Project-1

Project Description:

Build CI / CD Pipeline using Jenkins and deploy the real world Web Application in AWS Cloud.

Ans:

1.LAUNCH AN INSTANCE IN AWS

1. Open up the AWS Management Console

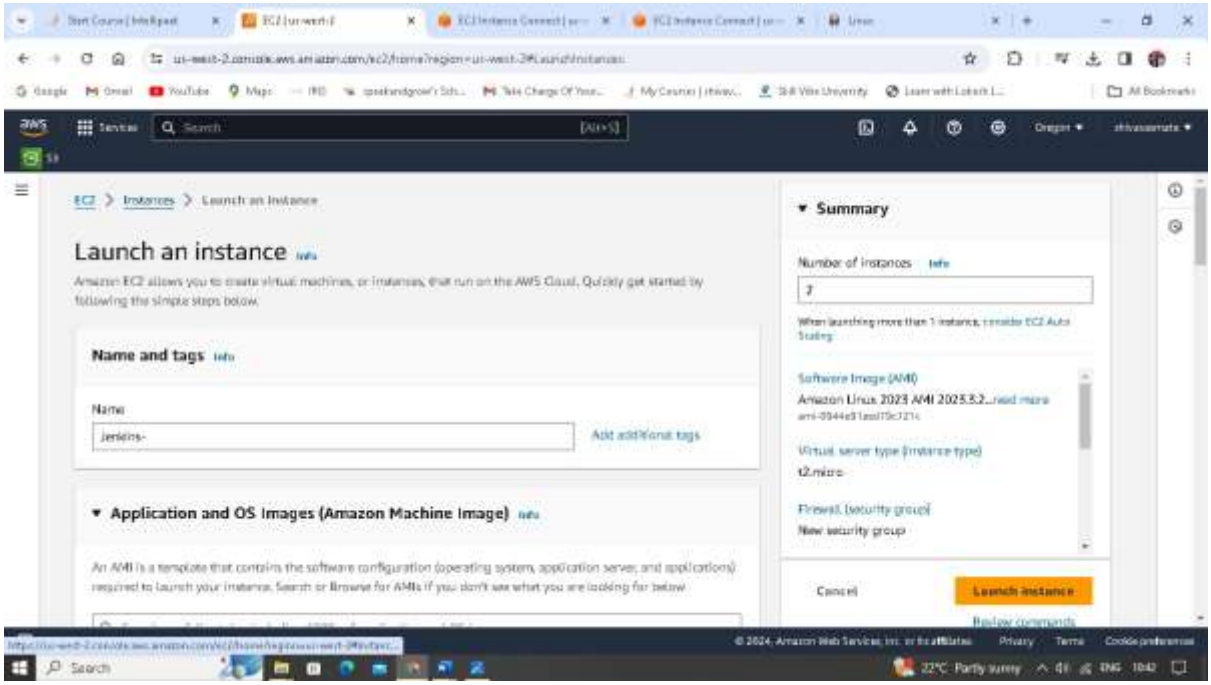
2. Check for the region [us-west-2(. Oregon)]

3. Search for EC2 in the search box

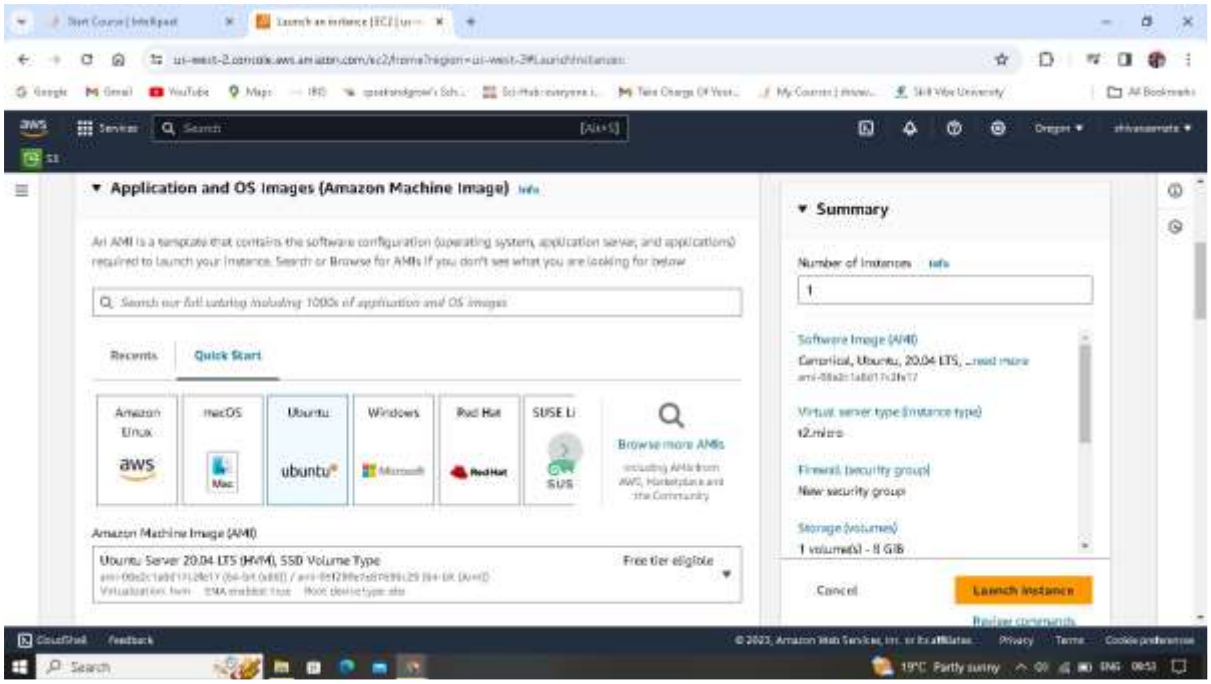
4. Click on instances to go to the EC2 console

5. Click on Launch Two Instances and setup the instance for UbuntuOS:

a. Name: Jenkins-

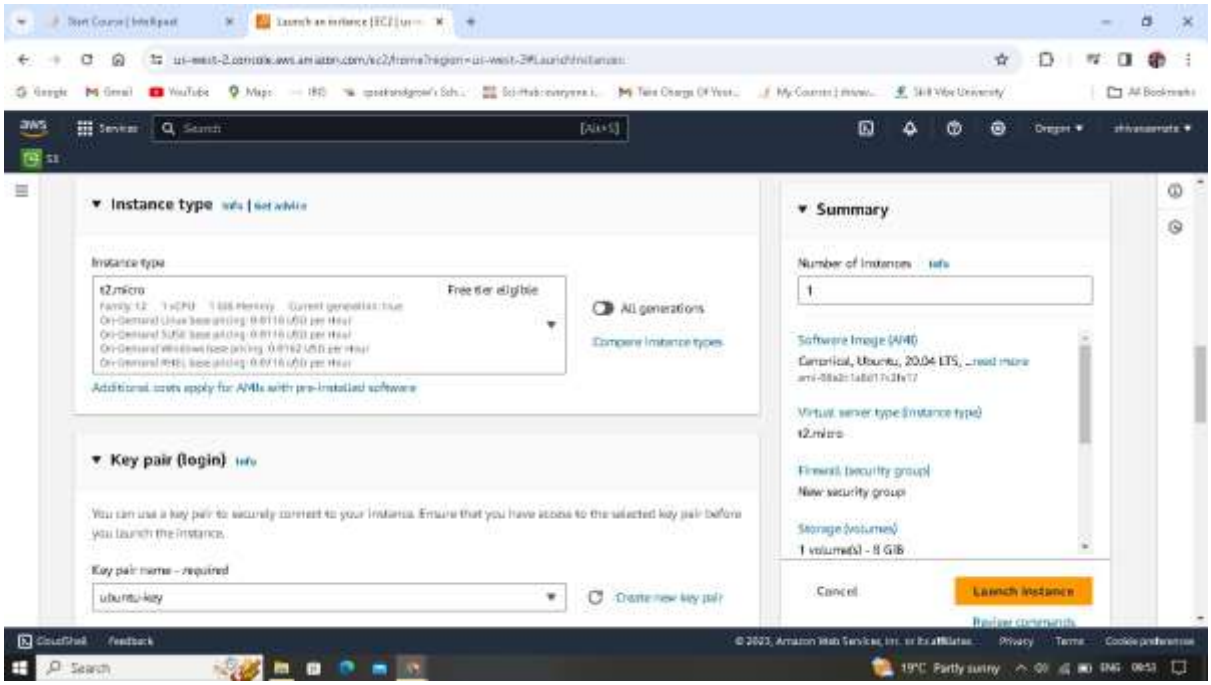


b. AMI: QuickStart >> Ubuntu [Any version which is free tier eligible]

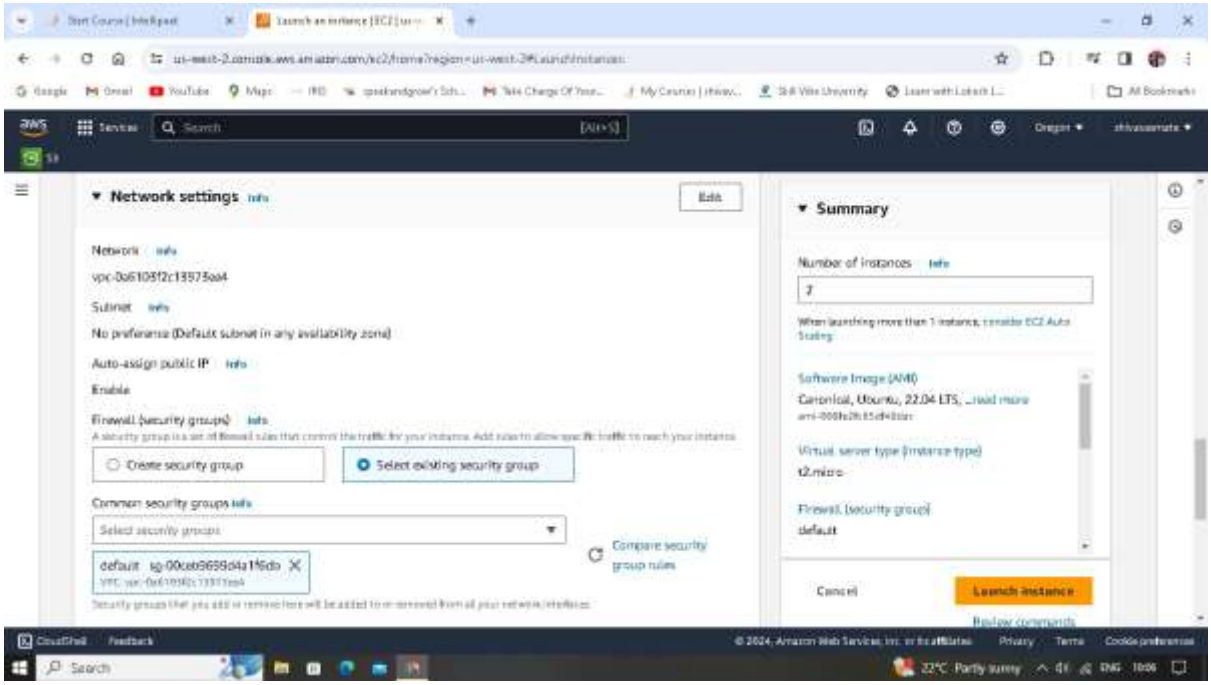


c. Instance type: t2. micro [free tier eligible]

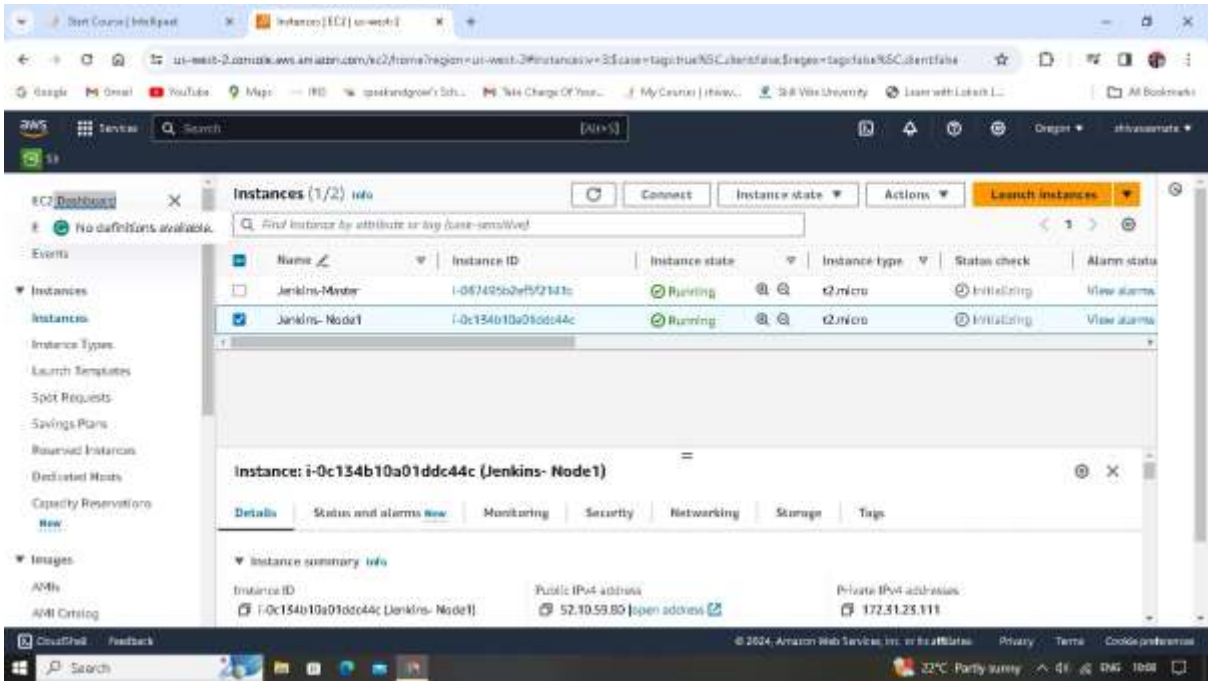
d. Key-pair: Create a key pair [rsa and .pem] with a name ubuntu-key



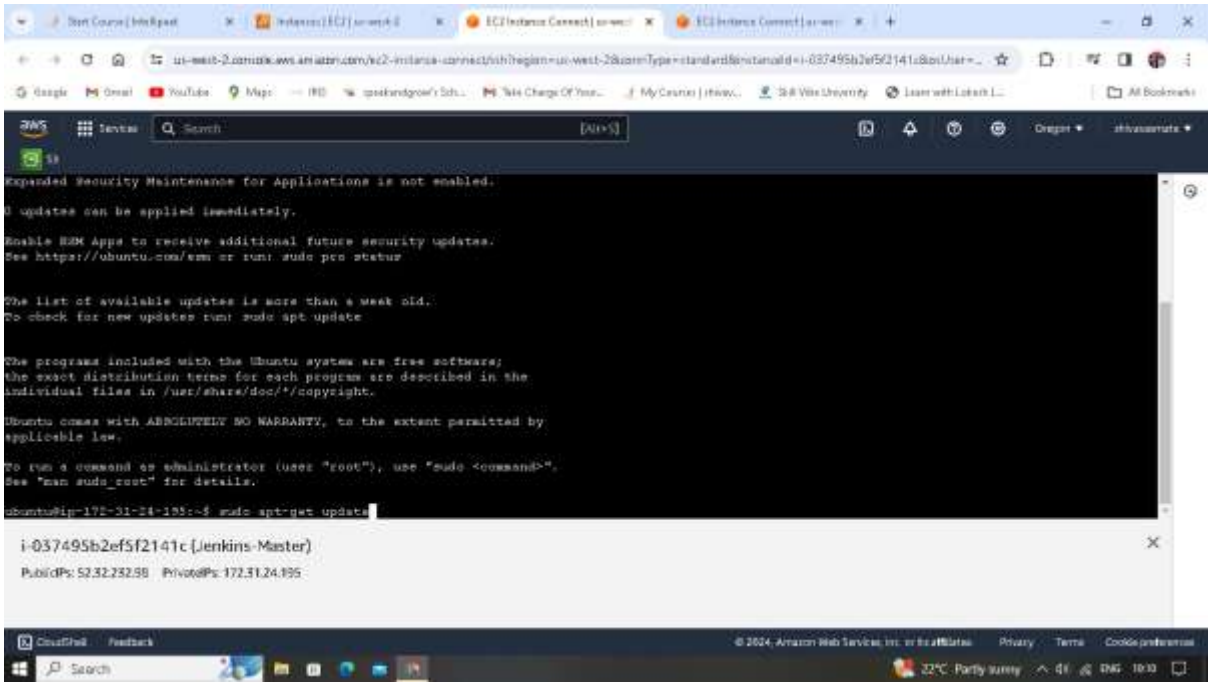
e. VPC and Security group as default with all traffic protocols allowed click on launch instance.



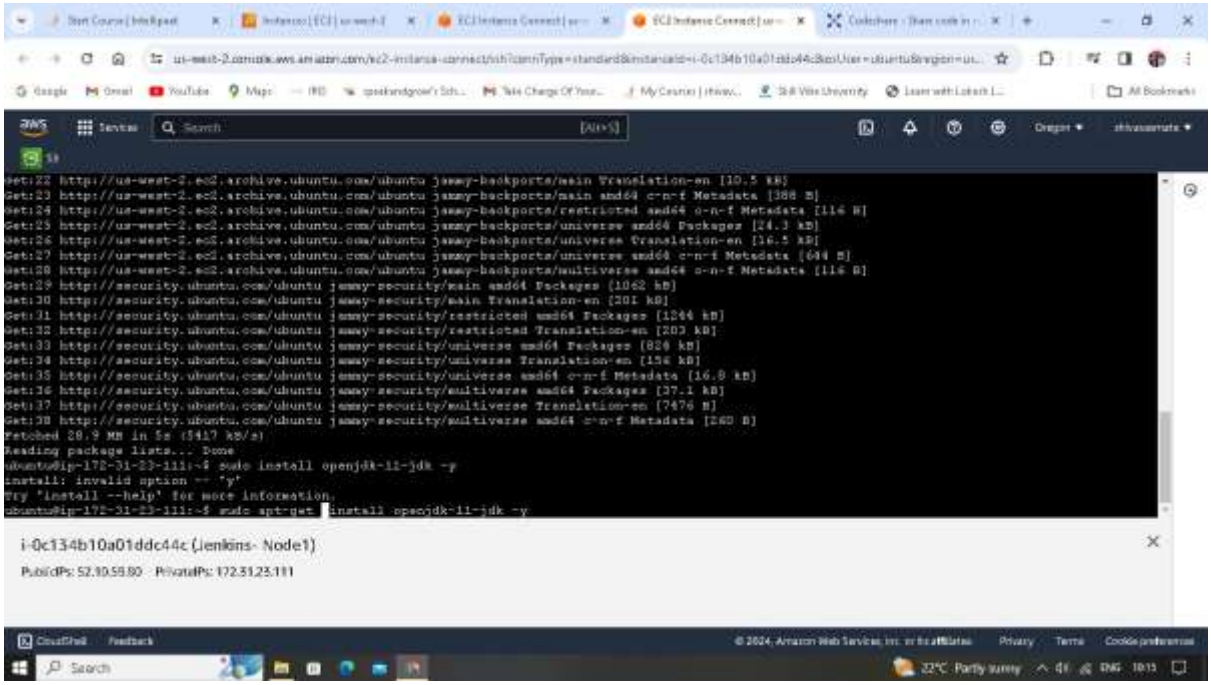
f. Name EC2 instances as Jenkins-Master and Jenkins-Node1 and check both are in running mode.



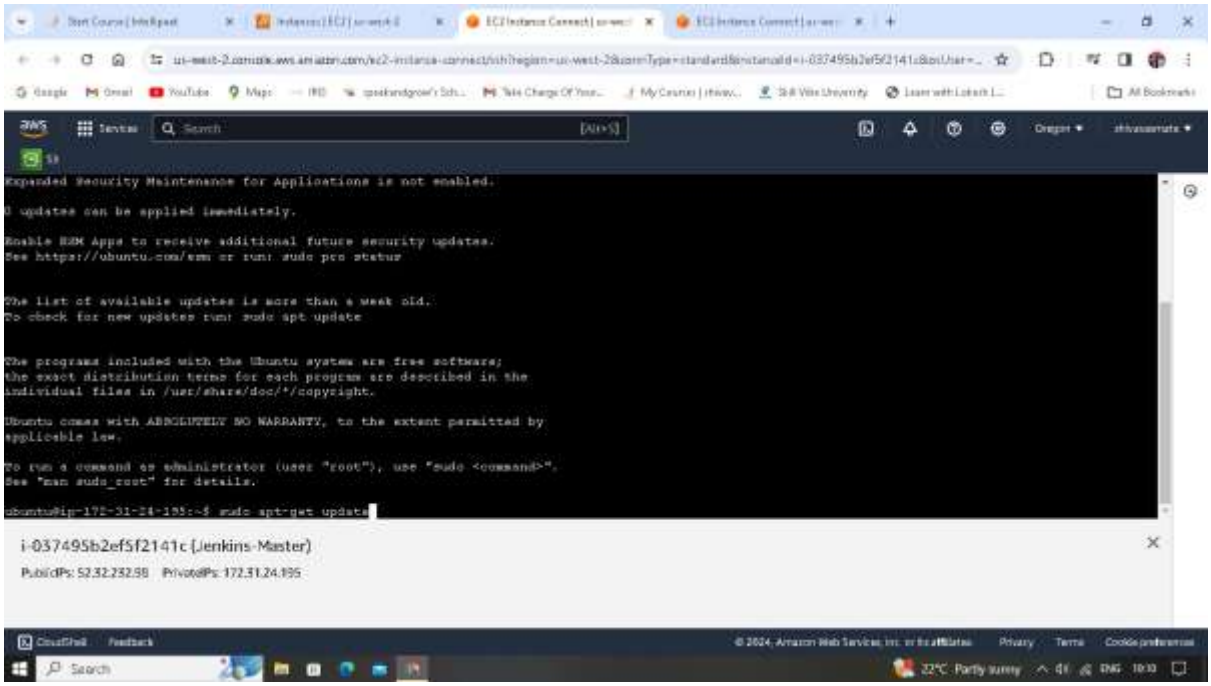
6.Connect Jenkins- Node1 and update it using command sudo apt-get update.

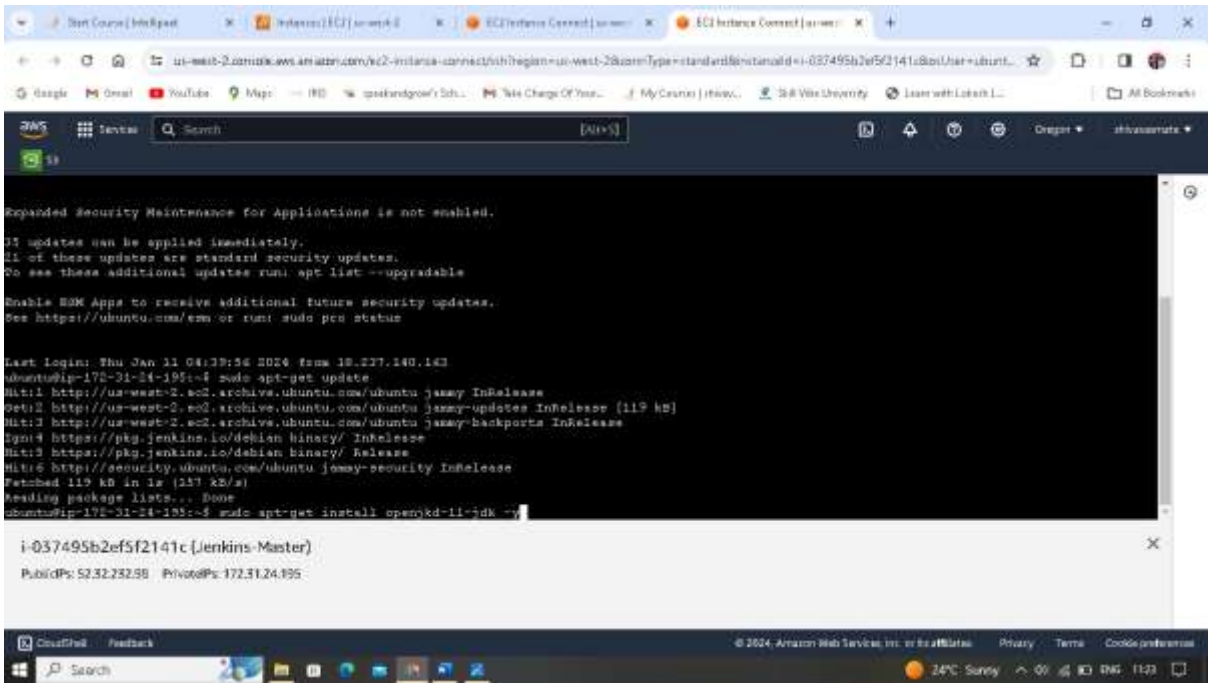


7. Install java using command sudo apt-get install openjdk-11-jkd -y



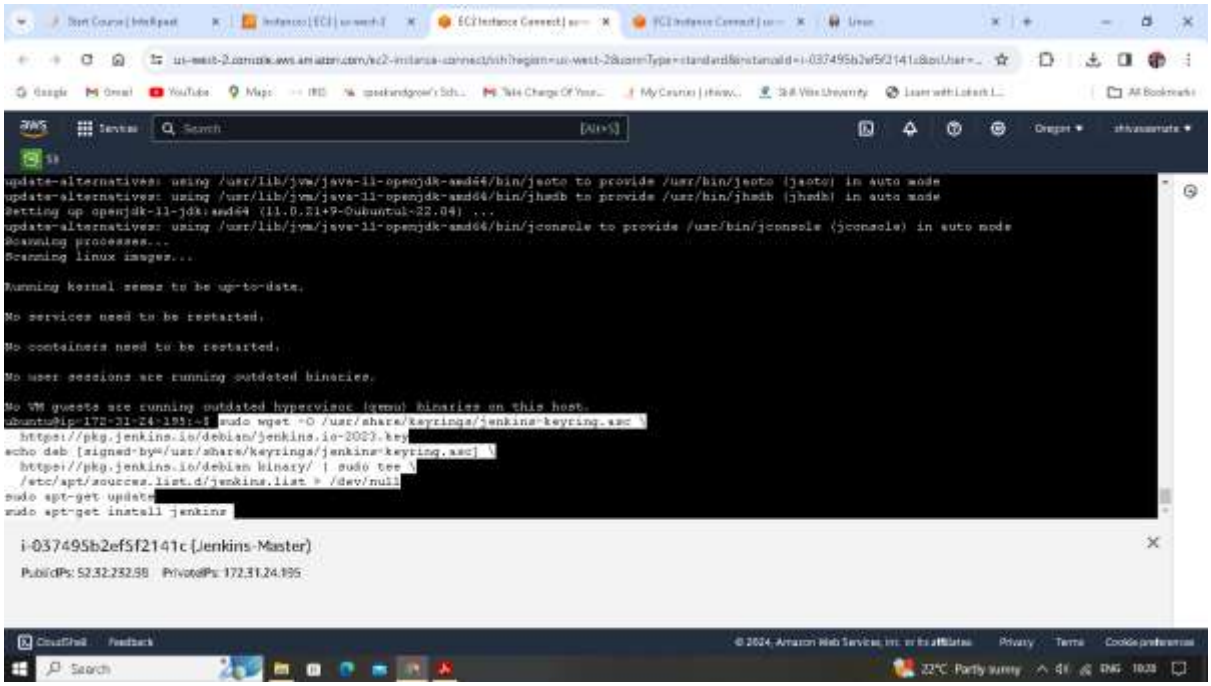
8. Similarly on Jenkins-Master also update and install java.

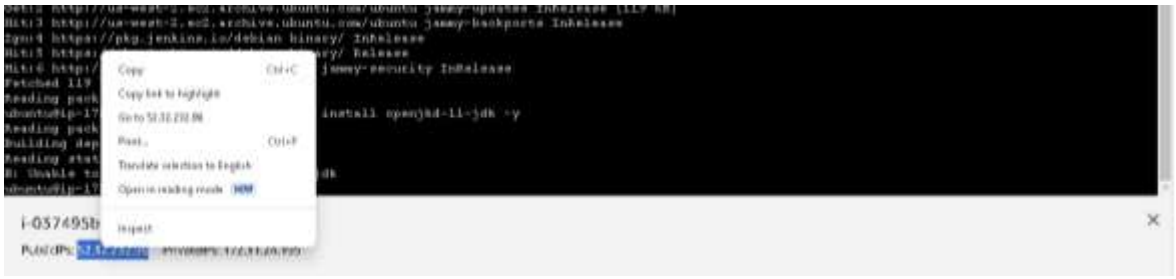




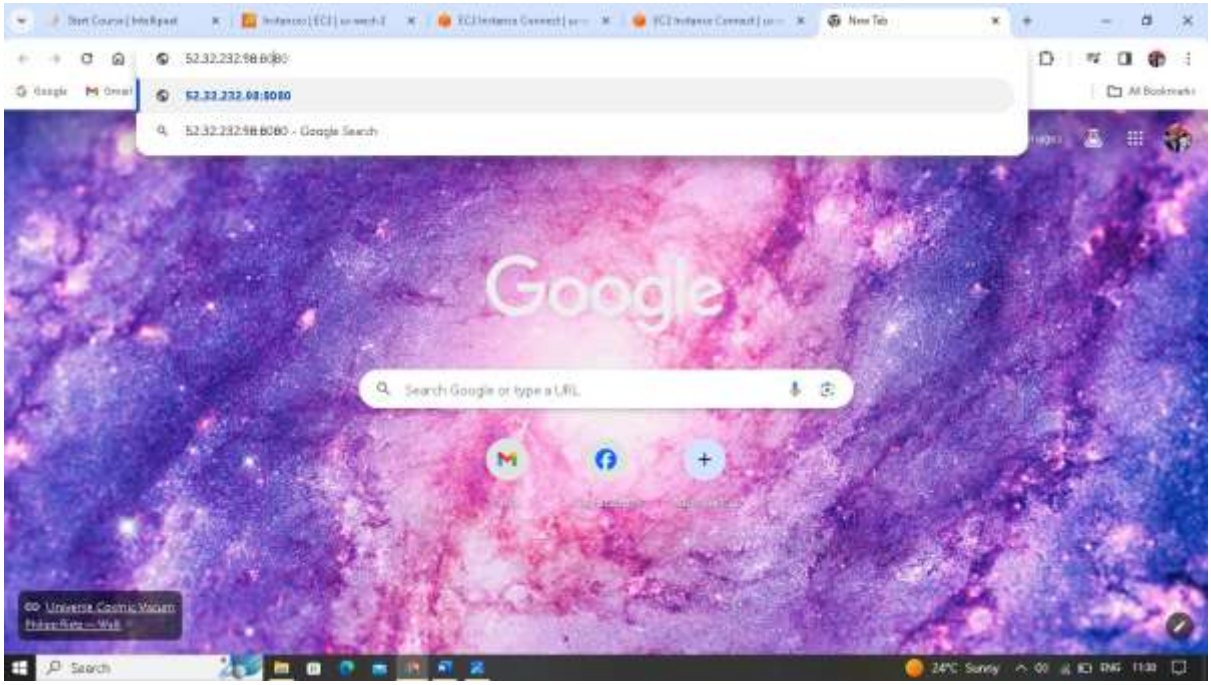
9. Only install Jenkins on the Jenkins-Master using command

```
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins
```





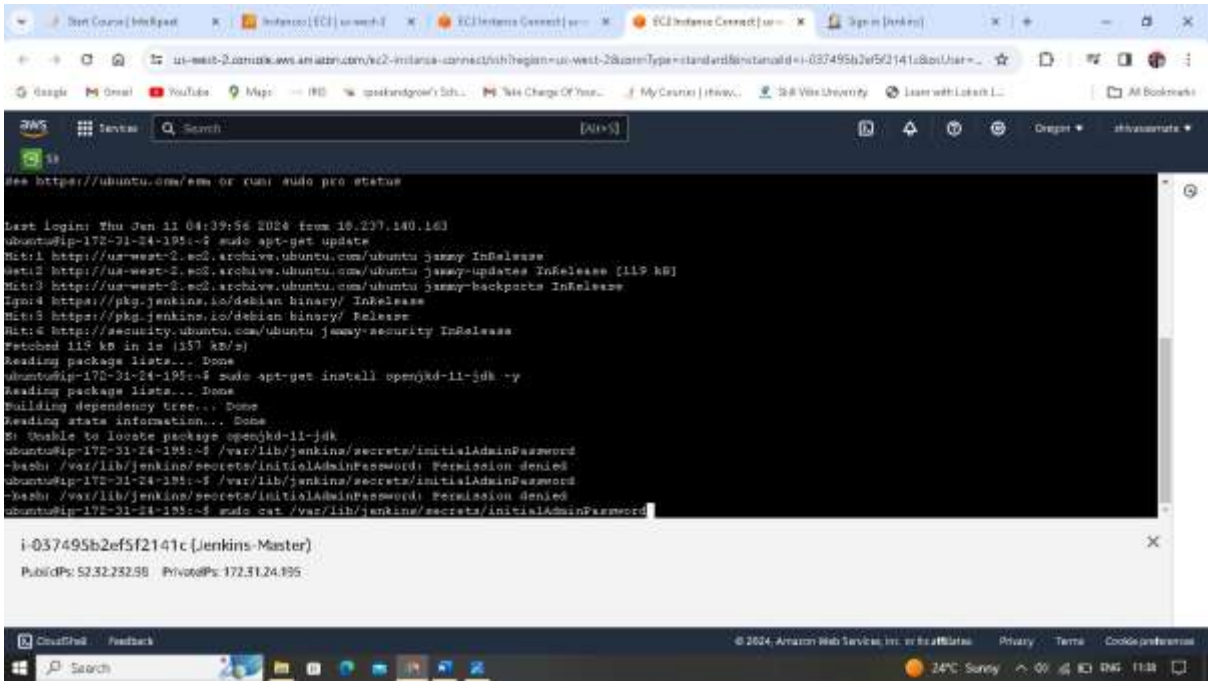
10. Open new browser copy and paste public IP with :8080 port.



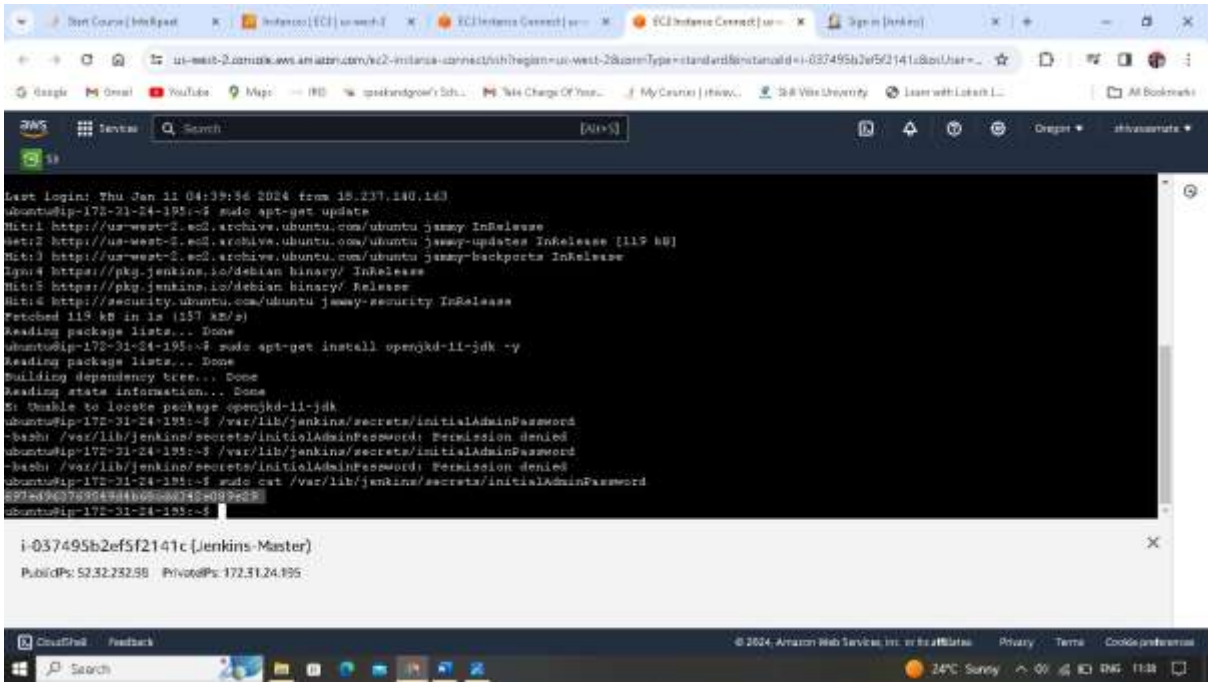
11. Jenkins sign-in page opens.

12 copy `/var/lib/jenkins/secrets/initialAdminPassword`

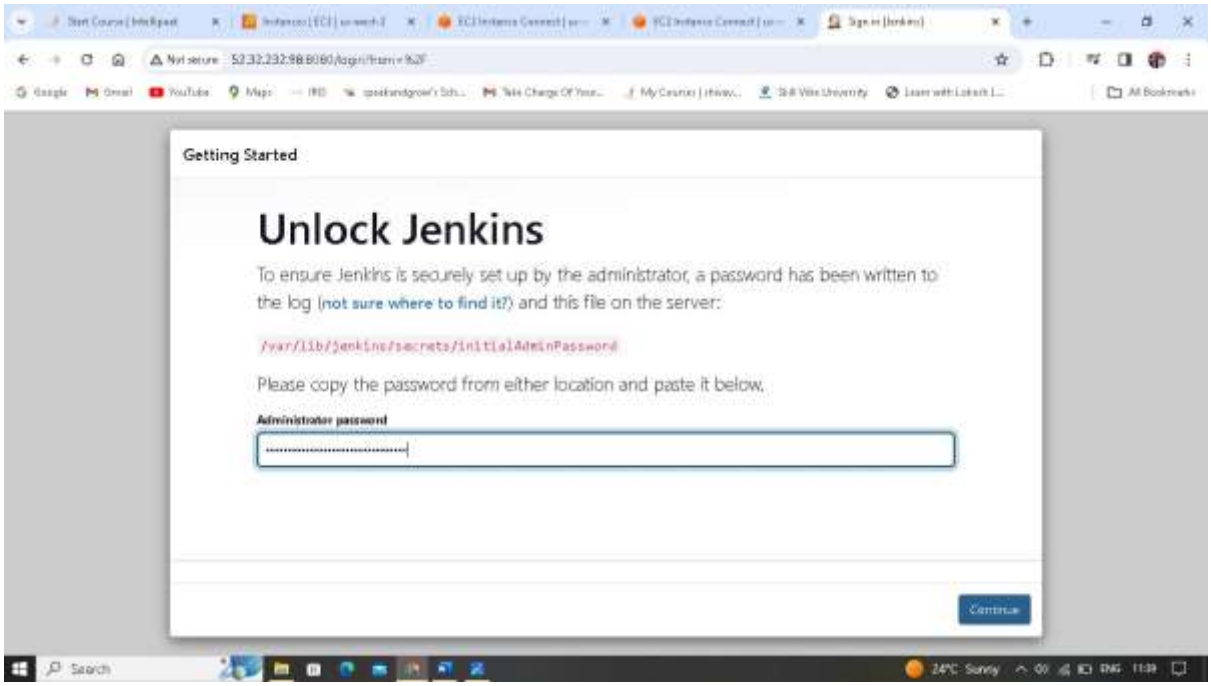
`Go to Jenkins-Master using command sudo cat paste it.`



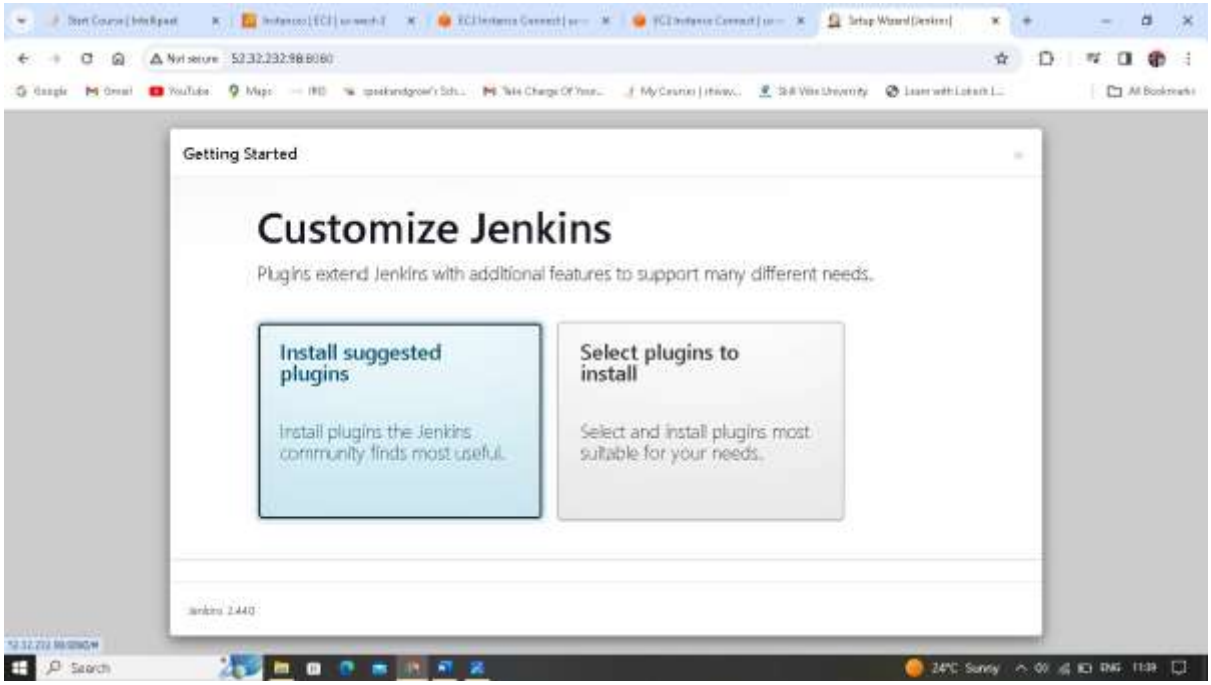
13. Hit enter to get password copy it.



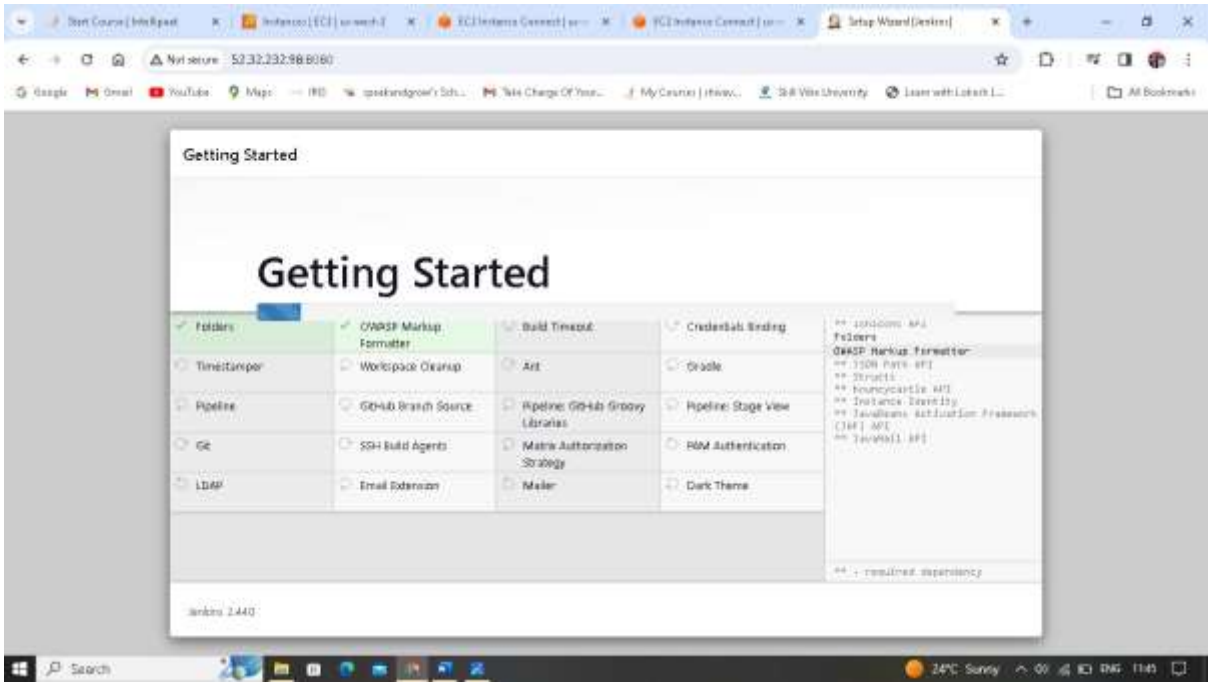
14. Paste the password and click on continue



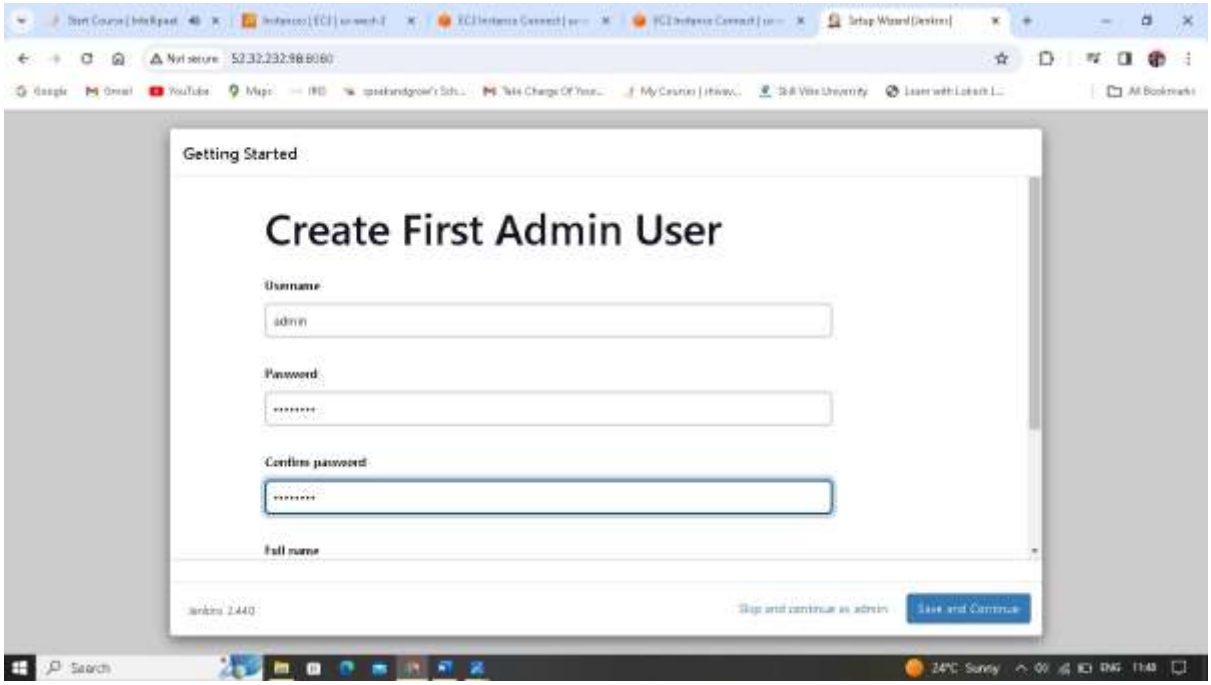
15. Click on install suggested plugins



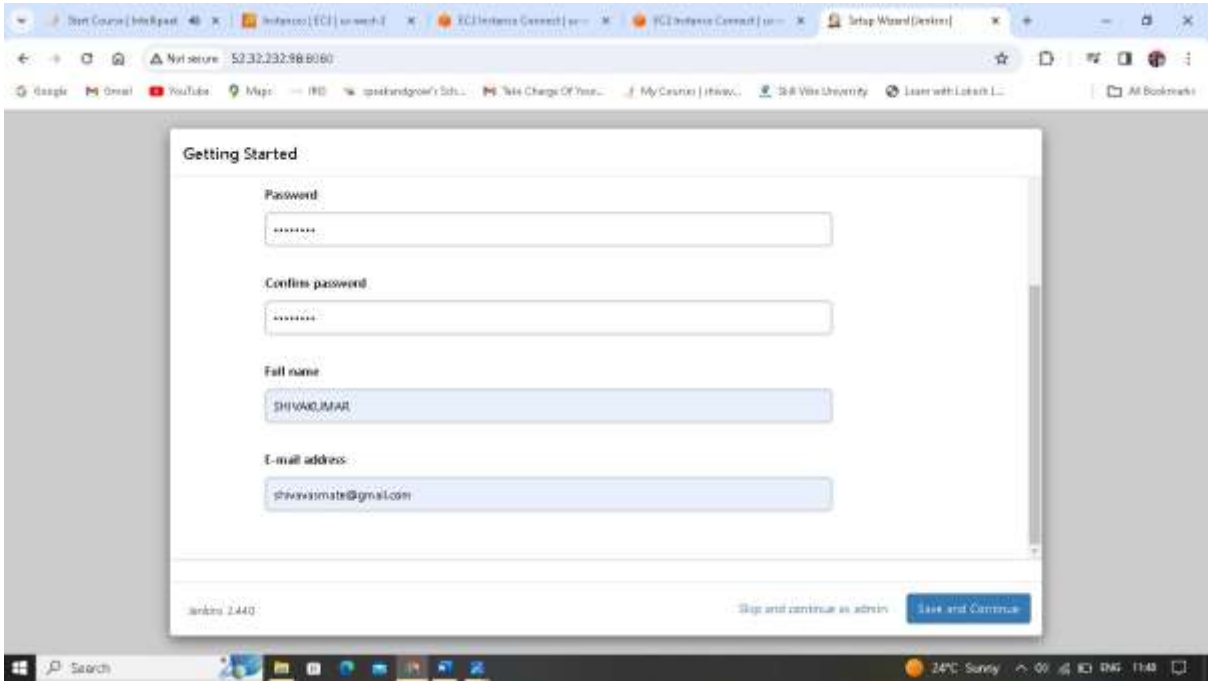
16. Just wait for all the plugins to install



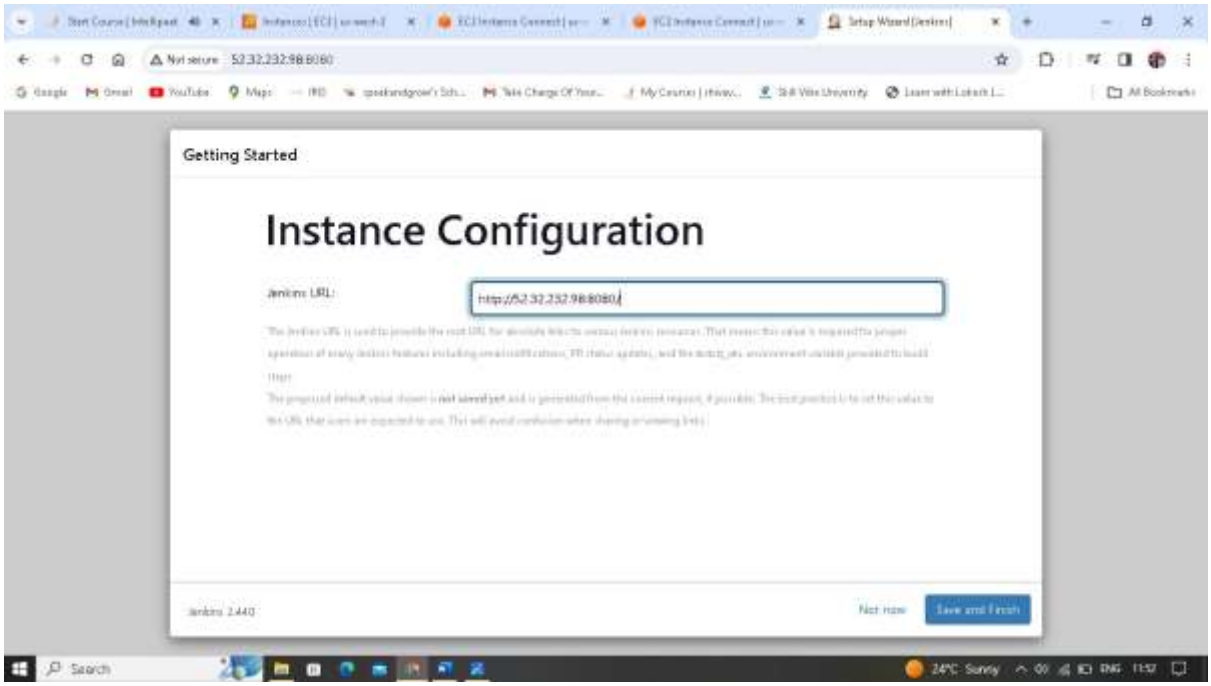
17. Now create first admin user.

