

Project-1

Project Description:

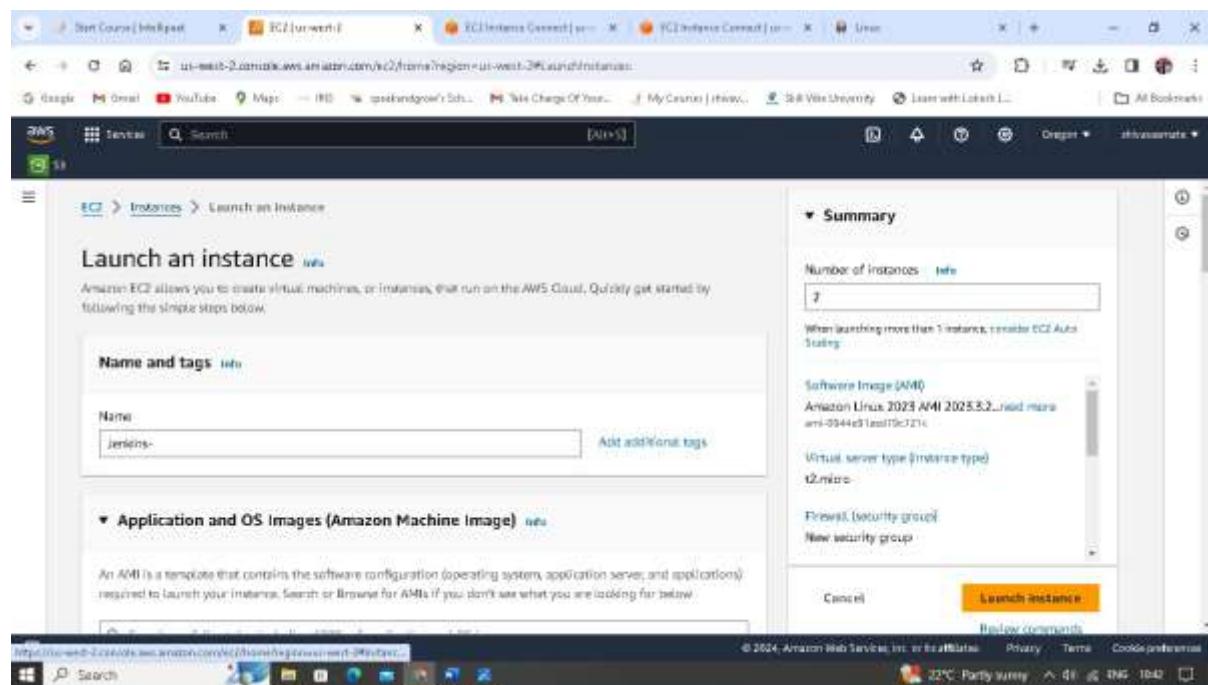
Build CI / CD Pipeline using Jenkins and deploy the real world Web Application in AWS Cloud.

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]

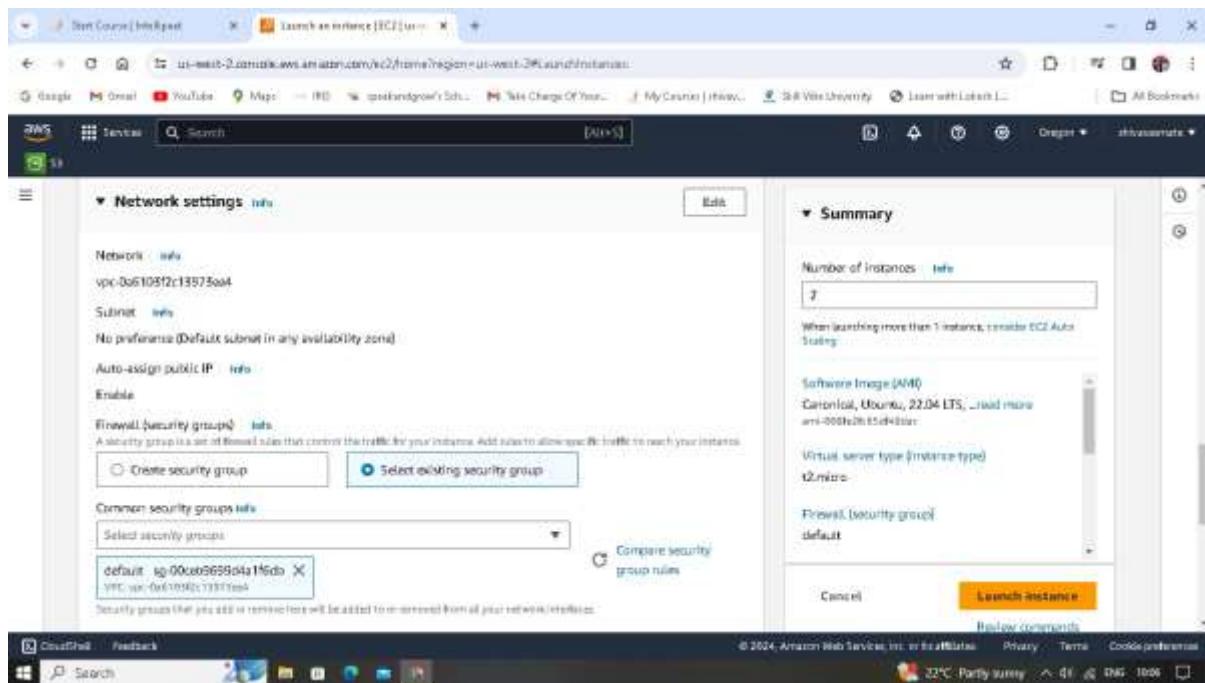
The screenshot shows the AWS Lambda console with the URL <https://ui-west-2.console.aws.amazon.com/lambda/v2/functions?region=us-west-2&launch/instances>. The page displays a search bar and a list of AMI icons: Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. A summary section on the right indicates 1 instance, a software image of Canonical, Ubuntu, 20.04 LTS, and an instance type of t2.micro. The 'Launch Instance' button is highlighted.

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

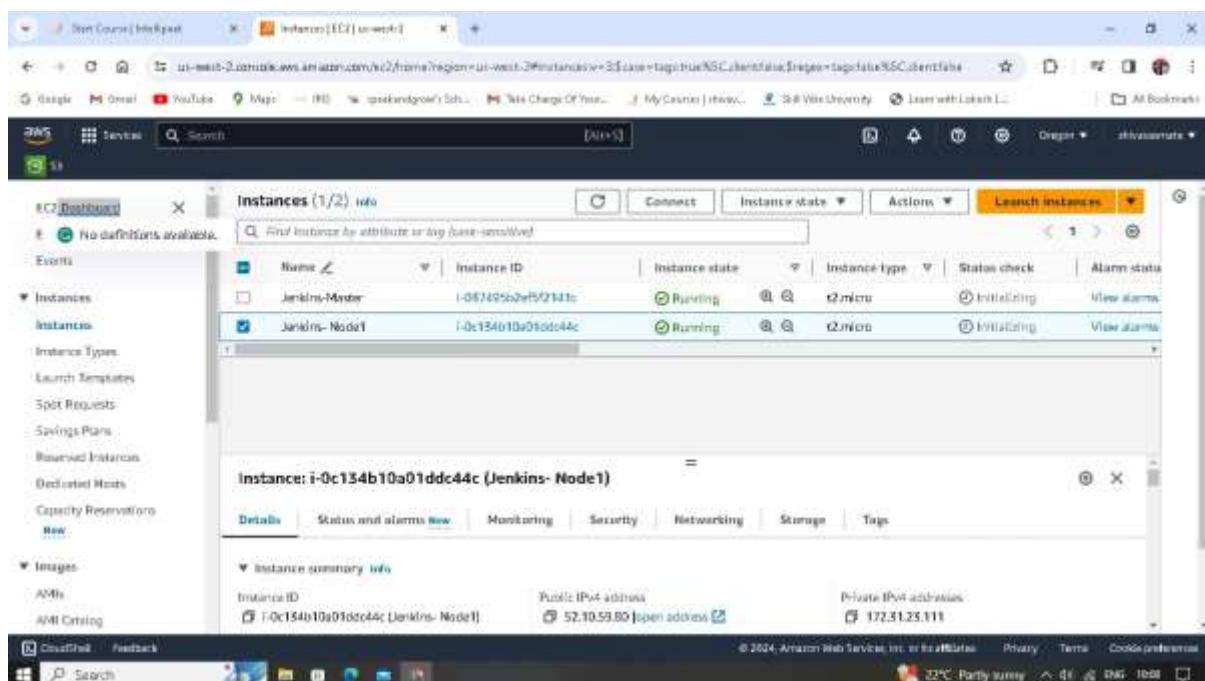
d. Key-pair: Create a key pair [rsa and .pem] with a name ubuntu-key

The screenshot shows the AWS Lambda console with the URL <https://ui-west-2.console.aws.amazon.com/lambda/v2/functions?region=us-west-2&launch/instances>. It displays the 'Instance type' section for t2.micro and the 'Key pair (login)' section where 'ubuntu-key' is selected. The summary on the right remains the same as in the previous step.

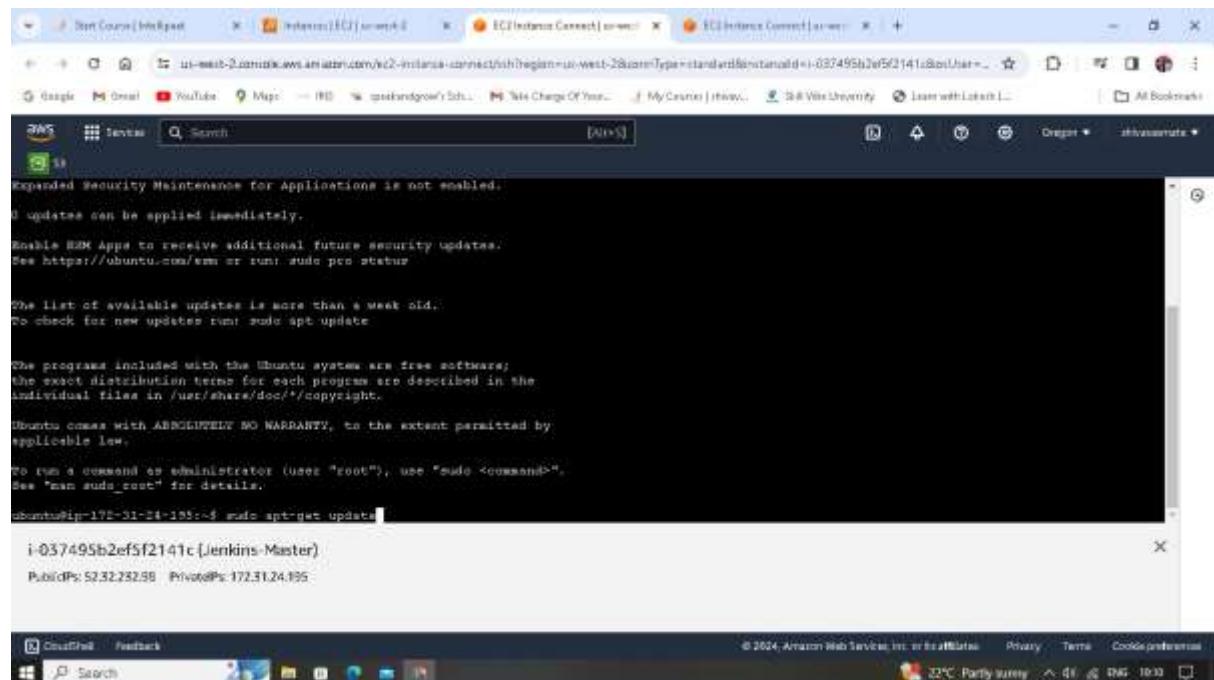
e. VPC and Security group as default with all traffic protocols allowed click on launch instance.



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

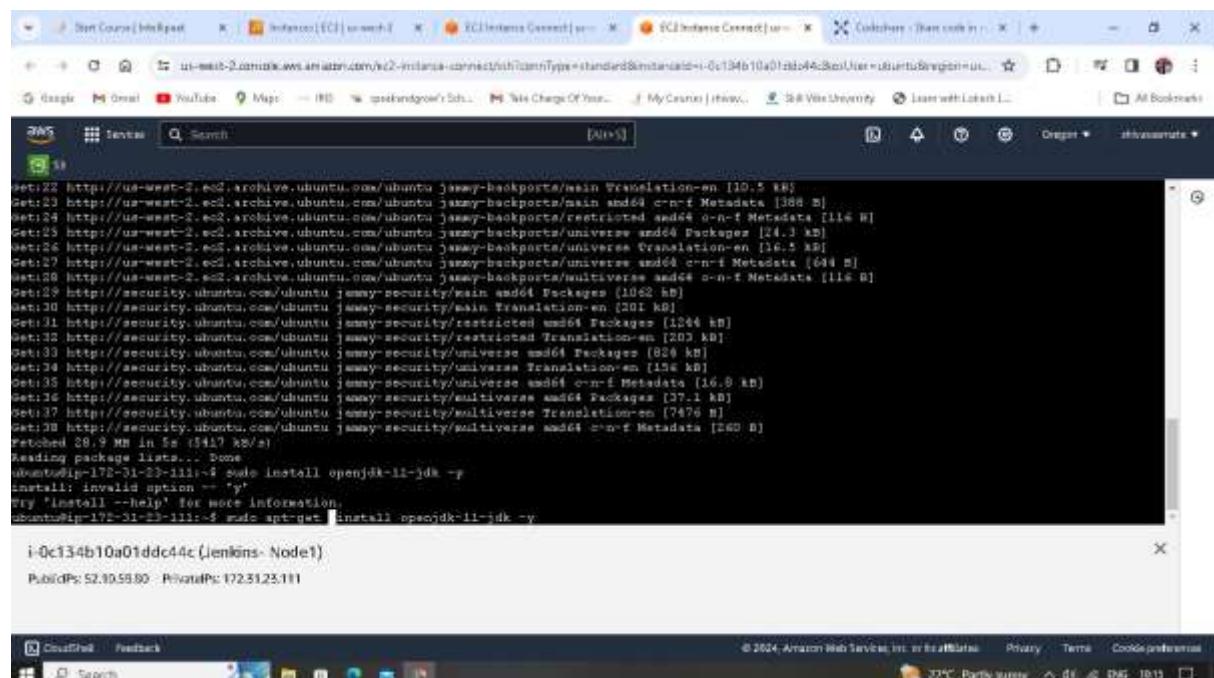


6. Connect Jenkins- Node1 and update it using command sudo apt-get update.



```
Expended Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
To run a command as administrator (use "root"), use "sudo <command>"  
See "man sudo_root" for details.  
ubuntu@ip-172-31-24-195:~$ sudo apt-get update  
i-037495b2ef5f2141c [Jenkins-Master]  
PublicIP: 52.32.232.99 PrivateIP: 172.31.24.195
```

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



```
get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/main Translation-en [10.5 kB]  
get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/main amd64 crn-f Metadata [388 B]  
get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/restricted amd64 crn-f Metadata [114 B]  
get:25 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/universe amd64 Packages [24.3 kB]  
get:26 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/universe Translation-en [16.5 kB]  
get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/universe amd64 crn-f Metadata [644 B]  
get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/multiverse amd64 crn-f Metadata [118 B]  
get:29 http://security.ubuntu.com/ubuntu jessy-security/main amd64 Packages [1062 kB]  
get:30 http://security.ubuntu.com/ubuntu jessy-security/main Translation-en [201 kB]  
get:31 http://security.ubuntu.com/ubuntu jessy-security/universe amd64 Packages [1244 kB]  
get:32 http://security.ubuntu.com/ubuntu jessy-security/restricted Translation-en [203 kB]  
get:33 http://security.ubuntu.com/ubuntu jessy-security/universe amd64 Packages [826 kB]  
get:34 http://security.ubuntu.com/ubuntu jessy-security/universe Translation-en [158 kB]  
get:35 http://security.ubuntu.com/ubuntu jessy-security/universe amd64 crn-f Metadata [16.8 kB]  
get:36 http://security.ubuntu.com/ubuntu jessy-security/multiverse amd64 Packages [37.1 kB]  
get:37 http://security.ubuntu.com/ubuntu jessy-security/multiverse Translation-en [7476 B]  
get:38 http://security.ubuntu.com/ubuntu jessy-security/multiverse amd64 crn-f Metadata [160 B]  
Fetched 28.9 MB in 5s (5417 kB/s)  
Reading package lists... done  
ubuntu@ip-172-31-23-111:~$ sudo apt-get install openjdk-11-jdk -y  
install: invalid option -- 'y'  
Try 'apt-get --help' for more information.  
ubuntu@ip-172-31-23-111:~$ sudo apt-get install openjdk-11-jdk -y  
i-0c134b10a01dddc44c [Jenkins- Node1]  
PublicIP: 52.39.59.80 · PrivateIP: 172.31.23.111
```

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

```
root@ip-172-31-24-195:~# sudo apt-get update
Get:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie InRelease [119 kB]
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-updates InRelease [119 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports InRelease [119 kB]
Get:4 https://pkg.jenkins.io/debian binary/ InRelease
Get:5 https://pkg.jenkins.io/debian binary/ Release
Get:6 http://security.ubuntu.com/ubuntu jessy-security InRelease
Fetched 119 kB in 1s (117 kB/s)
Reading package lists... Done
root@ip-172-31-24-195:~# apt list --upgradable
root@ip-172-31-24-195:~# sudo apt-get install openjdk-11-jdk
root@ip-172-31-24-195:~#
```

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
```

The screenshot shows a terminal window with the following command history:

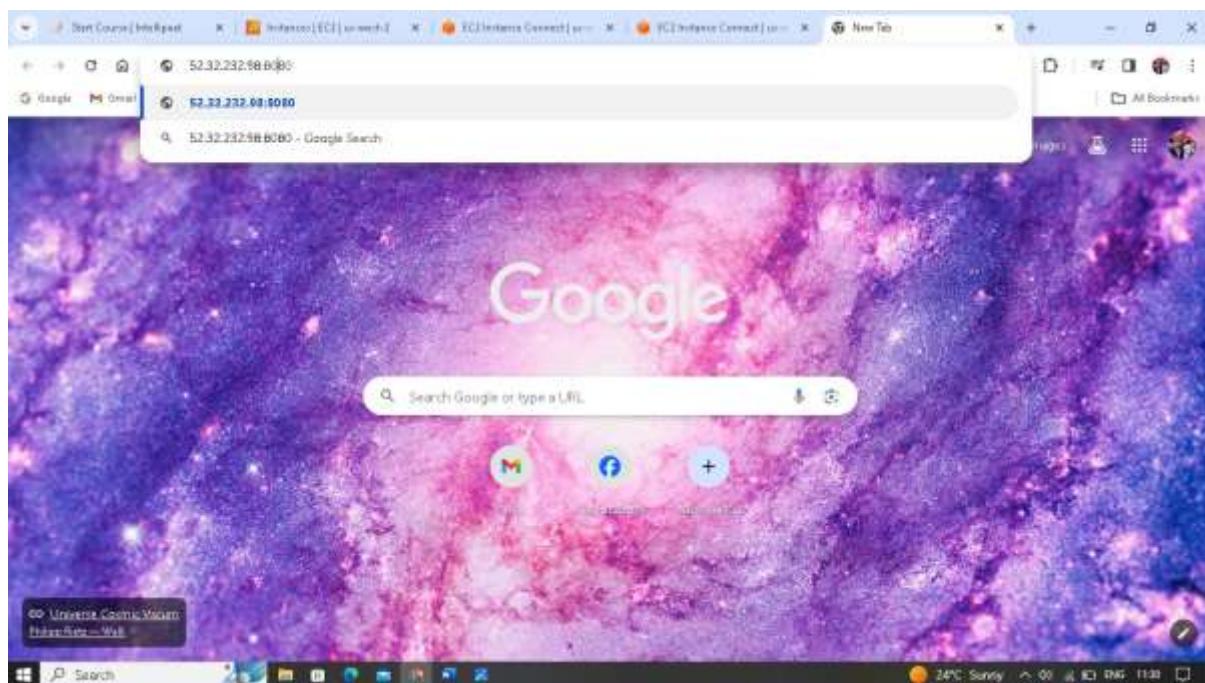
```
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javaws to provide /usr/bin/javaws (javaws) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jhsmb to provide /usr/bin/jhsmb (jhsmb) in auto mode
Setting up openjdk-11-jdk:amd64 (11.0.21+9-Ubuntu-22.04) ...
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (firmware) binaries on this host.
ubuntu@ip-172-31-24-195:~$ 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

i-037495b2ef5f2141c [Jenkins-Master]
BuildID: 5232232.98 PrivateIP: 172.31.24.195
```



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

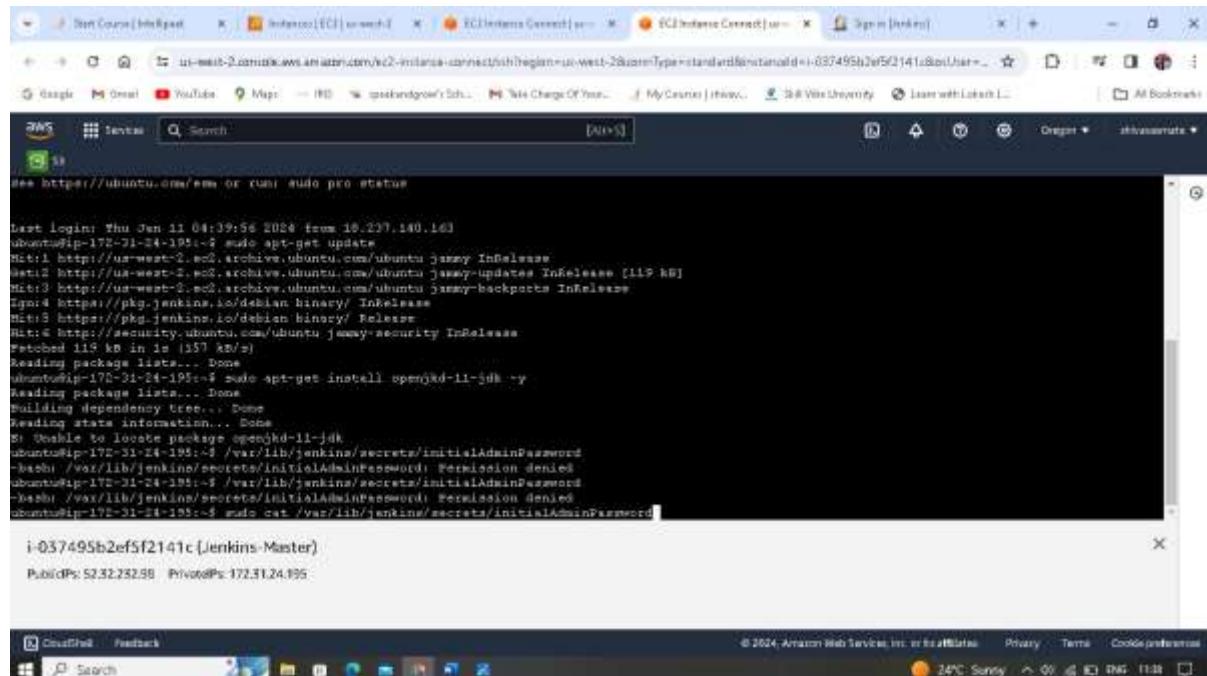


11. Jenkins sign-in page opens.



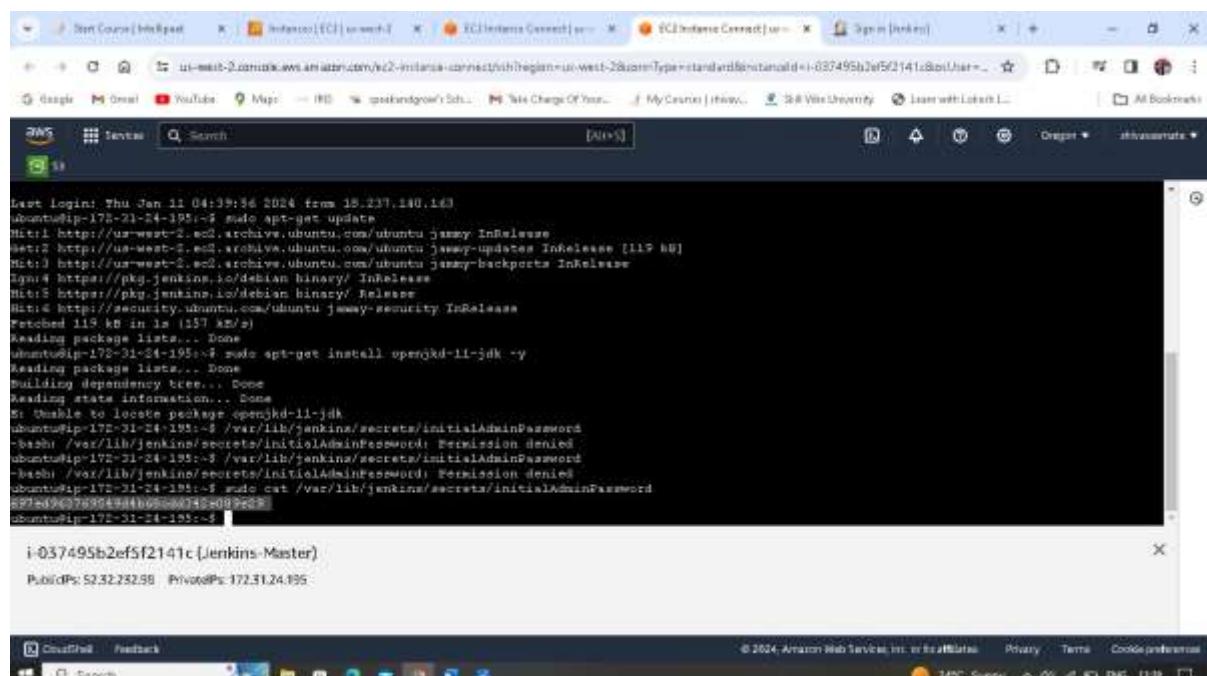
12 copy `/var/lib/jenkins/secrets/initialAdminPassword`

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



```
Last login: Thu Jan 11 04:39:56 2024 from 10.237.140.163
ubuntu@ip-172-31-24-195:~$ sudo apt-get update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-updates InRelease [119 kB]
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports InRelease
Ign:4 https://pkg.jenkins.io/debian binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian binary/ Release
Hit:6 https://security.ubuntu.com/ubuntu jessie-security InRelease
Fetched 119 kB in 1s (157 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-24-195:~$ sudo apt-get install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Building dependency tree... Done
The package openjdk-11-jdk is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
i-037495b2ef5f2141c (Jenkins-Master)
BuildID: S23223298 PrivateIP: 172.31.24.195
```

13. Hit enter to get password copy it.

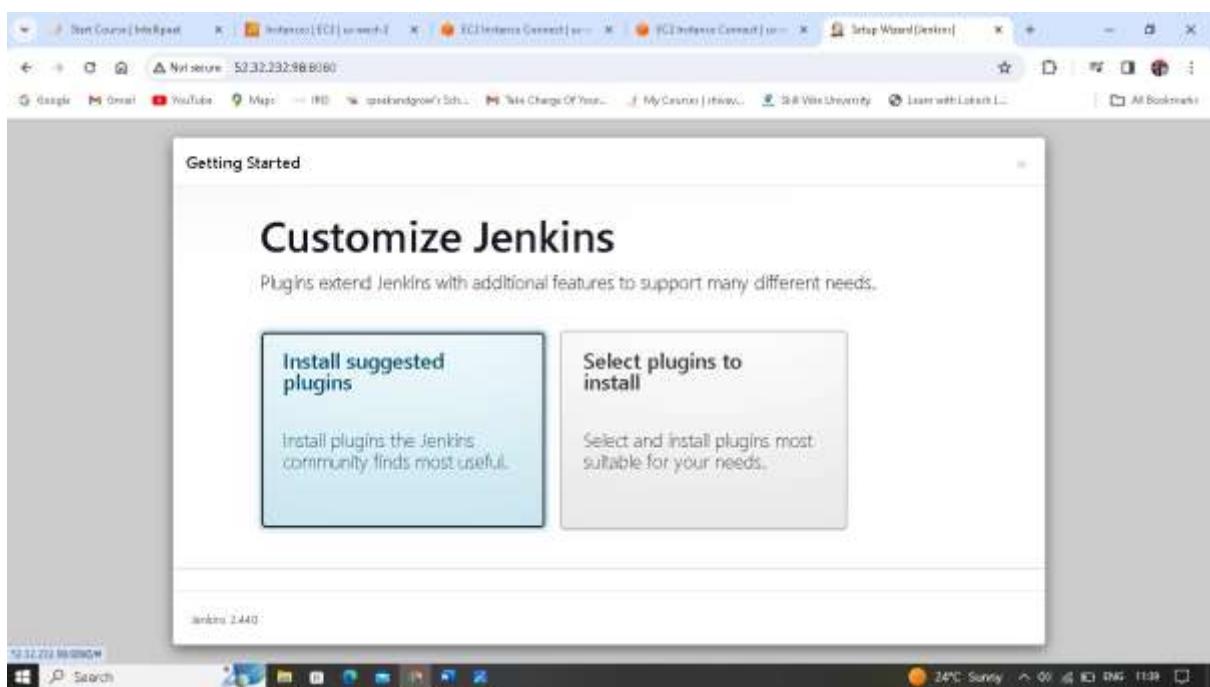


