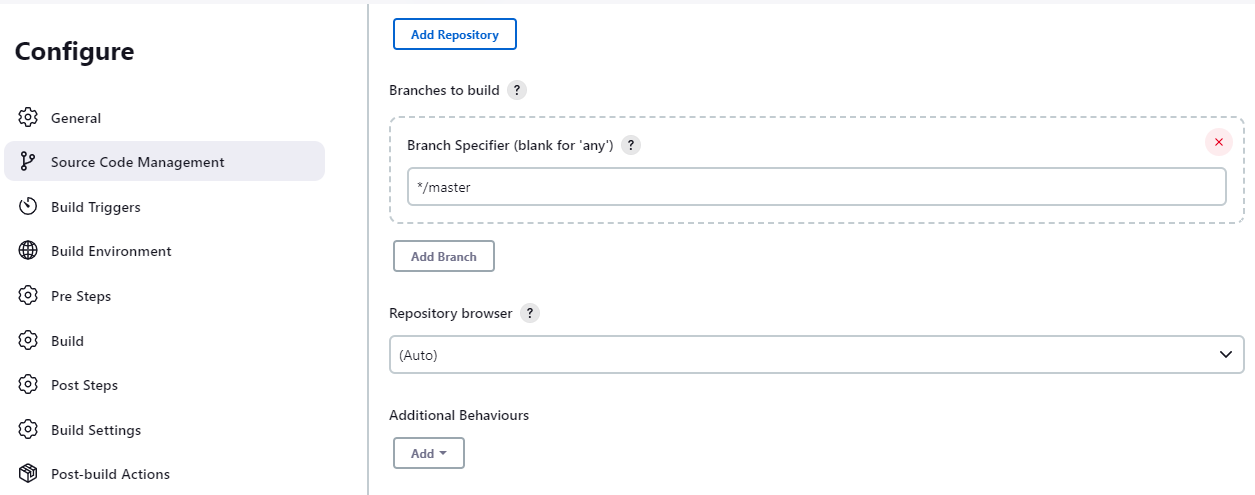




**Step4**: Source code management Section under configure only



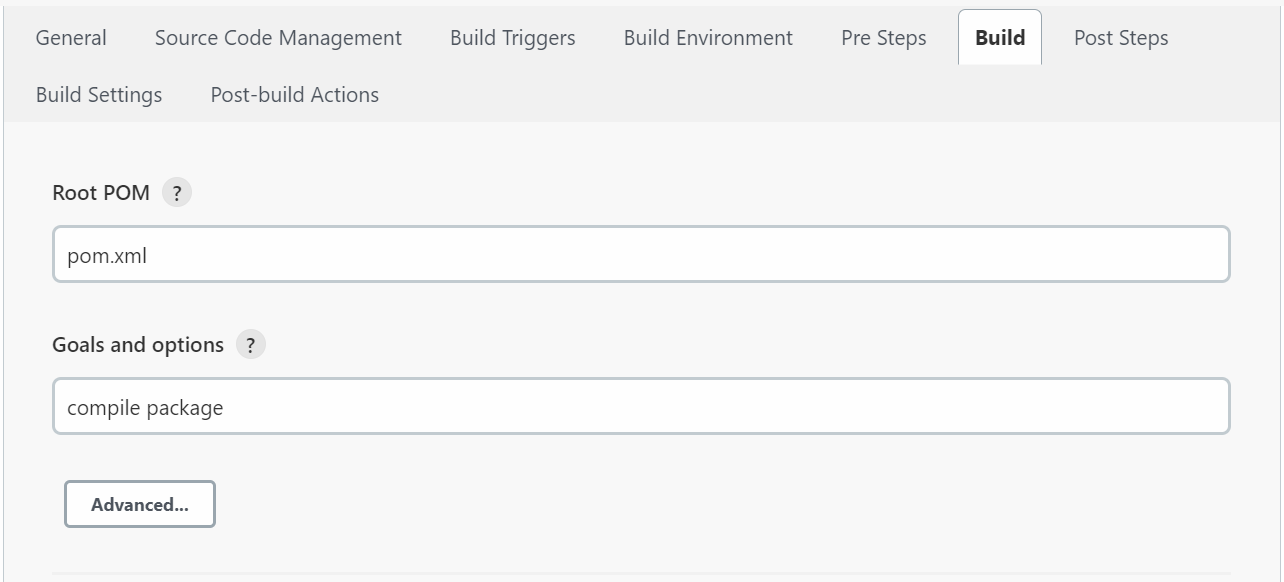
Now here specify which branch we want to pick from git and build purpose.



