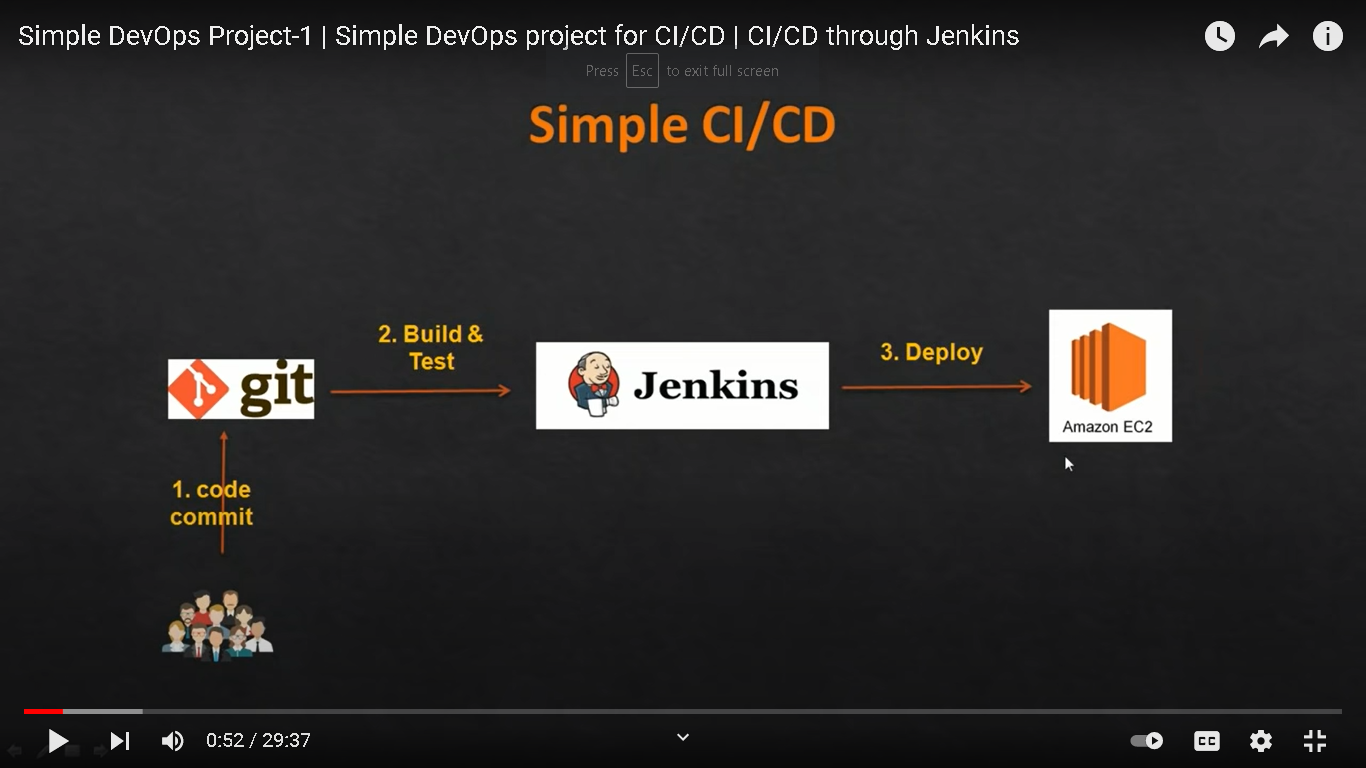
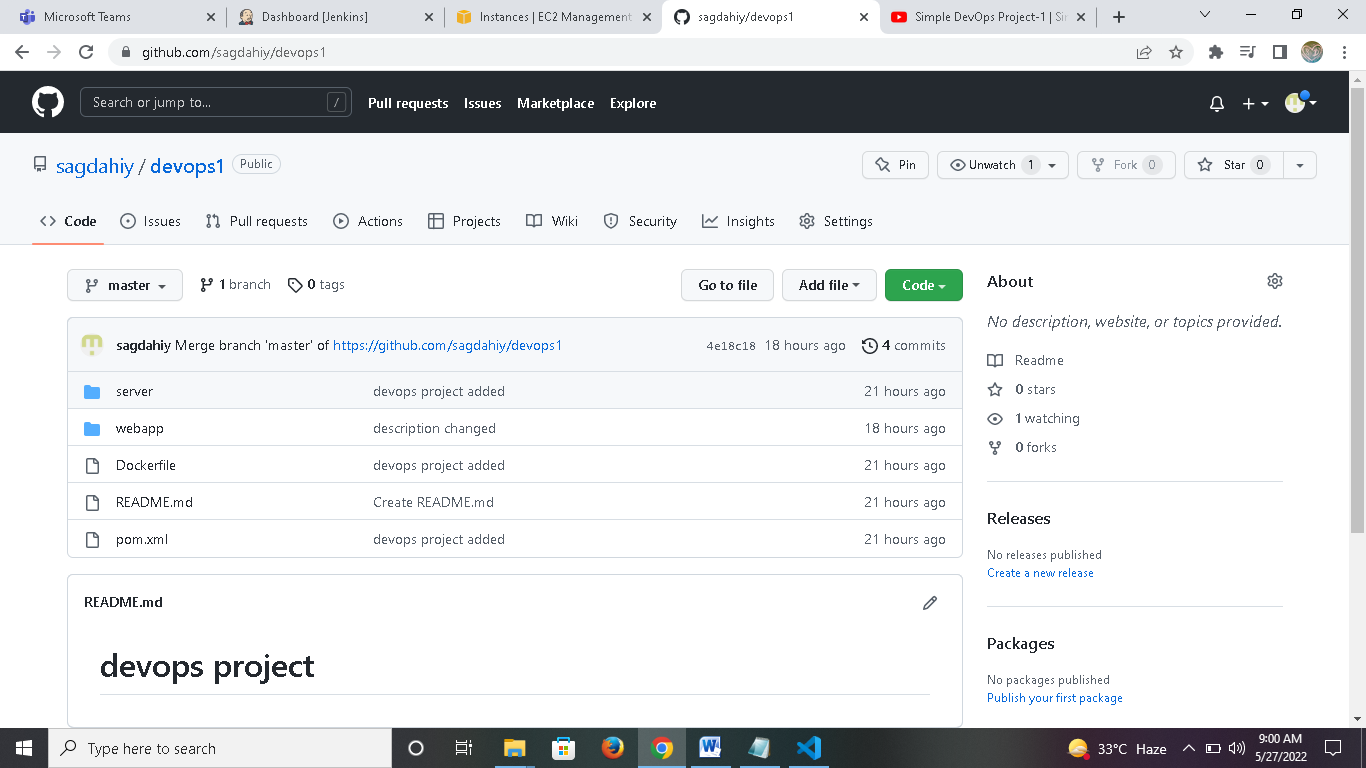
**DevOps Project 1**