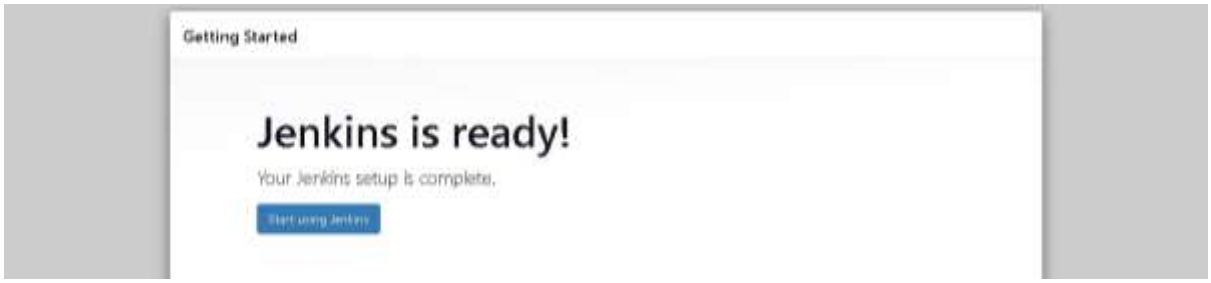


18. Give user name, password, full name and email ID and click save and continue.

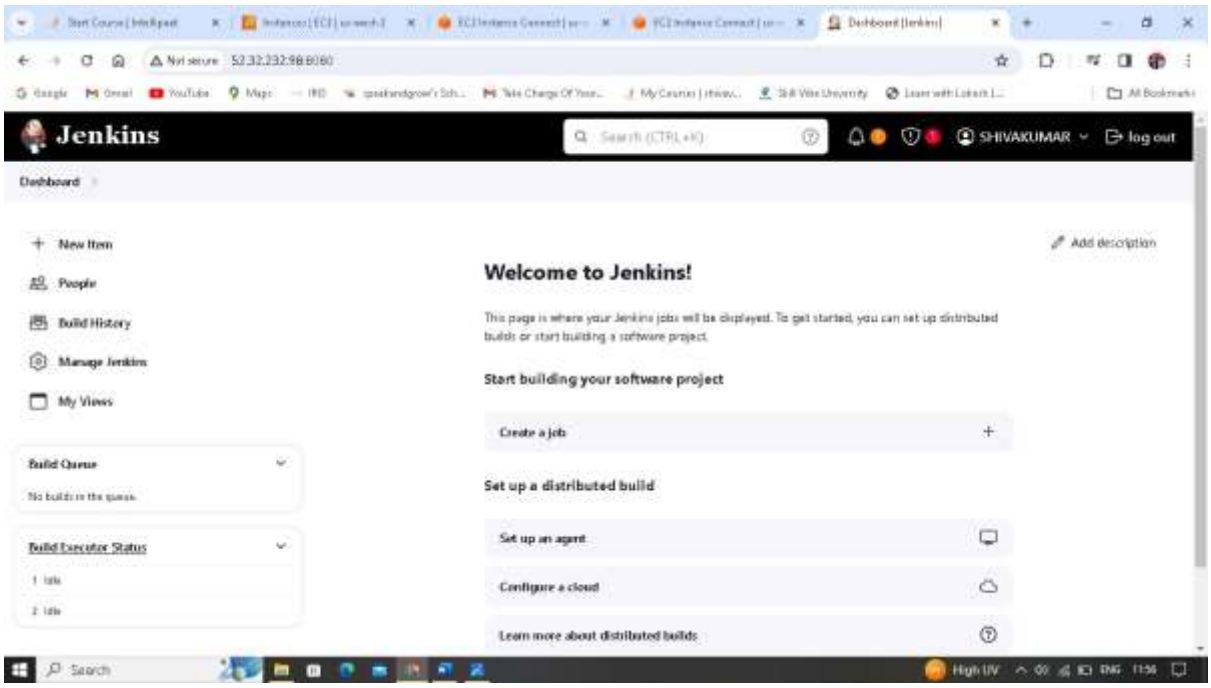


19 clicks save and finish





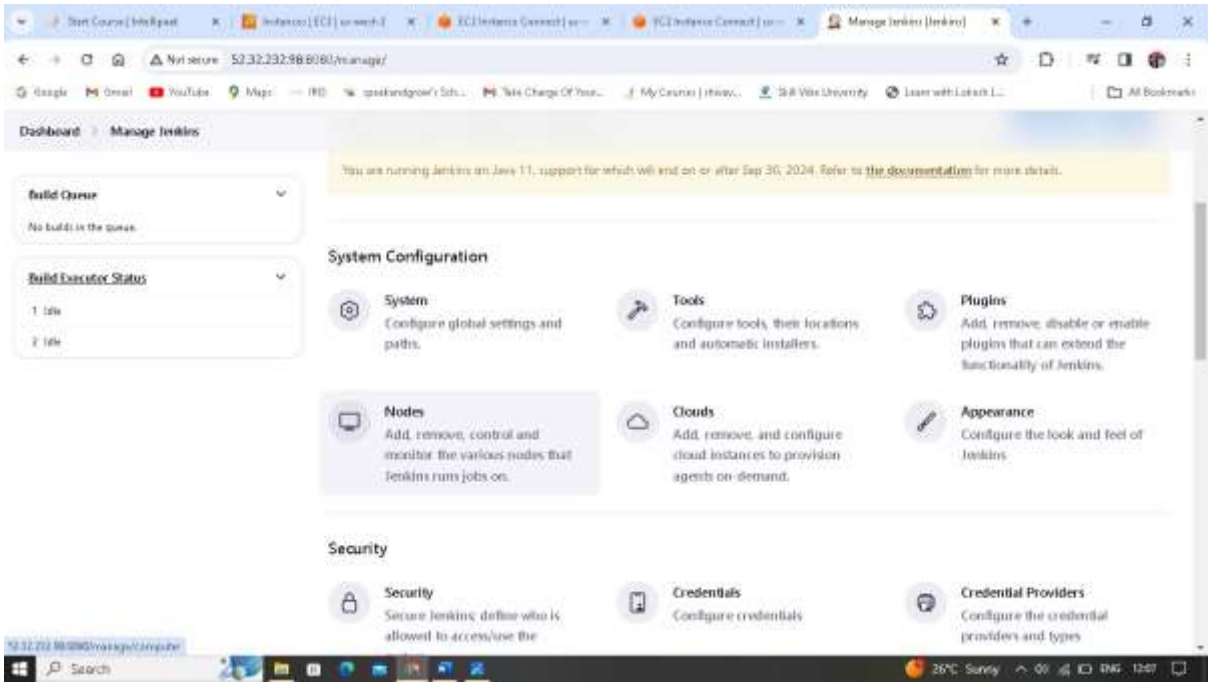
20. With this Jenkins dashboard is setup.



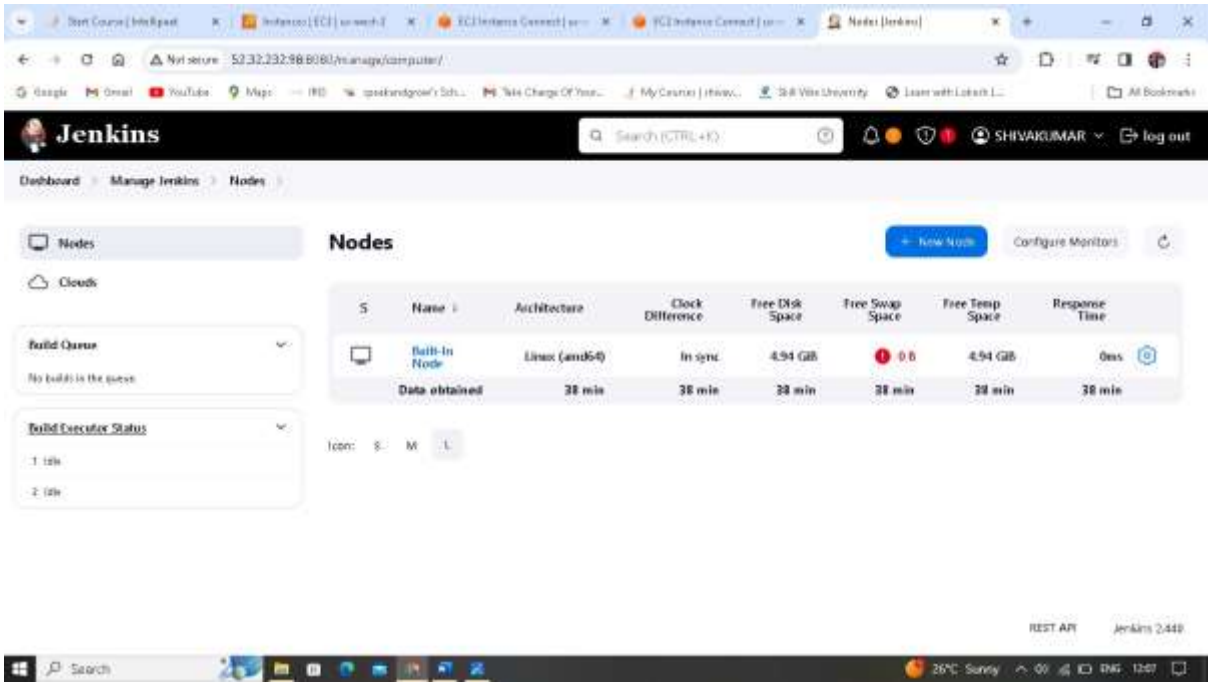
21. So far command line used.

22. Now connect Jenkins-Node1. Click on Manage Jenkins

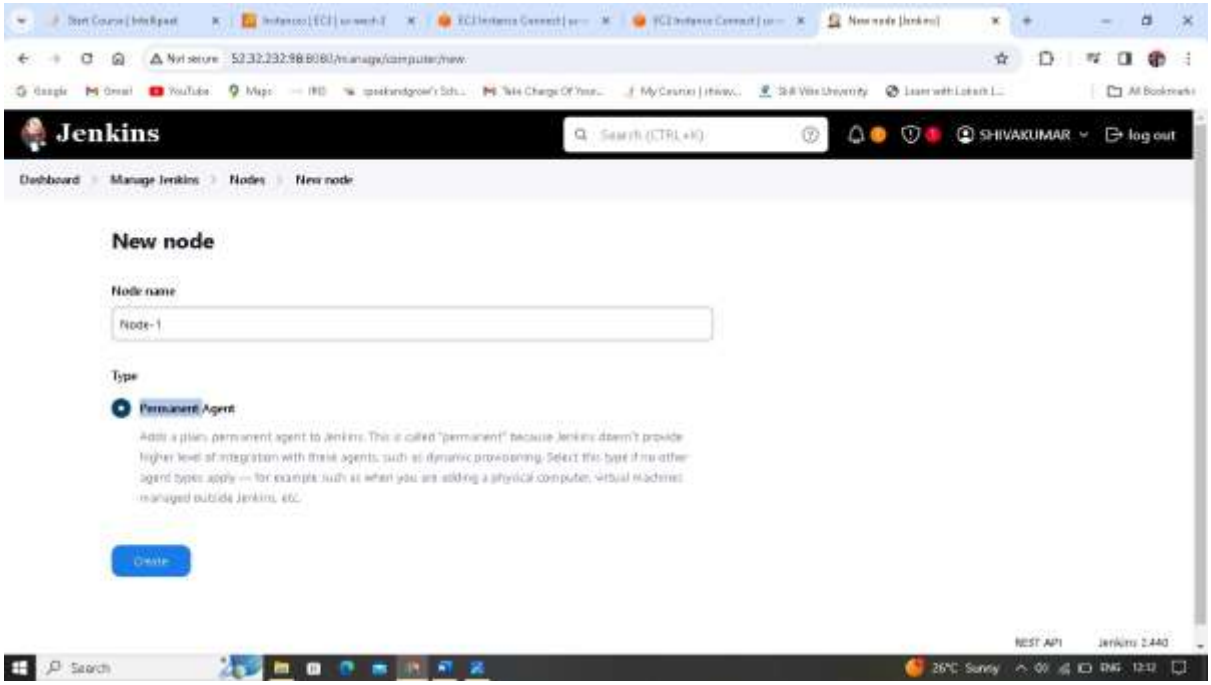
23. Go to system configuration and click on Nodes.



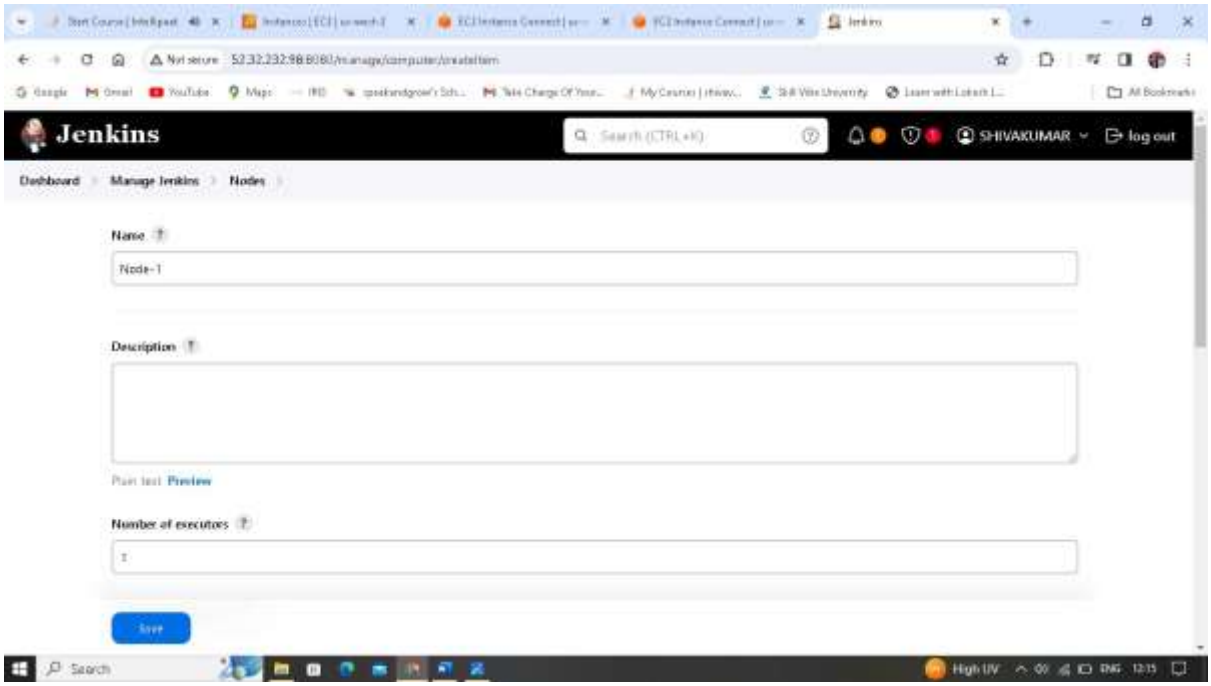
24. Click on +New Node



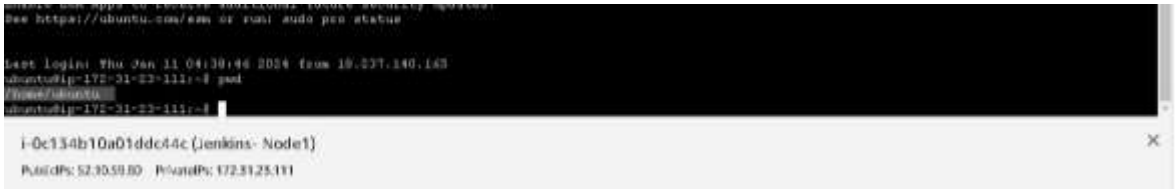
25. Create a new node with a name node-1 as a type permanent agent.



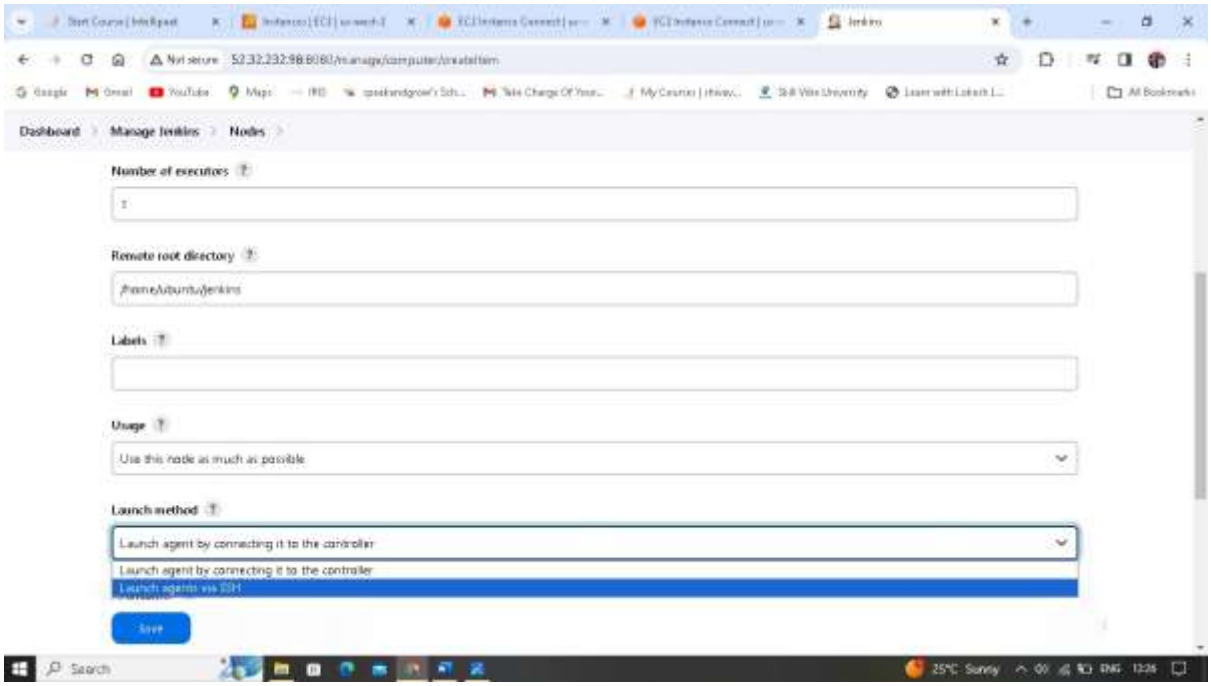
26. Now fill



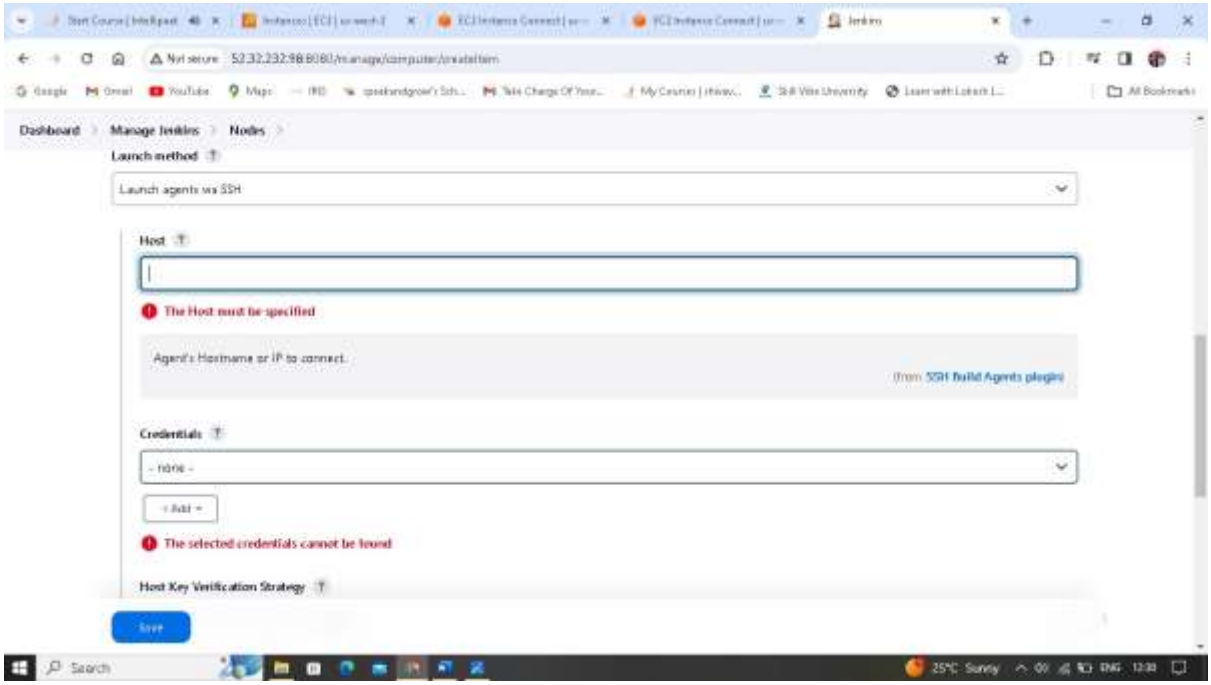
27. First of all give the Root directory. As to connect Jenkins-Node1 see the pwd (/home/ubuntu)



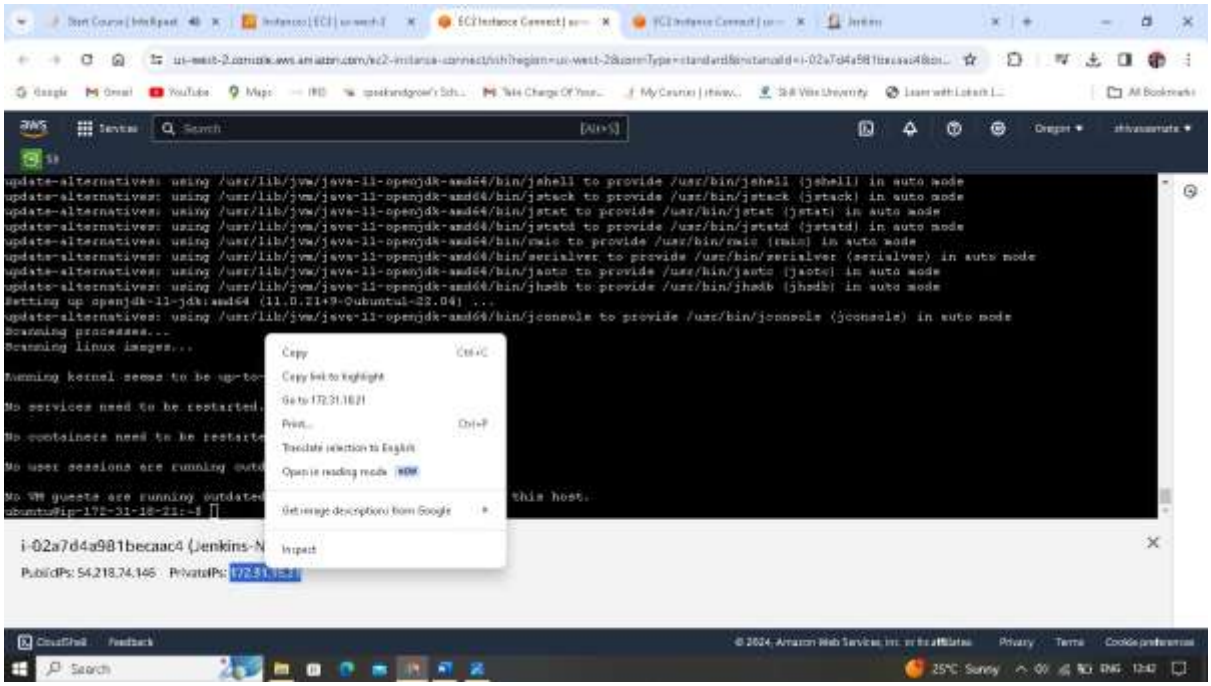
28. /home/ubuntu/jenkins as root directory.



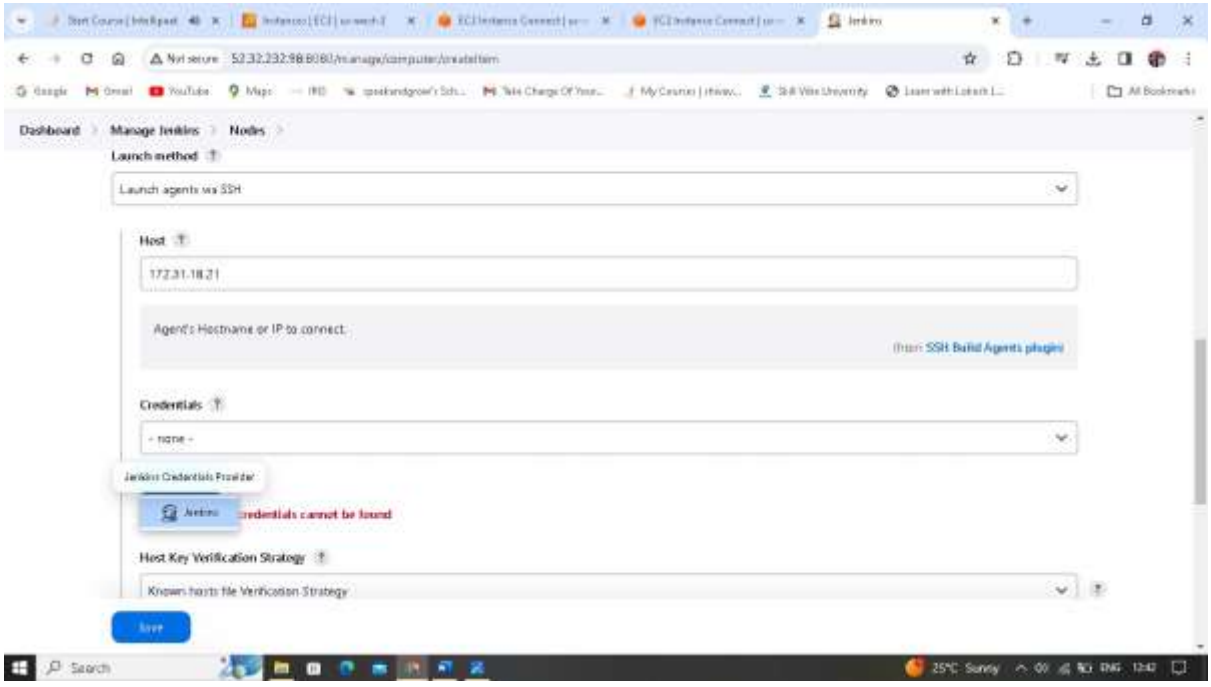
29. In Launch method select agents via SSH.



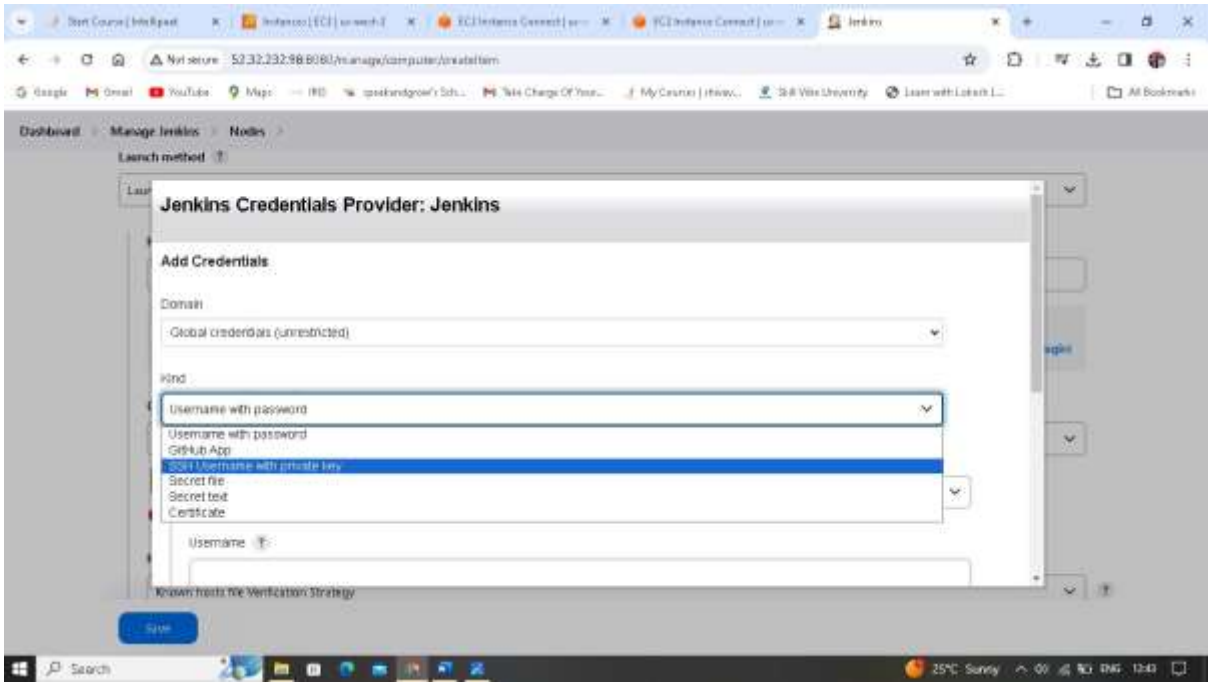
30. For Host copy and paste the private IP of Jenkins-Node1



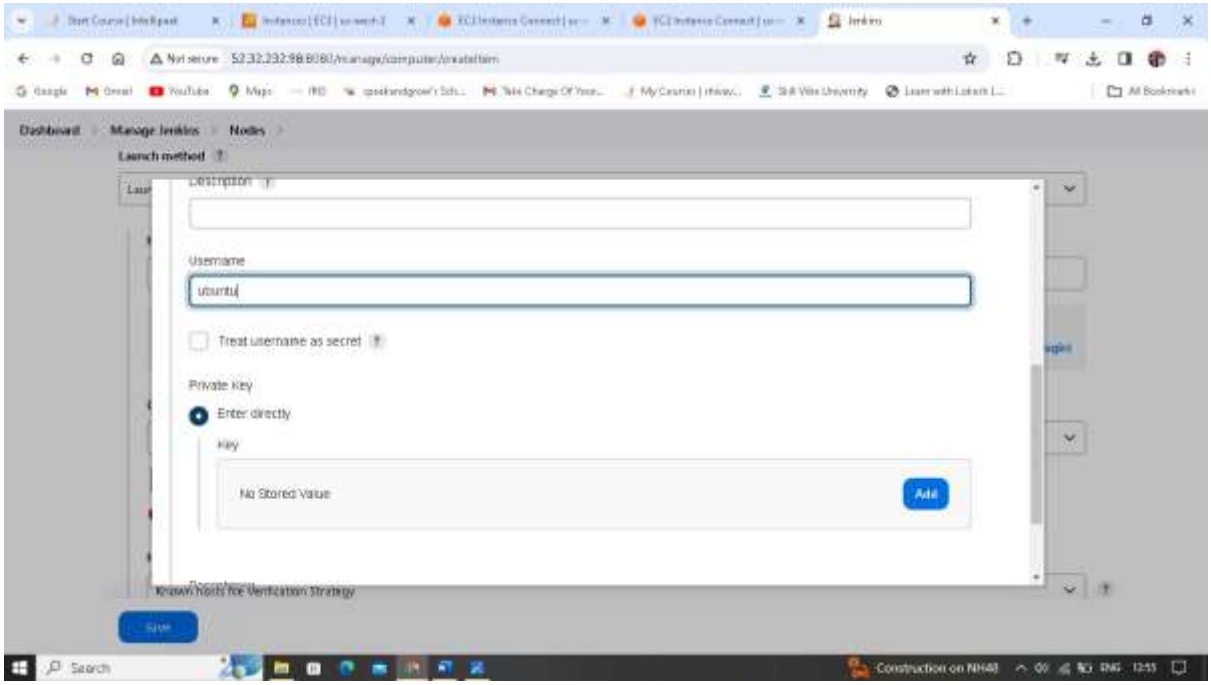
31. In the credentials click on add jenkins



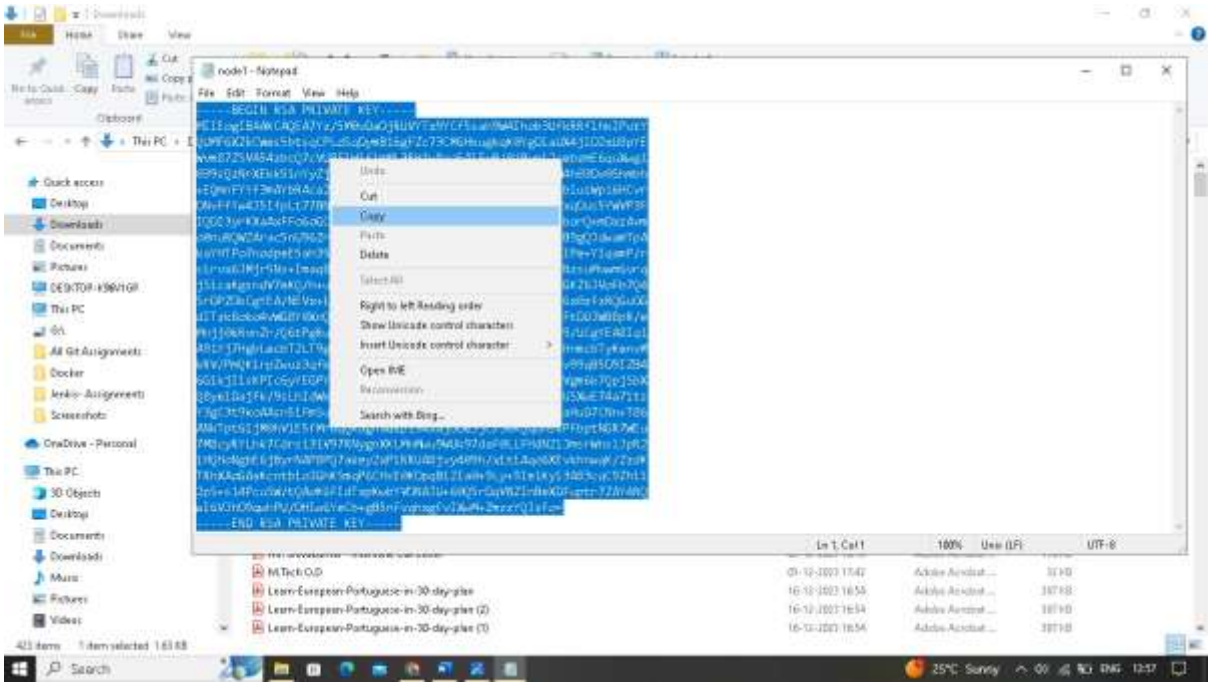
32. In kind select SSH Username with private key



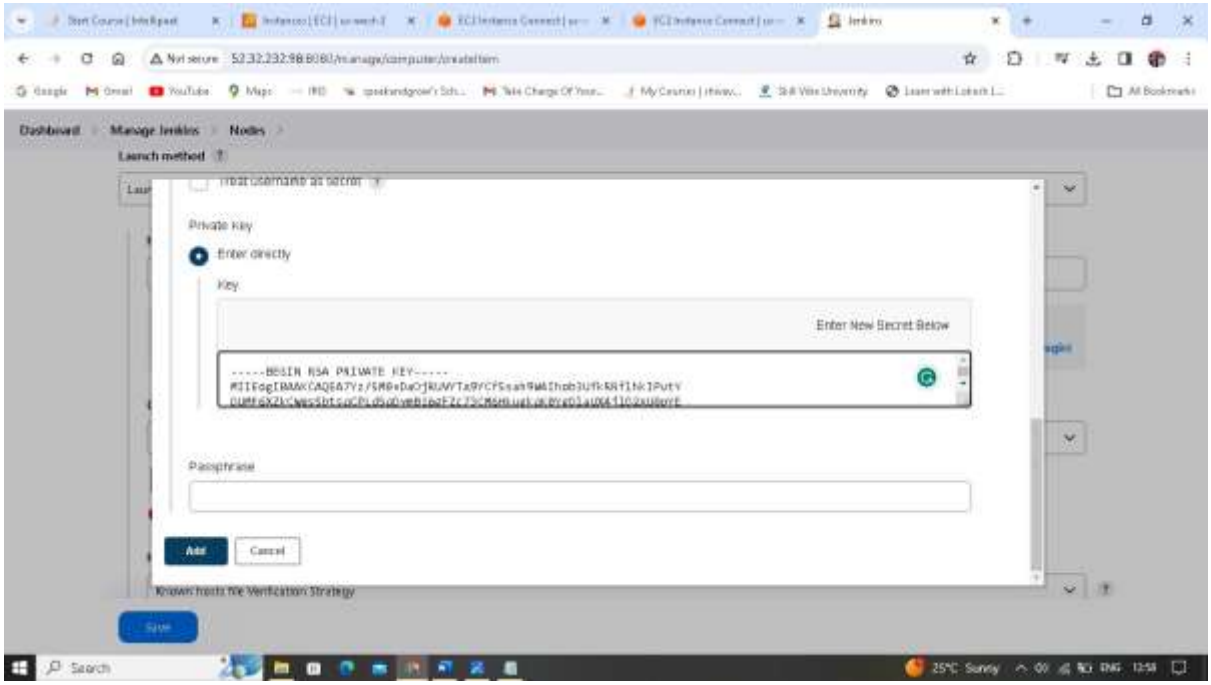
33. Give username as ubuntu click on enter directory and add



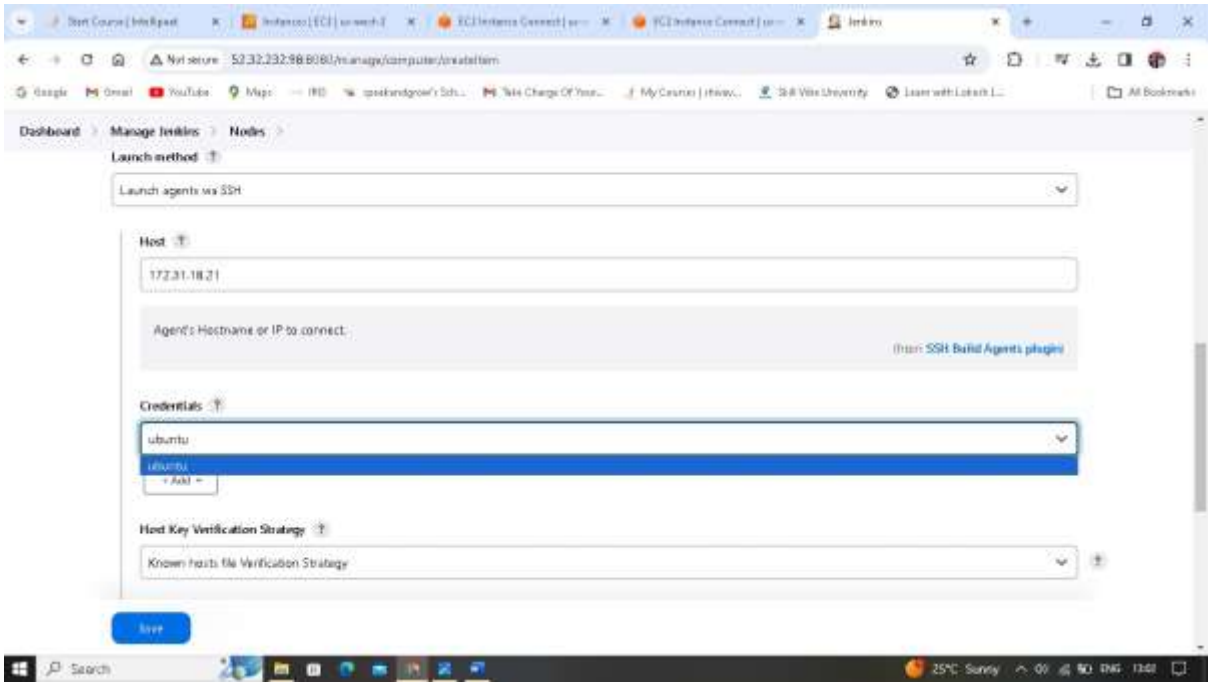
34. Copy the Jenkins-Node1 .pem file



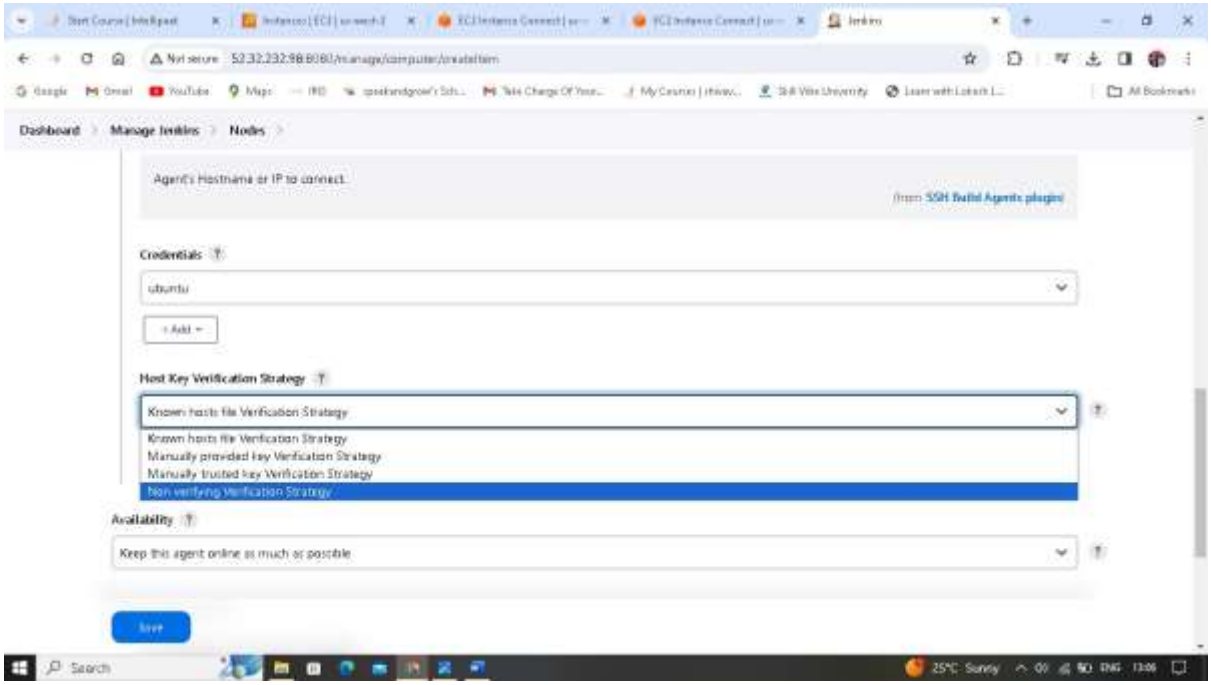
35. Paste the private key and click on add



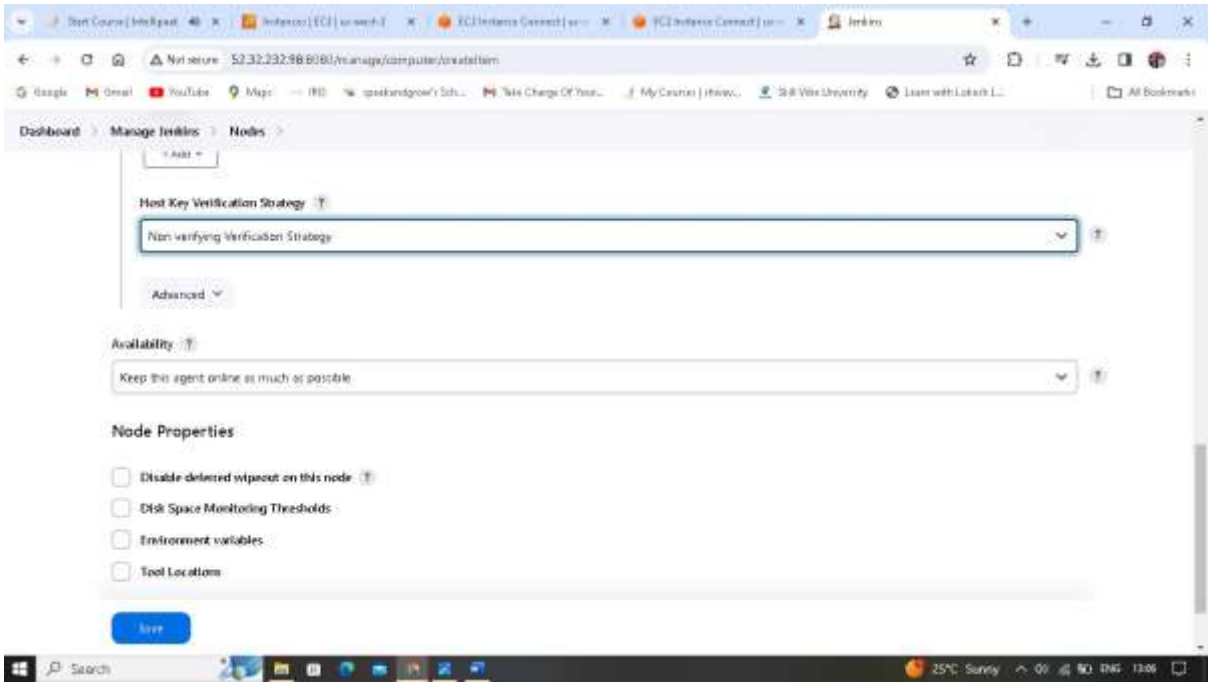
36. Now in the credentials select ubuntu