**Note**: Next Build Trigger and Build Environment section for now leave it as is.

We will talk about that later about use cases. Now lets come to **Build** section

**Build Section:**



**Note**: Click Apply and Save and Click on the **Build** Now, then it will start the compile the project and packaging.

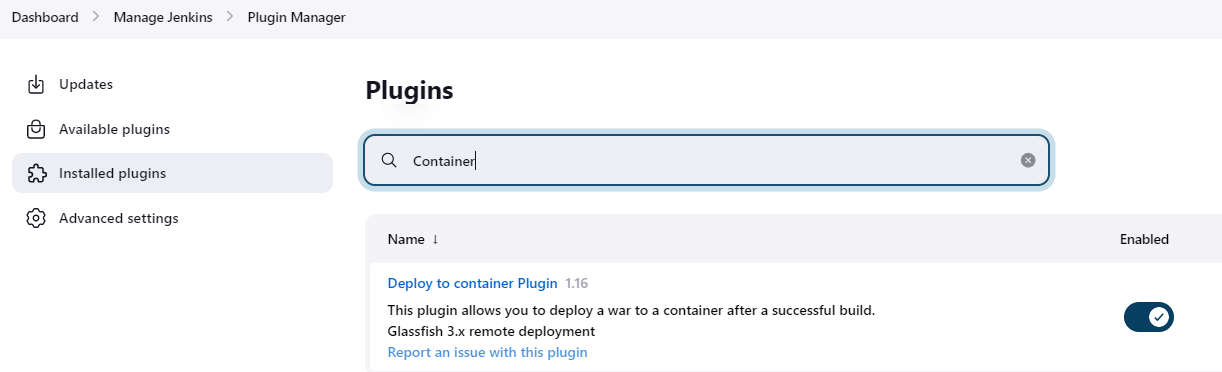


Note: So far we have just compile the maven web-app project and packaged that.

Now let’s deploy the war file into tomcat server for this edit the same Jenkins job and do the below things.

**Deploy War file to tomcat server:**

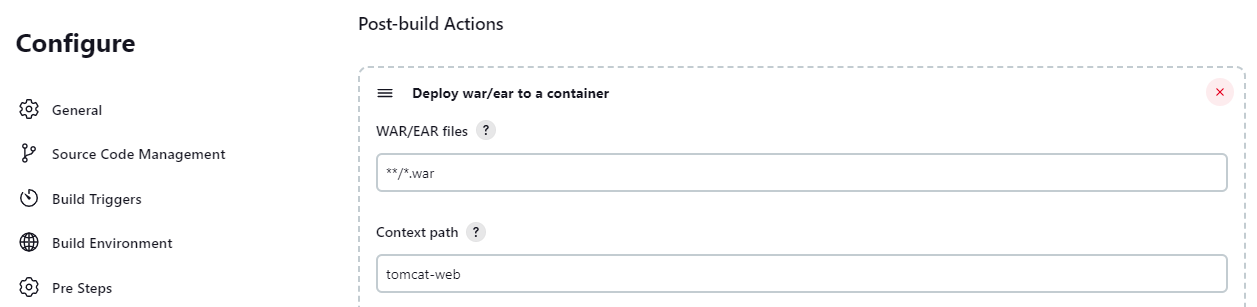
**Note**: login into Jenkins 🡪 Dashboard 🡪 Manage Jenkins -🡪 Plugin Manager 🡪 search for “Deploy to Container Pluggin” and check check box and click install.



**Note:** then only we can see Deploy war/ear to container under **pos-build Actions**.

**Step5**: Post Build Action Section

Choose Jenkins job name and click configure.