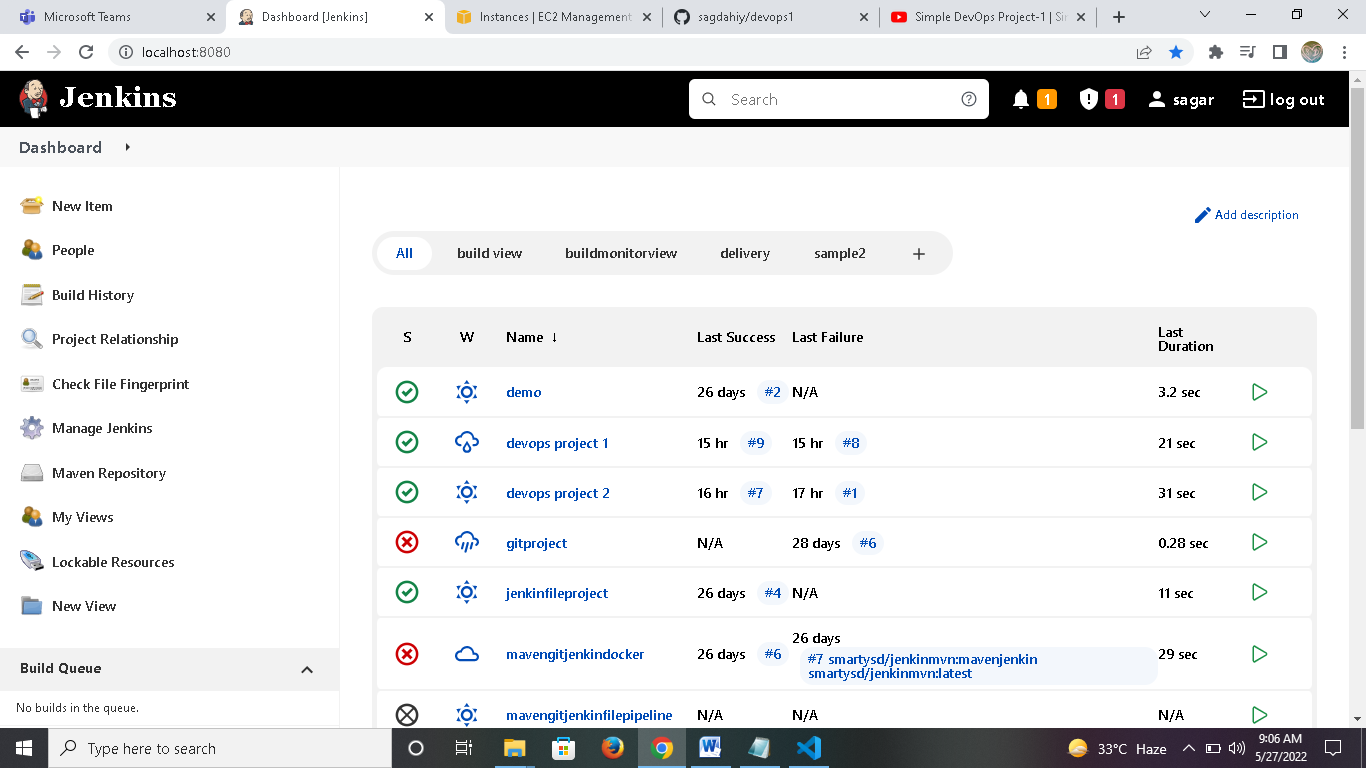


Make maven project and push it into github account

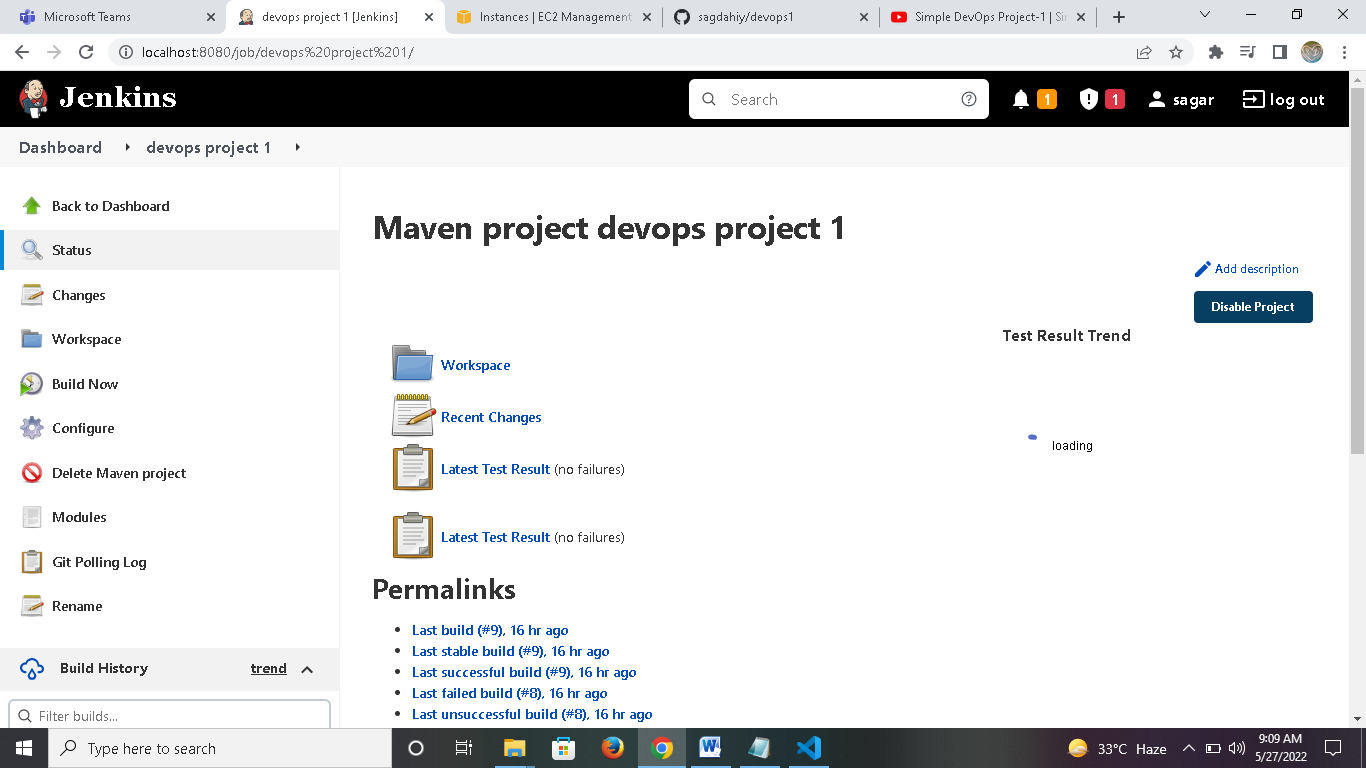


Pom.xml file is used to build maven project

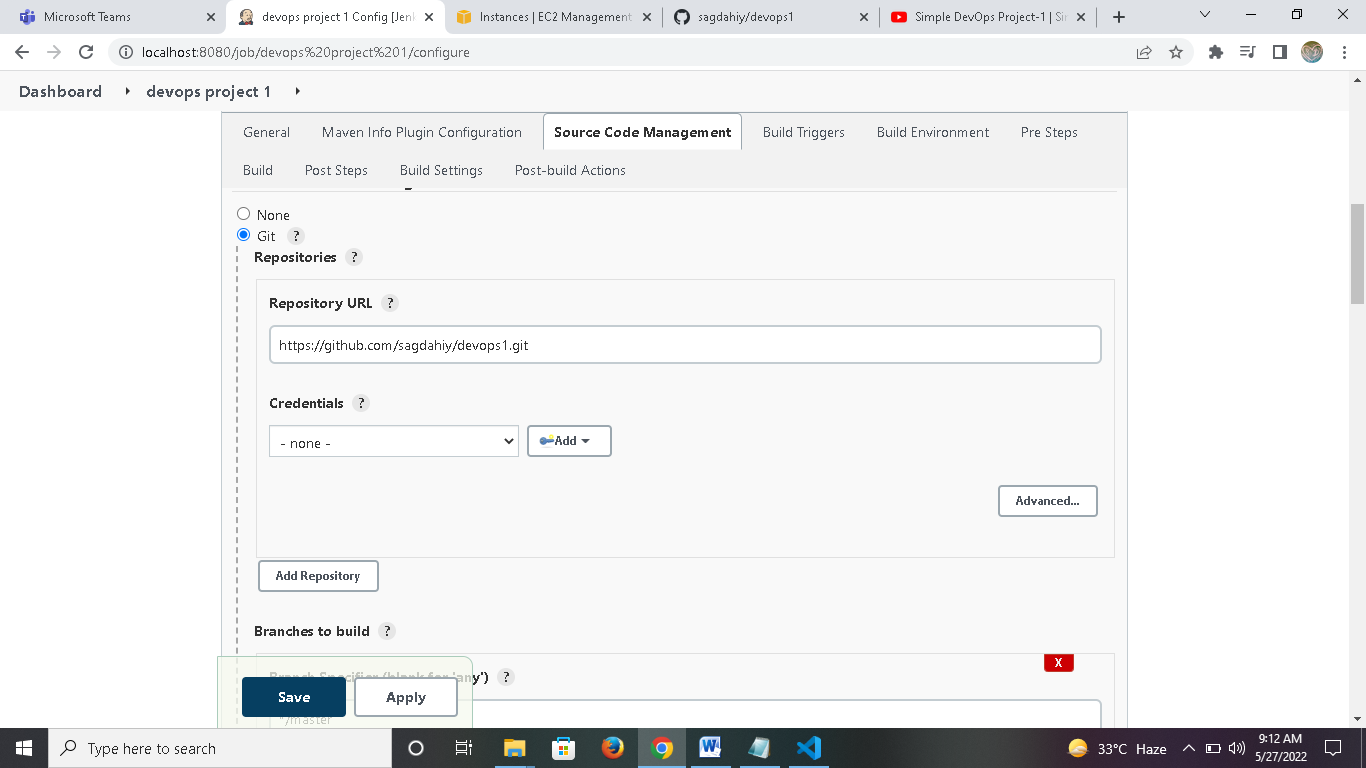
Build Jenkins server and access it on local host port no. 8080