```
Last login: Thu Jan 11 04:39:56 2024 from 10.237.140.163
ubuntu@ip-172-31-24-195:~$ sudo apt-get update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-updates InRelease [119 kB]
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports InRelease
Ign:4 https://pkg.jenkins.io/debian binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian binary/ Release
Hit:6 https://security.ubuntu.com/ubuntu jessie-security InRelease
Fetched 119 kB in 1s (157 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-24-195:~$ sudo apt-get install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Building dependency tree... Done
The package openjdk-11-jdk is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
i-037495b2ef5f2141c (Jenkins-Master)
BuildID: S23223298 PrivateIP: 172.31.24.195
```

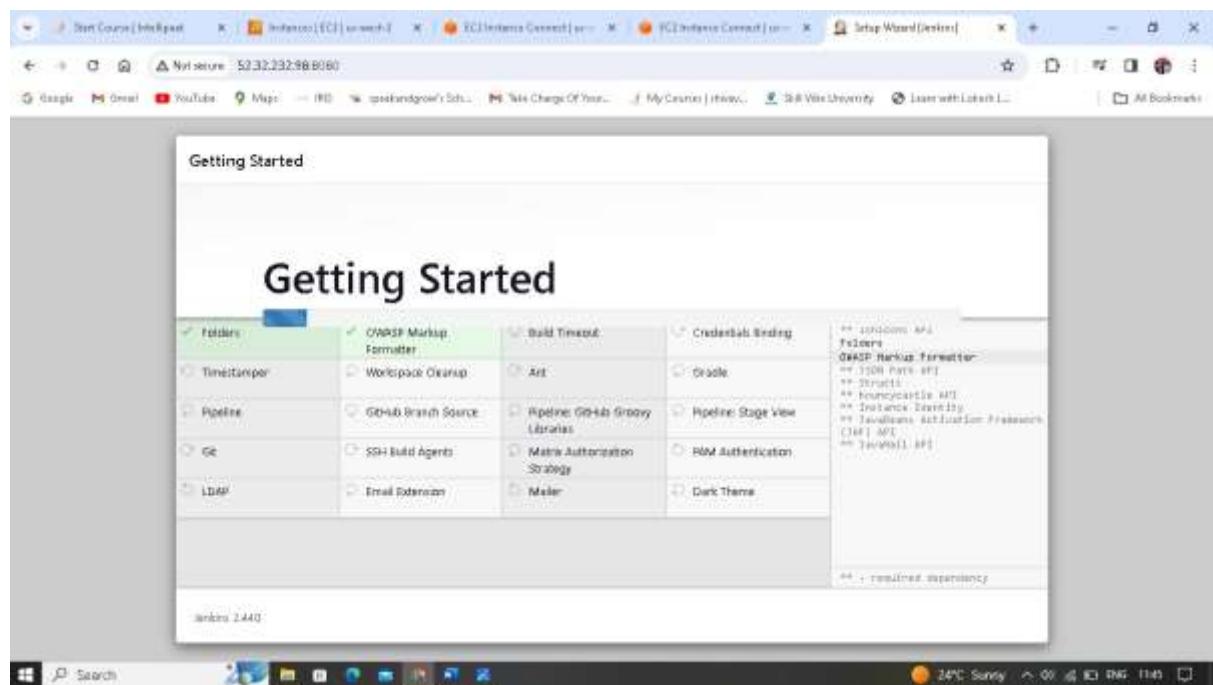
14. Paste the password and click on continue



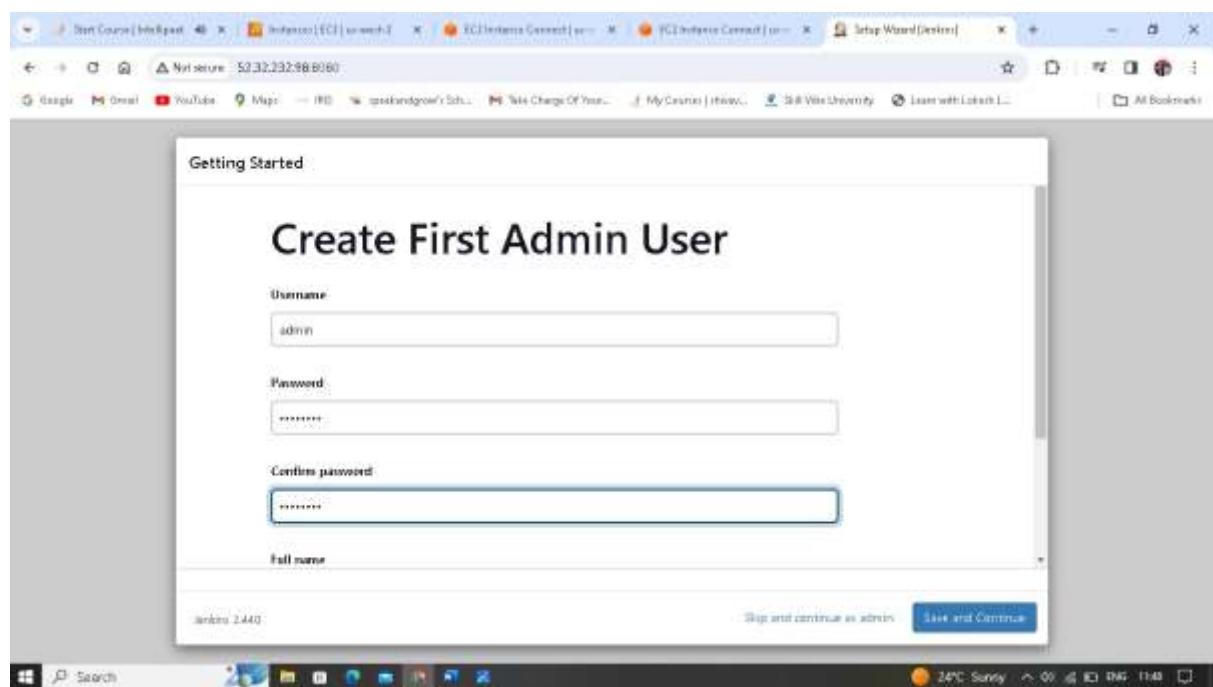
15. Click on install suggested plugins



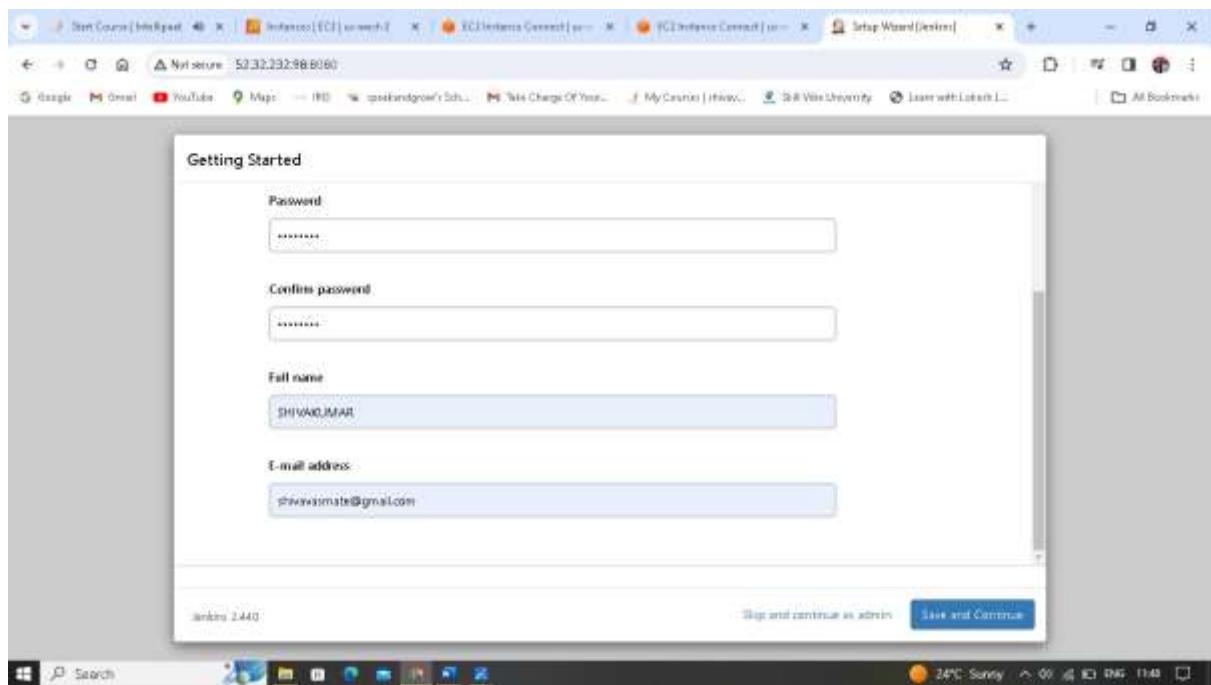
16. Just wait for all the plugins to install



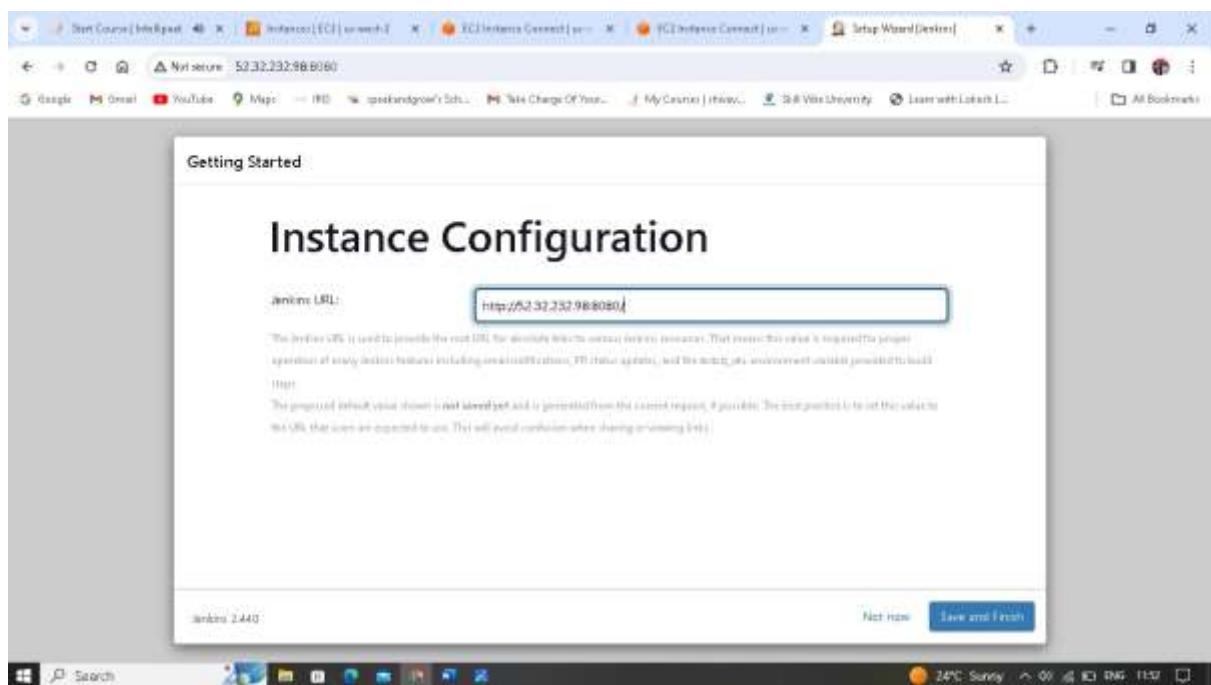
17. Now create first admin user.

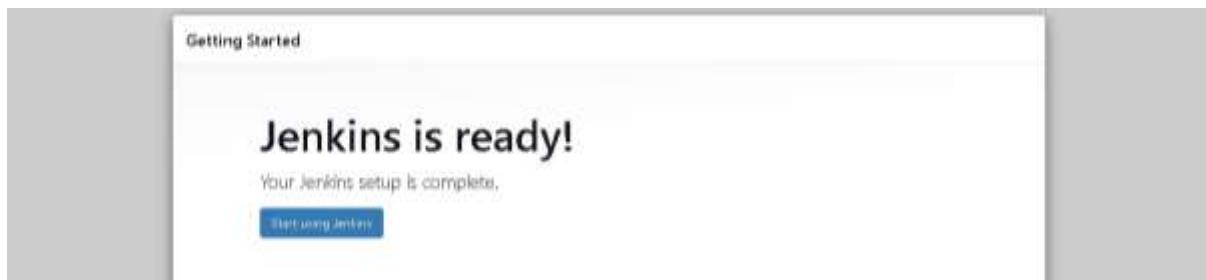


18. Give user name, password, full name and email ID and click save and continue.

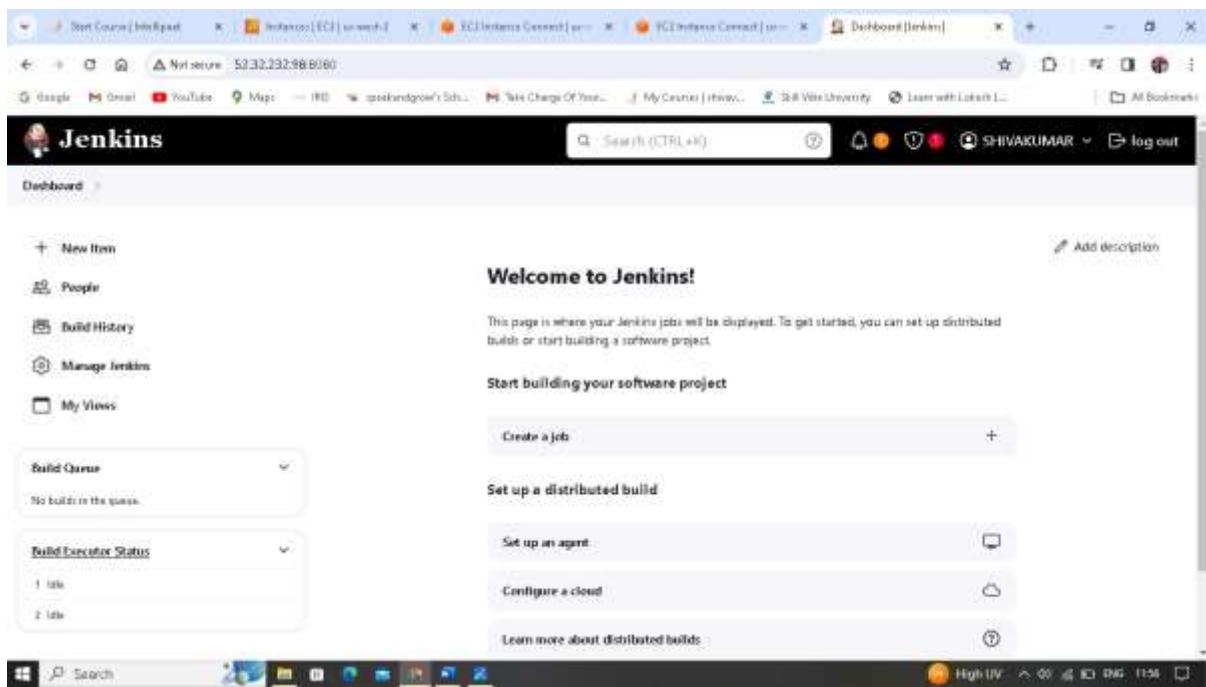


19 clicks save and finish





20. With this Jenkins dashboard is setup.



21. So far command line used.

```
1. Master
2. 1 sudo apt update
3. 2 sudo apt install openjdk-11-jdk -y
4. 3 sudo wget -O /usr/share/java/jenkins-keyring.jce - https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
5. 4 echo deb [signed-by=/usr/share/java/jenkins-keyring.jce] https://pkg.jenkins.io/debian-stable/ | sudo tee /etc/apt/sources.list.d/jenkins.list >> /dev/null
6. 5 sudo apt-get update
7. 6 sudo apt-get install jenkins -y
8. 7 sudo curl -s https://raw.githubusercontent.com/jenkinsci/docker/master/Dockerfile > /var/jenkins_home/jenkins/Dockerfile
9.
10. Slave
11. 1 sudo apt update
12. 2 sudo apt install openjdk-11-jdk -y
```

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

The screenshot shows the Jenkins Manage Jenkins interface. On the left, there's a sidebar with links: '+ New Item', 'People', 'Build History', 'Manage Jenkins' (which is selected and highlighted in blue), and 'My Views'. The main content area has a title 'Manage Jenkins'. Below it, a message states: 'Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.' It includes three buttons: 'Set up agent', 'Set up slave', and 'Slaves...'. A yellow banner at the bottom says 'Java 11 end of life in Jenkins' with 'Mark info' and 'Ignore' buttons. The bottom of the screen shows a Windows taskbar with icons for Start, Search, File Explorer, Task View, and others.

23. Go to system configuration and click on Nodes.

The screenshot shows the Jenkins System Configuration page. The sidebar on the left shows 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 idle, 2 idle). The main area has a title 'System Configuration'. It features several sections: 'System' (Configure global settings and paths), 'Tools' (Configure tools, their locations and automatic installers), 'Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), 'Nodes' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), 'Clouds' (Add, remove, and configure cloud instances to provision agents on-demand), 'Appearance' (Configure the look and feel of Jenkins), 'Security' (Secure Jenkins; define who is allowed to access/use the), 'Credentials' (Configure credentials), and 'Credential Providers' (Configure the credential providers and types). The bottom of the screen shows a Windows taskbar with icons for Start, Search, File Explorer, Task View, and others.

24. Click on +New Node

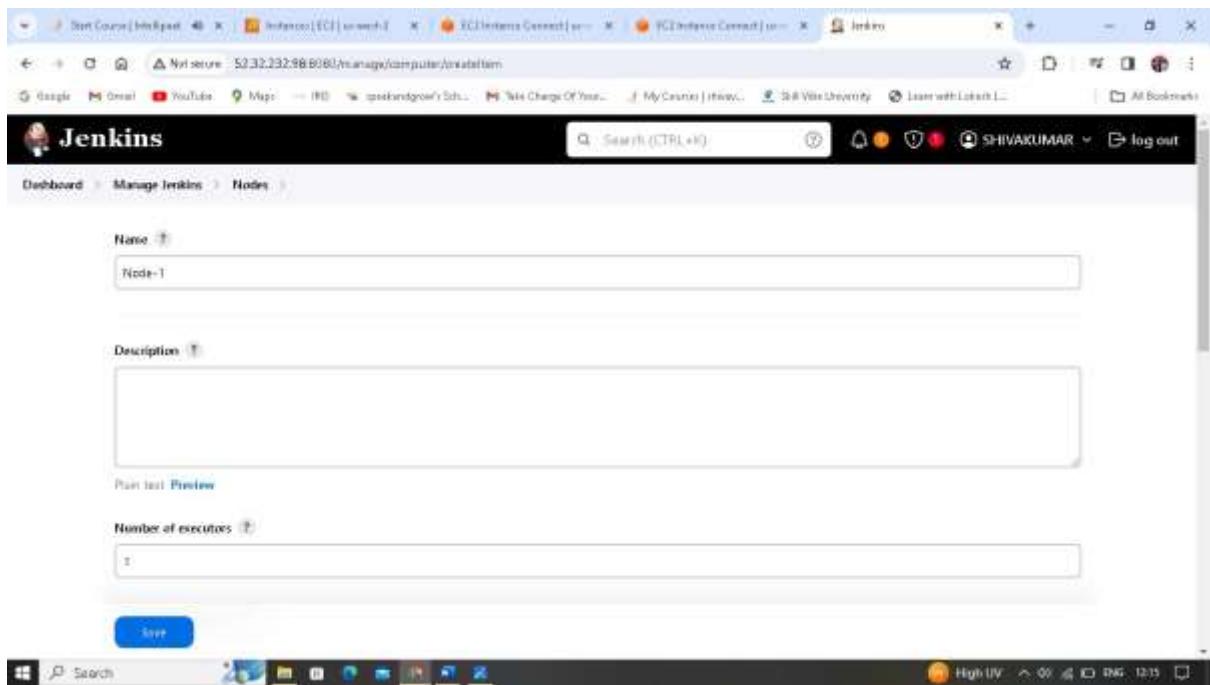
The screenshot shows the Jenkins 'Nodes' page. At the top, there is a search bar and a 'Configure Monitors' button. Below the header, there are two sections: 'Build Queue' and 'Build Executor Status'. The 'Build Queue' section shows 'No build in the queue'. The 'Build Executor Status' section shows two idle executors. The main table lists one node:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Built-In Node	Linux (amd64)	In sync	4.94 GB	0.6	4.94 GB	0ms

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

The screenshot shows the 'New node' configuration page. The 'Node name' field is set to 'Node-1'. The 'Type' section is set to 'Permanent Agent'. A descriptive note explains that this adds a plain permanent agent to Jenkins. The 'Create' button is visible at the bottom.

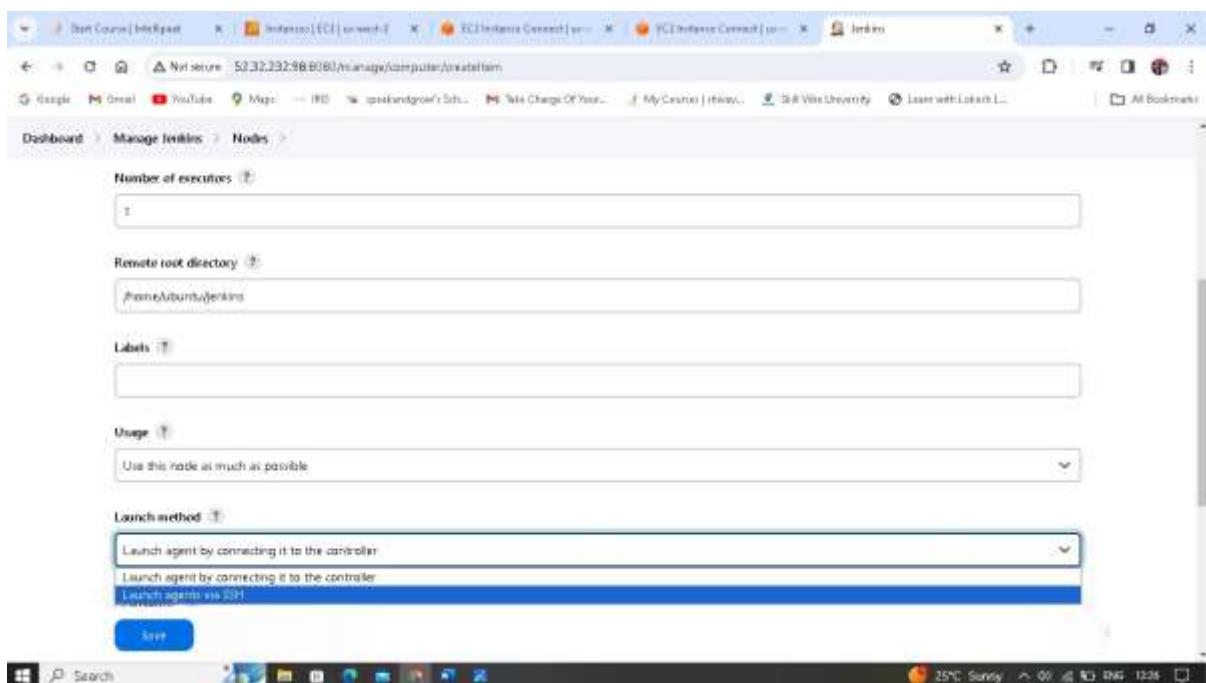
26. Now fill



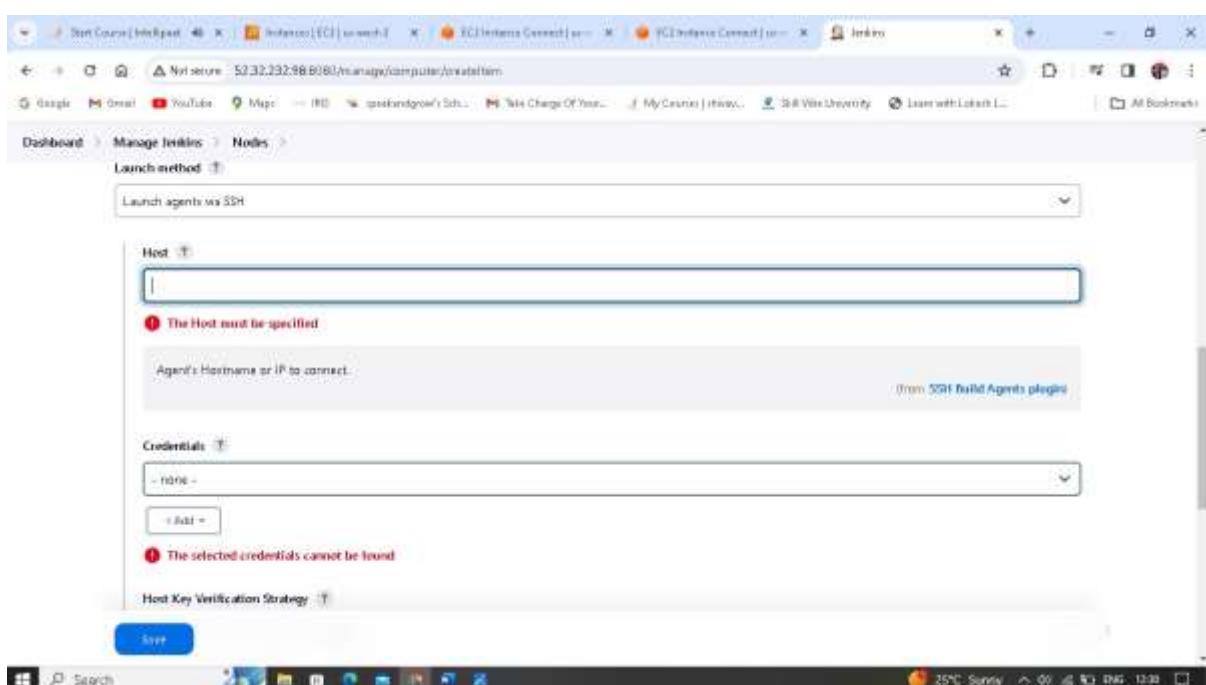
27. First of all give the Root directory. As to connect Jenkins-Node1 see the pwd (/home/ubuntu)

```
root@i-0c134b10a01dd04c:~# Last login: Thu Jan 11 04:30:46 2018 from 10.23.140.165
root@i-0c134b10a01dd04c:~# pwd
/home/ubuntu
root@i-0c134b10a01dd04c:~# i-0c134b10a01dd04c (Jenkins- Node1)
PublicIP: 52.10.59.80 PrivateIP: 172.31.25.111
```

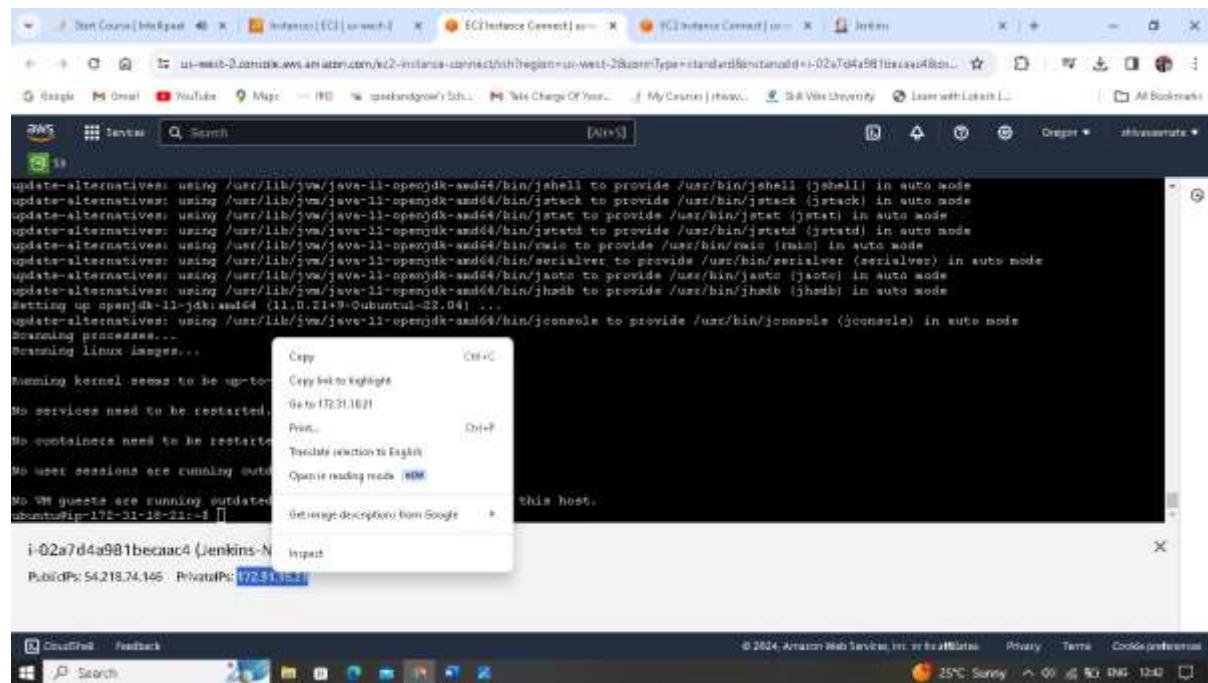
28. /home/ubuntu/jenkins as root directory.



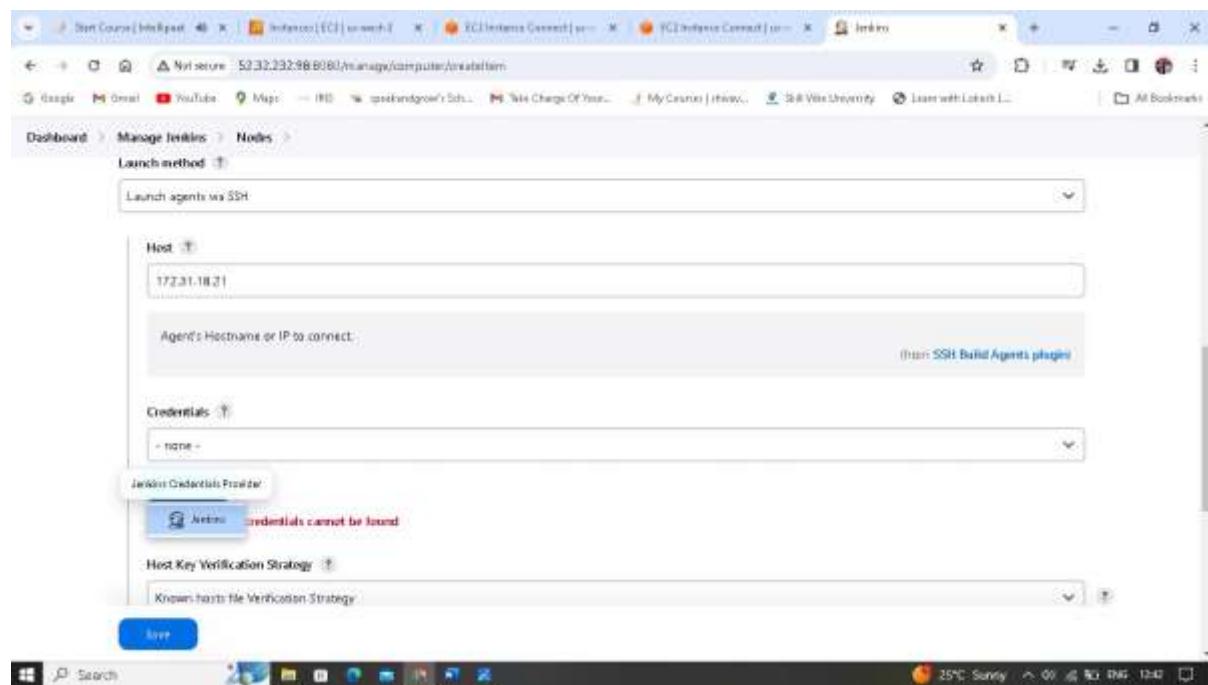
29. In Launch method select agents via SSH.



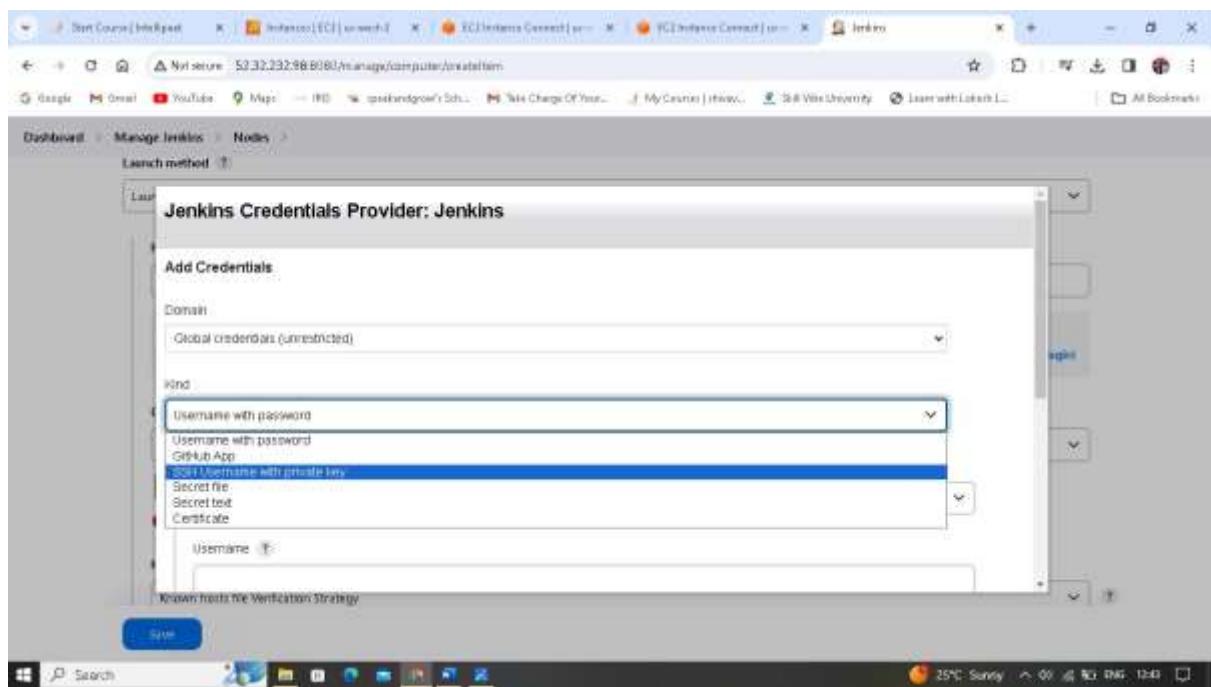
30. For Host copy and paste the private IP of Jenkins-Node1



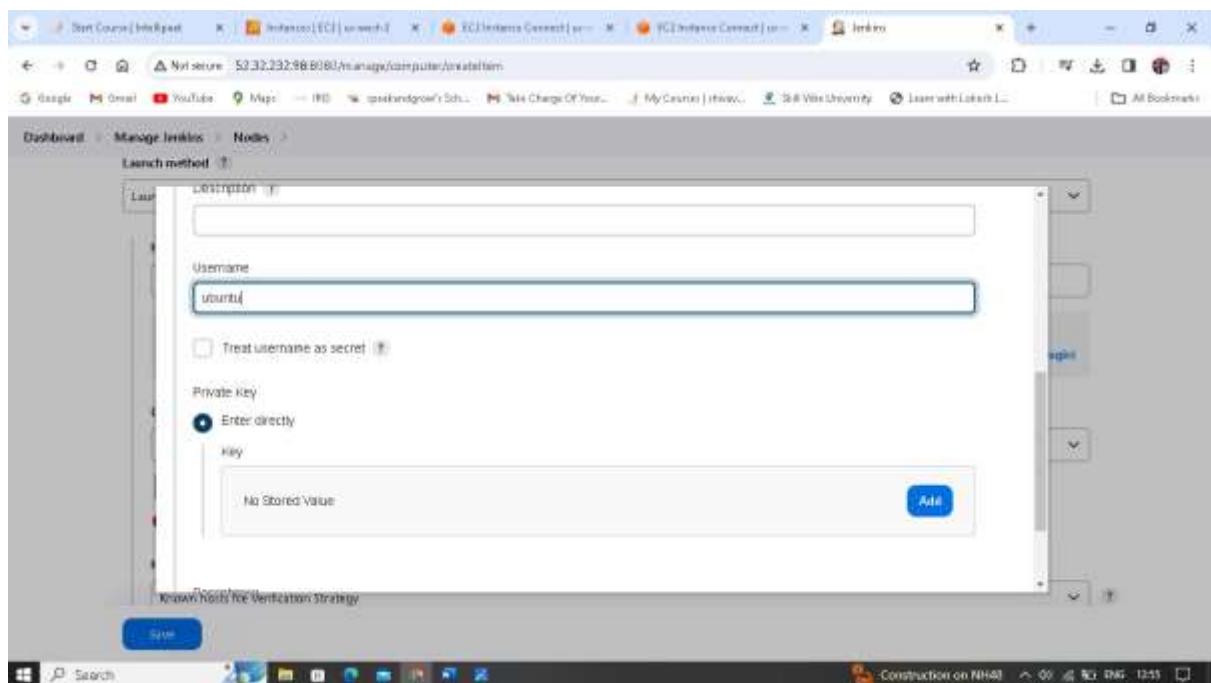
31. In the credentials click on add jenkins



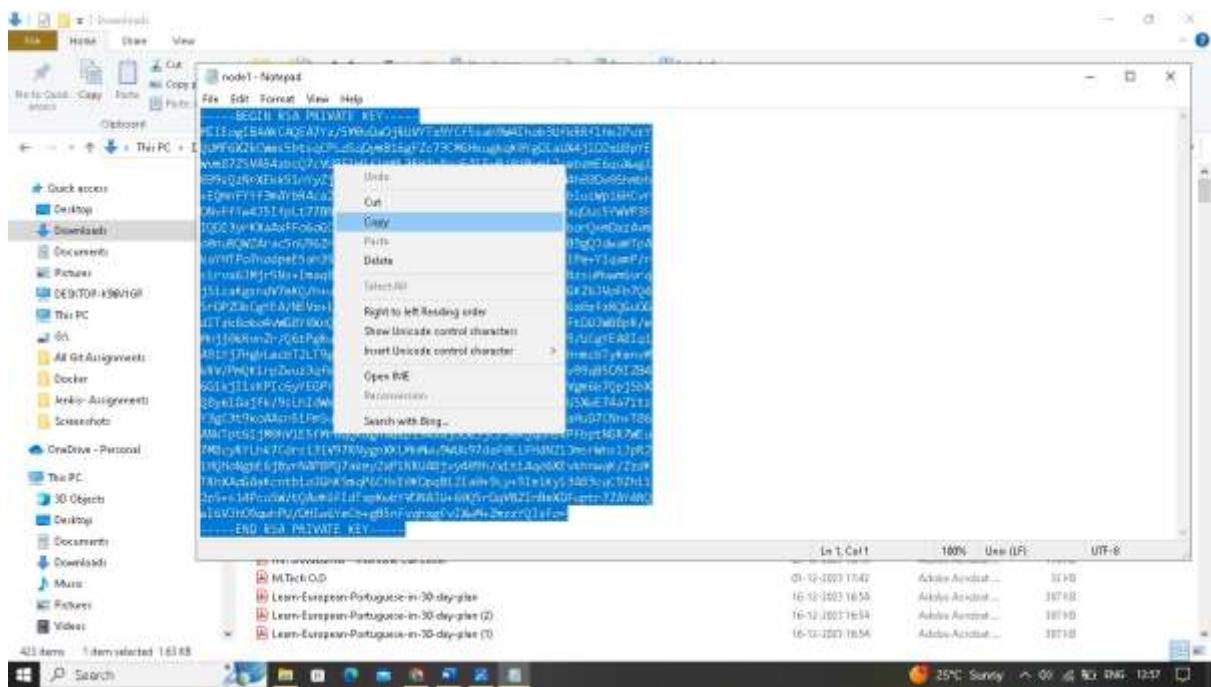
32. In kind select SSH Username with private key



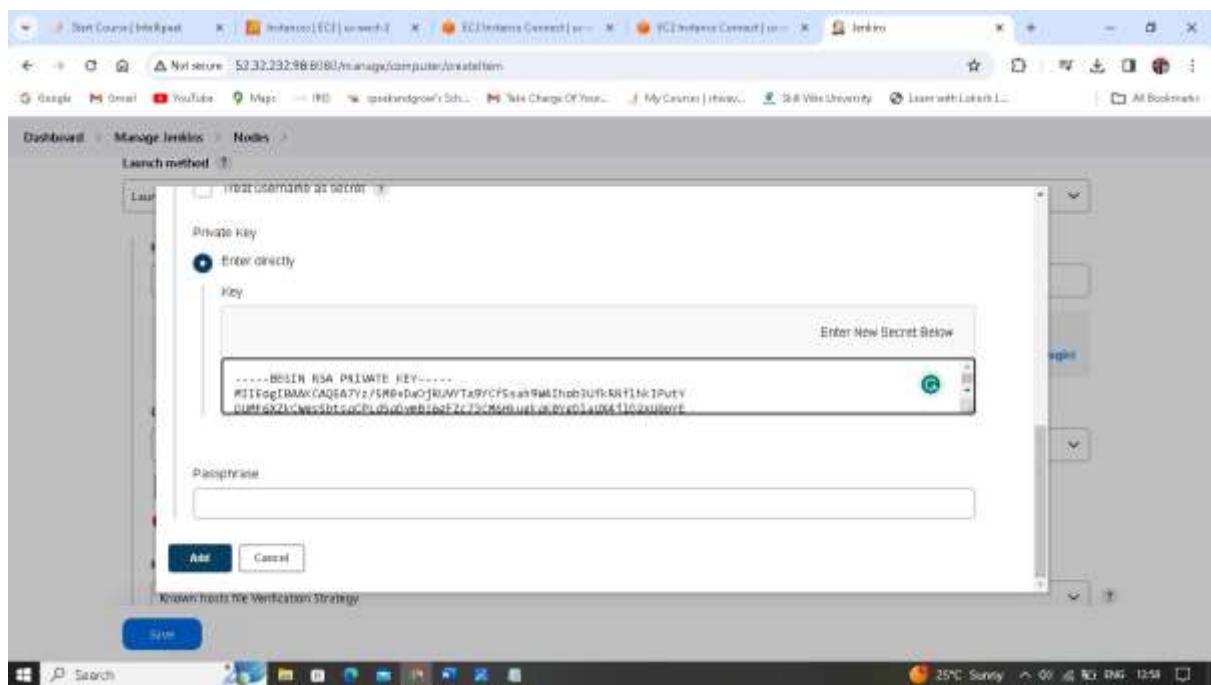
33. Give username as ubuntu click on enter directory and add



34. Copy the Jenkins-Node1 .pem file



35. Paste the private key and click on add



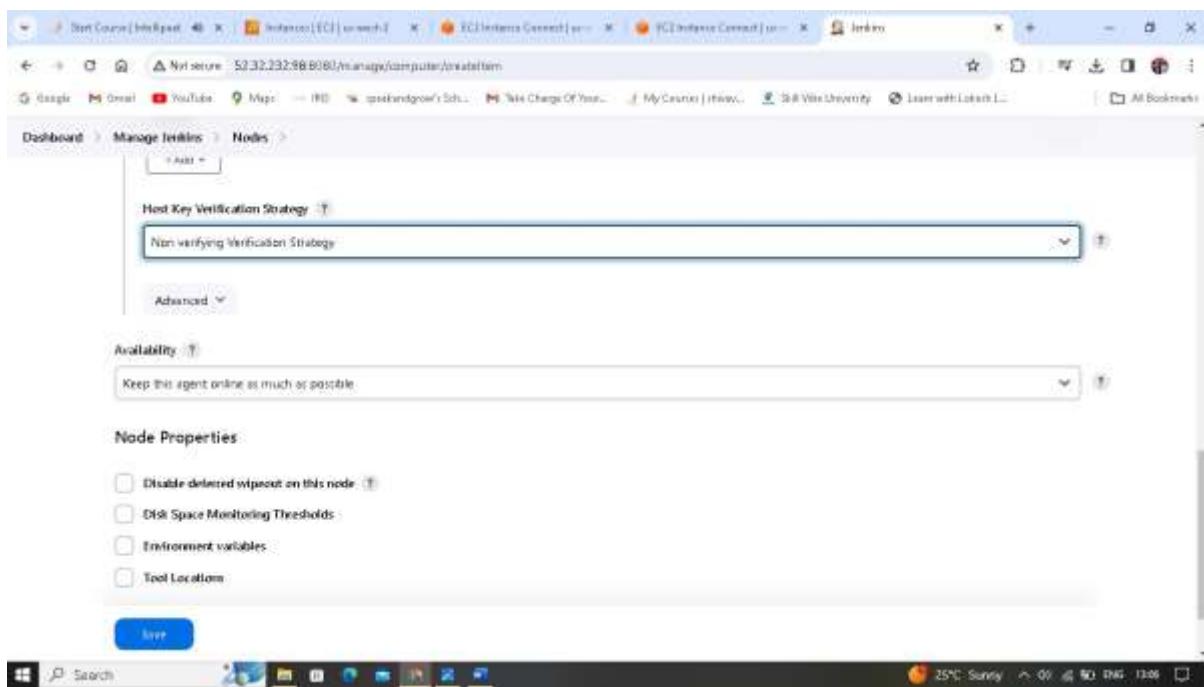
36. Now in the credentials select ubuntu

The screenshot shows the Jenkins 'Nodes' configuration page. The 'Launch method' is set to 'Launch agents via SSH'. The 'Host' field contains '172.31.18.21'. The 'Credentials' dropdown shows 'ubuntu' selected. The 'Host Key Verification Strategy' dropdown shows 'Known hosts file Verification Strategy' selected.

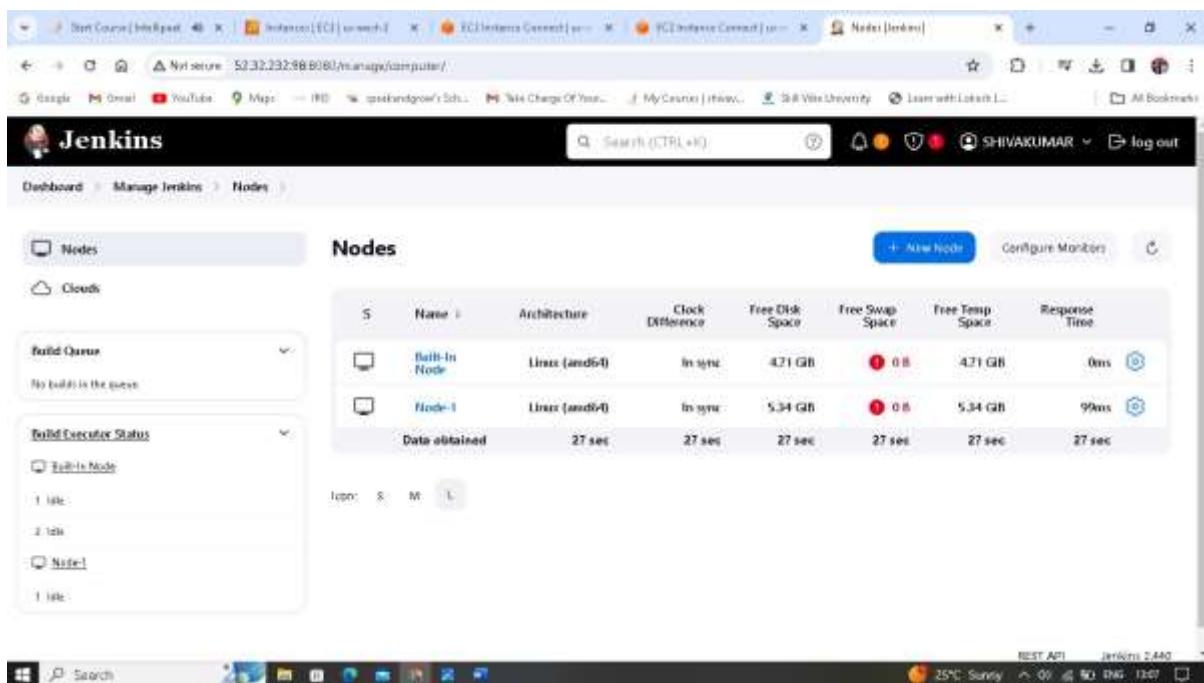
37. Select Non-verification strategy

The screenshot shows the Jenkins 'Nodes' configuration page. The 'Host Key Verification Strategy' dropdown is expanded, showing four options: 'Known hosts file Verification Strategy', 'Manually prompted key Verification Strategy', 'Manually trusted key Verification Strategy', and 'Non-verifying Verification Strategy'. The 'Non-verifying Verification Strategy' option is highlighted with a blue selection bar.

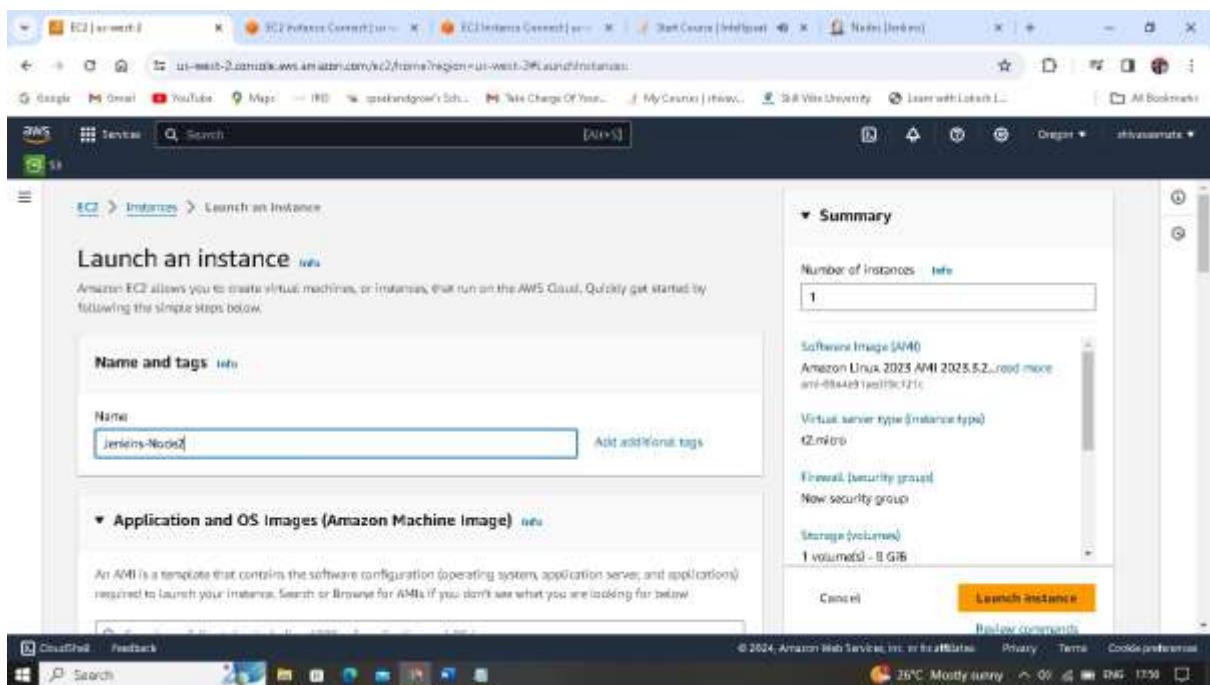
38. Click on save



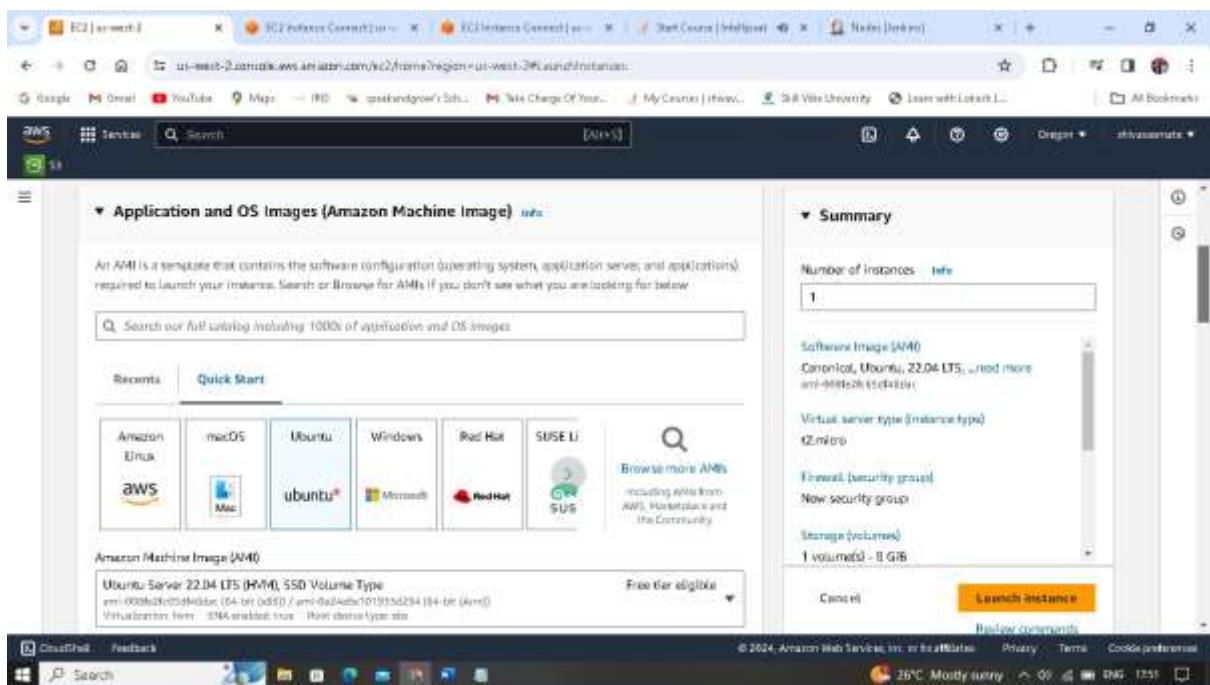
39. Successfully added Node-1



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



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



E2 | search

EC2 | Launch

EC2 Launch Configuration

EC2 Instances (1)

Start Cases (1)

Noti [link]

Google Gmail YouTube Maps

Search [A-Z]

Amazon Services

Instance

Search

Instance type

Free for eligible

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

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

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, > read more

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

Review changes

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Mostly sunny

E2 | search

EC2 | Launch

EC2 Launch Configuration

EC2 Instances (1)

Start Cases (1)

Noti [link]

Google Gmail YouTube Maps

Search [A-Z]

Amazon Services

Instance

Network settings

Edit

Network

vpc-0a610312:1597See4

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Firewall (security groups)

A security group is a set of network rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group

Selected existing security group

Selected security groups

Select security groups

default - sg-00cab0f090d1986b X

VPC: vpc-0a610312:1597See4

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, > read more

Virtual server type (instance type)

t2.micro

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

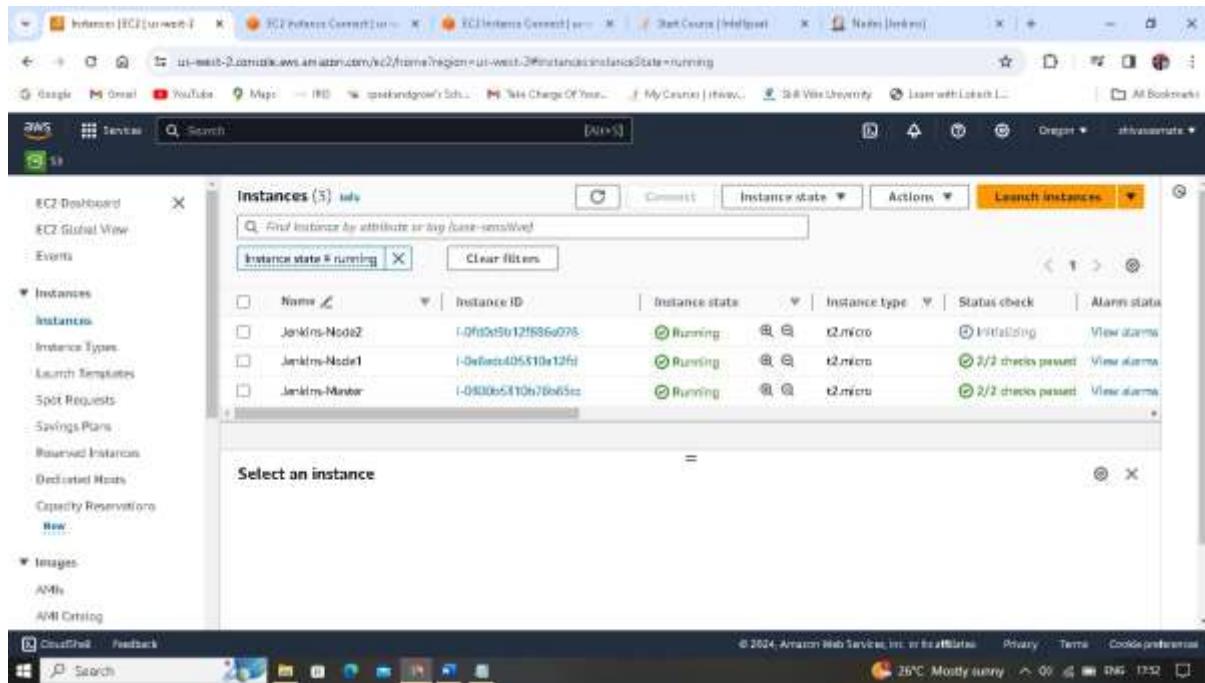
Launch instance

Review changes

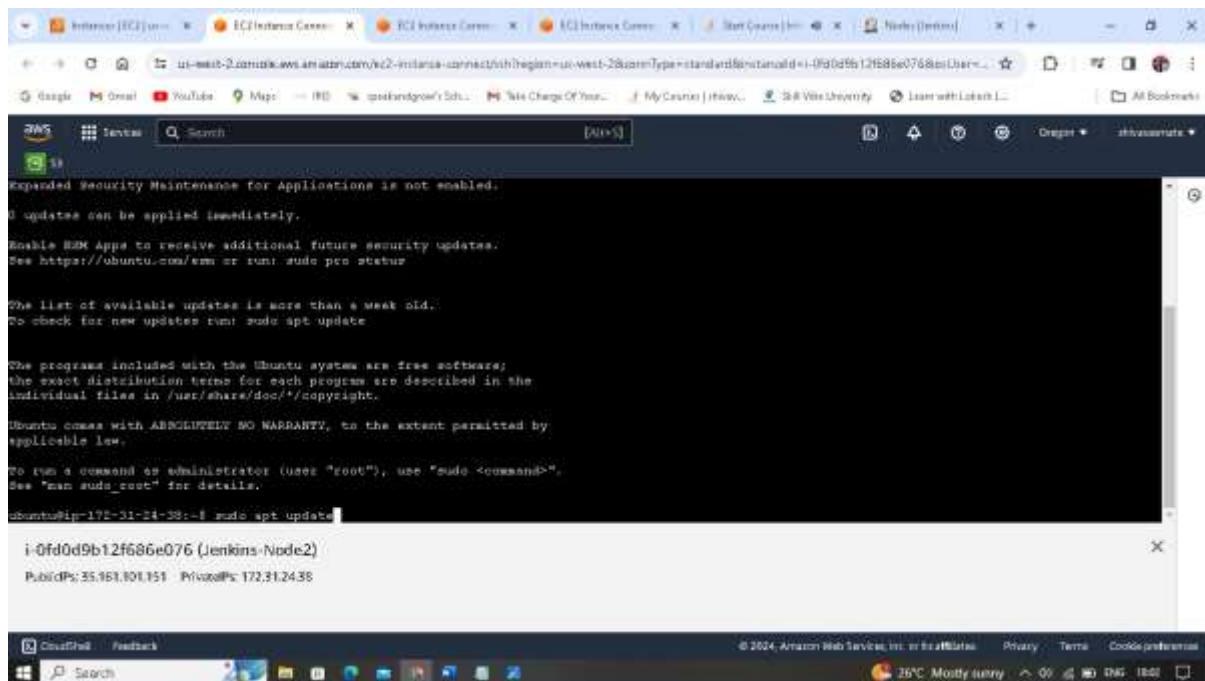
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

26°C Mostly sunny

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

