THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY

CONTINUOUS DELIVERY AND DEVOPS (LAB)

SUBJECT CODE- PCS217P

Submitted To:

Dr. Gurpal Singh Chhabra

(Assistant Professor)

Department of Computer Science and Engineering

Submitted by:

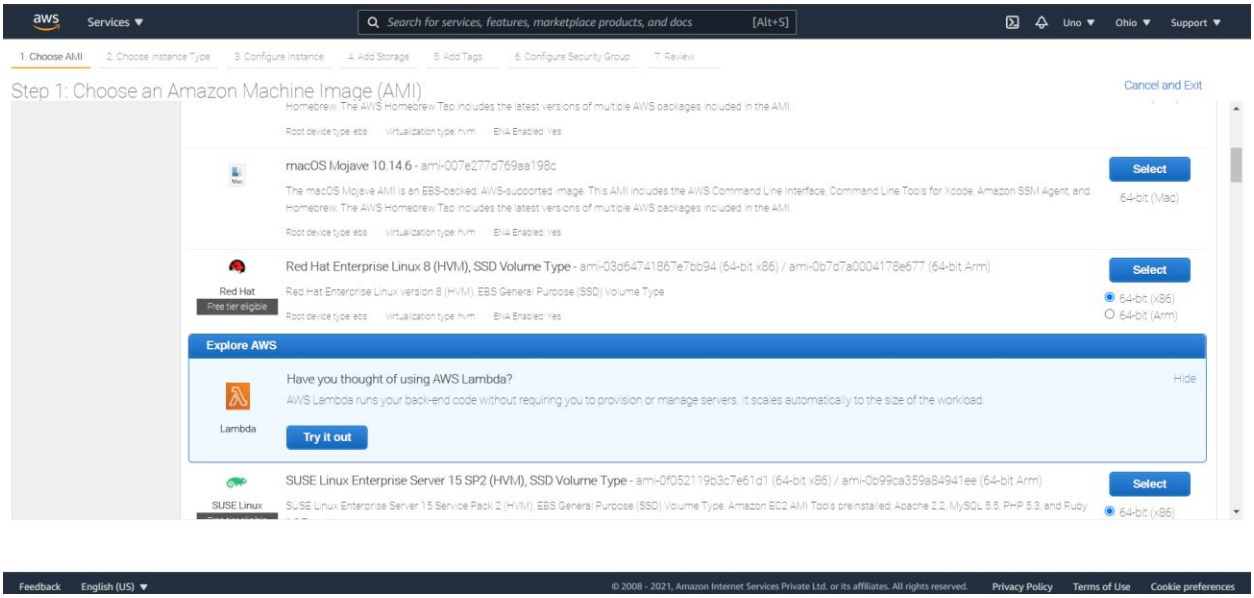
Ujjwal Ahuja

(802032055)

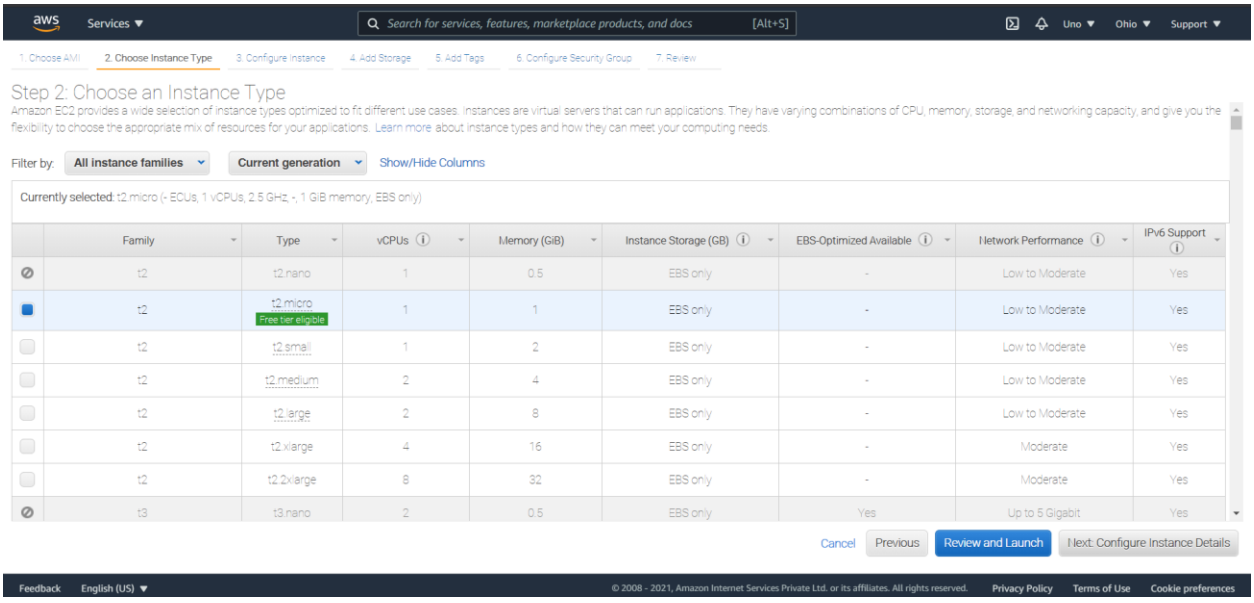
Project on Continuous Integration and Continuous Development through Jenkins