Make a maven project into Jenkins



Go to configure project and add scm of git repository

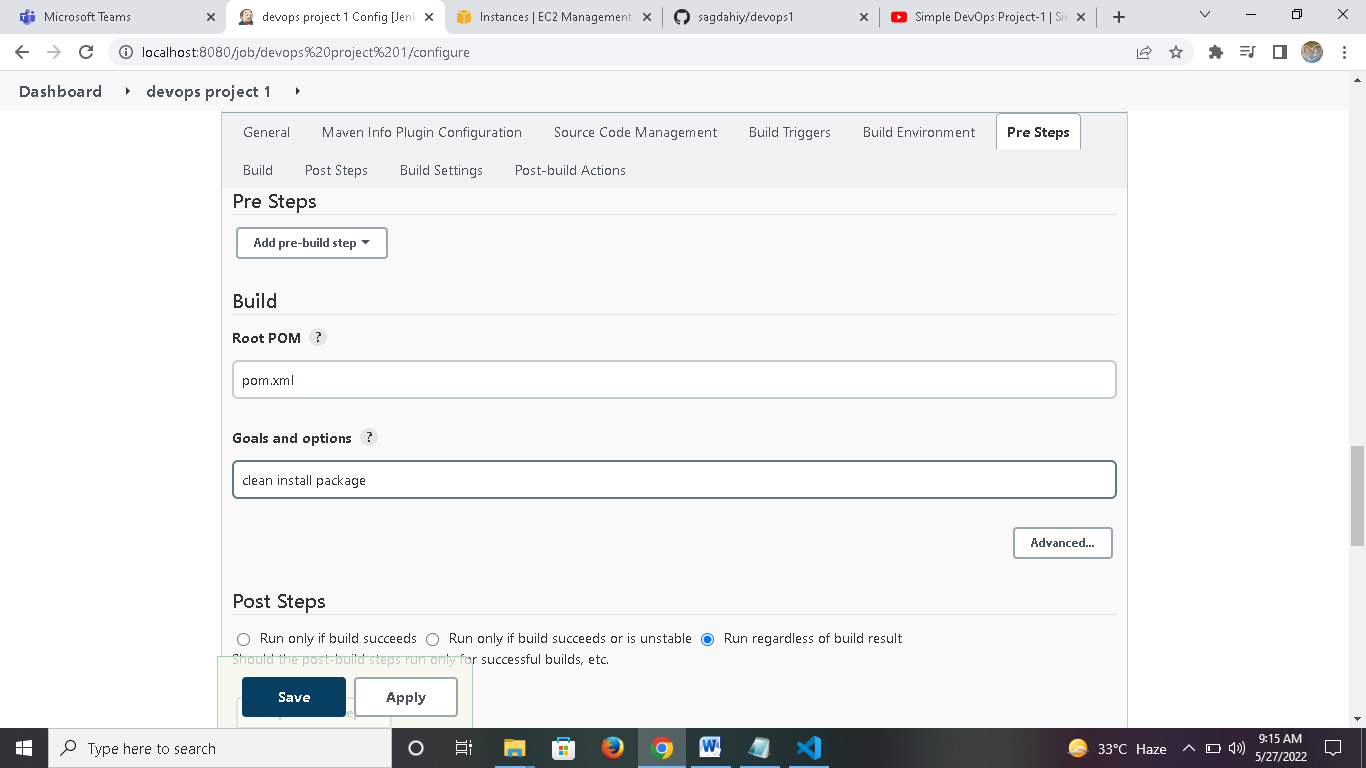


Go to build option and and goals :

Clean : for cleaning previos maven project

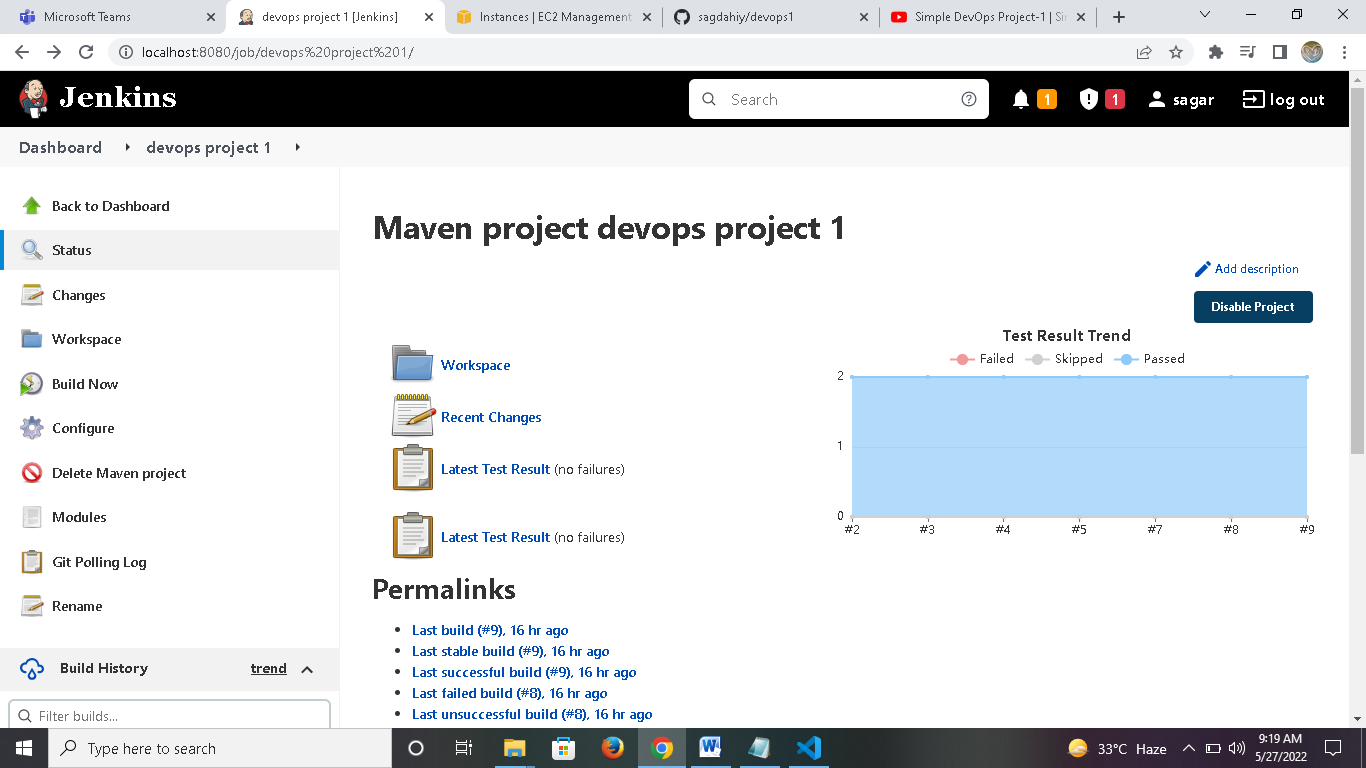
Install : for installing necessary packages and dependency