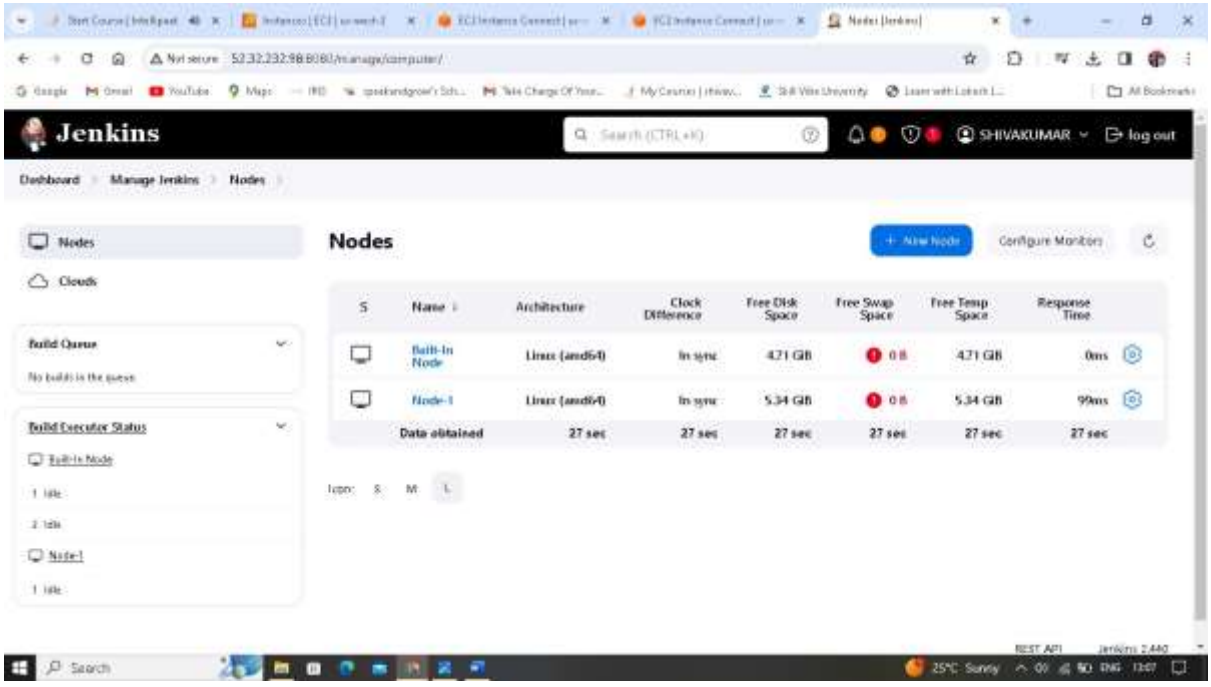
37. Select Non-verification strategy



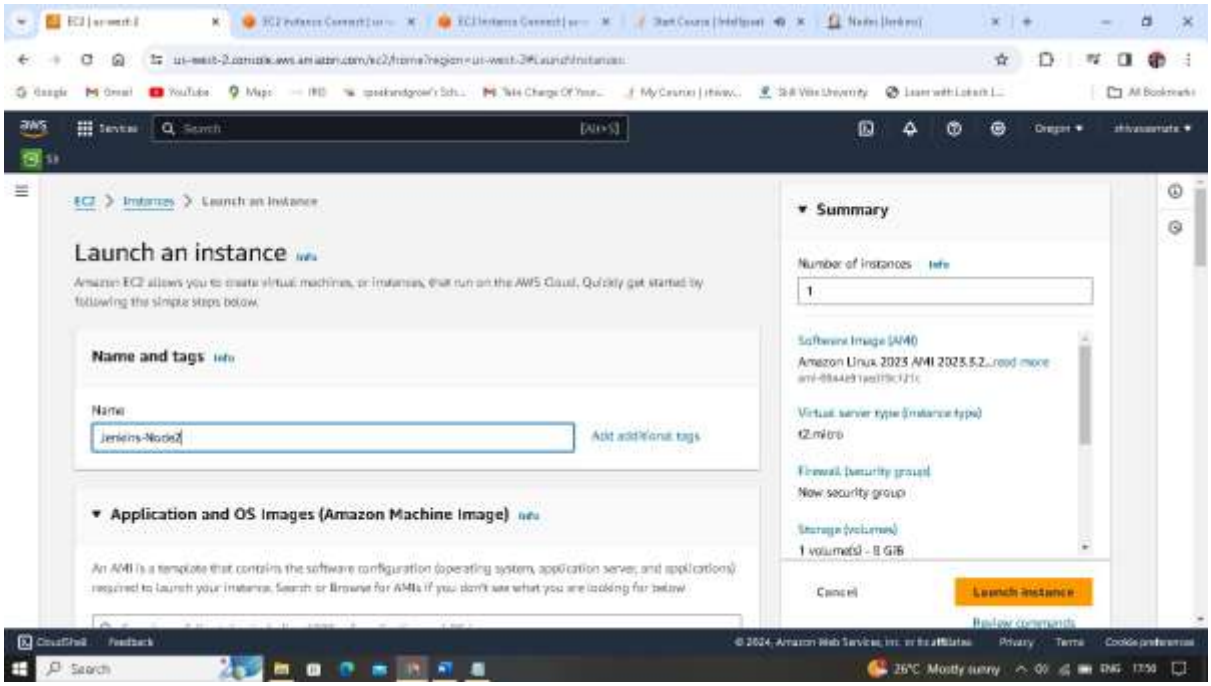
38. Click on save



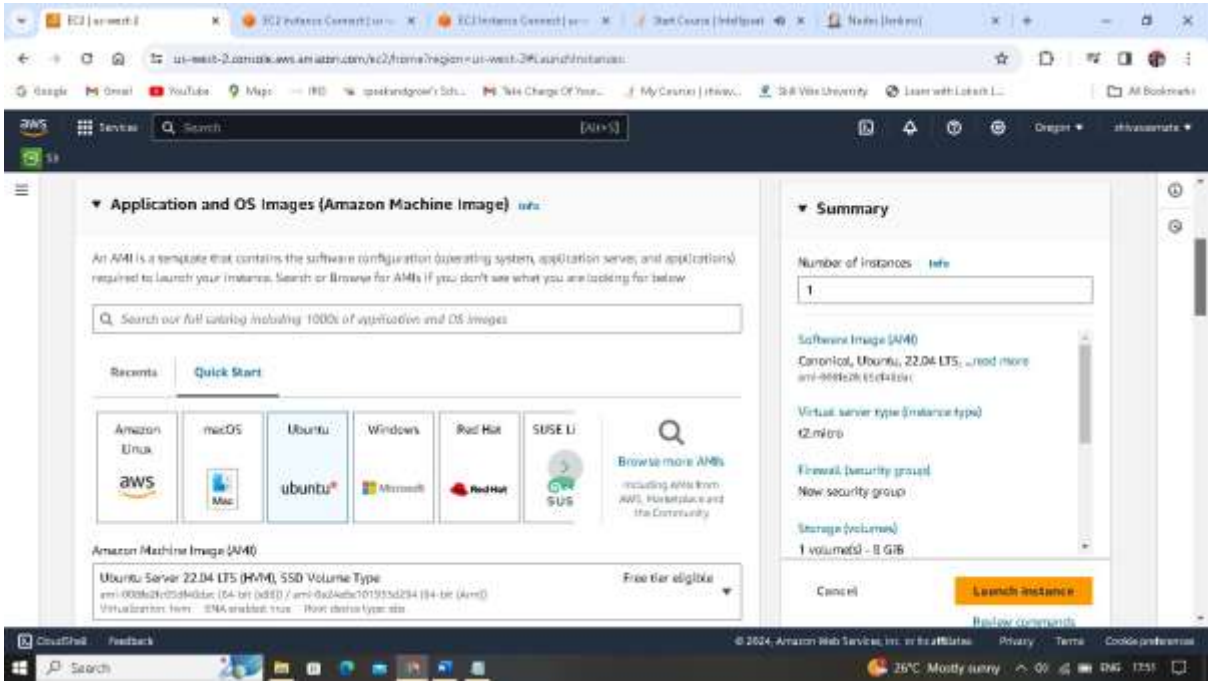
39. Successfully added Node-1

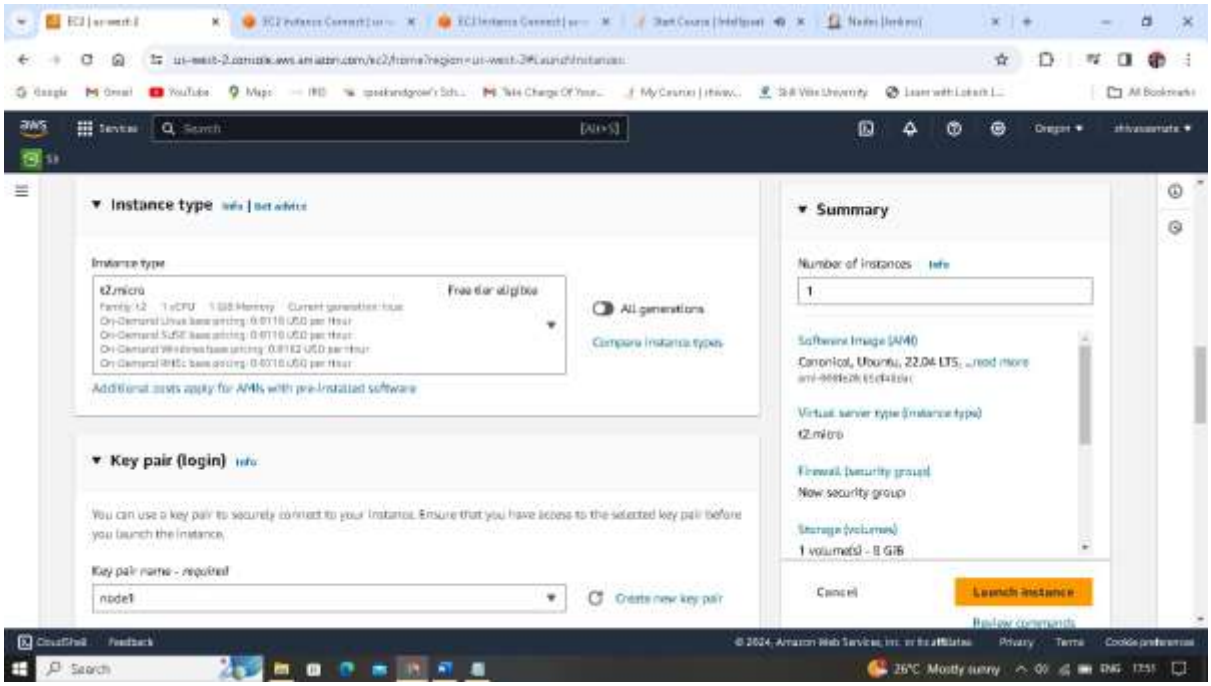


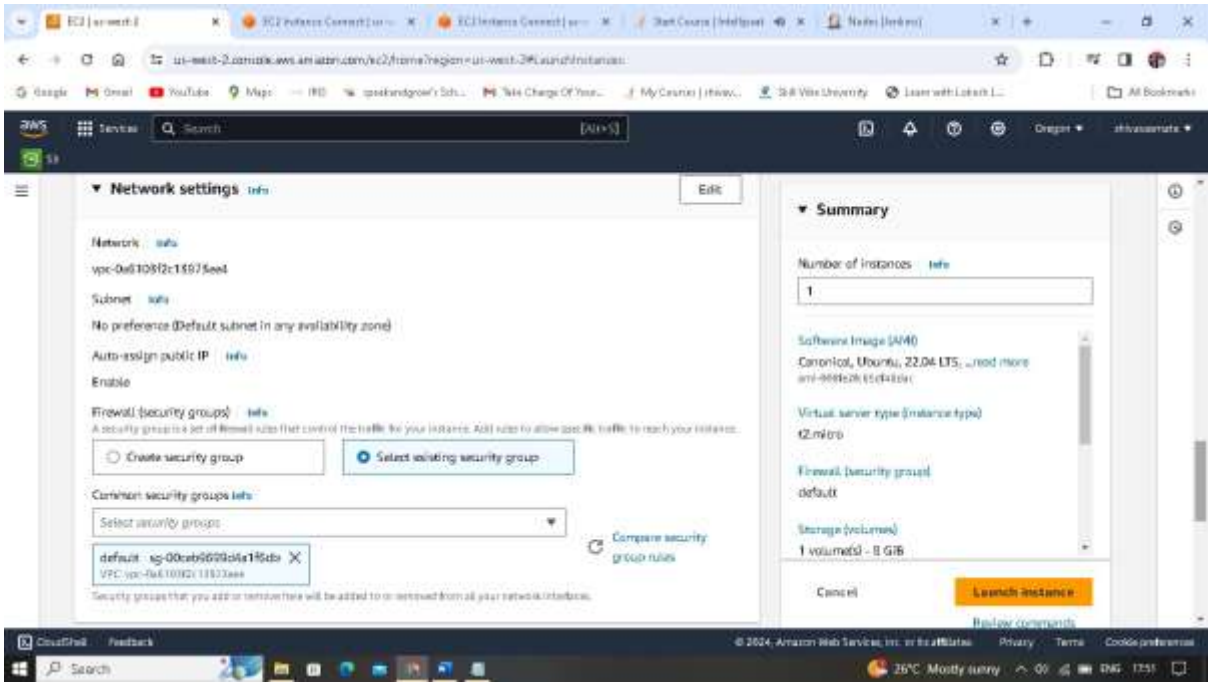
40. Create one more same instance. With a name Jenkins-node2.



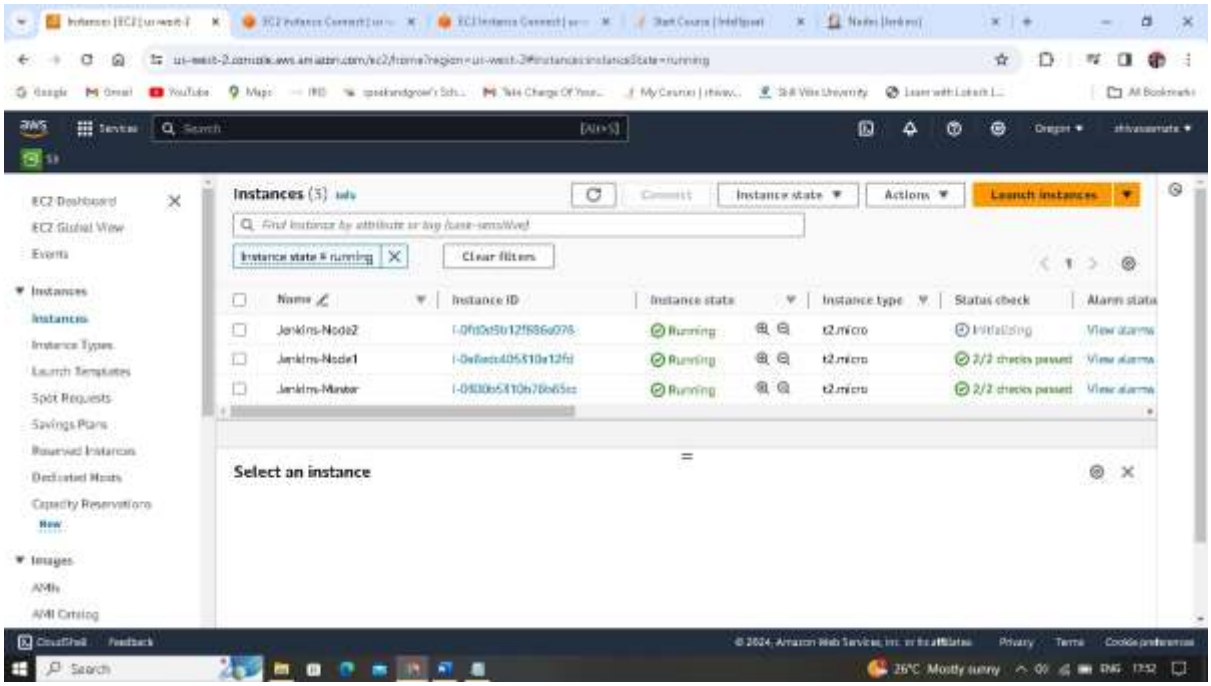
41. Keeping all other setting as same as Jenkins-Node-1



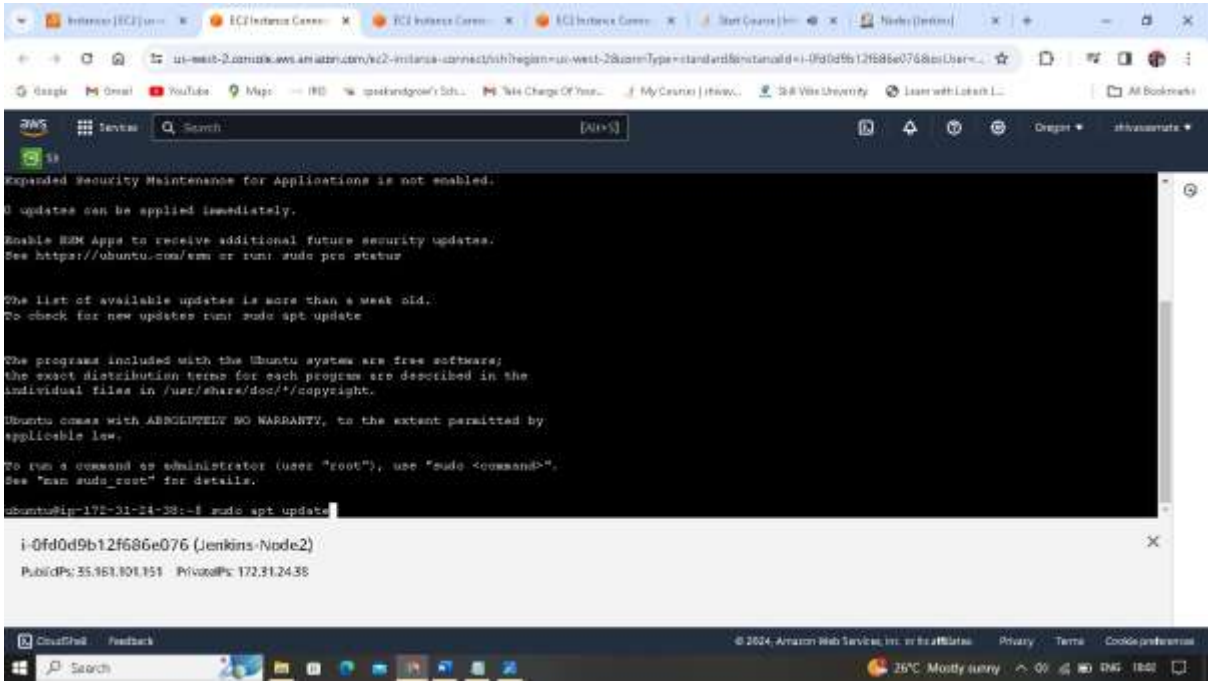

Instance type
Summary
Number of instances
Key pair (login)
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
node1
Create new key pair
Cancel
Launch instance


Network settings
Edit
Summary
Number of instances
Auto-assign public IP
Enable
Create security group
Select existing security group
Common security groups
Select security groups
Compare security group rules
Cancel
Launch instance

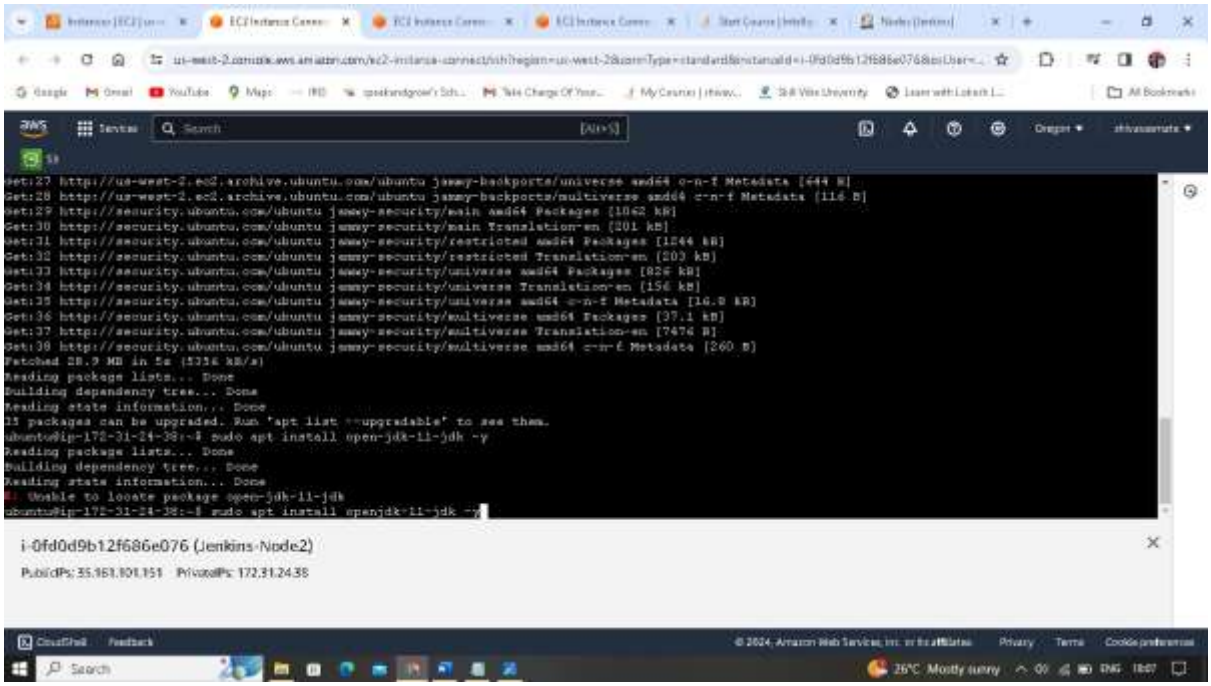
42. Now Jenkins-Node2 is also running status.



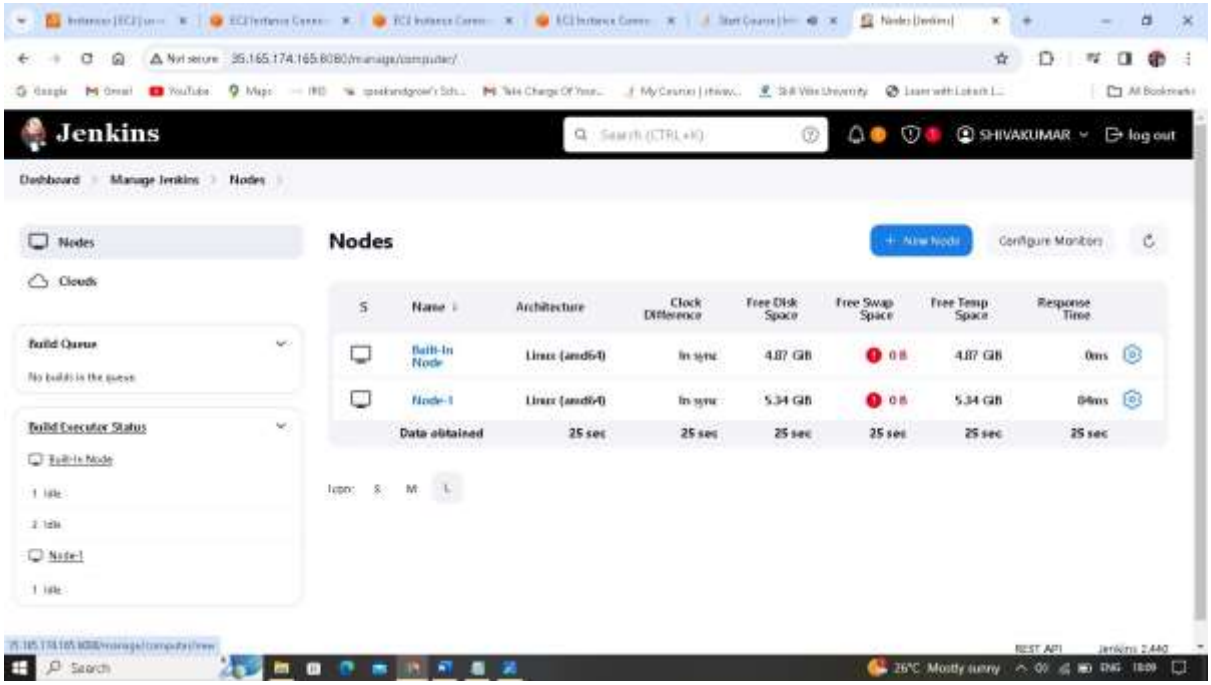
43. Connect Jenkins-Node2. and run command sudo apt update.



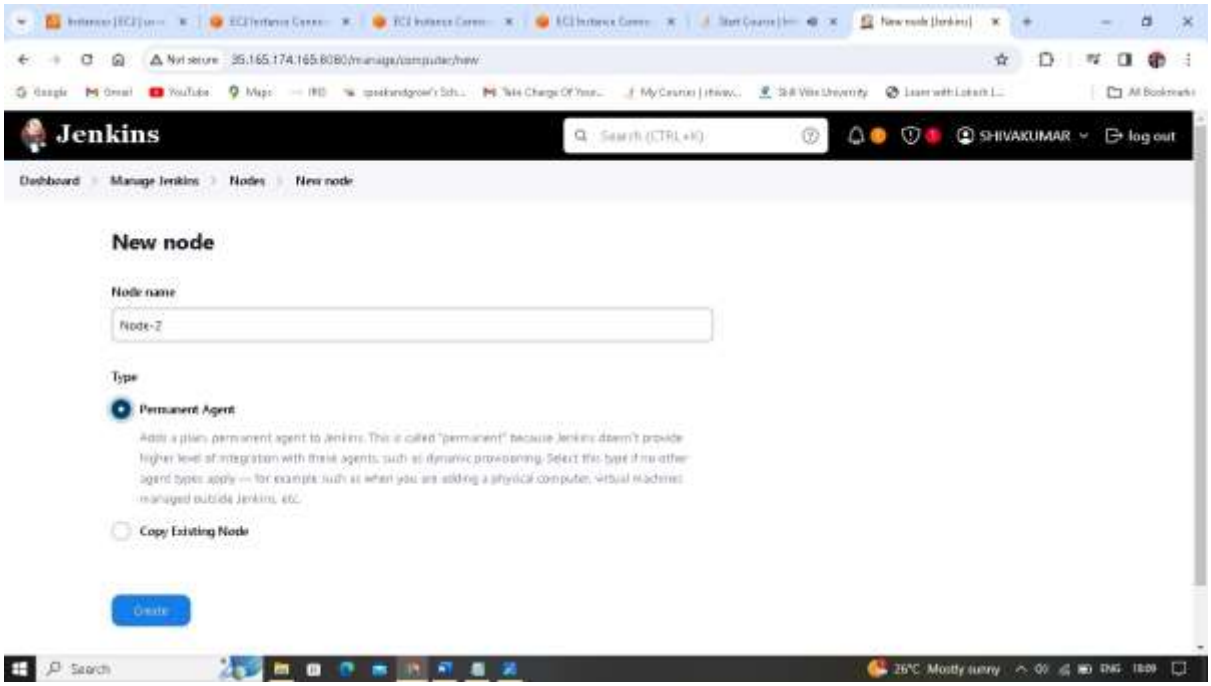
44. Install java by using sudo apt install openjdk-11-jdk -y



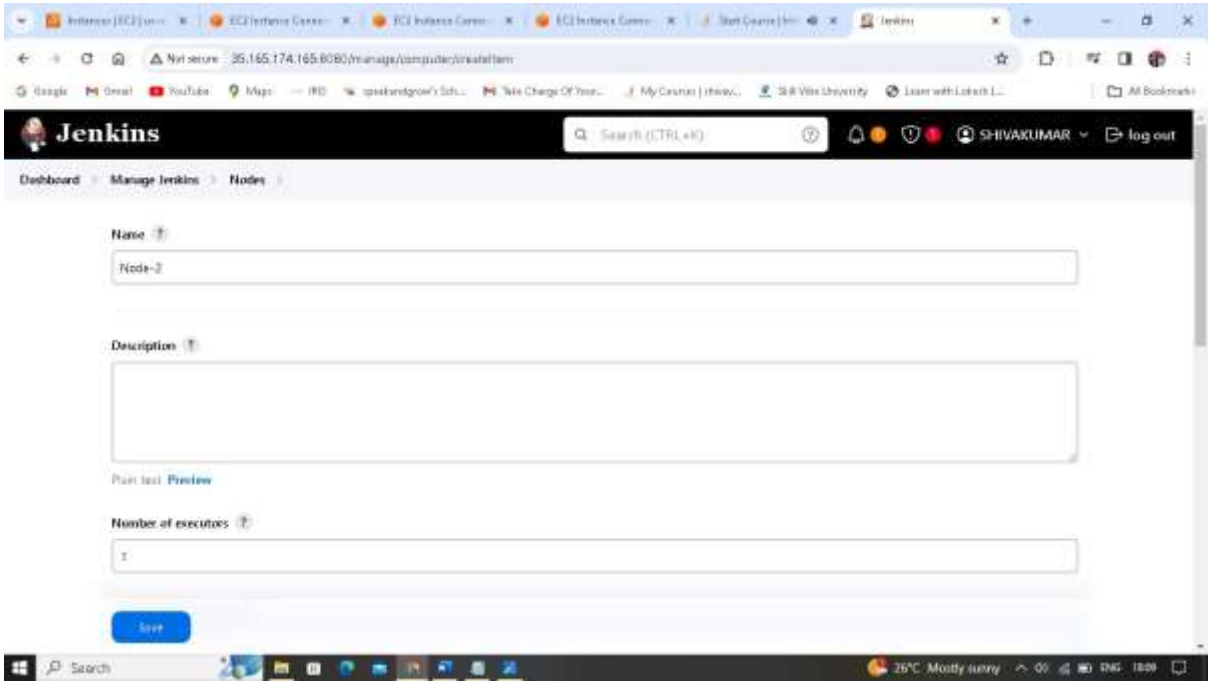
45. Go to Jenkins dashboard click on +New Node



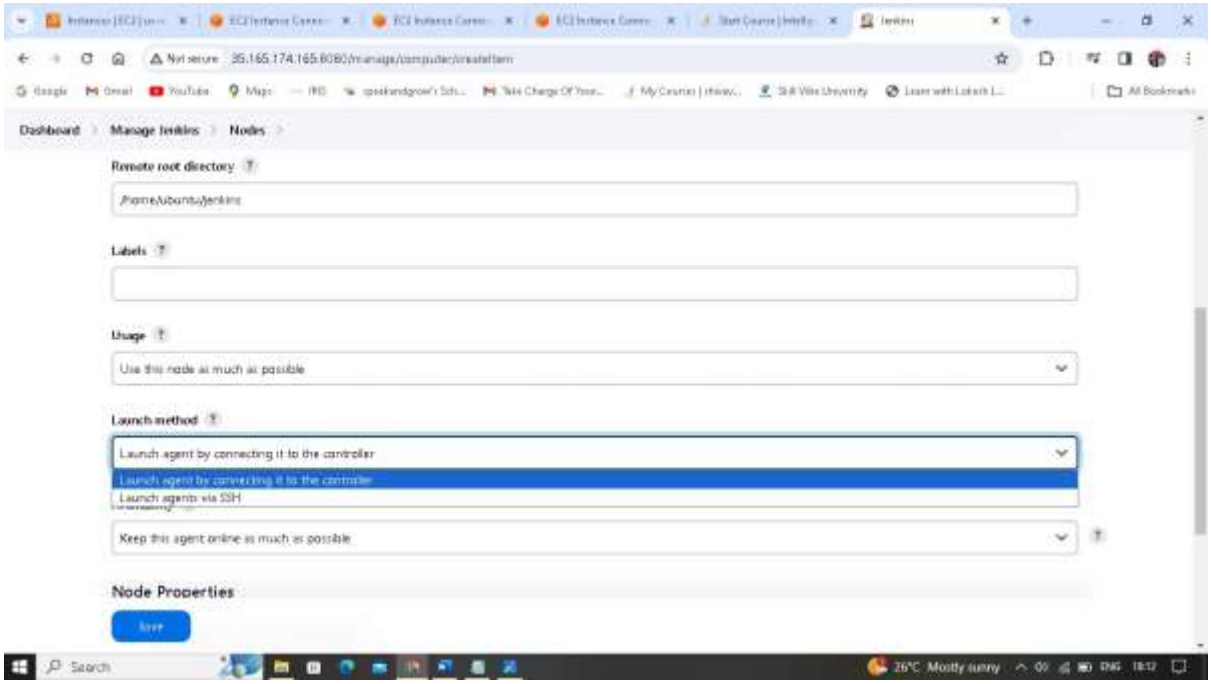
46. Create a new node with a name Node-2 as a type permanent agent.



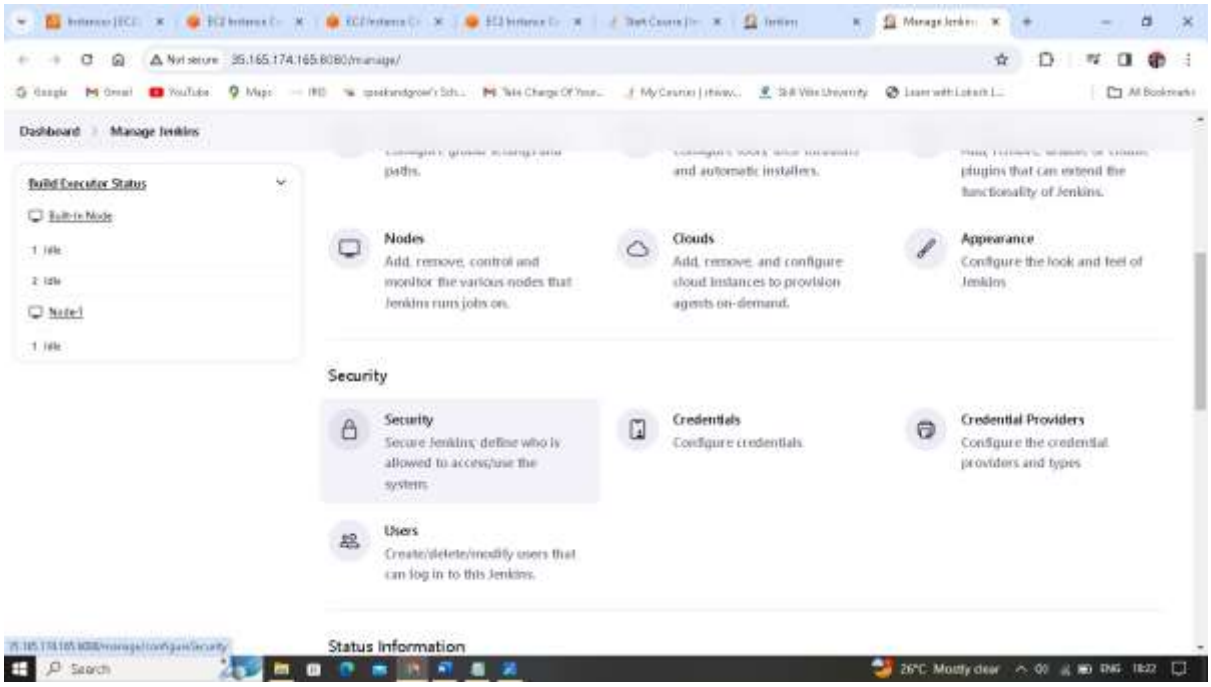
47. fill it



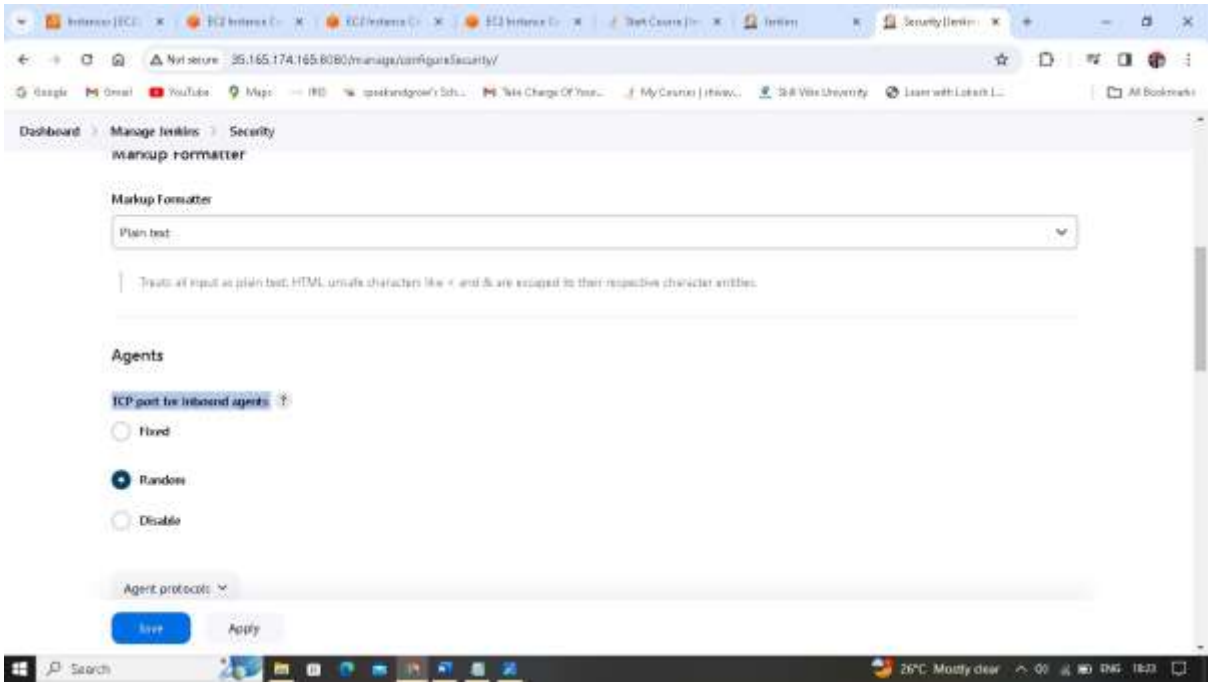
48. For root directory give /home/ubuntu/jenkins and In Launch method select Launch agent by connecting it to the controller.



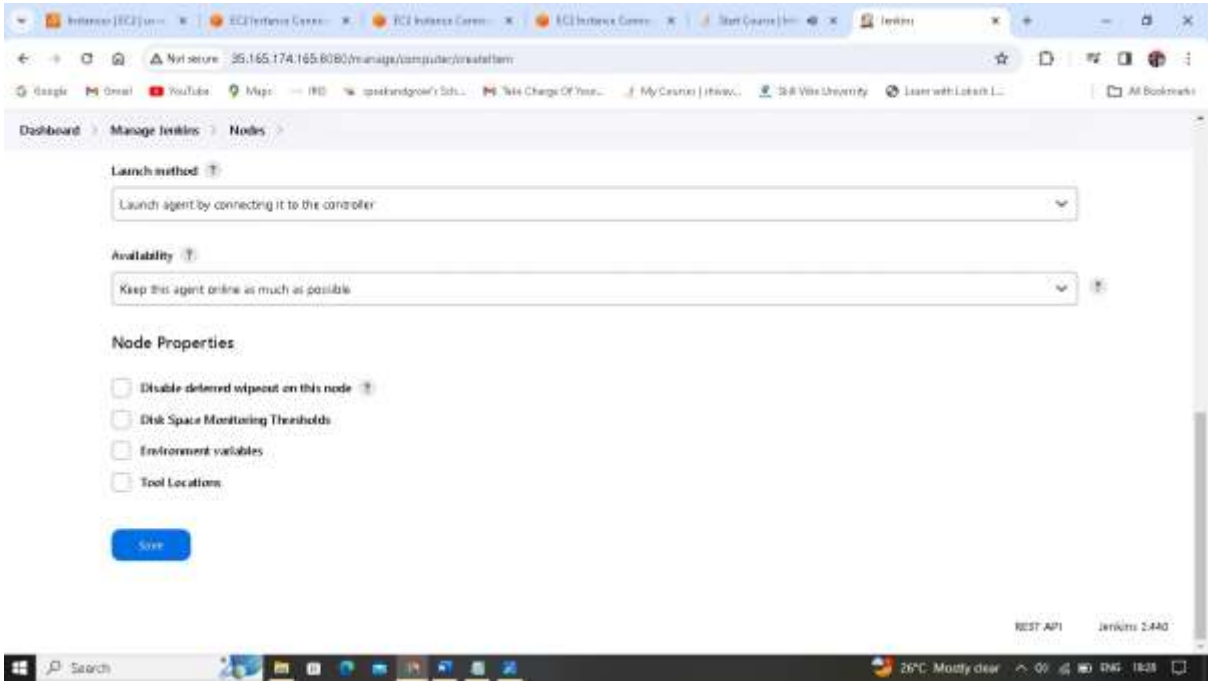
49. Now go to Manage Jenkins click on security



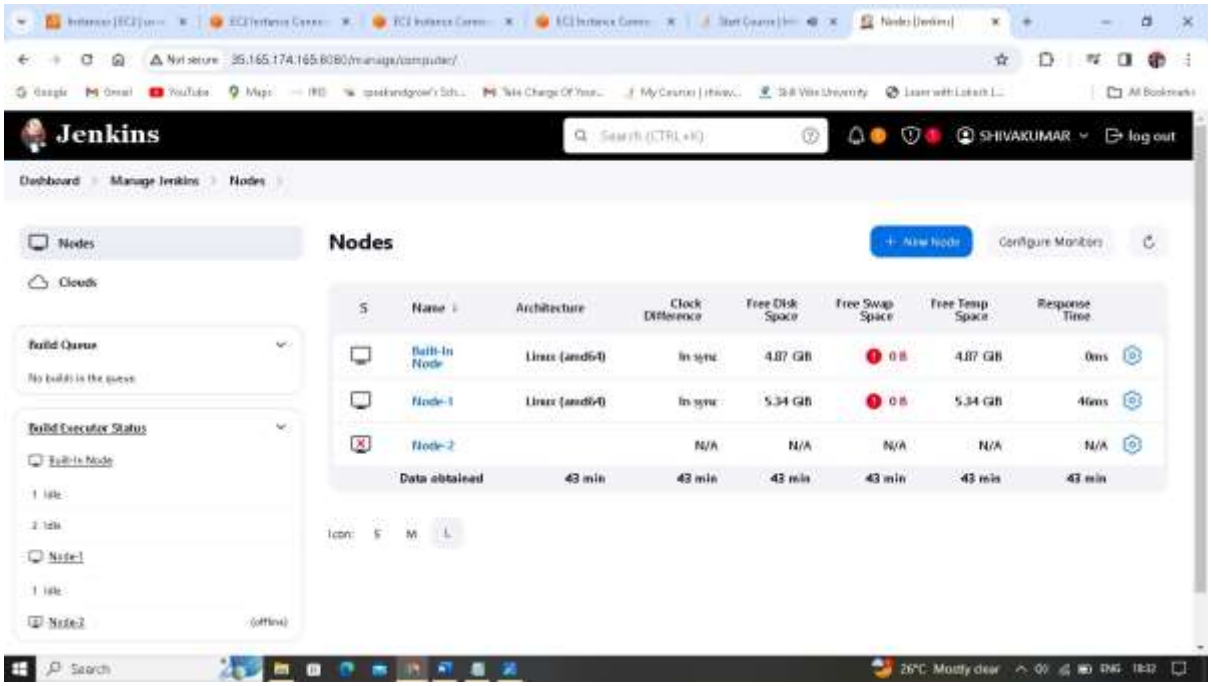
50. Under Agents TCP port for inbound agents, select as Random



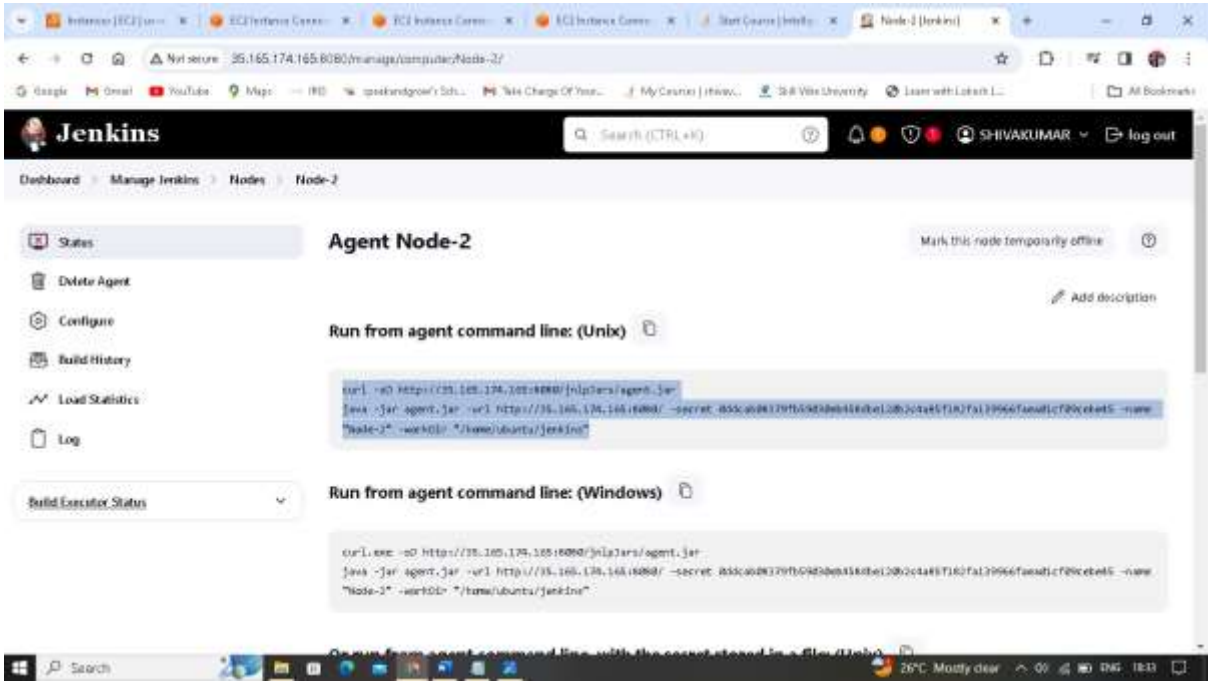
51. Now save it



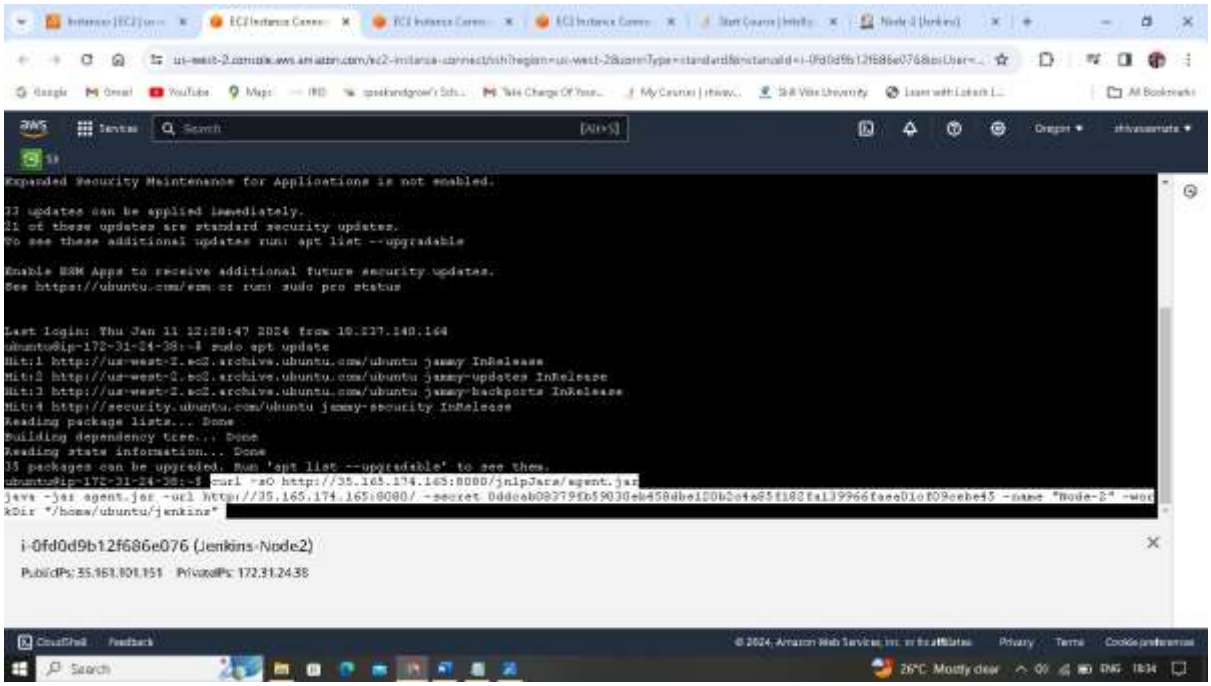
52. Still Node-2 is offline. So click on Node-2



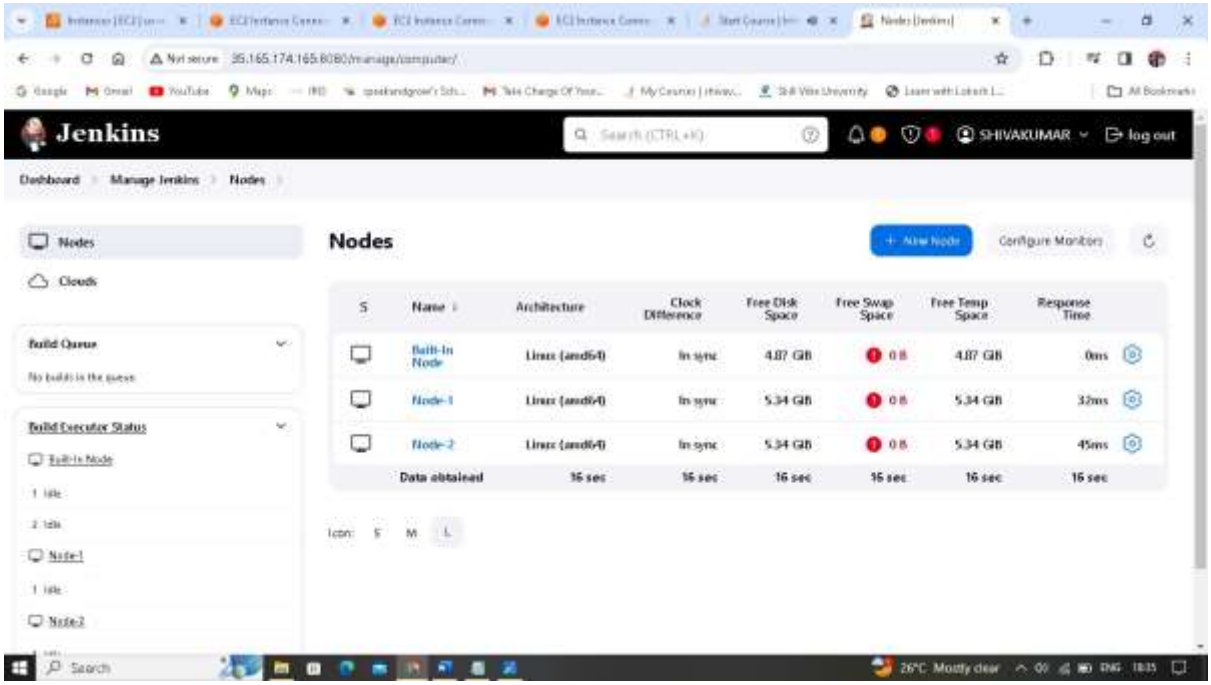
53. Copy the command



54. Paste it on the Jenkins-Node2



55. Now Node-2 also running.



Job1: Trigger a pipeline using Git when push on Develop branch

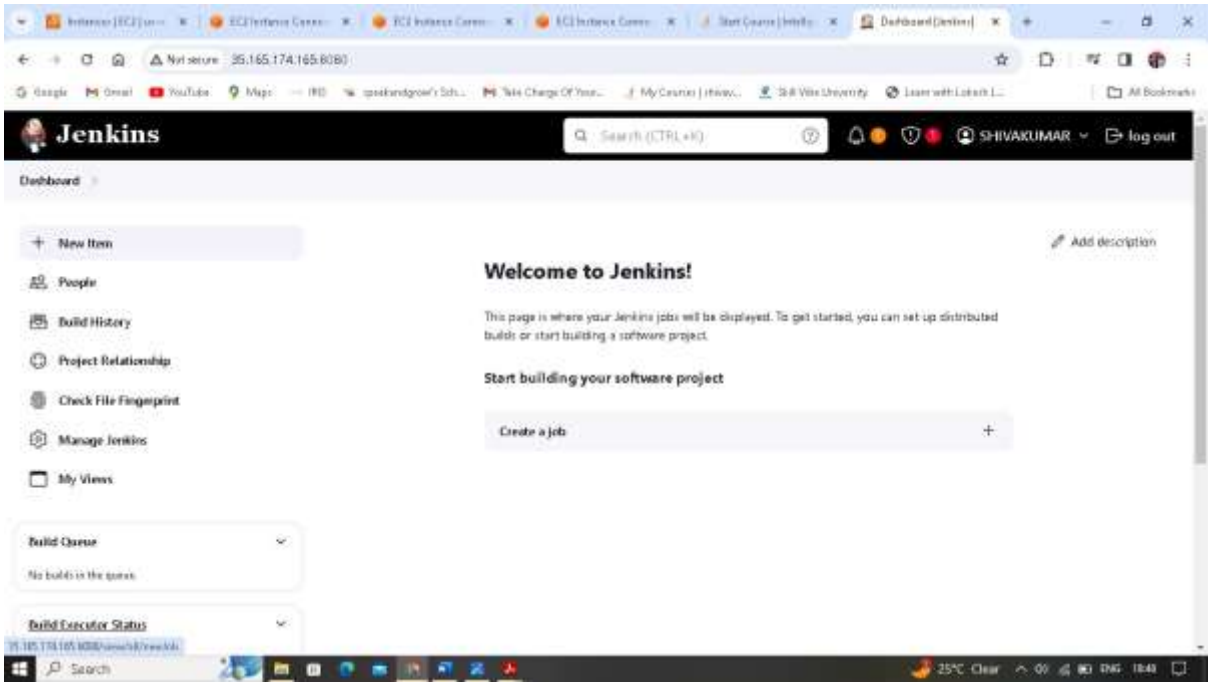
GitHub Repo → develop, master

Job1→ develop branch → Node-1 (test). Job2→ master branch → Node-2 (prod)