1. Create a free tier Amazon AWS account (Debit/credit card needed for a minimal transaction of Rs.2

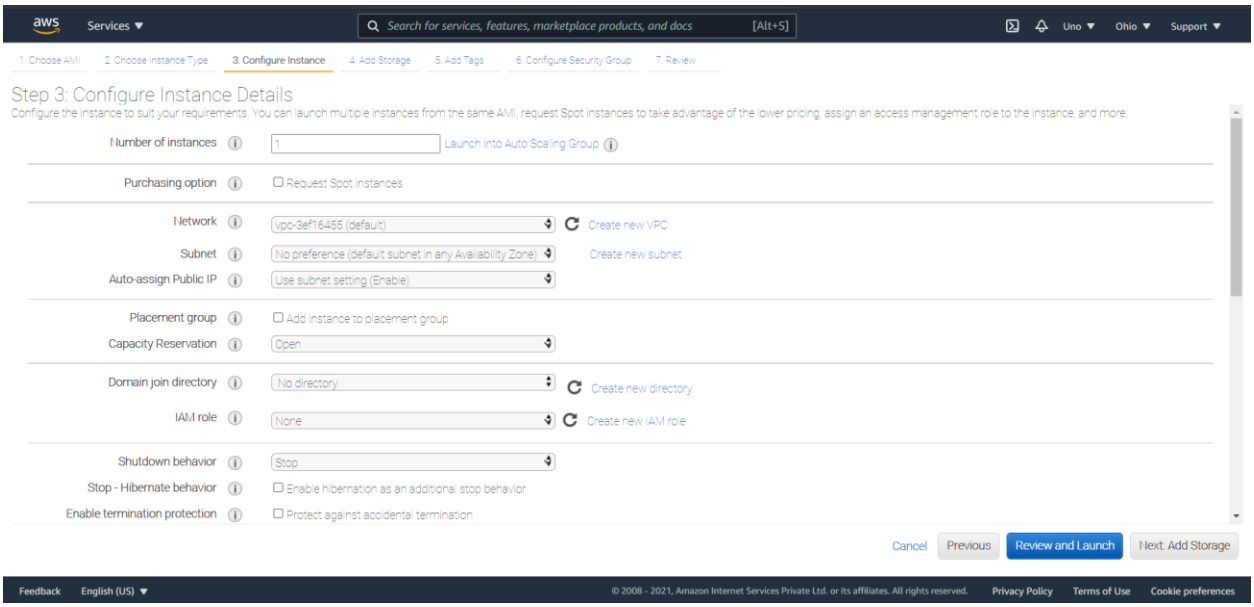
2. Create an Amazon ec2 instance



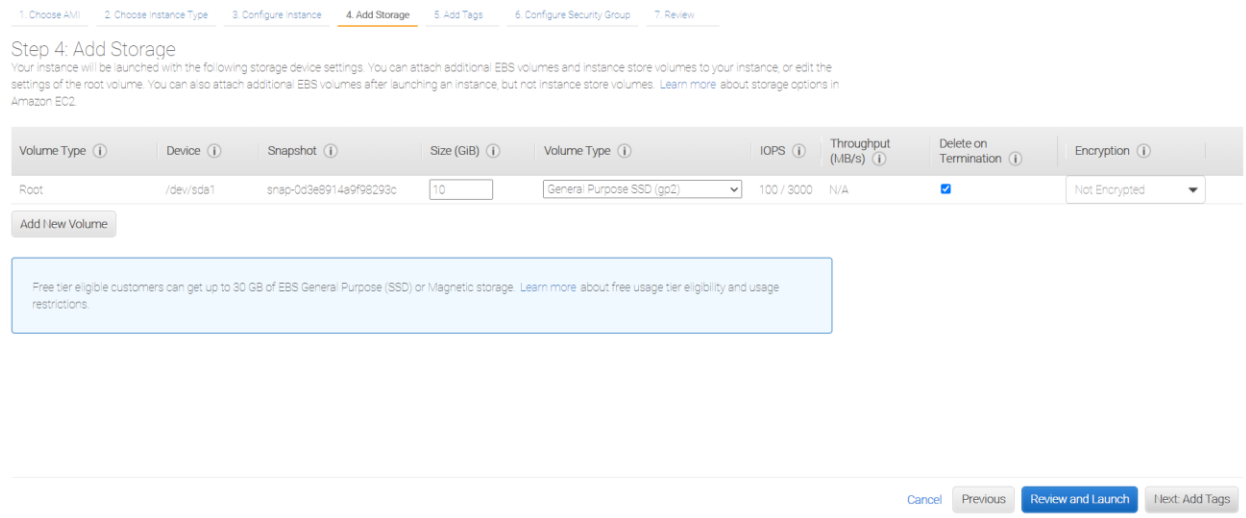
3. Choose your Instance type



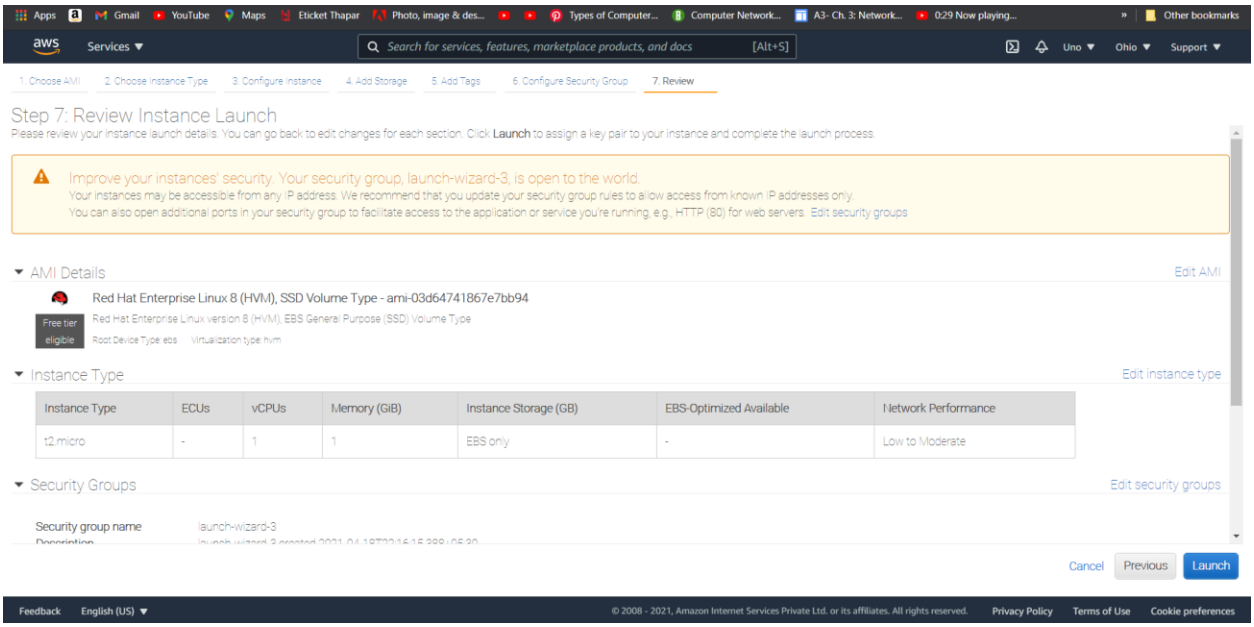
4. Configure Instance Details



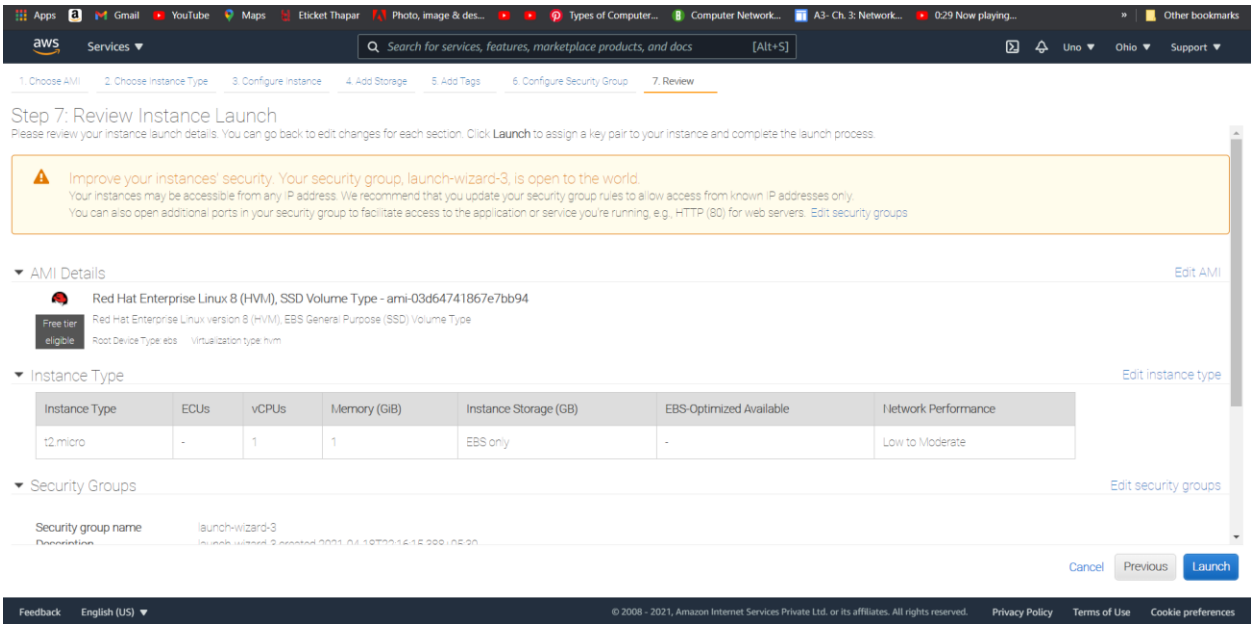
5. Add Storage



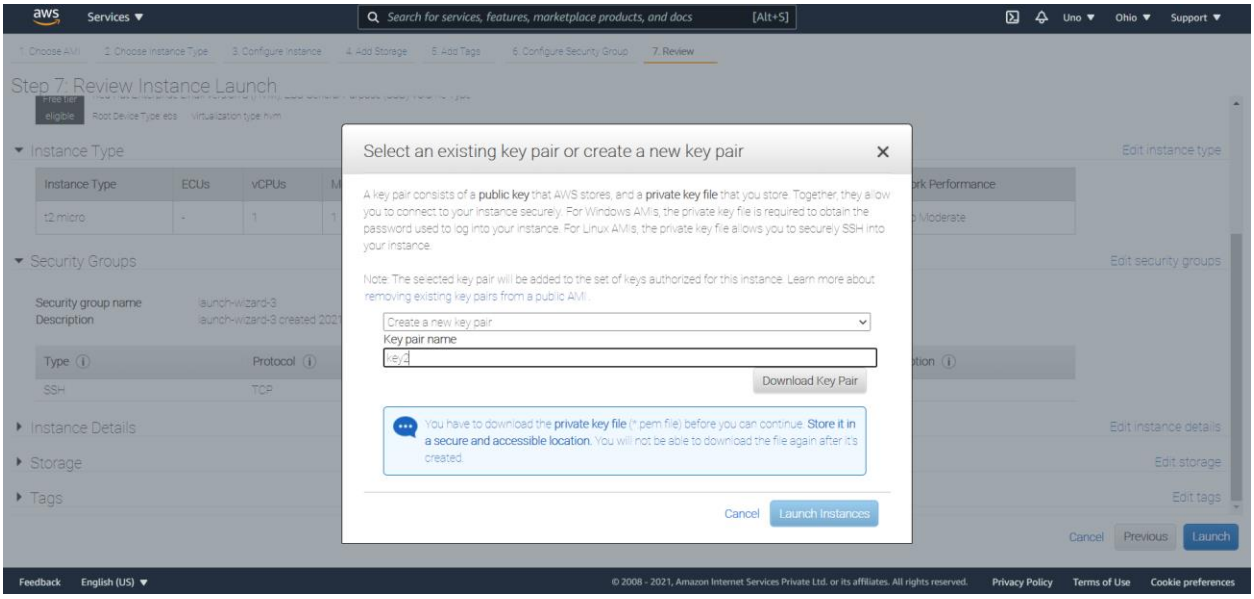
6. Configure Security Group



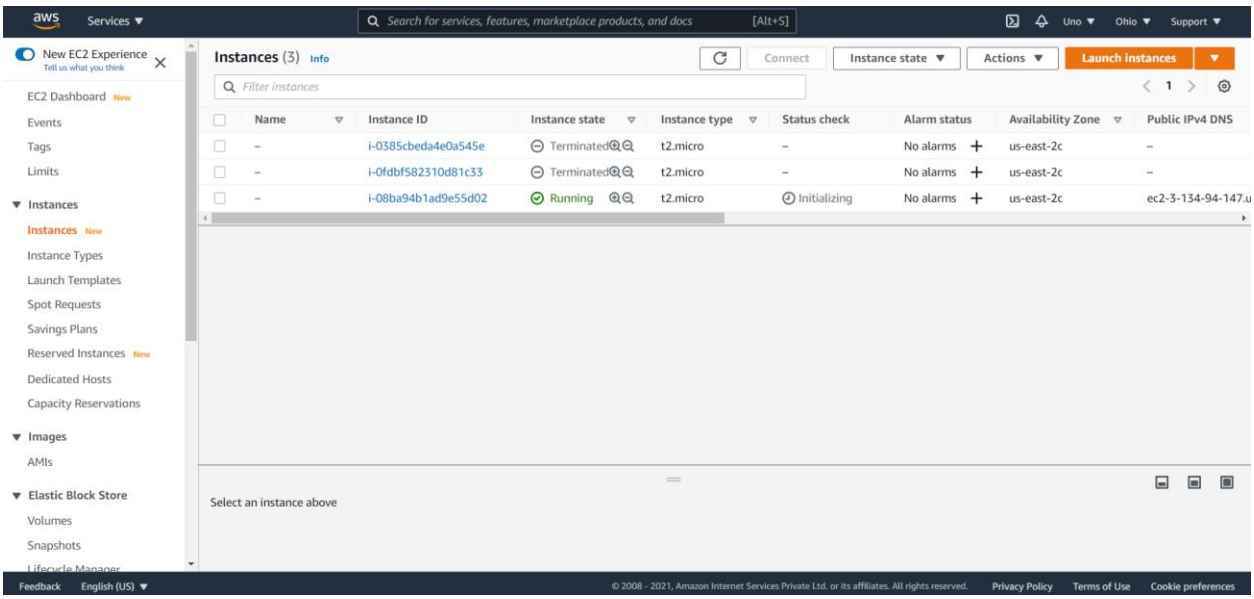
7. Review Instance Launch



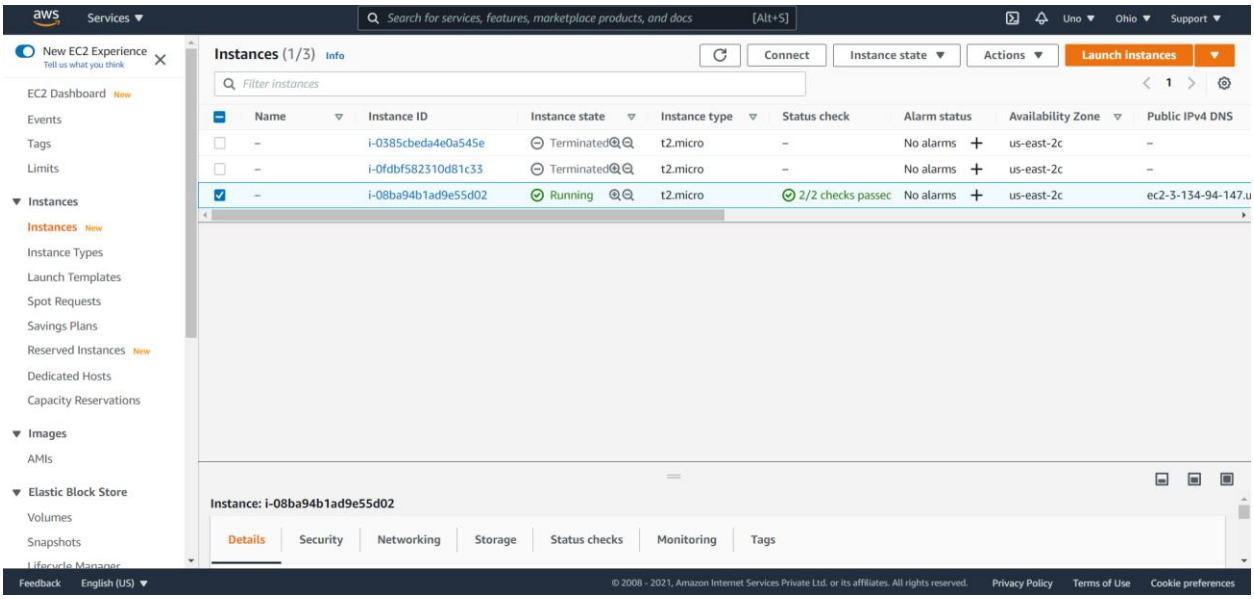
8. After clicking on launch, create a new key pair for your machine



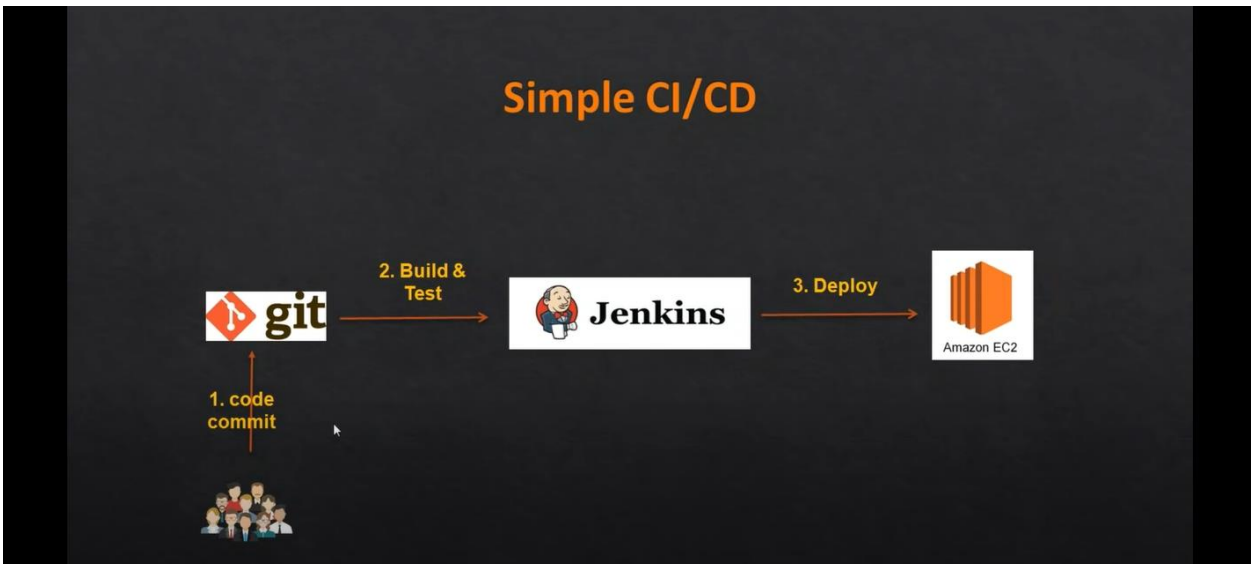
9. Now check for instances in EC2, it may take 1-2 minute for instance to initialize.