Package : for building jar,war file from java project

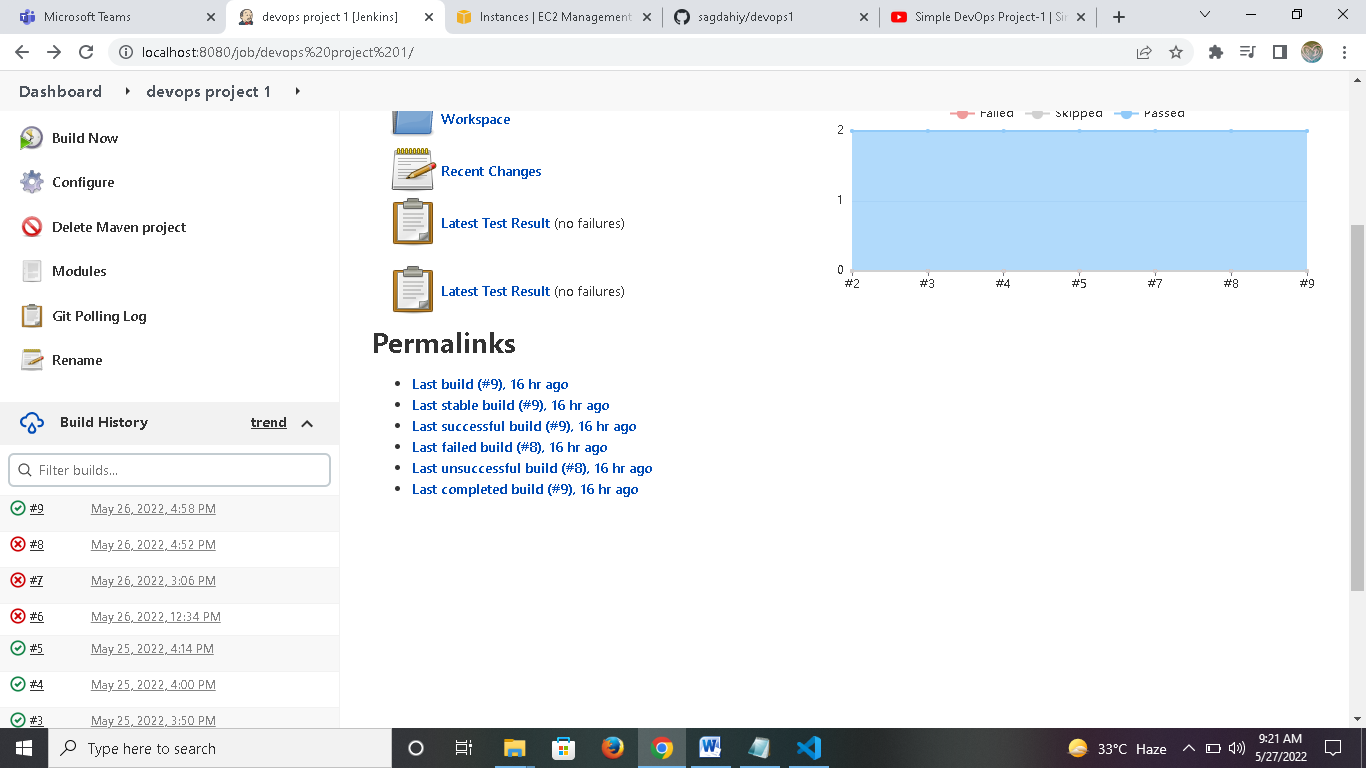


Then apply and save the project.

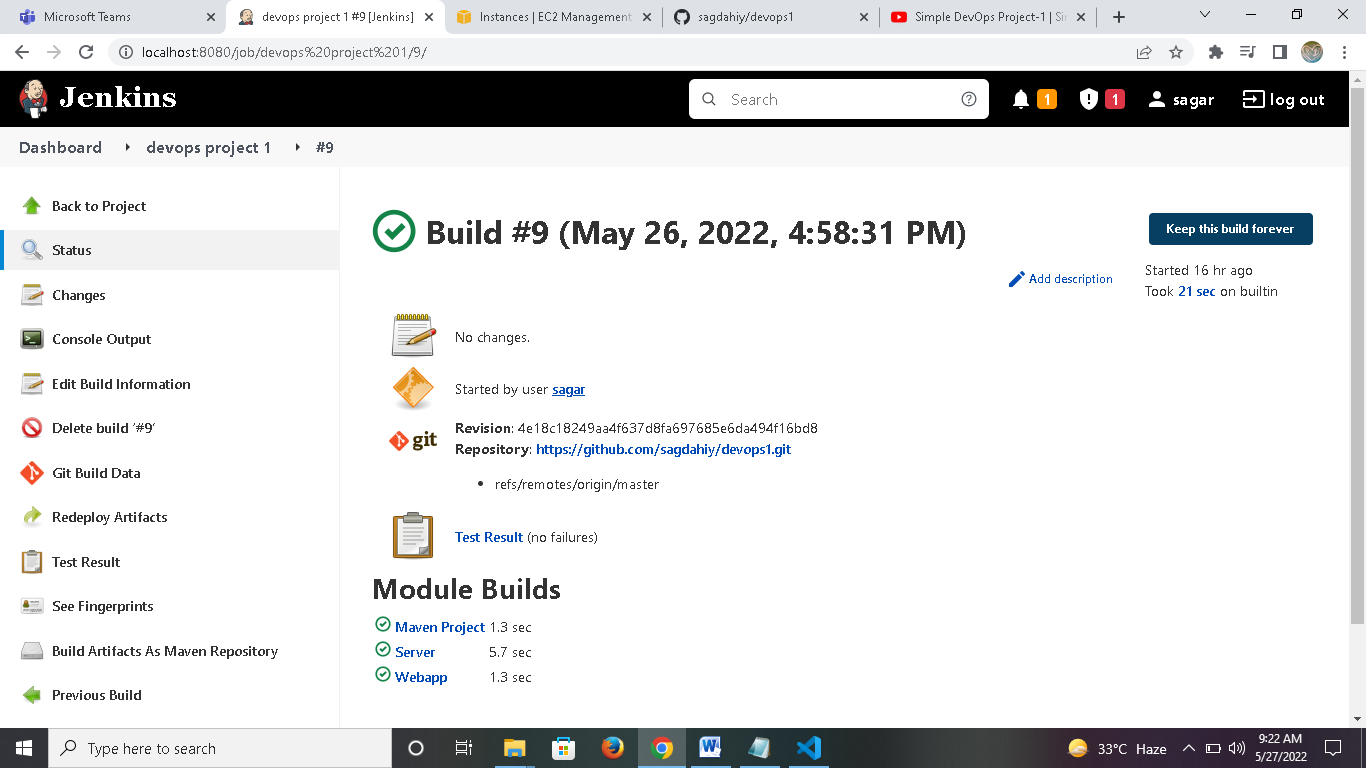
Then click on build now to build the project



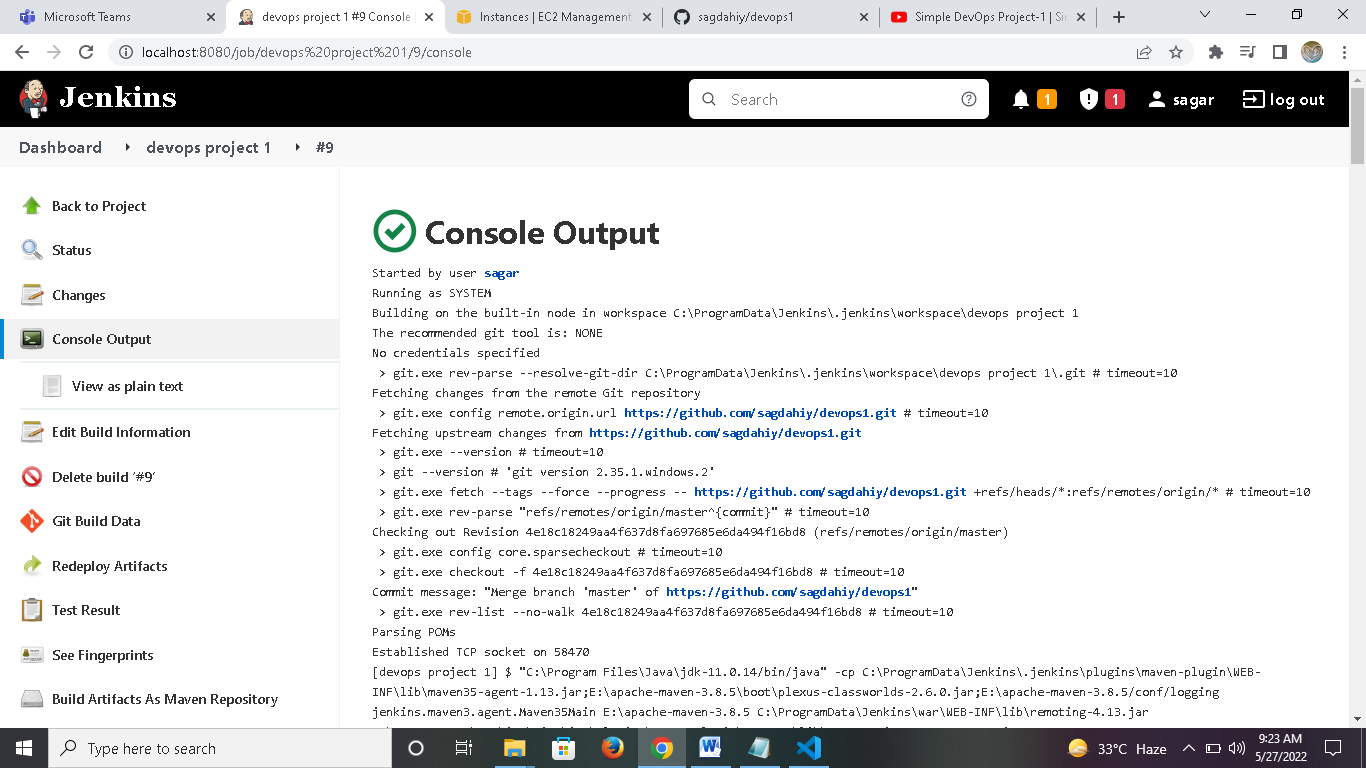
You can check it in build history



To see output and live build process,click on that build no.



Click on console ouput to see output .

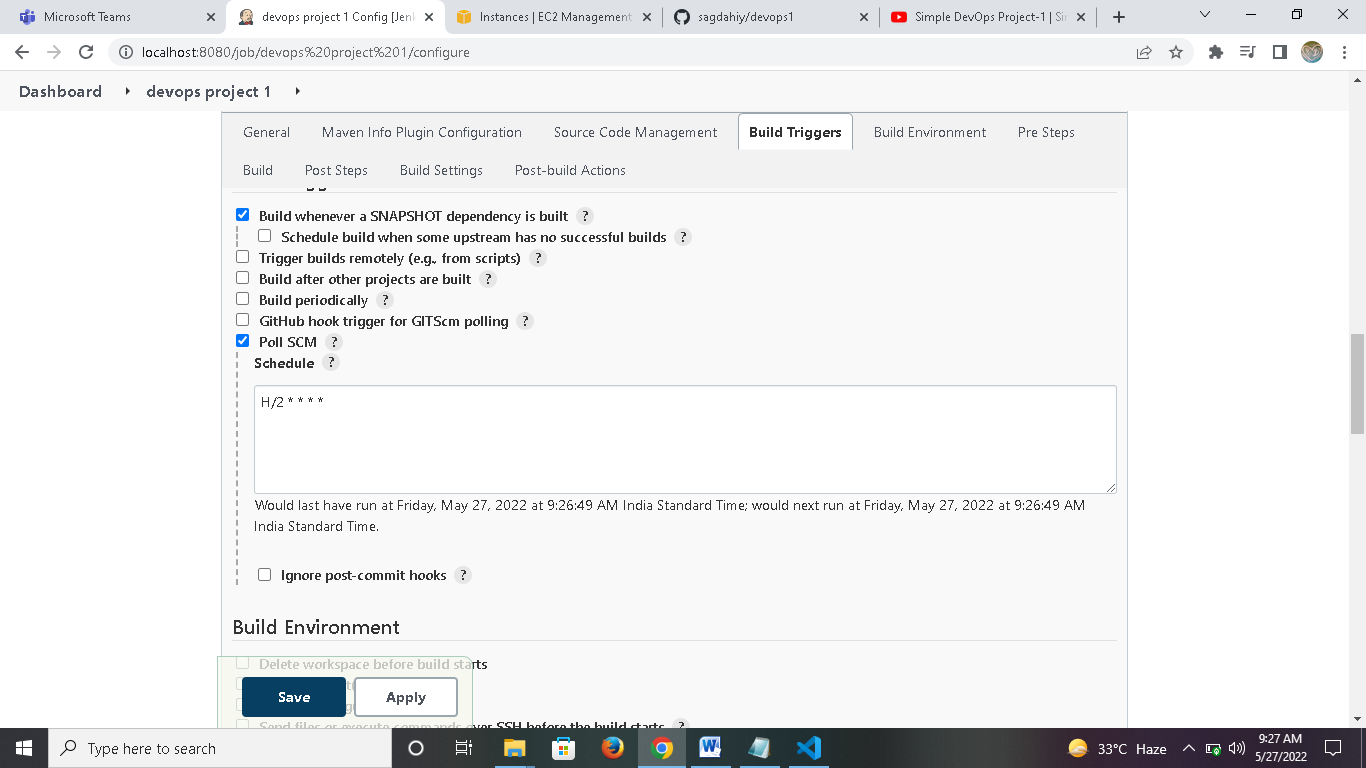


Also you can check build is success or failed in output.

Also our jar,war file is built.

Our continuous integration step is complete.

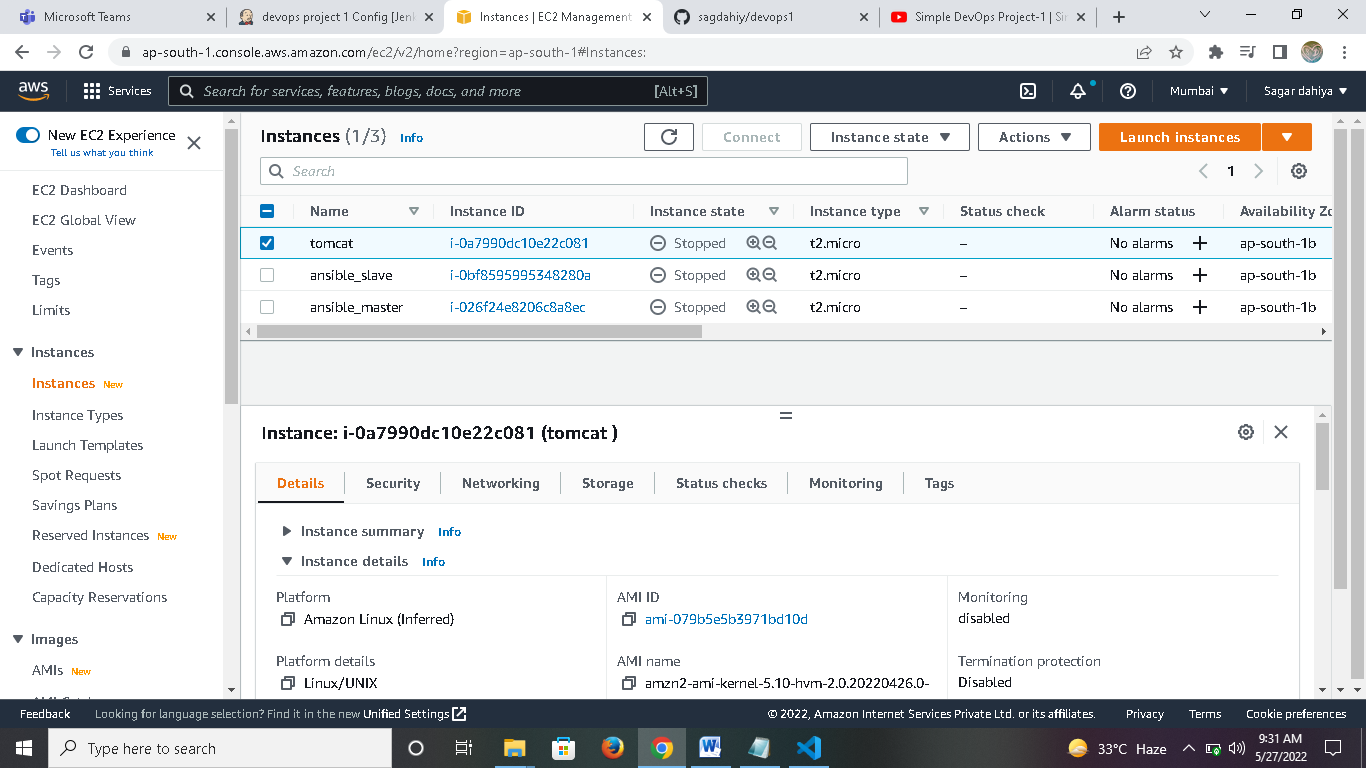
You can also add build trigger i.e if any change in project occur,Jenkins automatically build it



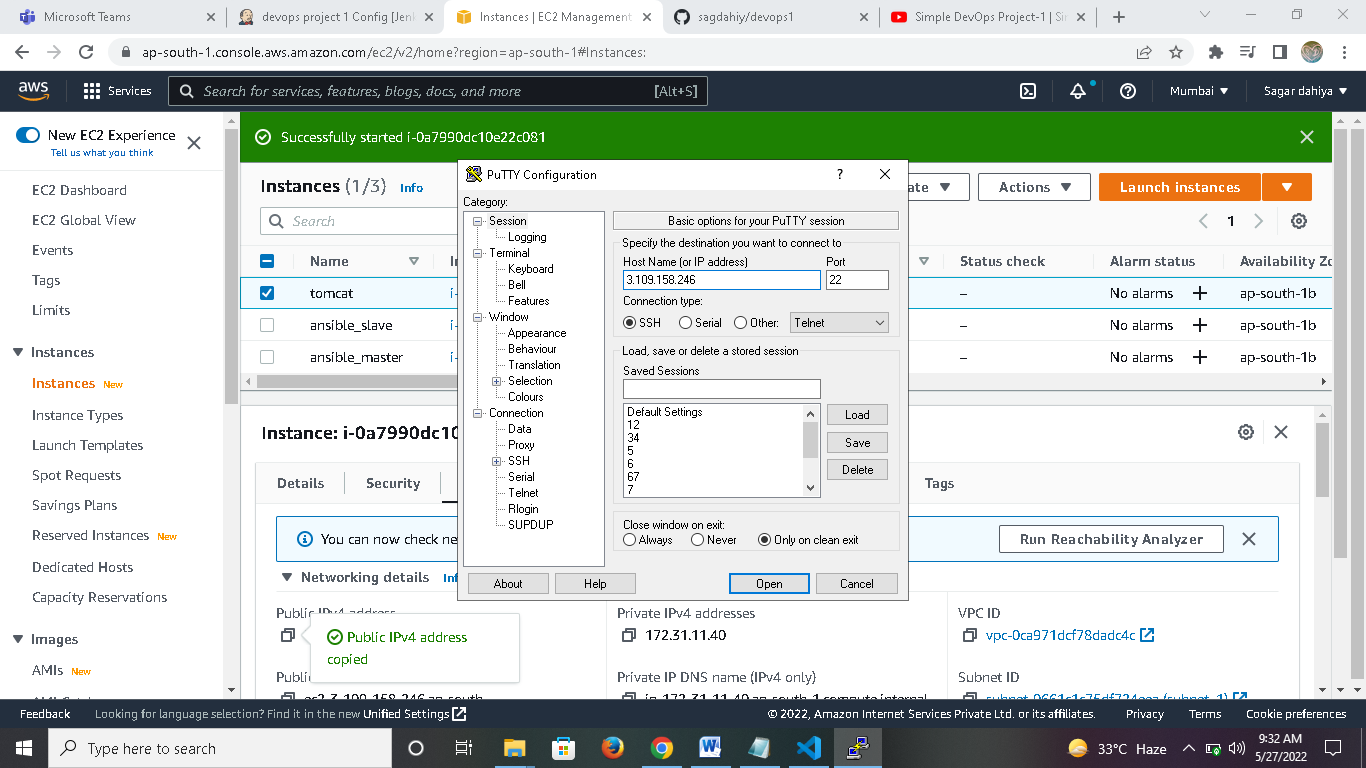
h/2 \* \* \* \* means it check git repo every 2 minutes,if any change is there,it automatically build project again.

CONTINUOUS DEPLOYMENT VIA TOMCAT SERVER

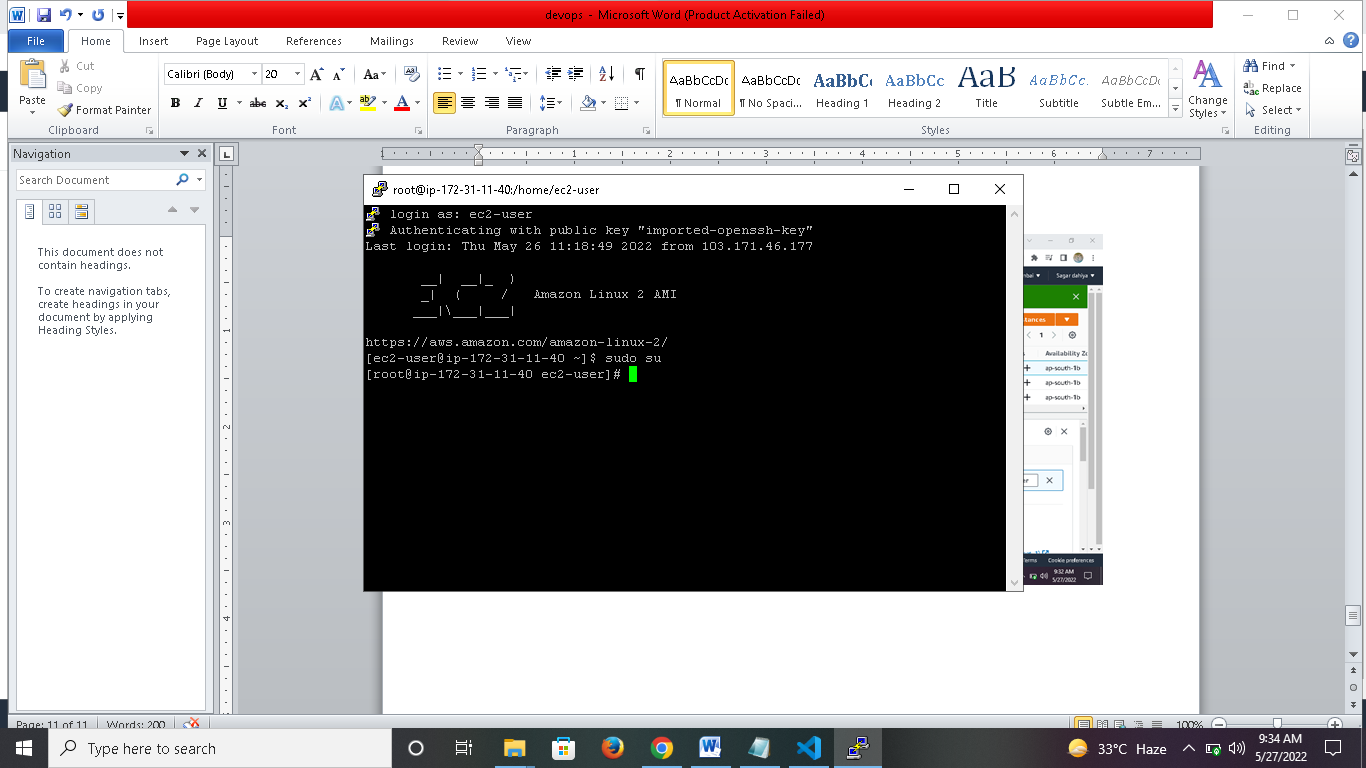
Make ec2 instance on aws and install tomcat8 into it



Launch ec2 instance via putty

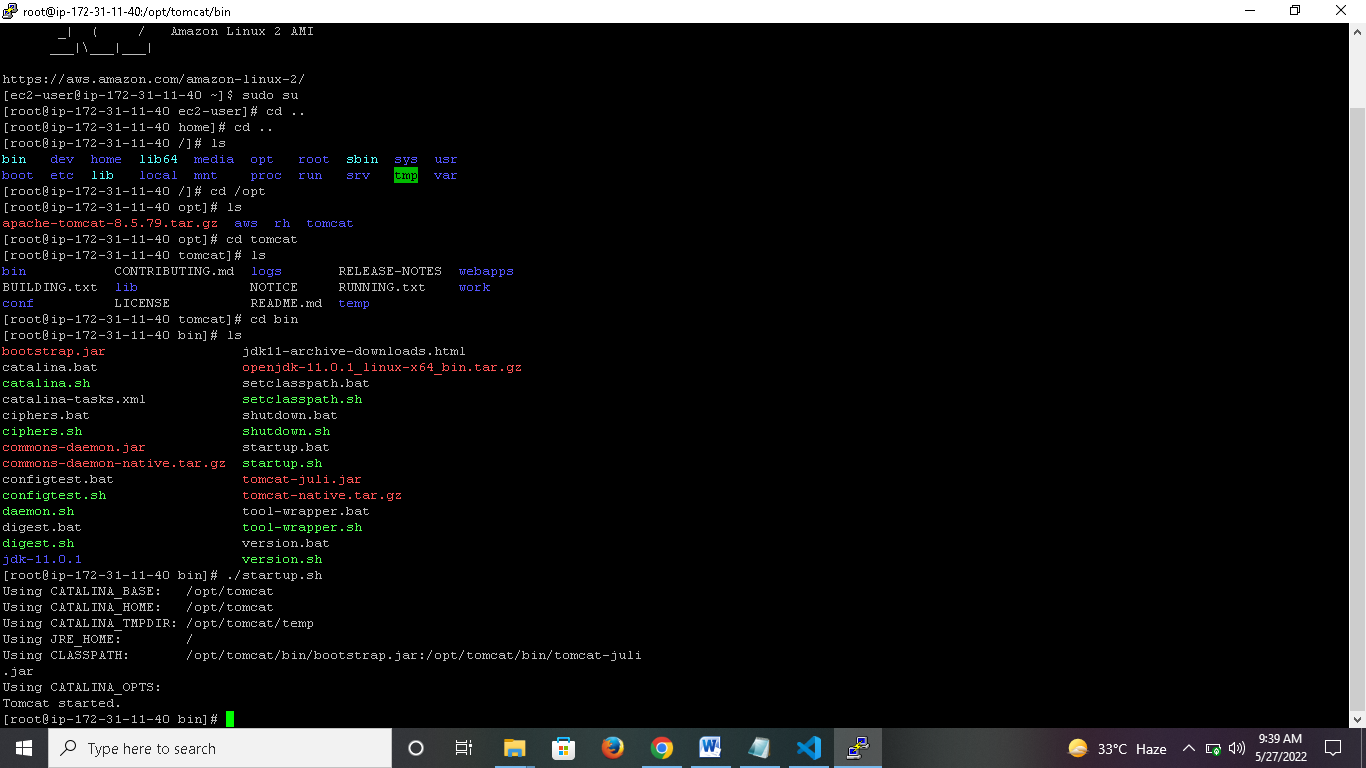


Login as ec2-user into it



Install tomcat8 tar.gz file using wget command and extract it in /opt dir.

<https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.63/bin/apache-tomcat-9.0.63.tar.gz>

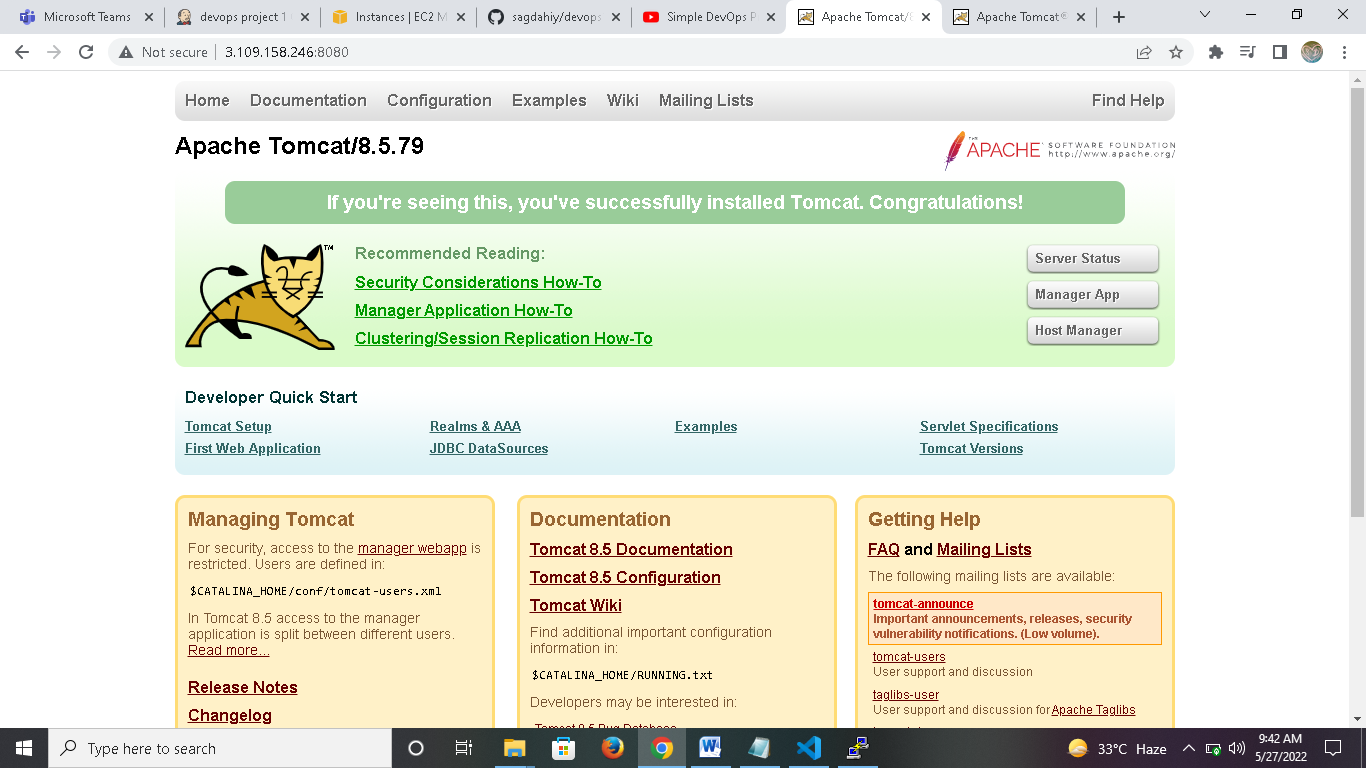


Go to tomcat dir and then in bin dir.

You will see startup.sh and shutdown.sh to start and stop tomcat server

Enter ./startup.sh to start tomcat server and you can see tomcat started.

Enter public ip and port no. to access it in browser.

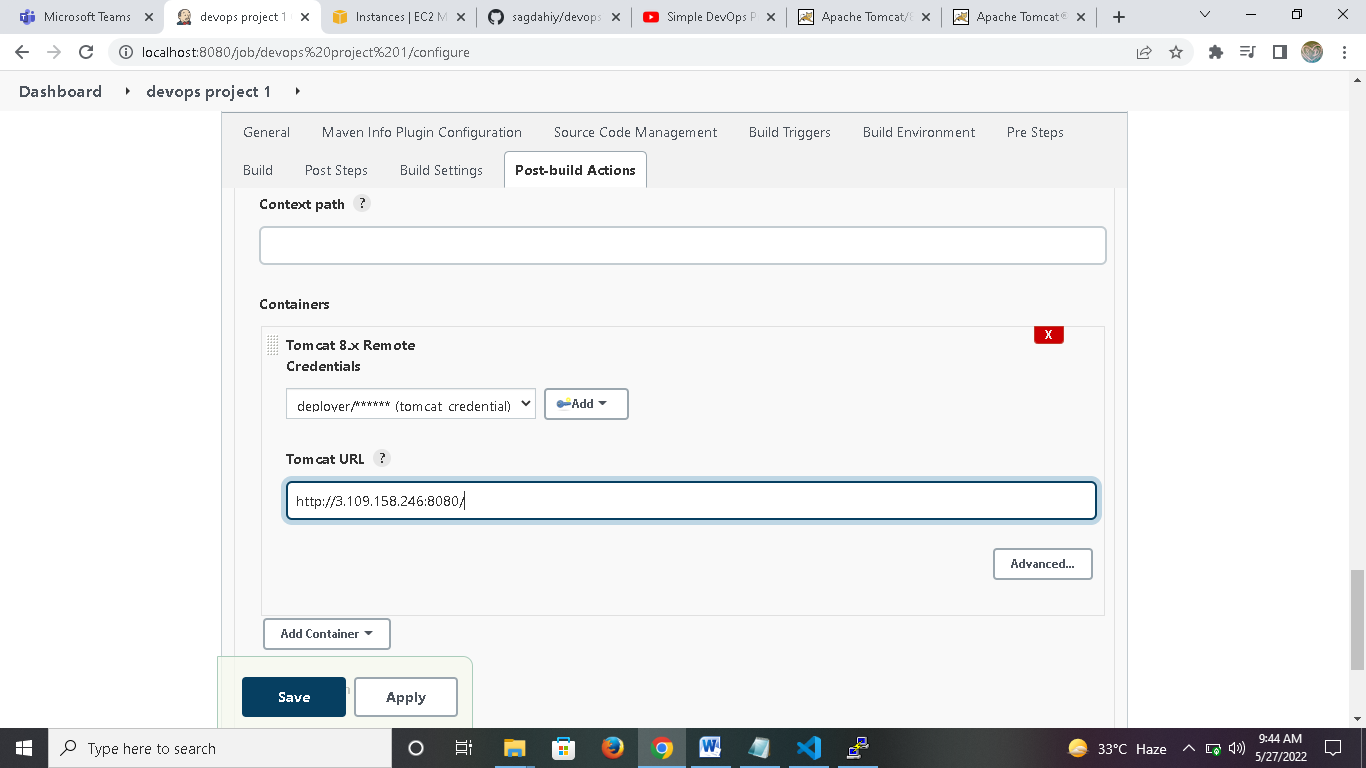


Go back to configure project in Jenkins and add post build action.

Enter url of tomcat server and credentials.

Apply and save the project.

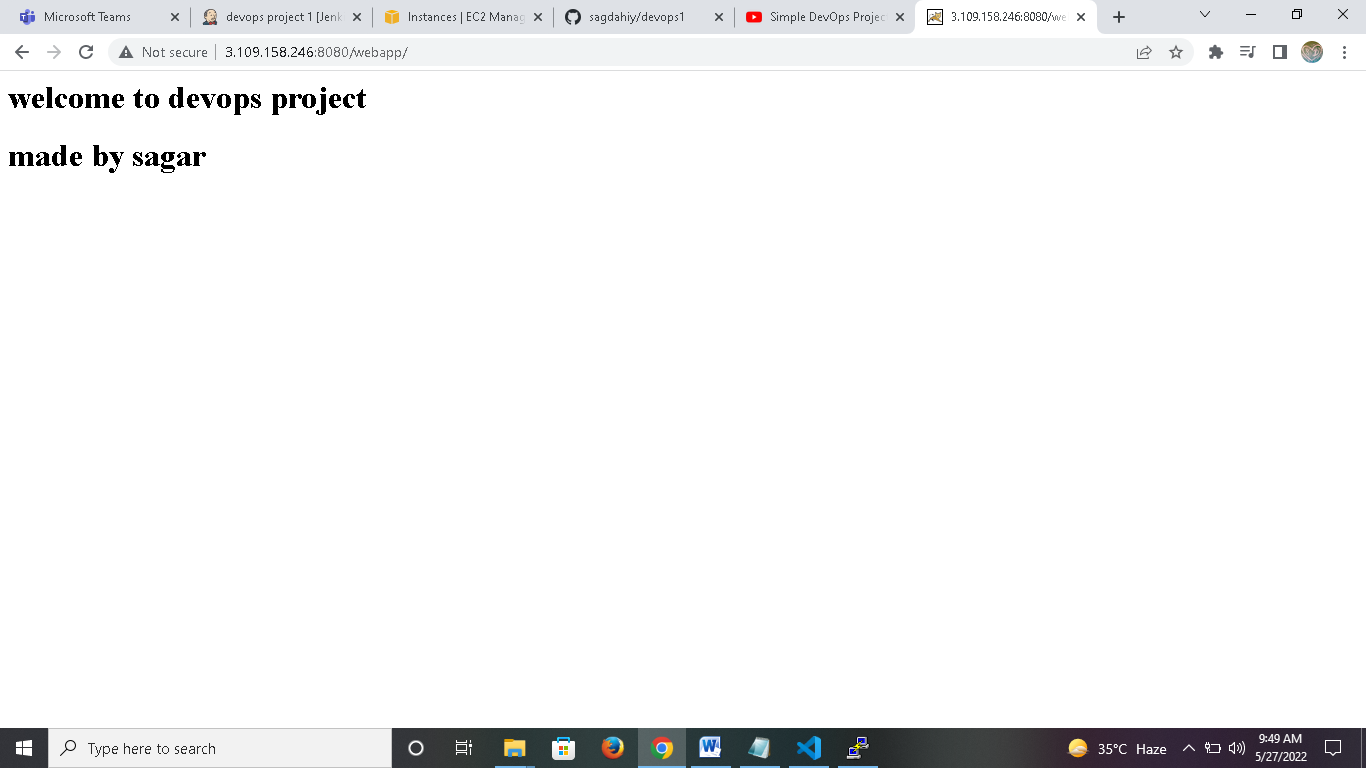
Again build it.



After build jar and war file,it upload them to tomcat server iin webapp dir.

You can access it in web browser

<http://3.109.158.246:8080/webapp/>



Now we complete continuous deployment step.