```
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/universe amd64 c-n-f Metadata [164 kB]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessie-backports/multiverse amd64 c-n-f Metadata [116 kB]
Get:29 http://security.ubuntu.com/ubuntu jessie-security/main amd64 Packages [1042 kB]
Get:30 http://security.ubuntu.com/ubuntu jessie-security/main Translation-en [201 kB]
Get:31 http://security.ubuntu.com/ubuntu jessie-security/restricted amd64 Packages [1244 kB]
Get:32 http://security.ubuntu.com/ubuntu jessie-security/restricted Translation-en [203 kB]
Get:33 http://security.ubuntu.com/ubuntu jessie-security/universe amd64 Packages [826 kB]
Get:34 http://security.ubuntu.com/ubuntu jessie-security/universe Translation-en [156 kB]
Get:35 http://security.ubuntu.com/ubuntu jessie-security/universe amd64 c-n-f Metadata [146 kB]
Get:36 http://security.ubuntu.com/ubuntu jessie-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jessie-security/multiverse Translation-en [7474 kB]
Get:38 http://security.ubuntu.com/ubuntu jessie-security/multiverse amd64 c-n-f Metadata [260 kB]
Fetched 28.9 MB in 5s (3316 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
25 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-24-38:~$ sudo apt install openjdk-11-jdk -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 packages to upgrade. 0 upgrades done.
0 packages to install. 0 newly installed.
0 packages to remove. 0 removed.
0 packages to autoremove. 0 autoremoved.
ubuntu@ip-172-31-24-38:~$ sudo apt install openjdk-11-jdk -y
[1] 0fd0d9b12f686e076 (Jenkins-Node2)
PublicIP: 35.165.101.151 PrivateIP: 172.31.24.38
```

45. Go to Jenkins dashboard click on +New Node

The screenshot shows the Jenkins dashboard with the 'Nodes' section selected. There are two nodes listed:

S.	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Build-In Node	Linux (amd64)	In sync	4.07 GB	0.8 GB	4.07 GB	0ms
2	Node-1	Linux (amd64)	In sync	5.34 GB	0.8 GB	5.34 GB	0ms

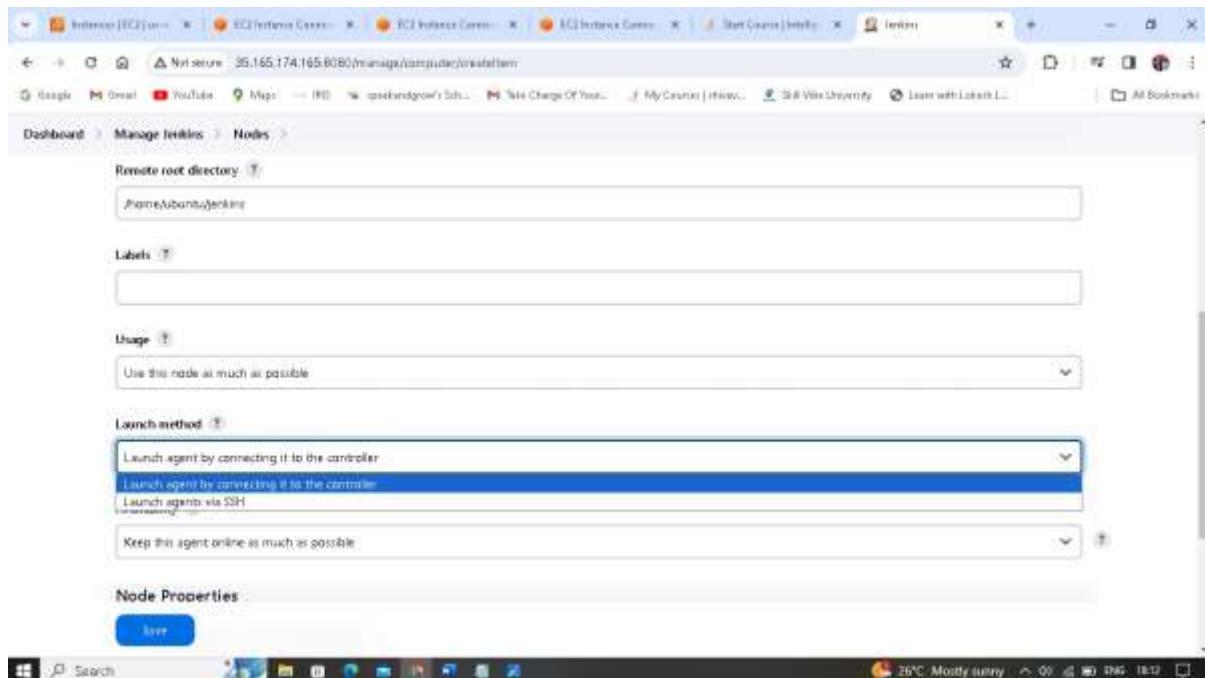
46. Create a new node with a name Node-2 as a type permanent agent.

The screenshot shows the Jenkins 'New node' configuration page. The URL in the browser is `http://192.168.174.165:8080/manage/computer/new`. The page title is 'Jenkins'. The navigation bar includes 'Dashboard', 'Manage Jenkins', 'Nodes', and 'New node'. The main section is titled 'New node'. It has a 'Node name' field containing 'Node-2'. A 'Type' section shows a radio button for 'Permanent Agent' (which is selected) and another for 'Copy Existing Node'. Below these are 'Advanced' and 'Configure' sections. At the bottom is a blue 'Create' button.

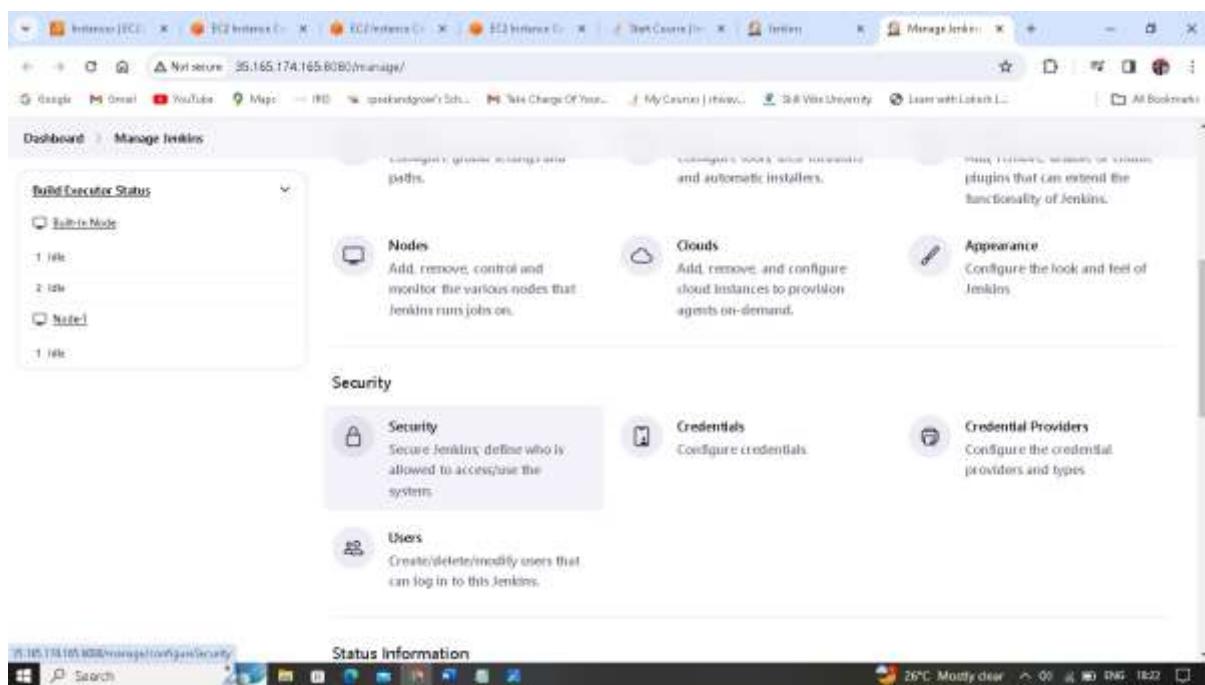
47. fill it

The screenshot shows the Jenkins 'New node' configuration page with fields filled. The URL is the same as the previous screenshot. The 'Name' field contains 'Node-2'. The 'Description' field is empty. The 'Number of executors' field contains '1'. The 'Advanced' and 'Configure' sections are visible at the bottom. The 'Create' button is present at the bottom.

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

The screenshot shows the Jenkins 'Manage Jenkins > Security' configuration page. In the 'Agents' section, the 'TCP port for inbound agents' dropdown is set to 'Random'. There are other options like 'Fixed' and 'Disable' available. Below this, the 'Agent protocols' dropdown is set to 'Live'. At the bottom, there is a 'Save' button.

51. Now save it

The screenshot shows the Jenkins 'Manage Jenkins > Nodes' configuration page. In the 'Launch method' dropdown, 'Launch agent by connecting it to the controller' is selected. In the 'Availability' dropdown, 'Keep this agent online as much as possible' is selected. Under 'Node Properties', several checkboxes are listed: 'Disable deferred wipeout on this node', 'Disk Space Monitoring Thresholds', 'Environment variables', and 'Tool Locations'. At the bottom, there is a 'Save' button.

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

The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with 'Build Queue' and 'Build Executor Status'. The main area has a table titled 'Nodes' with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. It lists three nodes: 'Built-In Node' (Architecture: Linux (amd64), State: In sync, Free Disk Space: 4.07 GB, Free Swap Space: 0.8 GB, Free Temp Space: 4.07 GB, Response Time: 0ms), 'Node-1' (Architecture: Linux (amd64), State: In sync, Free Disk Space: 5.34 GB, Free Swap Space: 0.8 GB, Free Temp Space: 5.34 GB, Response Time: 46ms), and 'Node-2' (Architecture: Data obtained, State: Offline, Free Disk Space: N/A, Free Swap Space: N/A, Free Temp Space: N/A, Response Time: N/A). Below the table, there are icons for 'S' (Sync), 'M' (Monitor), and 'L' (Log).

53. Copy the command

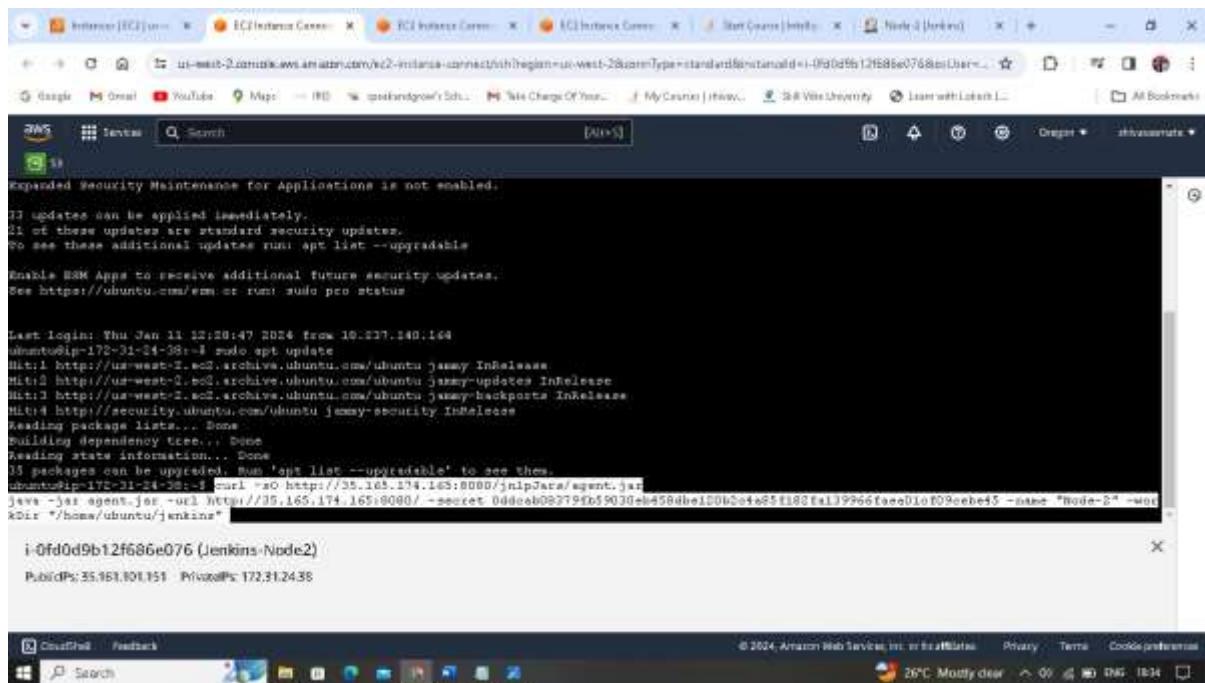
The screenshot shows the 'Agent Node-2' configuration page. On the left, there's a sidebar with 'States', 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', and 'Log'. The main area has two sections: 'Run from agent command line: (Unix)' and 'Run from agent command line: (Windows)'. Both sections contain a code editor with Jenkins agent installation commands. The Unix section contains:

```
curl -vO http://35.165.174.185:8080/jnlpJars/agent.jar
java -jar agent.jar -v1 http://35.165.174.185:8080/ -secret asdasd8379fb04898ea1580fe120b2ca4#T162fa139966fca0dcf08ec65 -name
"Node-2" -workdir "/home/ubuntu/jenkins"
```

The Windows section contains:

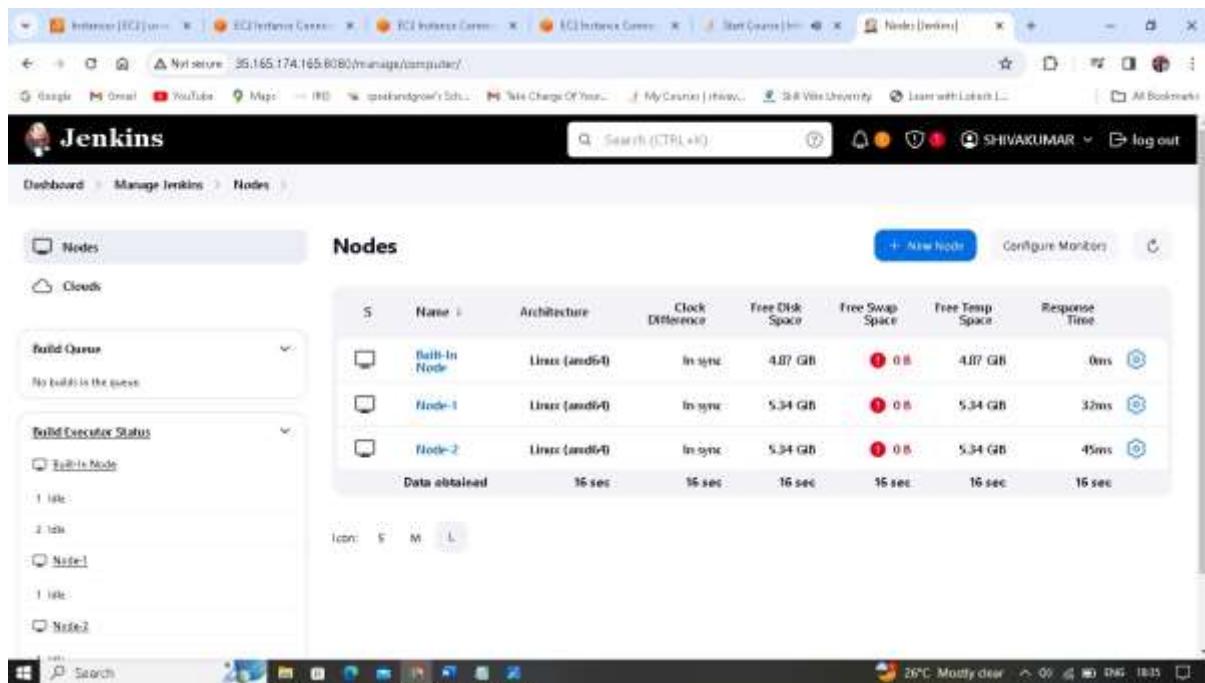
```
curl.exe -vO https://35.165.174.185:8080/jnlpJars/agent.jar
java -jar agent.jar -v1 http://35.165.174.185:8080/ -secret asdasd8379fb04898ea1580fe120b2ca4#T162fa139966fca0dcf08ec65 -name
"Node-2" -workdir "/home/ubuntu/jenkins"
```

54. Paste it on the Jenkins-Node2