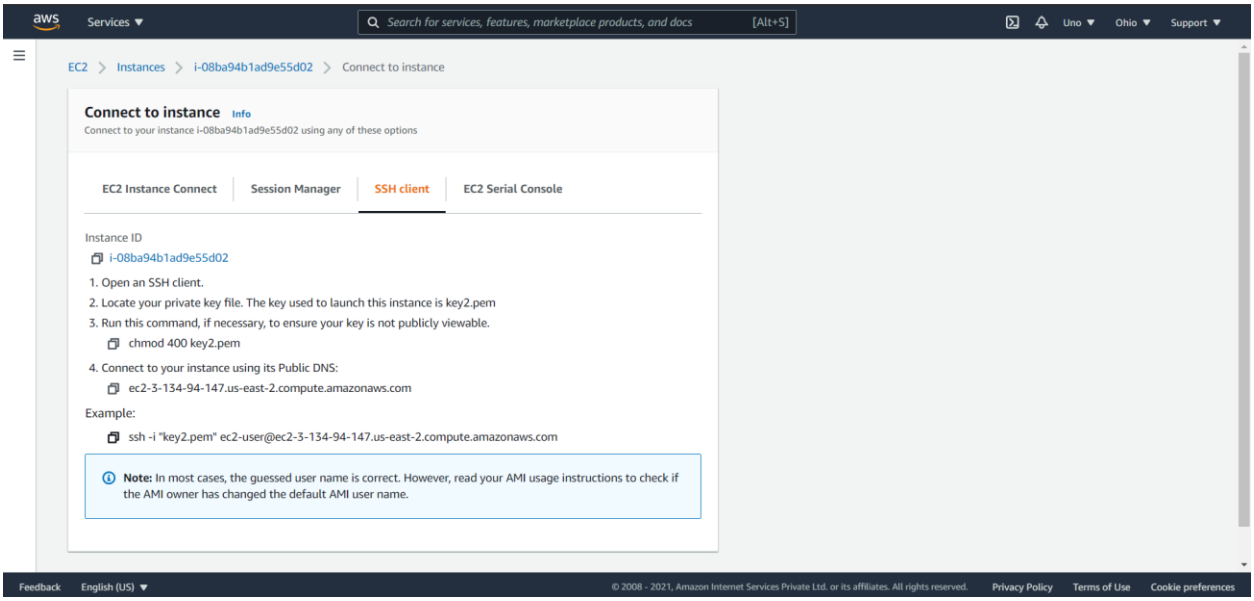
9.1.



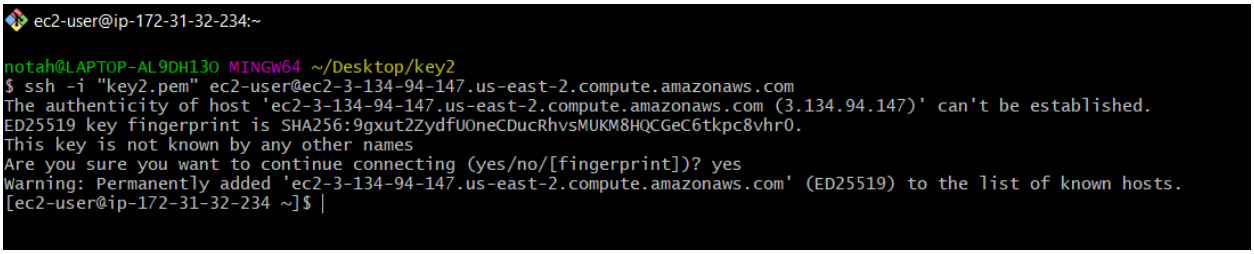
10. Connect to your AWS EC2 instance using git Bash



11. Connect to AWS instance



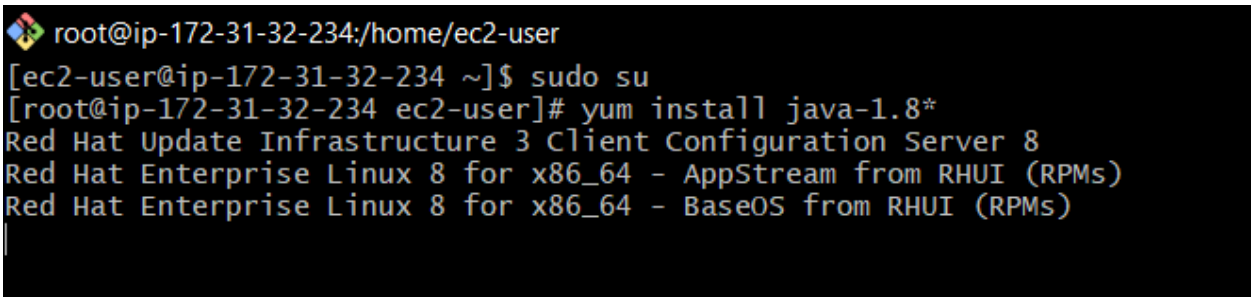
11.1.



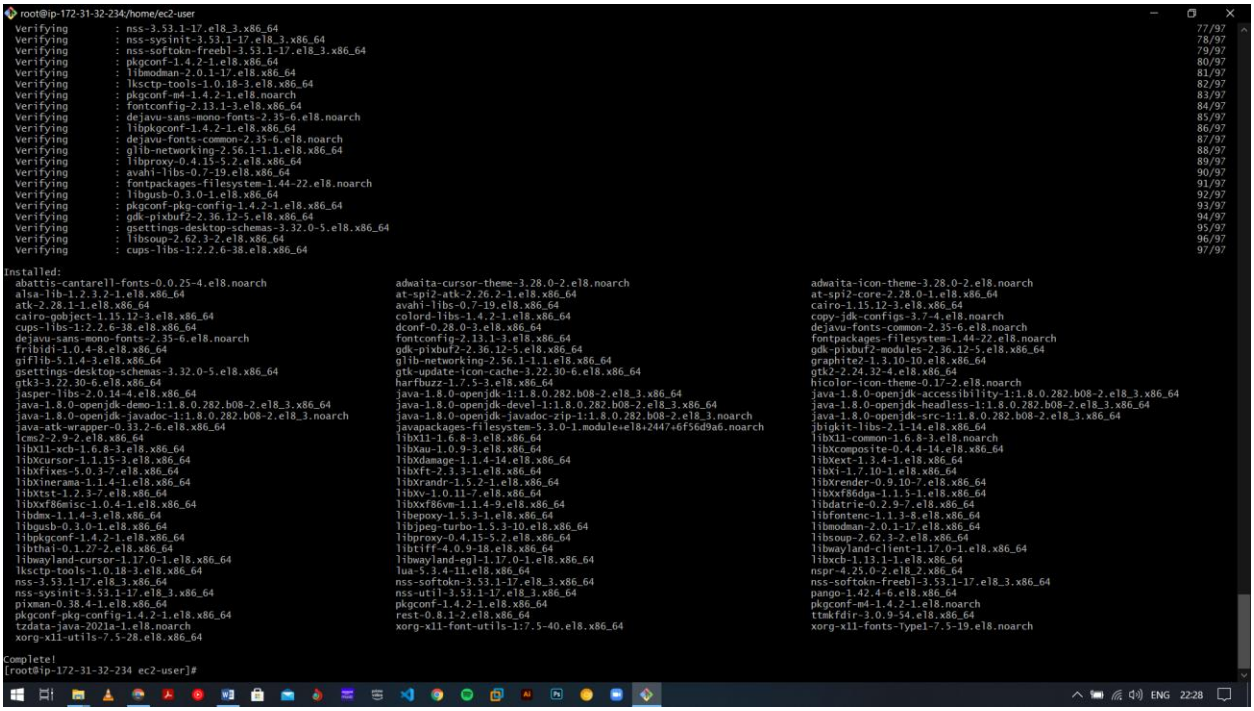
12. We need to deploy war file so we need tomcat server

13. Steps to configure EC2 instance on tomcat server:

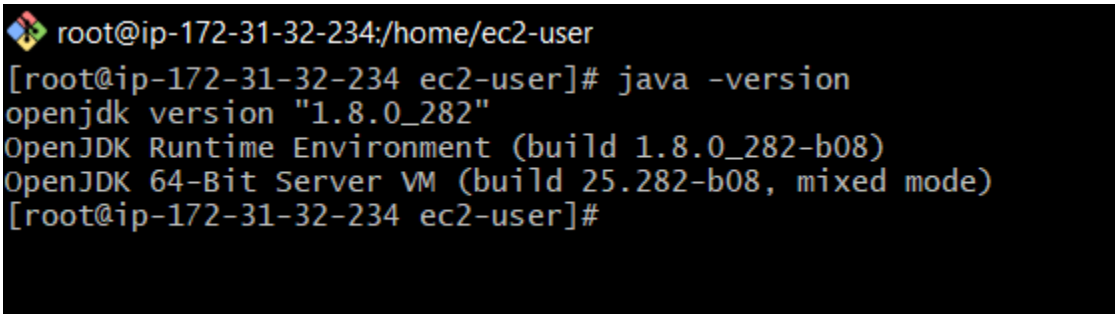
14. First we need to install java



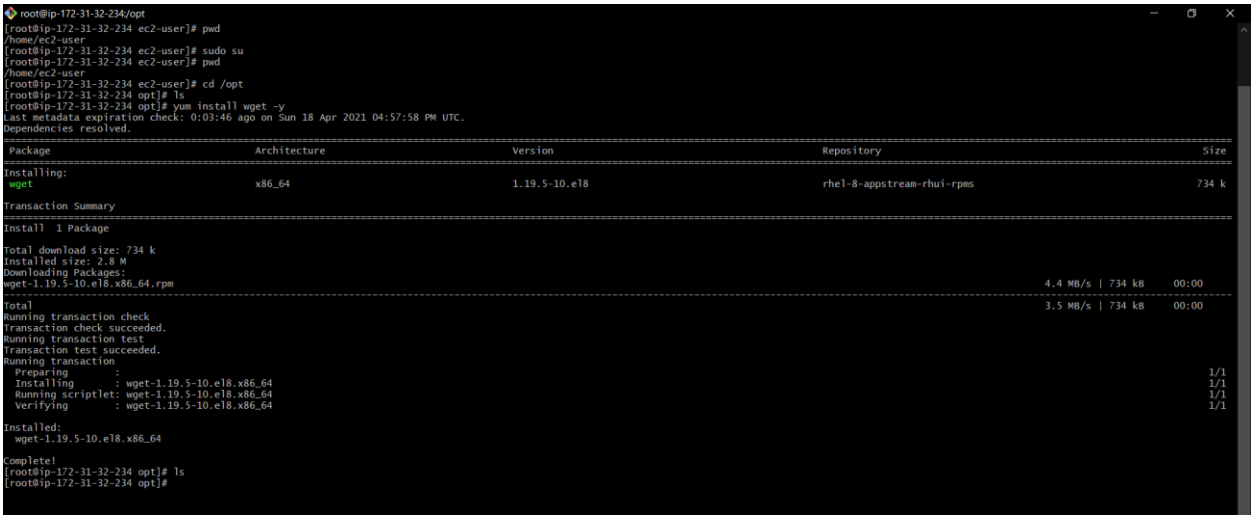
14.1.