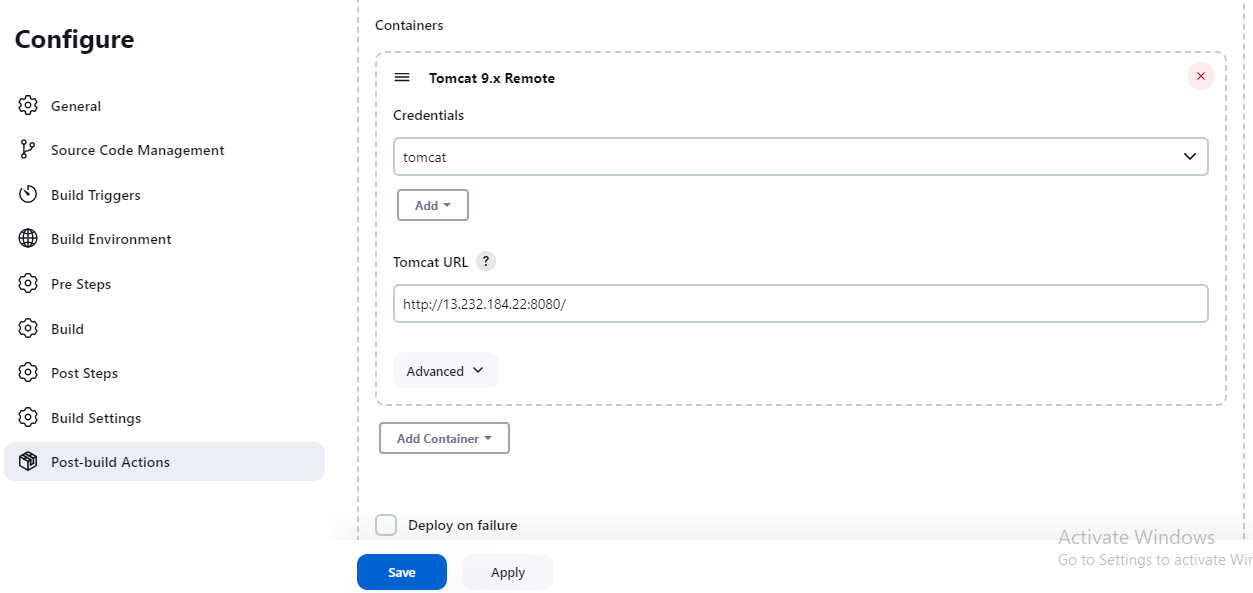
**Step6**: Now add containers and Add credentials for tomcat server.

**Note**: Create credentials for tomcat **after adding containers only**



**Note**: Select tomcat credential from credentials drop down.

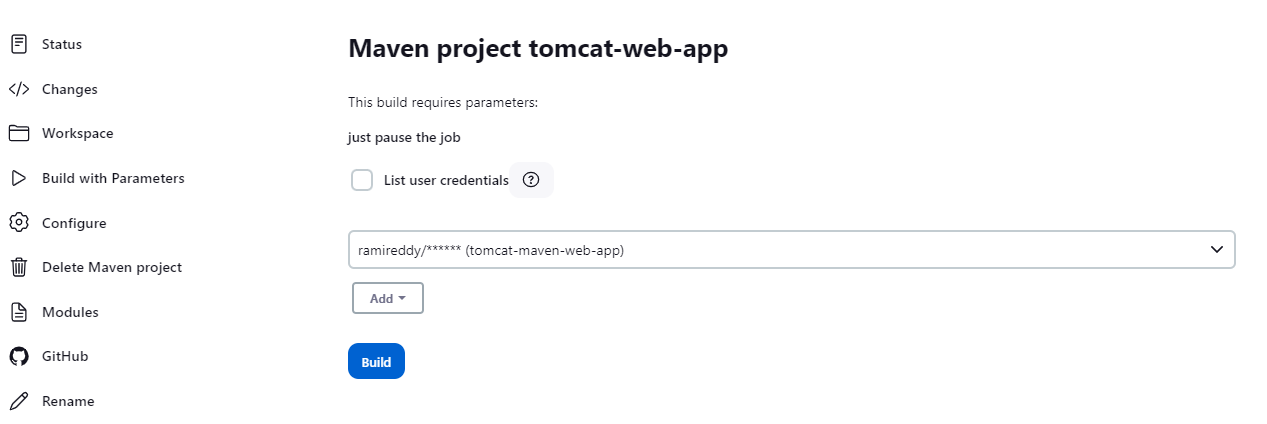
**Note**: Make sure tomcat server up and running first and also make sure 8080 port is enabled in ec2 **Security Group in-bound rule** and provide the tomcat URL.



**Step7**: Now click apply and Save button

**Step8**: Now go back to Dashboard 🡪 click on the tomcat-web-job Jenkins job name 🡪

Build with parameter.



**Step8**: Click build it will trigger the build

**Step9**: to see the console output go to the below build history section-🡪 click build number 🡪 console output.

**Step10**: now we can verify whether we receive an email or not by visiting the below url. https://www.mailinator.com/v4/public/inboxes.jsp?msgid=ramj-1681654501-114593268&to=ramj

**Step11**: Done.

**Step12**: we can check the war file where war will be deployed

Cd /opt/tomcat/webapps

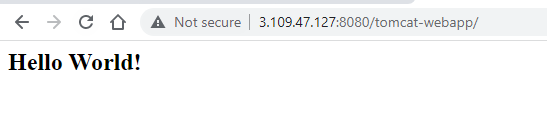
**Note**: Once build is successful



Now go to browser and paste the below url and you will see the output.

<http://13.232.184.22:8080/tomcat-web/>

**Output**: Hello world //this is coming from our maven web app project index.jsp.



**Note**: finally verify the **ls /opt/tomcat/webapps** folder contain the above **tomcat-demo.war** file.

[root@tomcat webapps]# ls /opt/tomcat/webapps

ROOT docs examples host-manager manager tomcat-demo tomcat-demo.war

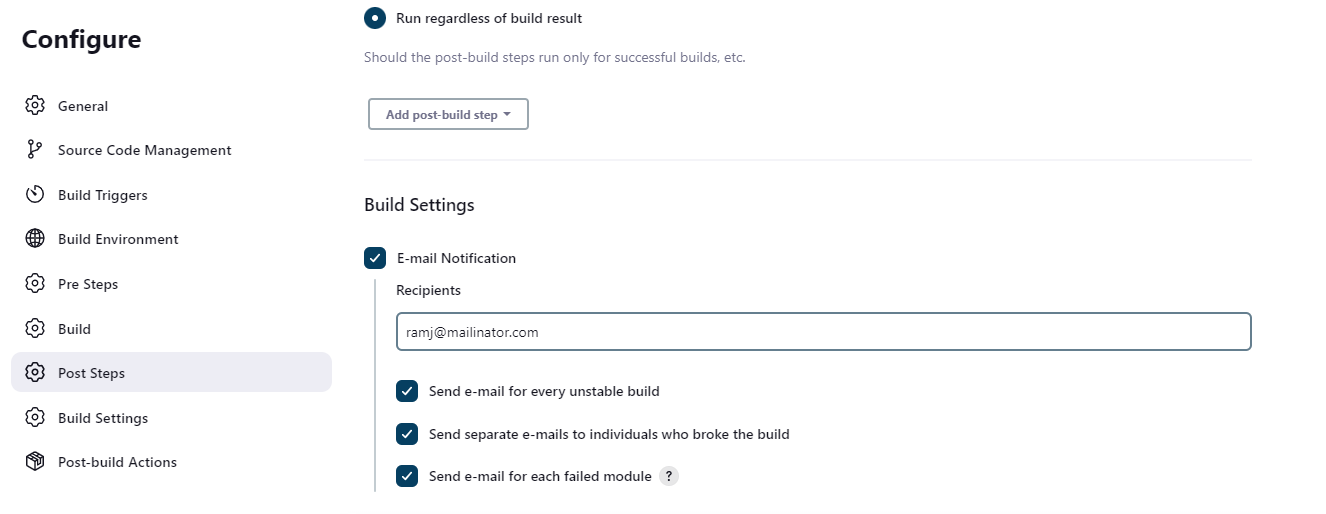
**Send Email:**

**Step6**: Build settings section.

Note: Go to ec2-server 🡪 Security group 🡪 inbound rule section 🡪 and enable

port number 25 for sending email, to check whether email came or not for this go to

https://www.mailinator.com/v4/public/inboxes.jsp?msgid=ramj-1681654501-114593268&to=ramj



https://github.com/RamiReddyFullstackDeveloper/maven-web-app

**Pipeline Job:**

https://github.com/vikas99341/tomcat-demo.git