```
Expanded security maintenance for Applications is not enabled.  
17 updates can be applied immediately.  
21 of those updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last Login: Thu Jan 11 12:28:47 2024 from 10.237.140.164  
ubuntu@ip-172-31-24-38:~$ sudo apt update  
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
35 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-172-31-24-38:~$ curl -sO http://35.165.174.165:8080/jnlpJars/agent.jar  
java -jar agent.jar -url http://35.165.174.165:8080/ -secret 0ddca092779fb5903d84578db100b2e4a5f180fa139966fae810d9cebe43 -name "Node-2" -nodeDir "/home/ubuntu/jenkins"  
i-0fd09b12f686e076 (Jenkins-Node2)  
PublicIP: 35.161.101.151 PrivateIP: 172.31.24.38
```

55. Now Node-2 also running.



The screenshot shows the Jenkins dashboard with the 'Nodes' section selected. The table displays the following data:

S.	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	Build-In Node	Linux (amd64)	In sync	4.07 GB	0.8 GB	4.07 GB	0ms
2	Node-1	Linux (amd64)	In sync	5.34 GB	0.8 GB	5.34 GB	32ms
3	Node-2	Linux (amd64)	In sync	5.34 GB	0.8 GB	5.34 GB	45ms

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

GitHub Repo → develop, master

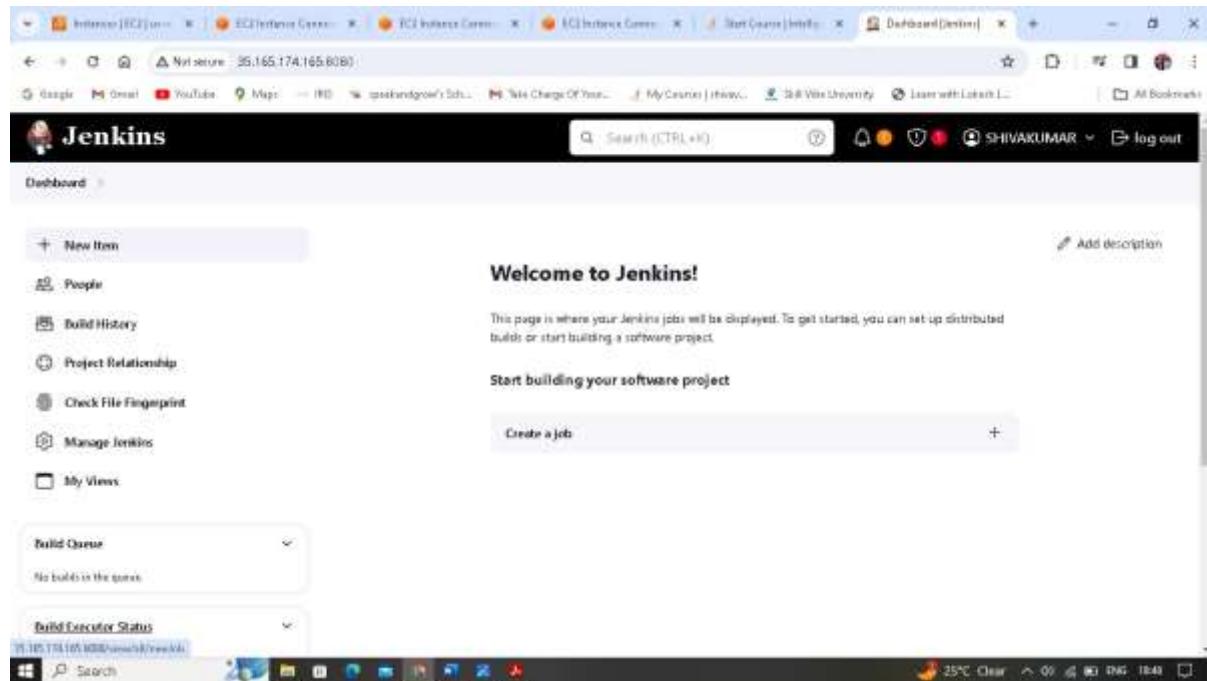
Job1 → develop branch → Node-1 (test).

Job2 → master branch → Node-2 (prod)

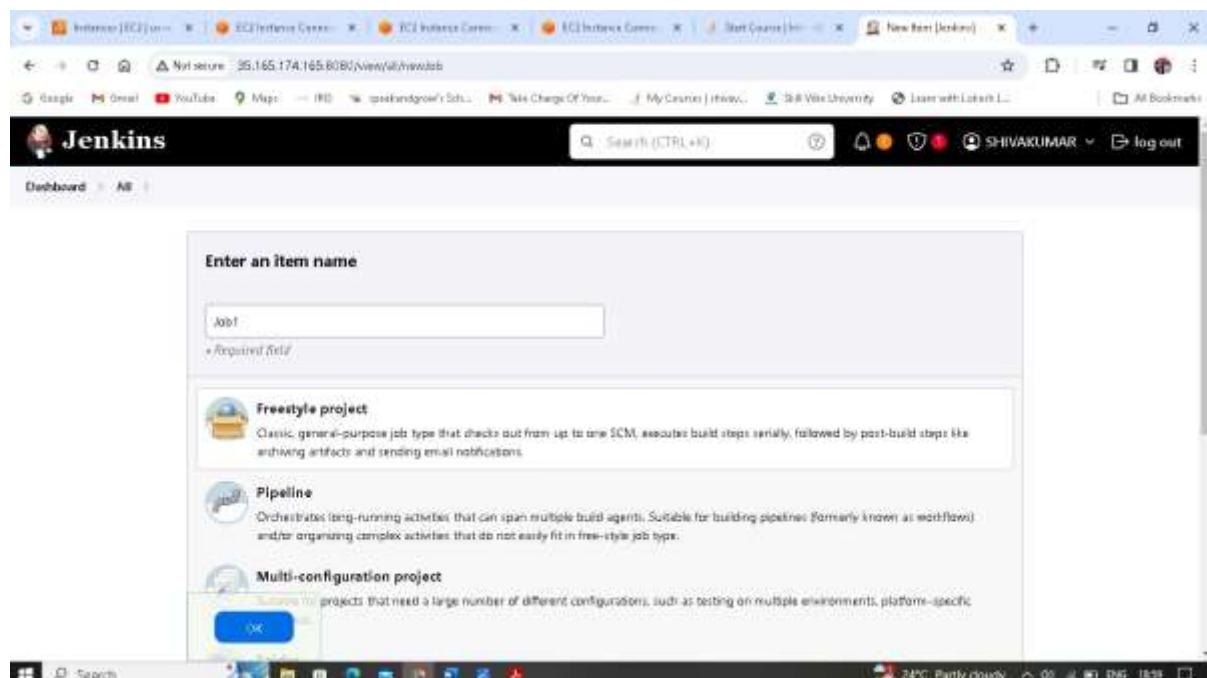
Job3 → Job1-Job2

2. FREESTYLE 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.



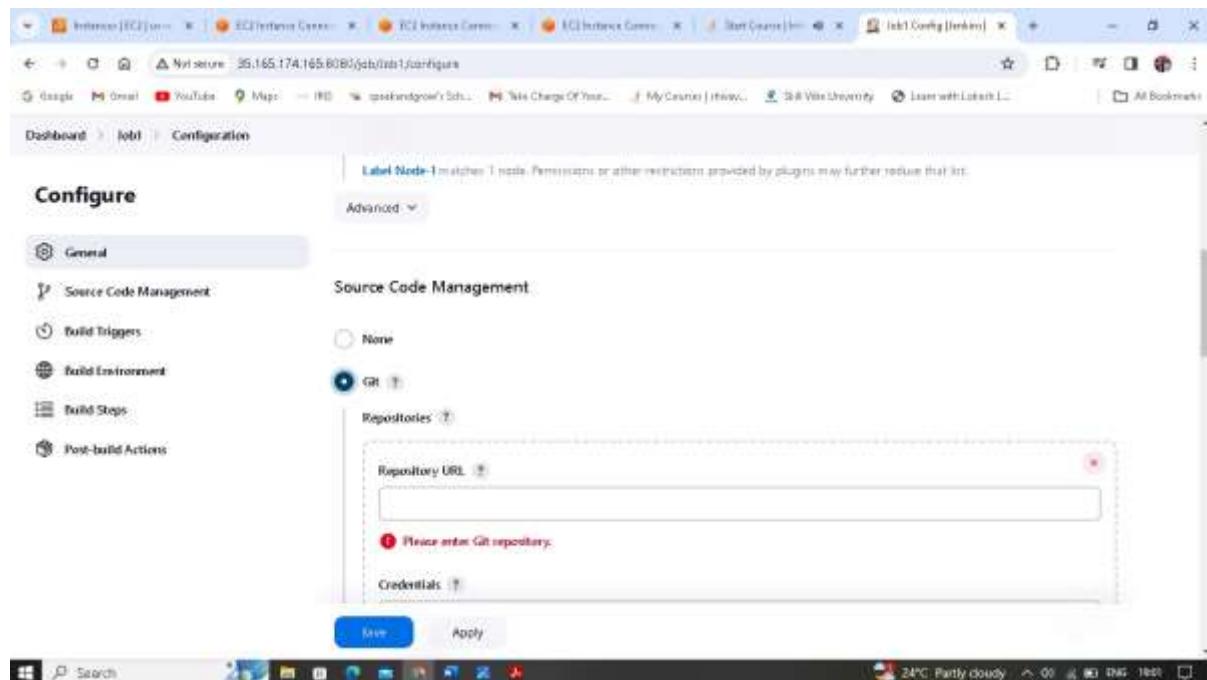
3. check it one by one and fill it

The screenshot shows the Jenkins configuration interface for a job named 'Job1'. The 'General' tab is selected. At the top right, there is a blue 'Enabled' toggle switch which is turned on. Below it, there is a 'Description' field containing a plain text preview. Under the 'Plain text' section, several checkboxes are present: 'Discard old builds', 'GitHub project', and 'This project is parameterized'. At the bottom of the configuration section are 'Save' and 'Apply' buttons.

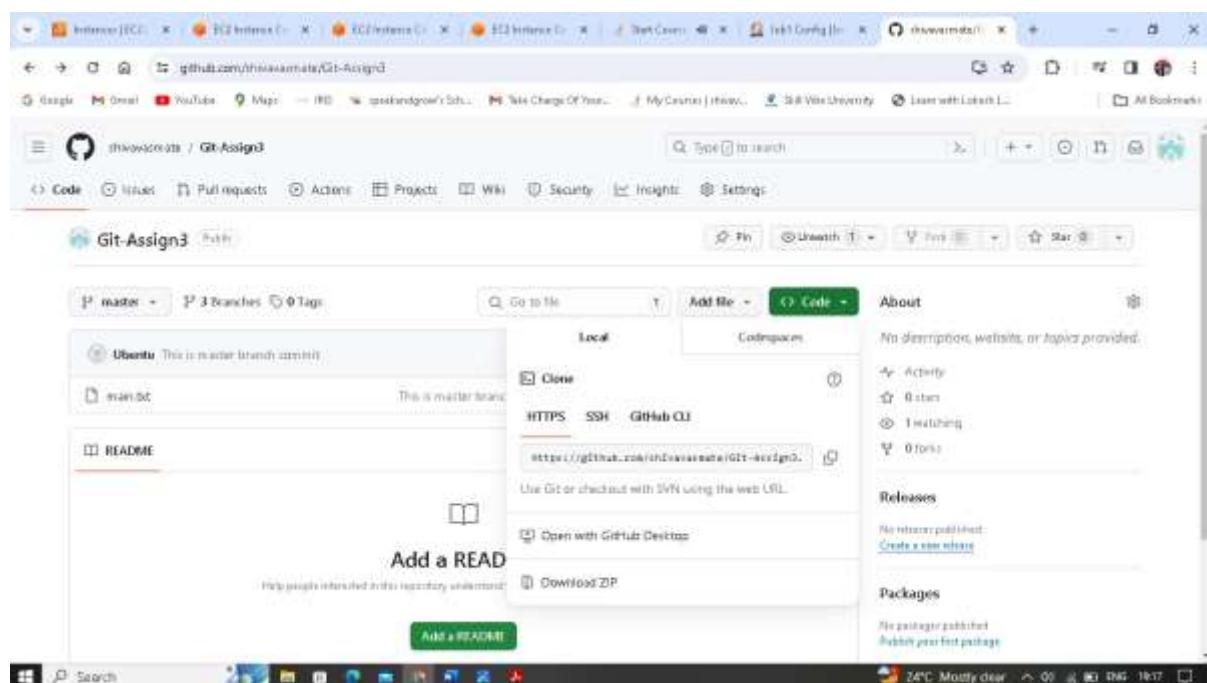
4. First of all check in Restrict where this project can be run. Label Expression as Node-1

This screenshot shows the same Jenkins configuration page for 'Job1', but with a different focus. The 'Restrict where this project can be run' checkbox is checked. Below it, the 'Label Expression' field contains the value 'Node-1'. A note below the expression states: 'Label Node-1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' There is also an 'Advanced' dropdown menu.

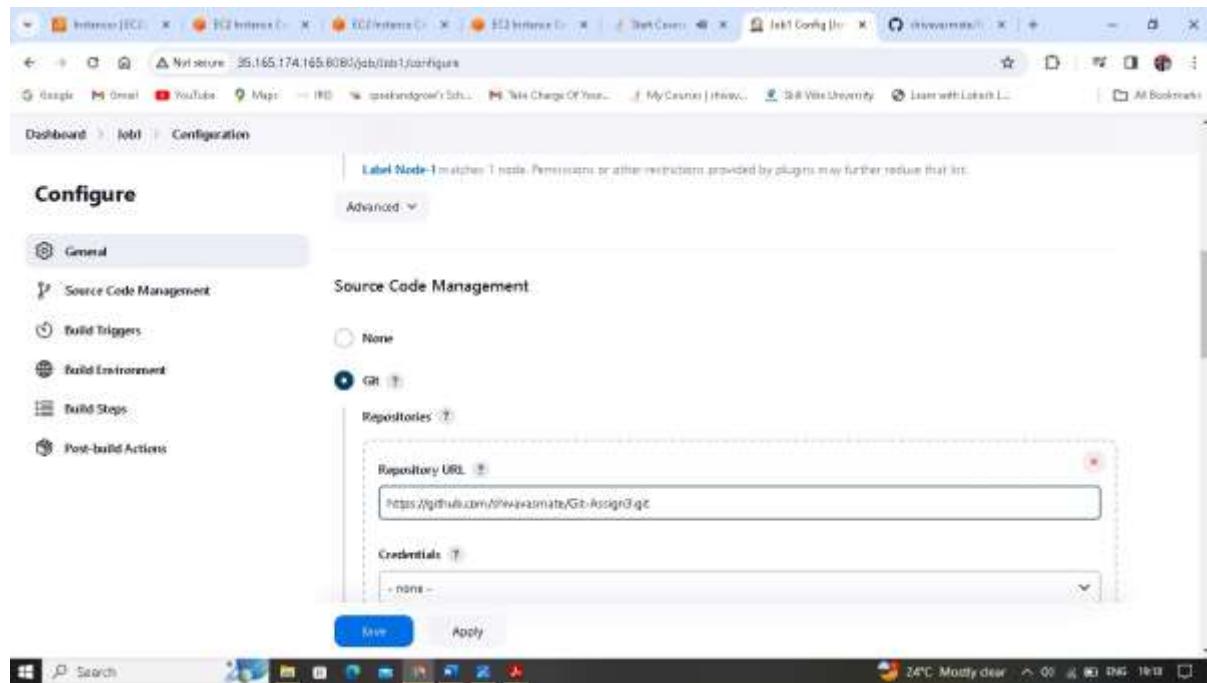
5. Under Source Code Management Check Git



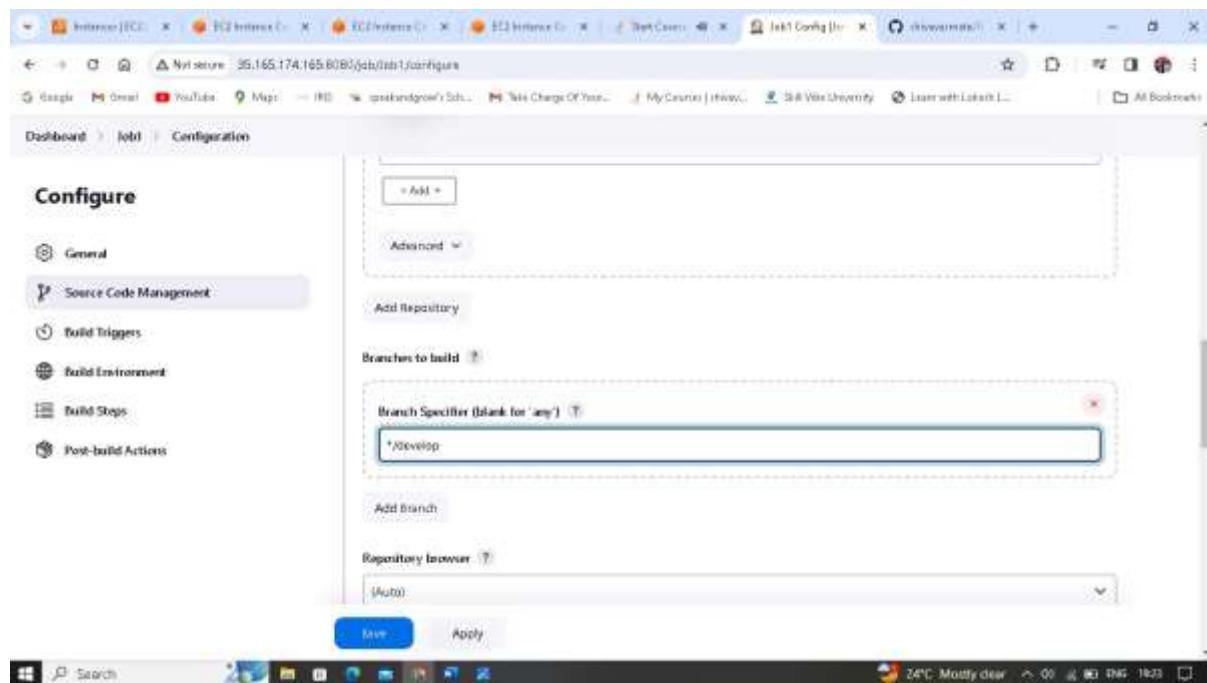
6. Now log in to your GitHub in another browser.



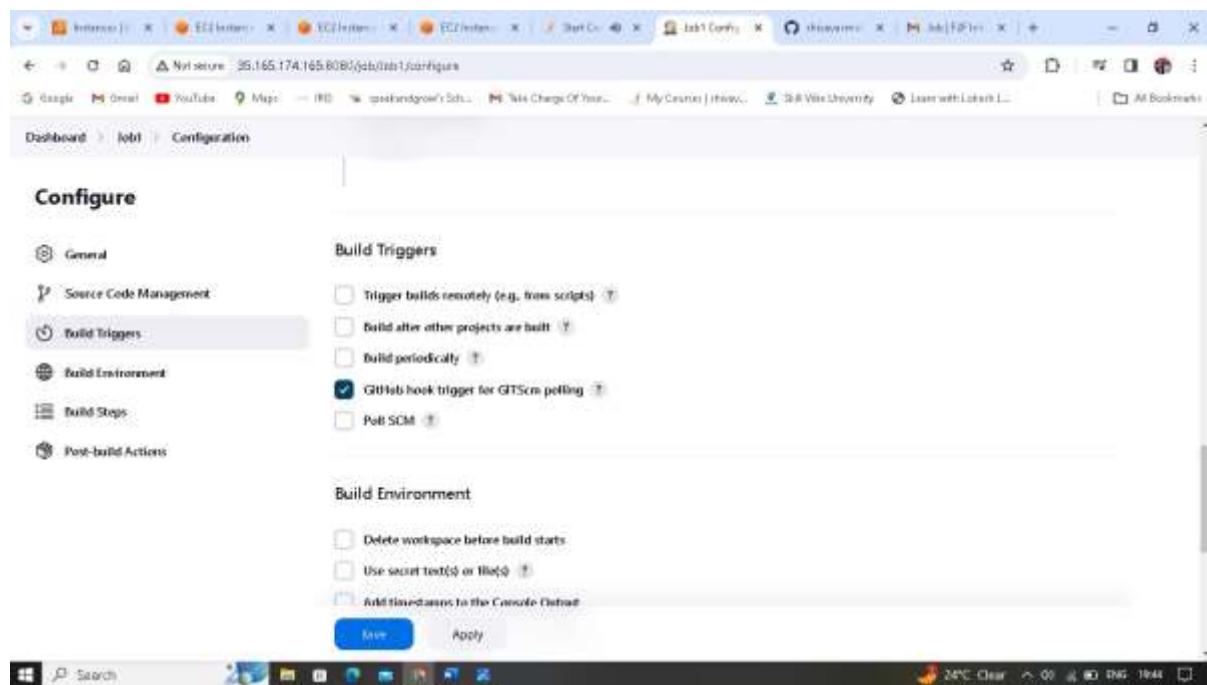
7. For Repository URL copy from code and choose the repo which has develop branch in it(Git-Assign3).



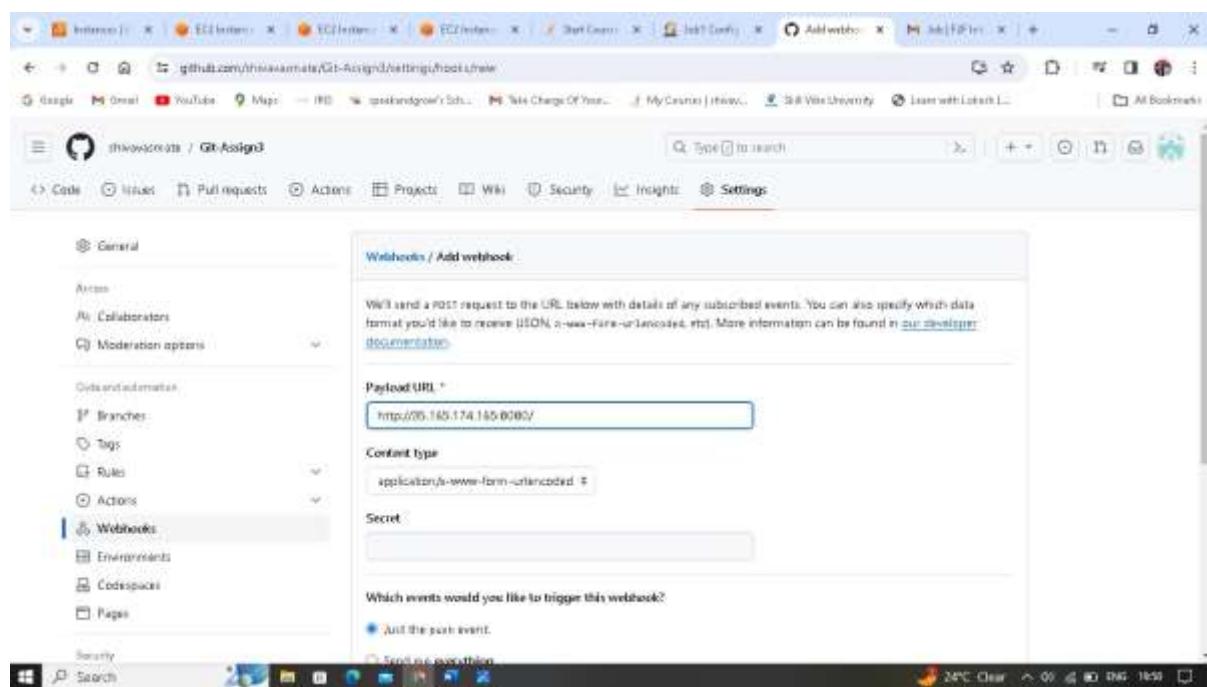
8. In Branches to build and Branch Specifier give */develop



9.Under Build Triggers choose GitHub hook 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/

The screenshot shows the Jenkins job configuration interface. The job name is 'Job1'. Under the 'Build Triggers' section, the 'GitHub hook trigger for GITScm polling' option is selected. Other options like 'Trigger builds remotely' and 'Build after other projects are built' are available but not selected. The 'Build Environment' and 'Post-build Actions' sections are also visible.

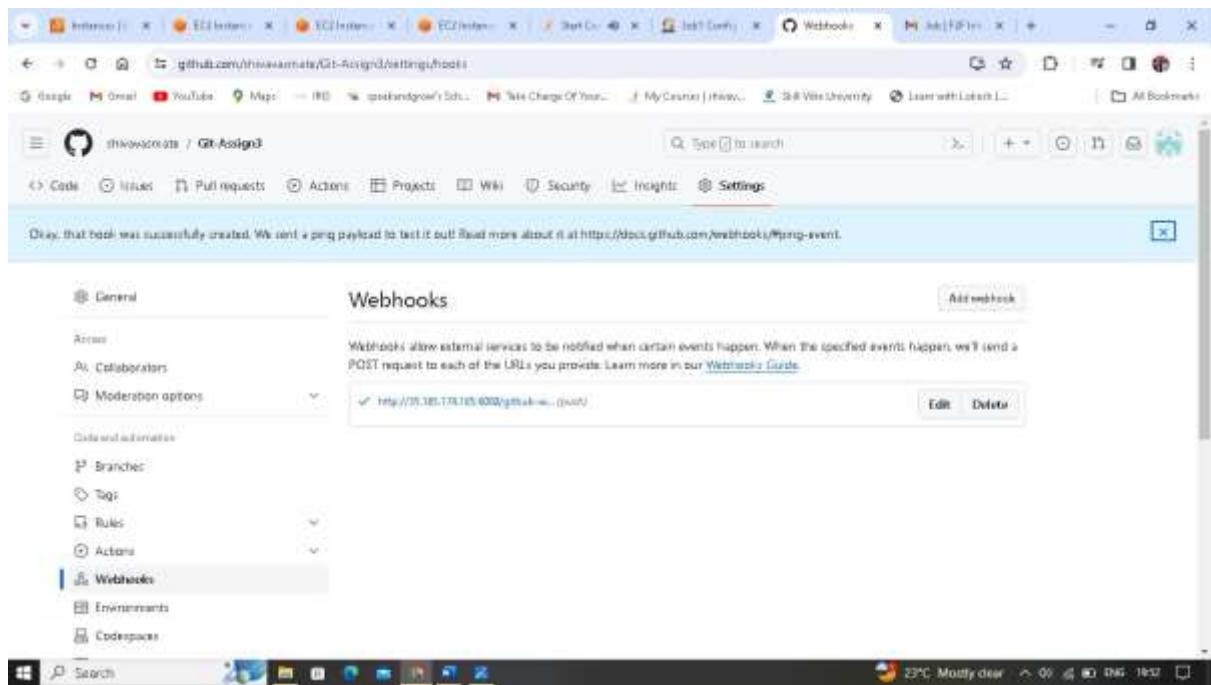
12. Give as http://35.165.174.165:8080/github-webhook/

The screenshot shows the GitHub repository settings for 'hiavamali/Cit-Agile'. In the left sidebar, 'Webhooks' is selected. On the right, a new webhook is being configured with the following details:

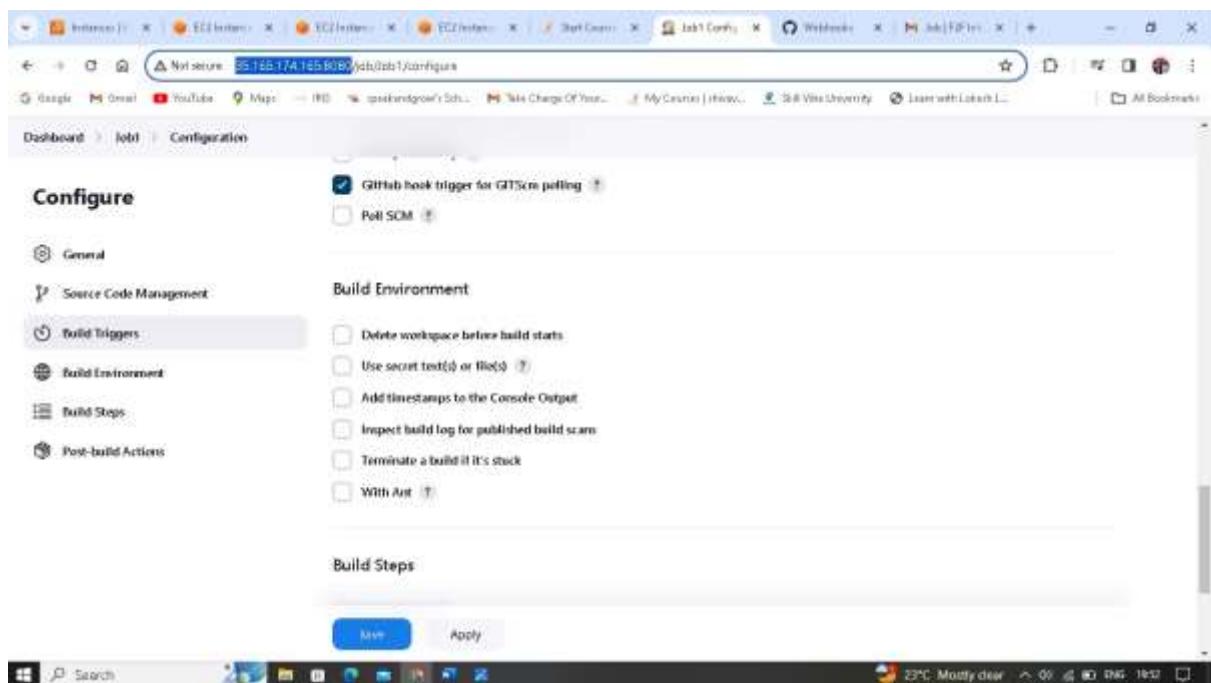
- Payload URL:** http://35.165.174.165:8080/github-webhook/
- Content type:** application/x-www-form-urlencoded
- Secret:** (empty field)
- Which events would you like to trigger this webhook?**
 - All the push event**
 - Send me everything**
 - Let me select individual events**
- Active:** (checkbox is checked)

A green 'Add webhook' button is at the bottom right.

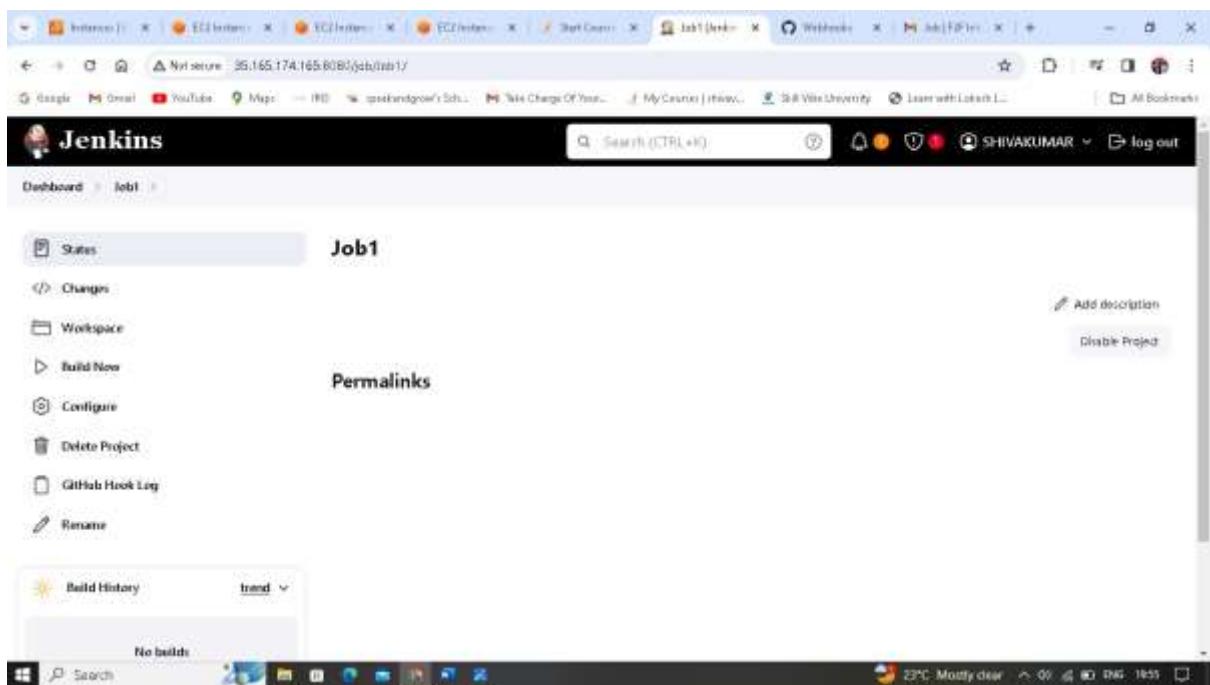
13. Check for right mark



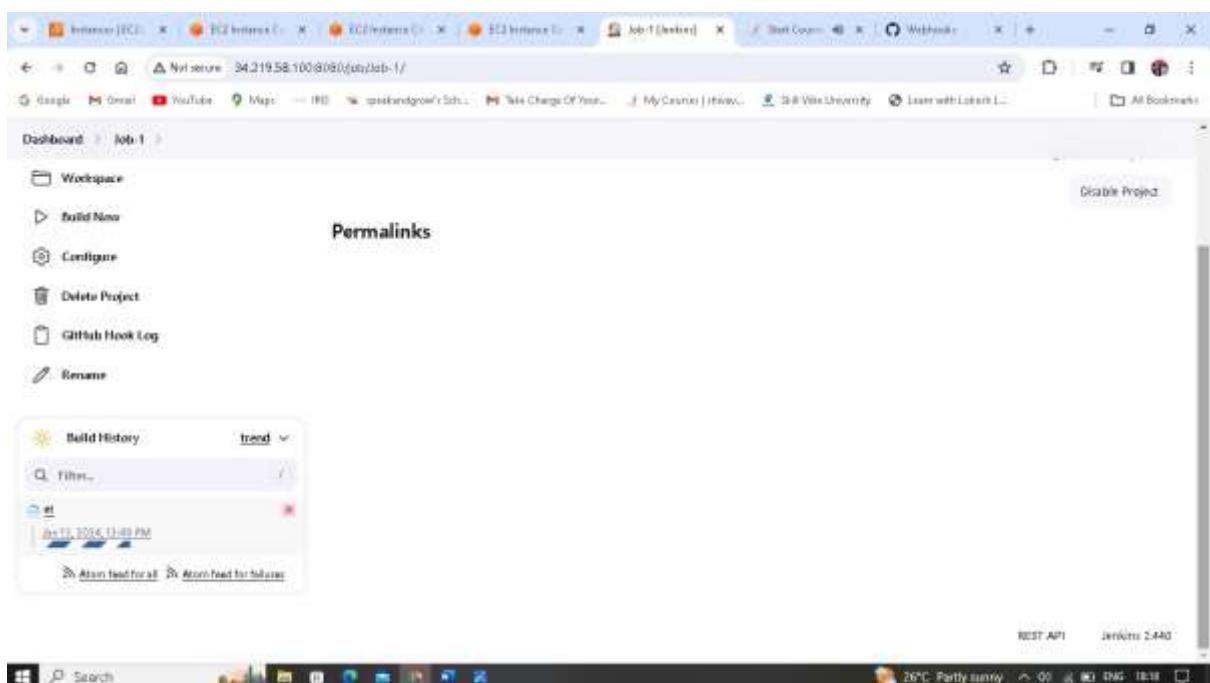
14. Go to Jenkins and click on save.



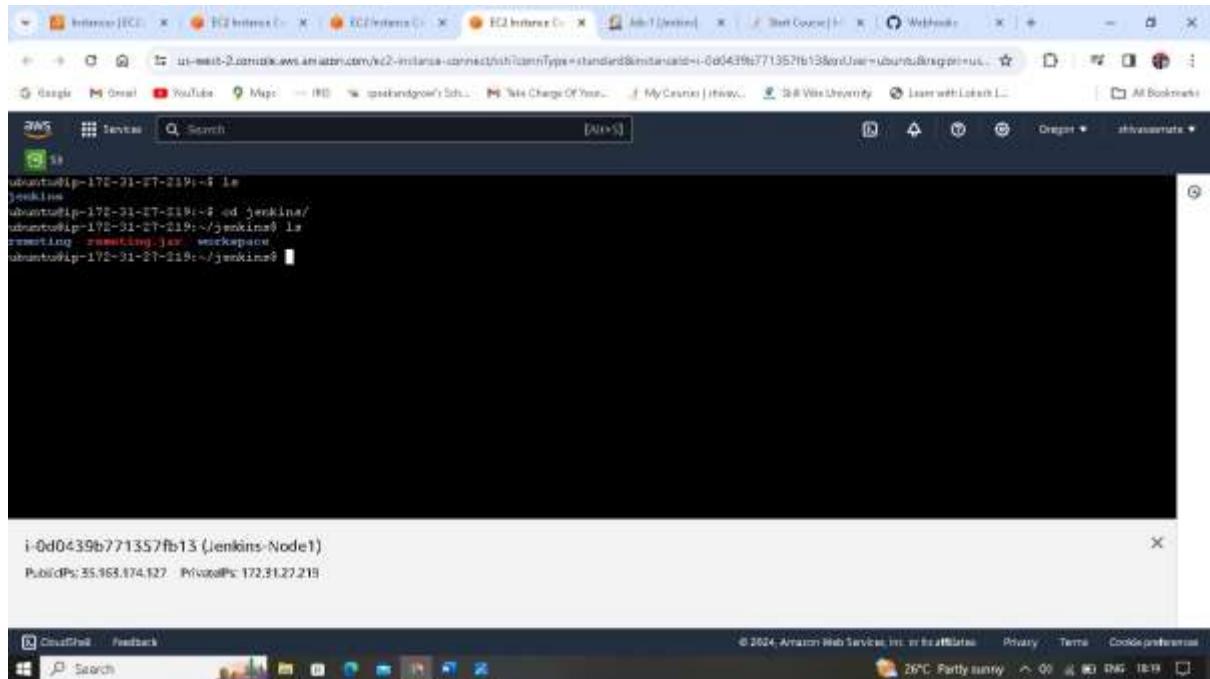
15. Job1 is created



16. Click on Build Now



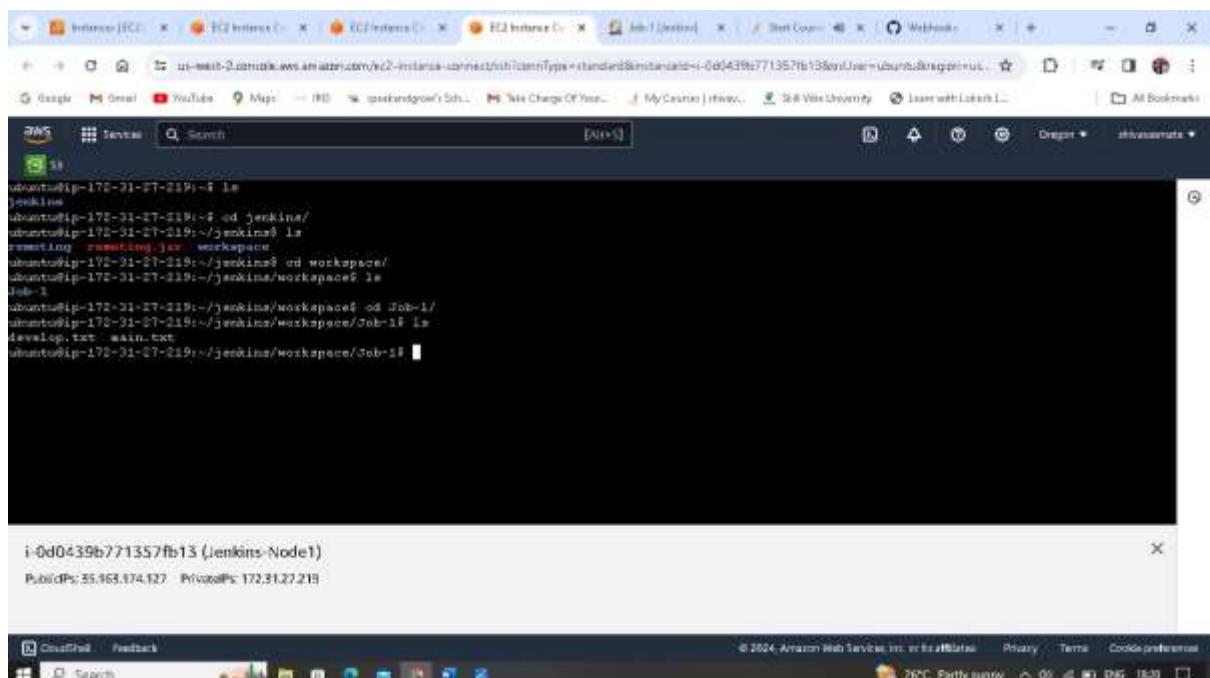
17. Go to Jenkins-Node1 do ls jenkins folder again do ls remoting remoting.jar and workspace (new one is create after build the Job1)



```
ubuntu@ip-172-31-27-219:~$ ls
jenkins
ubuntu@ip-172-31-27-219:~$ cd jenkins/
ubuntu@ip-172-31-27-219:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-27-219:~/jenkins$
```

i-0d0439b771357fb13 (Jenkins-Node1)
PublicIP: 35.163.174.127 PrivateIP: 172.31.27.219

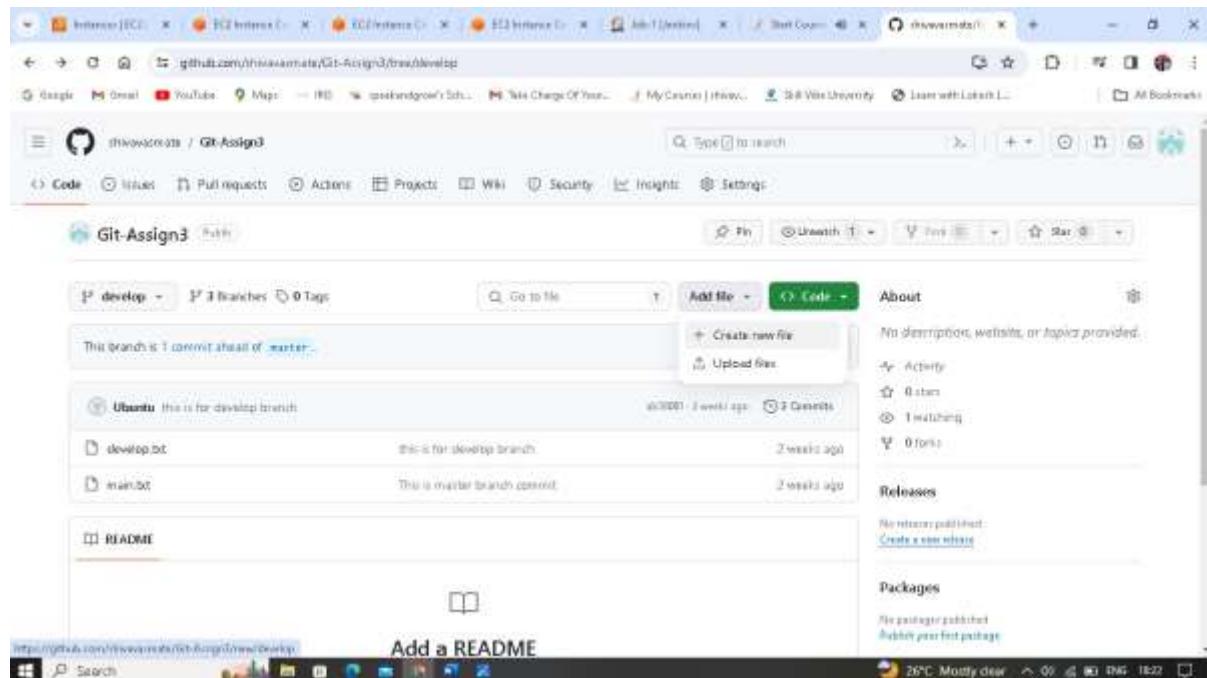
18.Under workspace do ls Job-1 folder is create and under Job-1 folder develop.txt and main.txt as same as GitHub repo.