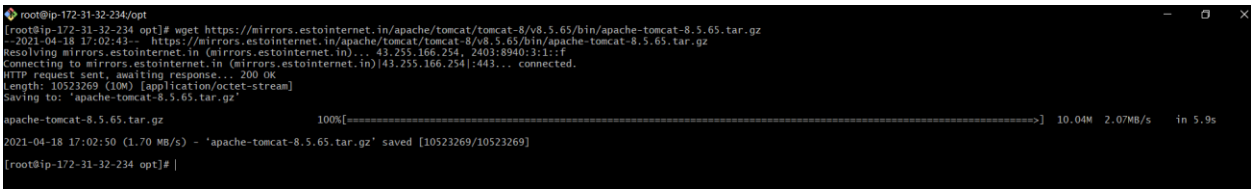
14.2. Check Version



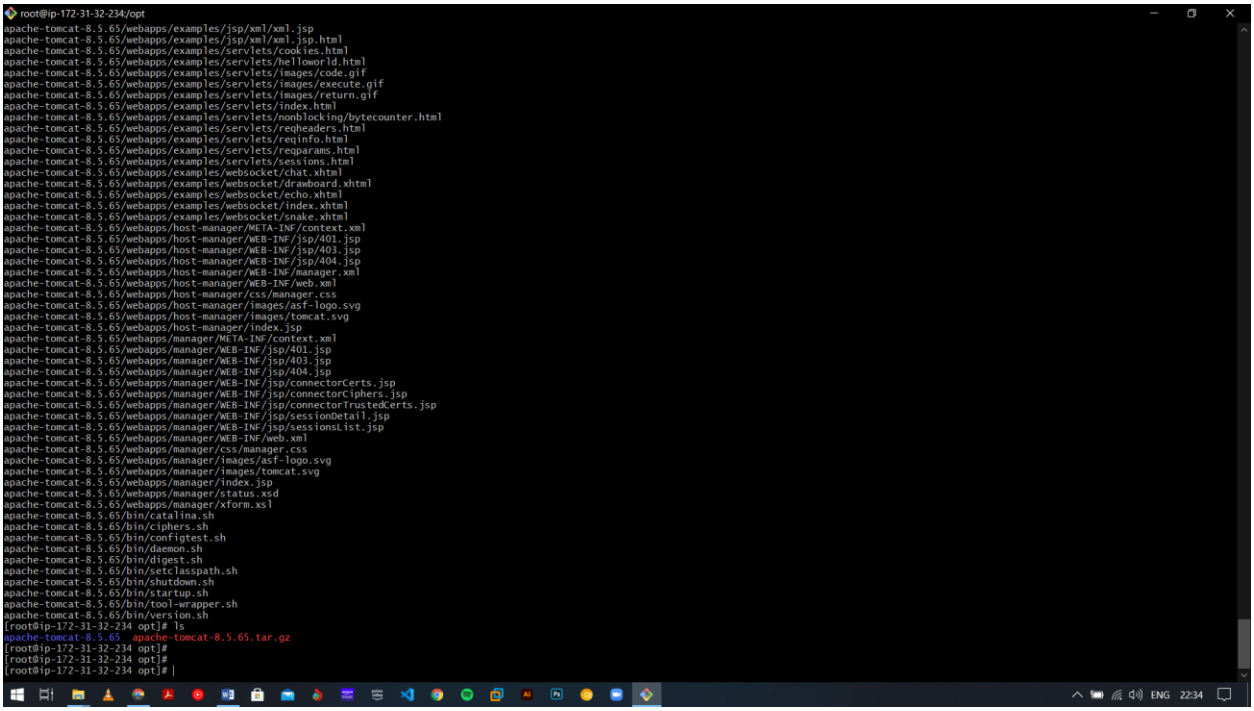
15. wget install command : yum install wget-y



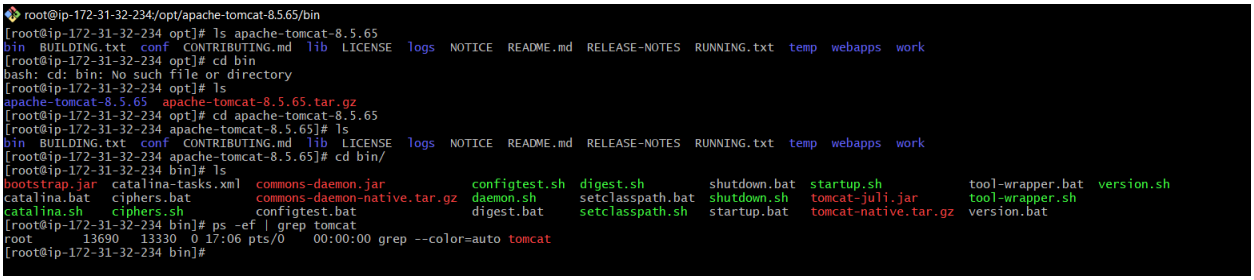
16. Download tomcat
    First change the path
    See on which path you are using ‘pwd’ command. (It will probably be /root)
    Change the path to ‘cd /opt’



16.1. Then untar and unzip using tar command “tar –zvxf apache-tomcat-8.5.64.tar.gz”



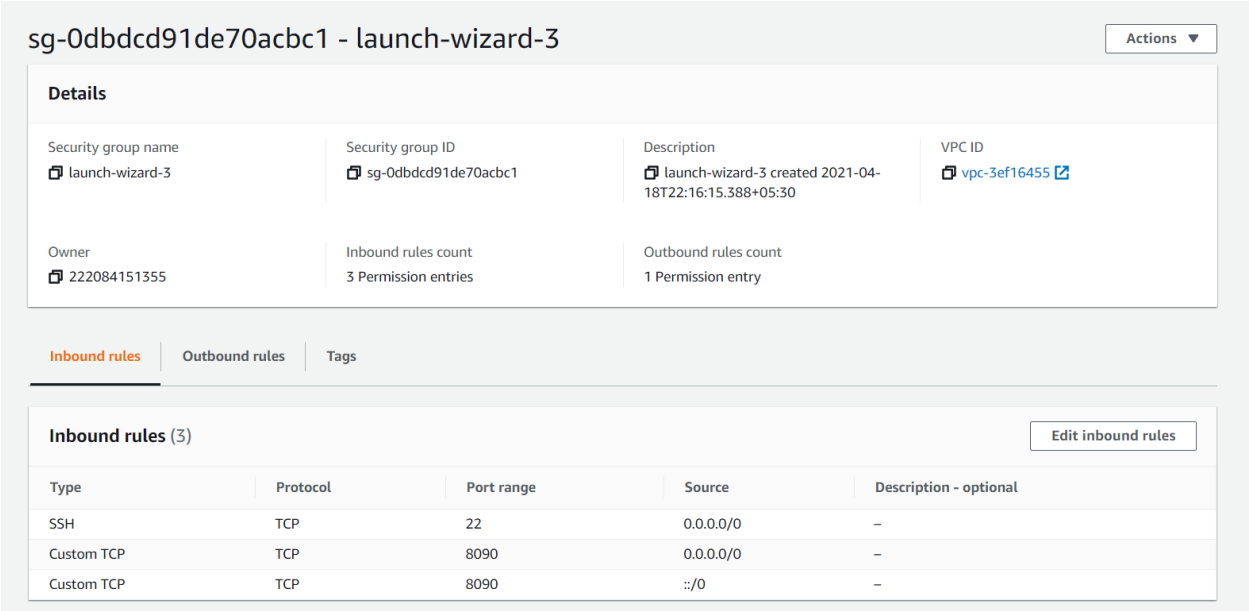
16.2. To check if tomcat is running or not (not)



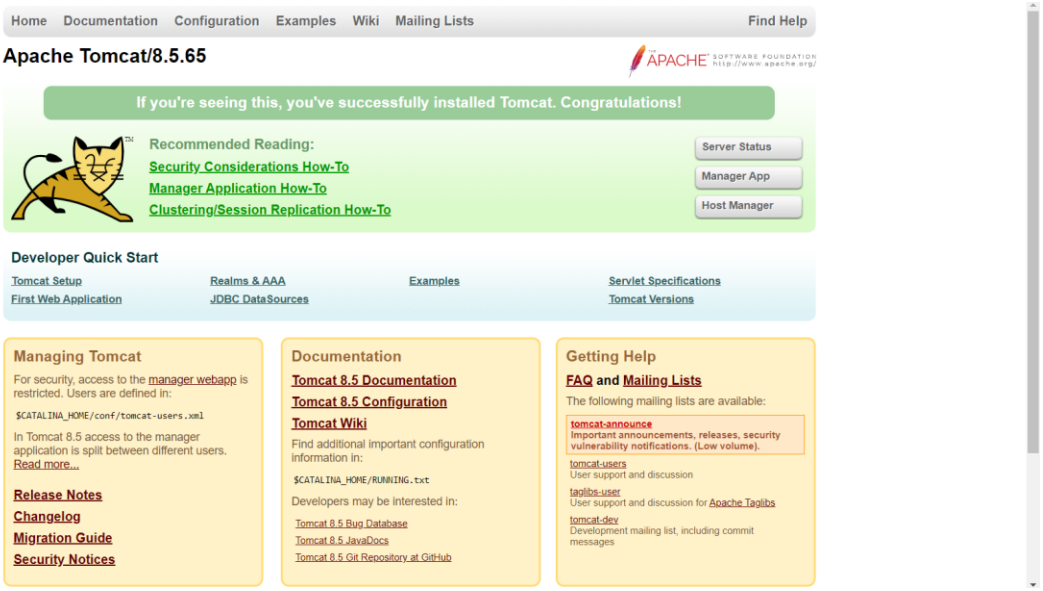
17. To access default tomcat page first access services under bin directory here is startup.sh and shutdown.sh file:
    17.1. Give a permission to startup.sh and shutdown.sh:

```
root@ip-172-31-38-55:/opt/apache-tomcat-10.0.5/bin
[root@ip-172-31-38-55 apache-tomcat-10.0.5]# ls
bin  BUILDING.txt  conf  CONTRIBUTING.md  lib  LICENSE  logs  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  temp  webapps  work
[root@ip-172-31-38-55 apache-tomcat-10.0.5]# cd bin
[root@ip-172-31-38-55 bin]# ps -ef | grep tomcat
root       13904   13882  0 13:00 pts/0    00:00:00 grep --color=auto tomcat
[root@ip-172-31-38-55 bin]# ls -ltr
total 880
-rwxr-x---. 1 root root   1908 Mar 30 08:19 version.sh
-rw-r-----. 1 root root   2026 Mar 30 08:19 version.bat
-rwxr-x---. 1 root root   5540 Mar 30 08:19 tool-wrapper.sh
-rw-r-----. 1 root root   4574 Mar 30 08:19 tool-wrapper.bat
-rw-r-----. 1 root root 423606 Mar 30 08:19 tomcat-native.tar.gz
-rw-r-----. 1 root root  46593 Mar 30 08:19 tomcat-juli.jar
-rwxr-x---. 1 root root   1904 Mar 30 08:19 startup.sh
-rw-r-----. 1 root root   2022 Mar 30 08:19 startup.bat
-rwxr-x---. 1 root root   1902 Mar 30 08:19 shutdown.sh
-rw-r-----. 1 root root   2020 Mar 30 08:19 shutdown.bat
-rwxr-x---. 1 root root   3708 Mar 30 08:19 setclasspath.sh
-rw-r-----. 1 root root   3460 Mar 30 08:19 setclasspath.bat
-rwxr-x---. 1 root root   1970 Mar 30 08:19 migrate.sh
-rw-r-----. 1 root root   2096 Mar 30 08:19 migrate.bat
-rwxr-x---. 1 root root   3382 Mar 30 08:19 makebase.sh
-rw-r-----. 1 root root   3606 Mar 30 08:19 makebase.bat
-rwxr-x---. 1 root root   1965 Mar 30 08:19 digest.sh
-rw-r-----. 1 root root   2091 Mar 30 08:19 digest.bat
-rwxr-x---. 1 root root   8836 Mar 30 08:19 daemon.sh
-rwxr-x---. 1 root root   1922 Mar 30 08:19 configtest.sh
-rw-r-----. 1 root root   2040 Mar 30 08:19 configtest.bat
-rw-r-----. 1 root root 207420 Mar 30 08:19 commons-daemon-native.tar.gz
-rw-r-----. 1 root root  25357 Mar 30 08:19 commons-daemon.jar
-rwxr-x---. 1 root root   1997 Mar 30 08:19 ciphers.sh
-rw-r-----. 1 root root   2123 Mar 30 08:19 ciphers.bat
-rw-r-----. 1 root root   1664 Mar 30 08:19 catalina-tasks.xml
-rwxr-x---. 1 root root  24814 Mar 30 08:19 catalina.sh
-rw-r-----. 1 root root  16266 Mar 30 08:19 catalina.bat
-rw-r-----. 1 root root  34727 Mar 30 08:19 bootstrap.jar
[root@ip-172-31-38-55 bin]# chmod +x startup.sh
[root@ip-172-31-38-55 bin]# chmod +x shutdown.sh
[root@ip-172-31-38-55 bin]#
```