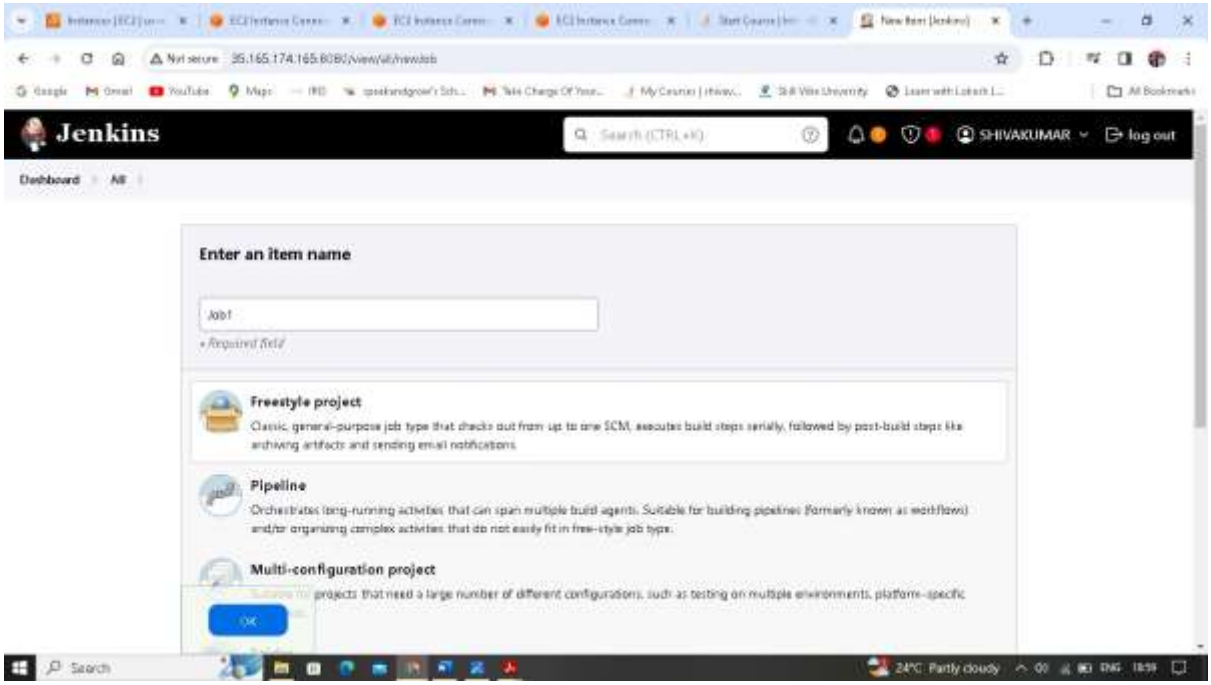
Job3→ Job1-Job2

2. FREESYSTYLE PROJECT

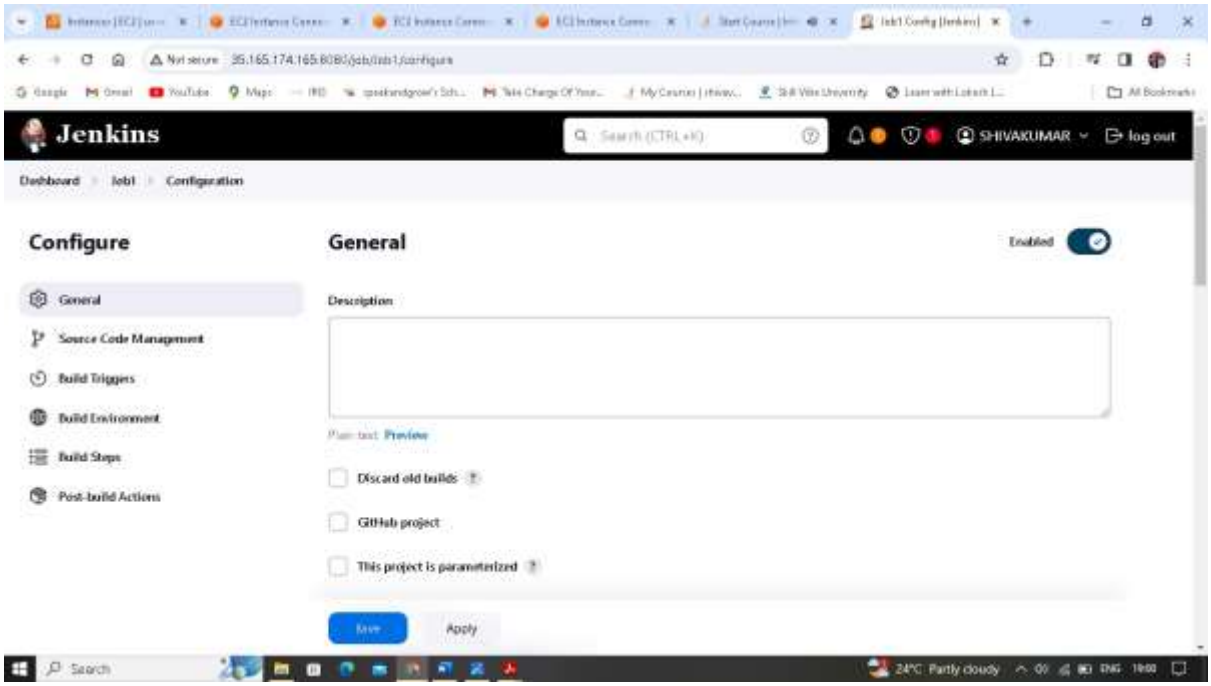
1. First go to Jenkins dashboard and click on New item



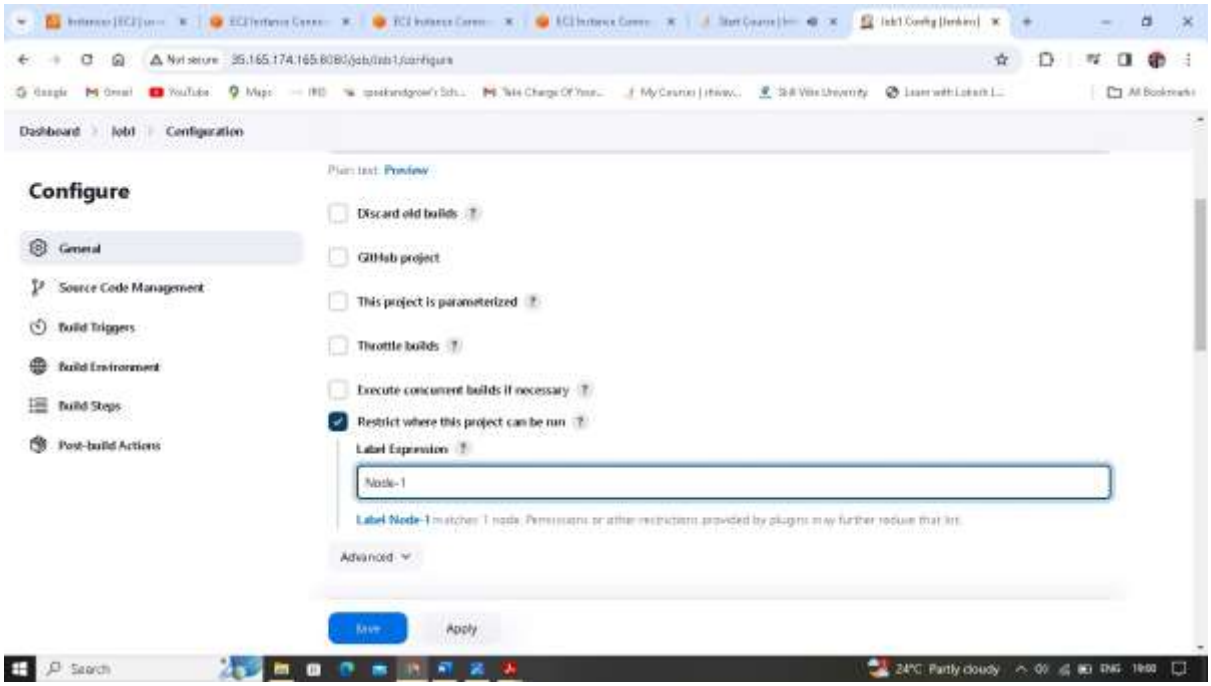
2. Enter an item name as Job1 choose Freestyle project and click on ok.



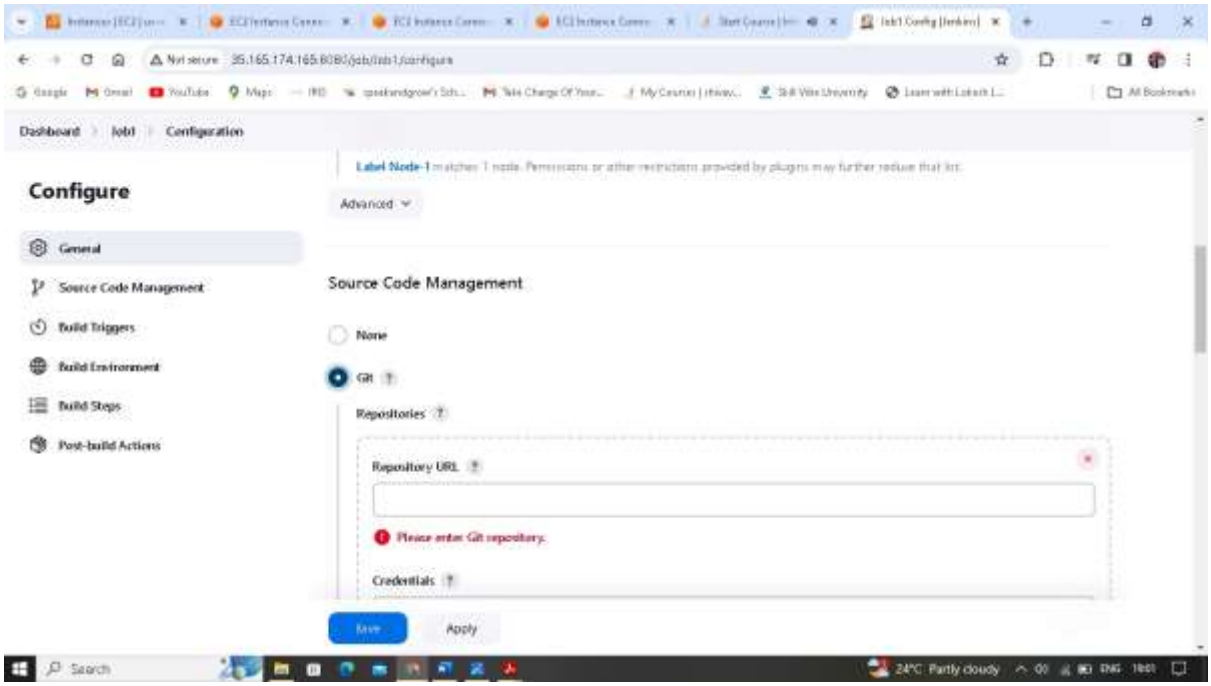
3. check it one by one and fill it



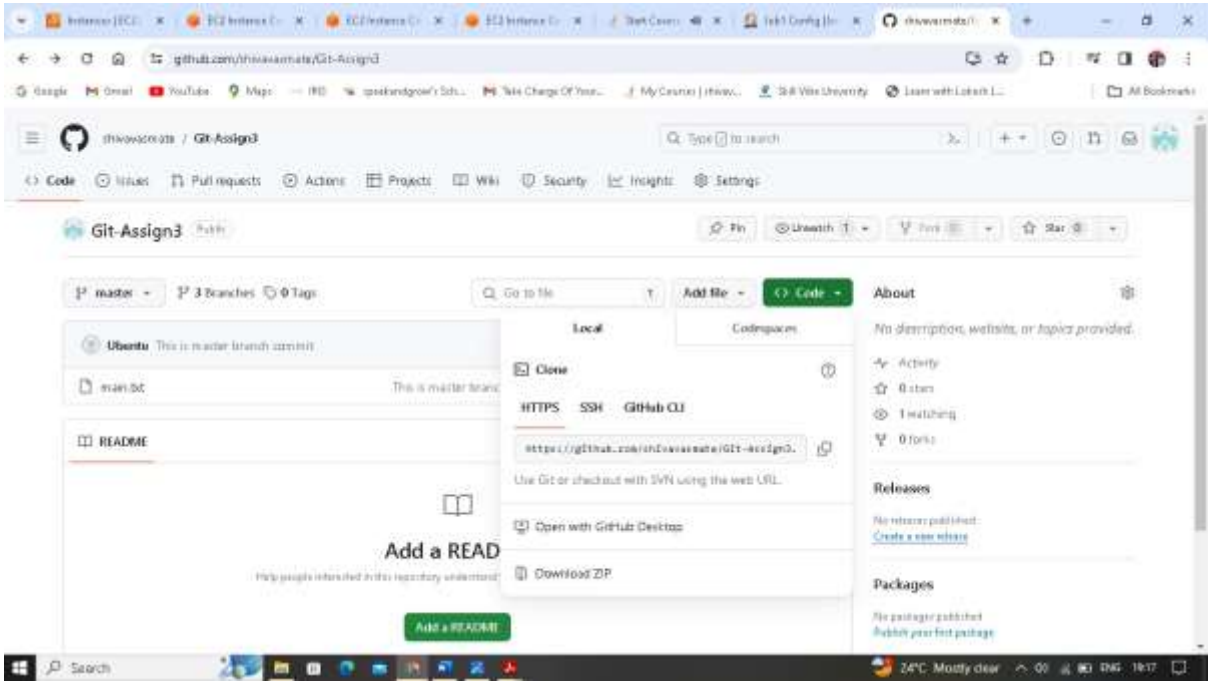
4. First of all check in Restrict where this project can be run. Label Expression as Node-1



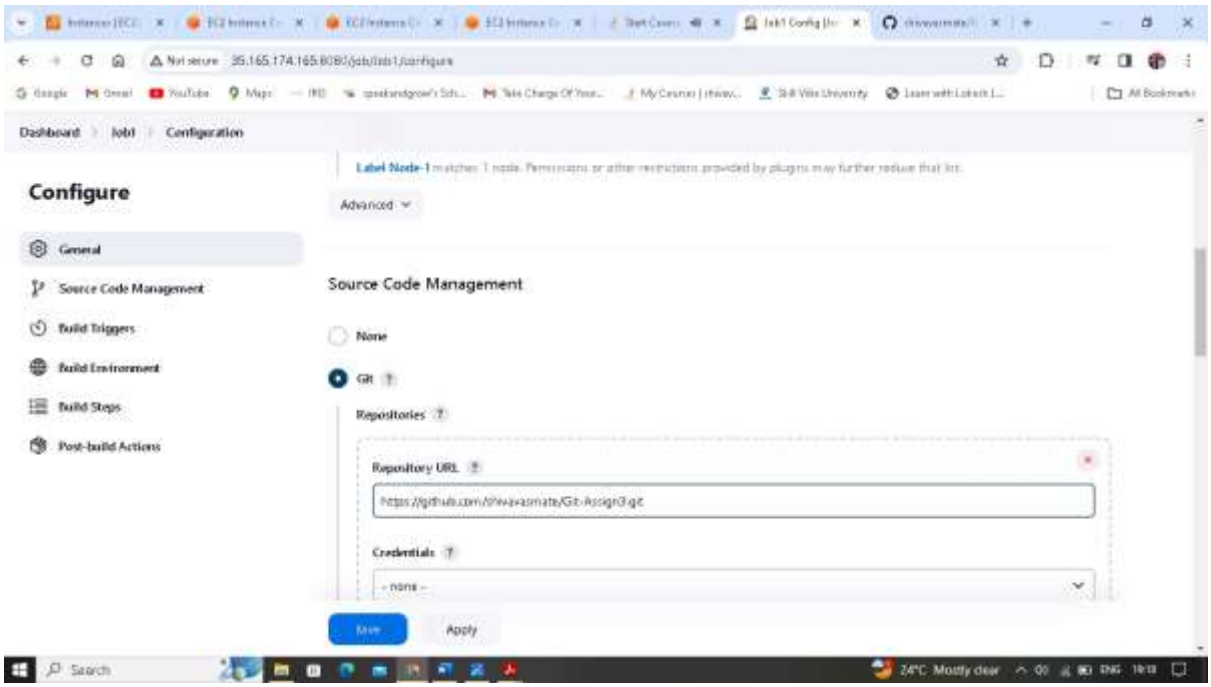
5. Under Source Code Management Check Git



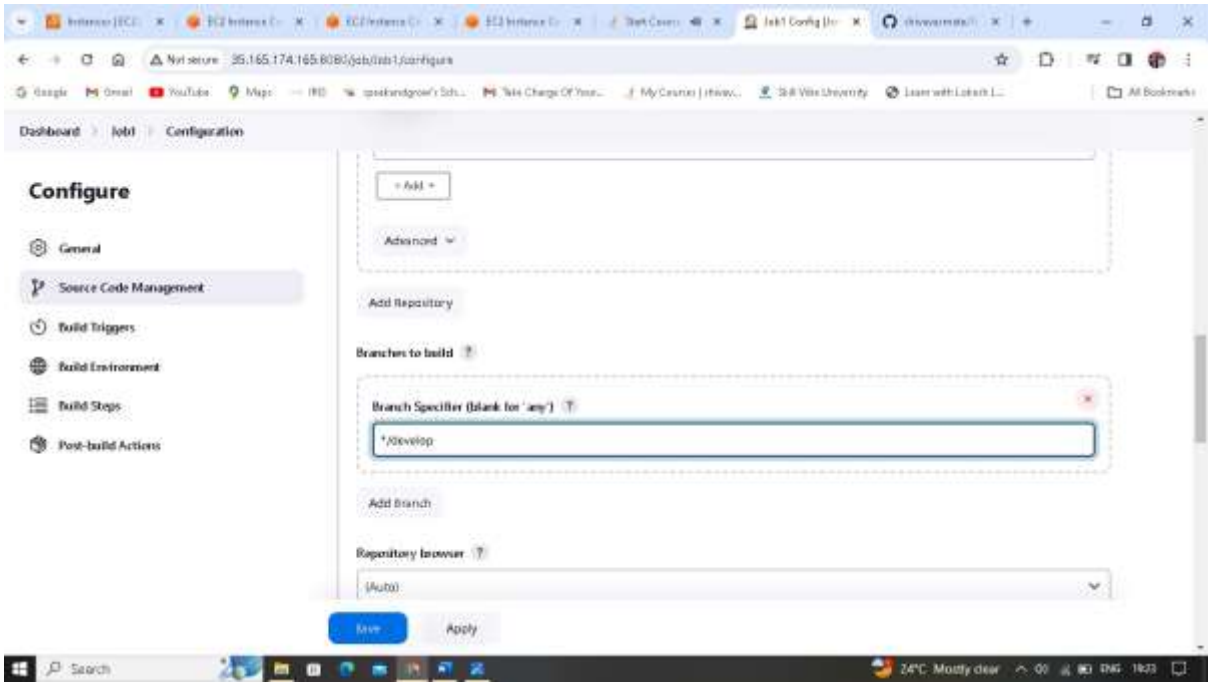
6.Now log in to your GitHub in another browser.



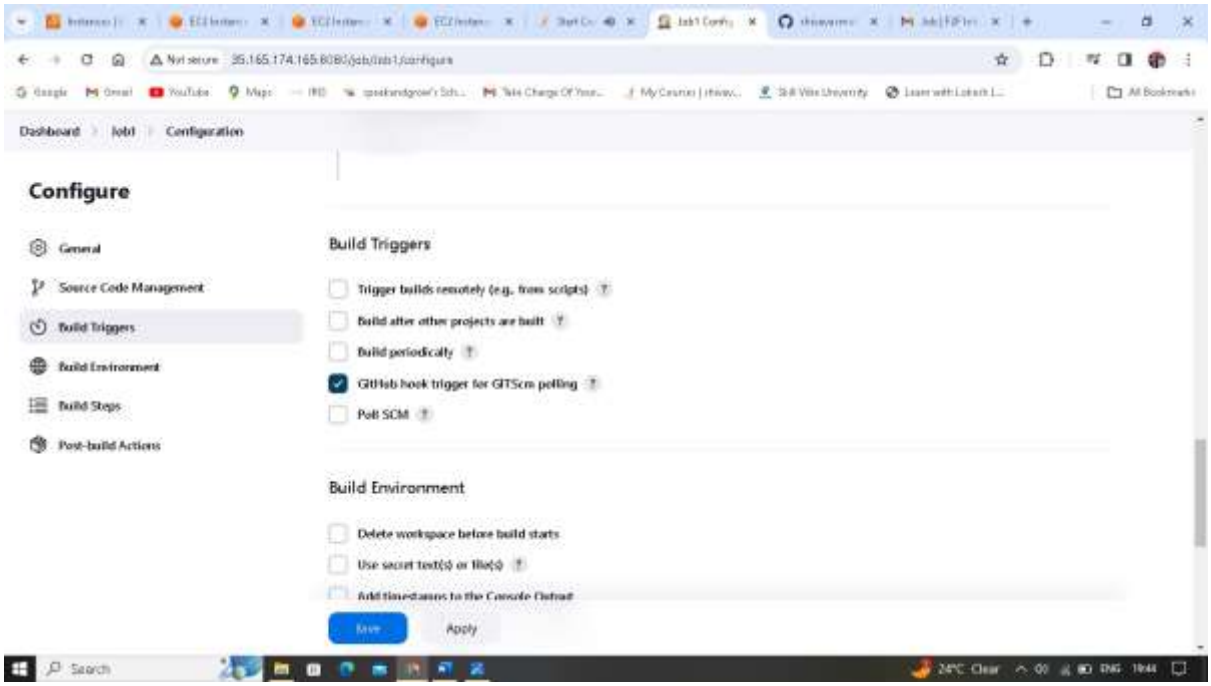
7. For Repository URL copy from code and choose the repo which has develop branch in it(Git-Assign3).



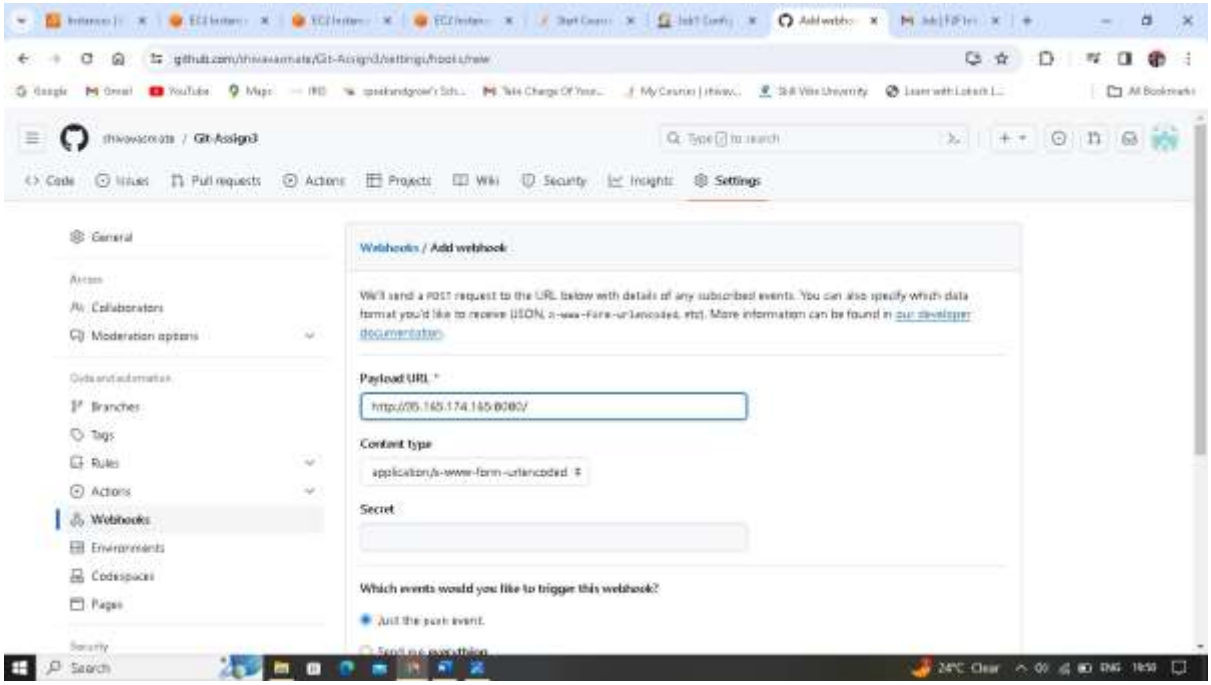
8. In Branches to build and Branch Specifier give */develop



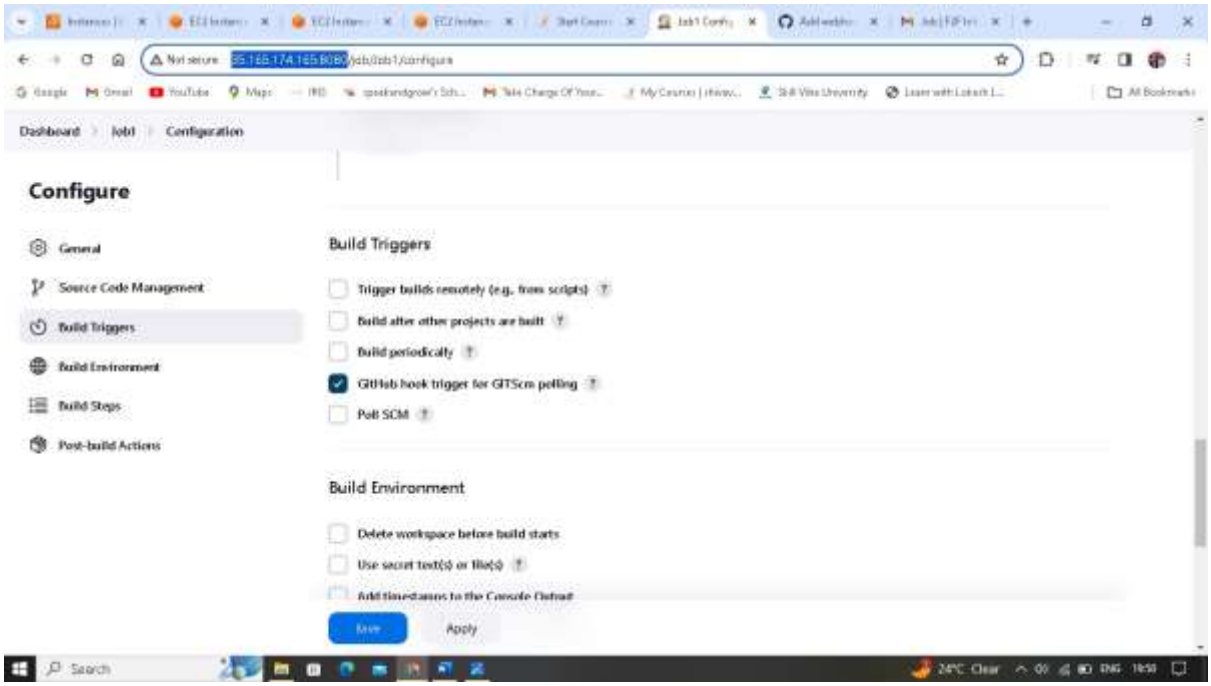
9.Under Build Triggers choose GitHub hock trigger for GIT Scm polling.



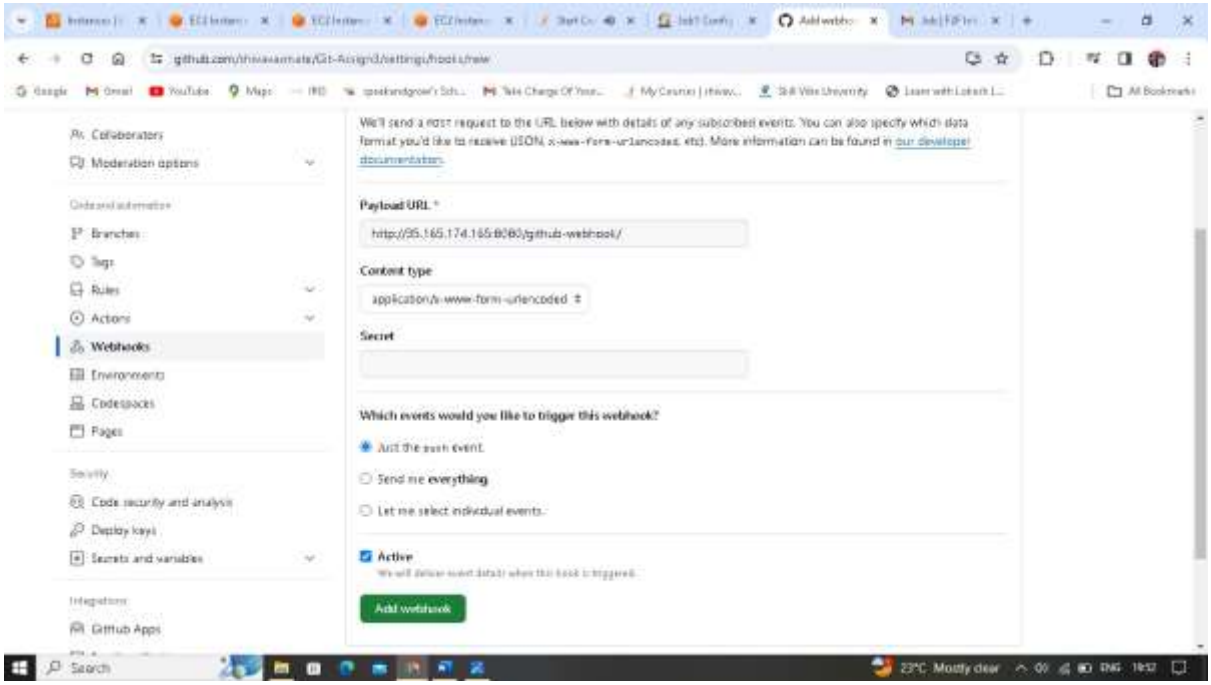
10.Now go to GitHub under repo settings look for Webhooks on left side



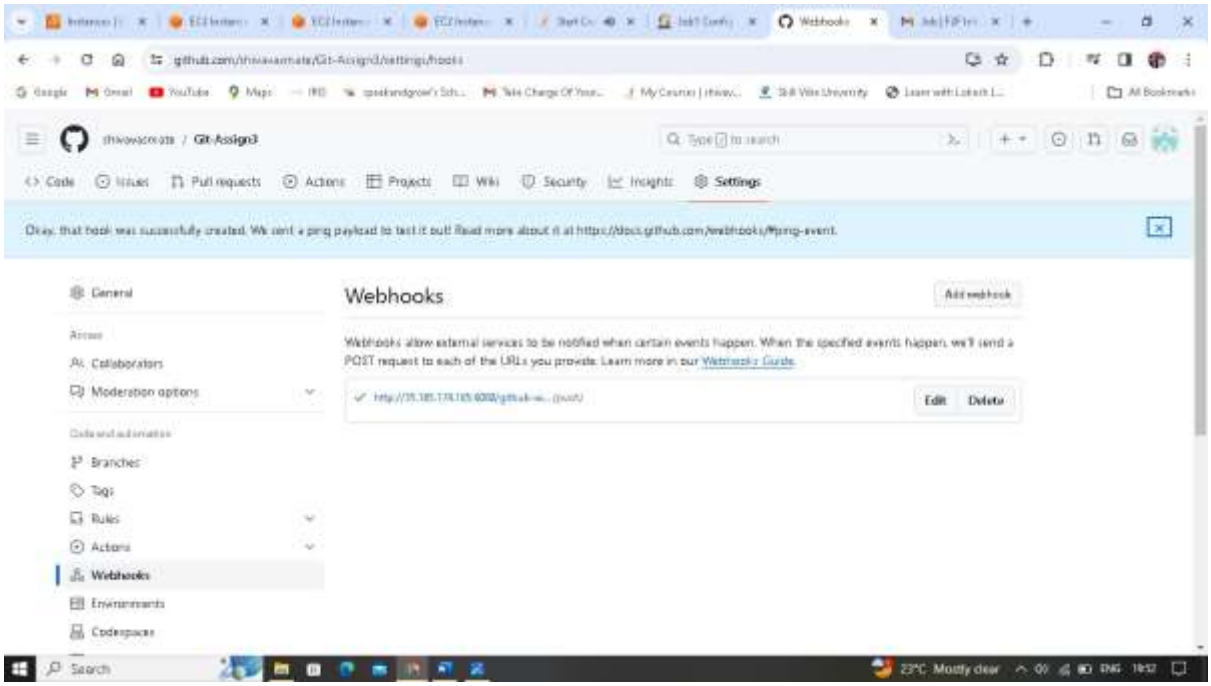
11. For Payload URL copy the http:// 35.165.174.165:8080/



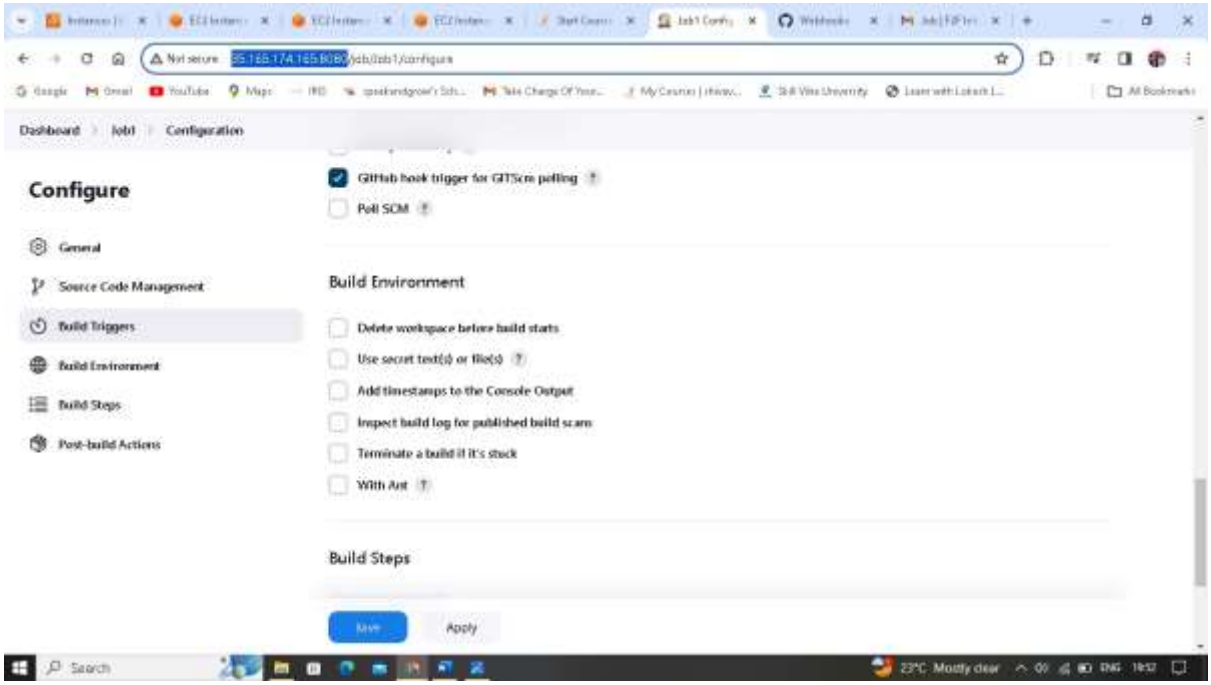
12. Give as http://35.165.174.165:8080/github-webhook/



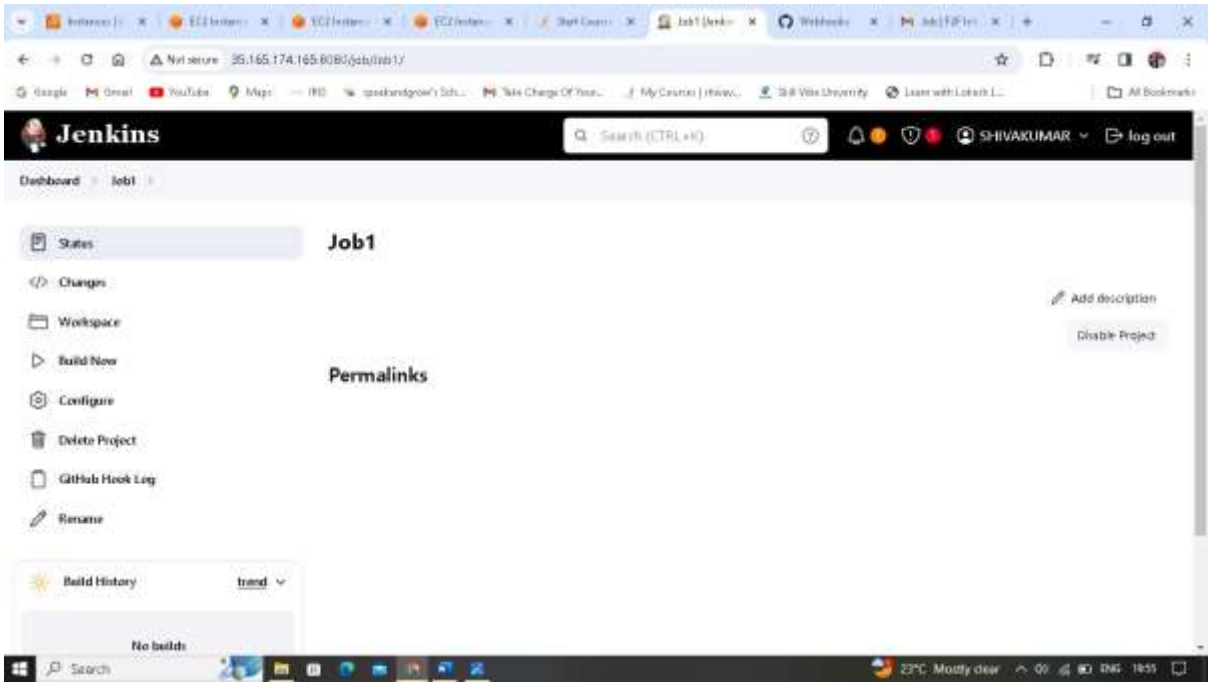
13. Check for right mark



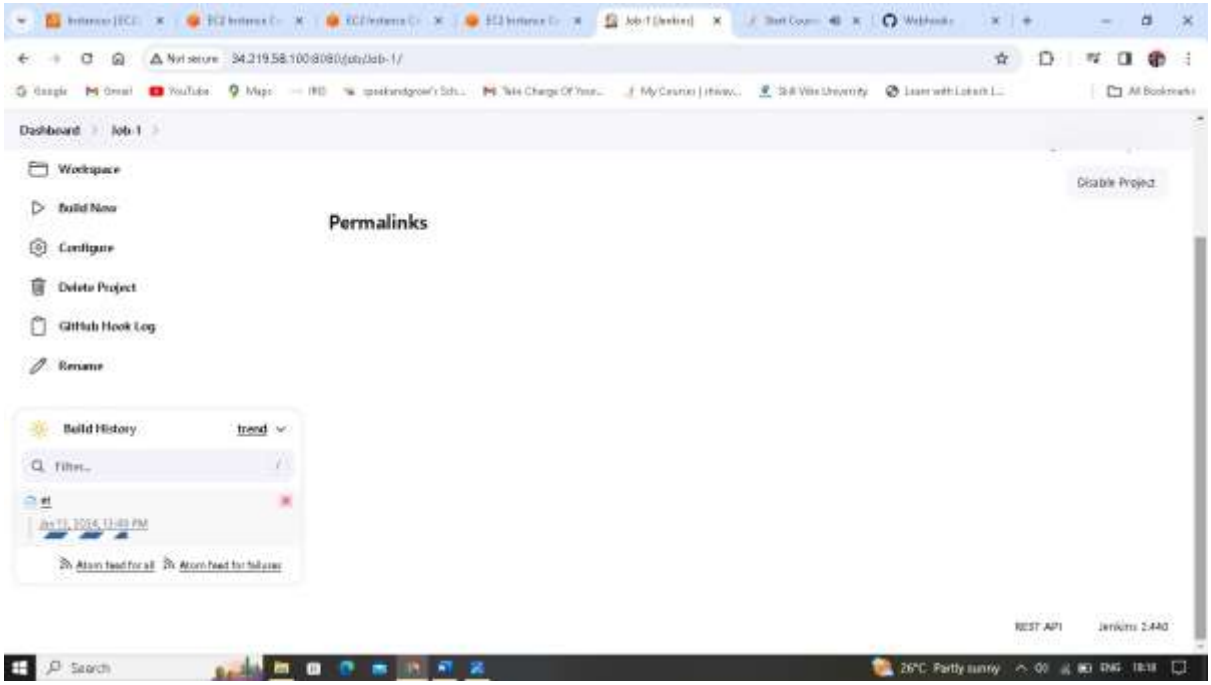
14. Go to Jenkins and click on save.



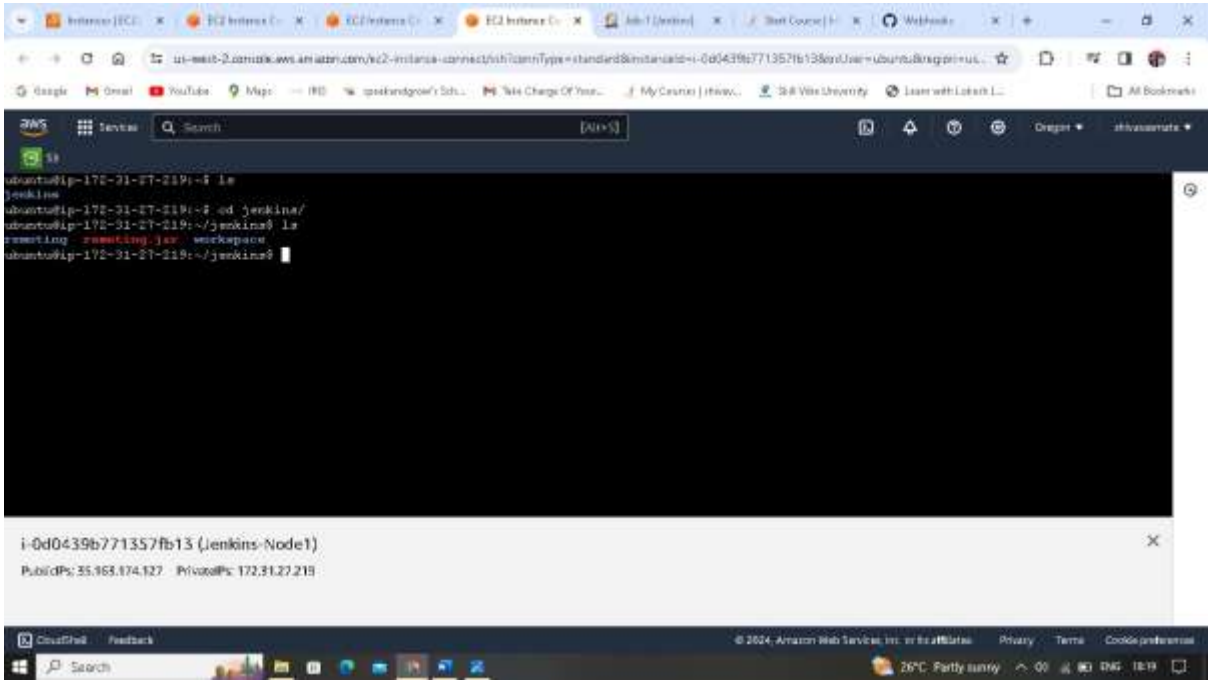
15. Job1 is create



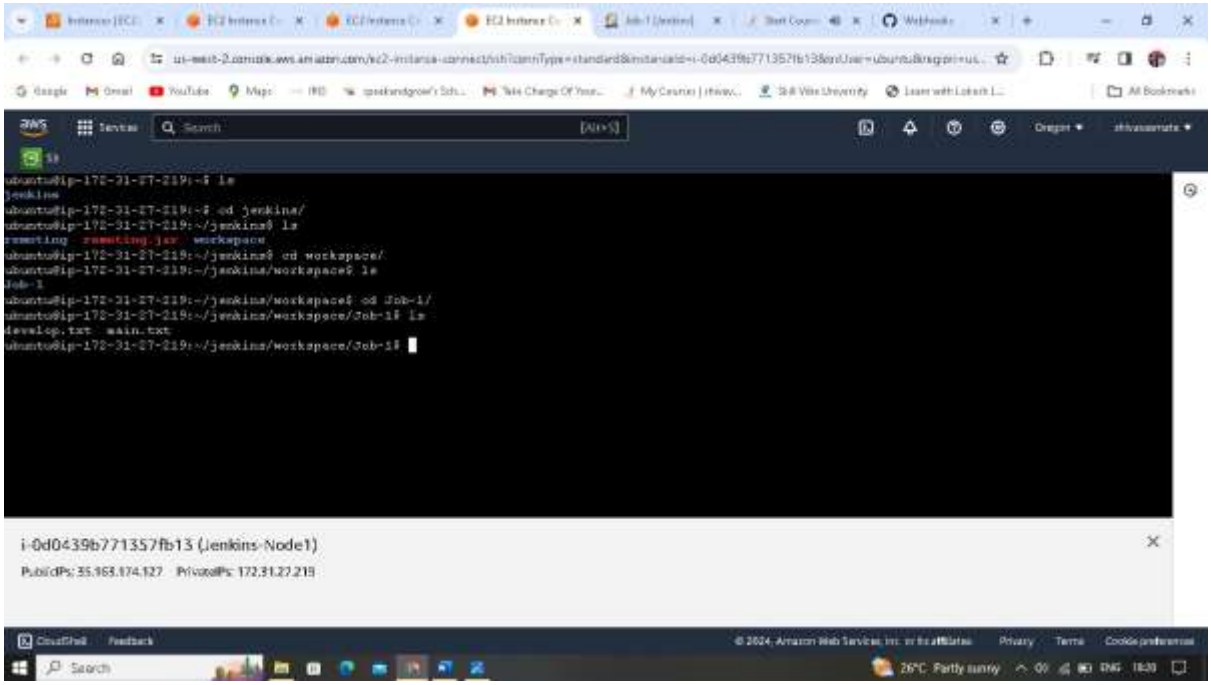
16. Click on Build Now



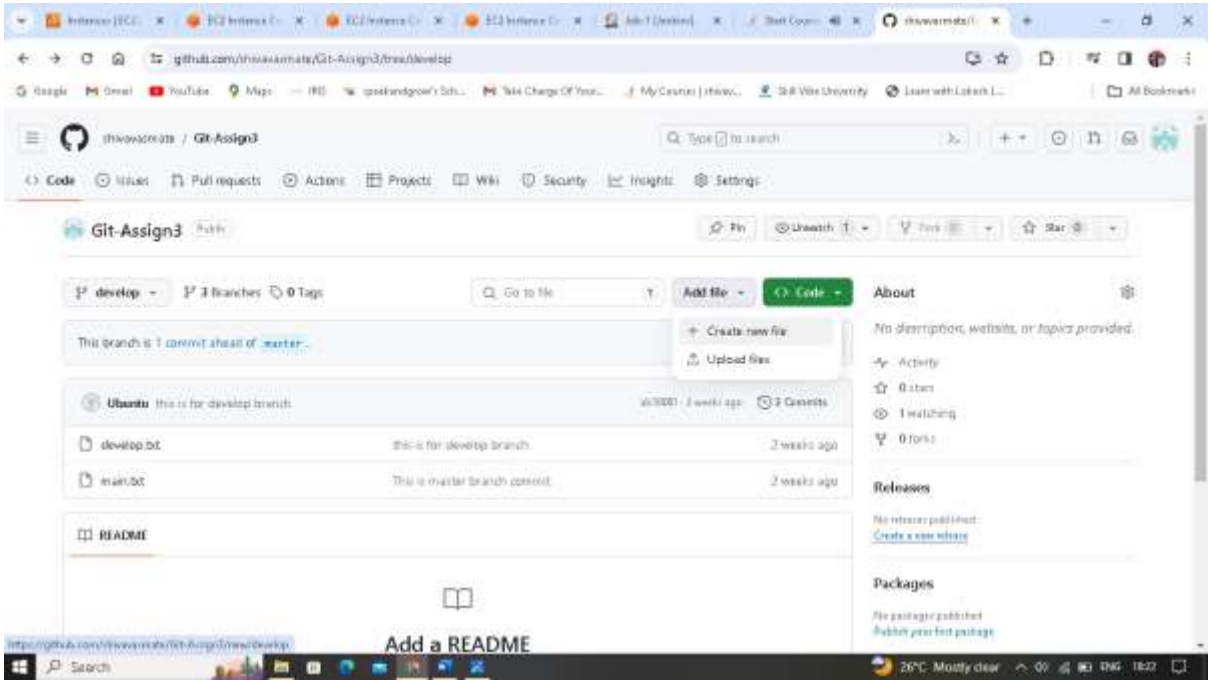
17. Go to Jenkins-Node1 do ls jenkins folder again do ls remoting remoting.jar and workspace (new one is create after build the Job1)



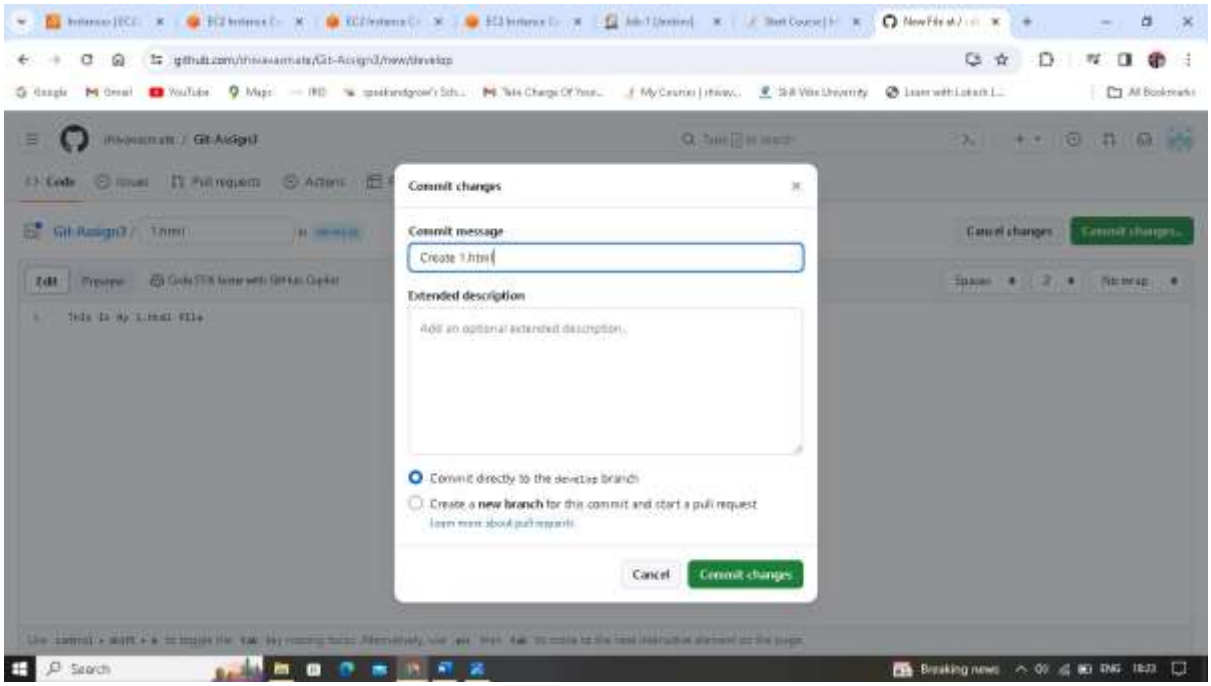
18.Under workspace do ls Job-1 folder is create and
under Job-1 folder develop.txt and main.txt as same as GitHub repo.



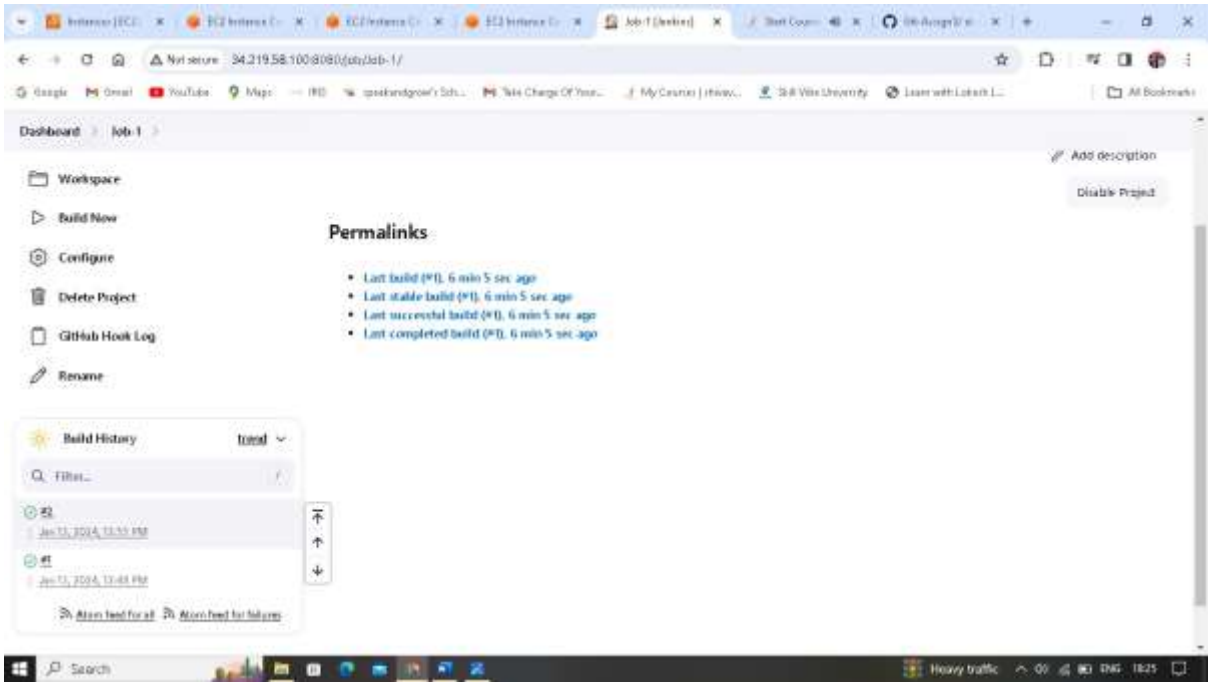
19. Now to check again go to GitHub repo under Git-Assign3 of develop branch add a file as 1.html.txt



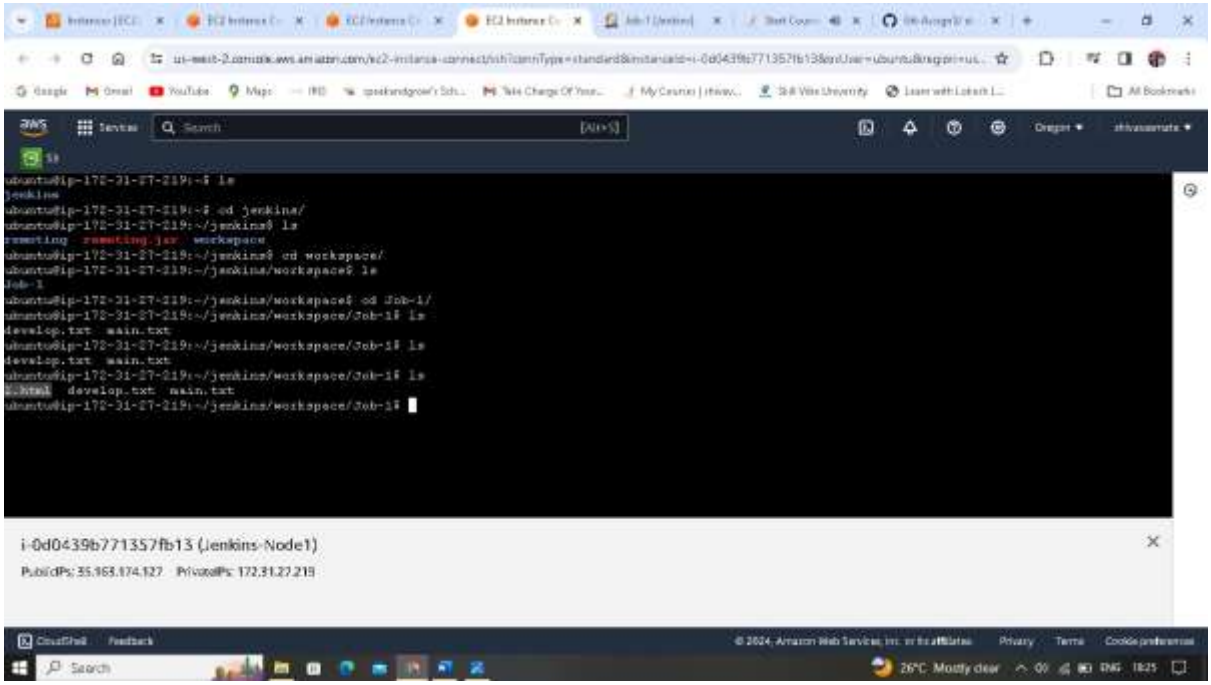
20. Now click on commit changes



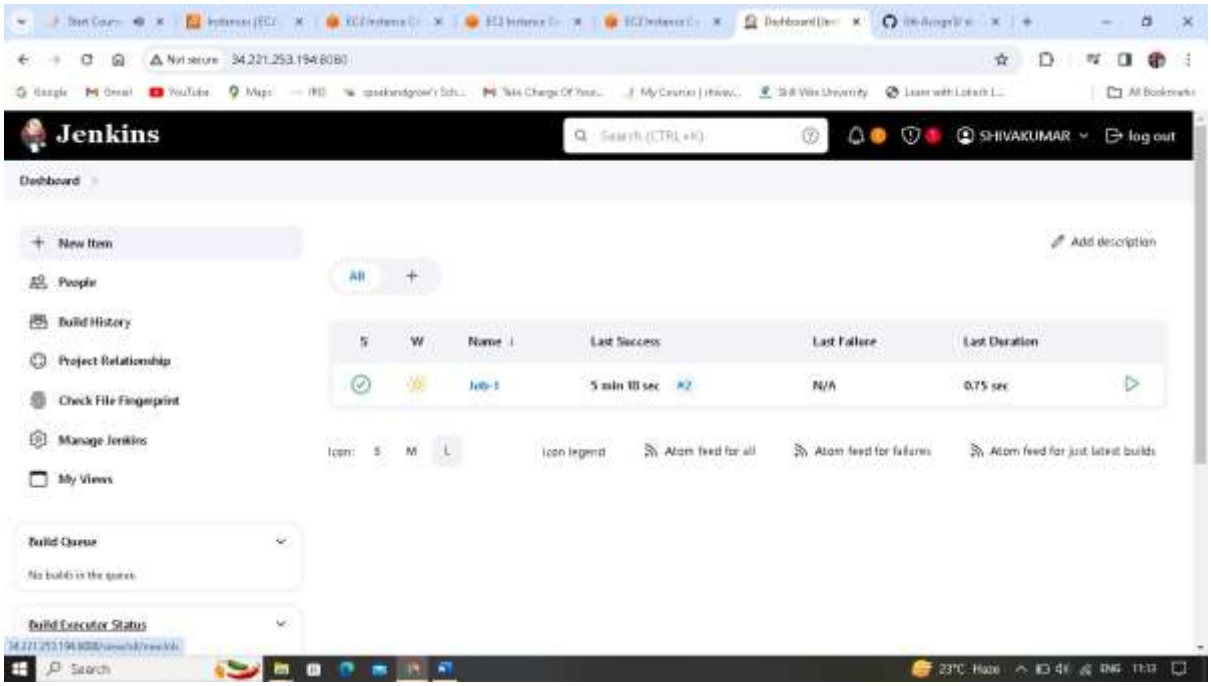
21. Automatic #2 build is created



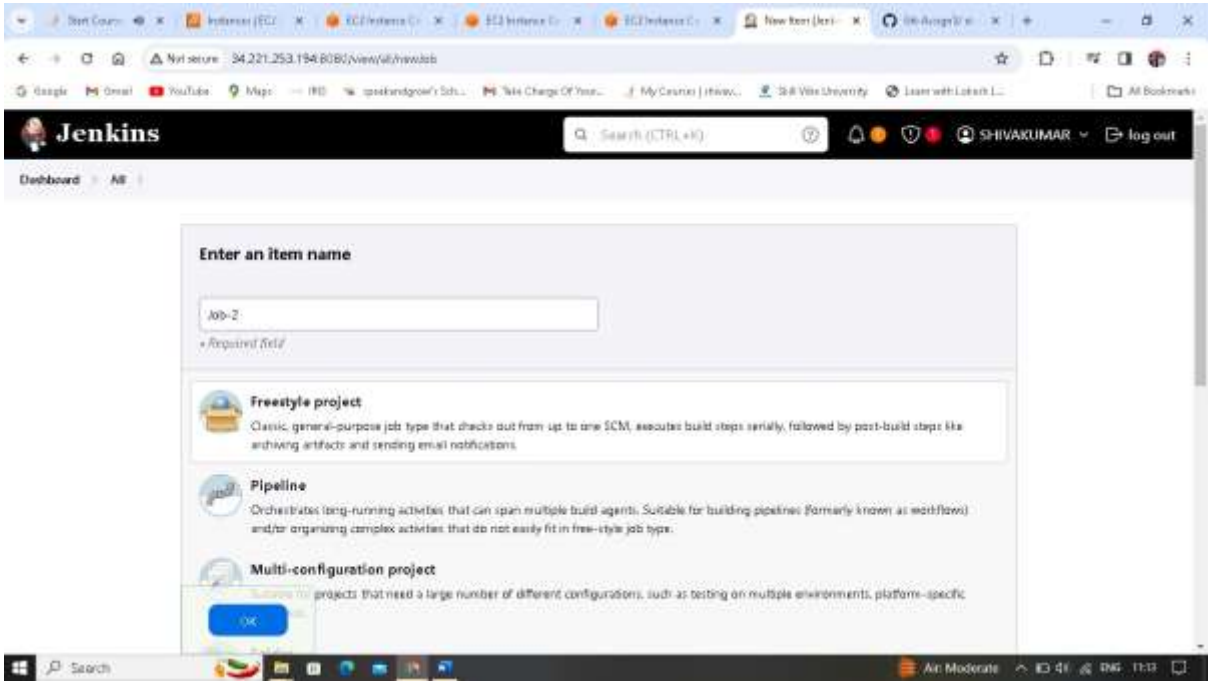
22.Now do ls again. This time 1.html also added with develop.txt and main.txt



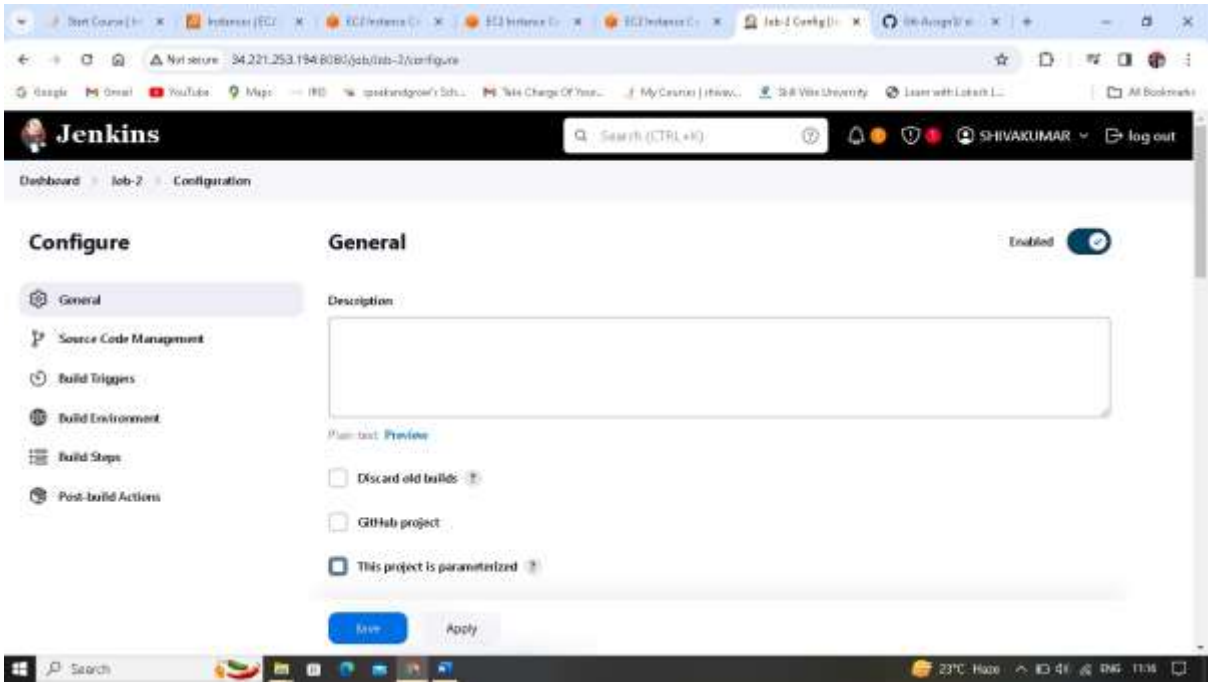
1. Go back to Dashboard and click on New item.



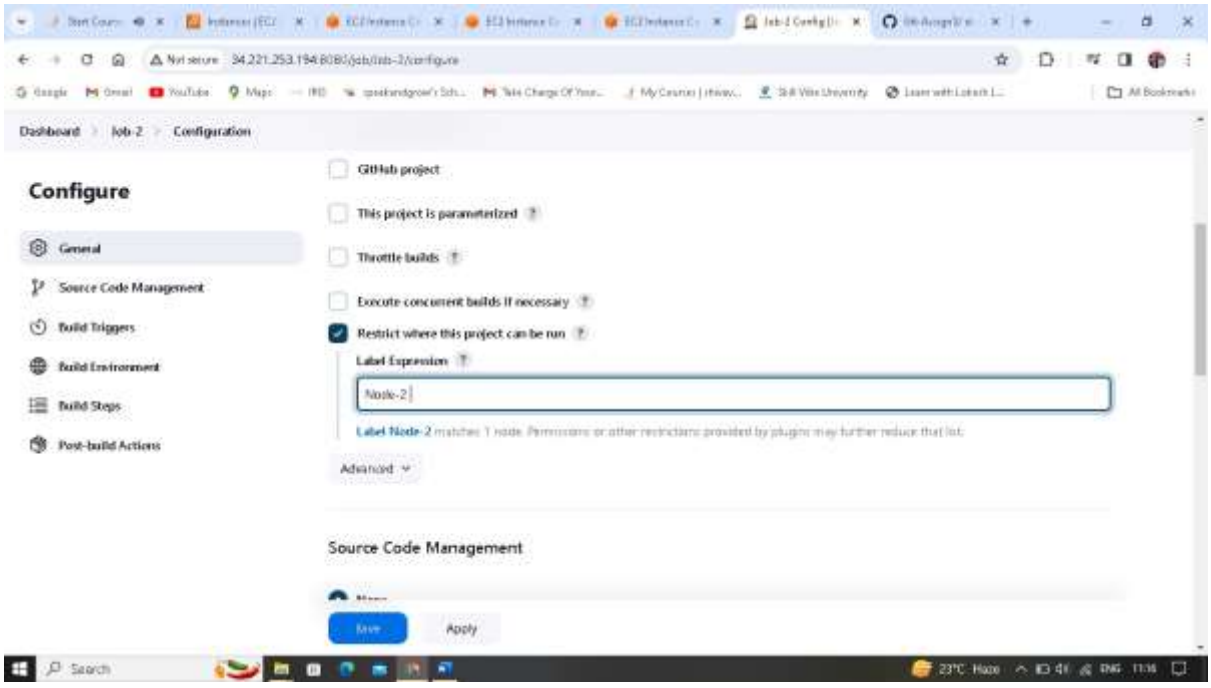
2. Enter an item name as Job-2



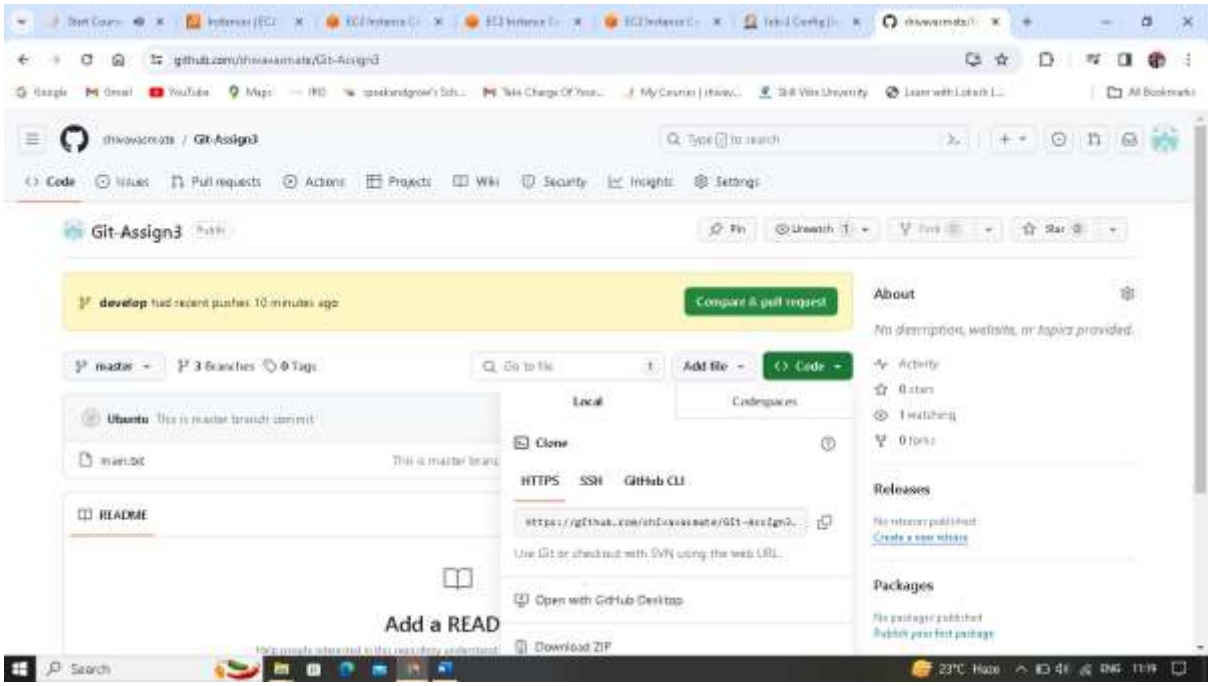
3. fill it



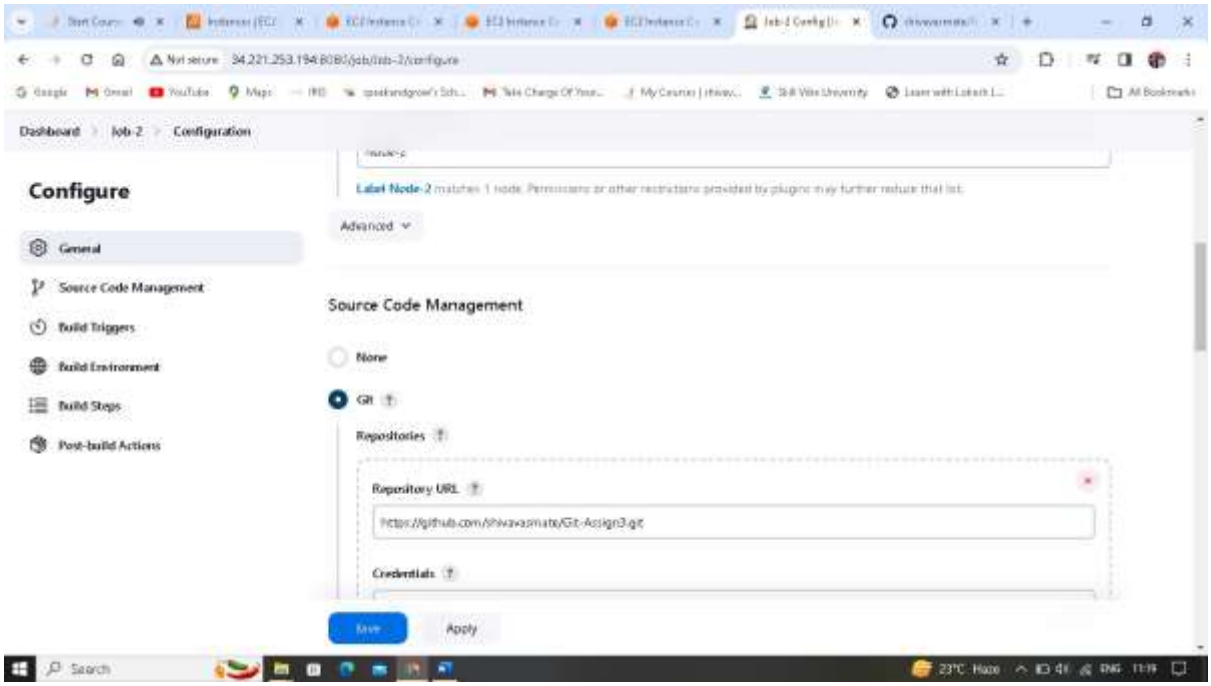
4.In Label Expression as Node-2



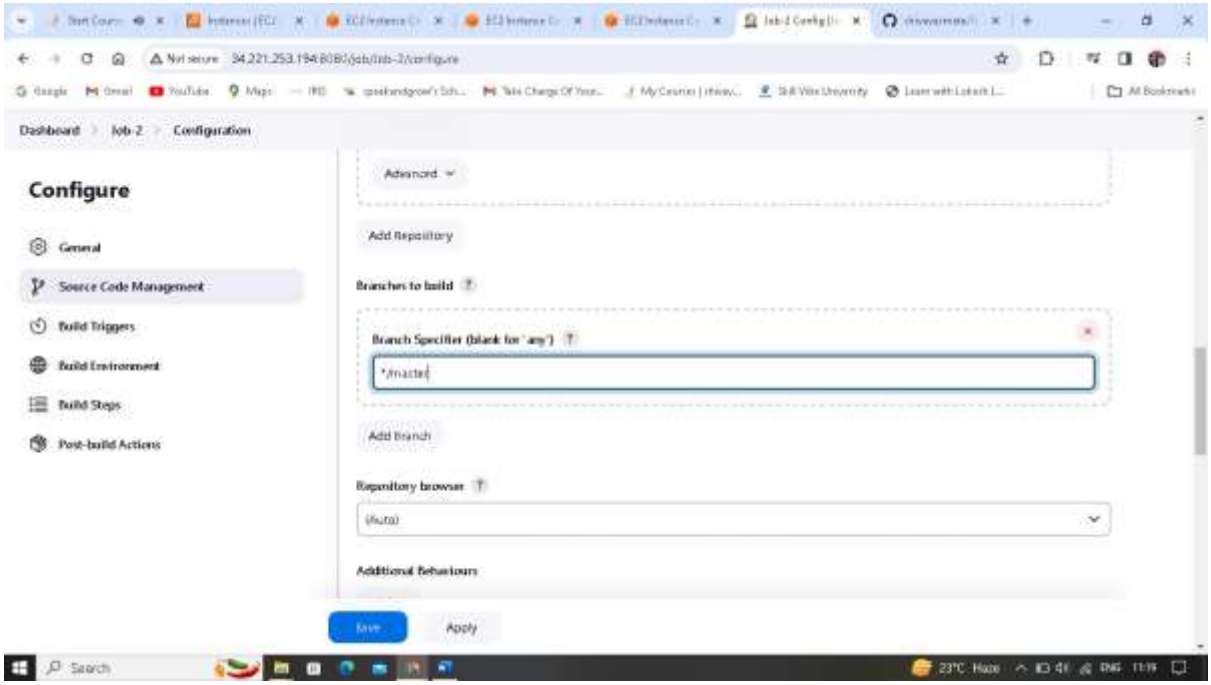
5. copy the URL for Github



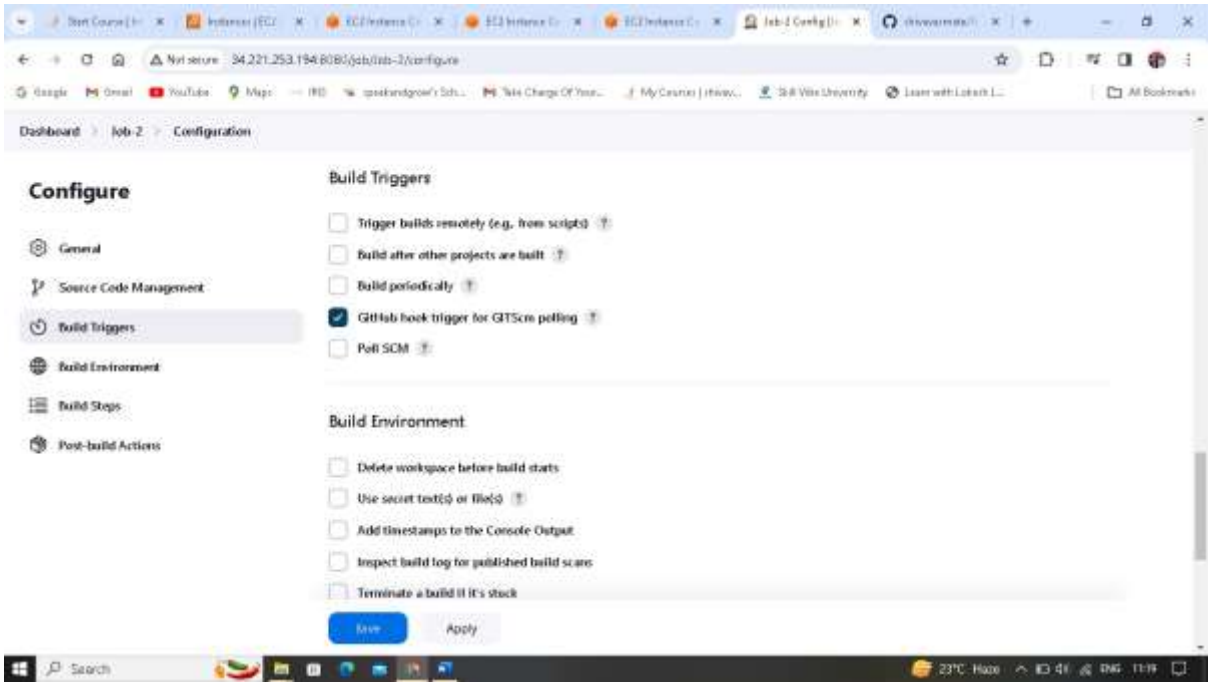
6.Paste in the Repository URL and choose the repo which has master branch in it (Git-Assign3).



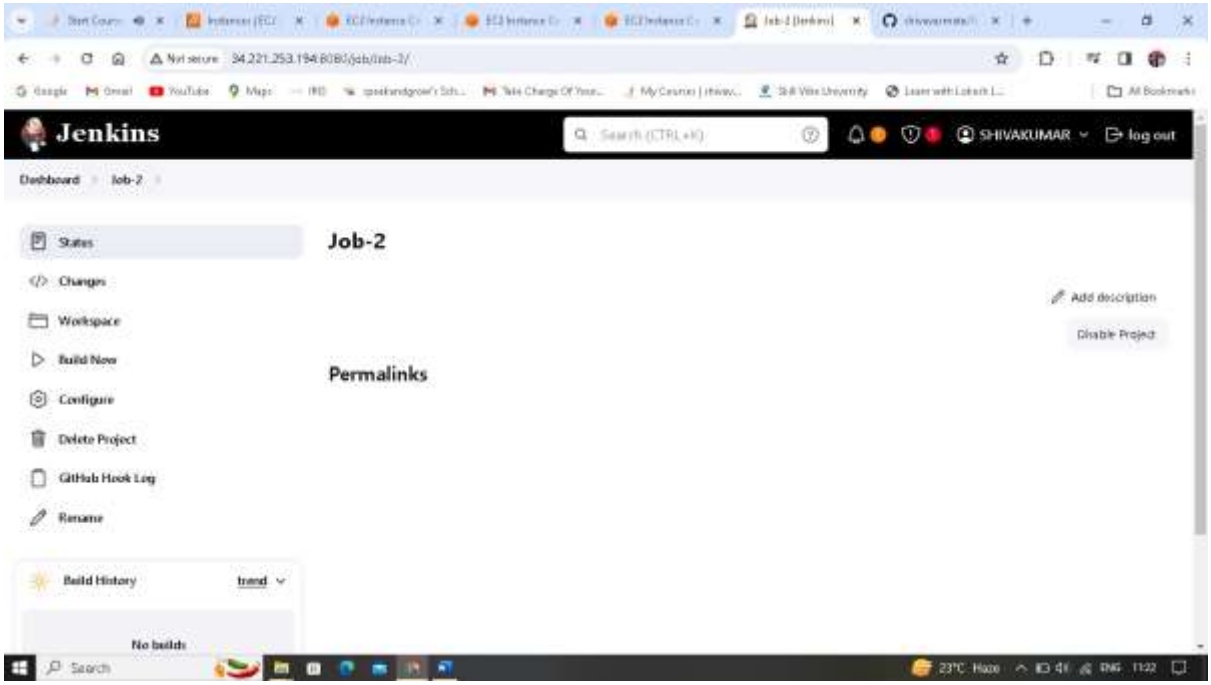
7. In Branches to build and Branch Specifier give */master



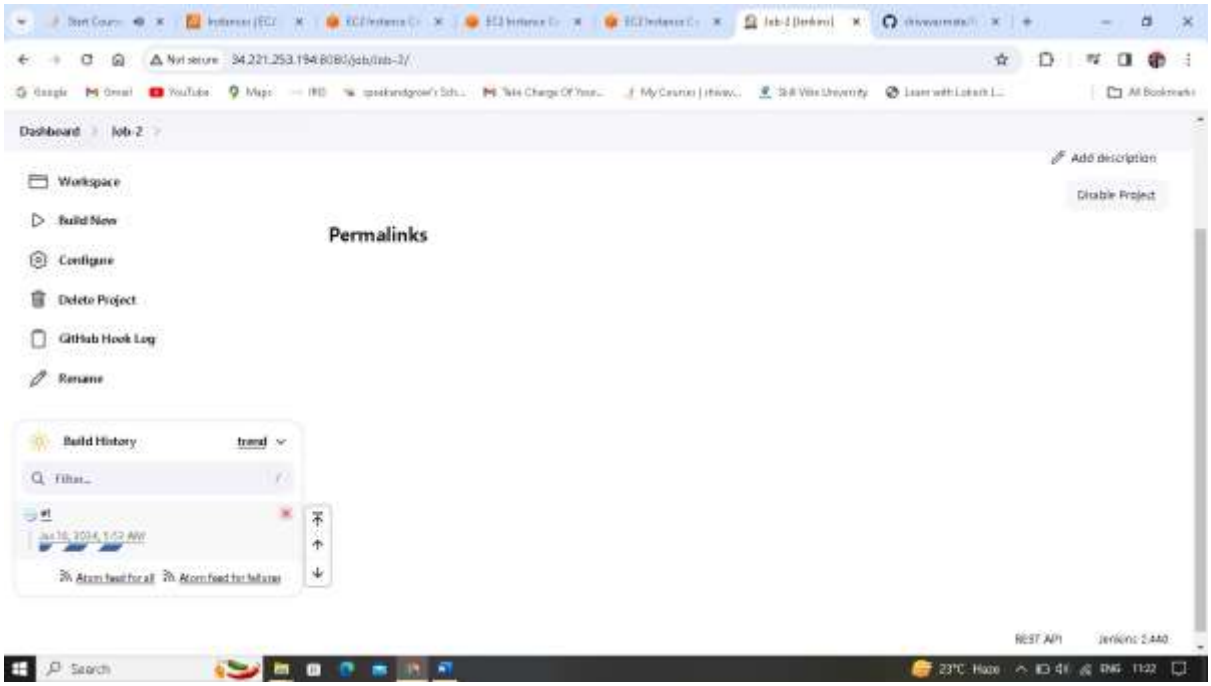
8. In Build Triggers choose GitHub hook trigger for Git Scm polling and click on save



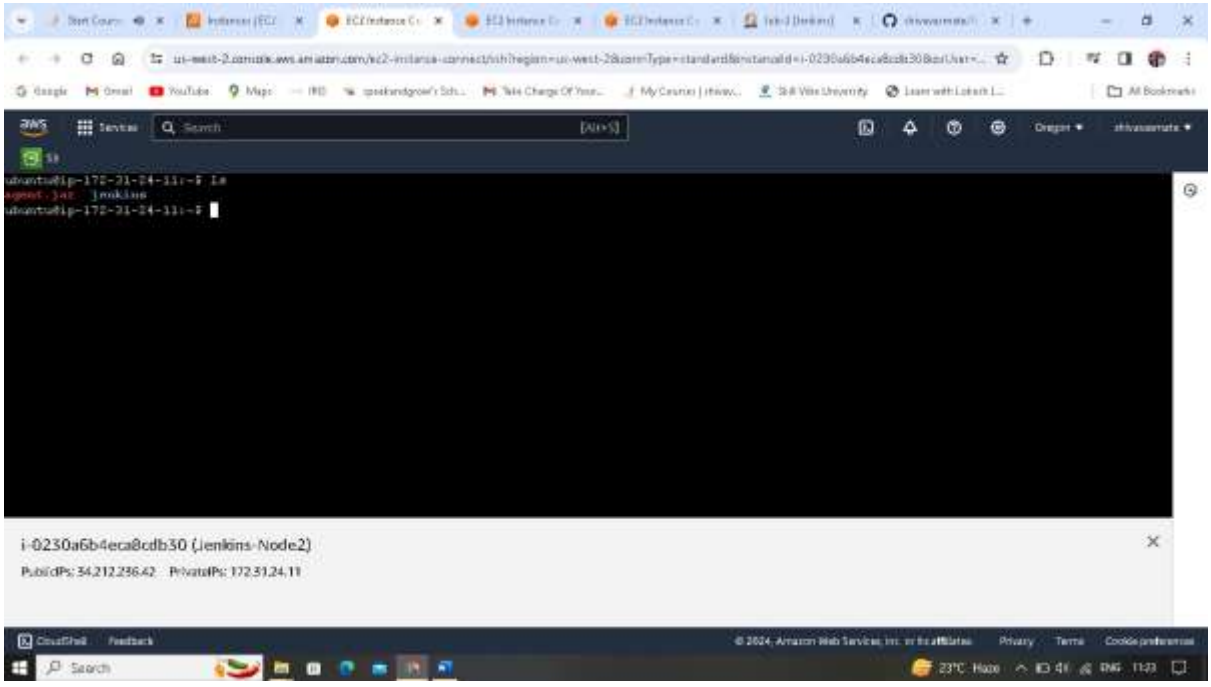
9. Job-2 is created



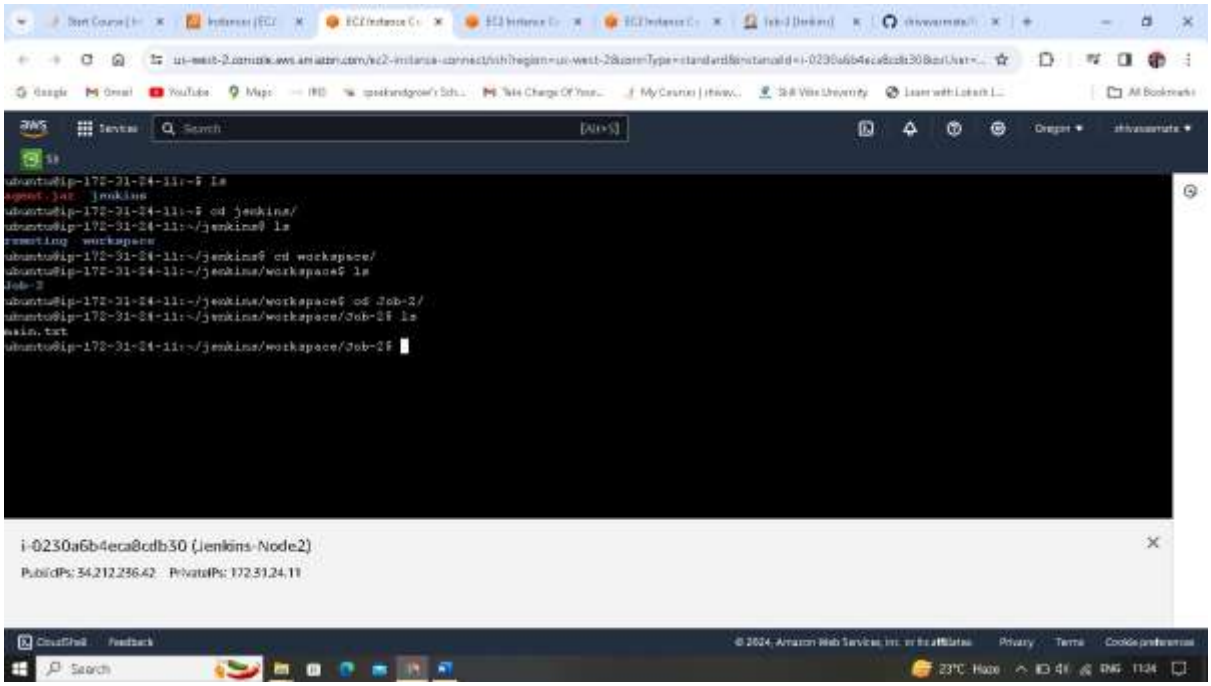
10. Click on Build Now.



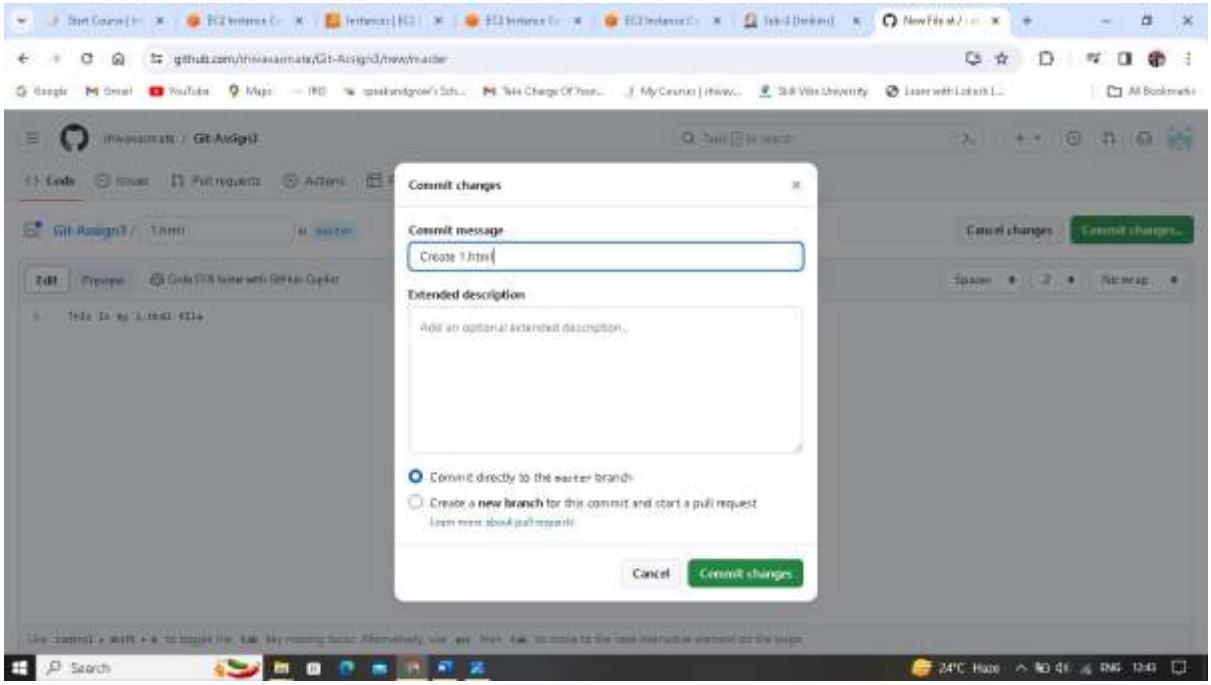
11. Go to Node-2 machine do ls, agent.jar and jenkins folders are present.



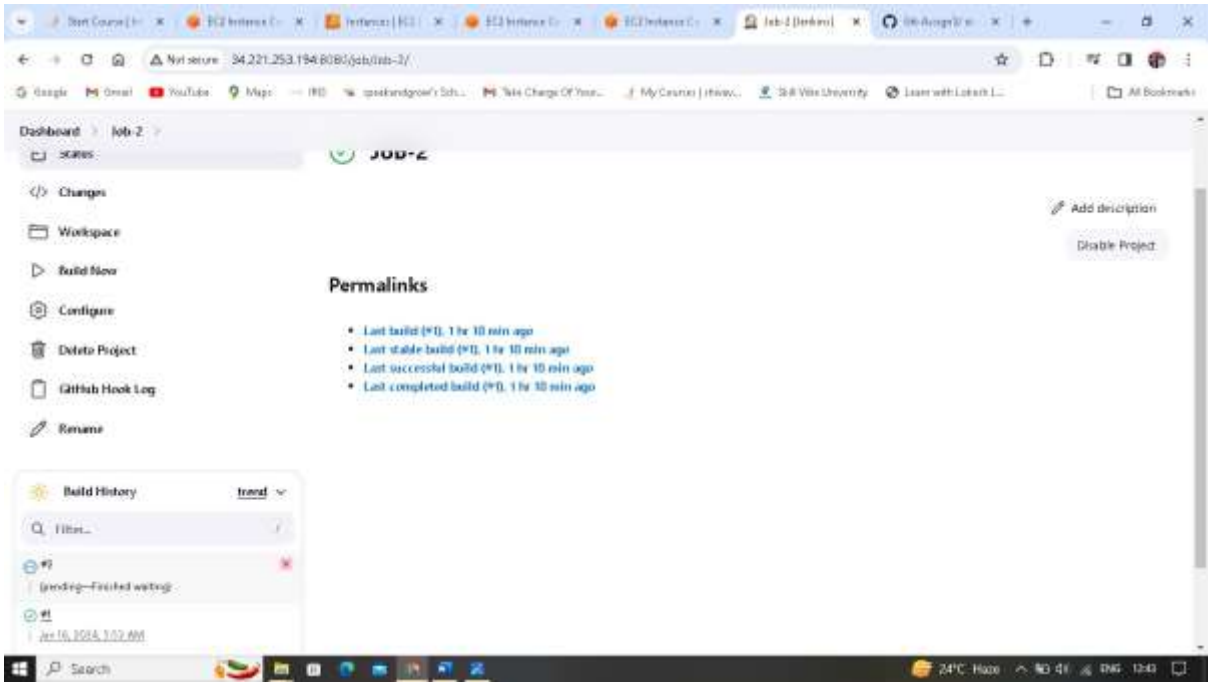
12. Finally Job-2 have main.txt from main branch of github.



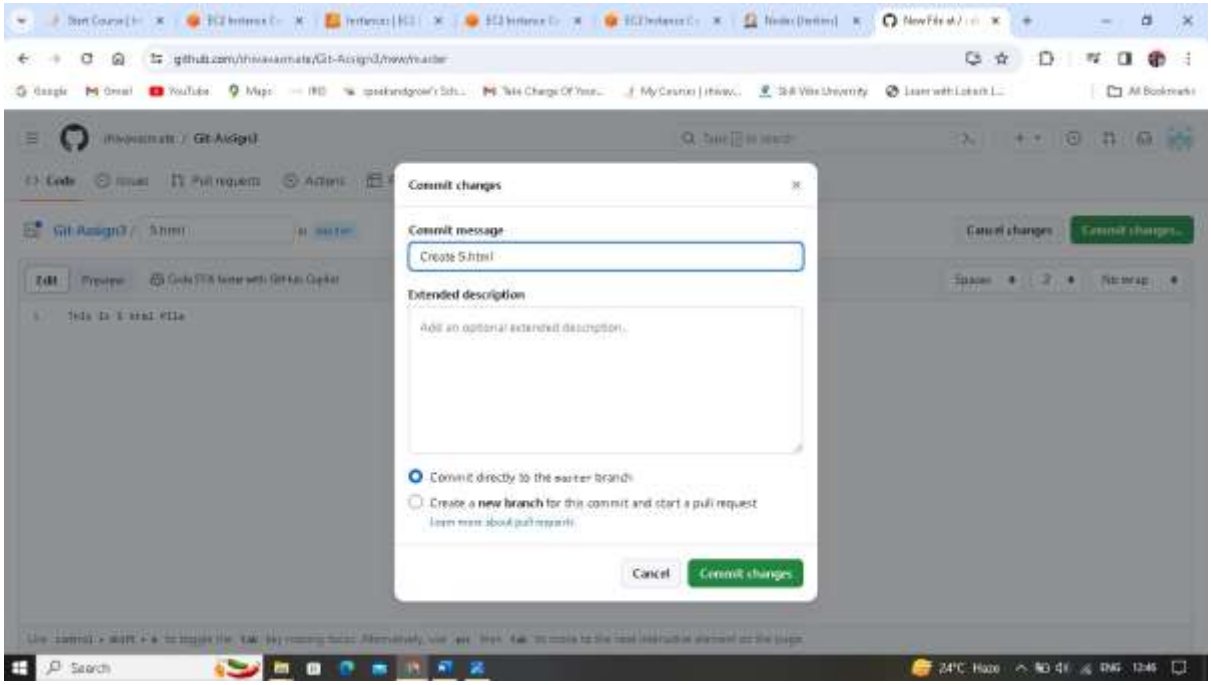
13. Now add a new file 1.html and click on Commit changes.



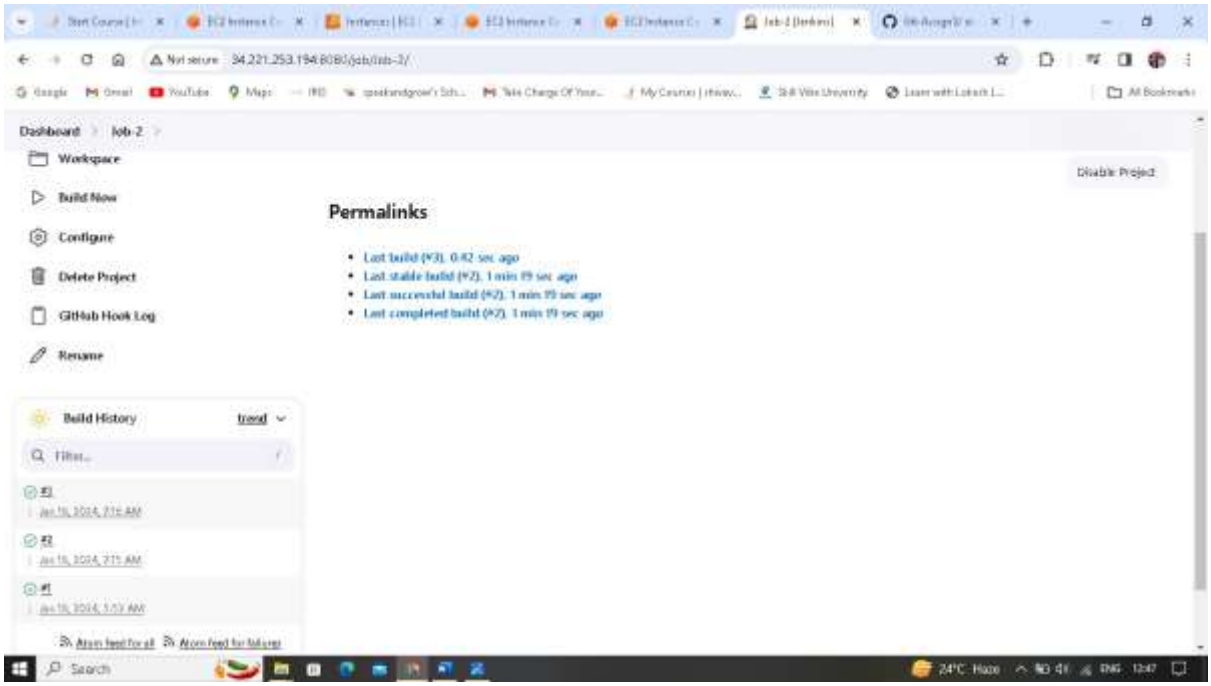
14.#2 build is created automatically



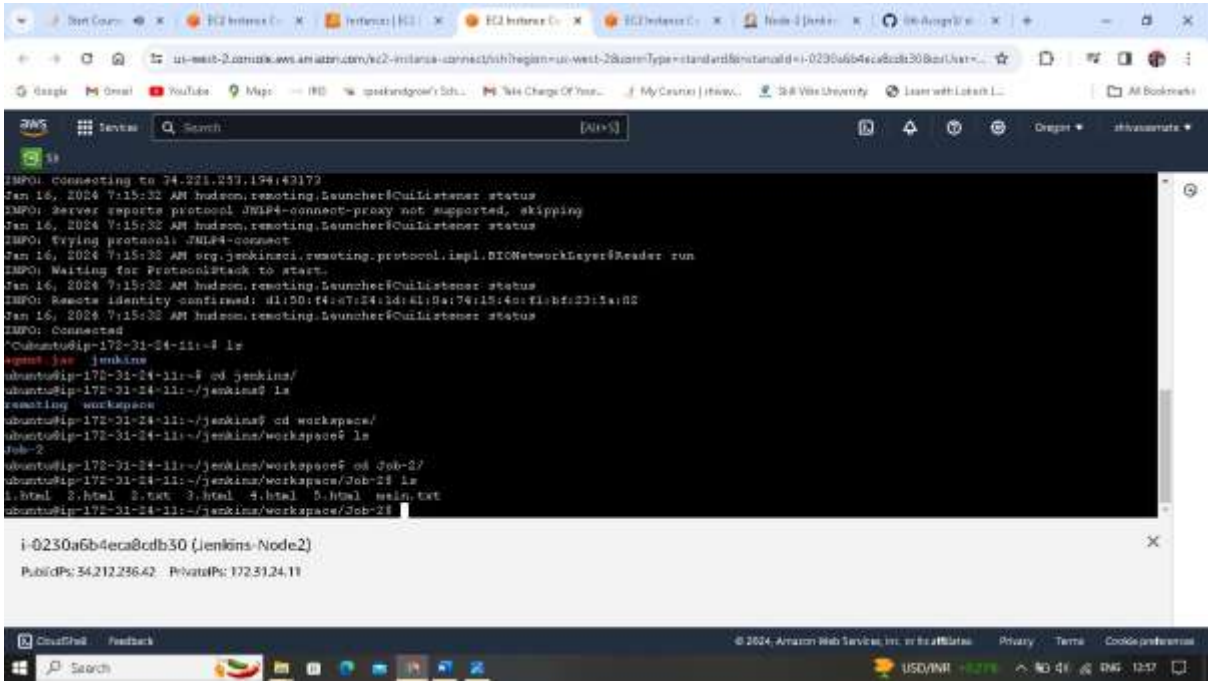
15. Similarly For 5.html file



16.#3 build also created automatically

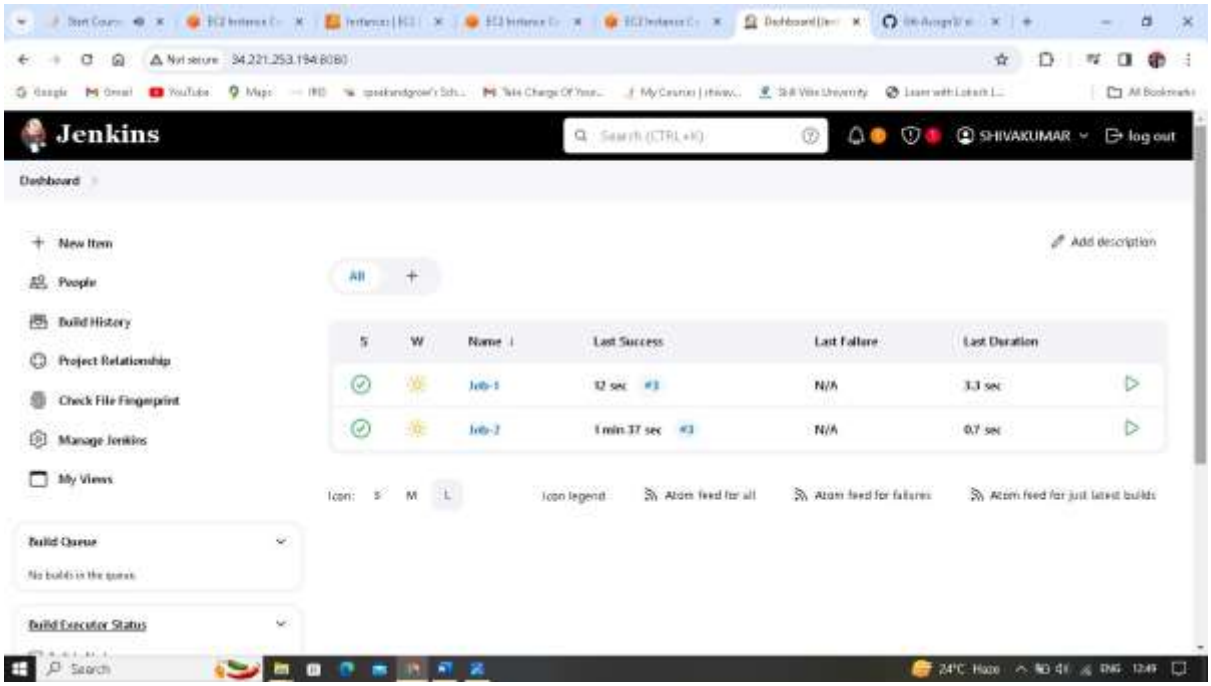


17. Go to Jenkins-Node 2 check for 1.html and 5.html files in the Job-2 folder.

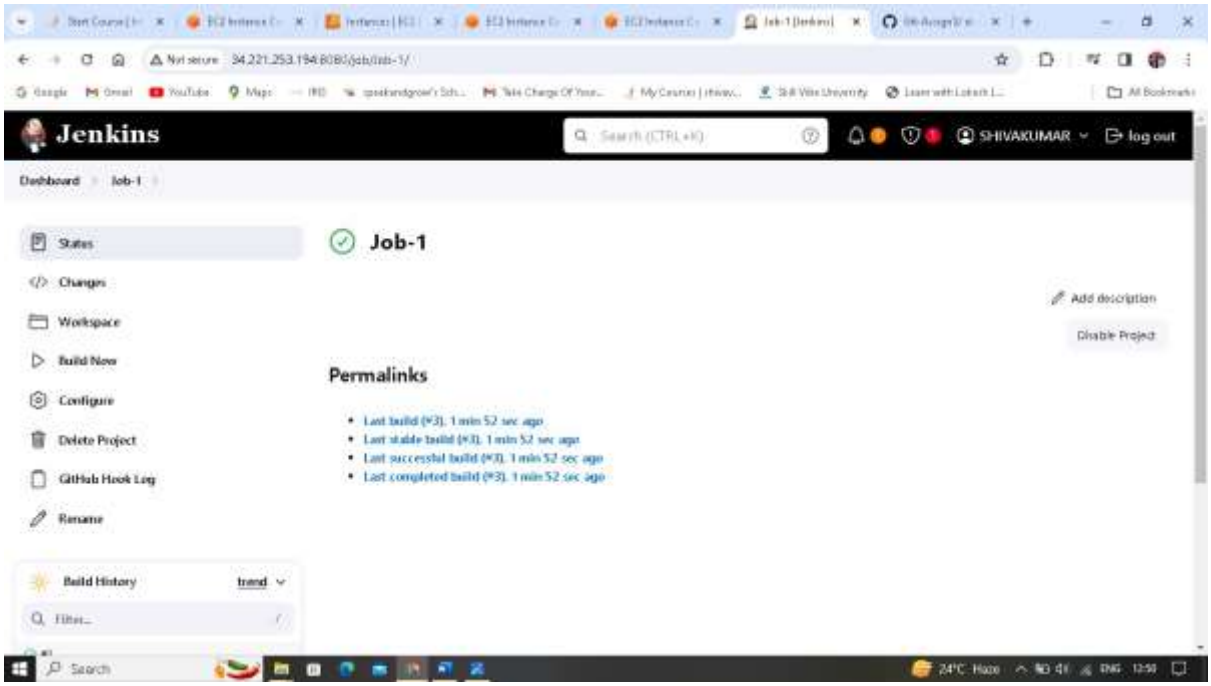


Job3 → Job1-Job2 (combine Jobs)

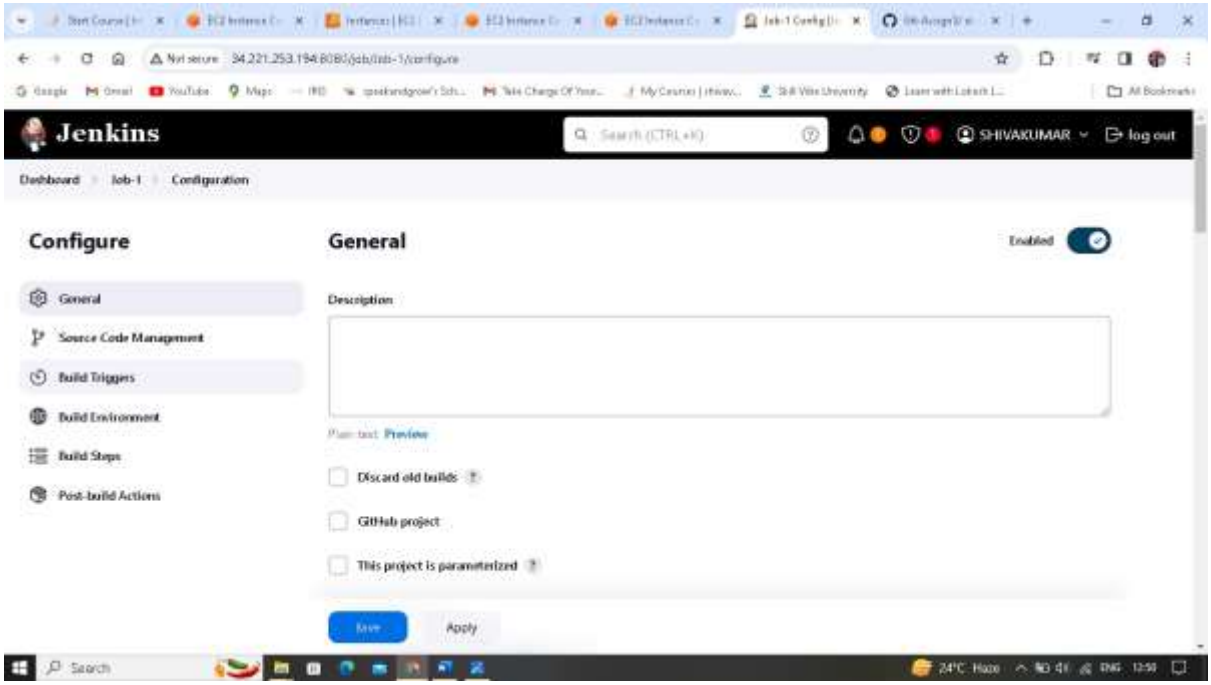
1. Go back to Dashboard and click on Job-1



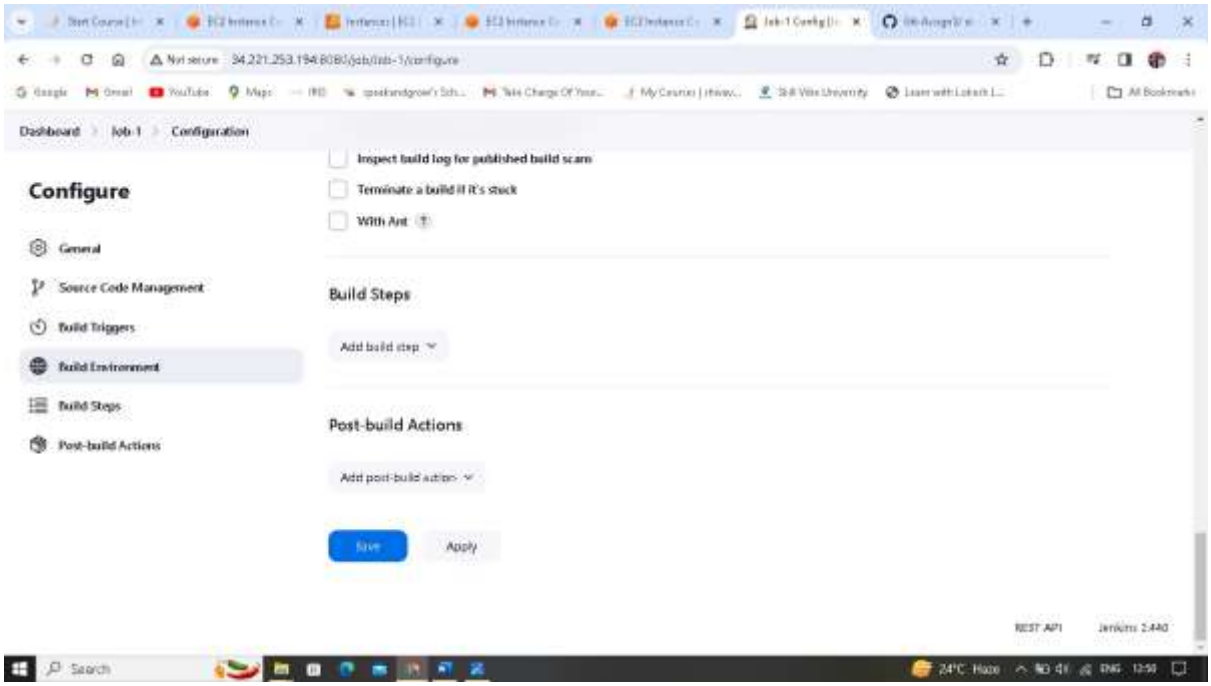
2. Now click on configure



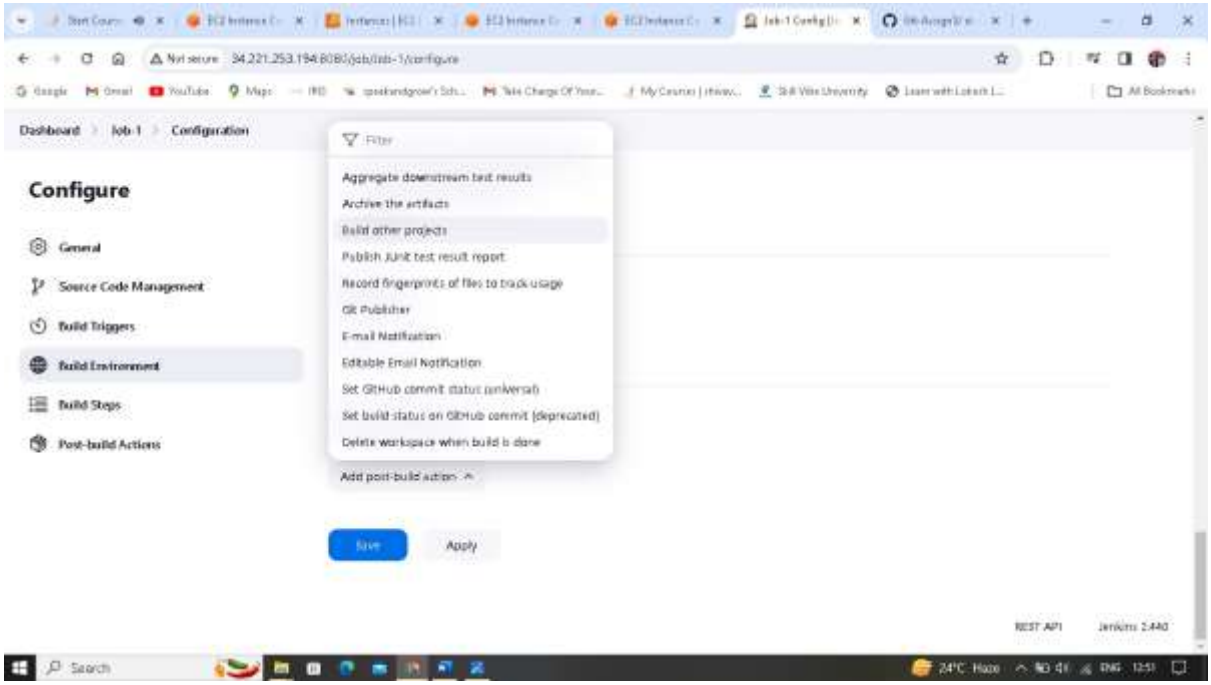
3. Go at the end



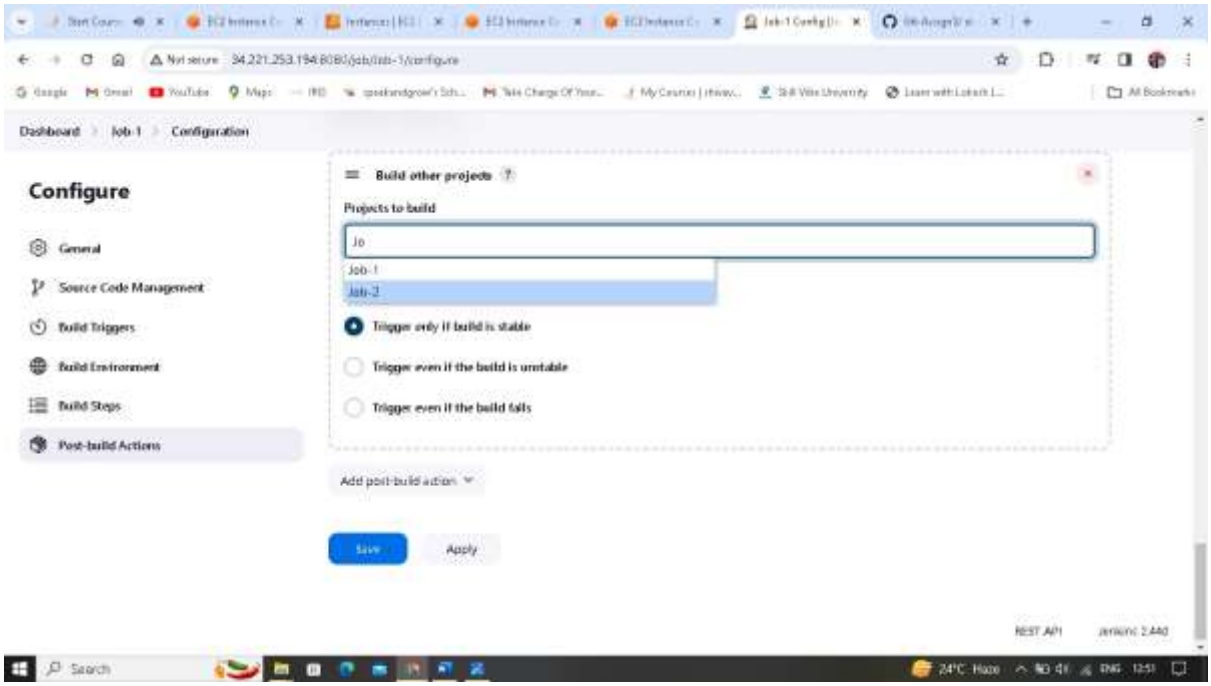
4. Under Post-build Actions



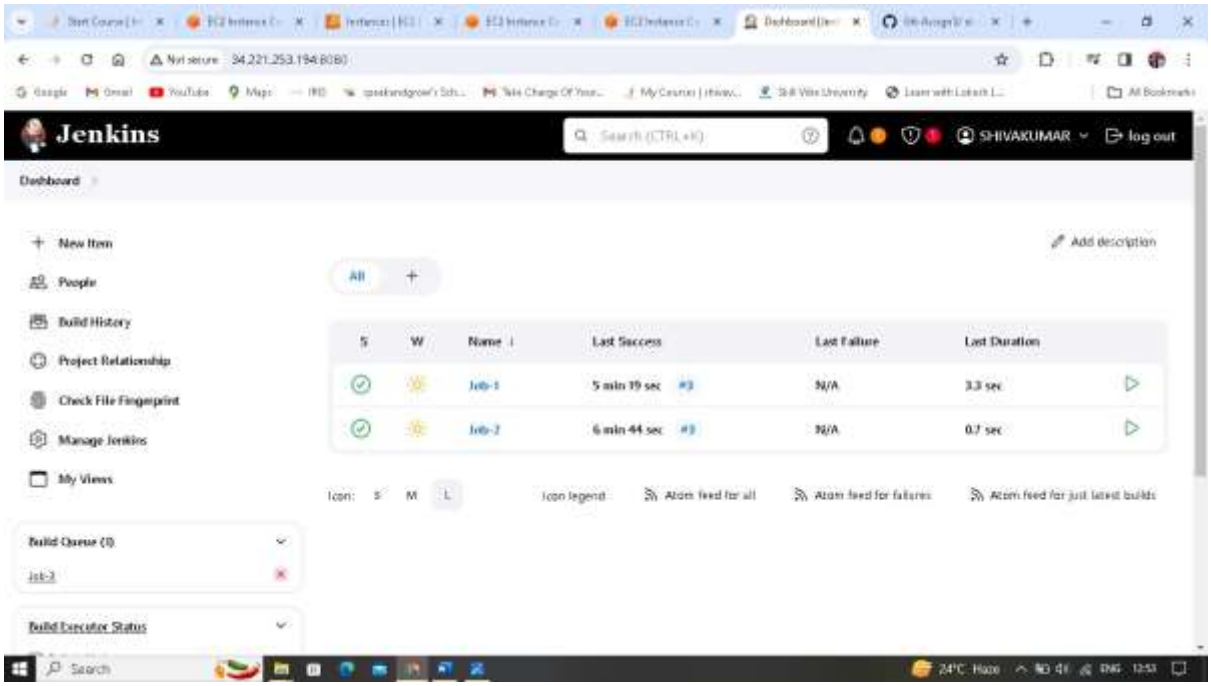
5. Choose Build other project (3rd)



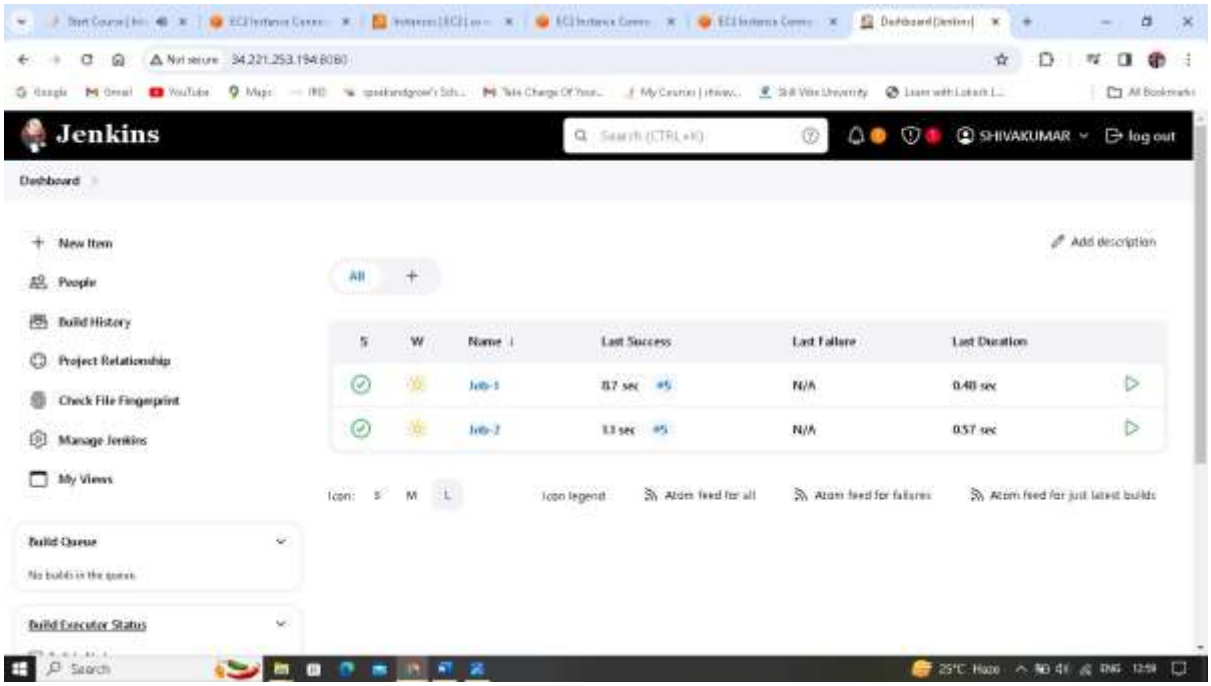
6. In project to build choose Job-2 and Trigger only if build is stable.



7.Go to Job-1 click on build and on the left side under Build Queue Job-2 appear.

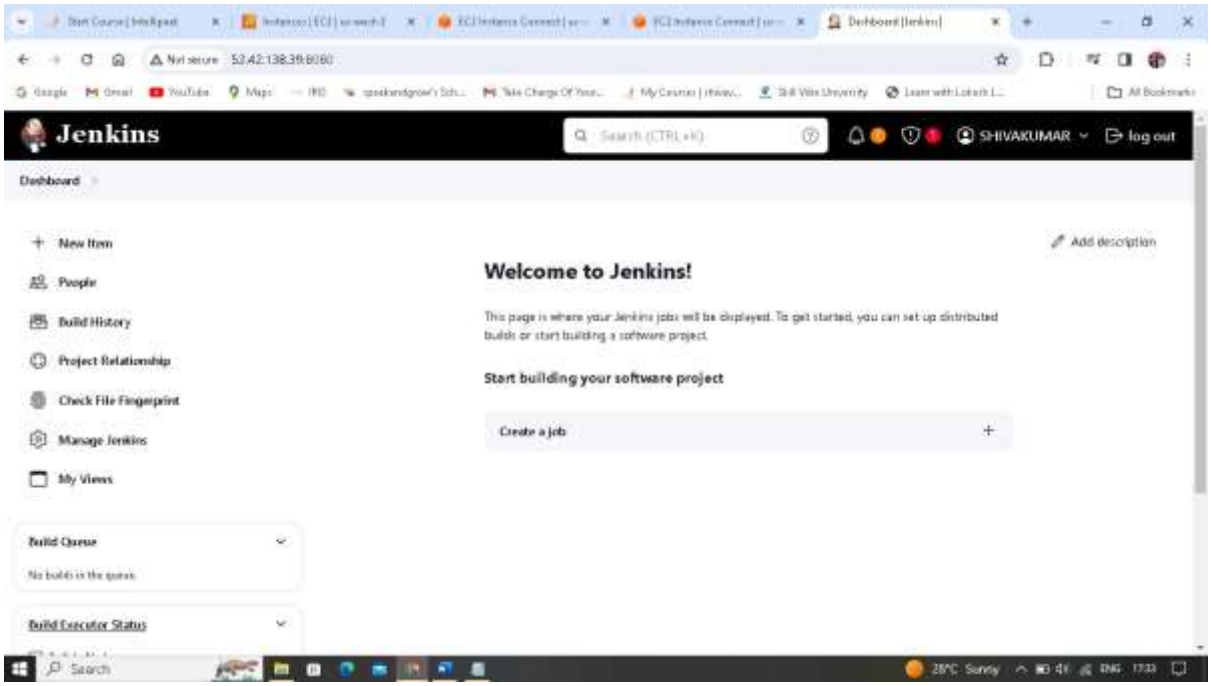


8. Check the last duration both nearly done build at same time for #5 build.

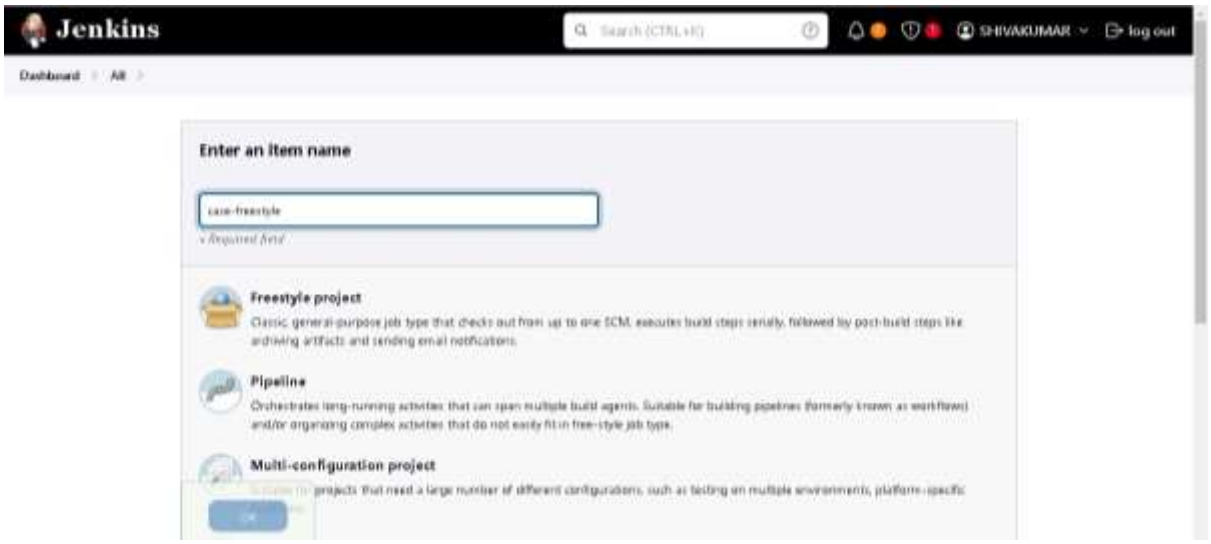

Jenkins
SHIVAKUMAR
log out
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
job-1
0.7 sec #5
N/A
0.48 sec
job-2
1.1 sec #5
N/A
0.57 sec
Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds
Build Queue
No builds in the queue.
Build Executor Status
Search

I. FOR FREESTYLE

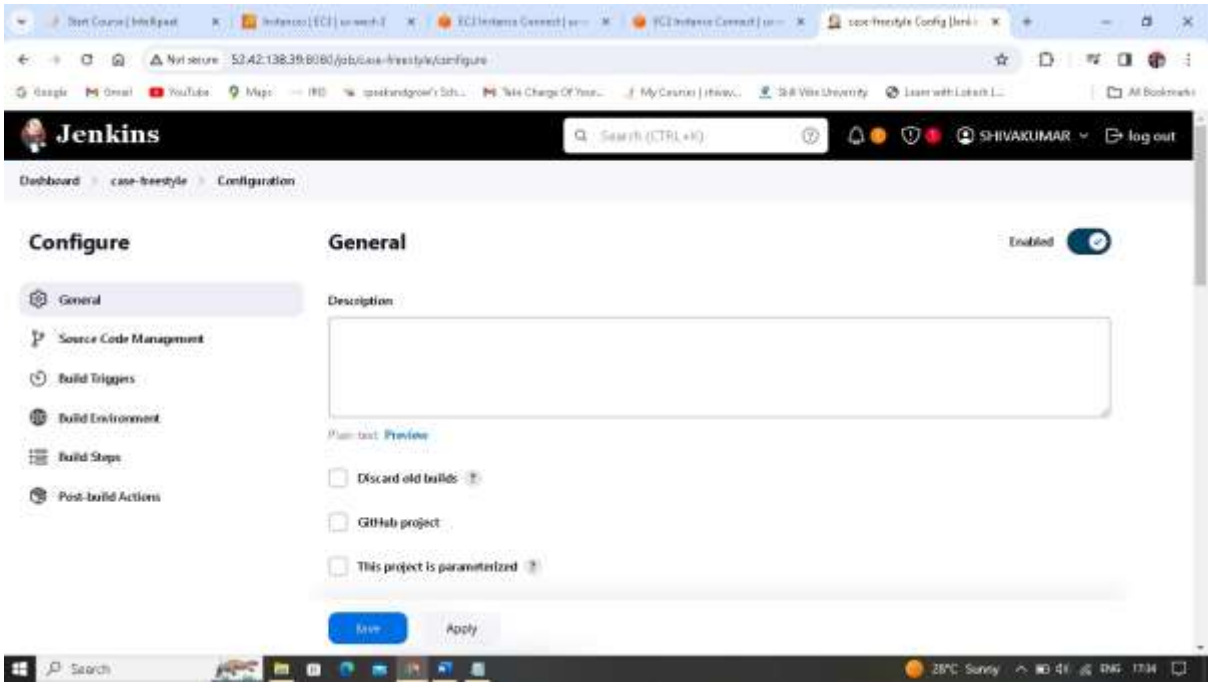
1. Go to Dashboard click on New Item.



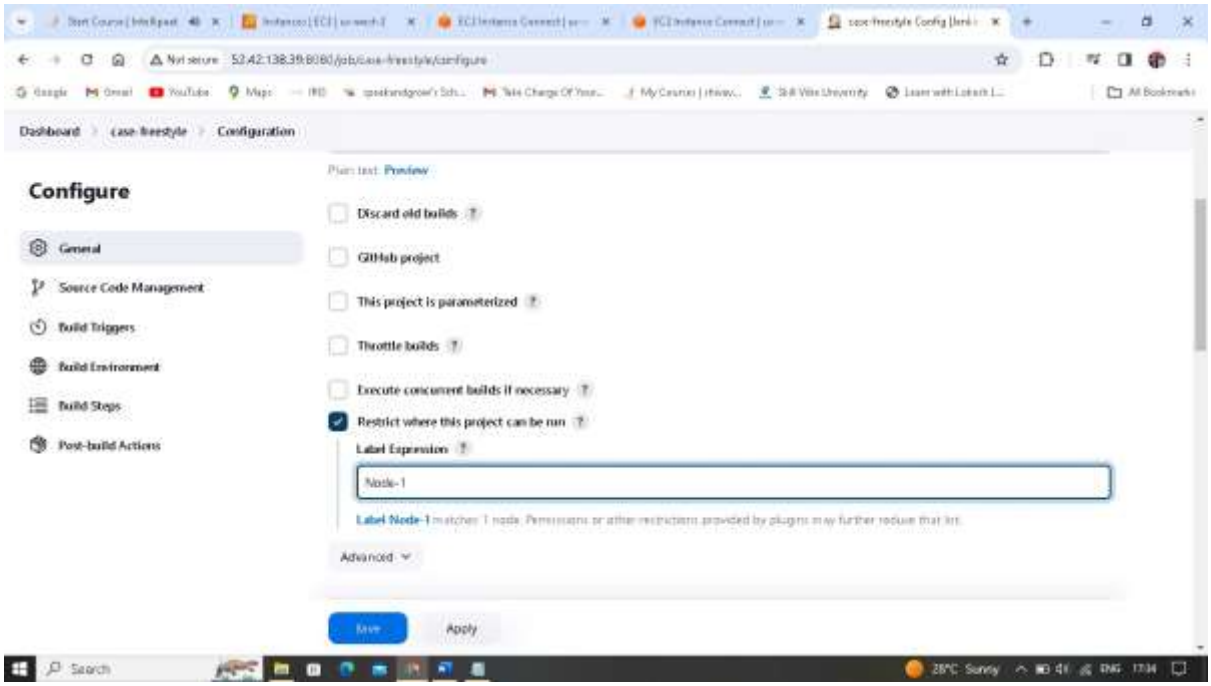
2.Enter an item name as “case-freestyle” and click on ok



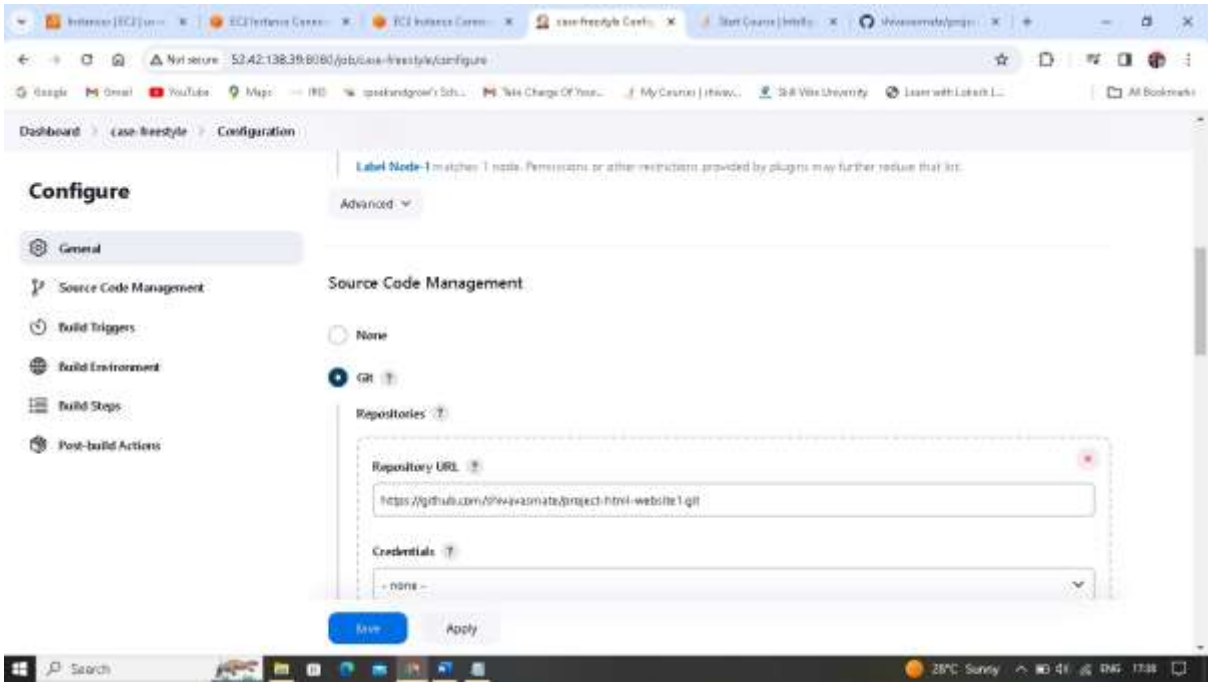
3. Fill it



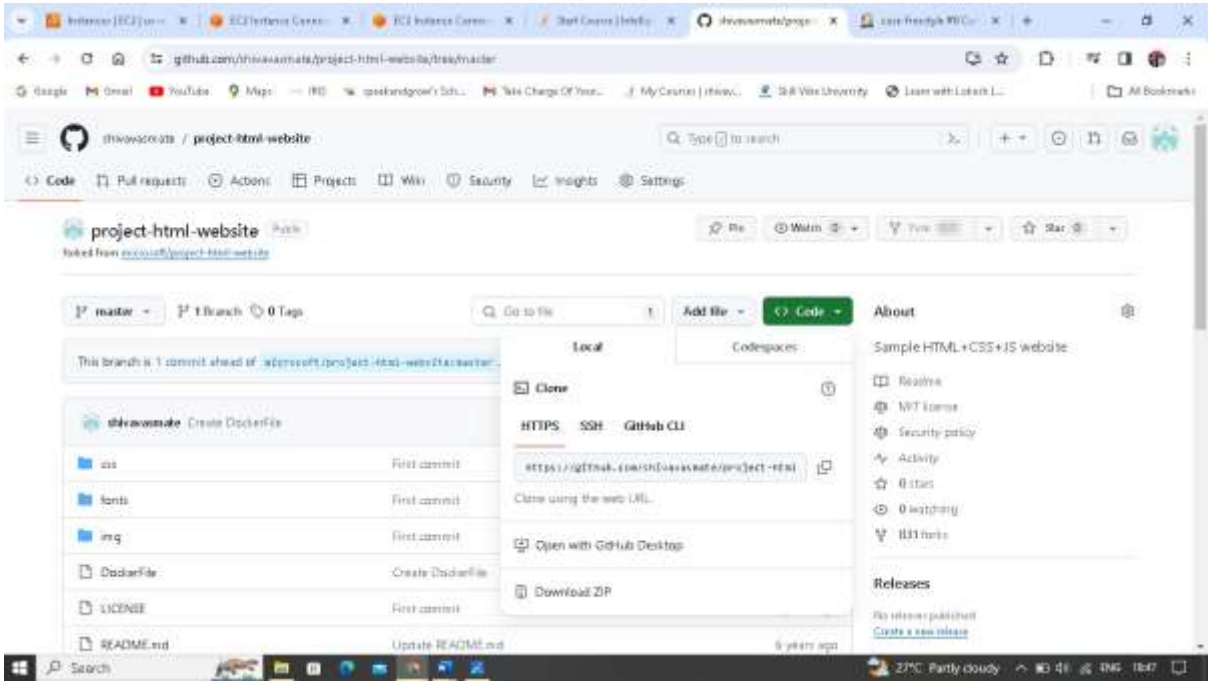
4. First of all check in Restrict where this project can be run. Label Expression as Node-1



5. Under Source Code Management Check Git



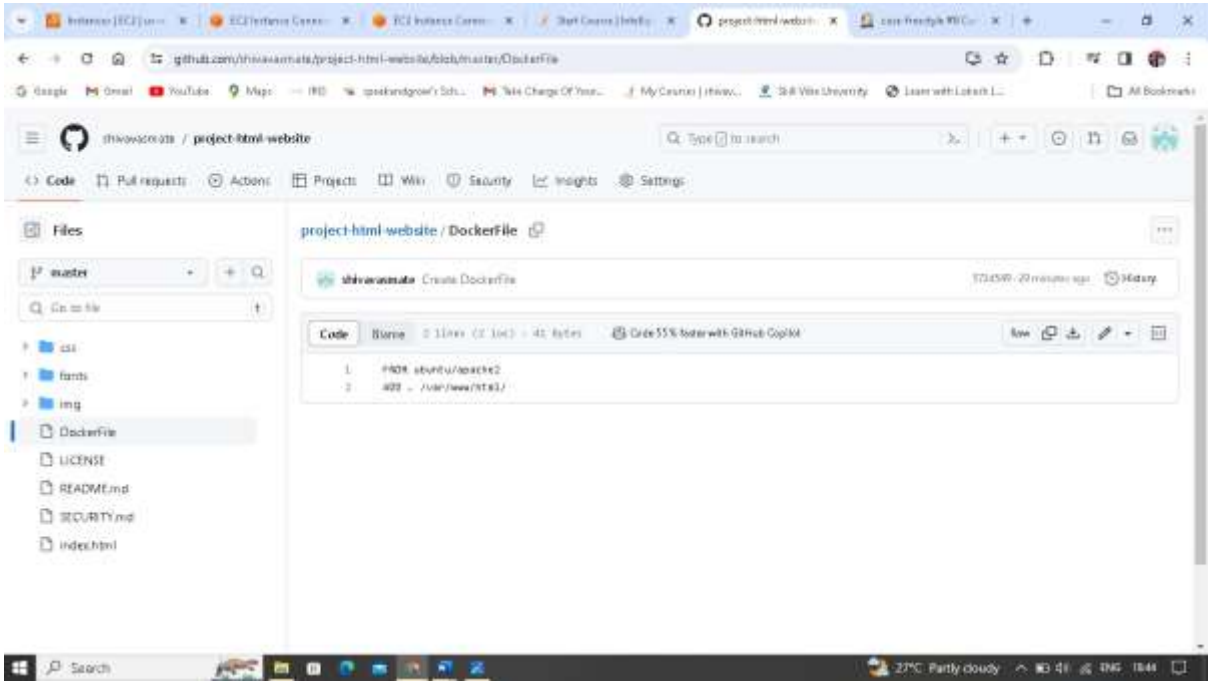
6.Now log in to your GitHub in another browser



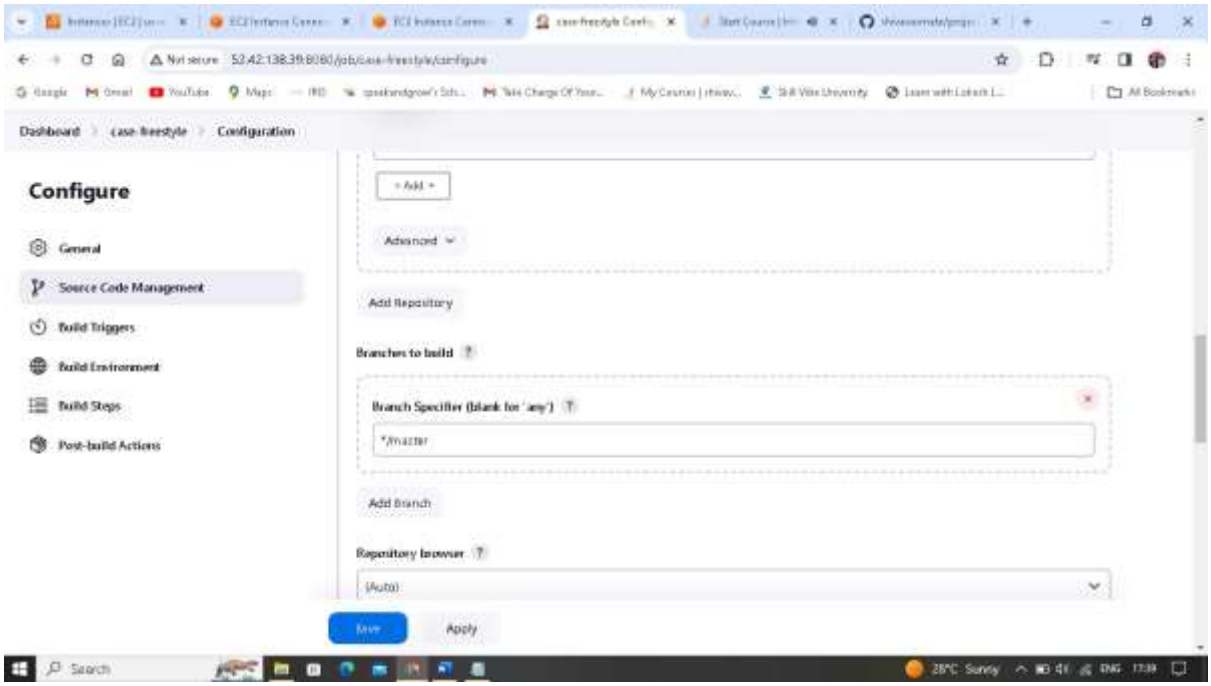
7. Create DockerFile with command

FROM ubuntu/apache2

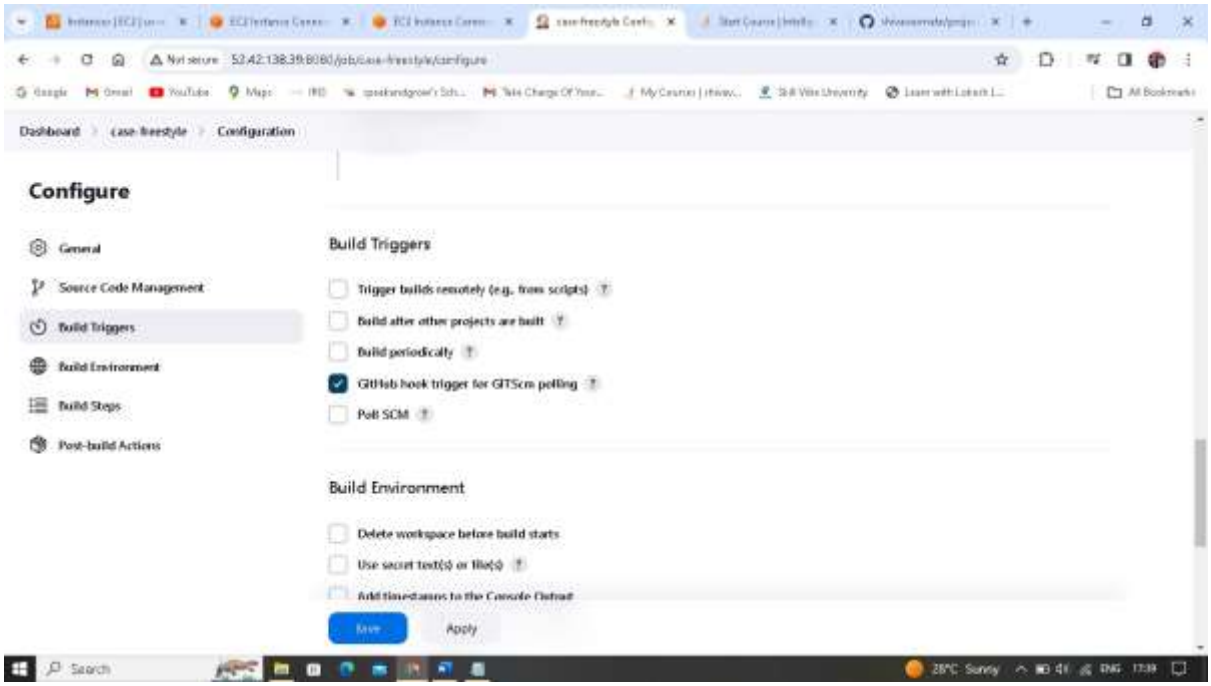
ADD . /var/www/html/ command changes.



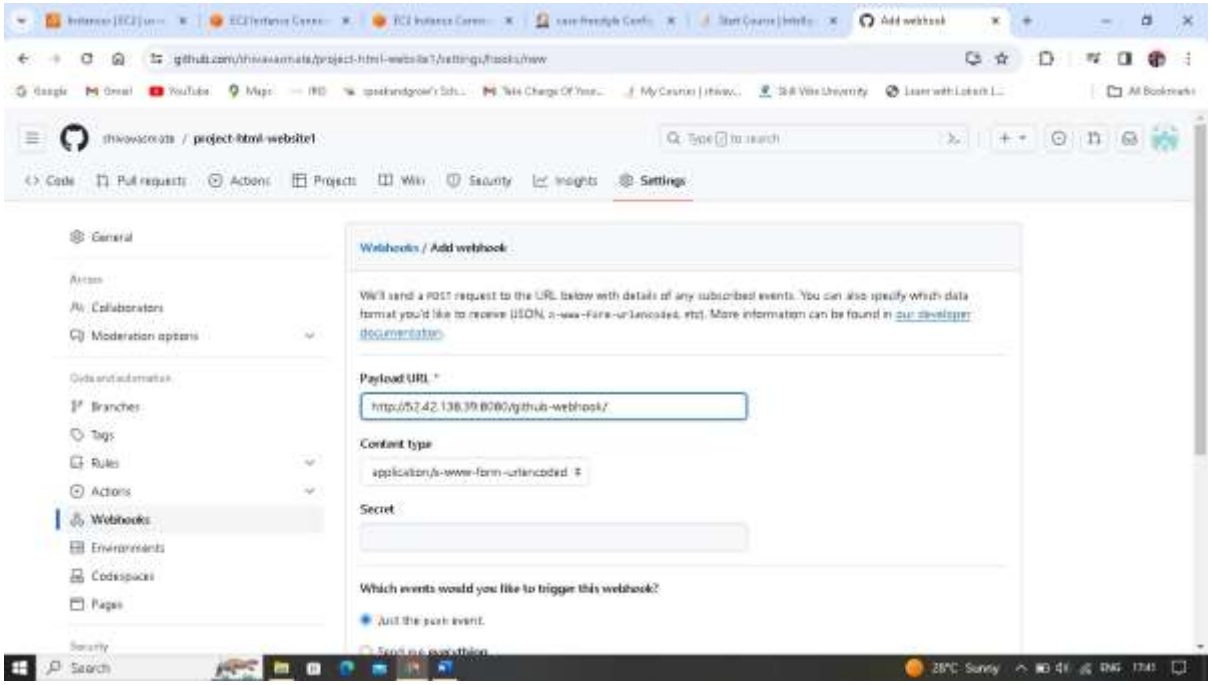
8. In Branches to build and Branch Specifier give */master



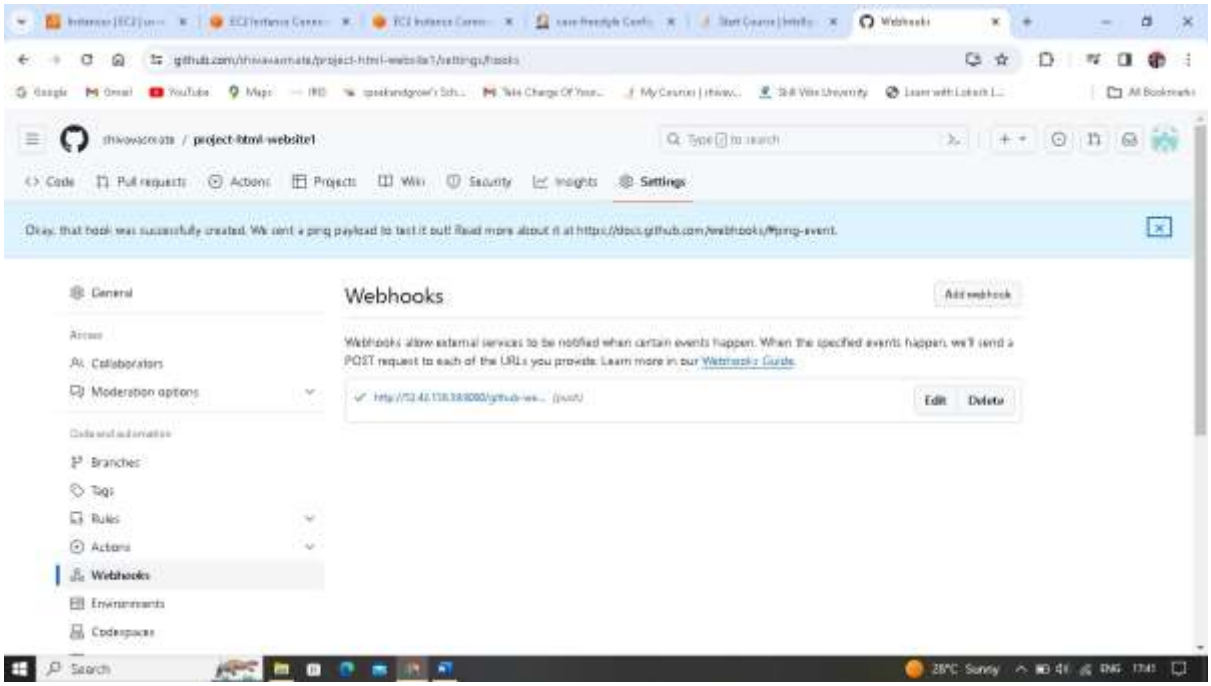
9. Under Build Triggers choose GitHub hock trigger for GIT Scm polling.



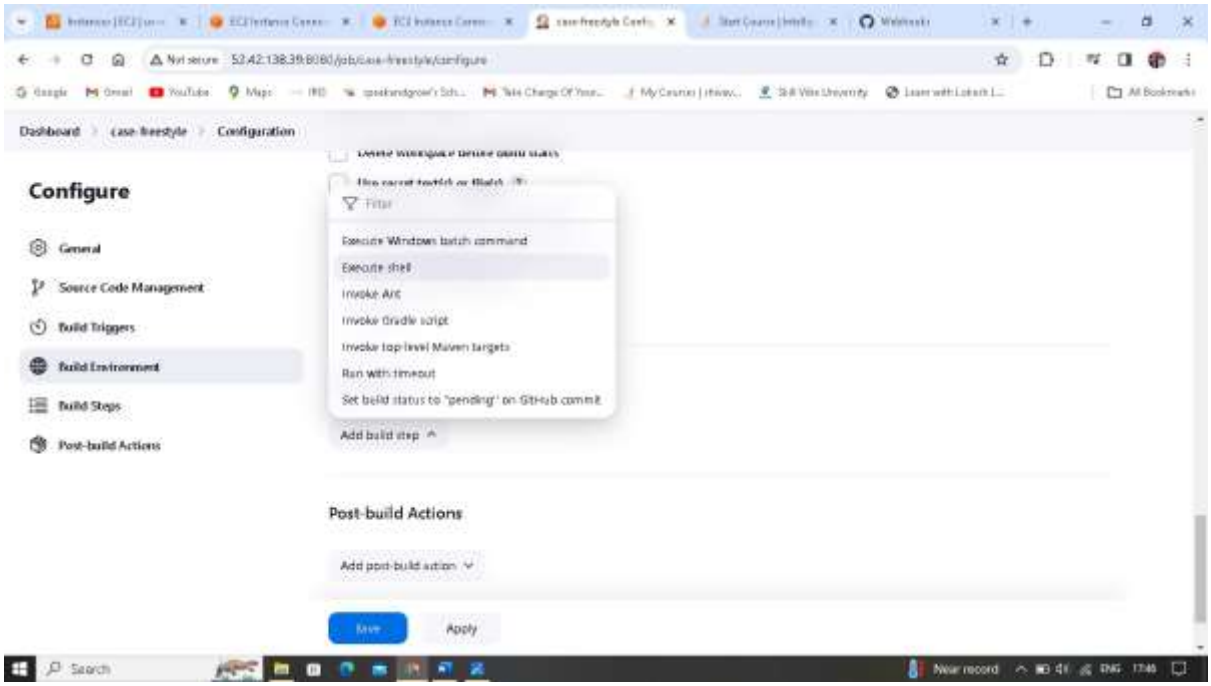
10.Now go to GitHub under repo settings look for Webhooks on left side



11. Check for right mark



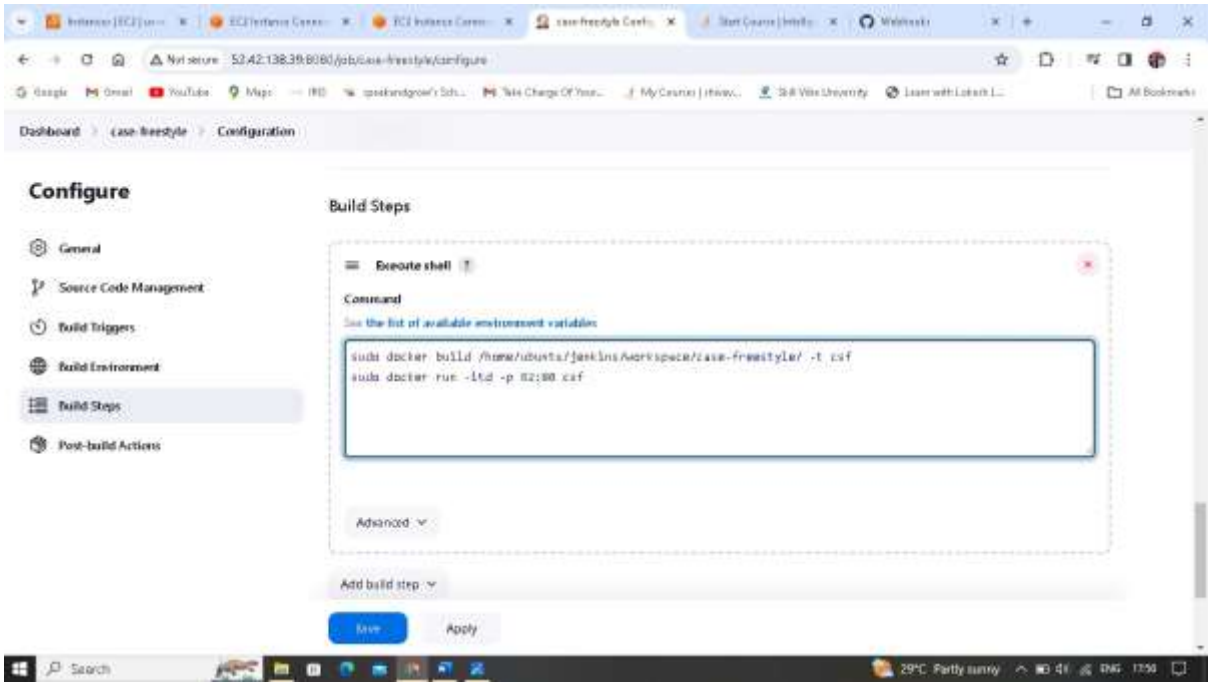
12. Under Build Steps click on Execute shell



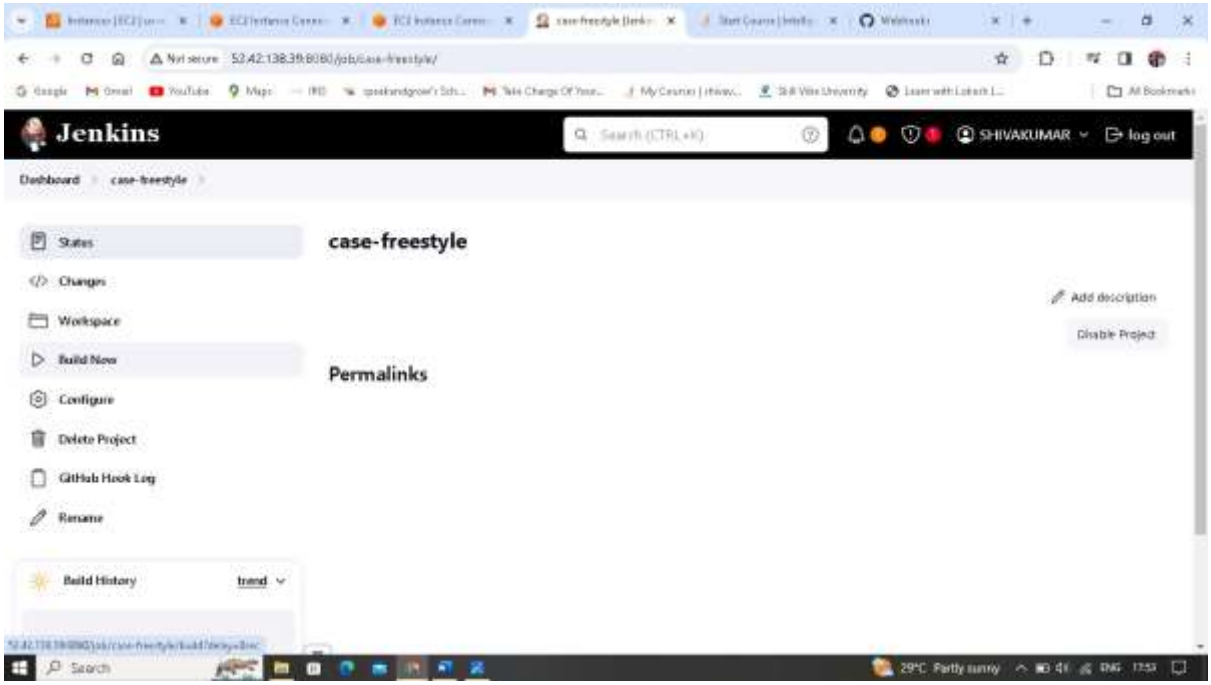
13. Under Command type

sudo docker build /home/ubuntu/Jenkins/workspace/case-freestyle/ -t csf

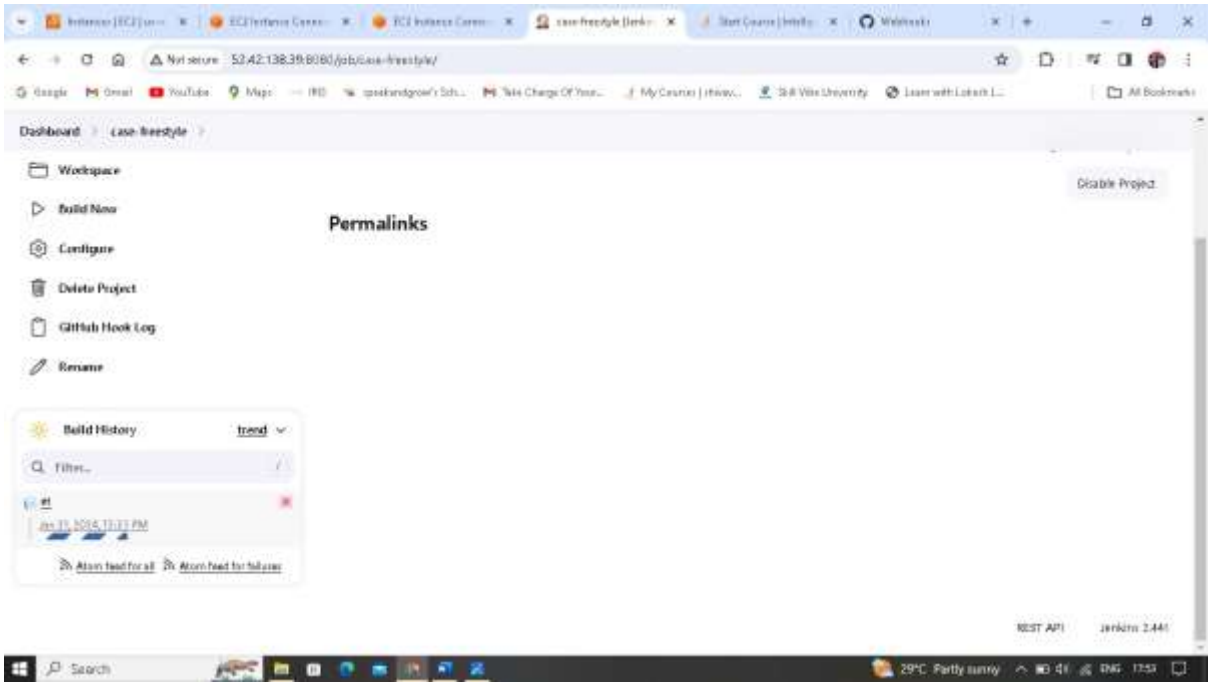
sudo docker run -itd -p 82:80 csf and click on save.



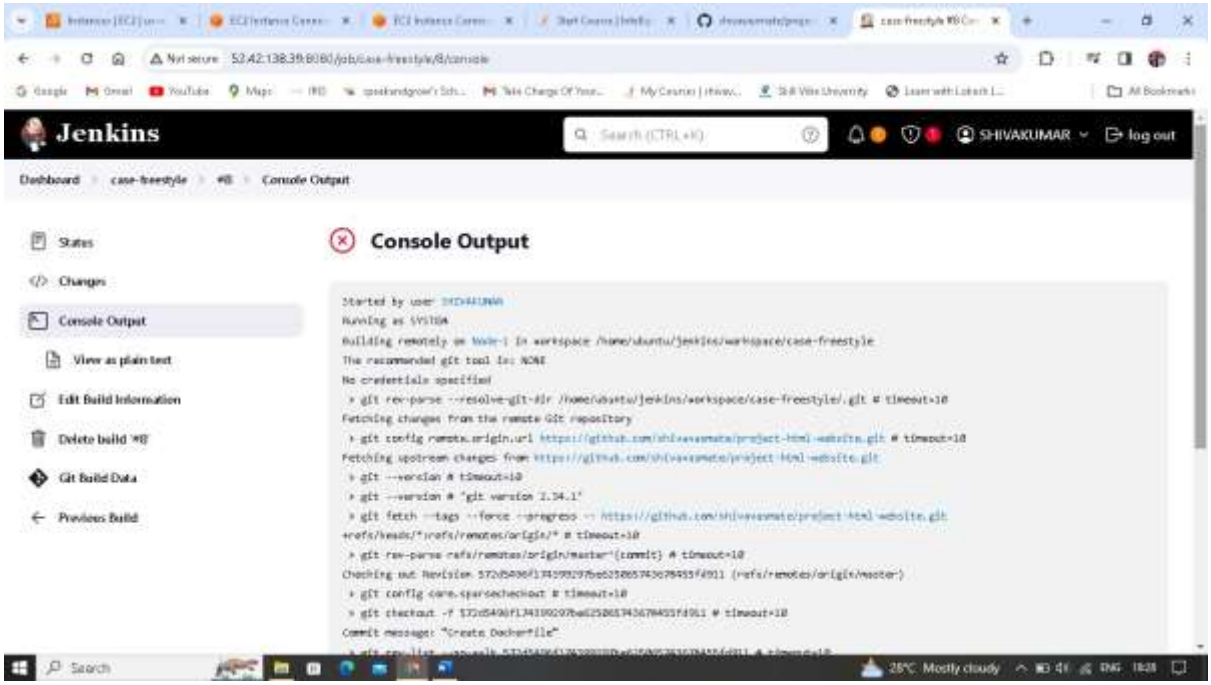
14.The case-freestyle is created



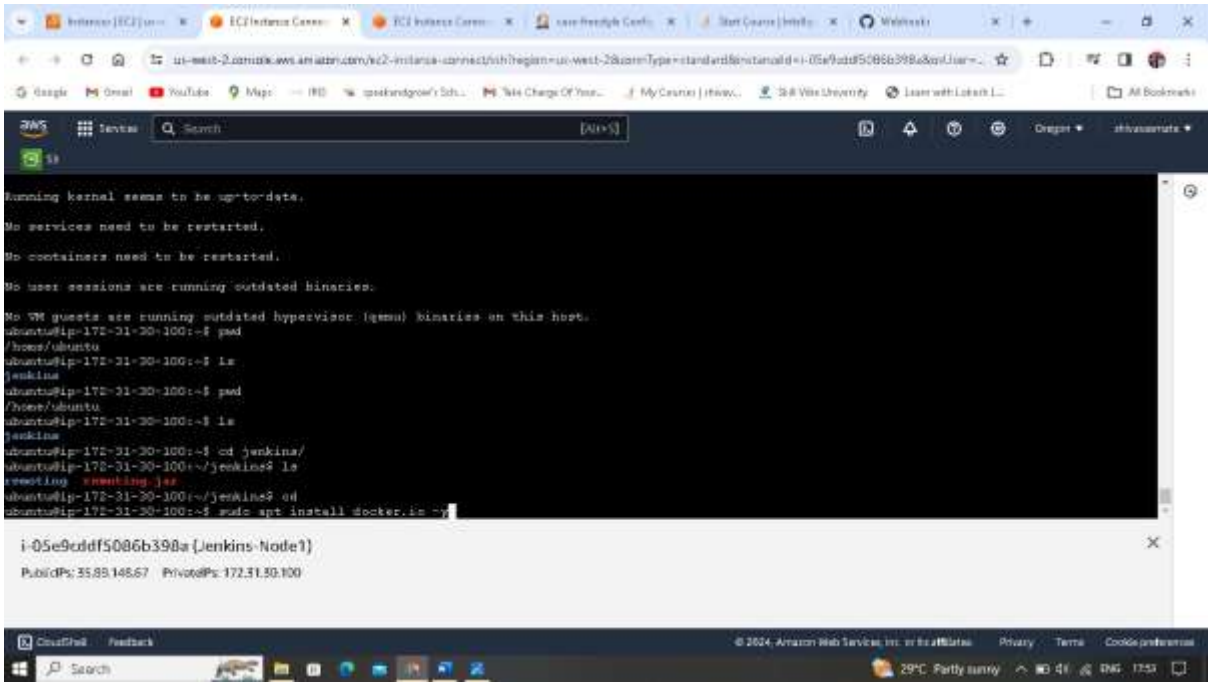
15. Click on Build Now



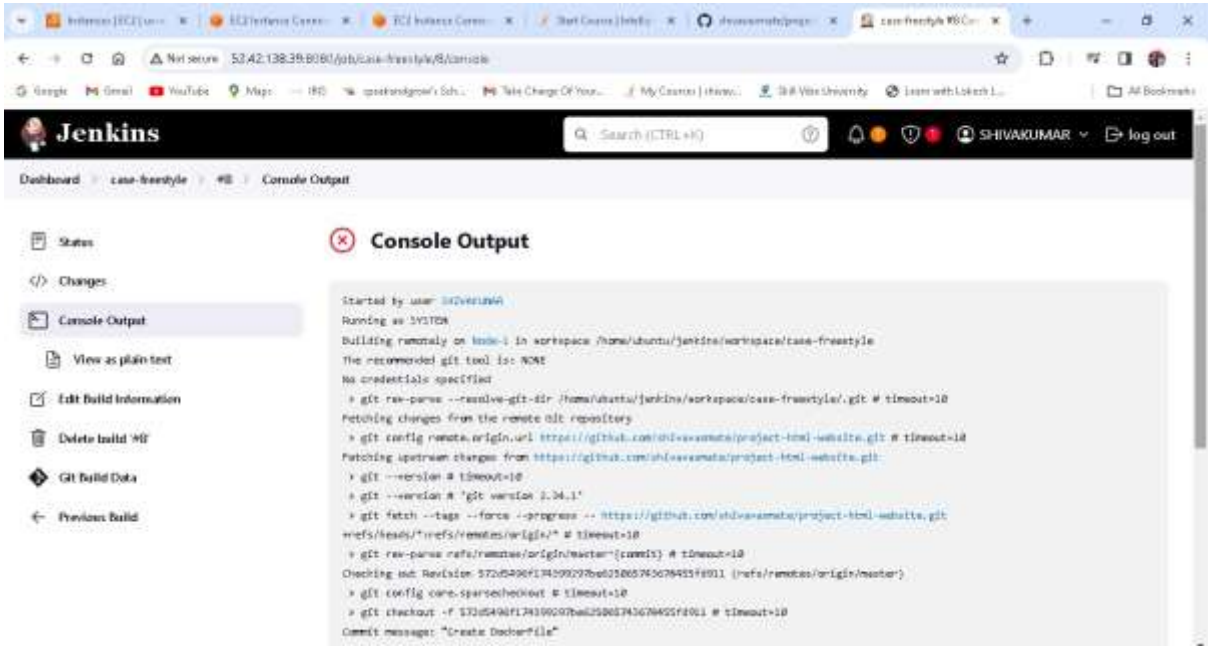
16. Check for error

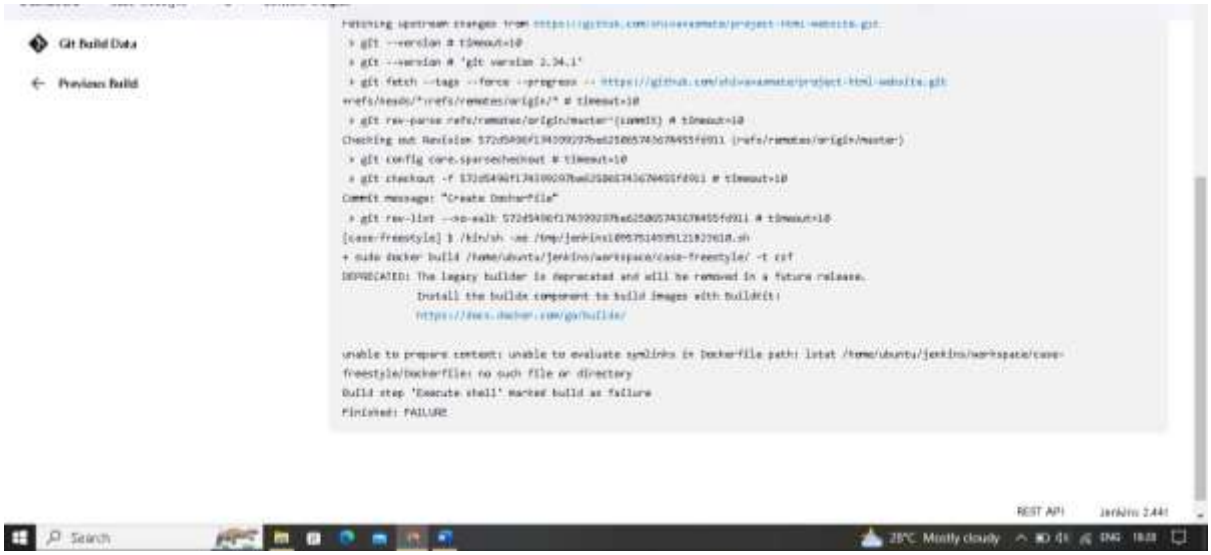


17. Install docker in Jenkins-Node1 using the command sudo apt install docker.io -y



18. Build now again and check for error





19. Error is

unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/ubuntu/jenkins/workspace/case-freestyle/Dockerfile: no such file or directory

Build step 'Execute shell' marked build as failure

Finished: FAILURE.

20. Go to Jenkins-Node1 check for it.

ubuntu@ip-172-31-30-100:~$ ls

jenkins

ubuntu@ip-172-31-30-100:~$ cd jenkins/

ubuntu@ip-172-31-30-100:~/jenkins$ ls

remoting remoting.jar workspace

ubuntu@ip-172-31-30-100:~/jenkins$ cd workspace/

ubuntu@ip-172-31-30-100:~/jenkins/workspace$ ls

case-freestyle

ubuntu@ip-172-31-30-100:~/jenkins/workspace$ cd case-freestyle/

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-freestyle$ ls

DockerFile LICENSE README.md SECURITY.md css fonts img index.html

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-freestyle$.

RESOLVING ERROR

21. $ mv DockerFile Dockerfile

a. Change the DockerFile into Dockerfile

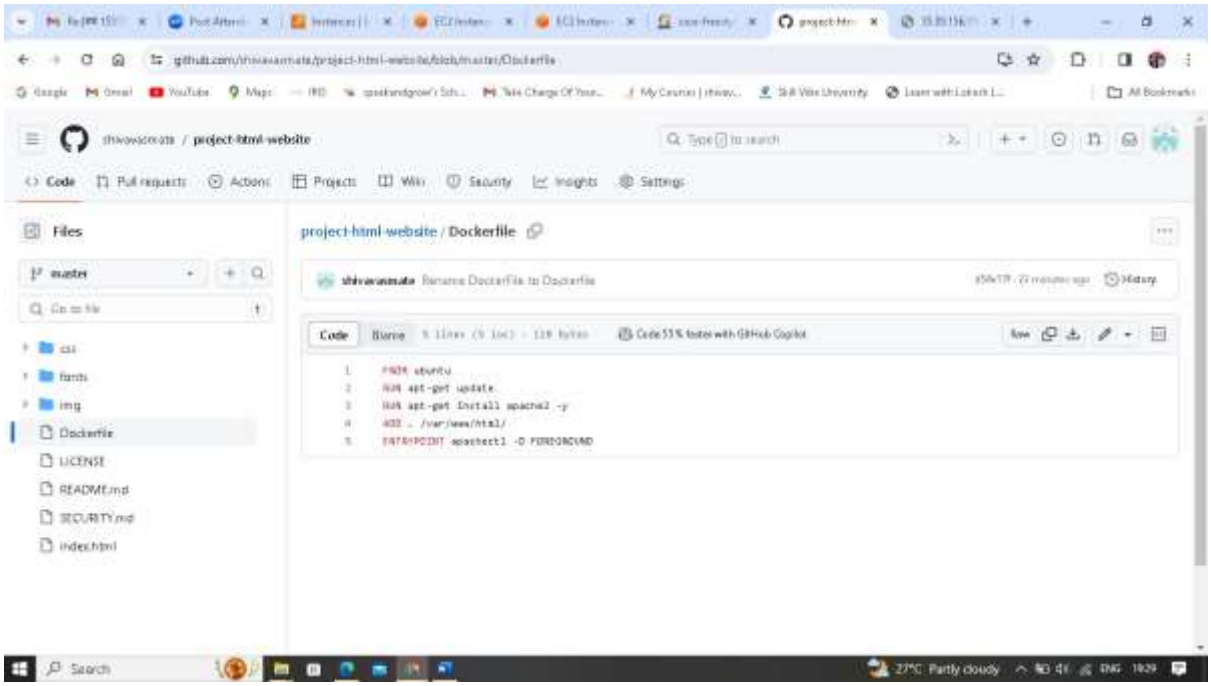
b. Update Dockerfile as
FROM ubuntu

RUN apt-get update

RUN apt-get install apache2 -y

ADD . /var/www/html/

ENTRYPOINT apachectl -D FOREGROUND



ubuntu@ip-172-31-30-100:~$ ls

jenkins

ubuntu@ip-172-31-30-100:~$ cd jenkins/

ubuntu@ip-172-31-30-100:~/jenkins$ ls

caches remoting remoting.jar workspace

ubuntu@ip-172-31-30-100:~/jenkins$ cd workspace/

ubuntu@ip-172-31-30-100:~/jenkins/workspace$ ls

case-freestyle case-pipeline case-pipeline@tmp

ubuntu@ip-172-31-30-100:~/jenkins/workspace$ cd case-freestyle/

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-freestyle$ ls

Dockerfile LICENSE README.md SECURITY.md css fonts img index.html

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-freestyle$ sudo cat Dockerfile

FROM ubuntu

RUN apt-get update

RUN apt-get install apache2 -y

ADD . /var/www/html/

ENTRYPOINT apachectl -D FOREGROUND

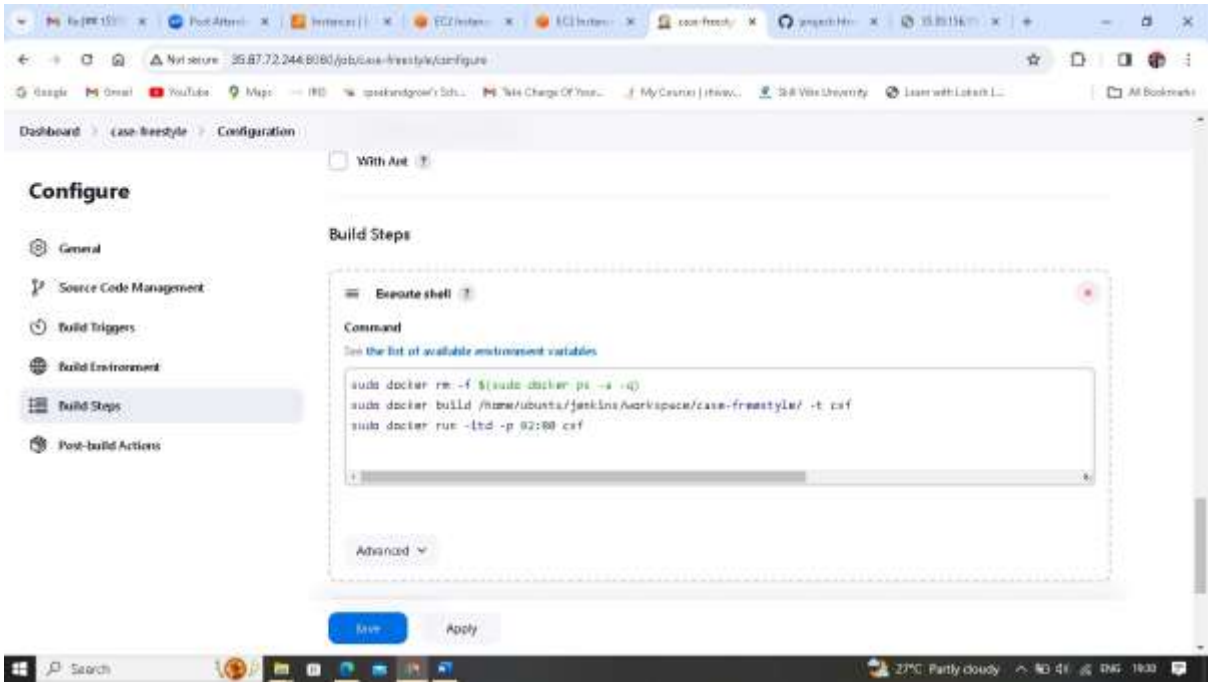
22. Update with command

sudo docker rm -f $(sudo docker ps -a -q)

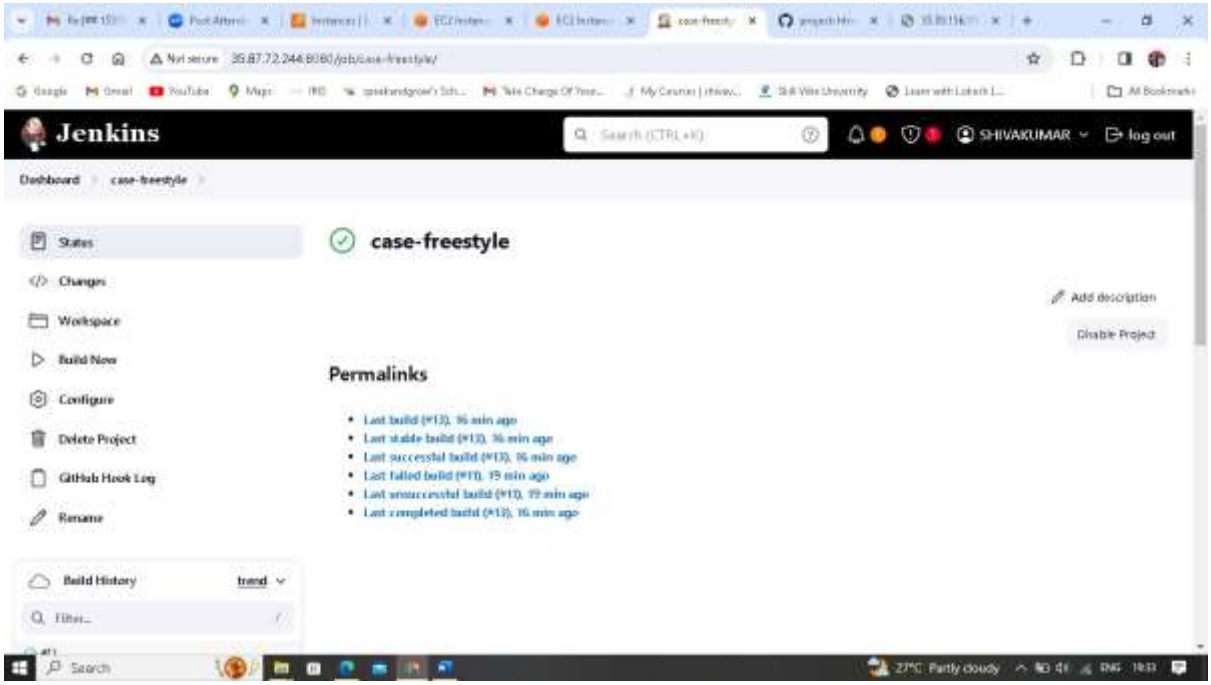
sudo docker build /home/ubuntu/jenkins/workspace/case-freestyle/ -t csf

sudo docker run -itd -p 82:80 csf

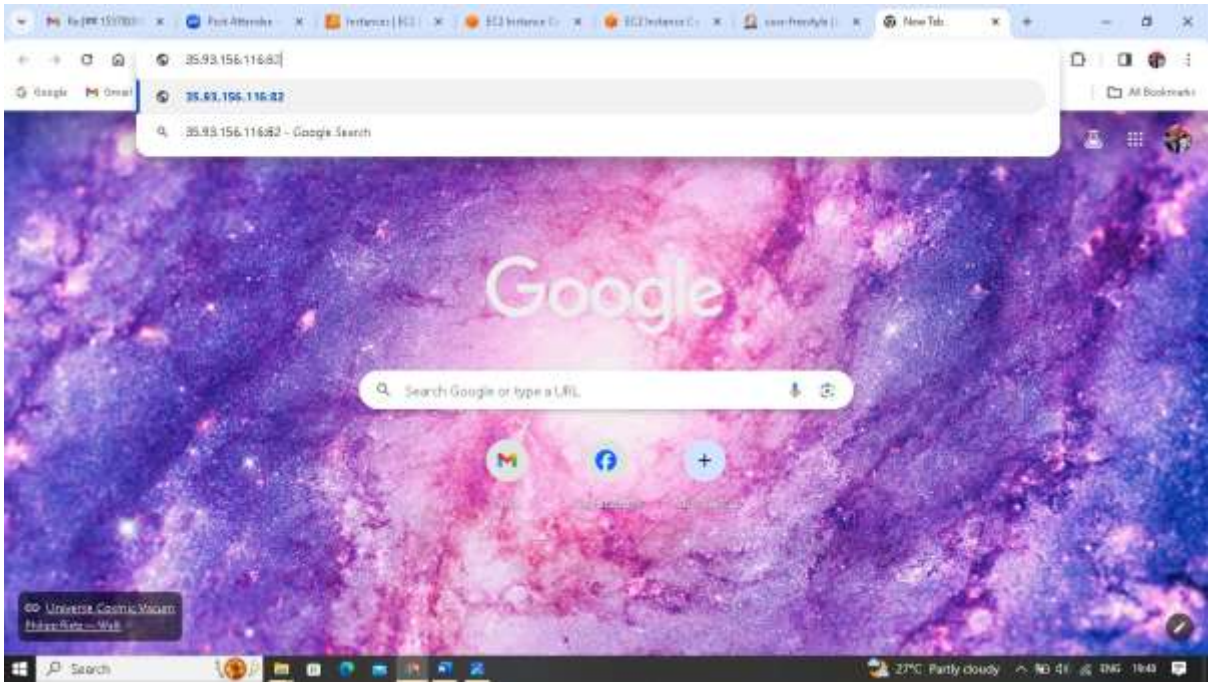
Click on apply and save.



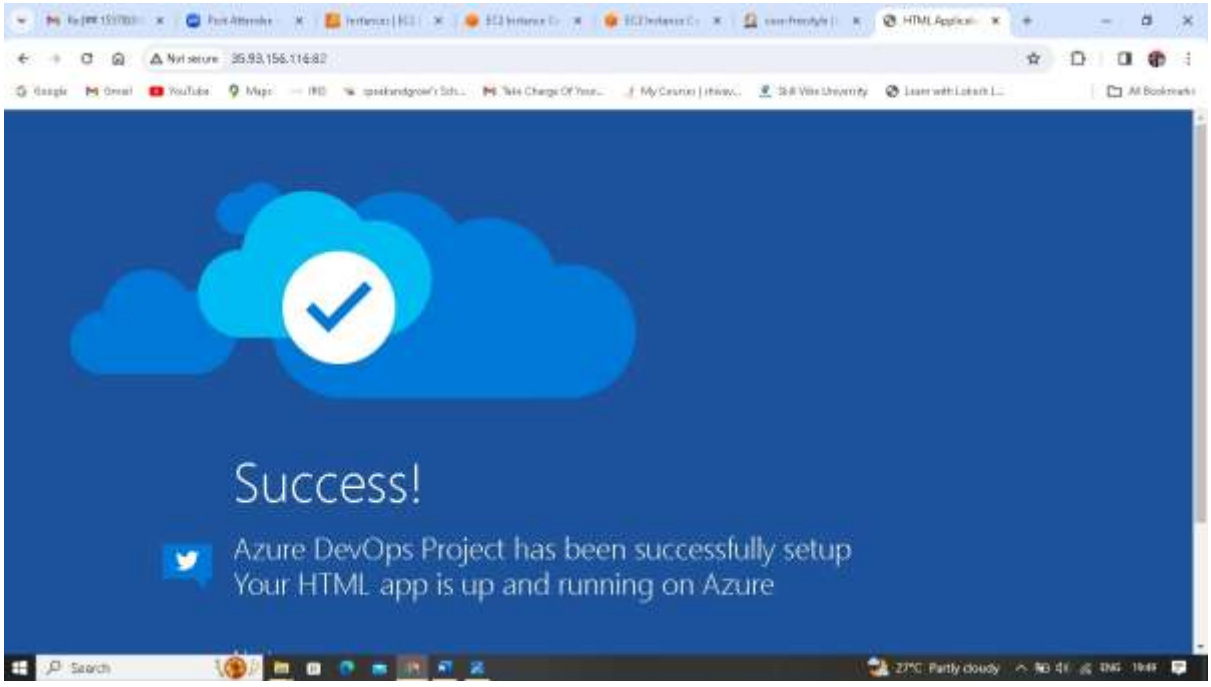
23. Click on Build Now. This time it is successful



24. Copy and paste the public IP with :82 in another browser.

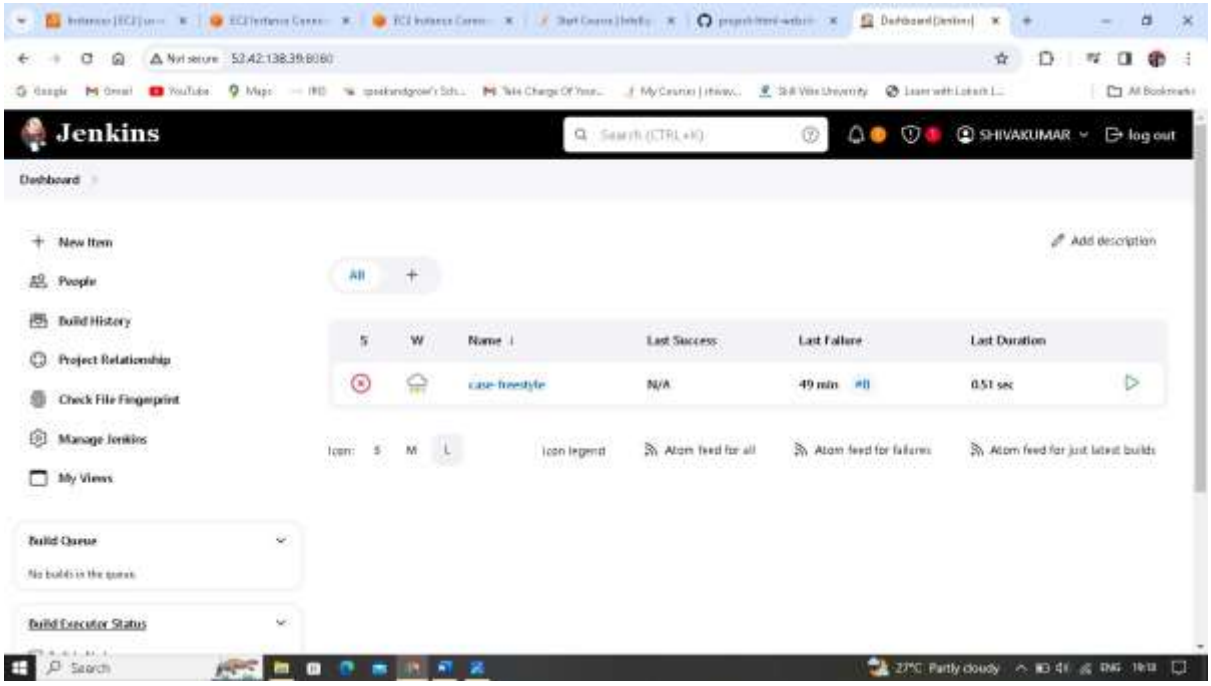


25. Now HTML Application opens.

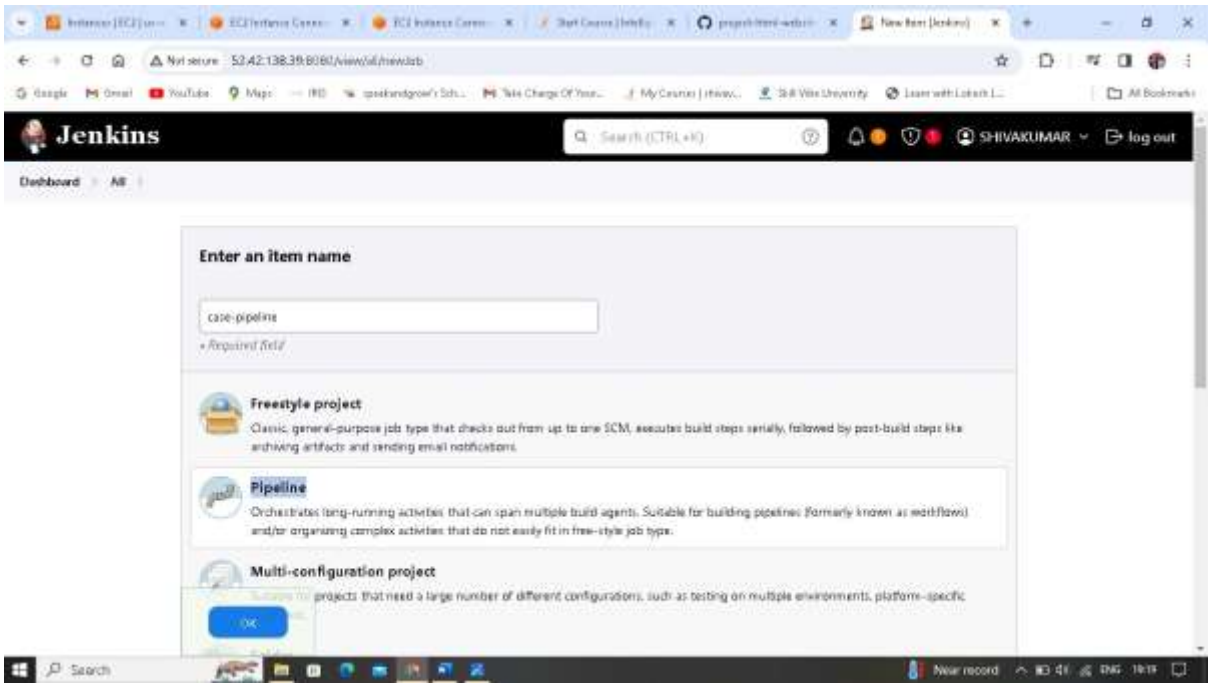


3. FOR PIPELINE

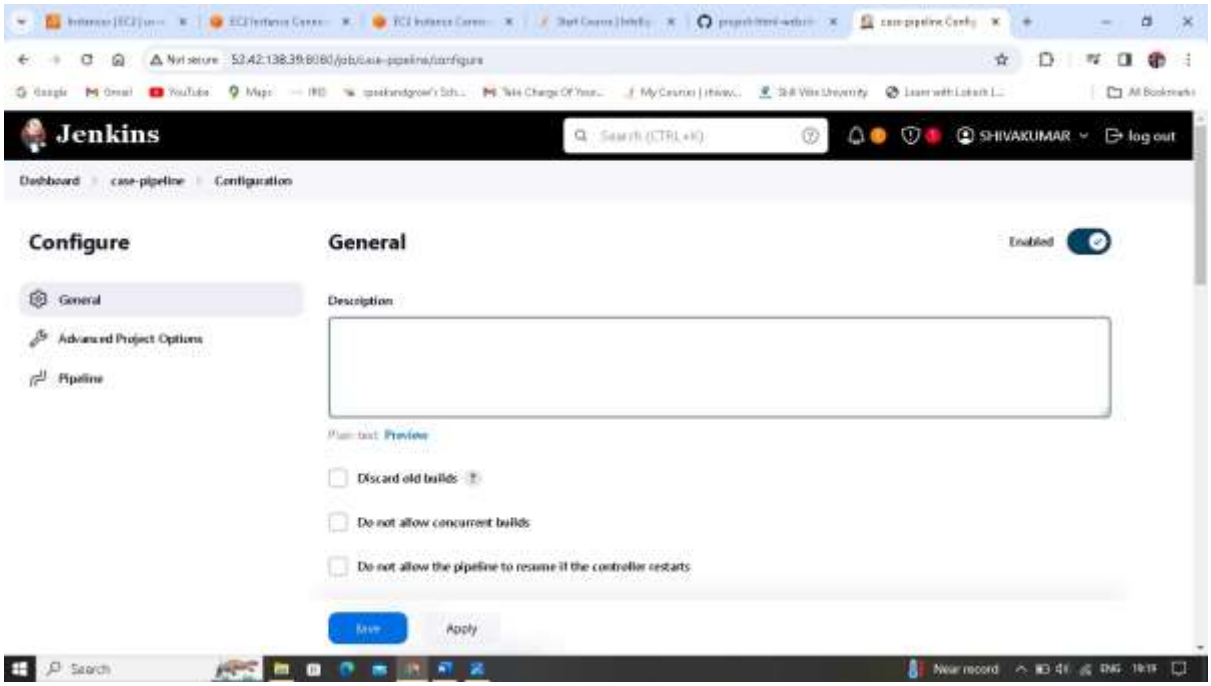
1. Go to Dashboard click on New Item.



2. Enter an item name as “case-pipeline” click on pipeline and ok

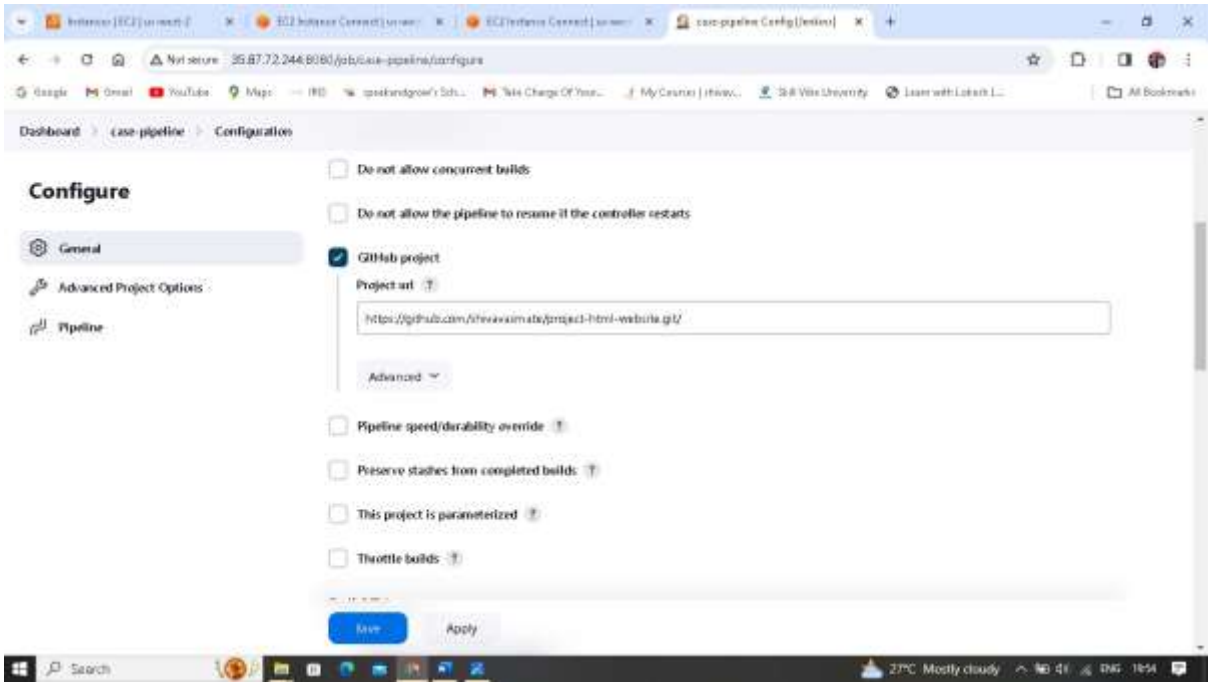


3. Fill it

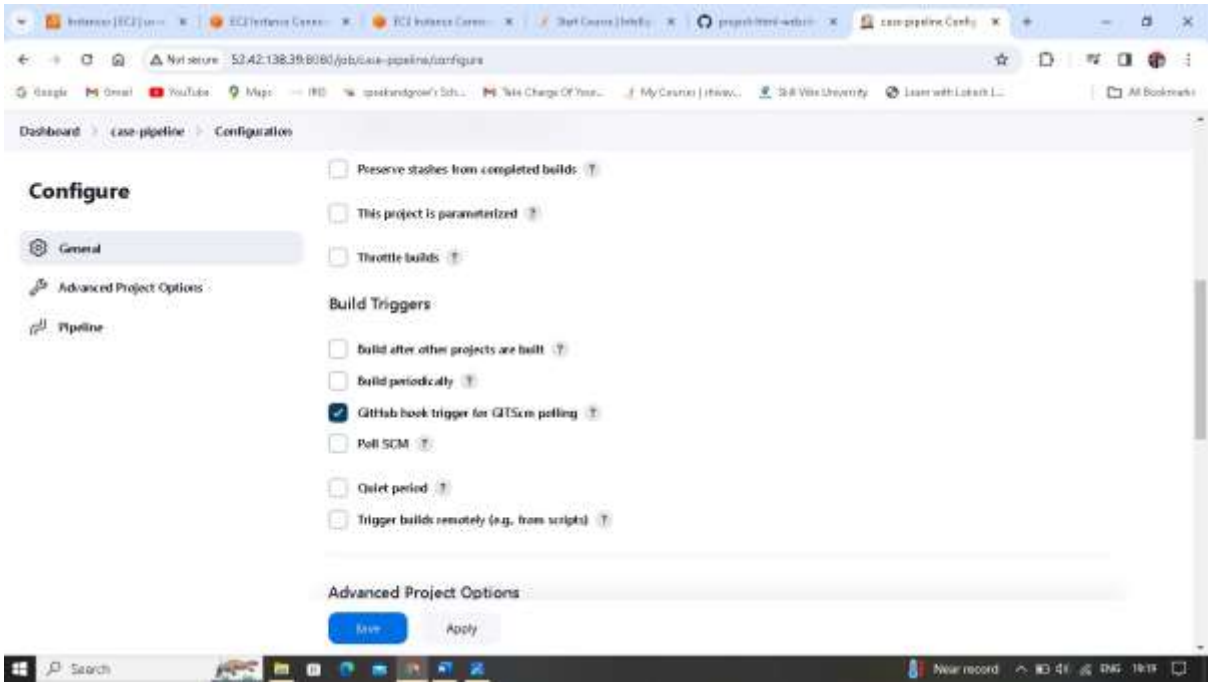


4. Check in GitHub Project

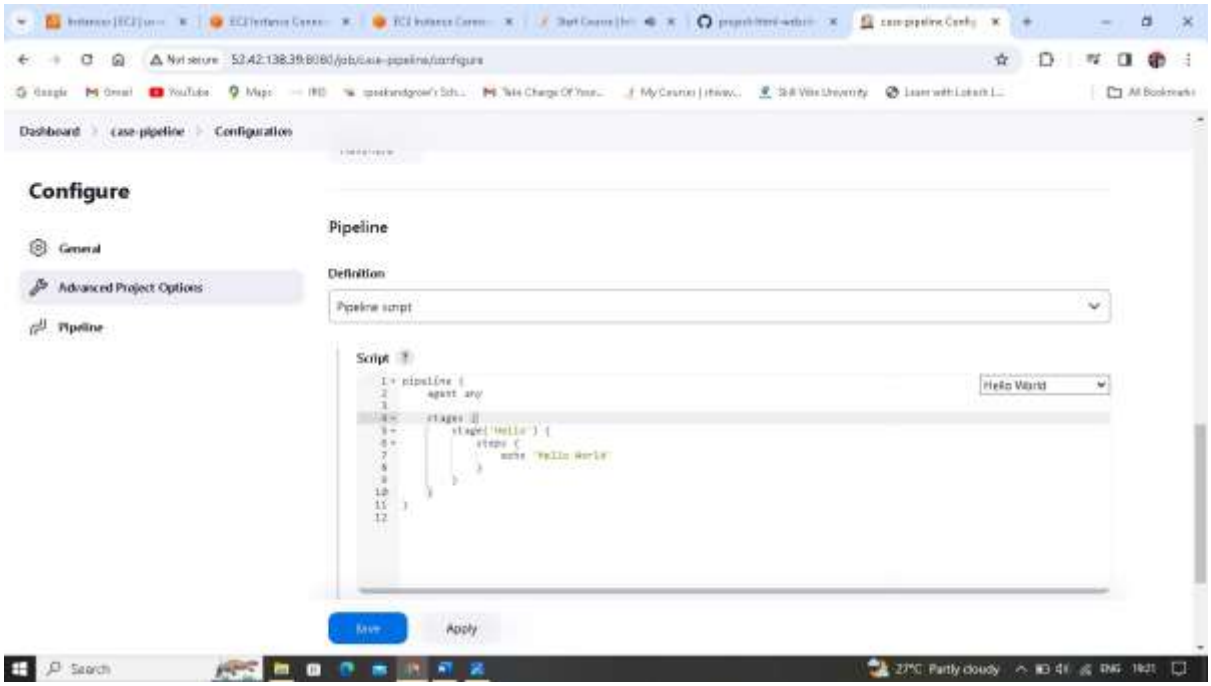
Project url https://github.com/shivavasmate/project-html-website.git/



5. Under Build Triggers check in for GitHub hook trigger for GITScm Polling.



6. Under pipeline script on the right-side corner click on “Hello World”

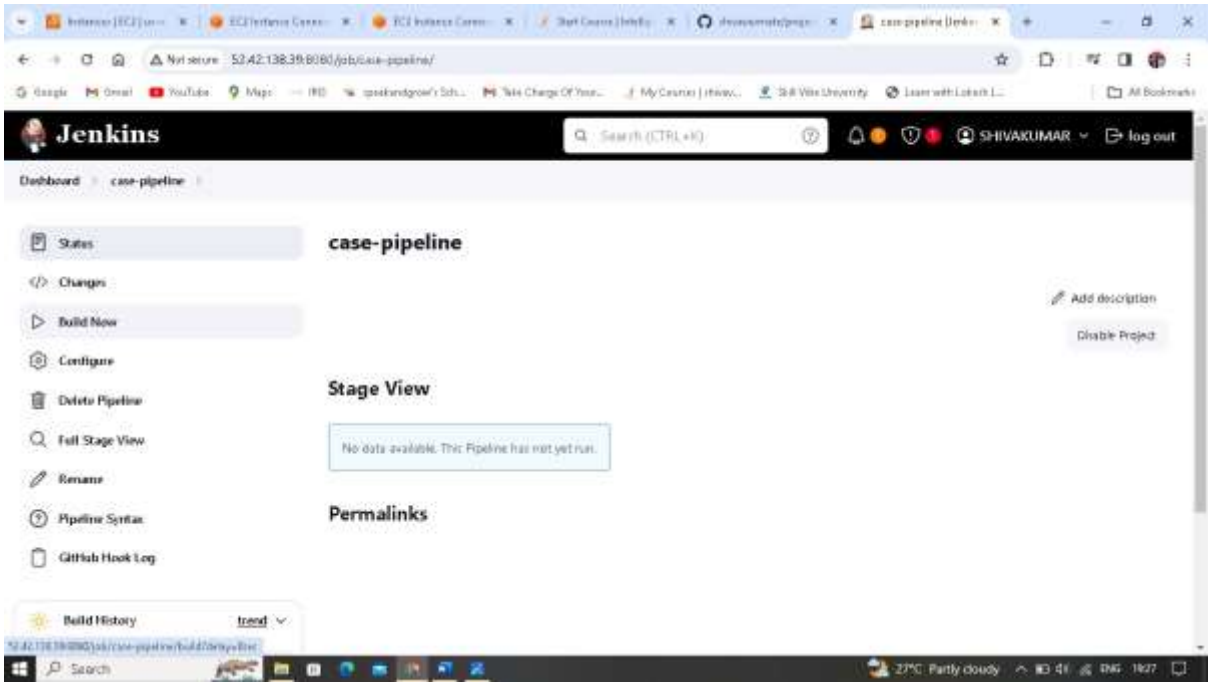


7. Edit the script as

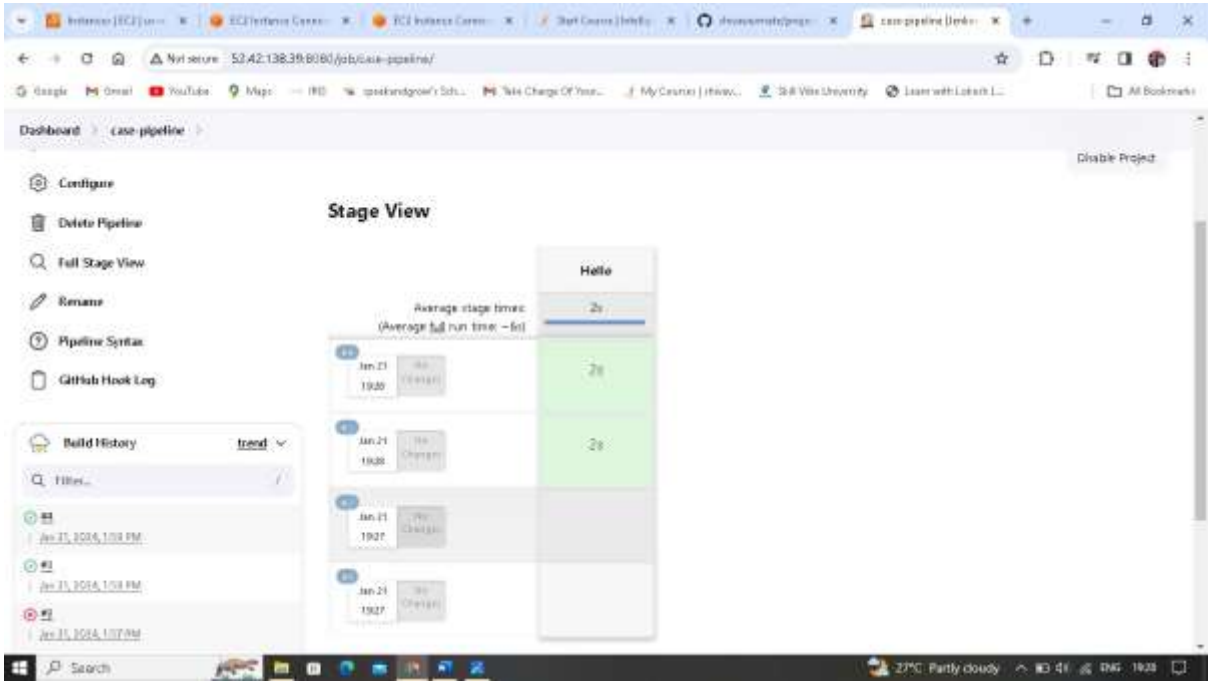
```
    pipeline {
          agent none
    stages {
        stage('Hello') {
           agent {
              label "Node-1"
           }
steps {
                echo 'Hello World'
                git 'https://github.com/shivavasmate/project-html-website.git'
            }
        }
    }
```

and click on save.

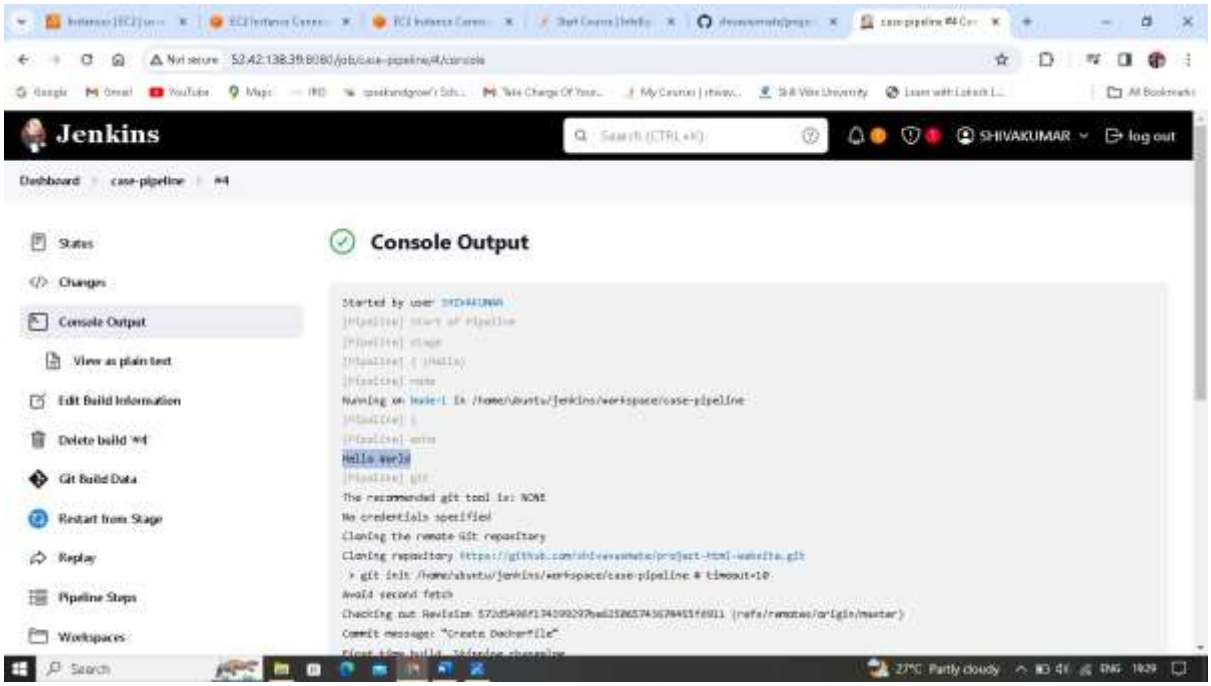
8. The case-pipeline is created and click on Build Now.



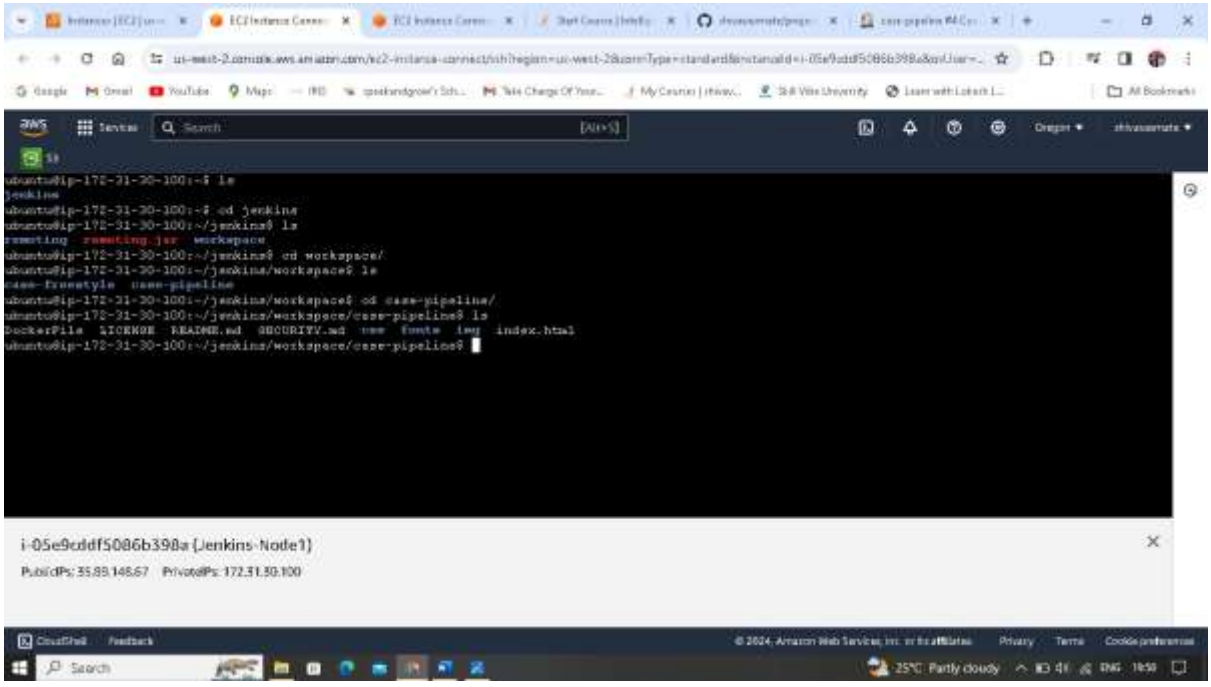
9. Build is successful.



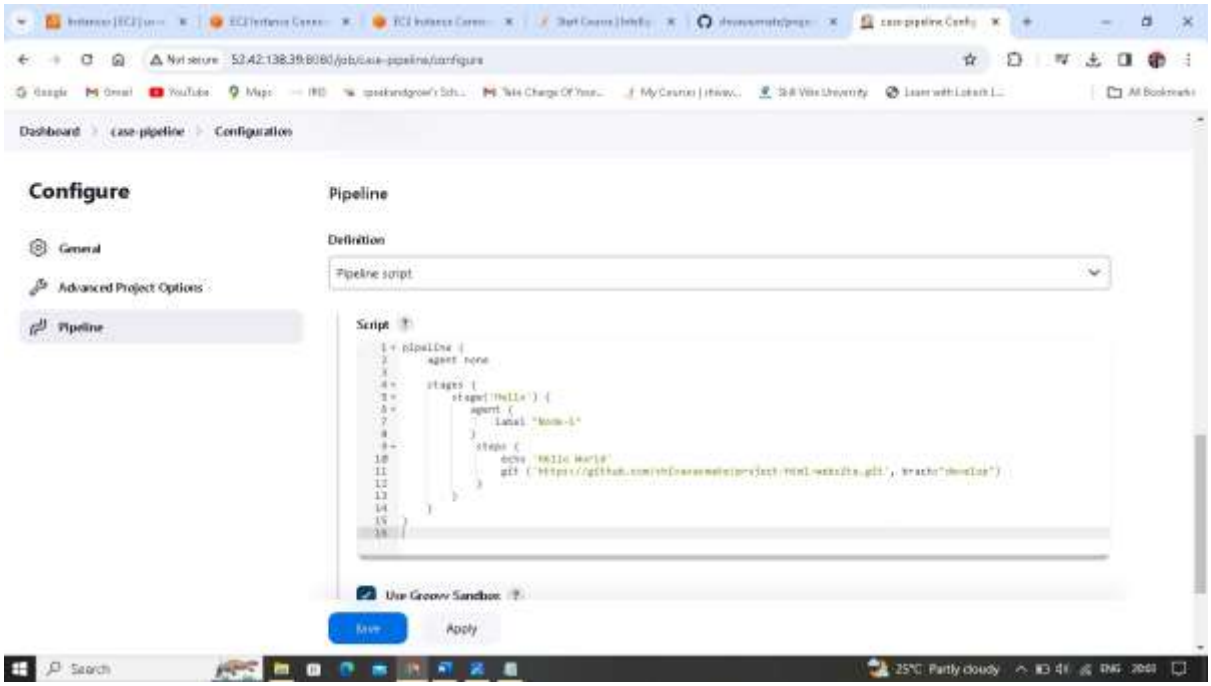
10. Check for it.



11. Go to Jenkins-Node1 check for clone under case-pipeline folder.



12. Now create for develop branch

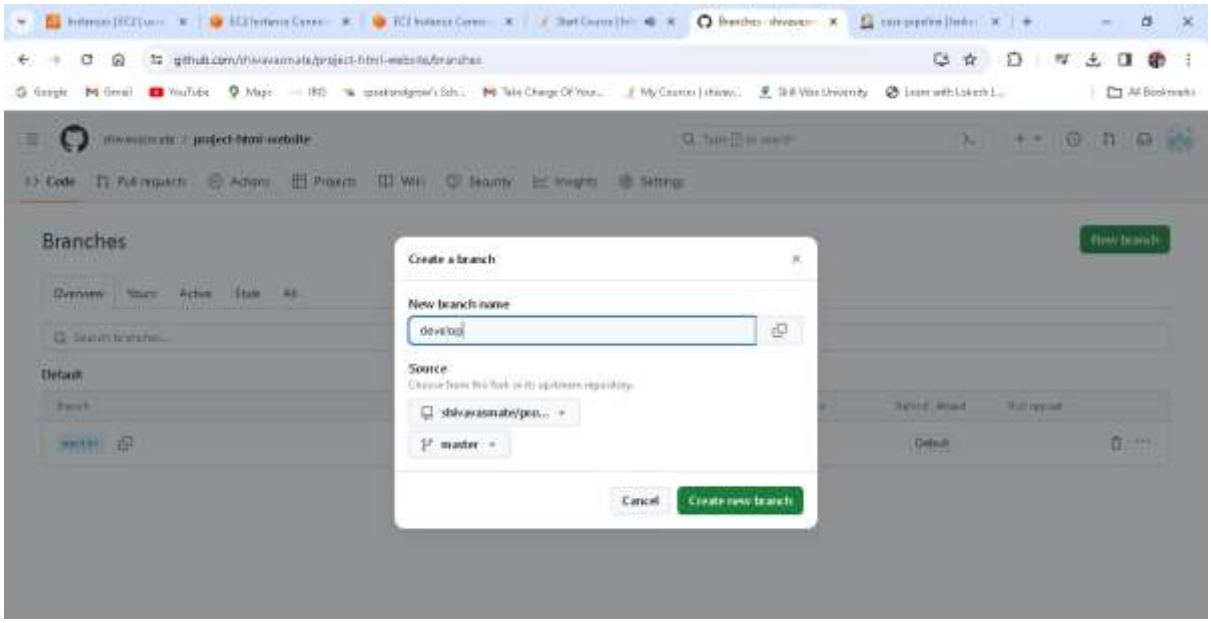


13. Edit the script as

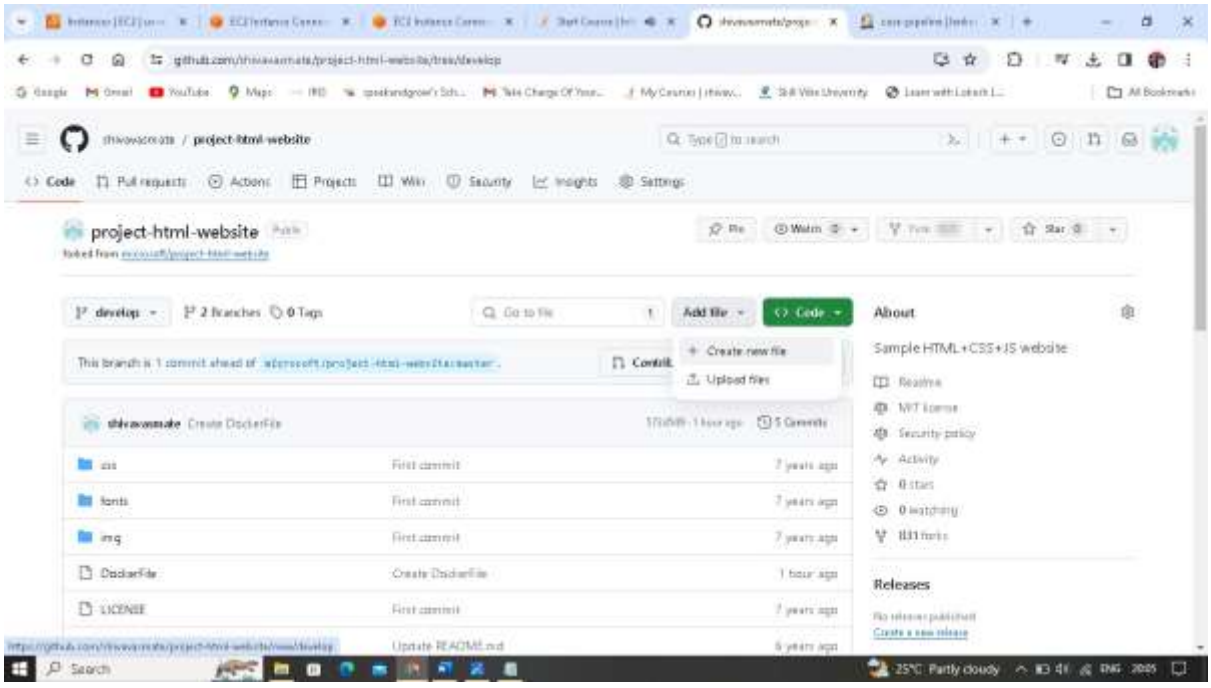
```
pipeline {
    agent none
stages {
    stage('Hello') {
      agent {
        label "Node-1"
      }
steps {
          echo 'Hello World'
        git (url: 'https://github.com/shivavasmate/project-html-website.git' branch:”develop”)
        }
    }
}
```

}

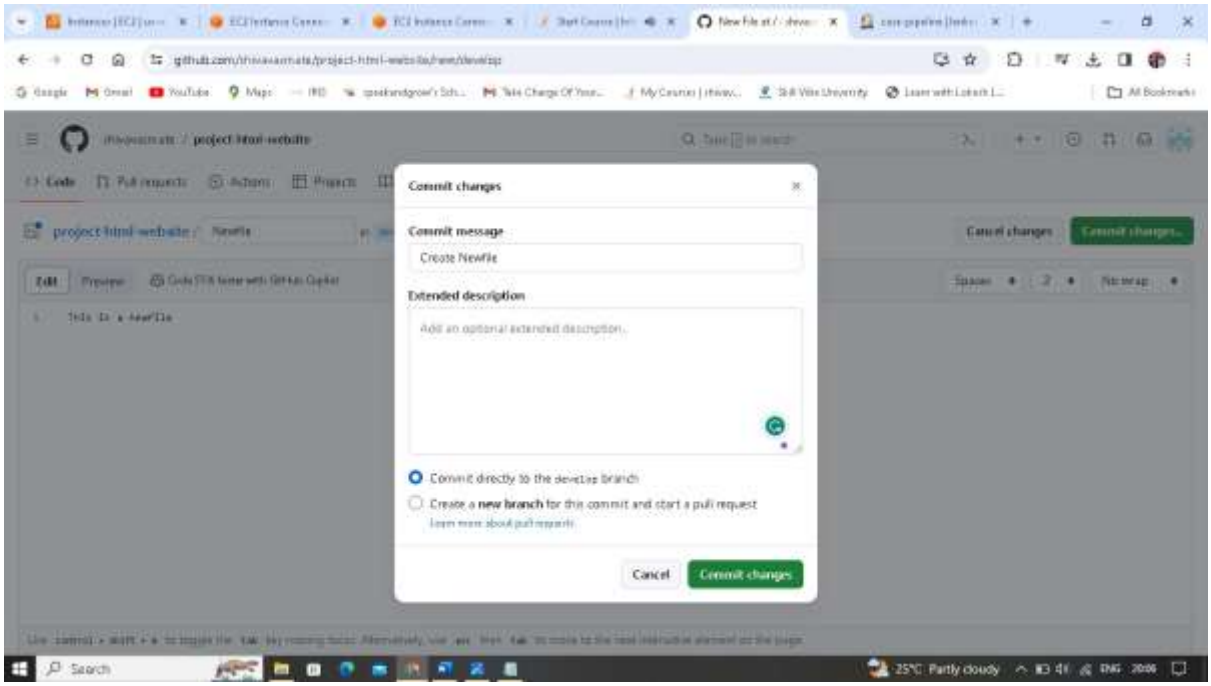
14. Create develop branch in the repo.



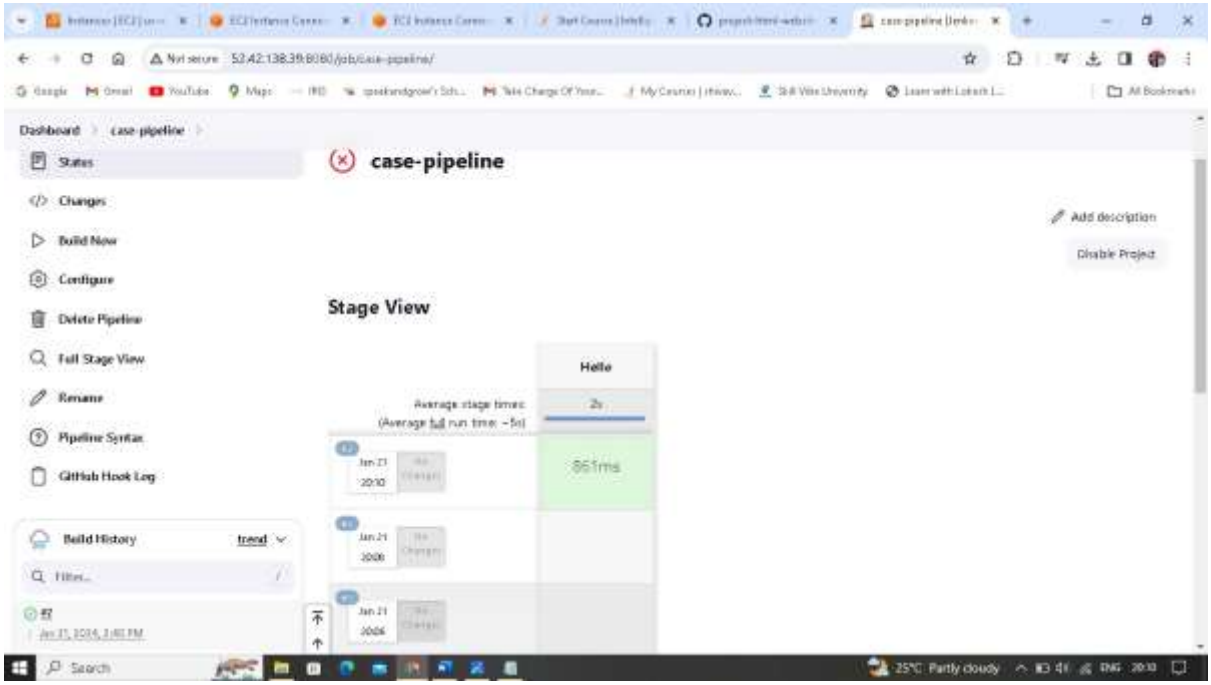
15. Now create new file



16. Commit changes for Nowfile



17. Click on Build Now and it is successful.

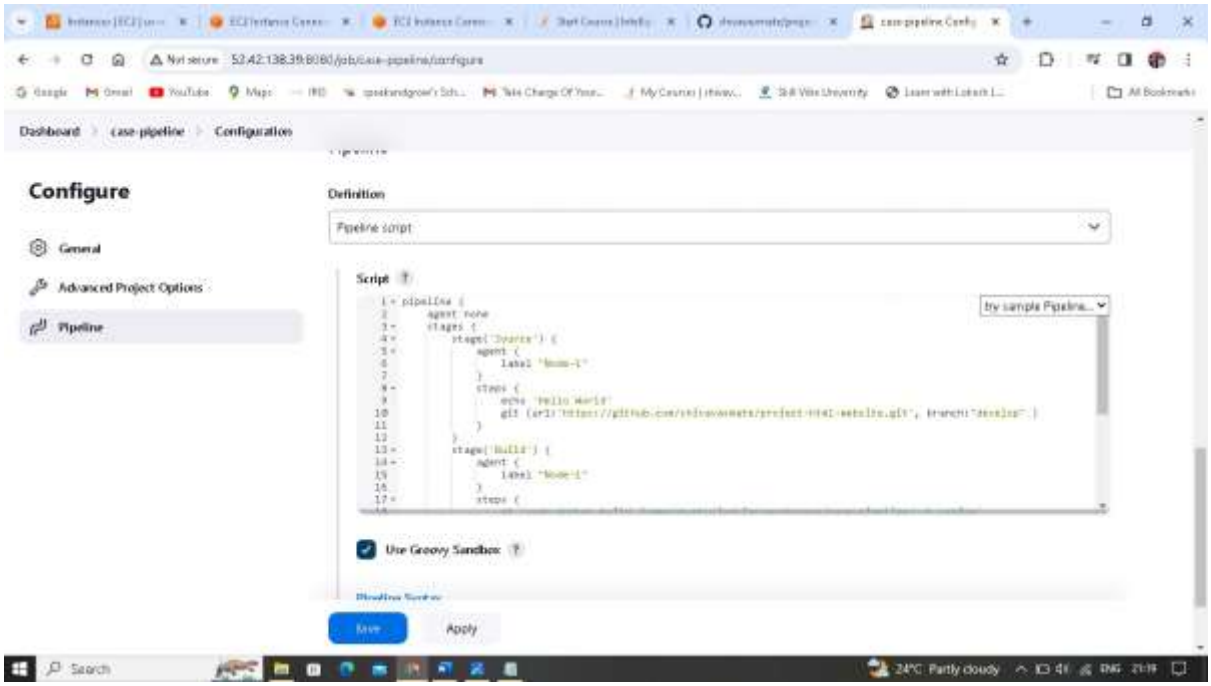


18. For running pipe line script. Edit it as

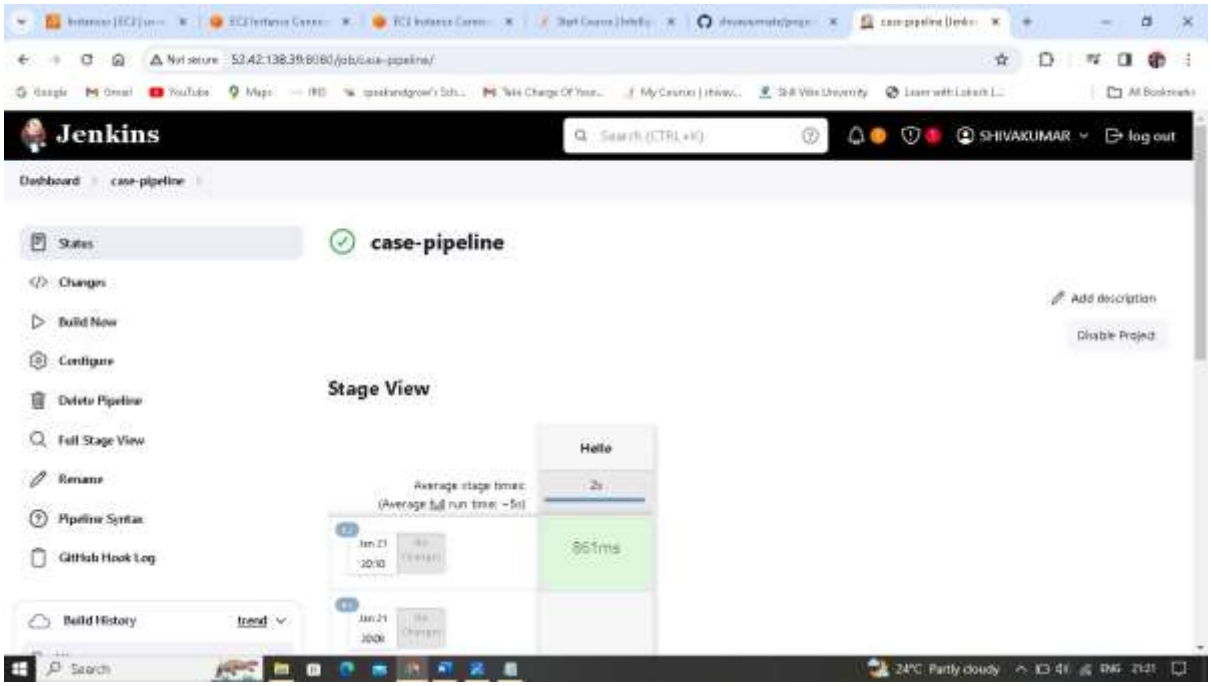
```
pipeline {
    agent none
    stages {
        stage('Source') {
            agent {
                label "Node-1"
            }
            steps {
                echo 'Hello World'
          git (url:'https://github.com/shivavasmate/project-html-website.git', branch:"develop")
            }
        }

        stage('Build') {
            agent {
                label "Node-1"
            }
            steps {
                sh 'sudo docker build /home/ubuntu/jenkins/workspace/case-pipeline/ -t cspipe1'
            }
        }
        stage('Deploy') {
            agent {
                label "Node-1"
```

```
        }
        steps {
            sh 'sudo docker rm -f $(sudo docker ps -aq)'
            sh 'sudo docker run -itd -p 84:80 cspipe1'
        }
    }
  }
}
```

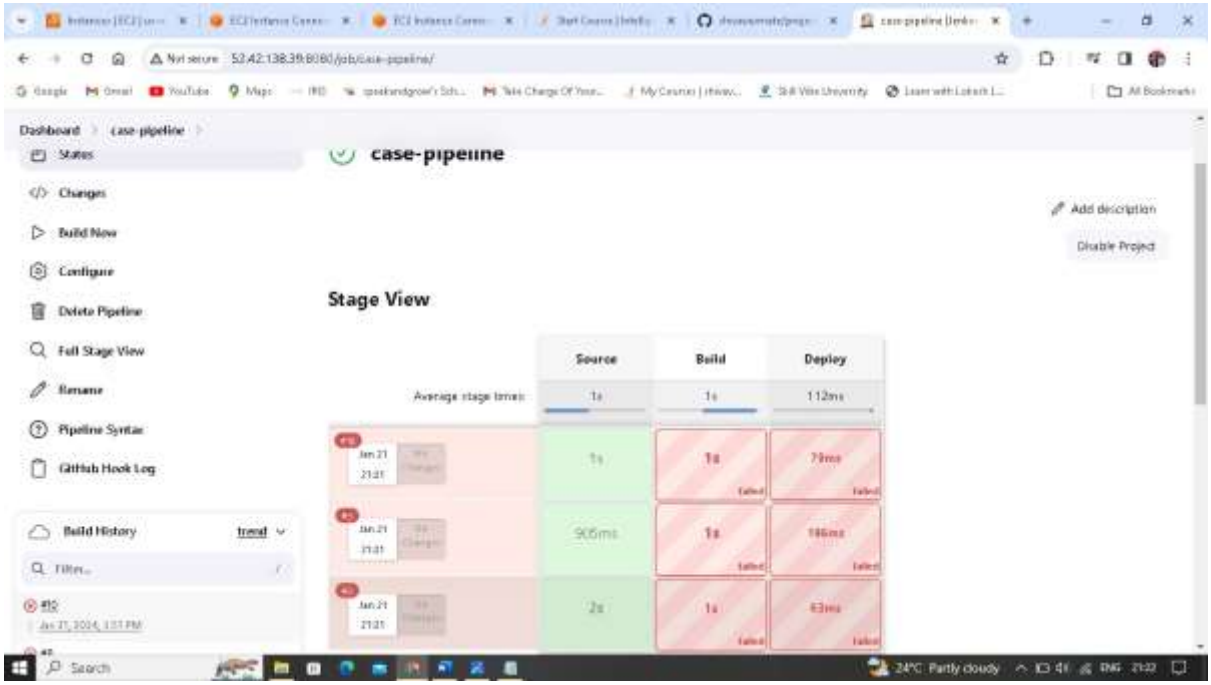
19. Click on save



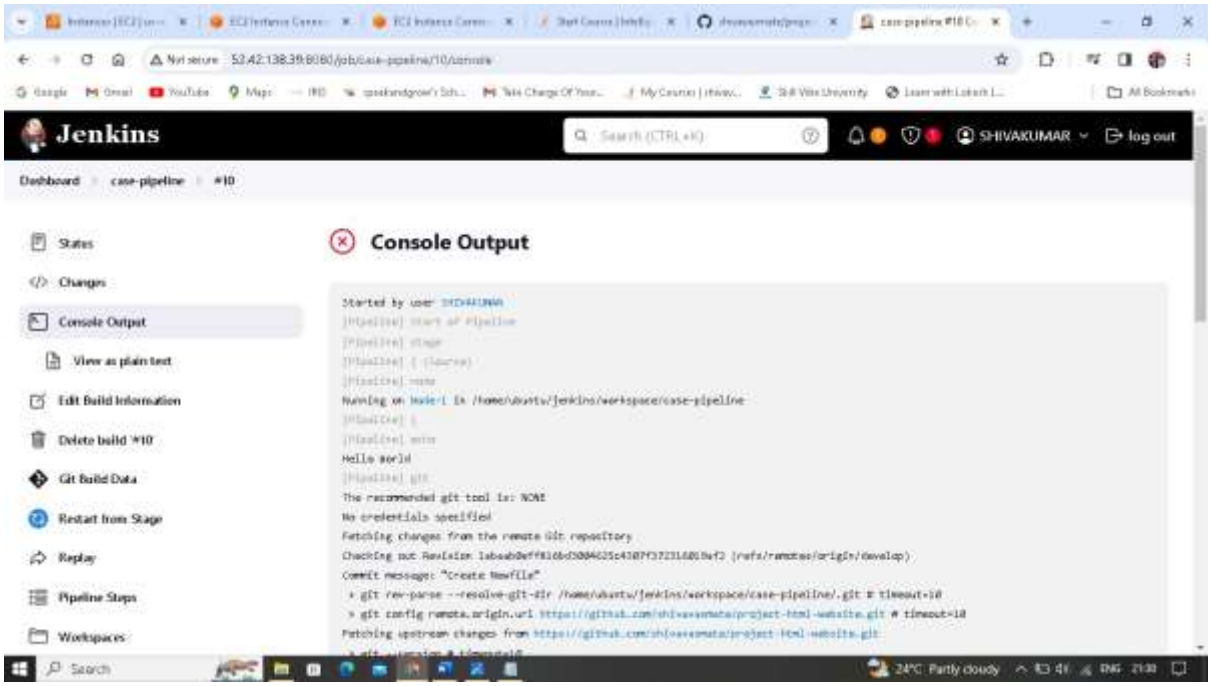
20. Click Build Now

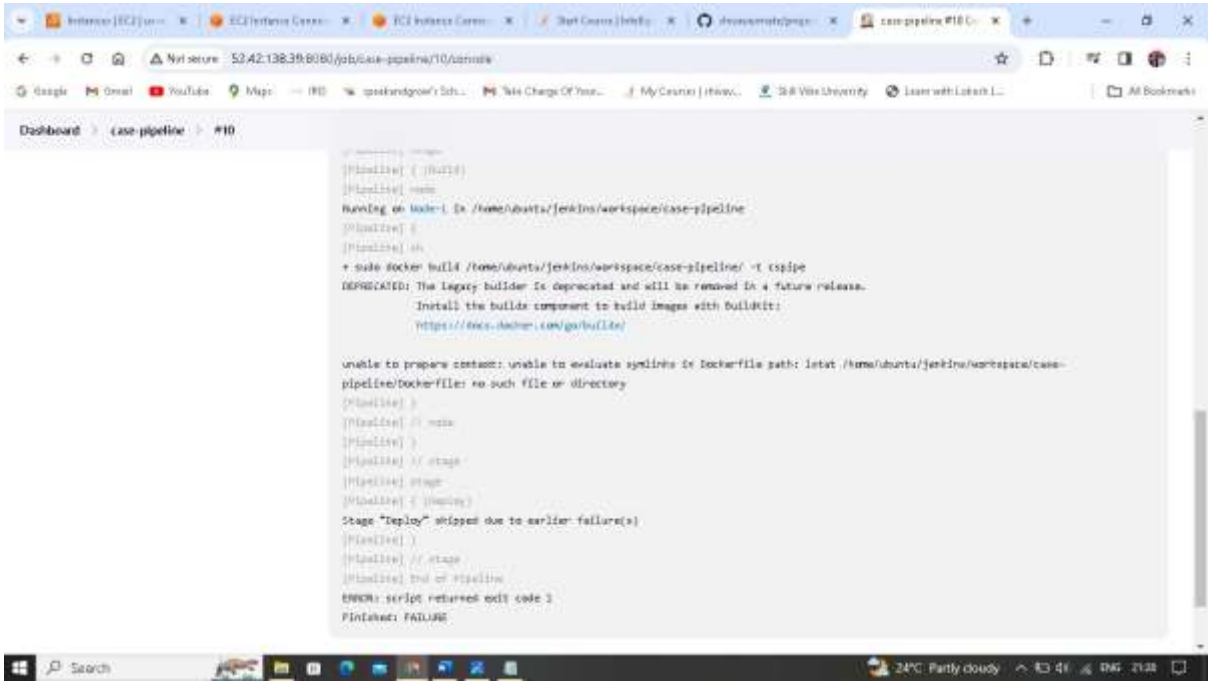


21. Error in Build and Deploy stages



22. Check for error.





unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/ubuntu/jenkins/workspace/case-pipeline/Dockerfile: no such file or directory

[Pipeline] }

[Pipeline] // node

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy)

Stage "Deploy" skipped due to earlier failure(s)

[Pipeline] }

[Pipeline] // stage

[Pipeline] End of Pipeline

ERROR: script returned exit code 1

Finished: FAILURE

RESOLVING ERROR

23. Go to Jenkins-Node1 check for it

ubuntu@ip-172-31-30-100:~/jenkins/workspace$ cd case-pipeline

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-pipeline$ ls

DockerFile LICENSE README.md SECURITY.md css fonts img index.html

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-pipeline$ rm DockerFile Dockerfile

rm: cannot remove 'Dockerfile': No such file or directory

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-pipeline$ mv DockerFile Dockerfile

mv: cannot stat 'DockerFile': No such file or directory

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-pipeline$ sudo docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

<none> <none> 686a9890498b 10 seconds ago 125MB

ubuntu latest e34e831650c1 10 days ago 77.9MB

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-pipeline$ sudo docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

<none> <none> 686a9890498b 17 seconds ago 125MB

ubuntu latest e34e831650c1 10 days ago 77.9MB

ubuntu@ip-172-31-30-100:~/jenkins/workspace/case-pipeline$ sudo docker images

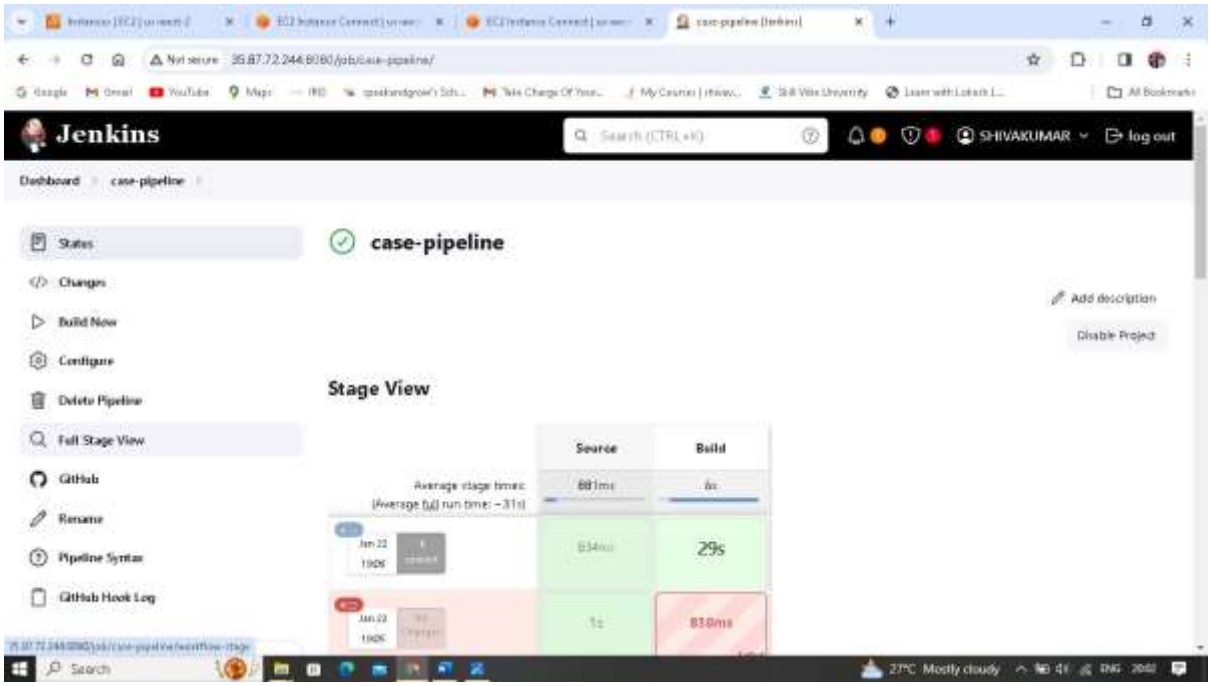
REPOSITORY TAG IMAGE ID CREATED SIZE

csf latest c1e14e16243f 12 seconds ago 234MB

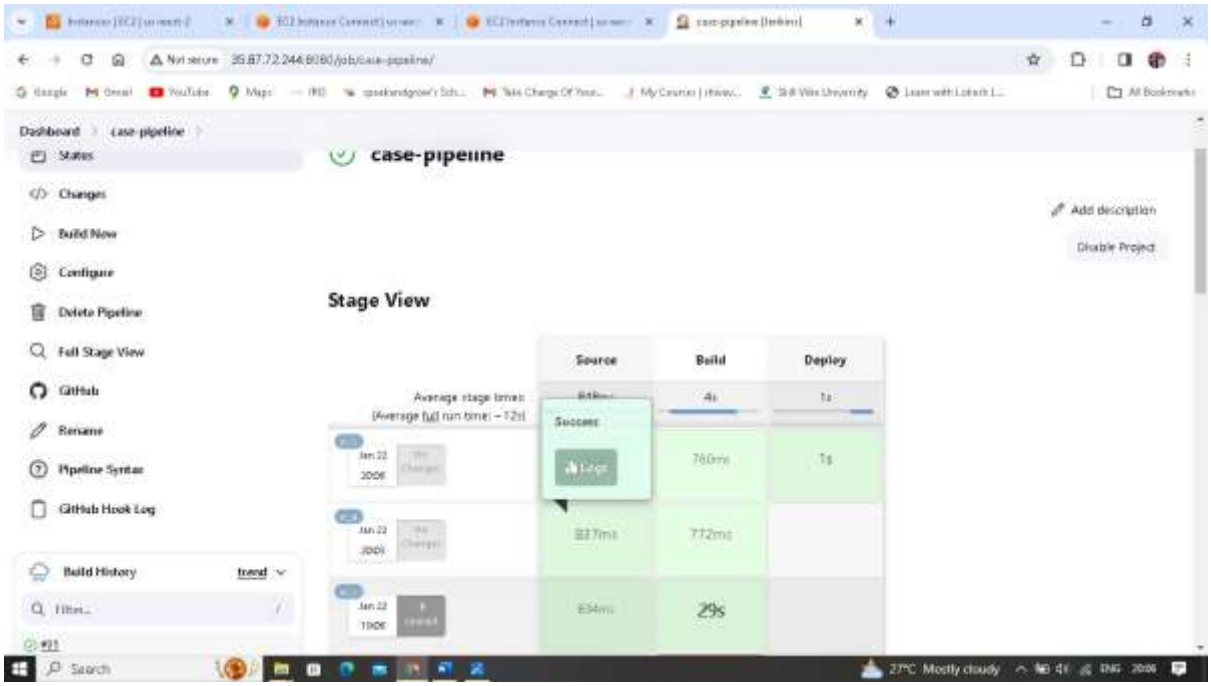
cspipe1 latest 2dc609ca0ae4 14 seconds ago 234MB

ubuntu latest e34e831650c1 10 days ago 77.9MB

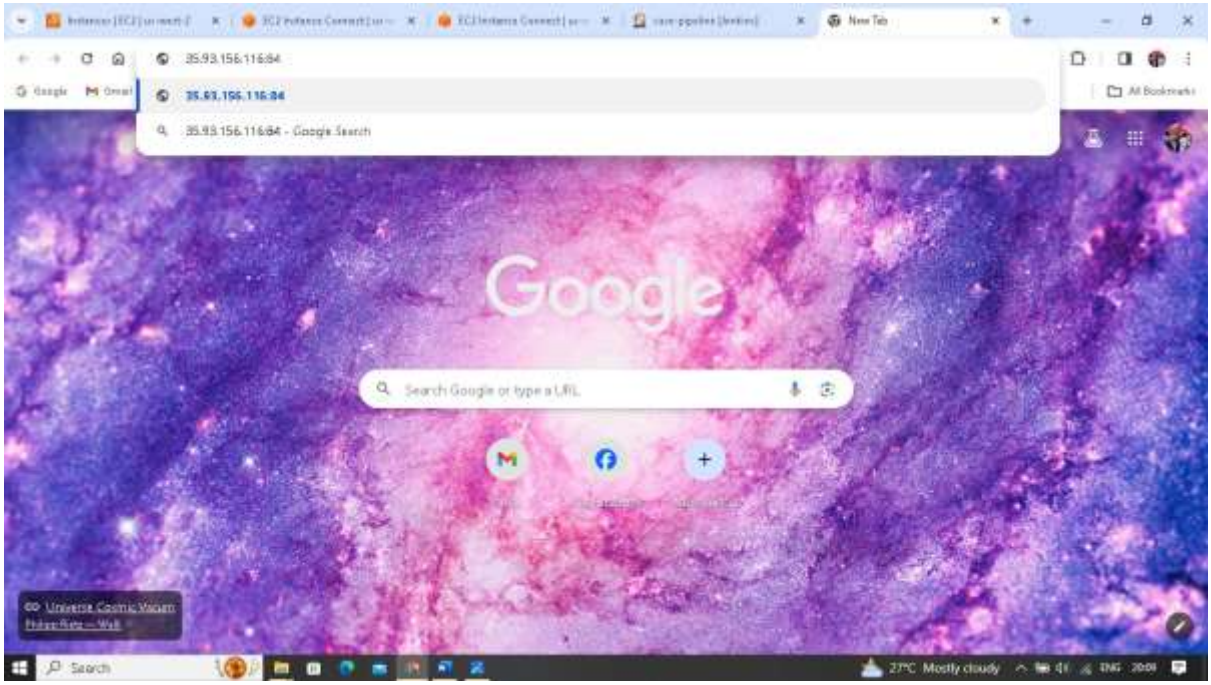
24. Click on Build Now



25. It is successful.



26. Copy and paste the public IP with port :84 in another browser.



27.Now HTML Application is running.