```
ubuntu@ip-172-31-27-219:~$ ls
jenkins
ubuntu@ip-172-31-27-219:~$ cd jenkins/
ubuntu@ip-172-31-27-219:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-27-219:~/jenkins$ cd workspace/
ubuntu@ip-172-31-27-219:~/jenkins/workspace$ ls
Job-1
ubuntu@ip-172-31-27-219:~/jenkins/workspace$ cd Job-1/
ubuntu@ip-172-31-27-219:~/jenkins/workspace/Job-1$ ls
develop.txt  main.txt
ubuntu@ip-172-31-27-219:~/jenkins/workspace/Job-1$
```

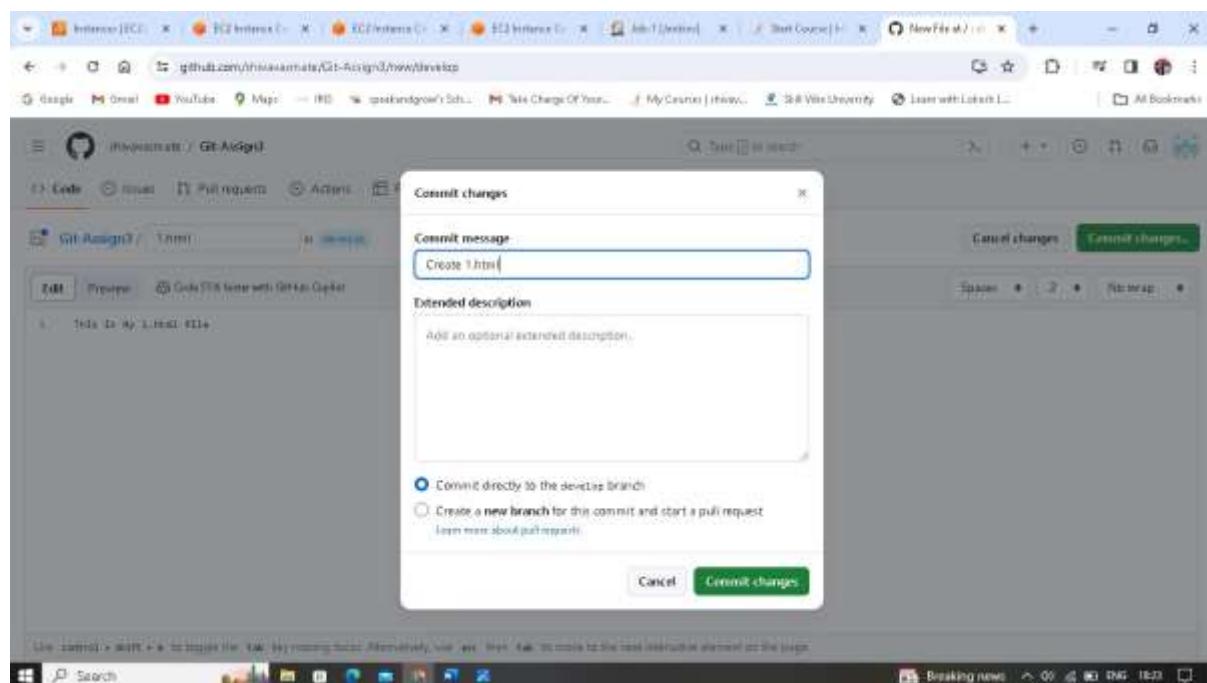
i-0d0439b771357fb13 (Jenkins-Node1)
PublicIP: 35.163.174.127 PrivateIP: 172.31.27.219

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



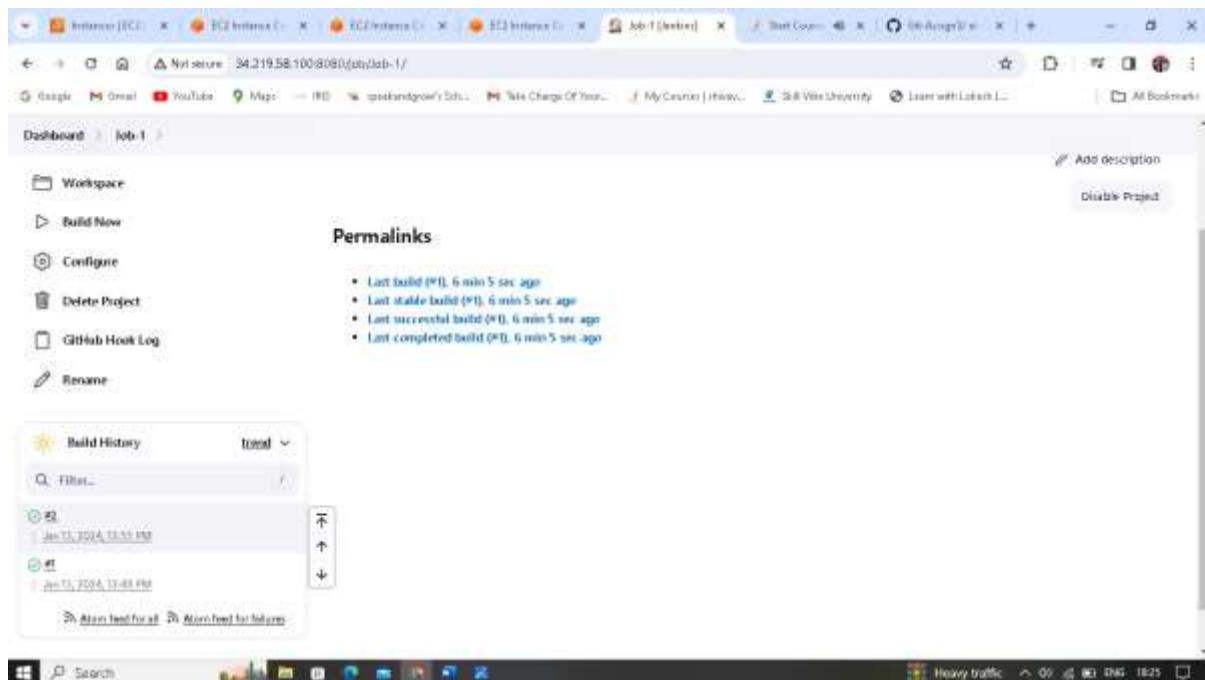
The screenshot shows a GitHub repository named 'Git-Assign3' on the 'develop' branch. There is one commit from 'Ubuntu' dated 2 weeks ago. The commit message is 'Ubuntu this is for develop branch', and the file added is '1.html.txt' with the content 'this is for develop branch'. The repository also contains a 'README' file.

20. Now click on commit changes

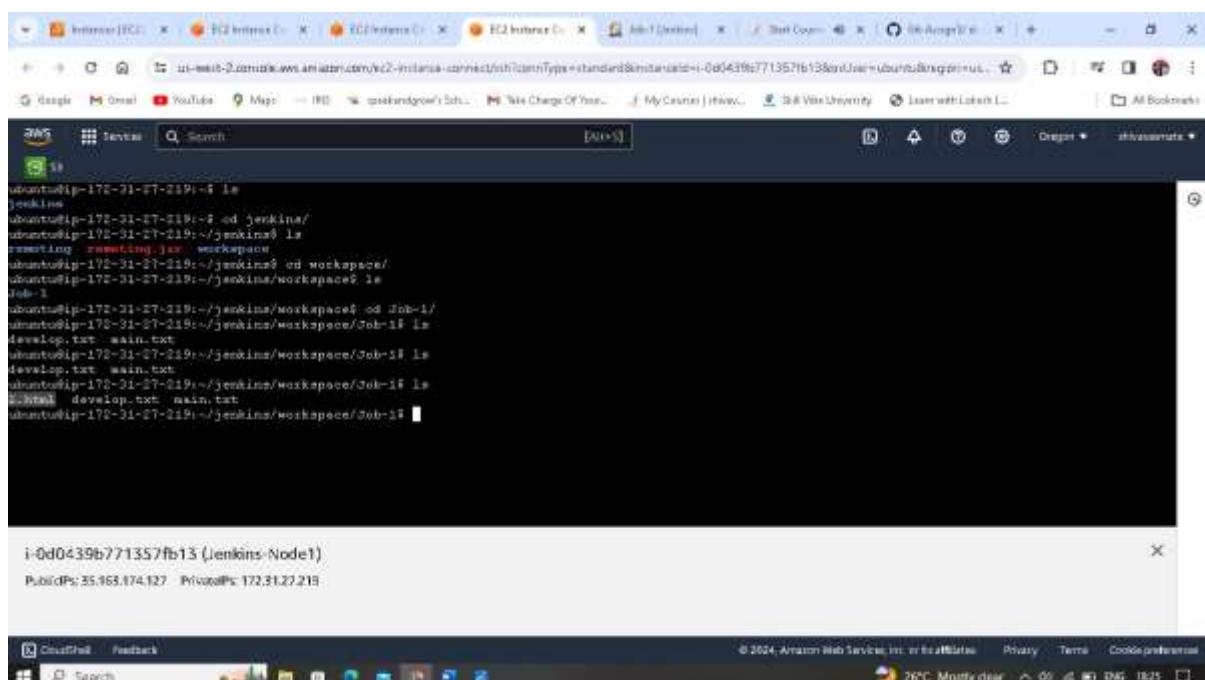


A 'Commit changes' dialog box is open. The 'Commit message' field contains 'Create 1.html'. The 'Commit directly to the develop branch' radio button is selected. The 'Commit changes...' button is highlighted in green.

21. Automatic #2 build is created



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



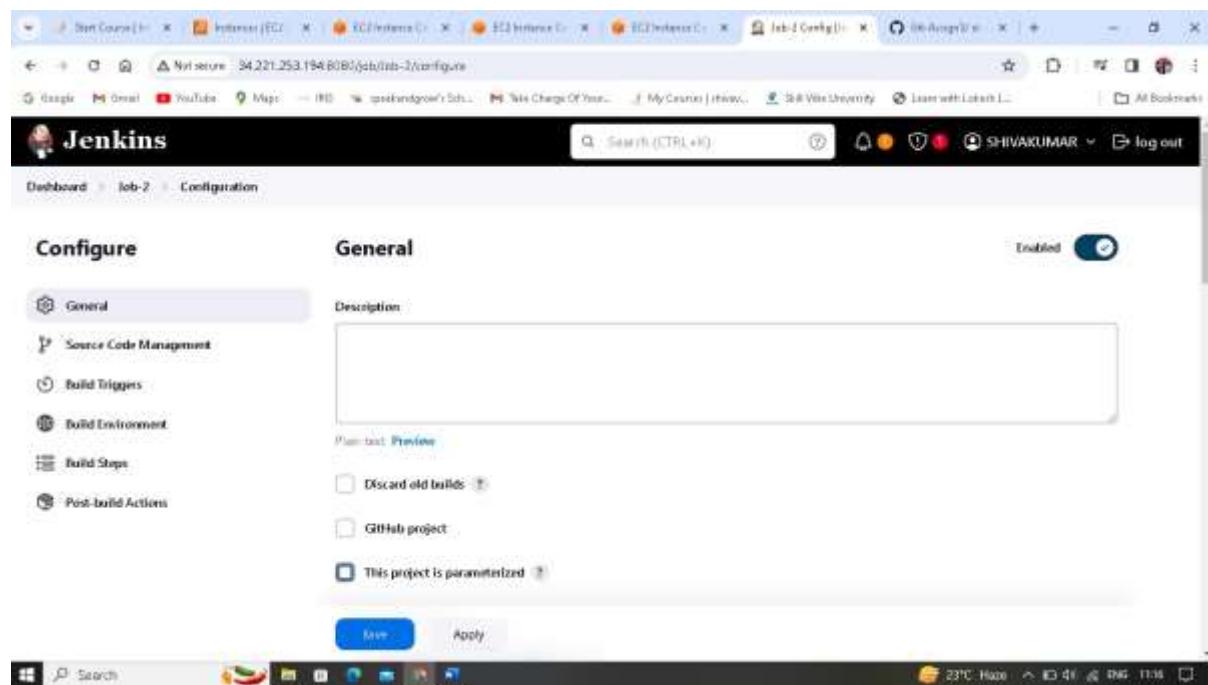
1. Go back to Dashboard and click on New item.

The screenshot shows the Jenkins dashboard with the title 'Jenkins'. At the top left, there is a 'New Item' button. Below it, there are sections for 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. A 'Build Queue' section indicates 'No builds in the queue'. A 'Build Executor Status' section shows a single executor with status 'idle'. The bottom of the screen shows a Windows taskbar with various icons and system status.

2. Enter an item name as Job-2

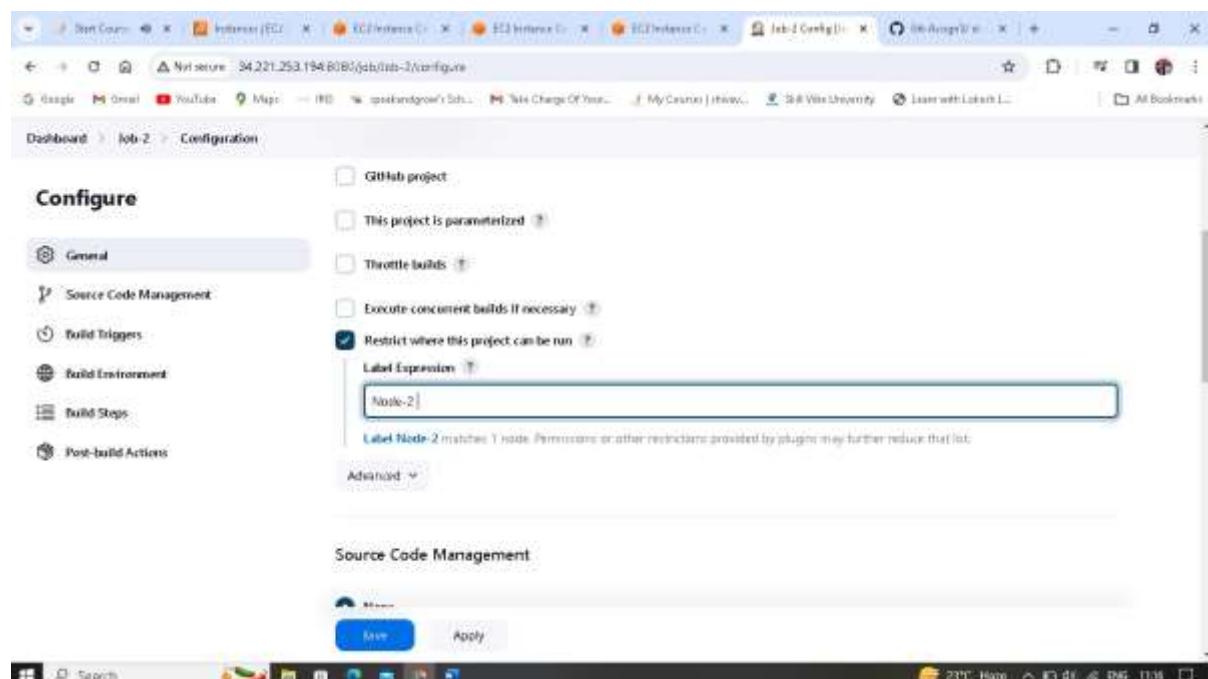
The screenshot shows the 'New Item' creation dialog. In the 'Enter an item name' field, the text 'Job-2' is entered. Below the field, there are three project types listed: 'Freestyle project', 'Pipeline', and 'Multi-configuration project'. Each has a brief description and an 'OK' button at the bottom. The background shows the Jenkins dashboard.

3. fill it



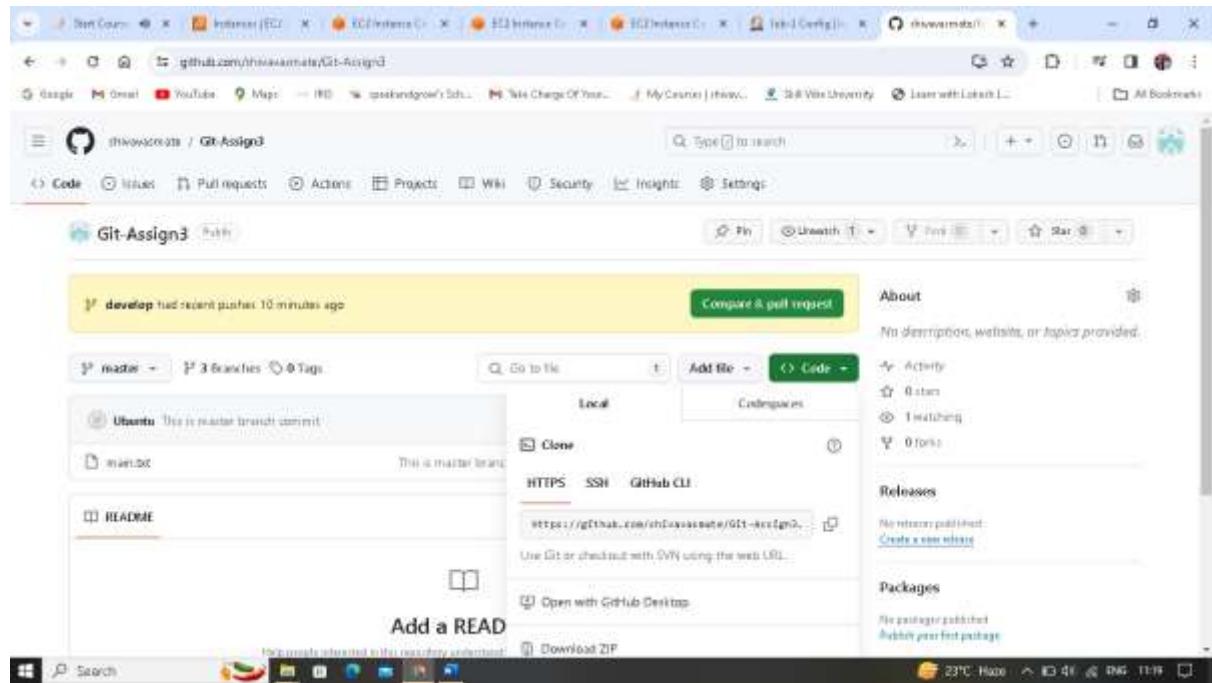
The screenshot shows the Jenkins configuration interface for a job named 'Job-2'. The 'General' tab is selected. On the left, there's a sidebar with links: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' tab has a 'Description' field which is currently empty. Below it are several checkboxes: 'Discard old builds', 'GitHub project', and 'This project is parameterized'. At the bottom are 'Save' and 'Apply' buttons.

4. In Label Expression as Node-2

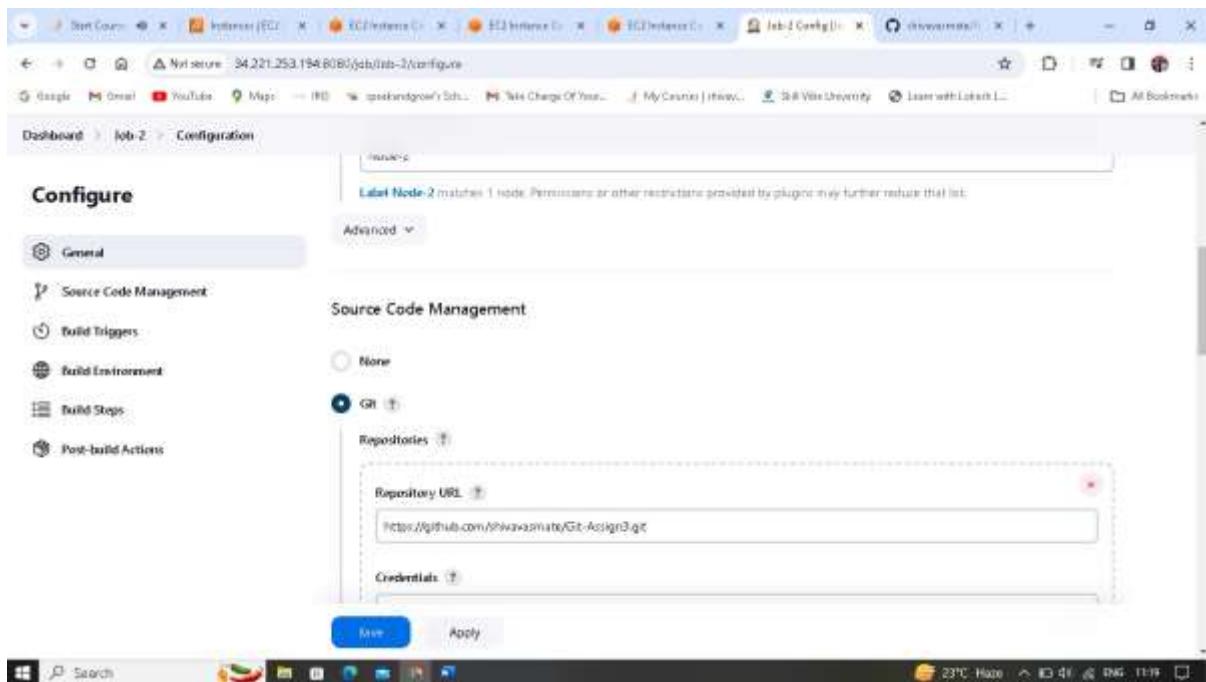


This screenshot shows the same Jenkins configuration page for 'Job-2', but with the 'Label Expression' field filled with 'Node-2'. The 'Restrict where this project can be run' checkbox is checked. The rest of the configuration is identical to the previous screenshot.

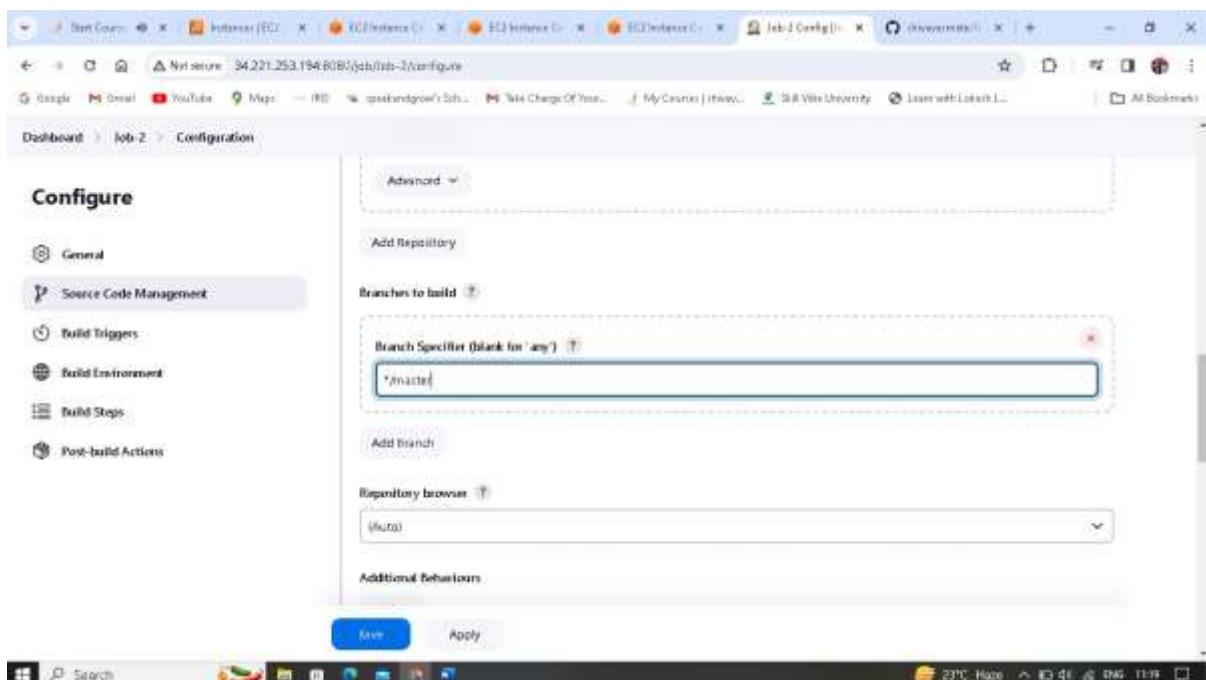
5. copy the URL for Github



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



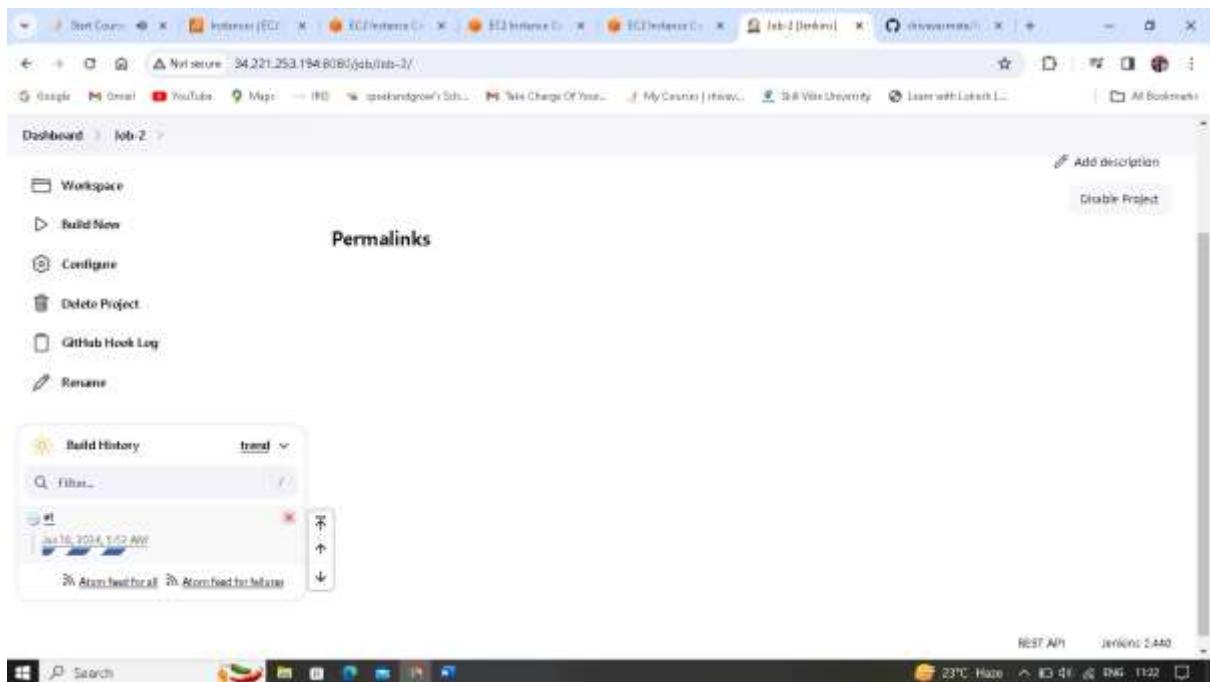
8. In Build Triggers choose GitHub hook trigger for Git Scm polling and click on save

The screenshot shows the Jenkins configuration interface for a job named 'Job-2'. The left sidebar lists several configuration sections: General, Source Code Management, Build Triggers (which is currently selected), Build Environment, Build Steps, and Post-build Actions. The 'Build Triggers' section contains several checkboxes: 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITScm polling' (which is checked), and 'Poll SCM'. The 'Build Environment' section includes options like 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published build scans', and 'Terminate a build if it's stuck'. At the bottom of the configuration page are 'Save' and 'Apply' buttons.

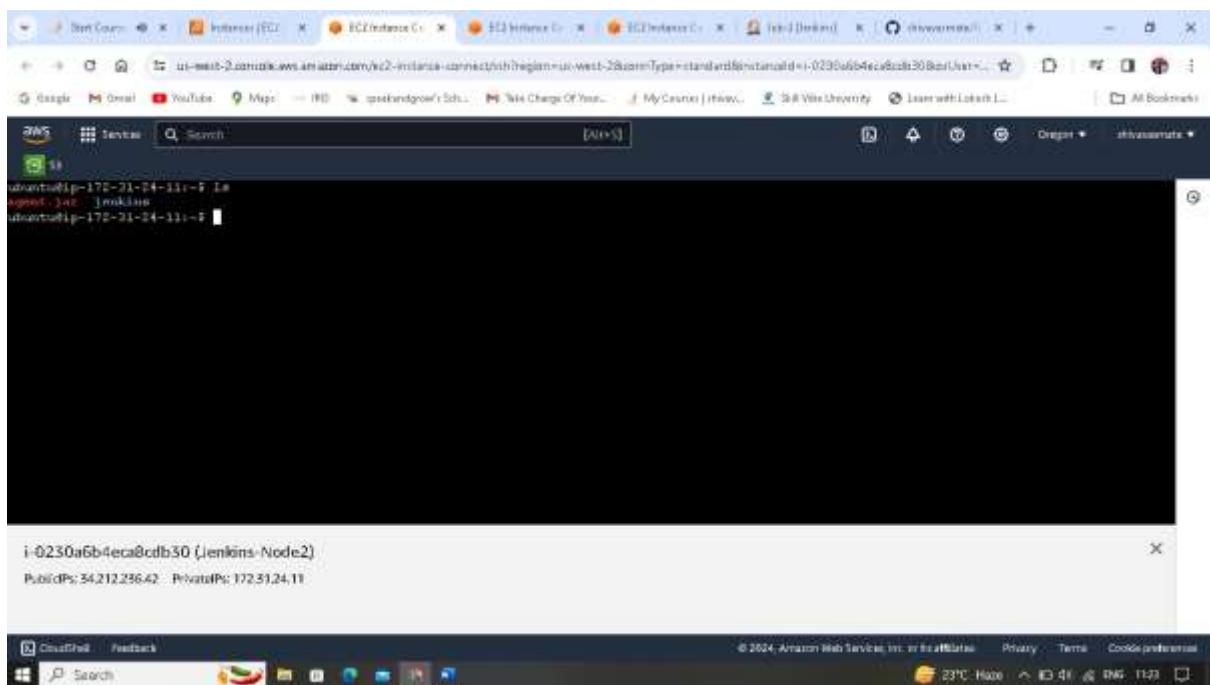
9. Job-2 is created

The screenshot shows the Jenkins dashboard with a single job listed: 'Job-2'. The job card includes links for 'States', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', and 'Rename'. On the right side of the card, there are buttons for 'Add description' and 'Disable Project'. Below the job card is a 'Build History' section which displays 'No builds'. The Jenkins logo and search bar are at the top, and the Windows taskbar is at the bottom.

10. Click on Build Now.



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



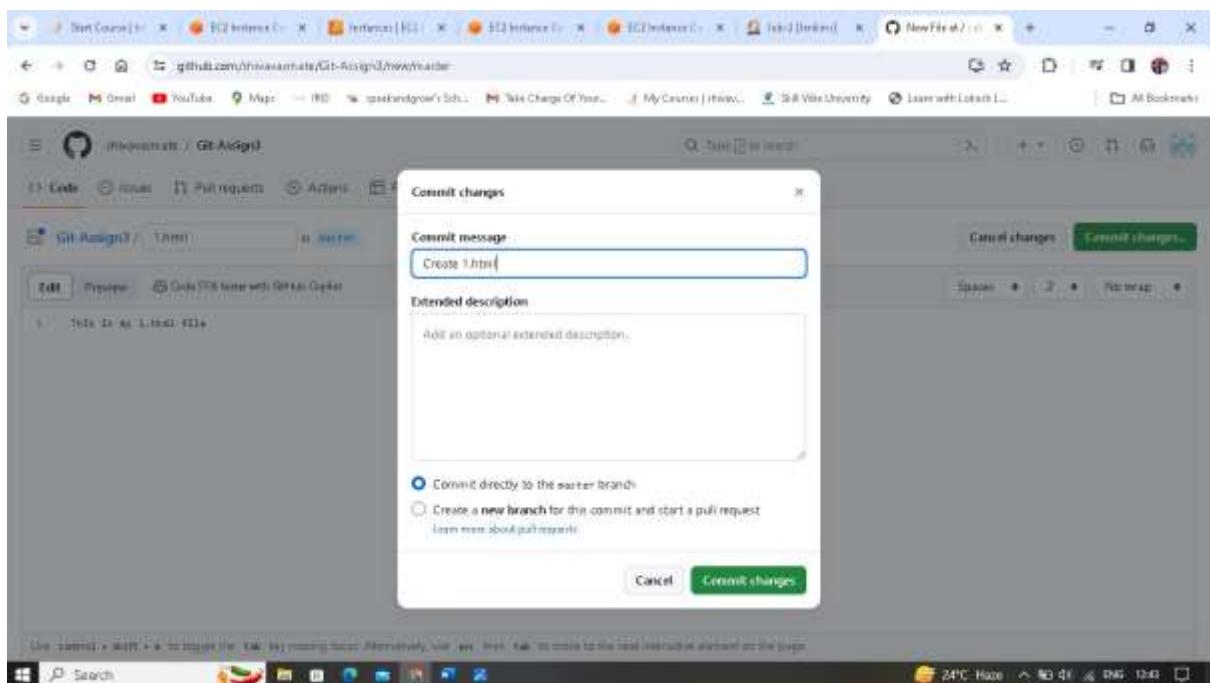
12. Finally Job-2 have main.txt from main branch of github.

A screenshot of a web browser window. The address bar shows a Jenkins instance URL. The main content area displays a terminal session on a Jenkins slave node (Ubuntu 22.04). The logs show the user navigating through Jenkins workspace and creating a new file named '1.html'. A modal dialog box is overlaid on the terminal window, providing Jenkins node details: ID: i-0230a6b4eca0cdb30 (Jenkins-Node2), Public IPs: 34.212.236.42, Private IP: 172.31.24.11. The browser's status bar at the bottom indicates the date and time.

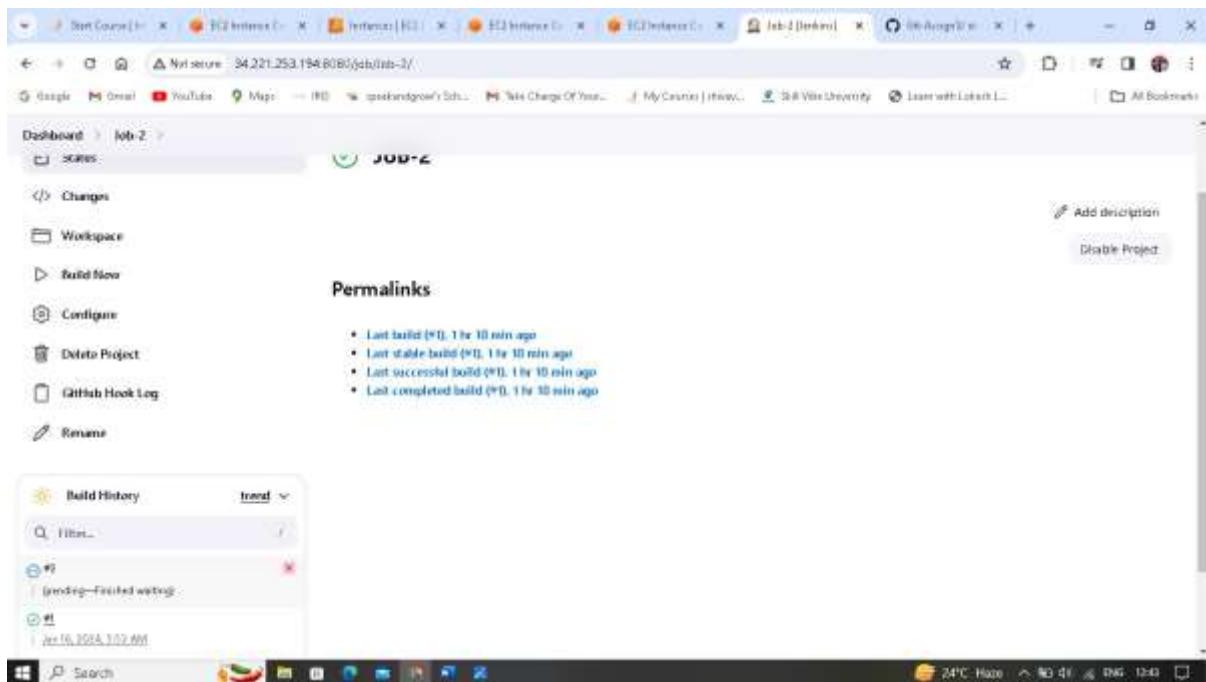
```
ubuntu@ip-172-31-24-11:~$ ls
agent.jar jenkins
ubuntu@ip-172-31-24-11:~$ cd jenkins/
ubuntu@ip-172-31-24-11:~/jenkins$ ls
executing workspace
ubuntu@ip-172-31-24-11:~/jenkins$ cd workspace/
ubuntu@ip-172-31-24-11:~/jenkins/workspace$ ls
Job-2
ubuntu@ip-172-31-24-11:~/jenkins/workspace$ cd Job-2/
ubuntu@ip-172-31-24-11:~/jenkins/workspace/Job-2$ ls
ssim.txt
ubuntu@ip-172-31-24-11:~/jenkins/workspace/Job-2$
```

i-0230a6b4eca0cdb30 (Jenkins-Node2)
PublicIPs: 34.212.236.42 PrivateIPs: 172.31.24.11

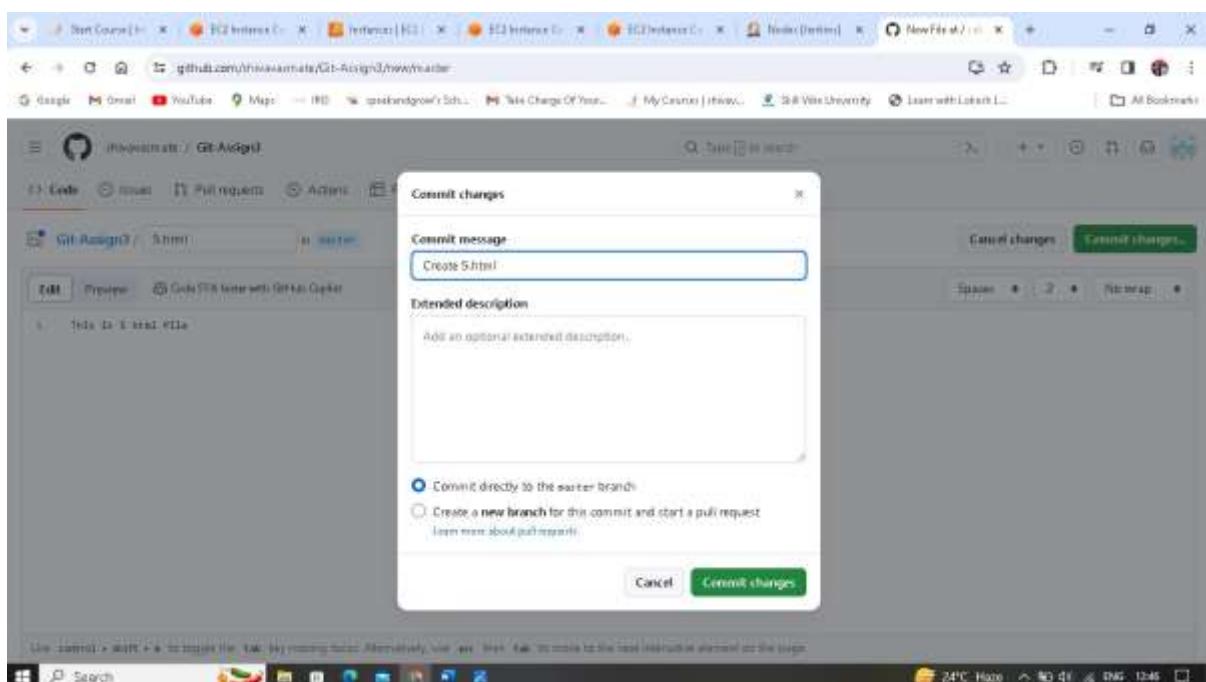
13. Now add a new file 1.html and click on Commit changes.



14.#2 build is created automatically



15. Similarly For 5.html file



16.#3 build also created automatically

The screenshot shows a web browser window with multiple tabs open. The active tab displays the Jenkins interface for a project named 'Job-2'. The page includes a sidebar with options like 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', and 'Rename'. Below this is a 'Permalinks' section with a bulleted list of build links. The main content area is titled 'Build History' and lists three builds: 'Jan 16, 2024, 215 AM', 'Jan 16, 2024, 215 AM' (selected), and 'Jan 16, 2024, 1:03 AM'. At the bottom of the page are 'Atom feed' and 'RSS feed' links. The browser's address bar shows the URL: '34.221.253.194:8080/jenkins/job/Job-2/'. The system tray at the bottom right indicates the date as '24°C - Friday, Jan 19, 2024'.

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

The screenshot shows a terminal window with a black background and white text. It displays Jenkins logs for a connection attempt from an IP address. The logs include messages about JNLP4-connect-proxy, protocol version, and remote identity confirmation. Below the logs, a command-line session is shown where the user navigates to the Jenkins workspace directory ('/jenkins/workspace/Job-2') and lists files. The files listed are '1.html', '2.html', '3.html', '4.html', and '5.html'. The terminal window has a title bar with 'aws' and 'jenkins' and a status bar at the bottom. The system tray at the bottom right shows the date as '24°C - Friday, Jan 19, 2024'.

Job3 → Job1-Job2 (combine Jobs)

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

The screenshot shows the Jenkins dashboard with two items listed:

S	W	Name	Last Success	Last Failure	Last Duration
✓	🟡	Job-1	42 sec #3	N/A	3.3 sec
✓	🟡	Job-2	1 min 37 sec #3	N/A	0.7 sec

Below the table, there are four links: 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just failed builds'. The 'Atom feed for all' link is underlined.

2. Now click on configure

The screenshot shows the Jenkins interface for a job named 'Job-1'. The dashboard includes a sidebar with options like 'States', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', and 'Rename'. The main area displays a green checkmark icon next to 'Job-1'. Below it, there's a section titled 'Permalinks' with a list of build links. A 'Build History' table is shown with one entry labeled 'tried'. On the right side, there are buttons for 'Add description' and 'Disable Project'. The top navigation bar shows the user 'SHIVAKUMAR' and a 'log out' link.

3. Go at the end

The screenshot shows the 'Configuration' page for 'Job-1'. The left sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'Post-build Actions' tab is selected. The main area contains a 'General' section with a 'Description' field and a 'Post-build Actions' section. Under 'Post-build Actions', there are three checkboxes: 'Discard old builds', 'GitHub project', and 'This project is parameterized'. At the bottom, there are 'Save' and 'Apply' buttons. The top navigation bar shows the user 'SHIVAKUMAR' and a 'log out' link.

4. Under Post-build Actions

The screenshot shows the Jenkins interface for configuring Job-1. The left sidebar lists configuration sections: General, Source Code Management, Build Triggers, Build Environment (which is selected), Build Steps, and Post-build Actions. Under 'Build Environment', there are three checkboxes: 'Inspect build log for published build scans', 'Terminate a build if it's stuck', and 'With Ant'. Below this is a 'Build Steps' section with a 'Add build step' dropdown. The 'Post-build Actions' section also has a 'Add post-build action' dropdown. At the bottom are 'Save' and 'Apply' buttons.

5. Choose Build other project (3rd)

This screenshot shows the same Jenkins configuration page for Job-1, but the 'Post-build Actions' section is now expanded. A dropdown menu is open, listing various actions: Aggregate downstream test results, Archive the artifacts, Build other projects (which is selected), Publish JUnit test result report, Record fingerprints of files to track usage, Git Publisher, E-mail Notification, Editable E-mail Notification, Set GitHub commit status (unstable), Set build status on GitHub commit (deprecated), and Delete workspace when build is done. Below the dropdown is an 'Add post-build action' dropdown and the familiar 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins configuration interface for 'Job-1'. On the left sidebar, under 'Build Triggers', the 'Post-build Actions' section is selected. A sub-section titled 'Build other projects' is expanded, showing a list of projects to build: 'Job', 'Job-1', and 'Job-2'. 'Job-2' is highlighted with a blue selection bar. Below the list are three trigger options: 'Trigger only if build is stable' (selected), 'Trigger even if the build is unstable', and 'Trigger even if the build fails'. At the bottom of the configuration panel are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins dashboard. On the left sidebar, under 'My Views', there is a 'Build Queue (1)' view. It lists one item: 'Job-2'. In the main content area, there is a table titled 'Build History' showing two builds for 'Job-2'. The first build (#5) was successful with a duration of 3.3 sec. The second build (#3) was successful with a duration of 0.7 sec. The table includes columns for Status (S), Warning (W), Name, Last Success, Last Failure, and Last Duration.

S	W	Name	Last Success	Last Failure	Last Duration
Green	Yellow	Job-2	5 min 19 sec #5	N/A	3.3 sec
Green	Yellow	Job-2	6 min 44 sec #3	N/A	0.7 sec

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

Start Course (In) ECLintence Center ECLintence Center ECLintence Center ECLintence Center Dashboard [Dev] -

Netshare 34.221.253.194:8080

Google Gmail YouTube Maps ... (10) % upanddowngroup's School Site Charge Of Your... My Courses | iHive... 3.8 Wits University Learn with Lakshmi L... All Bookmarks

Jenkins

SHIVAKUMAR log out

Dashboard >

+ New Item

All +

People

Build History Project Relationship Check File Fingerprint Manage Jenkins

Add description

Build Queue

No builds in the queue

Build Executor Status

Icon: S M L Icon legend: ⚡ Alarm feed for all ⚡ Alarm feed for failures ⚡ Alarm feed for just latest builds

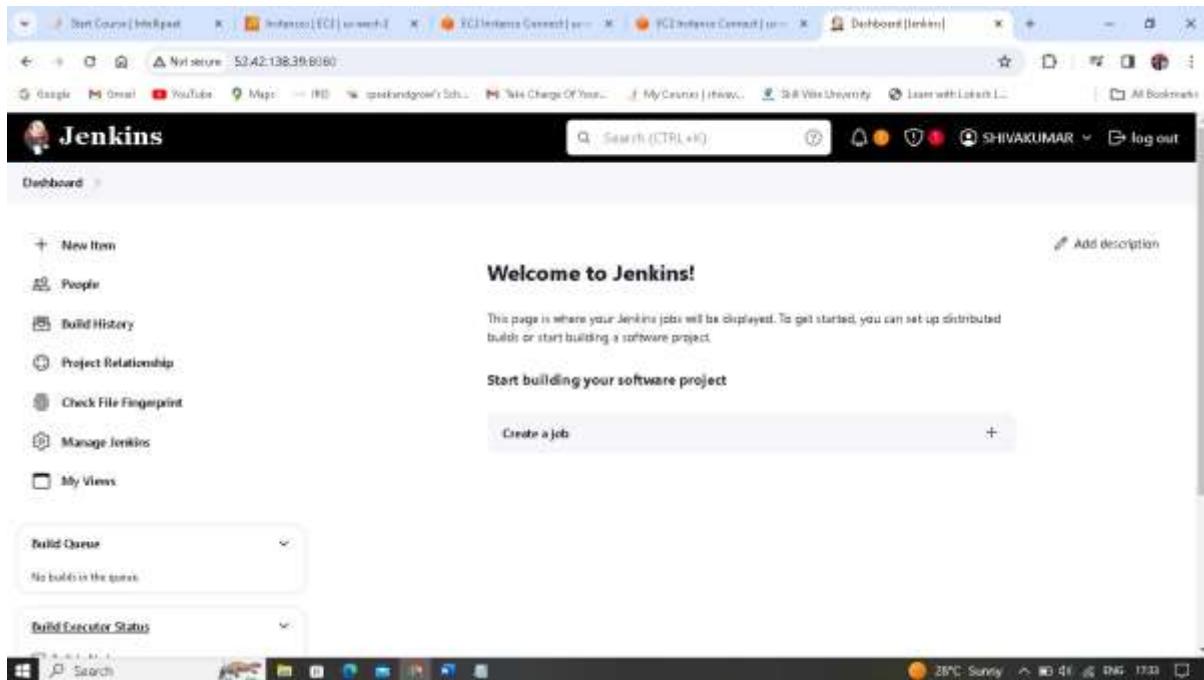
S	W	Name	Last Success	Last Failure	Last Duration
✓	⚠	Job-1	87 sec ⚡	N/A	0.48 sec ⚡
✓	⚠	Job-2	13 sec ⚡	N/A	0.57 sec ⚡

Search

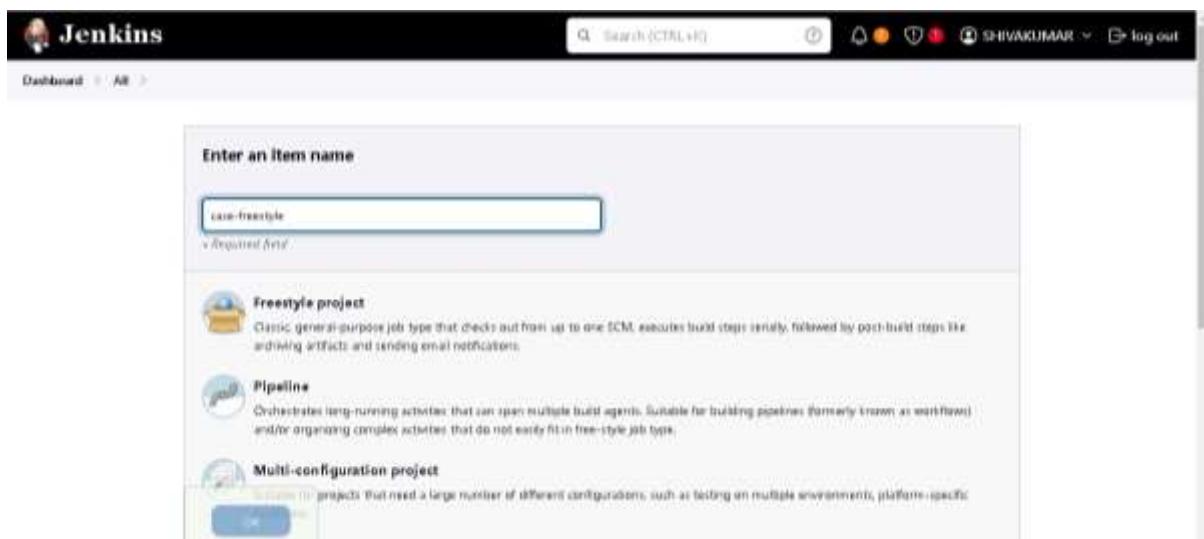
25°C Haze 80 46 88 12:59

I. FOR FREESTYLE

1. Go to Dashboard click on New Item.



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



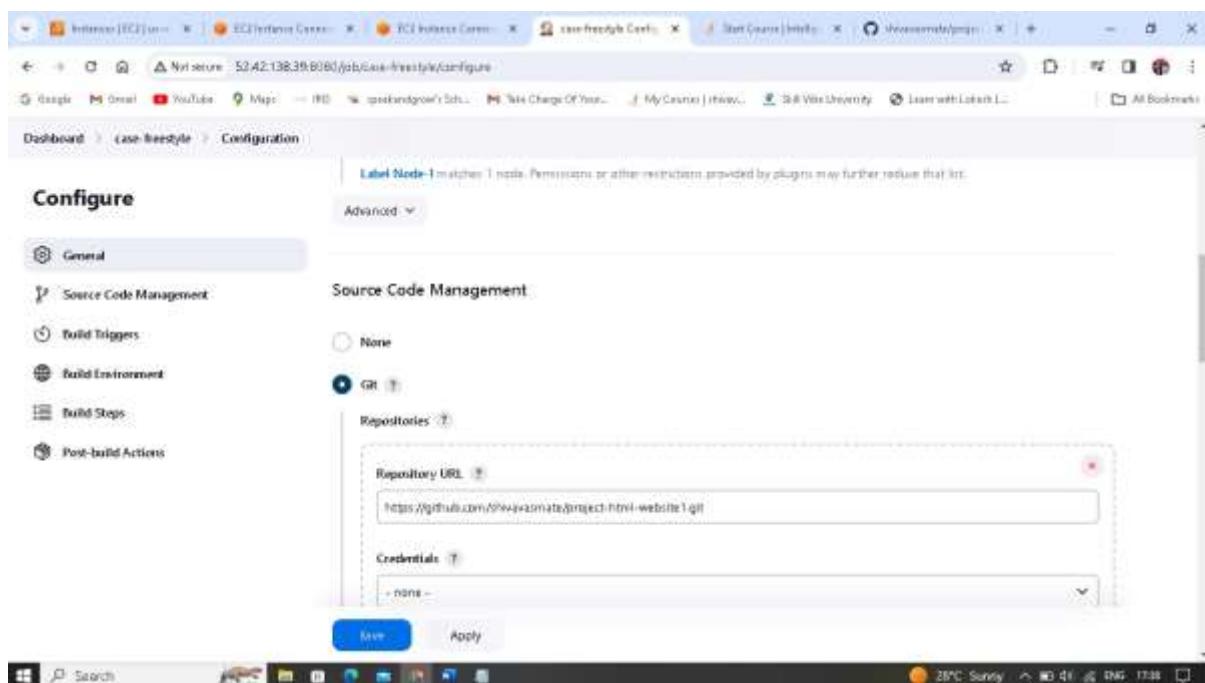
3. Fill it

The screenshot shows the Jenkins configuration interface for a project named 'case-freestyle'. The 'General' tab is selected. The 'Enabled' checkbox is checked. In the 'Description' field, there is no text. Under 'Post-build Actions', three checkboxes are present: 'Discard old builds', 'GitHub project', and 'This project is parameterized'. At the bottom are 'Save' and 'Apply' buttons.

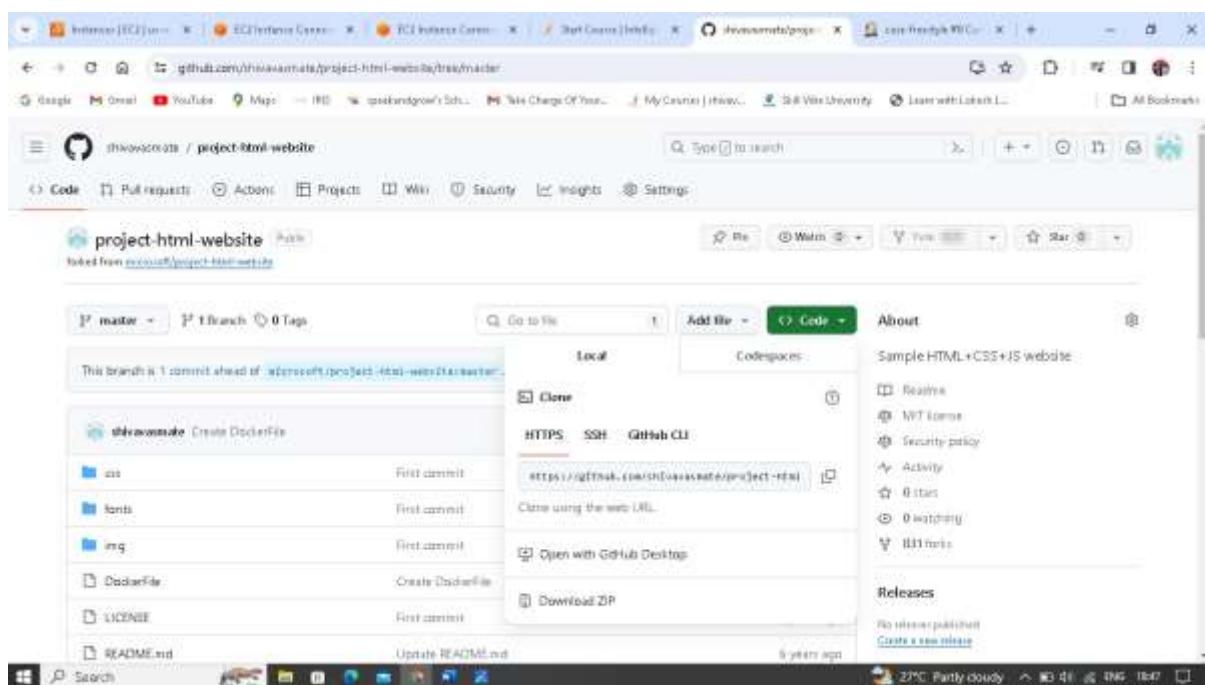
4. First of all check in Restrict where this project can be run. Label Expression as Node-1

The screenshot shows the Jenkins configuration interface for the same project. The 'Restrict where this project can be run' checkbox is checked. Below it, the 'Label Expression' field contains 'Node-1'. A note below the field states: 'Label Node-1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' At the bottom are 'Save' and 'Apply' buttons.

5. Under Source Code Management Check Git



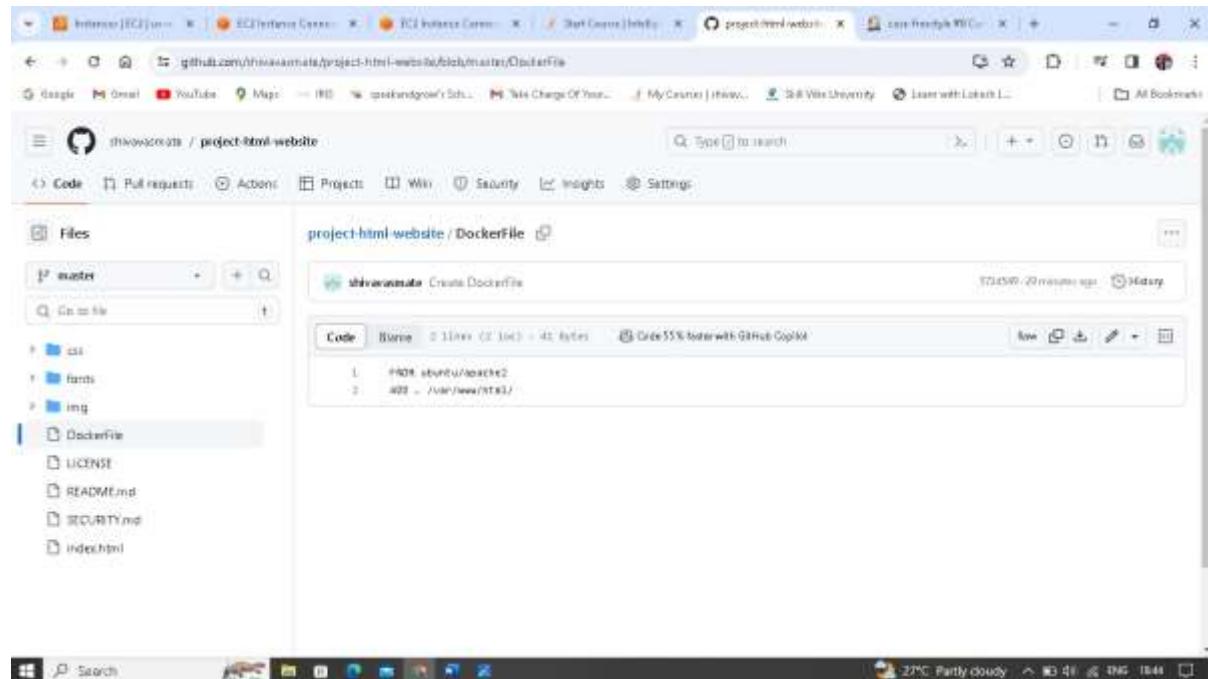
6. Now log in to your GitHub in another browser



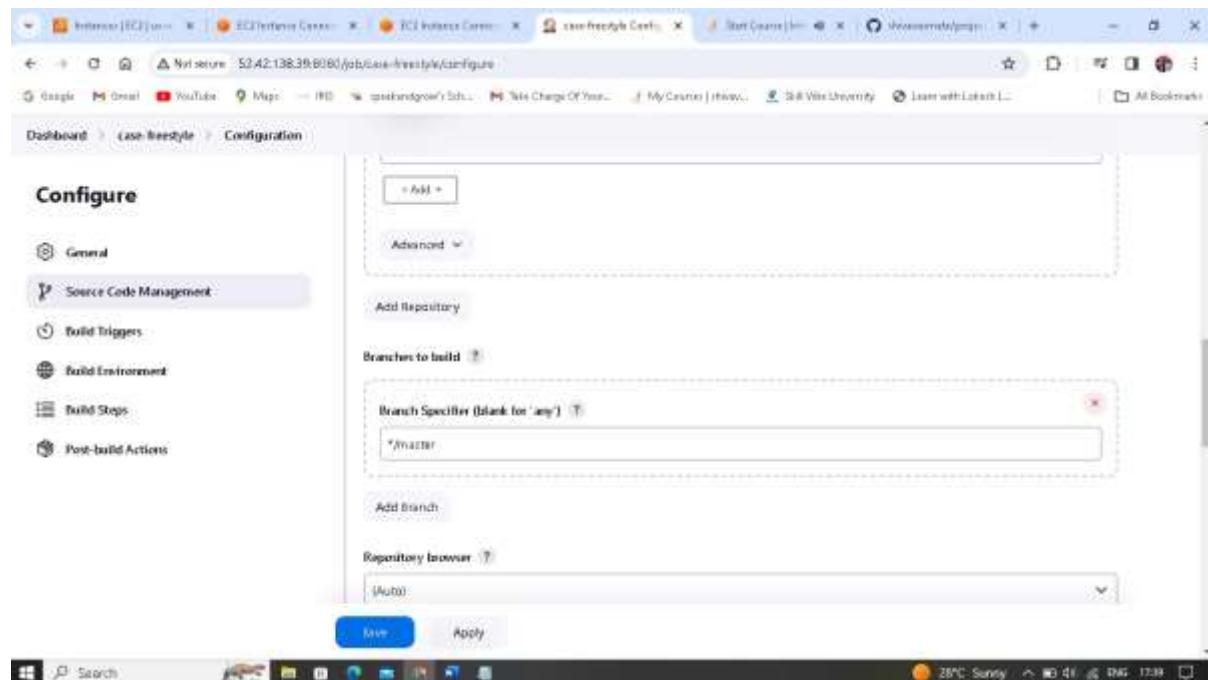
7. Create DockerFile with command