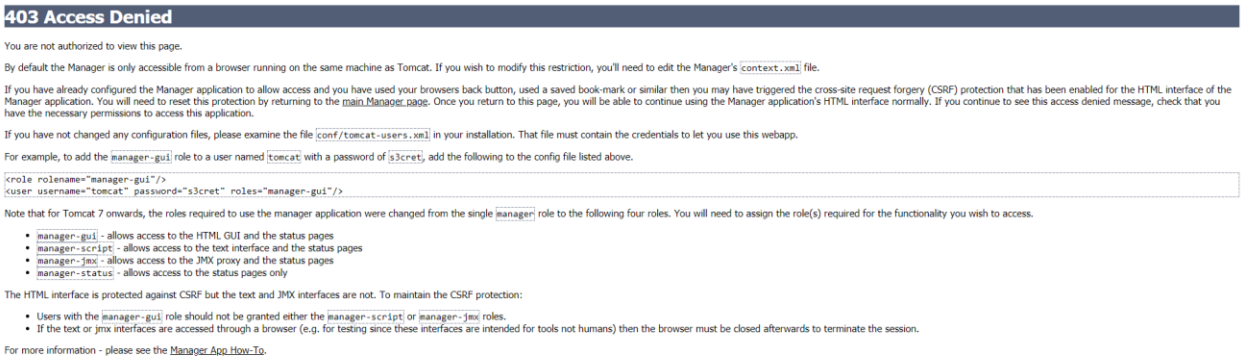
18. By default it will work on port no. 22 so in security group we need to add port 8090:



19. To access tomcat in a browser, copy the public ip of your aws instance and add ‘:8090’ at the end.
    For example: http://3.21.162.39:8090/



20. If we want to access these manager file then it will access from the same machine as tomcat but we access it from other so we need to edit the context.xml file



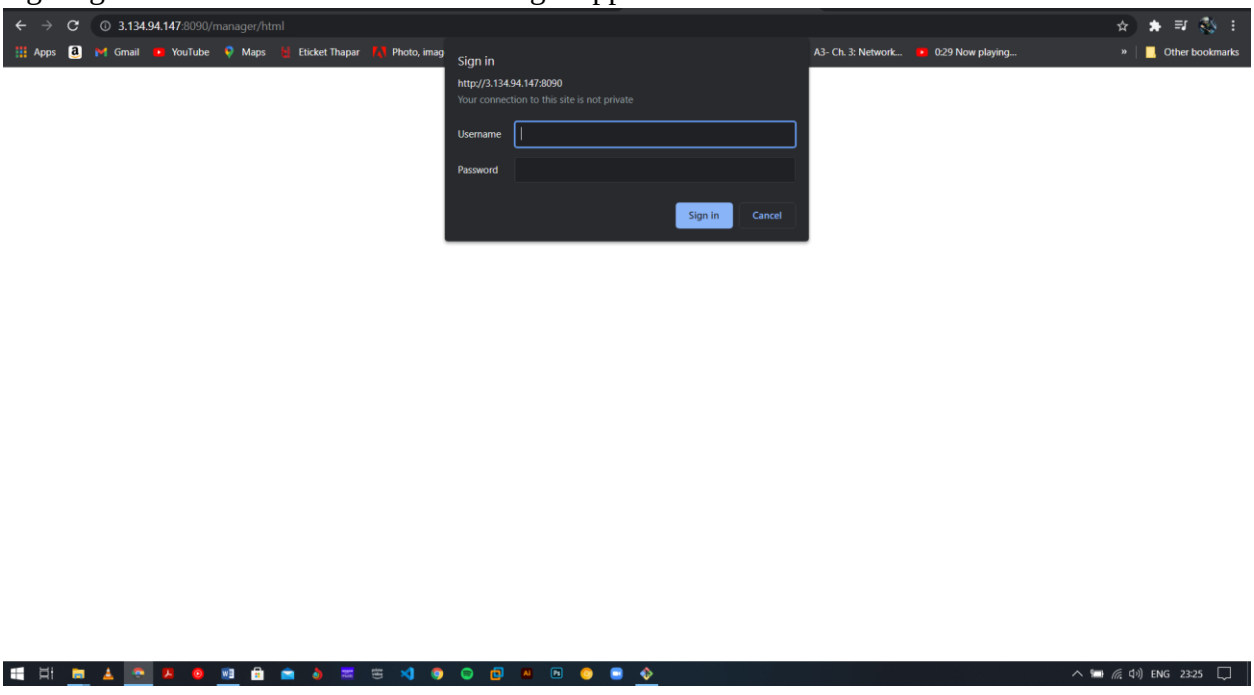
21. Search context.xml file via find command if will give the location

    Command: find / -name context.xml

    21.1. Edit the context.xml file using vi command

```
[root@ip-172-31-32-234 conf]# find / -name context.xml
/opt/apache-tomcat-8.5.65/conf/context.xml
/opt/apache-tomcat-8.5.65/webapps/examples/META-INF/context.xml
/opt/apache-tomcat-8.5.65/webapps/host-manager/META-INF/context.xml
/opt/apache-tomcat-8.5.65/webapps/manager/META-INF/context.xml
[root@ip-172-31-32-234 conf]# vi /opt/apache-tomcat-8.5.65/webapps/host-manager/META-INF/context.xml
[root@ip-172-31-32-234 conf]# vi /opt/apache-tomcat-8.5.65/webapps/manager/META-INF/context.xml
[root@ip-172-31-32-234 conf]# tomcatdown
Using CATALINA_BASE:   /opt/apache-tomcat-8.5.65
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.65
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.65/temp
Using JRE_HOME:        /
Using CLASSPATH:       /opt/apache-tomcat-8.5.65/bin/bootstrap.jar:/opt/apache-tomcat-8.5.65/bin/tomcat-juli.jar
Using CATALINA_OPTS:
[root@ip-172-31-32-234 conf]# womcatup
bash: womcatup: command not found
[root@ip-172-31-32-234 conf]# tomcatup
Using CATALINA_BASE:   /opt/apache-tomcat-8.5.65
Using CATALINA_HOME:   /opt/apache-tomcat-8.5.65
Using CATALINA_TMPDIR: /opt/apache-tomcat-8.5.65/temp
Using JRE_HOME:        /
Using CLASSPATH:       /opt/apache-tomcat-8.5.65/bin/bootstrap.jar:/opt/apache-tomcat-8.5.65/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
[root@ip-172-31-32-234 conf]#
```

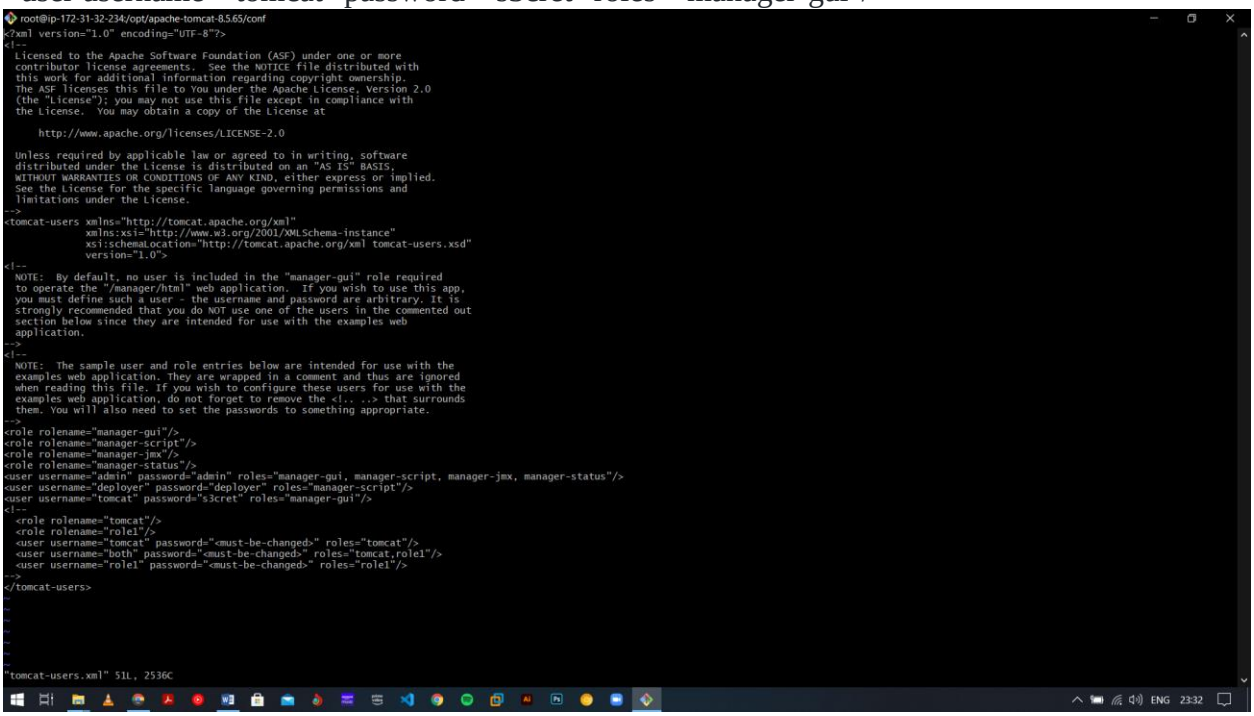
22. Again go back to server and access manager apps then it ask for credential



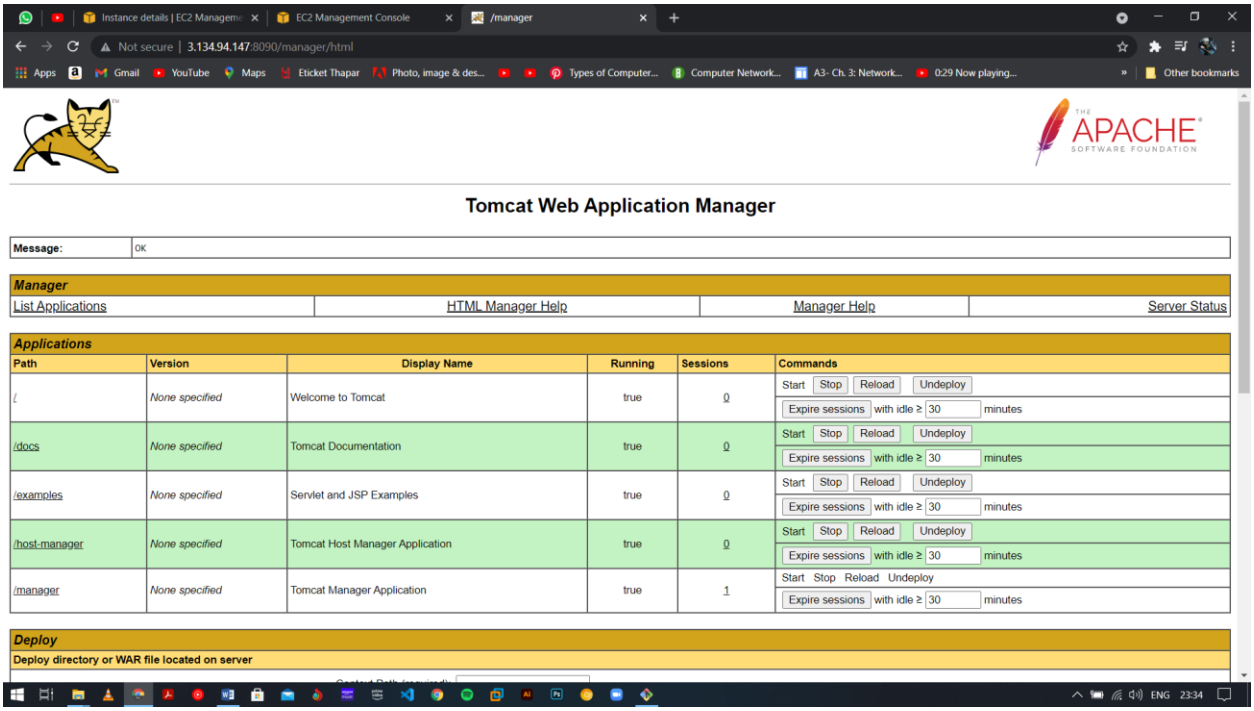
23. In conf there is a file call tomcat.users.xml , here we need to add user and role