```
FROM ubuntu/apache2
```

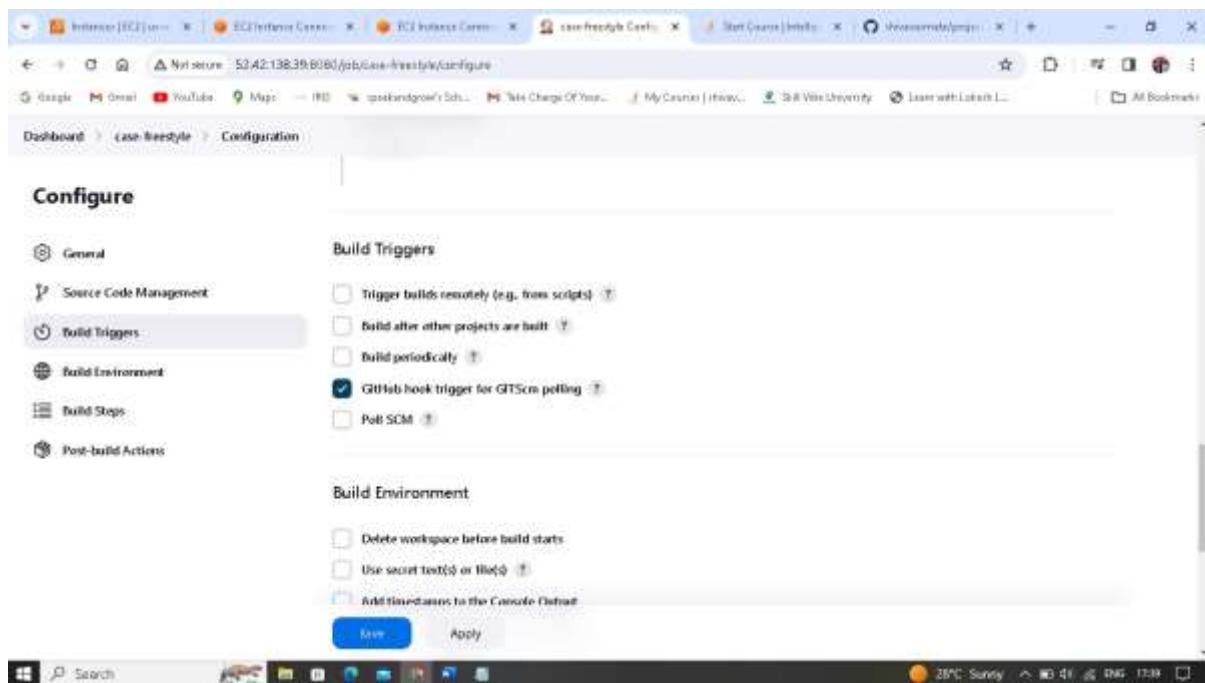
ADD . /var/www/html/ command changes.



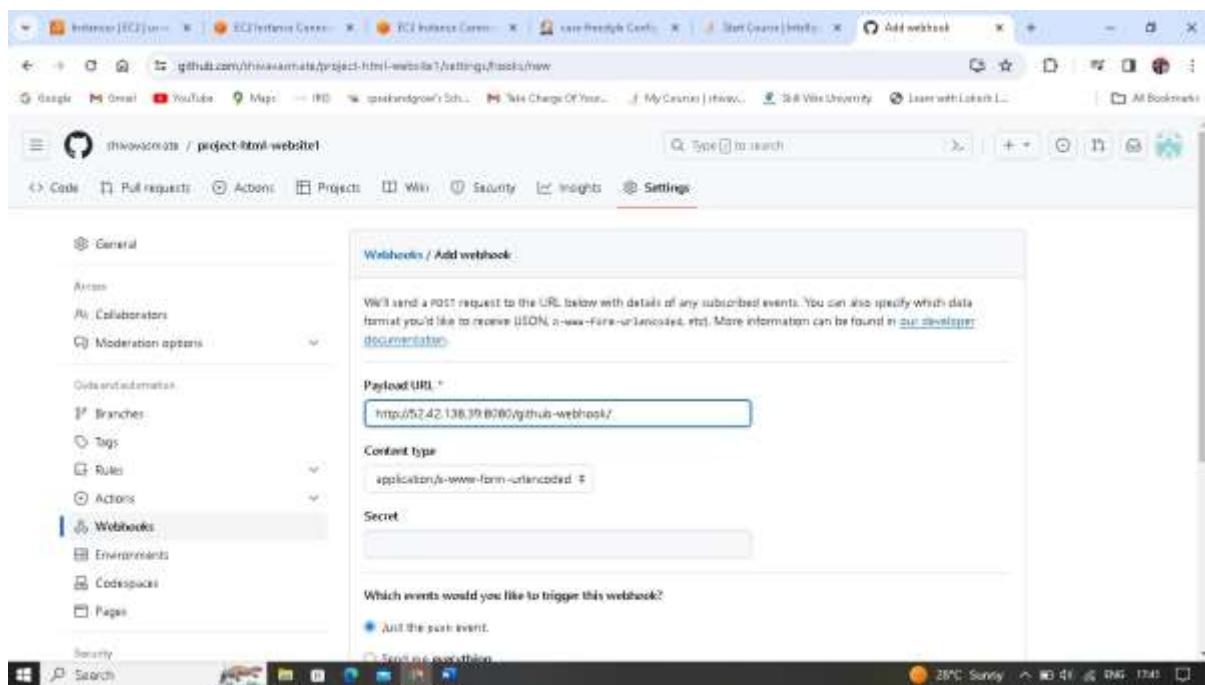
8. In Branches to build and Branch Specifier give */master



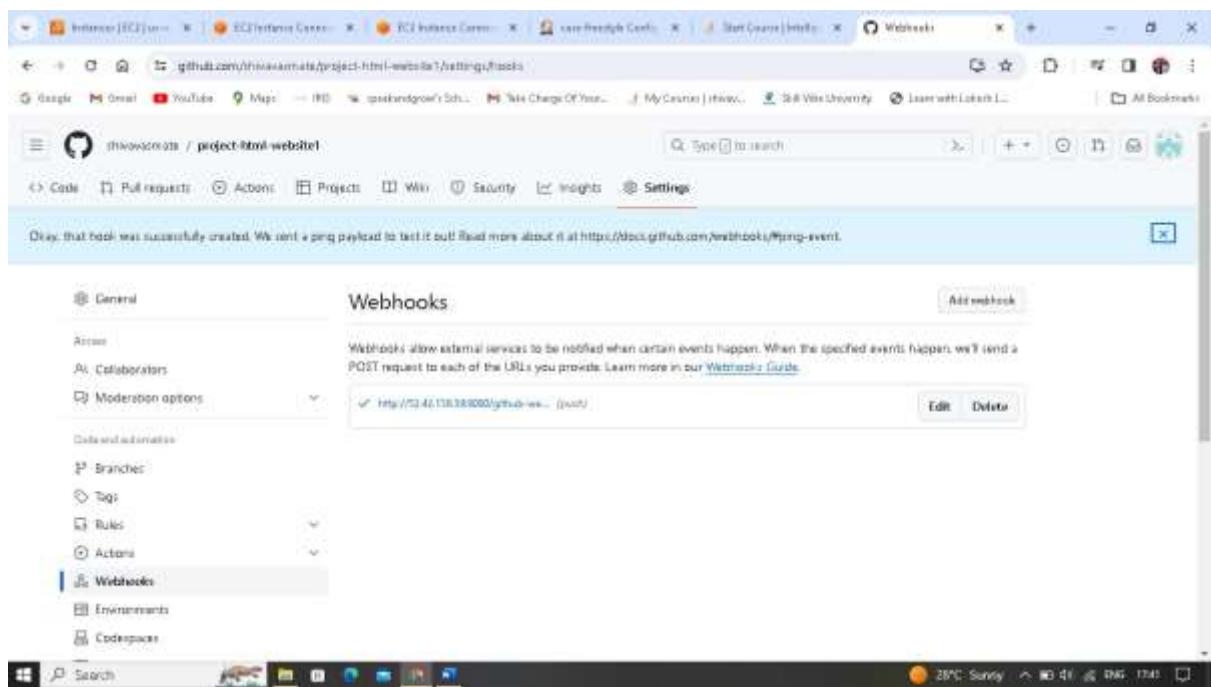
9. Under Build Triggers choose GitHub hook trigger for GIT Scm polling.



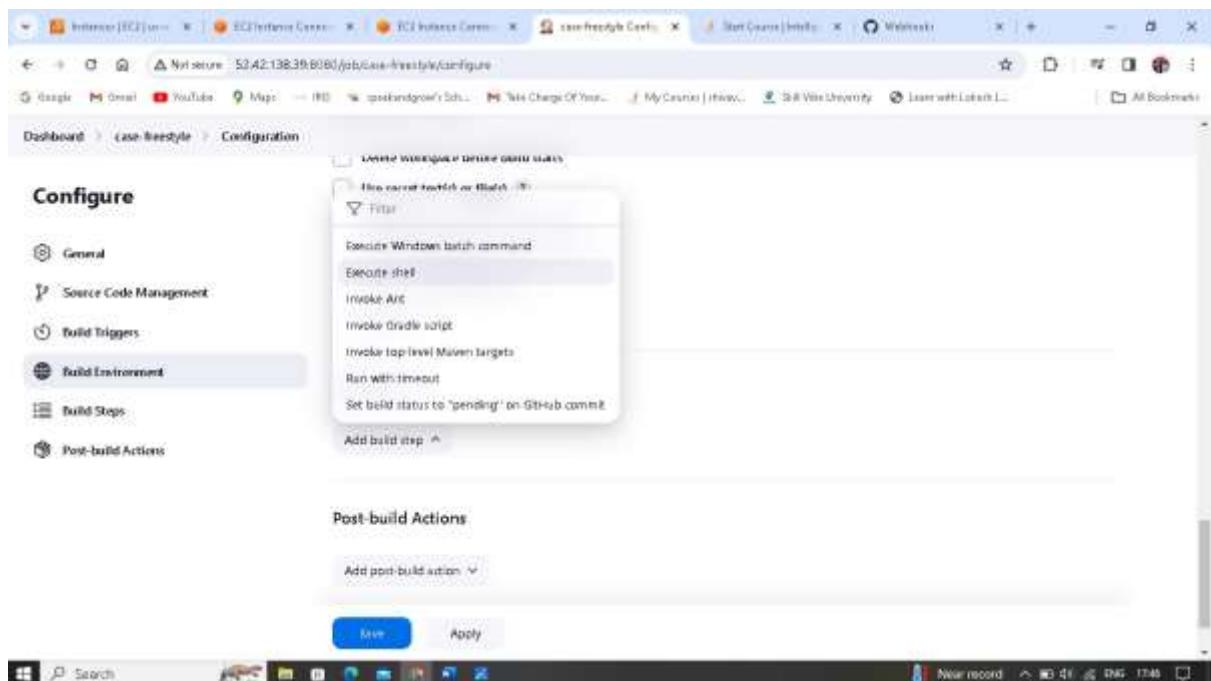
10. Now go to GitHub under repo settings look for Webhooks on left side



11. Check for right mark



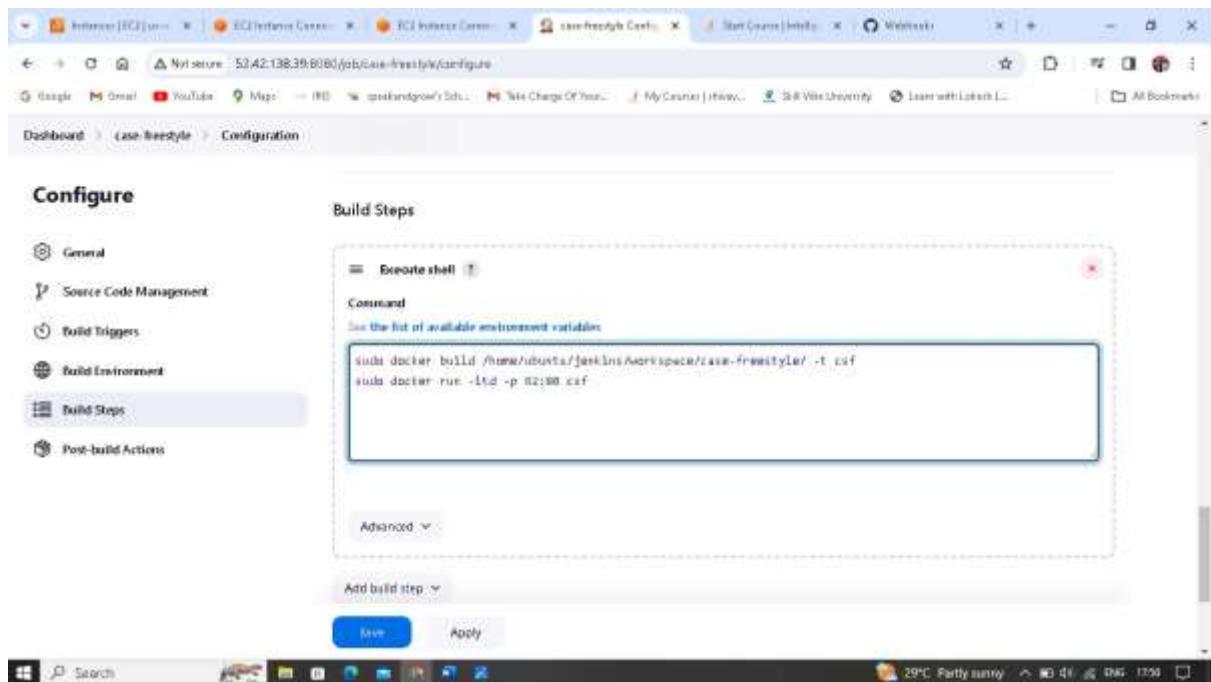
12. Under Build Steps click on Execute shell



13. Under Command type

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

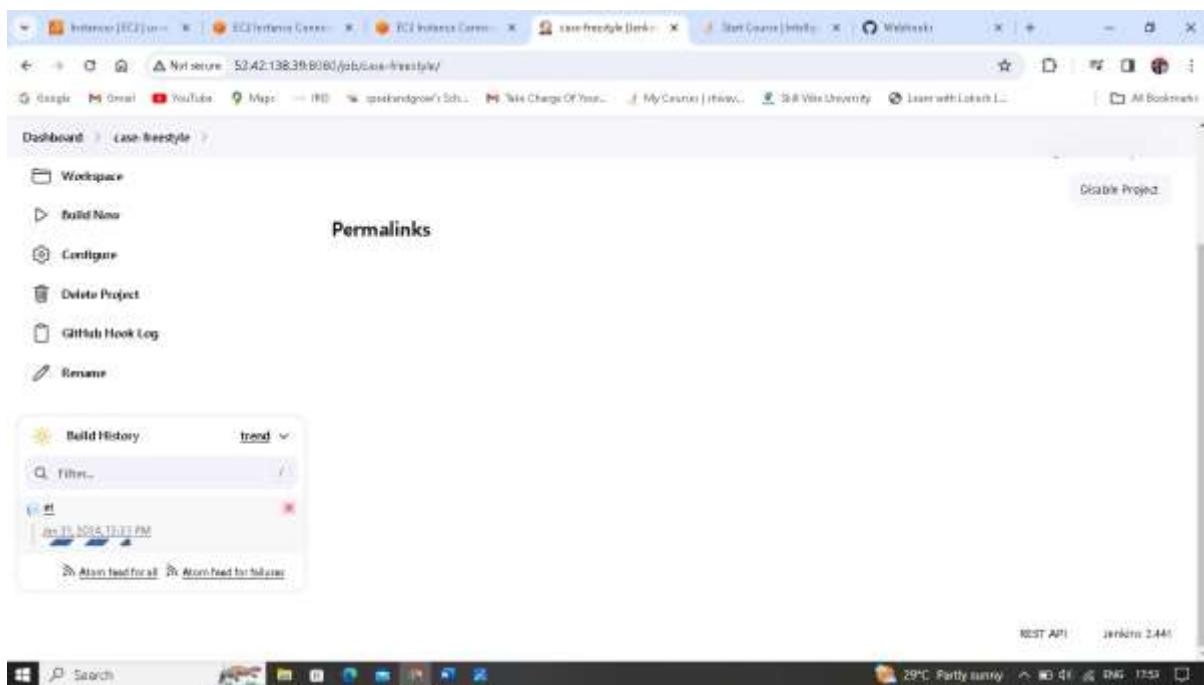
sudo docker run -itd -p 82:80 csf and click on save.



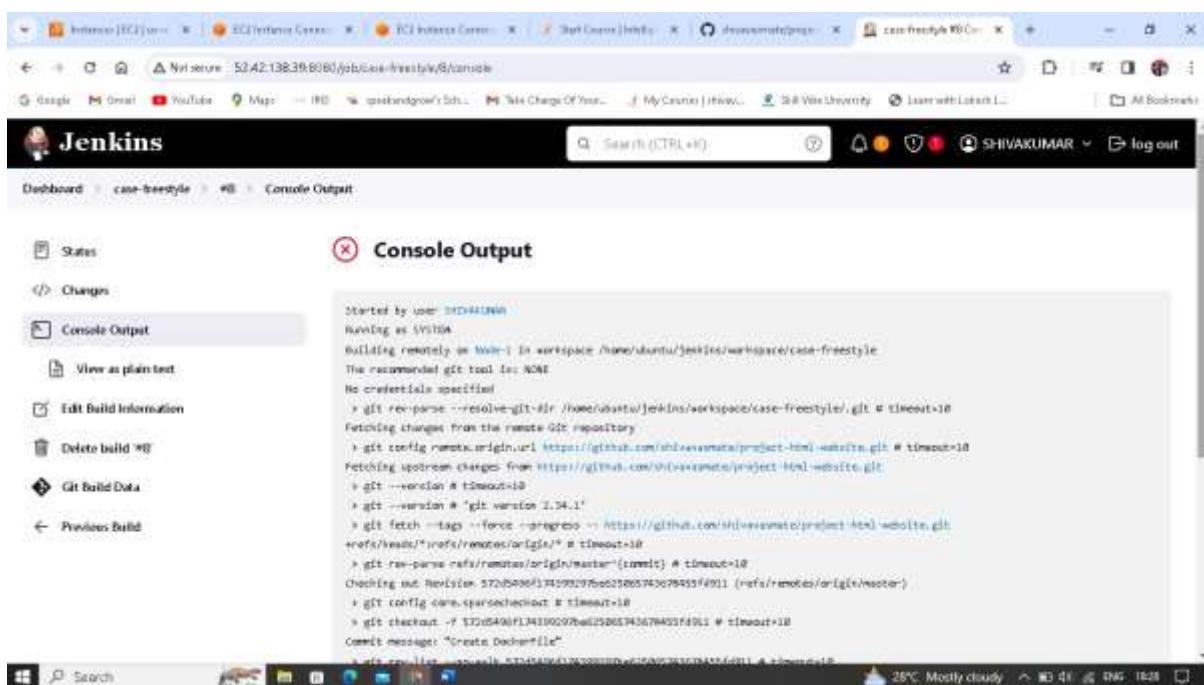
14.The case-freestyle is created



15. Click on Build Now



16. Check for error



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

```
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (Qemu) binaries on this host.  
ubuntu@ip-172-31-30-100:~$ ps  
/root/ubuntu  
ubuntu@ip-172-31-30-100:~$ ls  
jenkins  
ubuntu@ip-172-31-30-100:~$ ps  
/root/ubuntu  
ubuntu@ip-172-31-30-100:~$ ls  
jenkins  
ubuntu@ip-172-31-30-100:~$ cd jenkins/  
ubuntu@ip-172-31-30-100:~/jenkins$ ls  
creating vhostconfig.jenkins  
ubuntu@ip-172-31-30-100:~/jenkins$ cd  
ubuntu@ip-172-31-30-100:~$ sudo apt install docker.io -y  
i-05e9cdddf5006b398a [Jenkins-Node1]  
PublicIP: 35.89.148.67 PrivateIP: 172.31.30.100
```

18. Build now again and check for error

The screenshot shows the Jenkins interface with the following details:

- Job Name:** case-freestyle
- Build Status:** Success
- Console Output:** Available (highlighted in blue)
- Output Content:**

```
Started by user SHIVAKUMAR  
Running on Node-1 in workspace /home/ubuntu/jenkins/workspace/case-freestyle  
The recommended git tool is: NONE  
No credentials specified  
+ git raw-pulls --remote=git-dir /home/ubuntu/jenkins/workspace/case-freestyle/.git # timeout=10  
Fetching changes from the remote git repository  
+ git config remote.origin.url https://github.com/shivakumar/project-html-website.git # timeout=10  
Fetching upstream changes from https://github.com/shivakumar/project-html-website.git  
+ git --version # timeout=10  
+ git --version # git version 2.34.1  
+ git fetch -t --force --progress -- https://github.com/shivakumar/project-html-website.git  
refs/heads/* refs/remotes/origin/* # timeout=10  
+ git raw-pulls ref/remotes/origin/master:[remote] # timeout=10  
Checking out Revision 372d549ef134299278be2580874562764559f011 (refs/remotes/origin/master)  
+ git config core.sparsecheckout # timeout=10  
+ git checkout -f 372d549ef134299278be2580874562764559f011 # timeout=10  
Commit message: "Create Dockerfile"
```

The screenshot shows a Jenkins build log window. At the top left are 'Git Build Data' and 'Previous Build' buttons. The main area contains a large amount of command-line output. Key lines include:

```
git --version # timeout=10
git --version # 'git version 2.34.1'
git fetch --tags --force --progress -- https://github.com/oliviercamon/project-HDL-wm011a.git
refs/heads/* refs/remotes/origin/* # timeout=10
git rev-parse refs/remotes/origin/master^{revparse}: # timeout=10
Checking out Revision 572d946f174399070bae0250027433278450fd911 (#refs/remotes/origin/master)
git config core.sparsecheckout # timeout=10
git checkout -f 572d946f174399070bae0250027433278450fd911 # timeout=10
Commit message: "Create Dockerfile"
git rev-list --no-walk 572d946f174399070bae0250027433278450fd911 # timeout=10
[case-freestyle] $ /bin/sh -c '/tmp/jenkins118957514959123827618.sh'
+ sudo docker build /home/ubuntu/jenkins/workspace/case-freestyle/ -t rcf
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
        Detail the build step to build images with buildkit!
        https://www.docker.com/getbuildkit

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.
```

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

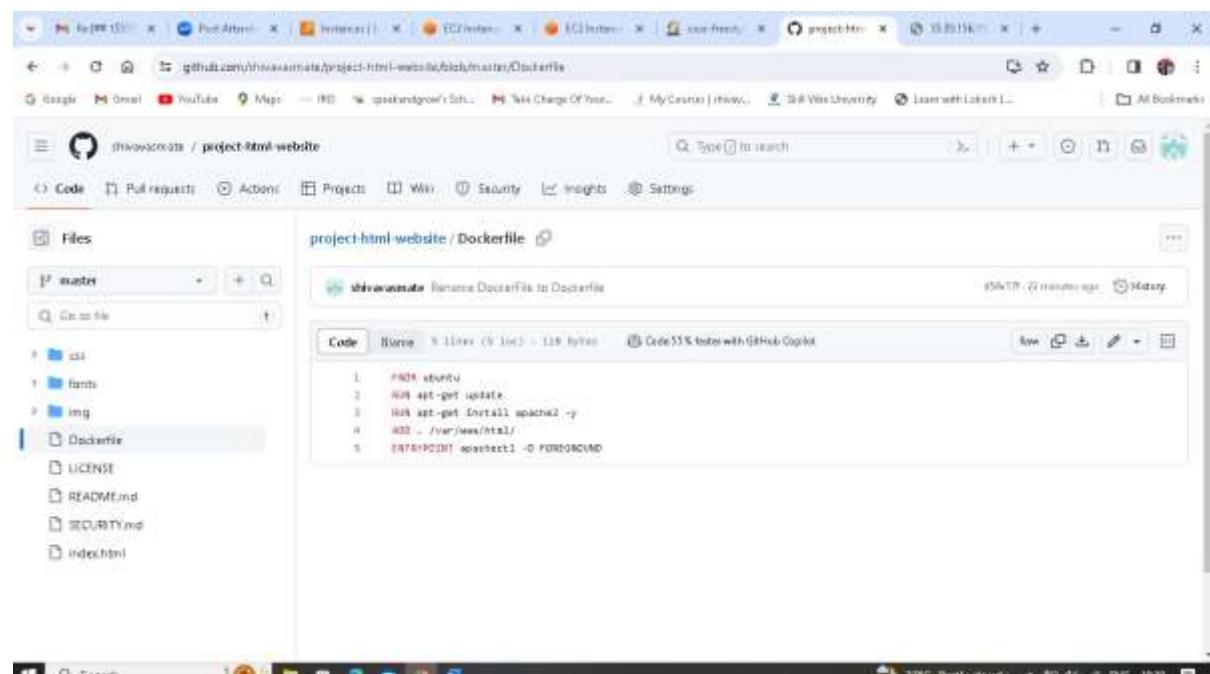
b. Update Dockerfile as
FROM ubuntu

RUN apt-get update

RUN apt-get install apache2 -y

ADD ./var/www/html/

ENTRYPOINT apachectl -D FOREGROUND



The screenshot shows a GitHub repository named 'project-html-website'. The 'Dockerfile' tab is selected, displaying the following code:

```
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.
```

The screenshot shows the Jenkins configuration interface for a 'case-freestyle' job. On the left, a sidebar lists 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps' (which is selected), and 'Post-build Actions'. The main area is titled 'Configure' and contains a 'Build Steps' section. Under 'Build Steps', there is one step named 'Execute shell'. The 'Command' field contains the following Jenkinsfile code:

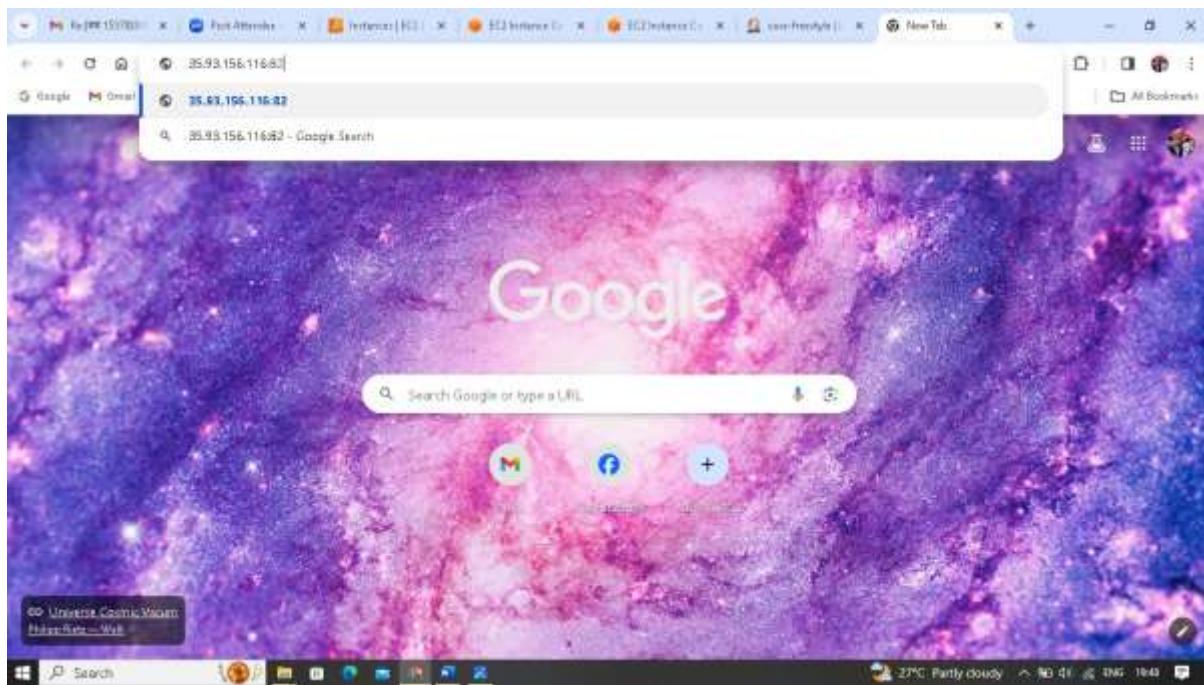
```
sudo docker rm -f $(sudo docker ps -q)
sudo docker build /home/ubuntu/jenkins/workspace/case-freestyle/ -t cxf
sudo docker run -td -p 92:80 cxf
```

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

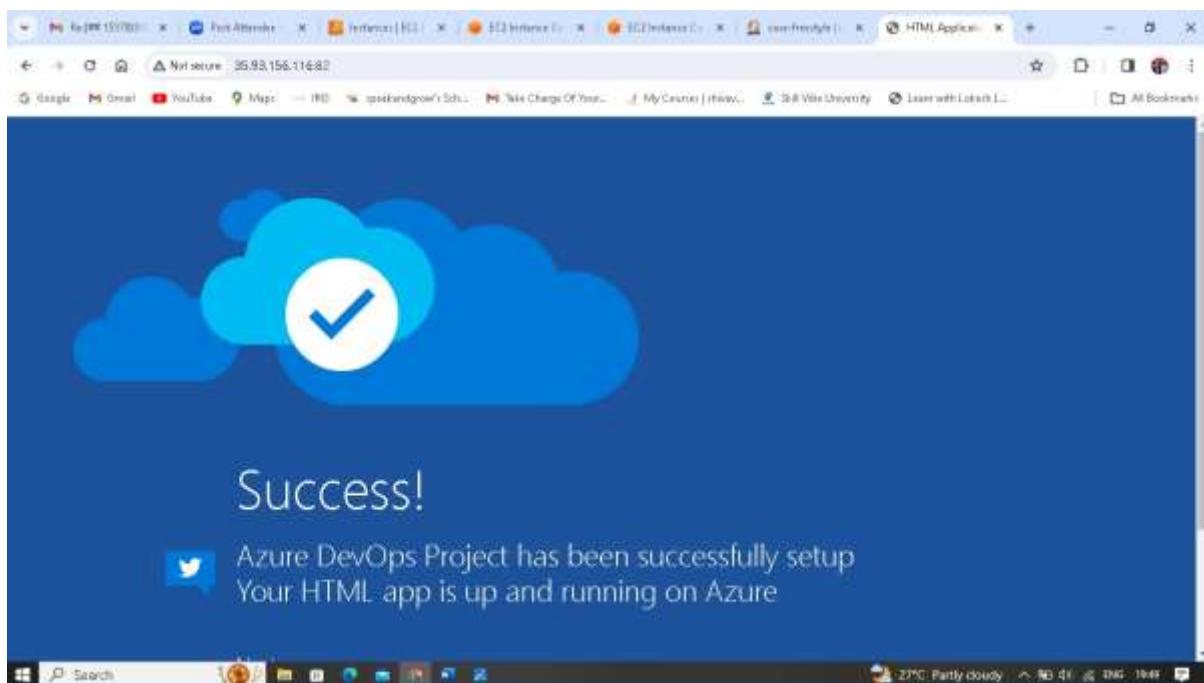
The screenshot shows the Jenkins dashboard for the 'case-freestyle' project. The top navigation bar includes links for 'Dashboard', 'case-freestyle', 'Build History', 'Build Now', 'Configure', 'Delete Project', 'GitHub Hook Log', and 'Rename'. The main content area displays the project name 'case-freestyle' with a green checkmark icon. Below the project name is a 'Permalinks' section listing various build logs. At the bottom of the page is a 'Build History' table with columns for 'Build #', 'Status', 'Duration', and 'Last Result'. The first row in the history table is highlighted with a blue background.

Build #	Status	Duration	Last Result
1	Success	16 min ago	Success

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

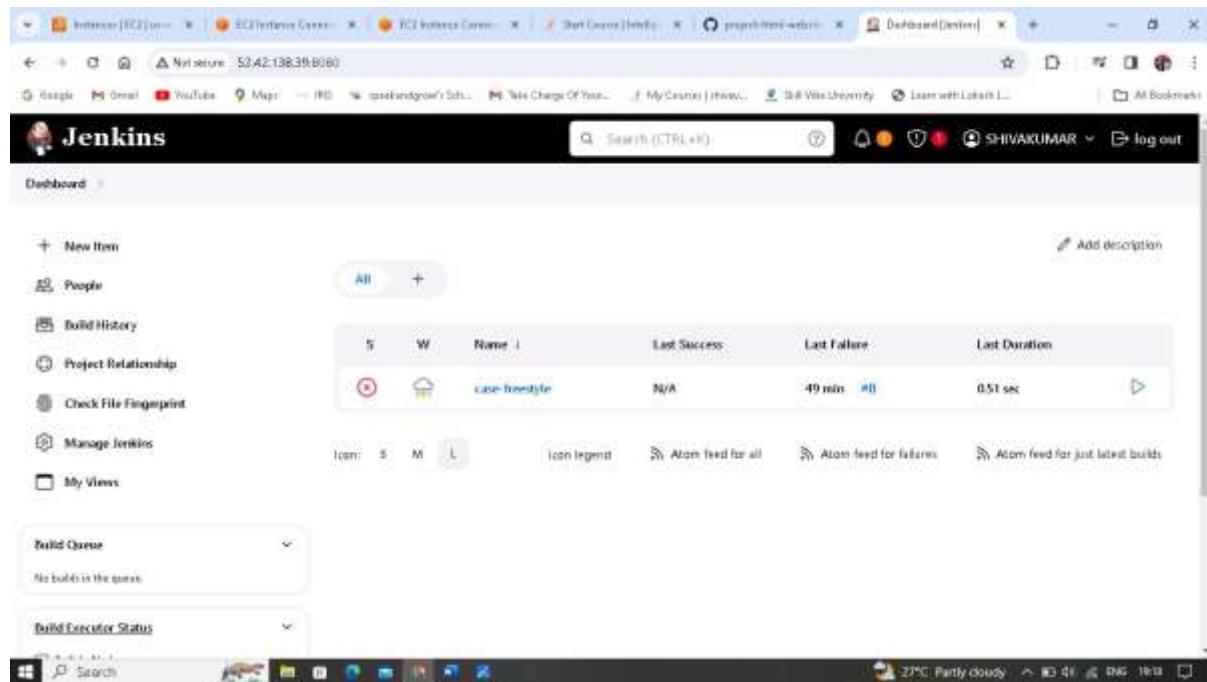


25. Now HTML Application opens.



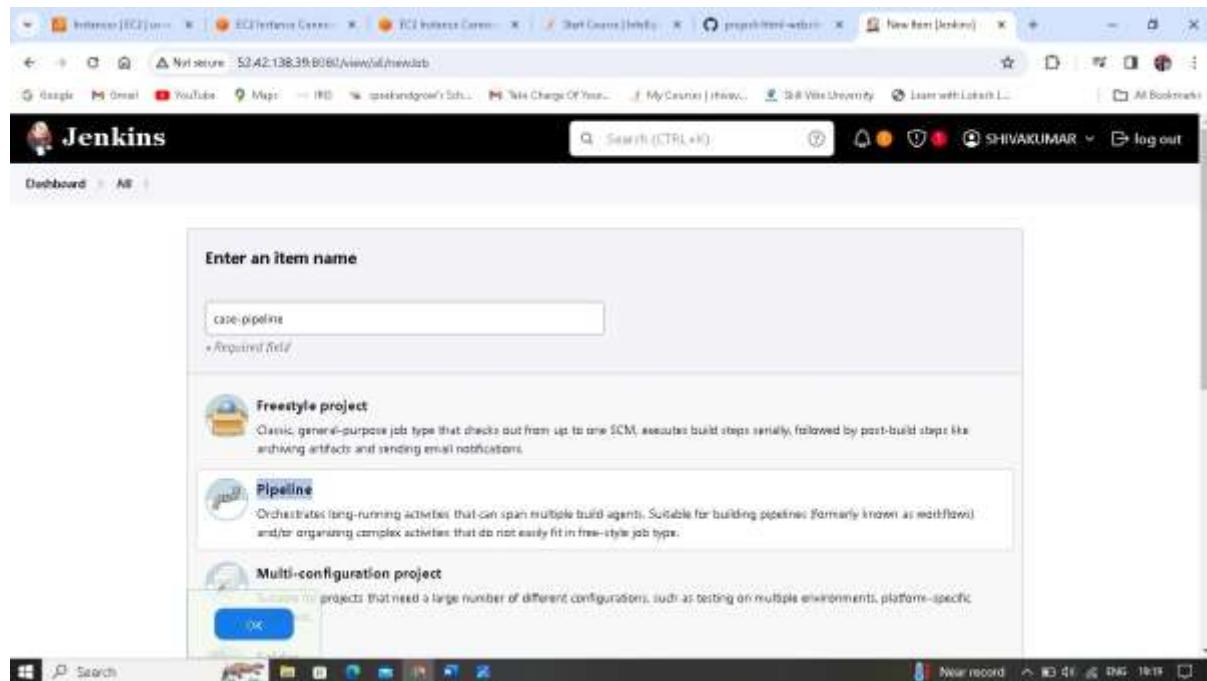
3. FOR PIPELINE

1. Go to Dashboard click on New Item.



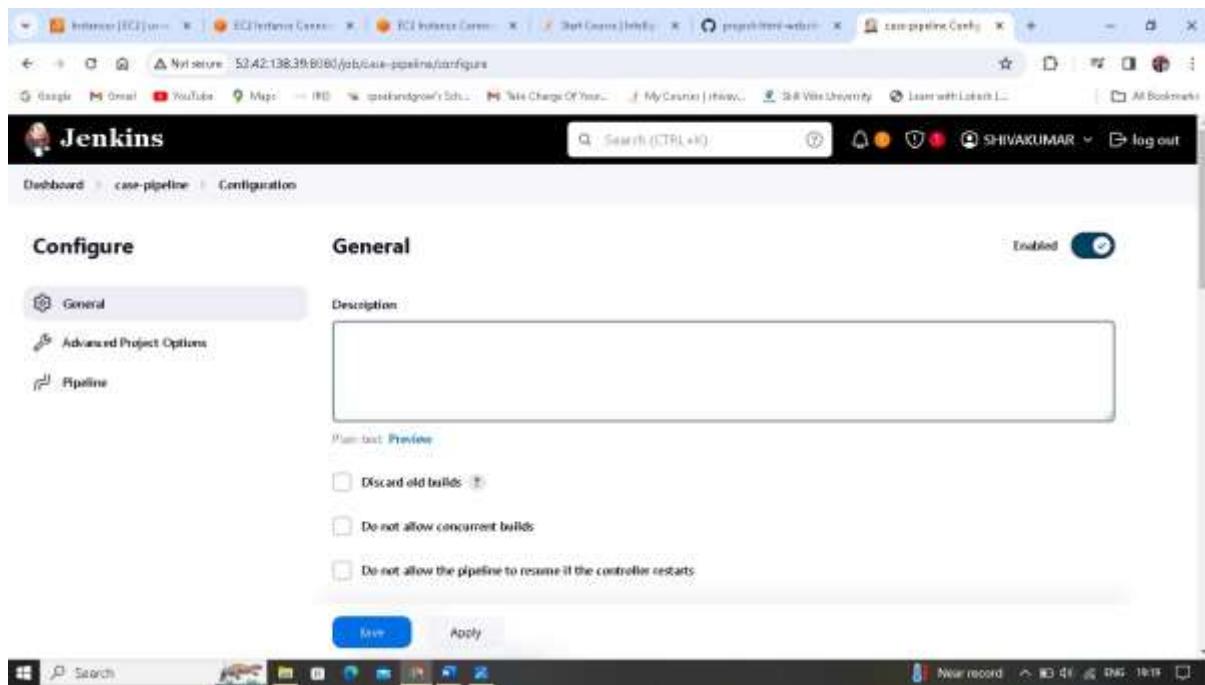
The screenshot shows the Jenkins dashboard with the title bar "Jenkins". On the left, there's a sidebar with links like "+ New Item", "People", "Build History", "Project Relationship", "Check File Fingerprint", "Manage Jenkins", and "My Views". The main area has a table titled "All" showing one item: "case-freestyle" with status "N/A", last failure at "49 min ago", and duration "0.51 sec". Below the table are filters for "Item type" (S, M, L) and "Icon legend". At the bottom, there are three "Atom feed" options: "Atom feed for all", "Atom feed for failures", and "Atom feed for just latest builds". The taskbar at the bottom shows system status: 23% battery, Partly cloudy, 10:46 AM, and a search bar.

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



The screenshot shows a modal dialog box titled "Enter an item name". In the input field, "case-pipeline" is typed. Below the input field, there are three project types listed: "Freestyle project", "Pipeline", and "Multi-configuration project". The "Pipeline" option is highlighted with a blue border. At the bottom of the dialog is a blue "OK" button. The background of the dialog is white, while the rest of the Jenkins interface is greyed out.

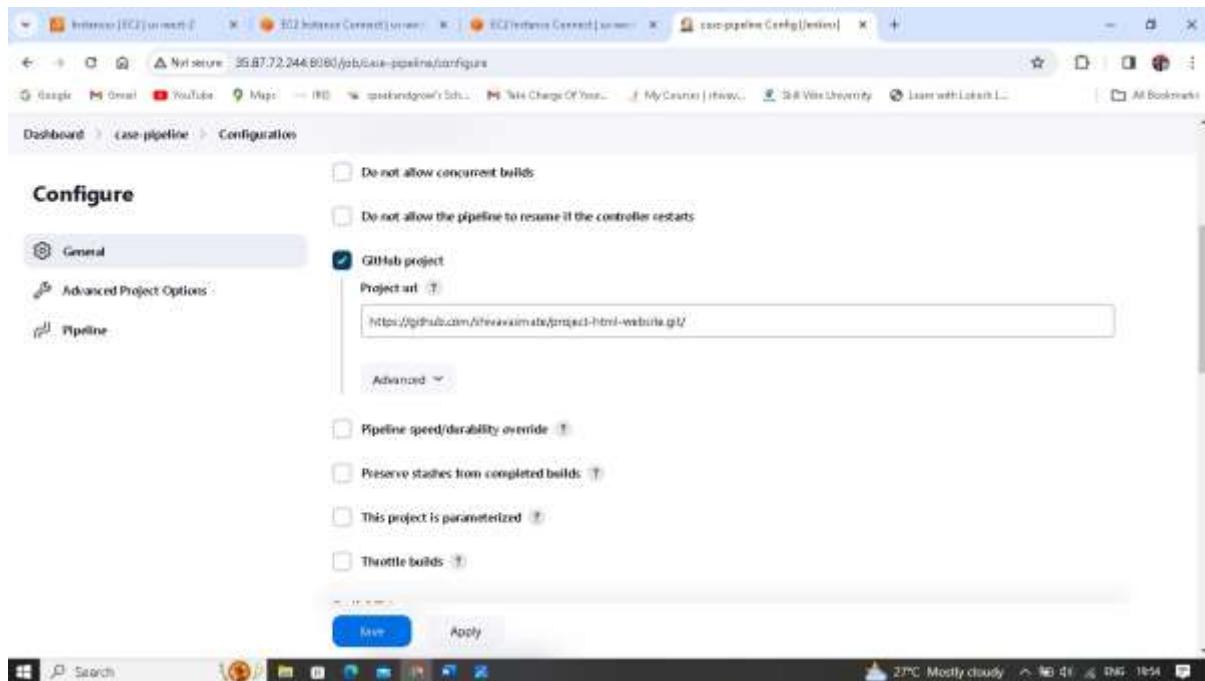
3. Fill it



The screenshot shows the Jenkins Pipeline Configuration page for a project named 'case-pipeline'. The 'General' tab is selected. The 'Enabled' switch is turned on. The 'Description' field is empty. Under 'Plan text', there are three checkboxes: 'Discard old builds', 'Do not allow concurrent builds', and 'Do not allow the pipeline to resume if the controller restarts'. At the bottom are 'Save' and 'Apply' buttons.

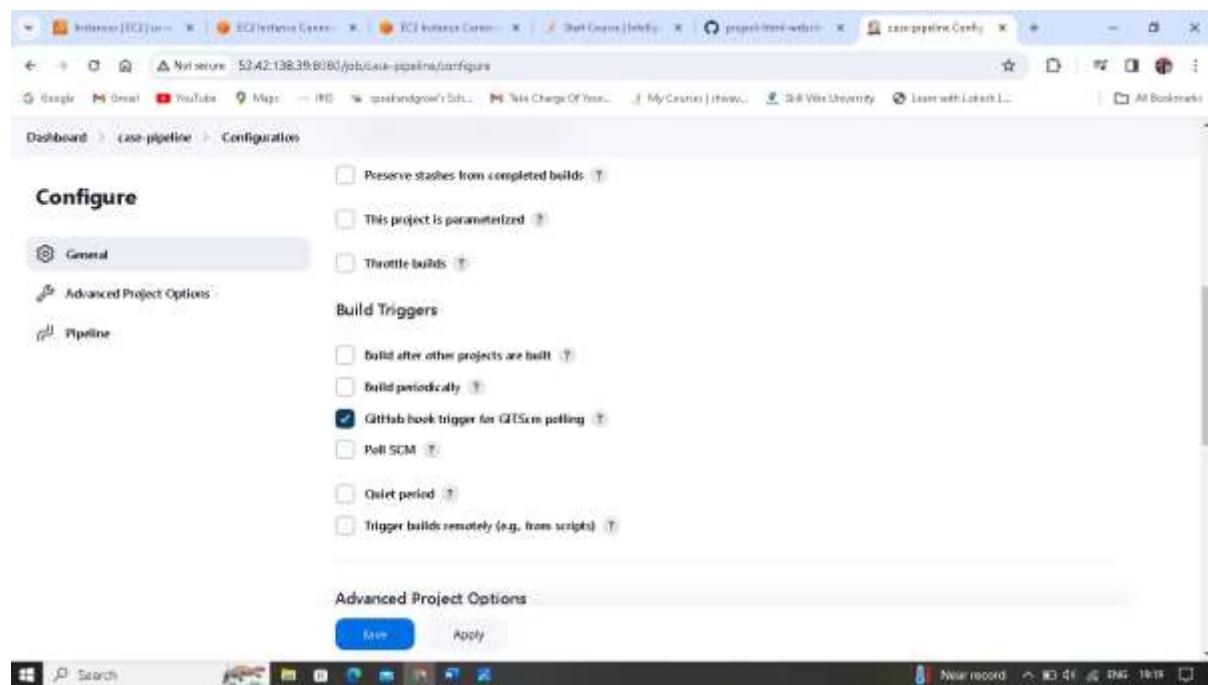
4. Check in GitHub Project

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

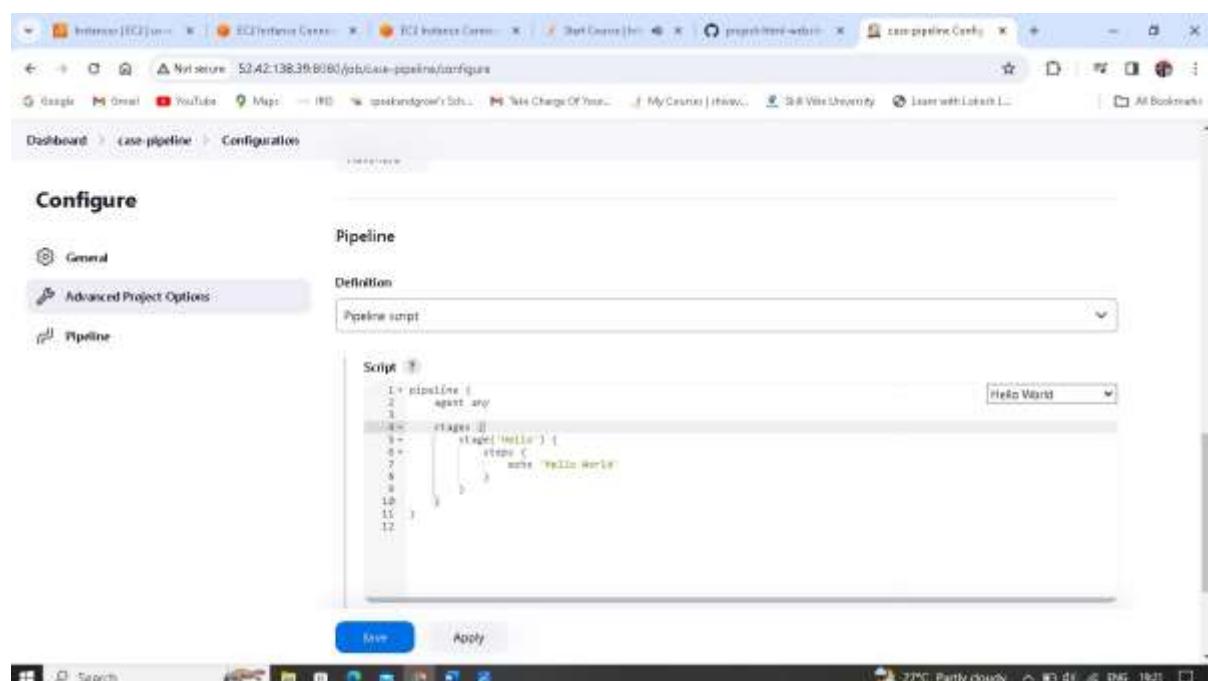


The screenshot shows the Jenkins Pipeline Configuration page for the same project. The 'GitHub project' checkbox is selected. The 'Project url' field contains the value 'https://github.com/shivavasmate/project-html-website.git'. Other checkboxes available but not selected include 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'Pipeline speed/durability override', 'Preserve stashes from completed builds', 'This project is parameterized', and 'Throttle builds'. At the bottom are 'Save' and 'Apply' buttons.

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.

The screenshot shows the Jenkins interface for the 'case-pipeline' project. On the left, there's a sidebar with options like 'States', 'Changes', 'Build Now' (which is highlighted in blue), 'Configure', 'Delete Pipeline', 'Full Stage View', 'Rename', 'Pipeline Syntax', and 'GitHub Hook Log'. The main area is titled 'Stage View' and displays a message: 'No data available. This Pipeline has not yet run.' At the top right, there are buttons for 'Add description' and 'Disable Project'. The bottom left shows a 'Build History' section with a dropdown set to 'trend'. The history table lists four builds: 'Jan 21 1929' (Status: 'Success'), 'Jan 21 1929' (Status: 'CANCELED'), 'Jan 21 1927' (Status: 'CANCELED'), and 'Jan 21 1927' (Status: 'CANCELED'). The bottom right shows system status: 22°C, Partly cloudy, 10:41 AM, 1927.

9. Build is successful.

This screenshot shows the same Jenkins project page after a build was triggered. The 'Build History' table now includes a successful build from 'Jan 21 1929' (Status: 'Success'). The other three builds remain in the 'CANCELED' state. The rest of the interface, including the sidebar and top status bar, remains identical to the previous screenshot.

10. Check for it.

The screenshot shows a Jenkins job named 'case-pipeline' with a build number of 4. The 'Console Output' tab is selected. The output log shows the following:

```
Started by user SHIVAKUMAR
[Pipeline] Start of Pipeline
[Pipeline] stage
[Pipeline] | (Hello)
[Pipeline] name
Running on Jenkins: /home/ubuntu/jenkins/workspace/case-pipeline
[Pipeline] |
[Pipeline] with
Hello world
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/diveevamini/project-hello-world
> git init /home/ubuntu/jenkins/workspace/case-pipeline # timeout=10
Avoid second fetch
Checking out Revision 572d549e6f134209297bd15265743678455ff911 (refs/remotes/origin/master)
Commit message: "Create Dockerfile".
First time build. Wiping previous state
```

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

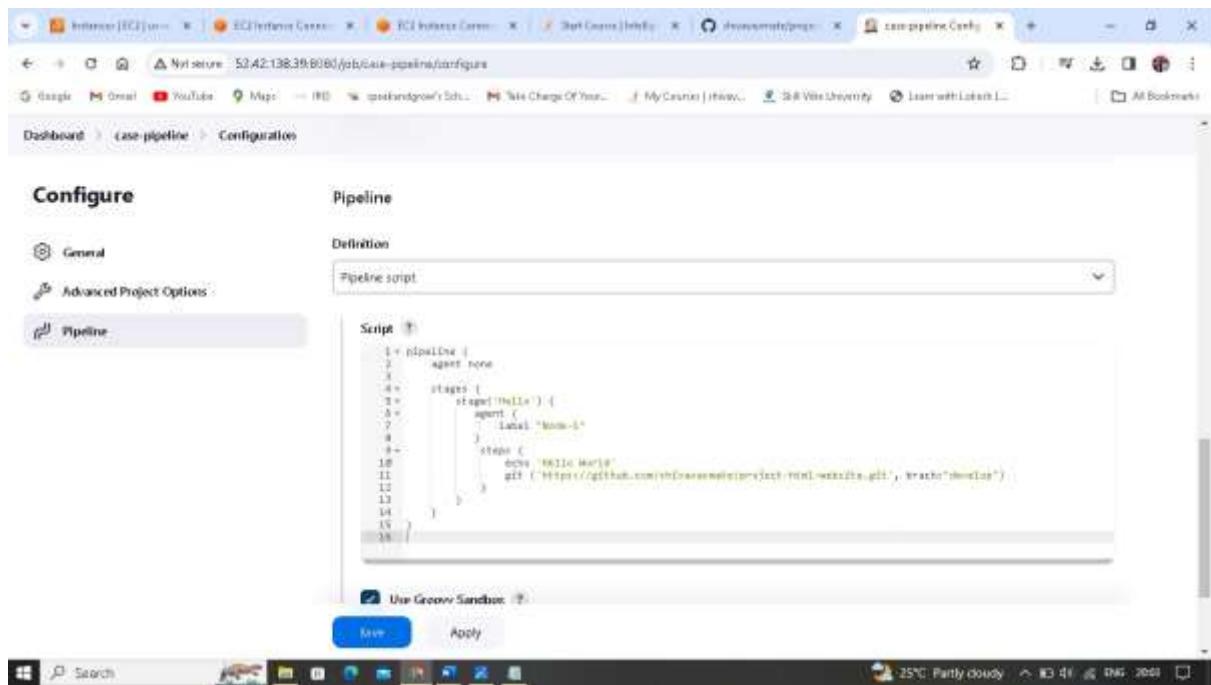
The screenshot shows a terminal window on an AWS instance connected via SSH. The user is in the directory `/home/ubuntu/jenkins/workspace/case-pipeline`. The terminal output shows the following command and its result:

```
ubuntu@ip-172-31-30-100:~$ ls
jenkins
ubuntu@ip-172-31-30-100:~$ cd jenkins
ubuntu@ip-172-31-30-100:~/jenkins$ ls
running warining.jar workspace
ubuntu@ip-172-31-30-100:~/jenkins$ cd workspace/
ubuntu@ip-172-31-30-100:~/jenkins/workspace$ ls
case-pipeline
```

Below the terminal window, a modal dialog box displays the Jenkins instance details:

i-05e9cddf5006b390a [Jenkins-Node1]
Public IP: 35.88.148.67 Private IP: 172.31.30.100

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")

            }

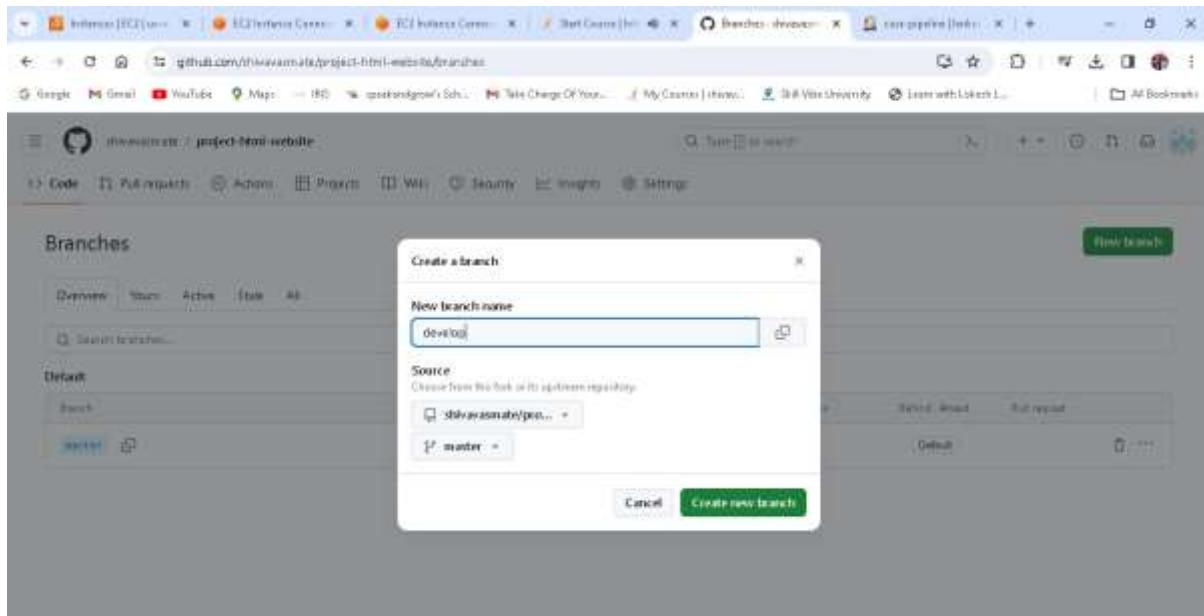
        }

    }

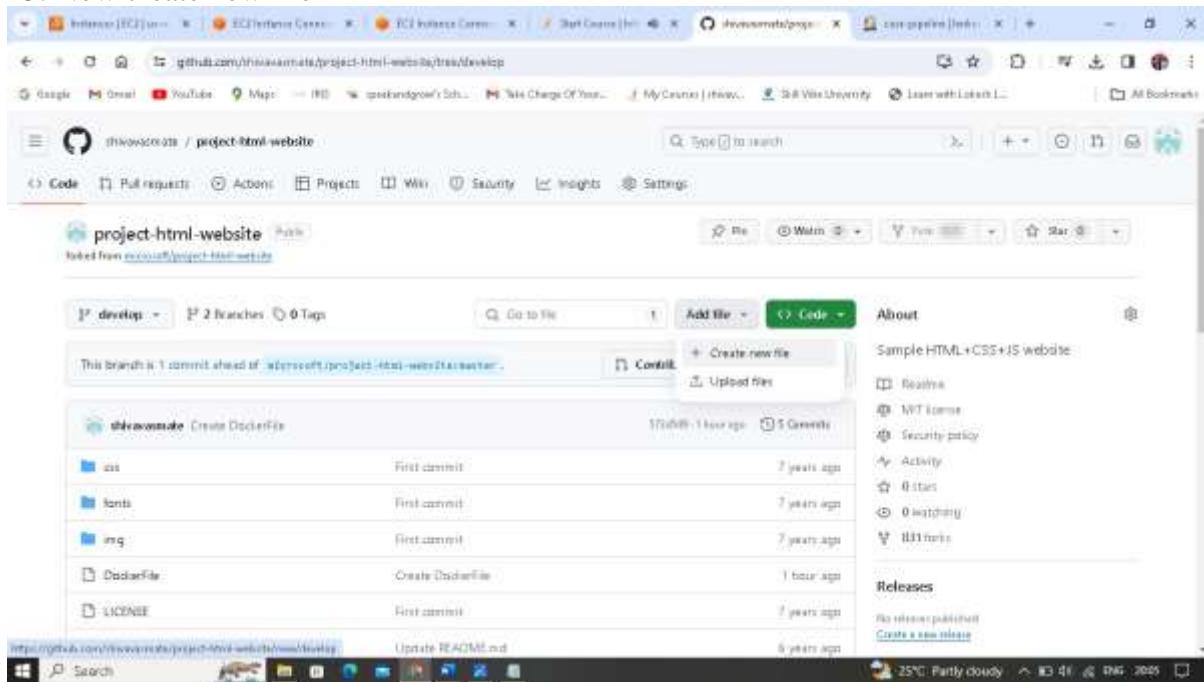
}
```

}

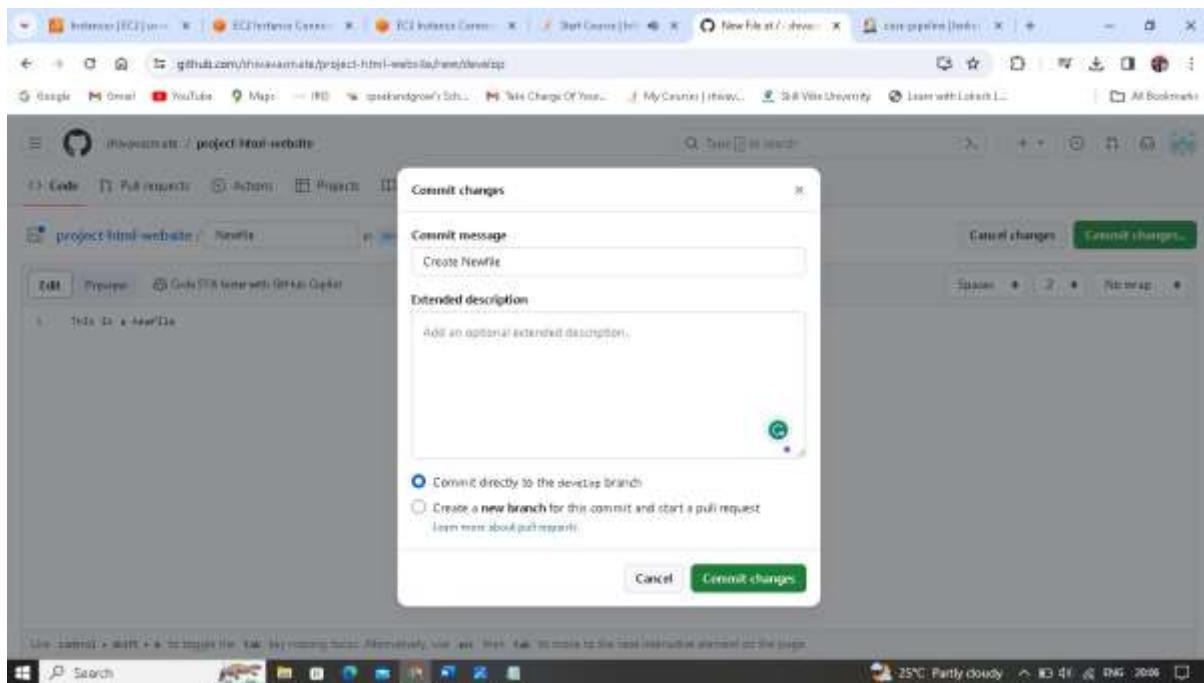
14. Create develop branch in the repo.



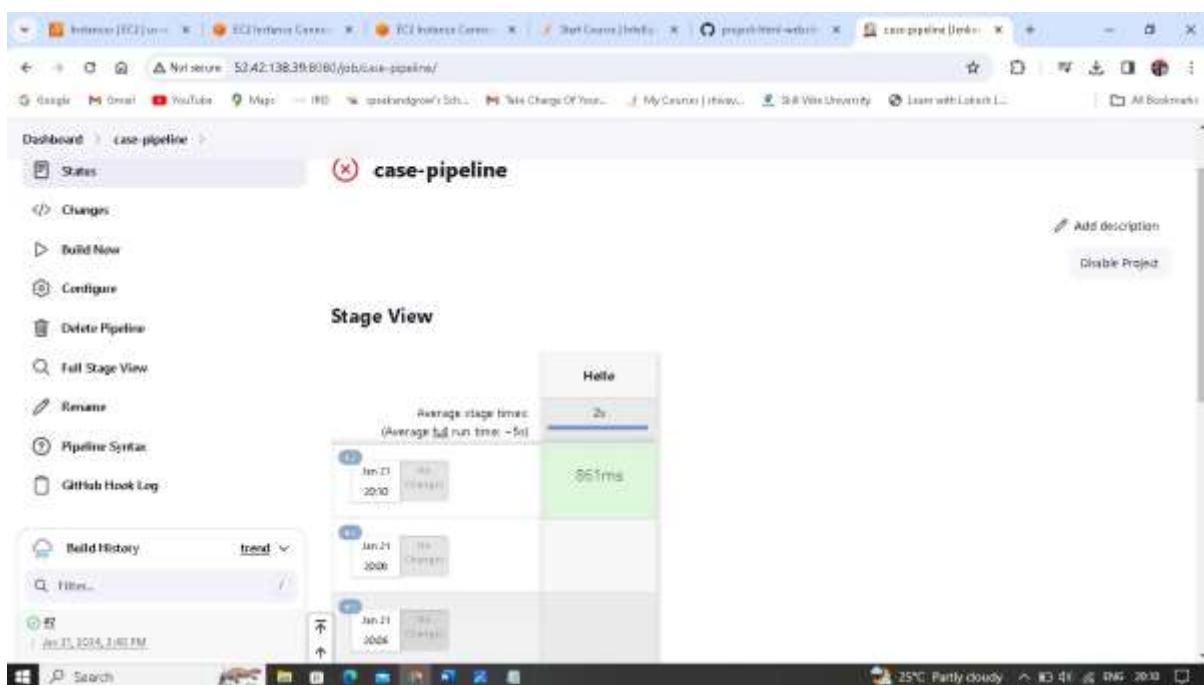
15. Now create new file



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'

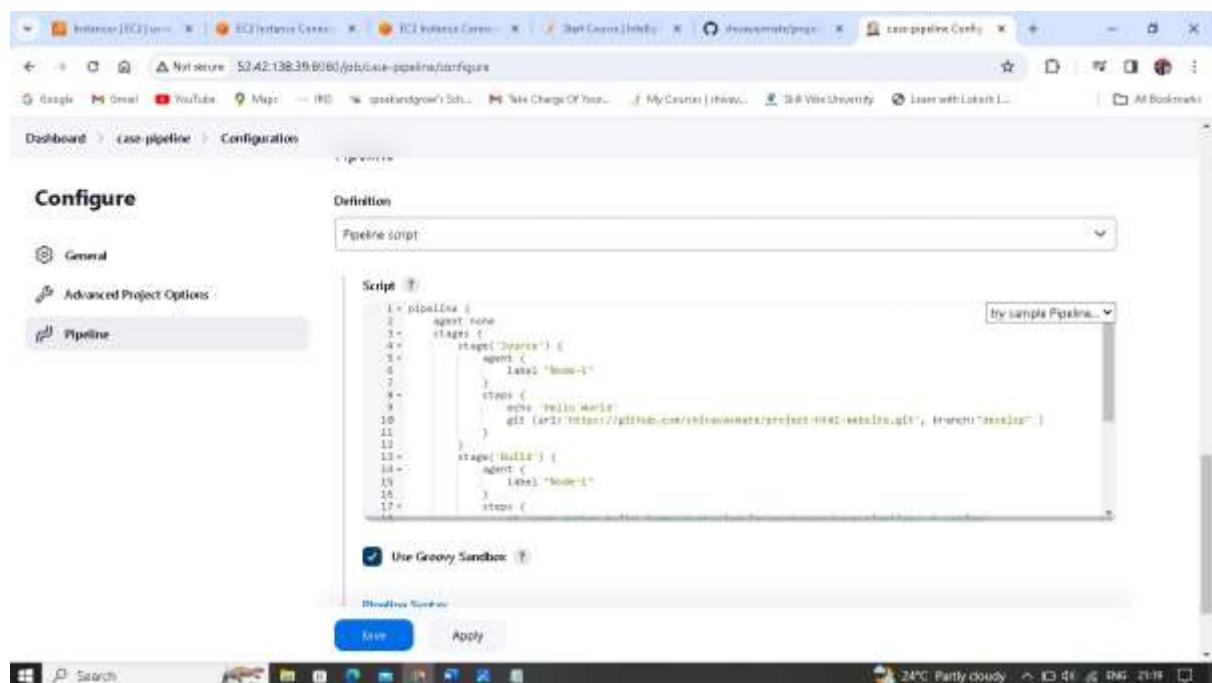
    }

}

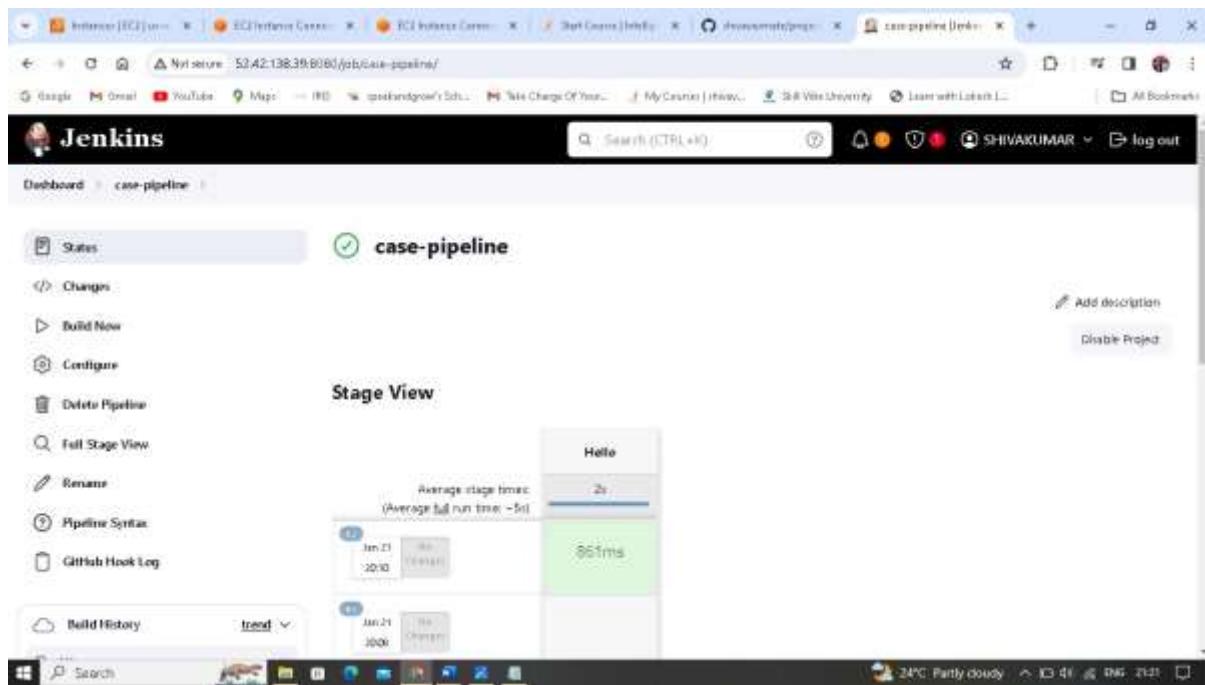
}

```

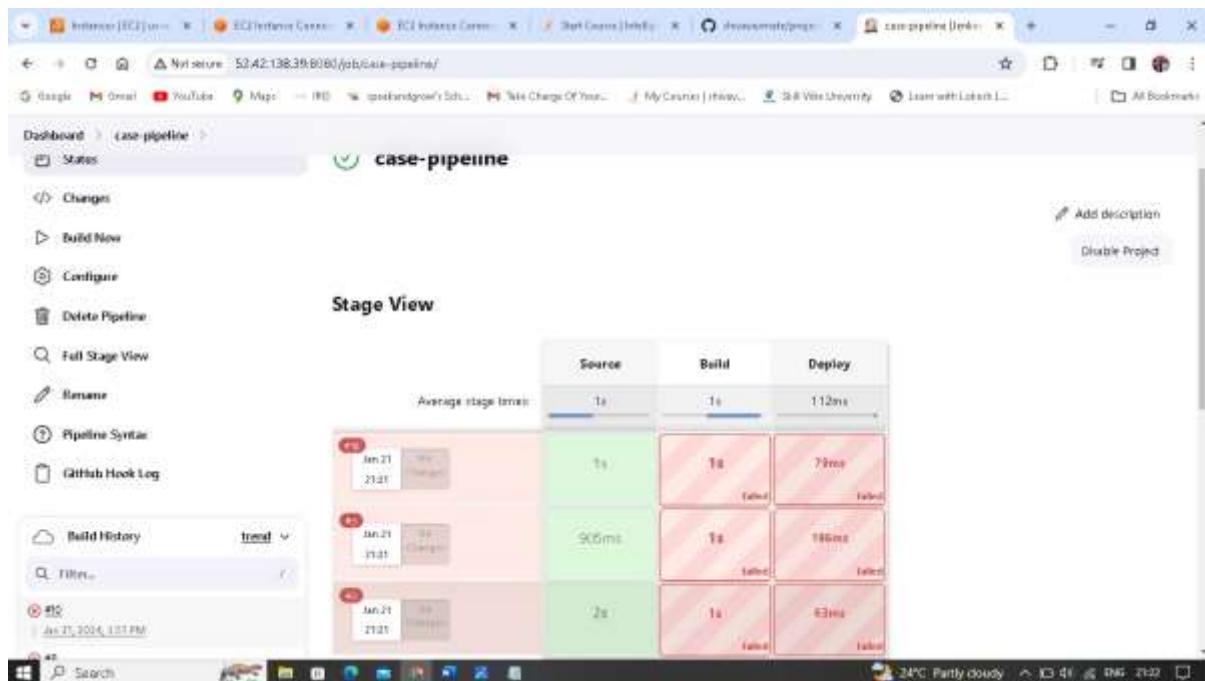
19. Click on save



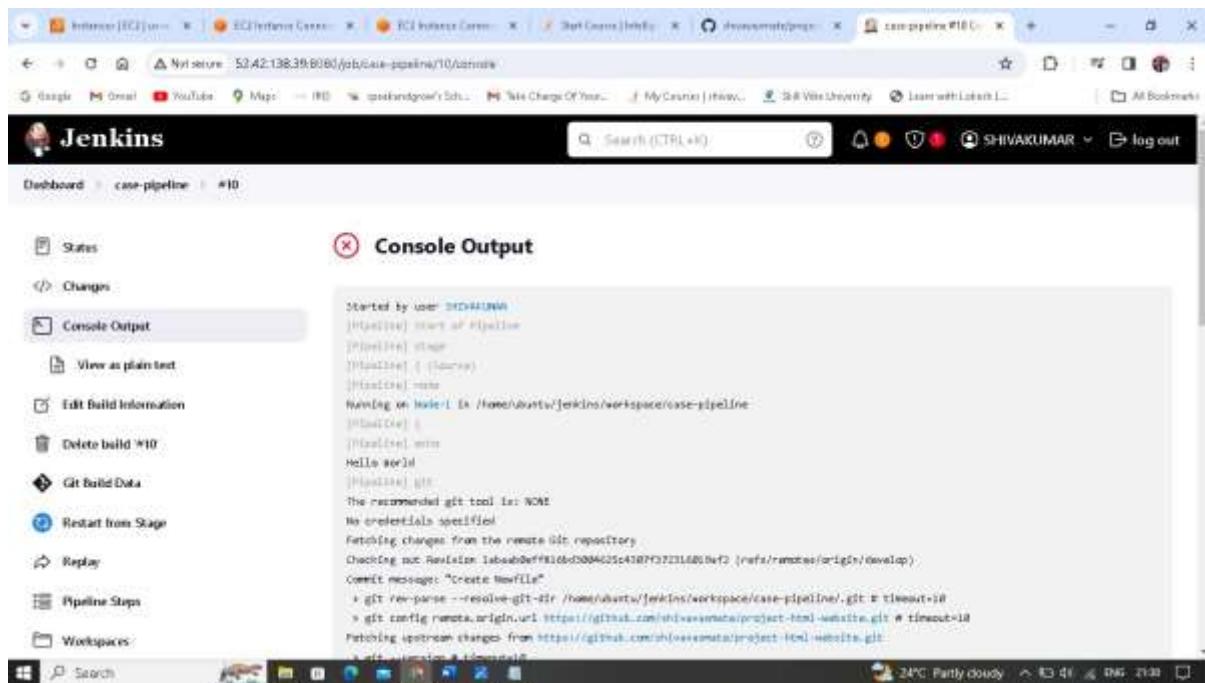
20. Click Build Now



21. Error in Build and Deploy stages

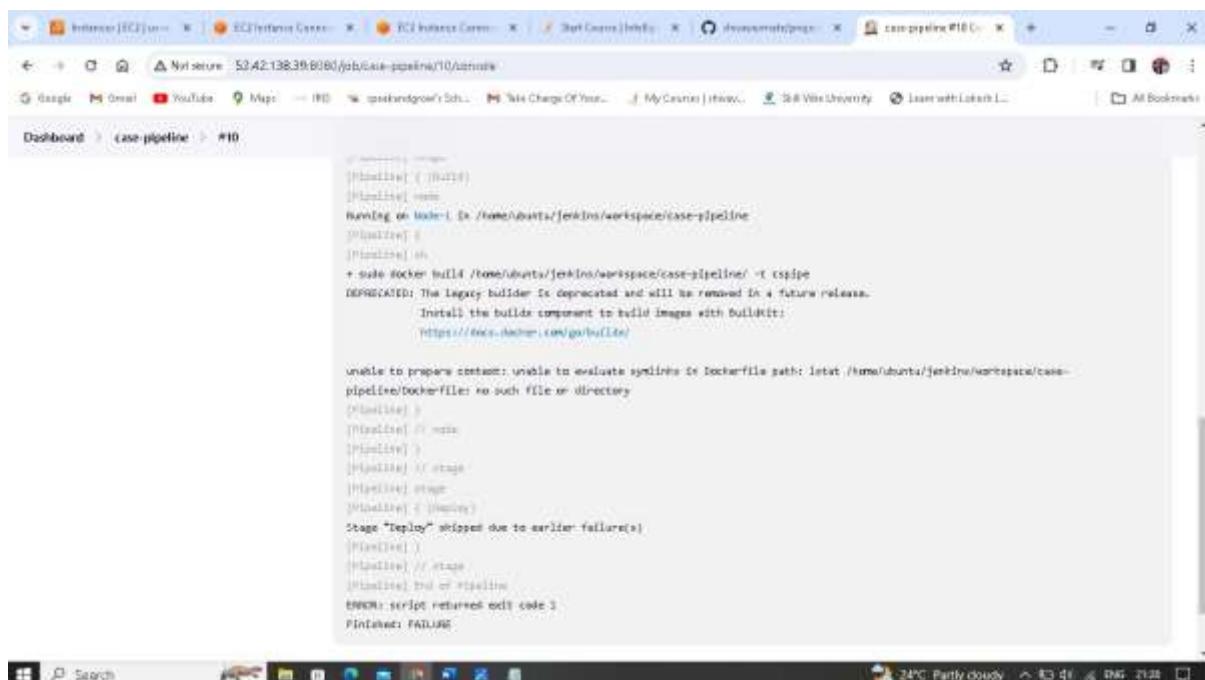


22. Check for error.



The screenshot shows the Jenkins interface for a job named 'case-pipeline'. The 'Console Output' tab is selected. The output shows a successful build process:

```
Started by user SHIVAKUMAR
[Pipeline] start of pipeline
[Pipeline] stage
[Pipeline] { (source)
[Pipeline] name
Running on master in /home/ubuntu/jenkins/workspace/case-pipeline
[Pipeline] {
[Pipeline] with
Hello world
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
Checking out Revision 1abab0eff8106d5004625c4387f372316809ef2 (refs/remotes/origin/develop)
Commit message: "Create NewFile".
+ git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/case-pipeline/.git & timeout=10
+ git config remote.origin.url https://github.com/shivakumar1991/project-HCI-website.git & timeout=10
Fetching upstream changes from https://github.com/shivakumar1991/project-HCI-website.git
```



The screenshot shows the Jenkins interface for the same job. The 'Console Output' tab is selected. The output shows an error in the Dockerfile path:

```
[Pipeline] {
[Pipeline] with
Running on master in /home/ubuntu/jenkins/workspace/case-pipeline
[Pipeline] {
[Pipeline] with
+ sudo docker build /home/ubuntu/jenkins/workspace/case-pipeline/ -t csippe
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildkit component to build images with buildkit:
https://docs.docker.com/guide/buildkit/
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] { (memory)
Stage "Deploy" skipped due to earlier failure(s)
[Pipeline] }
[Pipeline] // stage
[Pipeline] end of pipeline
ERROR: script returned exit code 1
FinDocker: FAILURE
```

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
```

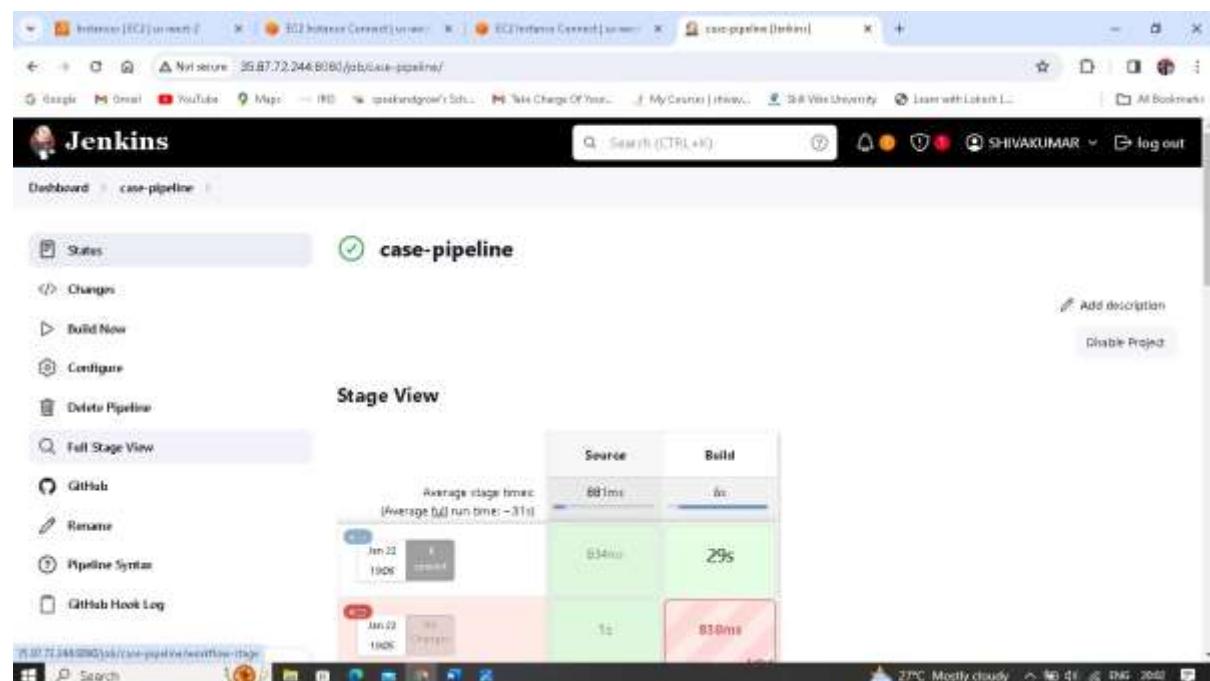
```
REPOSITORY TAG IMAGE ID CREATED SIZE
```

```
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.

The screenshot shows a Jenkins pipeline status page titled "case-pipeline". The pipeline has three stages: Source, Build, and Deploy. The Source stage is labeled "Success" with a green background. The Build stage shows a duration of 760ms. The Deploy stage shows a duration of 78ms. The pipeline has run 4 times successfully. The pipeline history shows three successful runs from Jan 22, 2020.

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

The screenshot shows a web browser displaying the Google homepage. The address bar shows the URL "35.93.156.116:84". The search bar contains the placeholder text "Search Google or type a URL.". The page features the classic Google logo and search interface.

27.Now HTML Application is running.