23.1. User and role are given below add in between tomcat users:

```
<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="deployer" password="deployer" roles="manager-script"/>
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

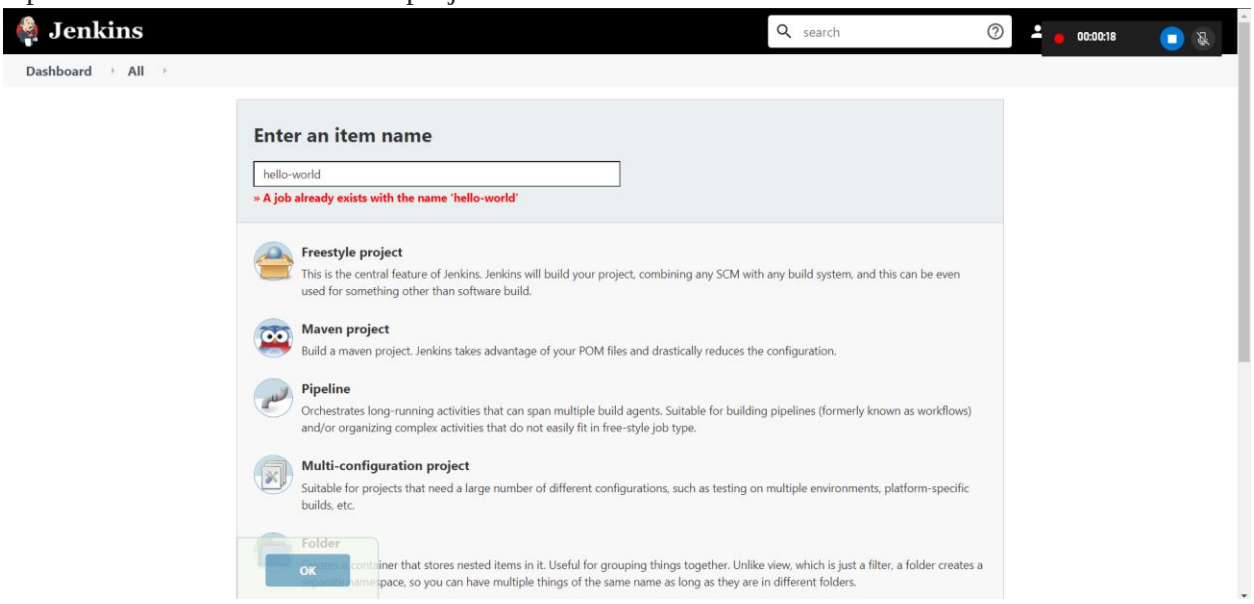


24. Again refresh tomcat server and access manager app sign in with username and password it look like:

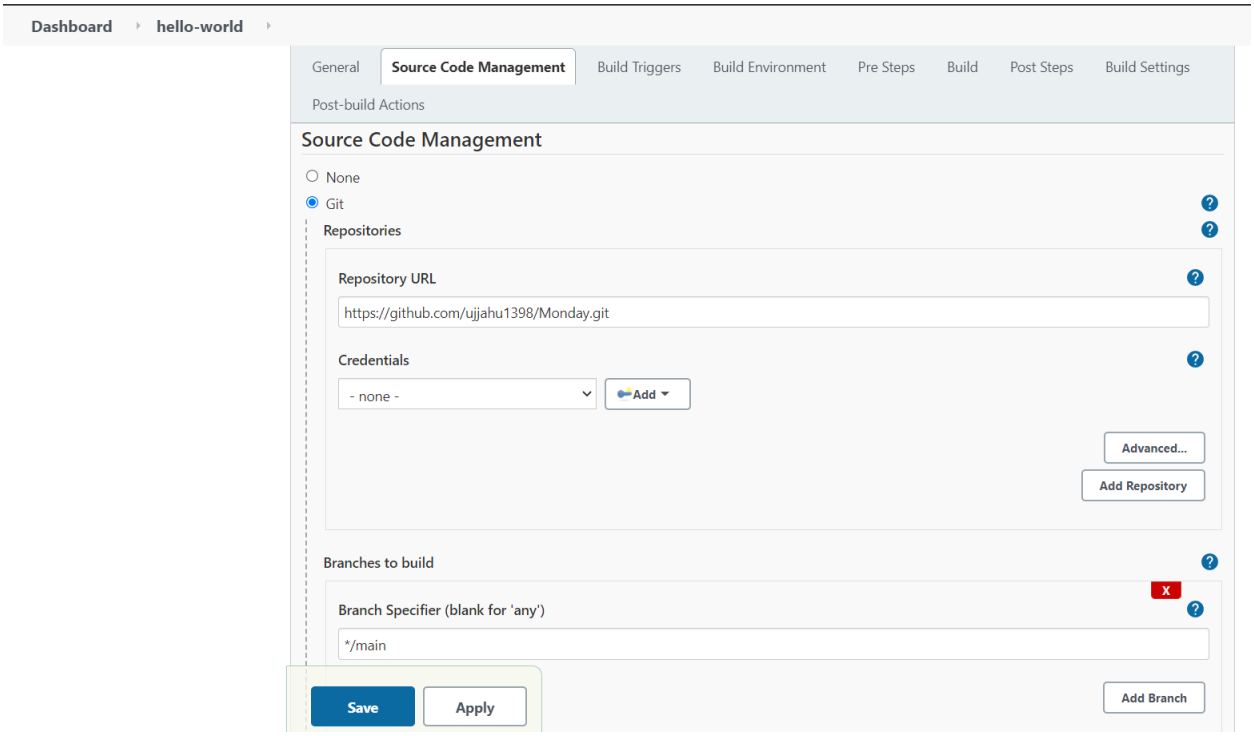


25. Project is cloned where we saved our AWS key.
Link:
https://github.com/ujjahu1398/Monday/blob/main/webapp/src/main/webapp/index.jsp

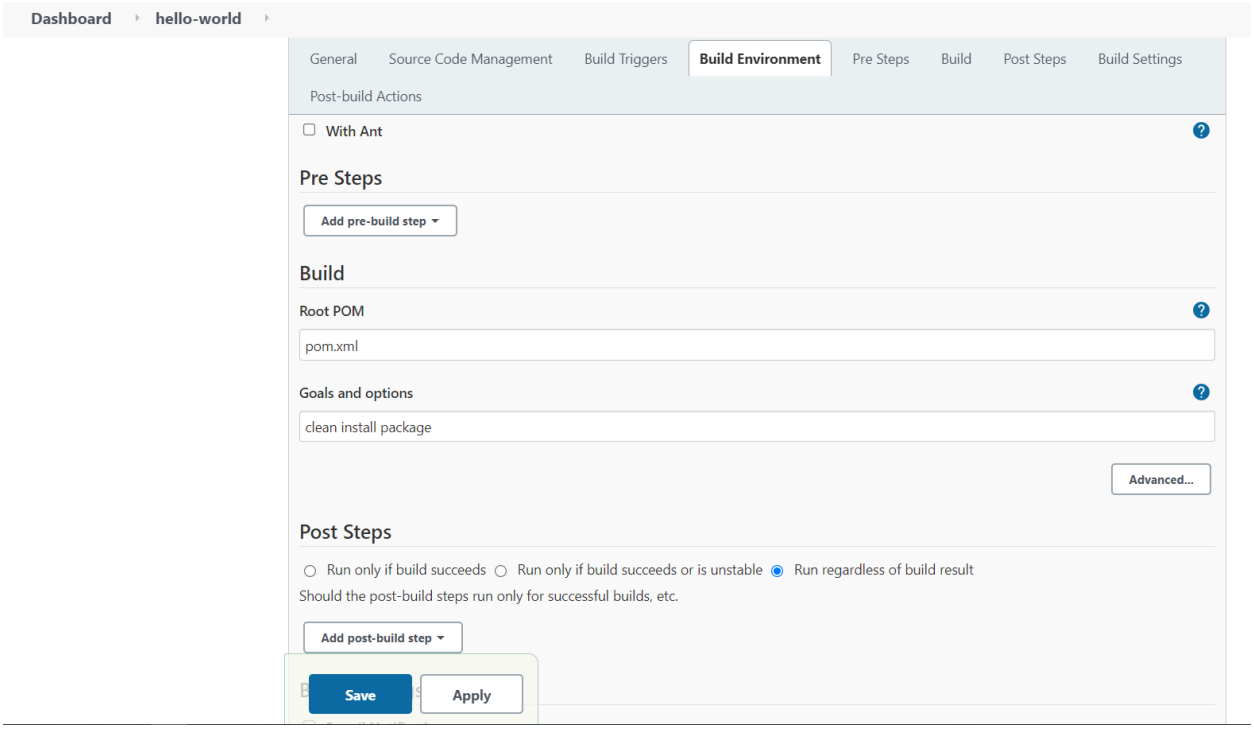
26. Open Jenkins and create a new project



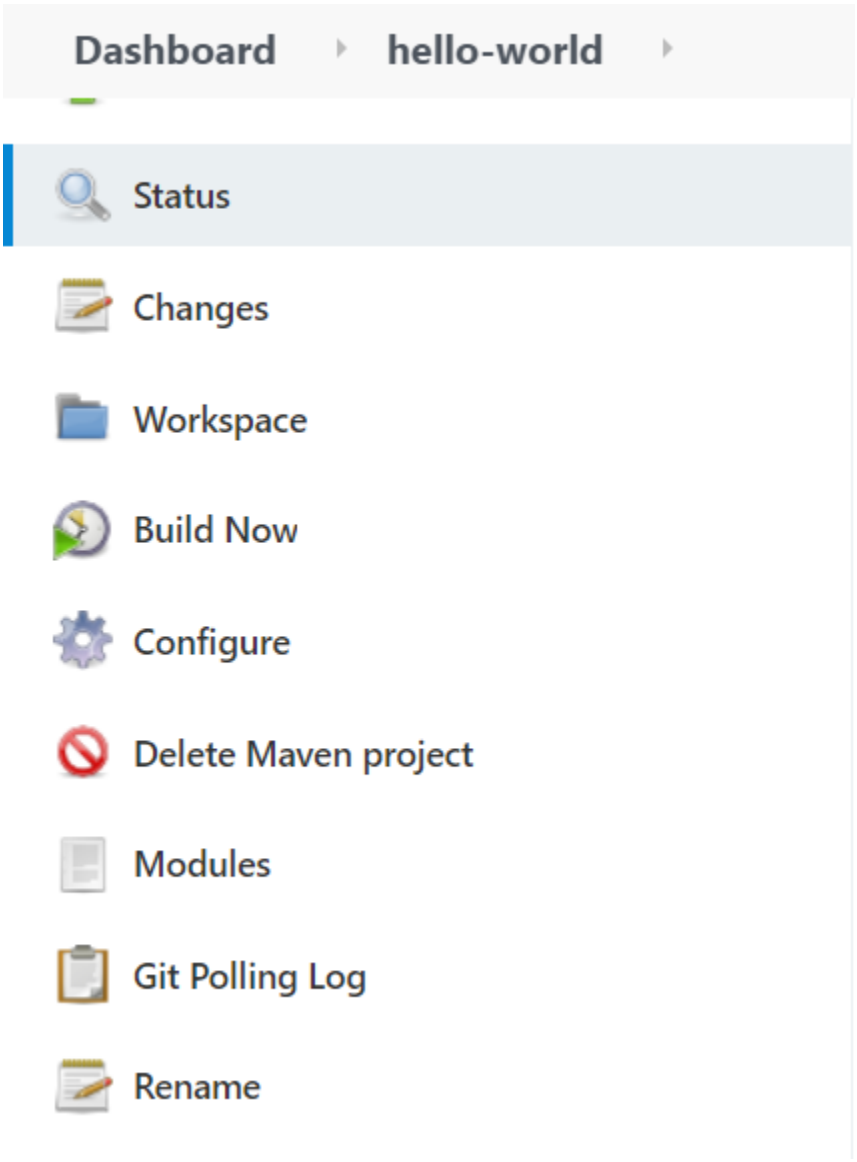
27. Set your git url path



28. Set the goal that means clean old build and just install the maven package



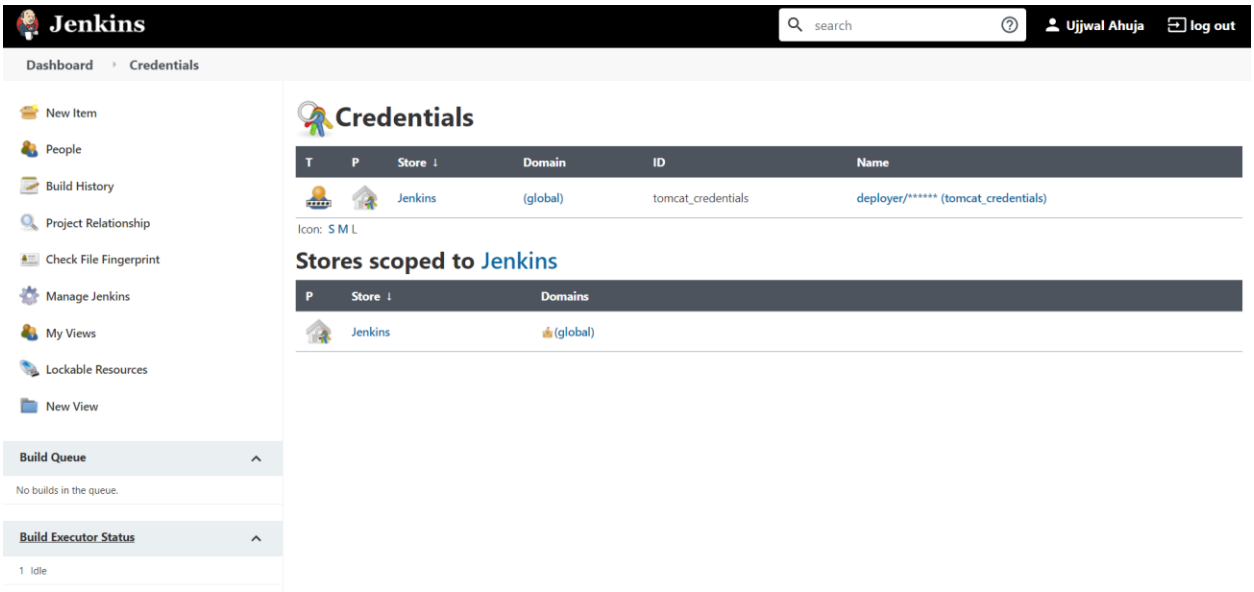
29. Then click on Build Now Manually



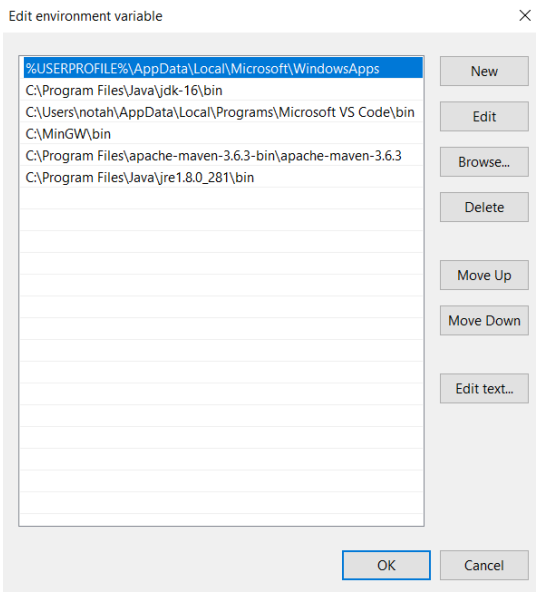
30. After build we need to deploy on server. First open your tomcat server, After that we need to give our tomcat credential to our Jenkins therefore we need to install the deploy plugin on Jenkins.

Go with:
Dashboard-> Manage Jenkins-> manage plugin-> Available-> Deploy to container-> Install w/o restart

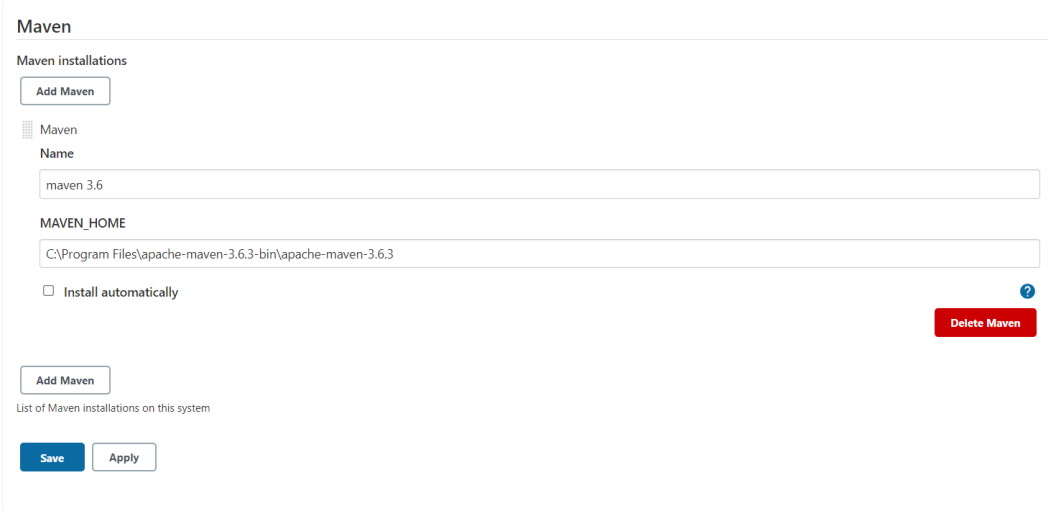
After this we need to set the credentials then,
Go with:
Manage Jenkins-> manage credentials-> global credentials-> add credential



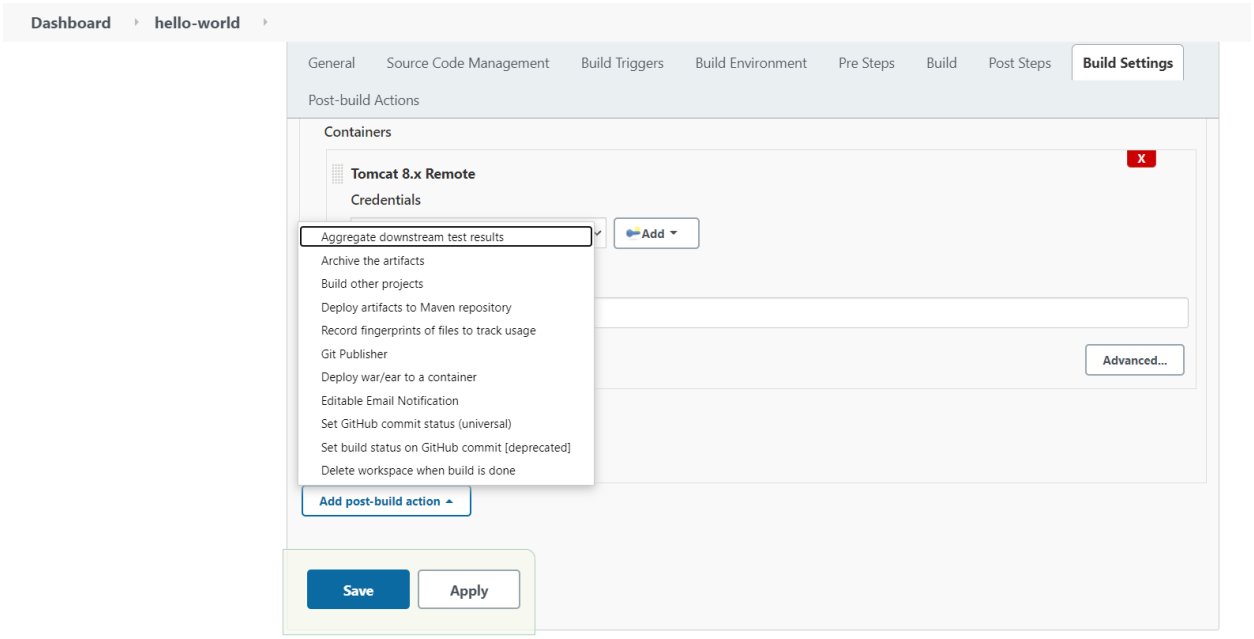
31. Download Apache Mavin and set path in Environment Variables



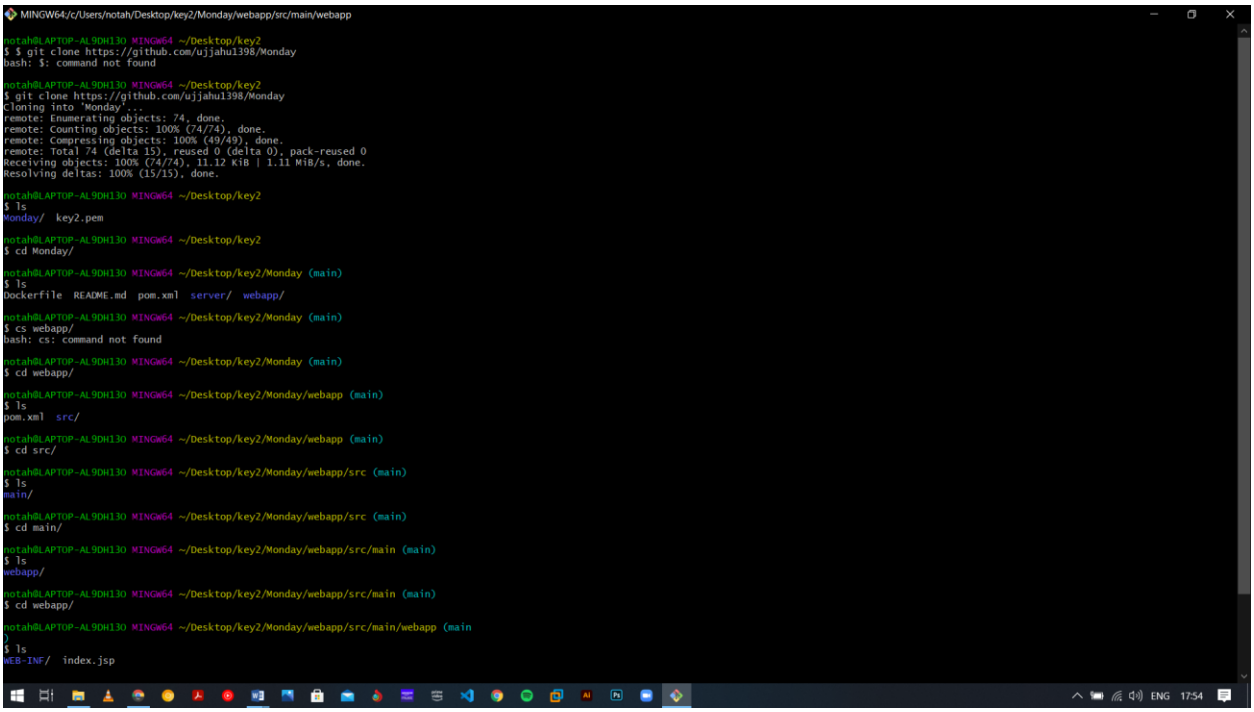
32. For adding maven manually go with:
    Manage Jenkins-> manage global tool configuration-> add maven

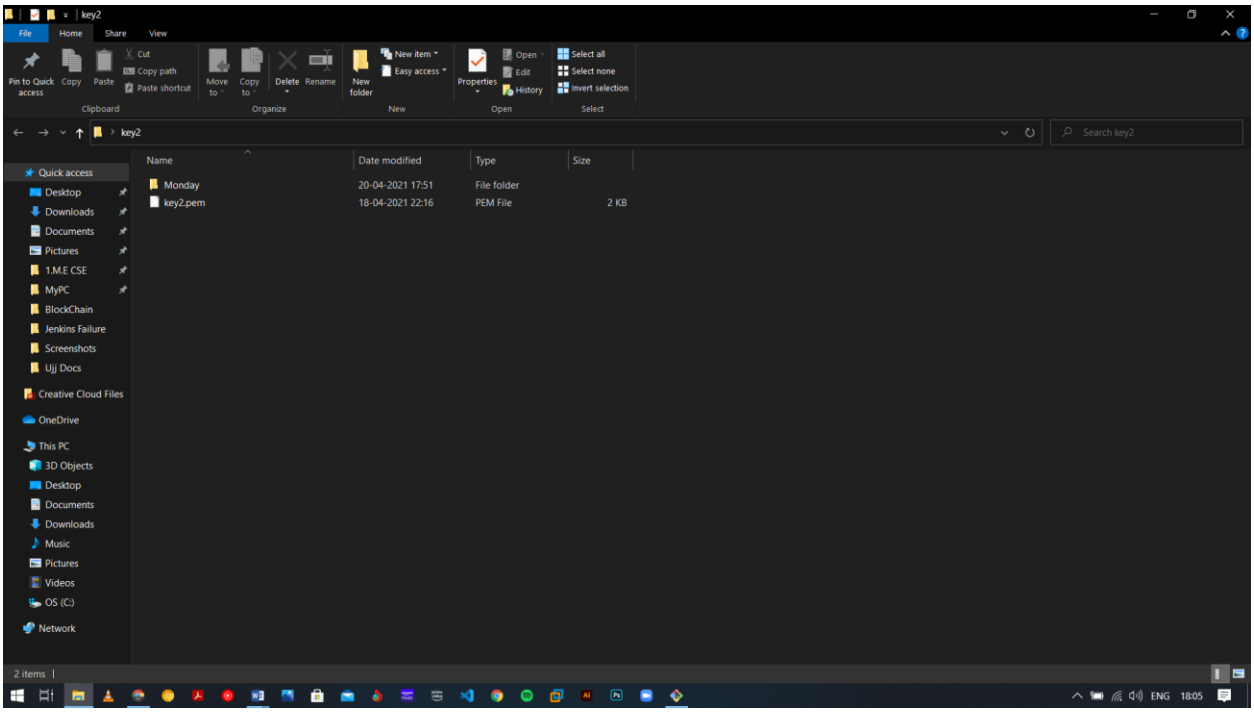


33. Deploy war to container:

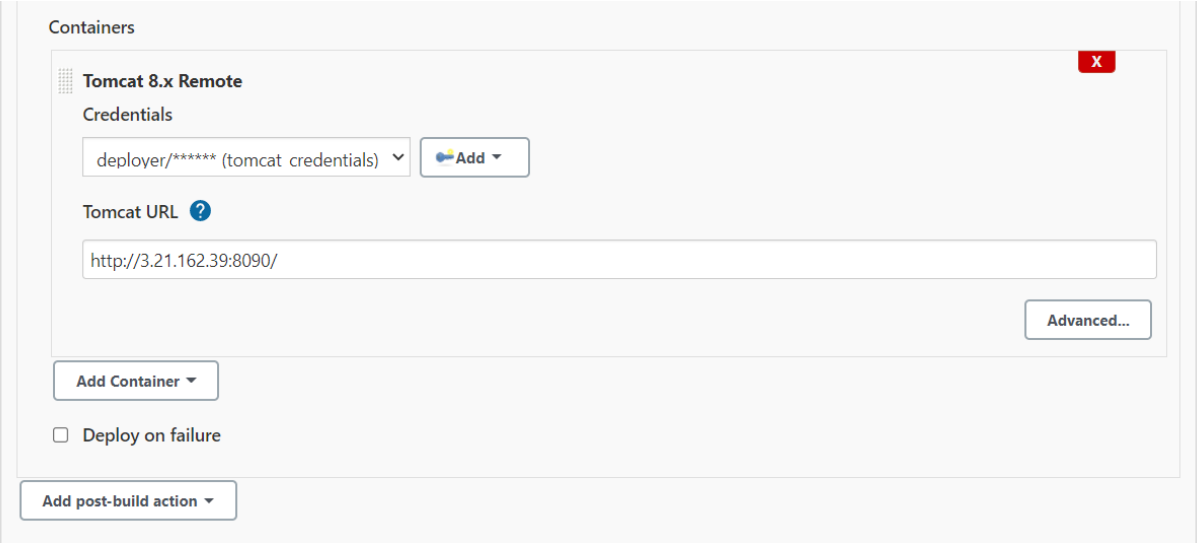


34. Clone the code where your key is saved:

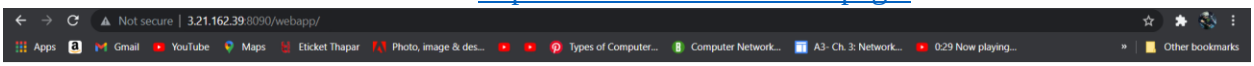




35. When we give the right credential of tomcat then only this war file will deploy on tomcat server



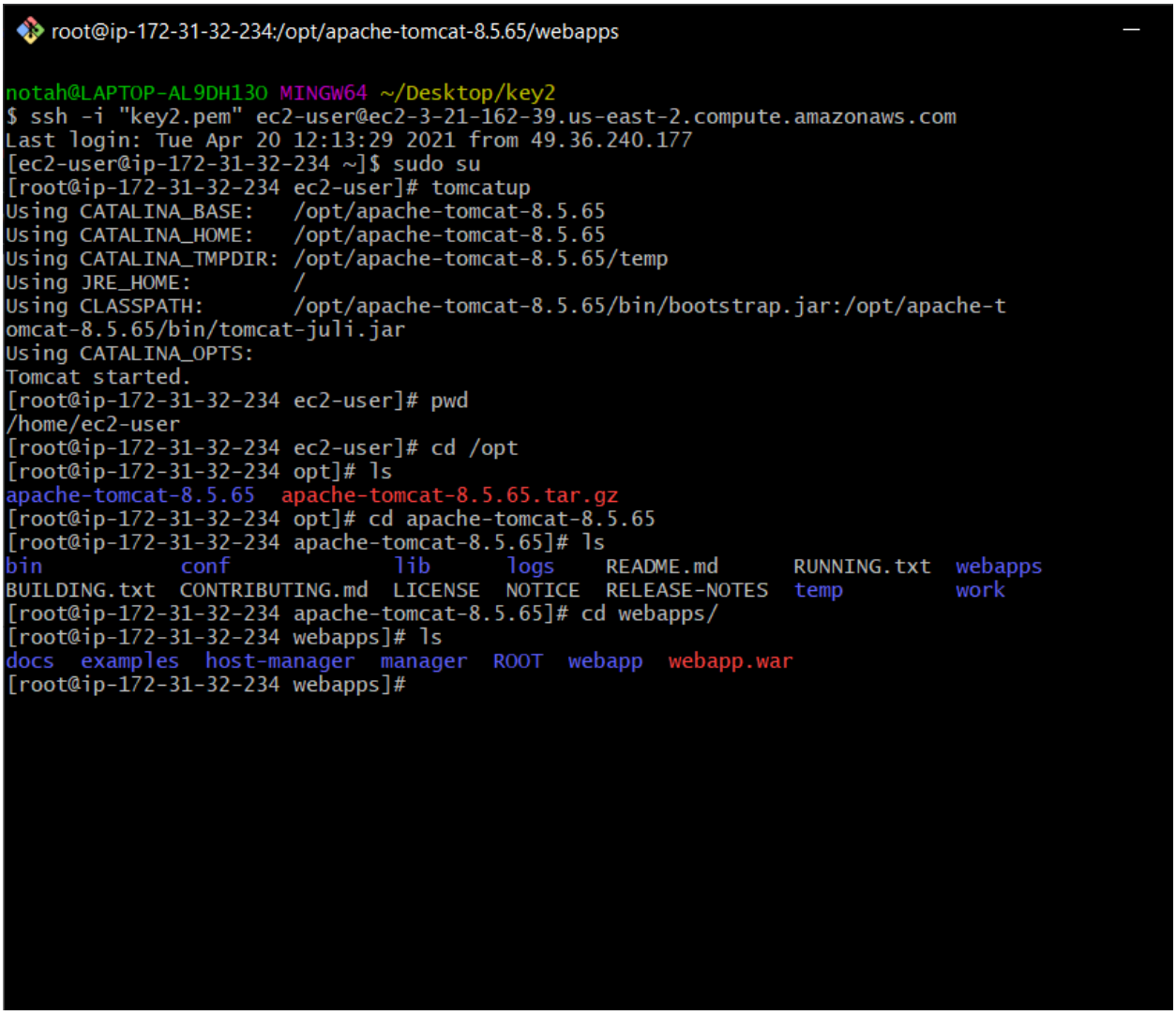
36. Then build now and check the link " http://3.21.162.39:8090/webpage/"



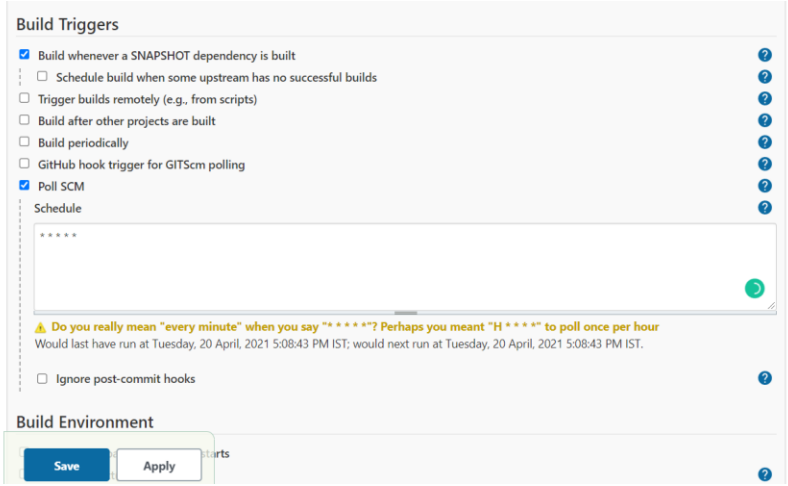
CONTINUOUS DELIVERY AND DEVOPS (LAB) SUBJECT CODE- PCS217P

37. Commit changes in your git repository.

38. Then build it if should able to create the war file once war file is create successfully it simply deploy to server



39. Build automatic trigger for 1 minute.



40. check the link " http://3.21.162.39:8090/webpage/" again: