

## Project-1

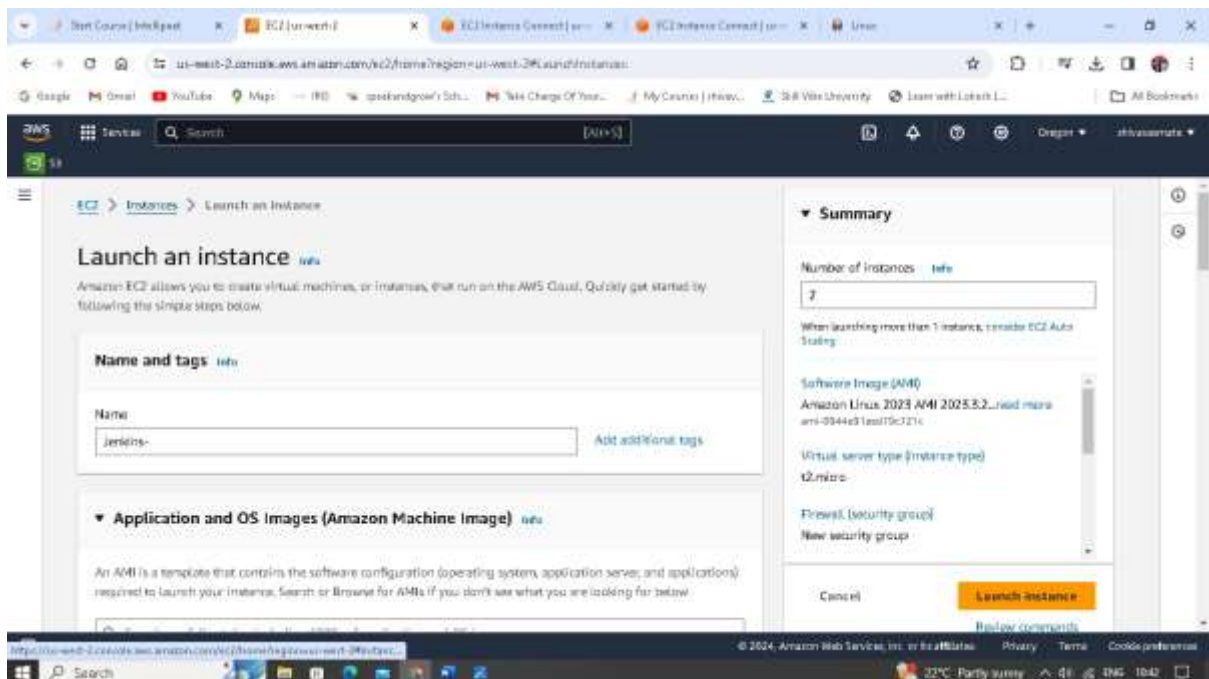
Project Description:

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

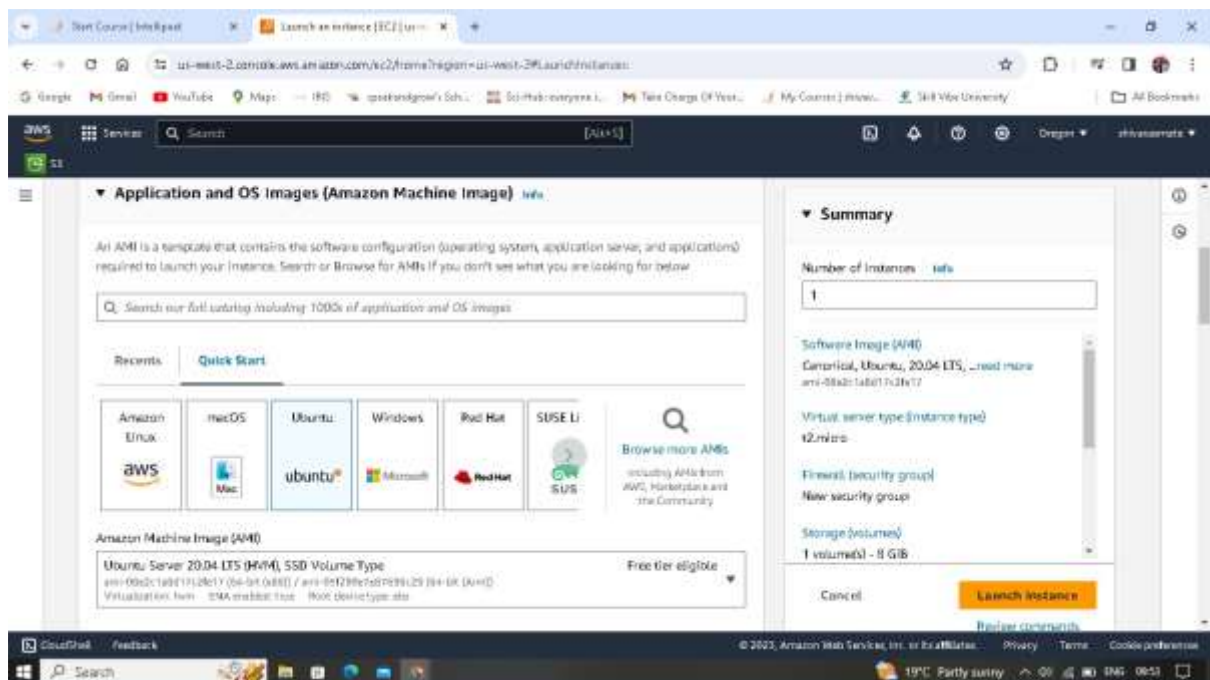
Ans:

### 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-

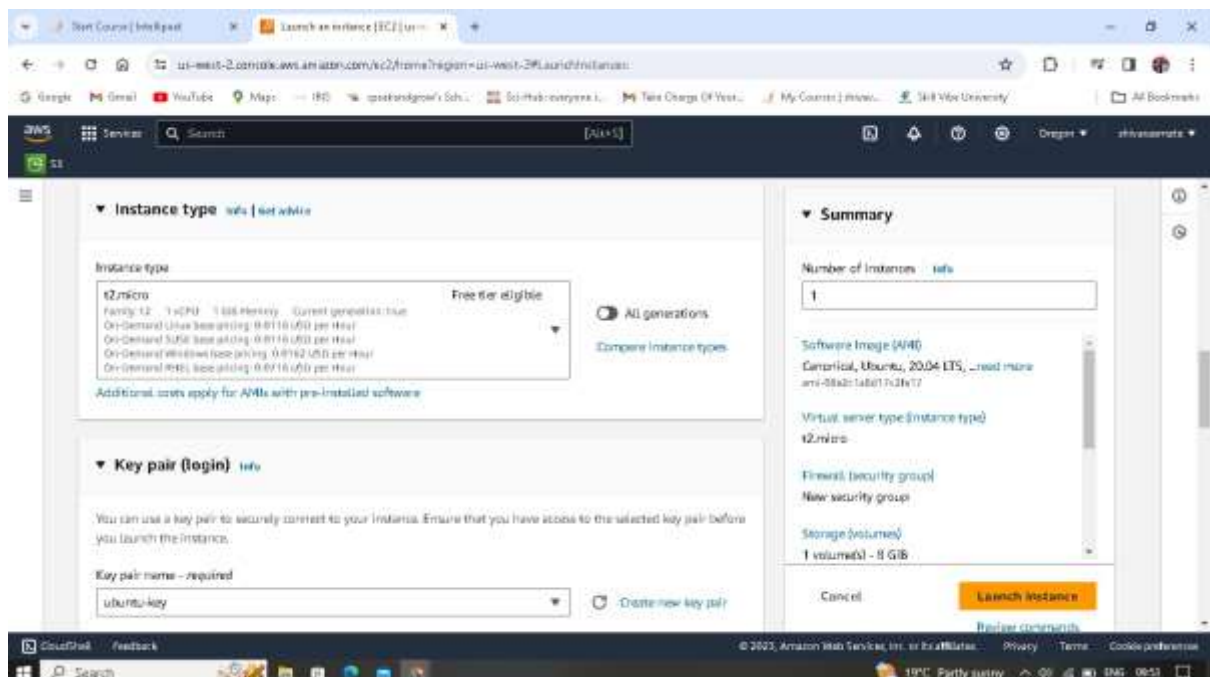


b. AMI: QuickStart >> Ubuntu [Any version which is free tier eligible]

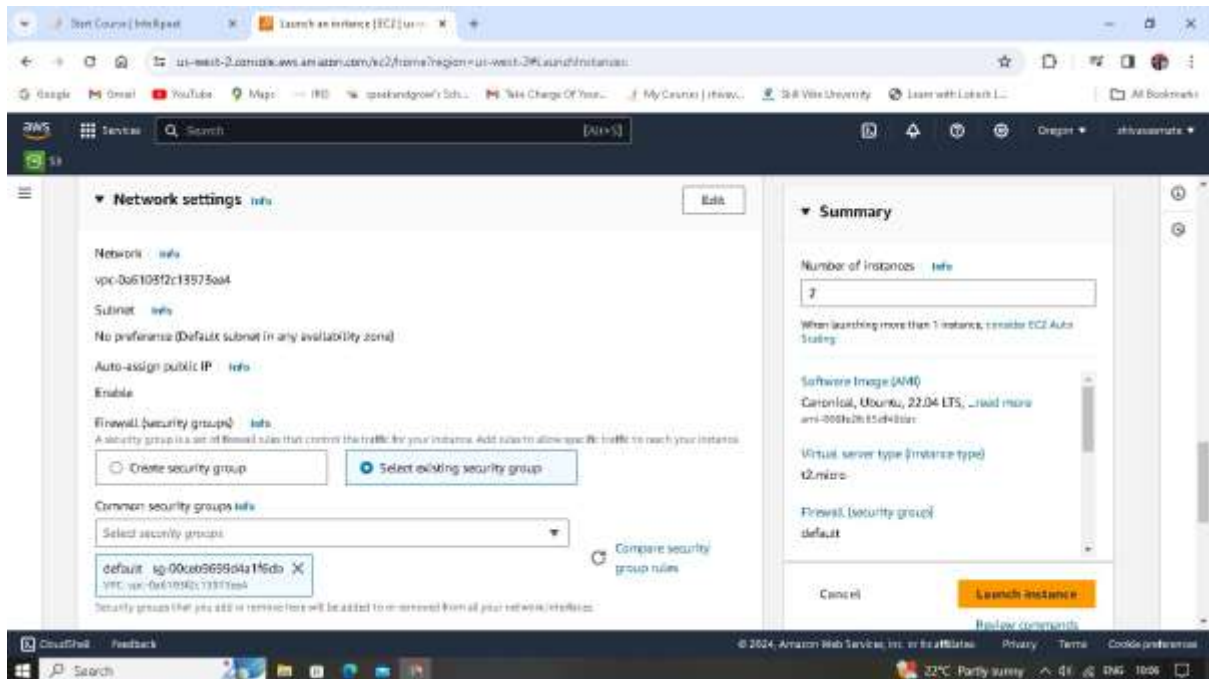


c. Instance type: t2. micro [free tier eligible]

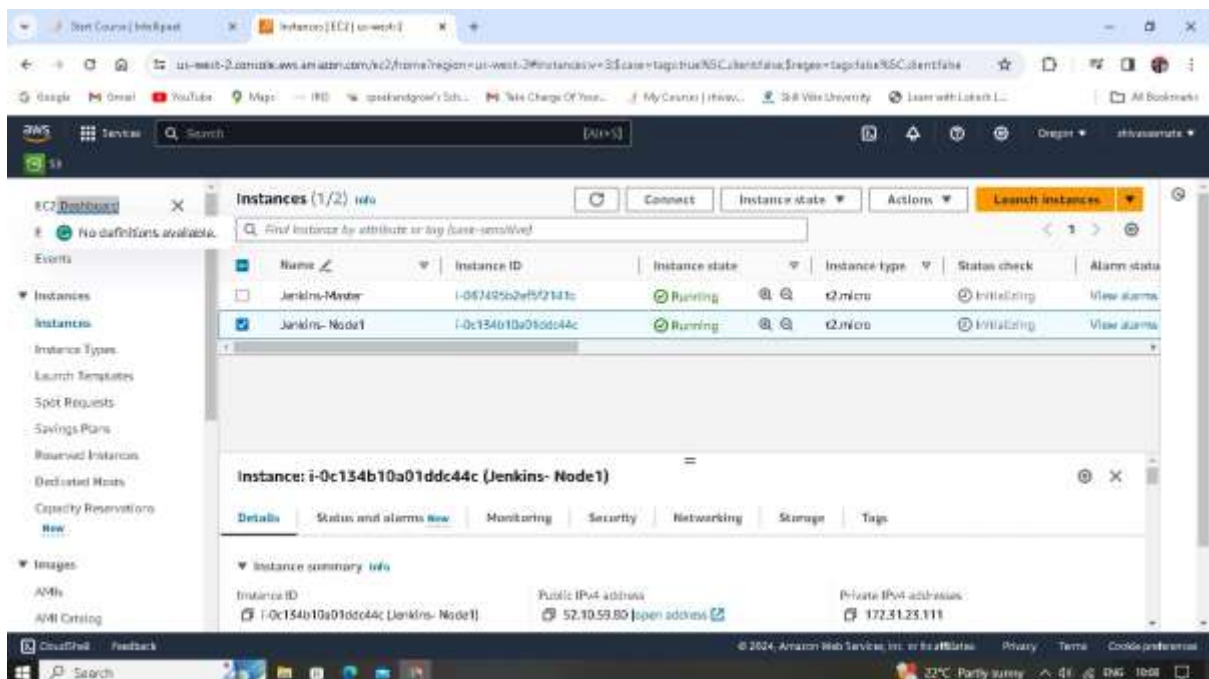
d. Key-pair: Create a key pair [rsa and .pem] with a name ubuntu-key



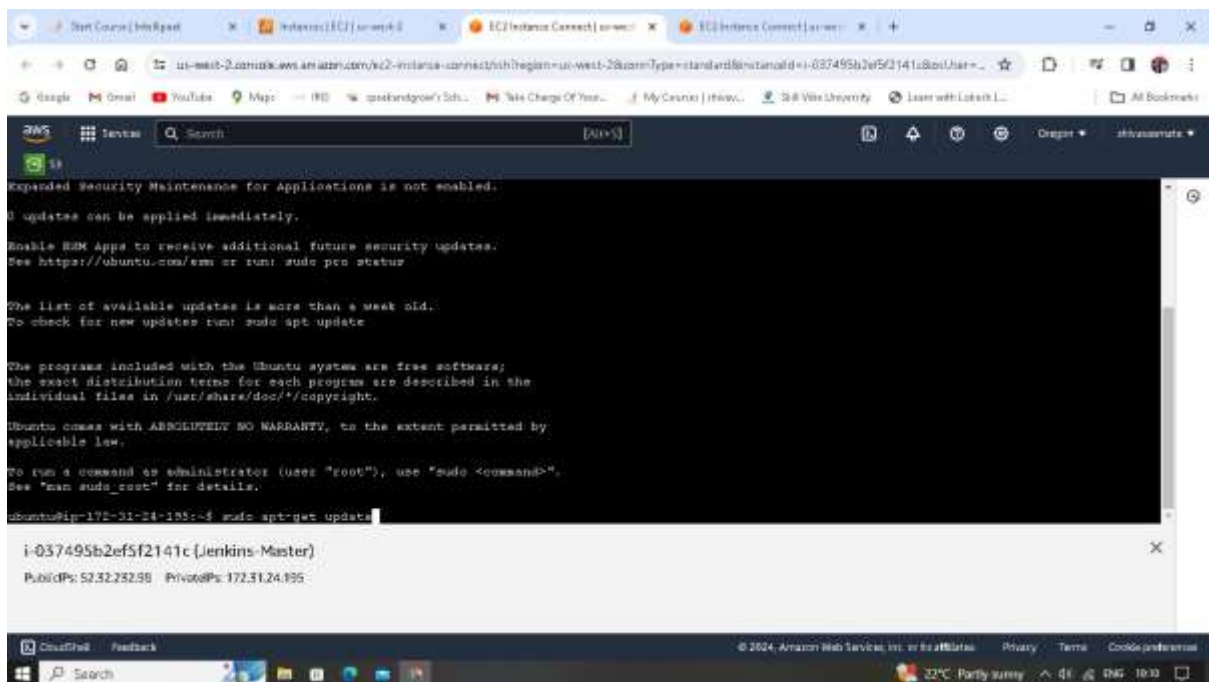
e. VPC and Security group as default with all traffic protocols allowed click on launch instance.



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`.



```
Expanded 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

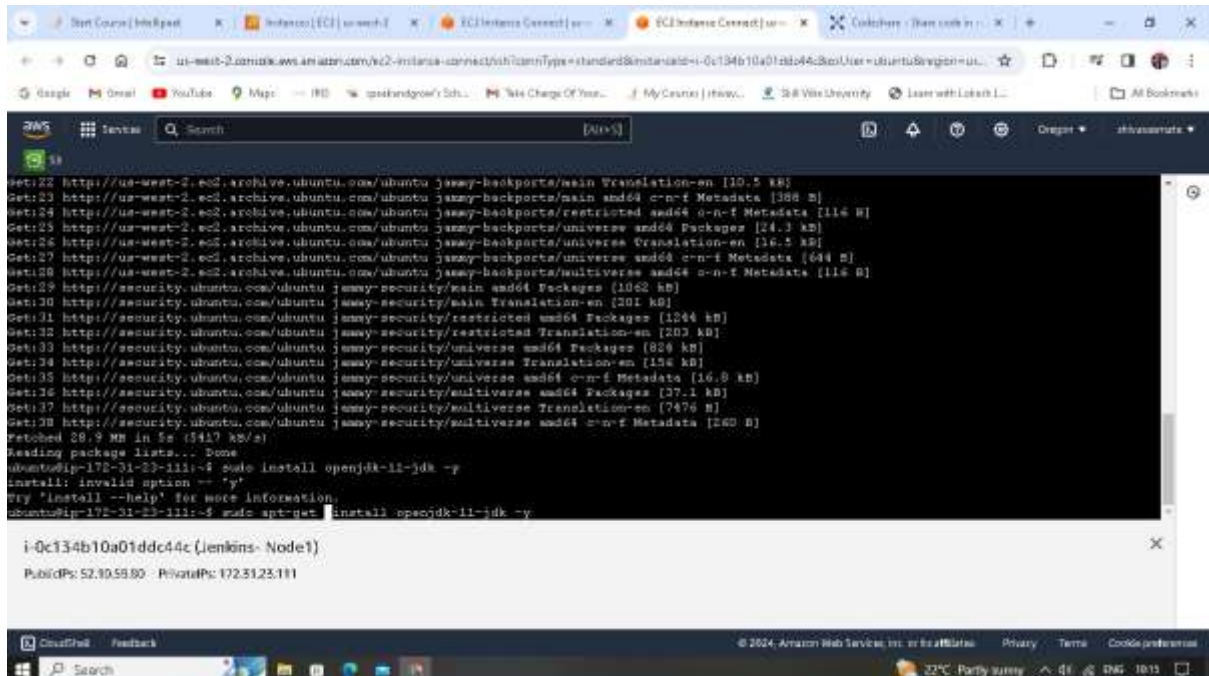
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-24-193:~$ sudo apt-get update
i-037495b2ef5f2141c (Jenkins-Master)
PublicPis: 52.82.252.58 PrivatePis: 172.31.24.195
```

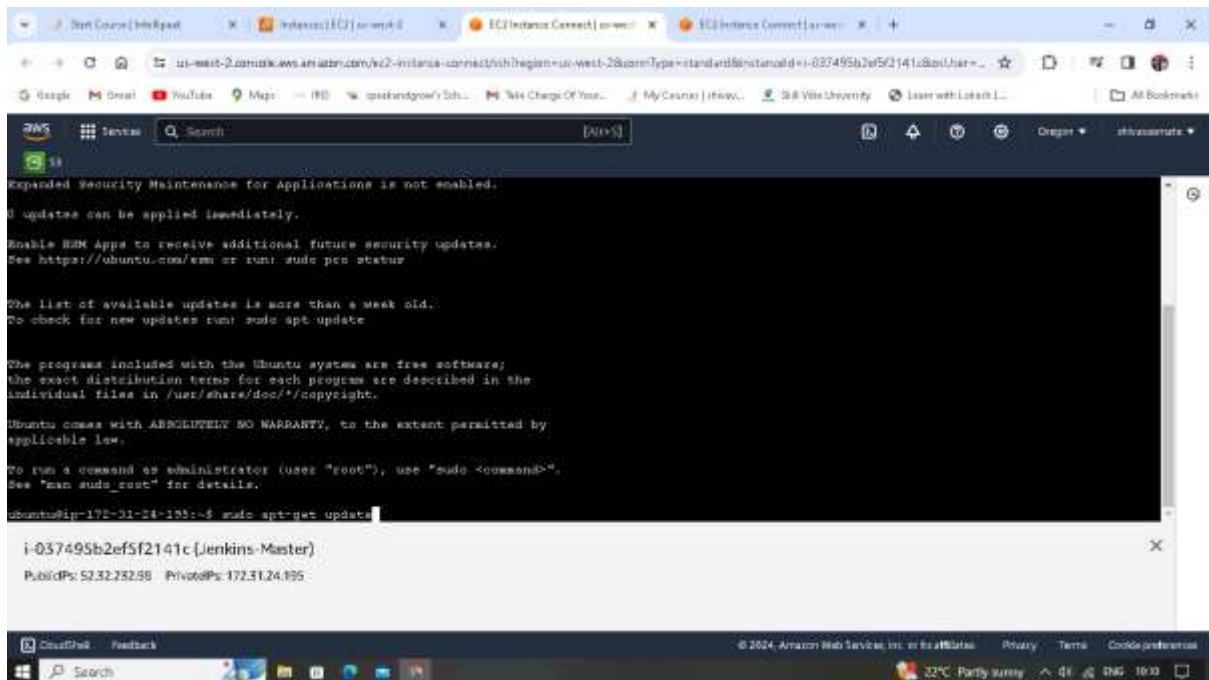
7. Install java using command `sudo apt-get install openjdk-11-jdk -y`



```
Get:22 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/main Translation-en [10.5 kB]
Get:23 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:25 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/universe amd64 Packages [24.3 kB]
Get:26 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/universe Translation-en [16.5 kB]
Get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/universe amd64 c-n-f Metadata [644 B]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jessy-security/main amd64 Packages [1052 kB]
Get:30 http://security.ubuntu.com/ubuntu jessy-security/main Translation-en [201 kB]
Get:31 http://security.ubuntu.com/ubuntu jessy-security/restricted amd64 Packages [1244 kB]
Get:32 http://security.ubuntu.com/ubuntu jessy-security/restricted Translation-en [202 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 [156 kB]
Get:35 http://security.ubuntu.com/ubuntu jessy-security/universe amd64 c-n-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 c-n-f Metadata [260 B]
Fetched 28.9 MB in 5s (5417 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-23-111:~$ sudo install openjdk-11-jdk -y
install: invalid option -- 'y'
Try 'install --help' for more information.
ubuntu@ip-172-31-23-111:~$ sudo apt-get install openjdk-11-jdk -y
i-0c134b10a01ddc44c (Jenkins-Node1)
PublicPis: 52.80.59.80 PrivatePis: 172.31.23.111
```



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



```
Expanded Security Maintenance for Applications is not enabled.
3 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pcs status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

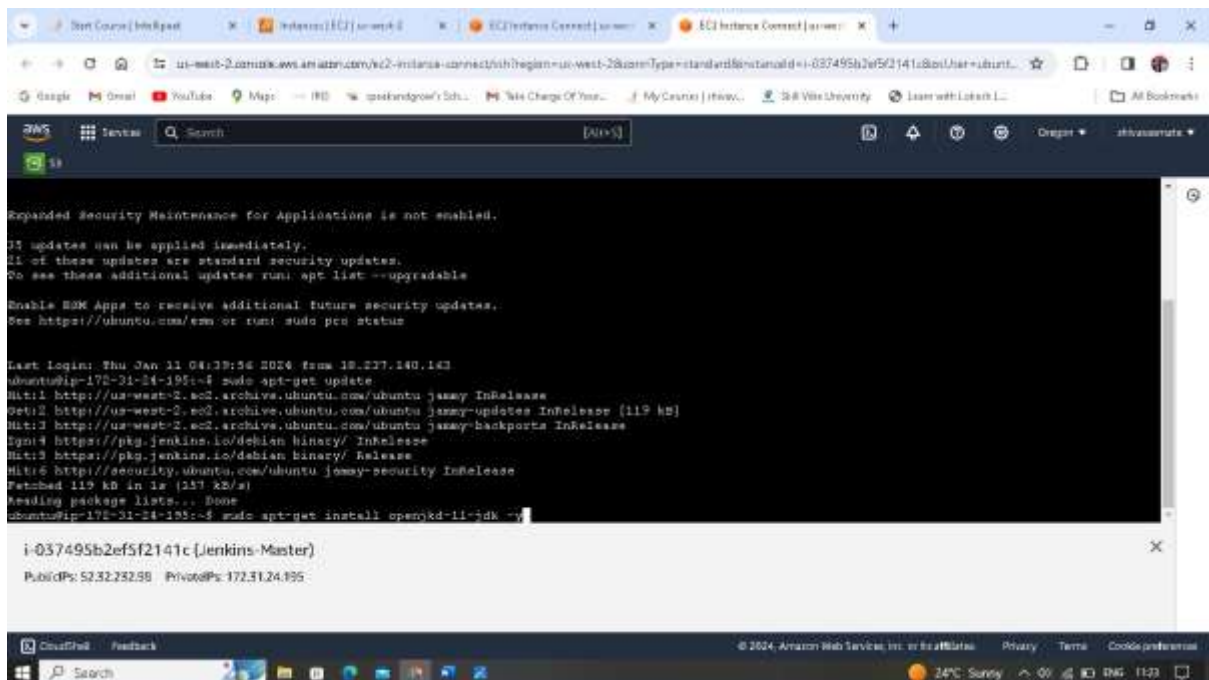
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-24-195:~$ sudo apt-get update
```

i-037495b2ef5f2141c (Jenkins-Master)  
PublicIPs: 52.32.232.98 PrivateIPs: 172.31.24.195



```
Expanded Security Maintenance for Applications is not enabled.
11 updates can be applied immediately.
11 of these 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 pcs status

Last login: Thu Jan 11 04:39:56 2024 from 10.227.140.143
ubuntu@ip-172-31-24-195:~$ sudo apt-get update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-updates InRelease (119 kB)
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian binary/ Release
Hit:6 http://security.ubuntu.com/ubuntu jessy-security InRelease
Fetched 119 kB in 1s (257 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-24-195:~$ sudo apt-get install openjdk-11-jdk -y
```

i-037495b2ef5f2141c (Jenkins-Master)  
PublicIPs: 52.32.232.98 PrivateIPs: 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
```

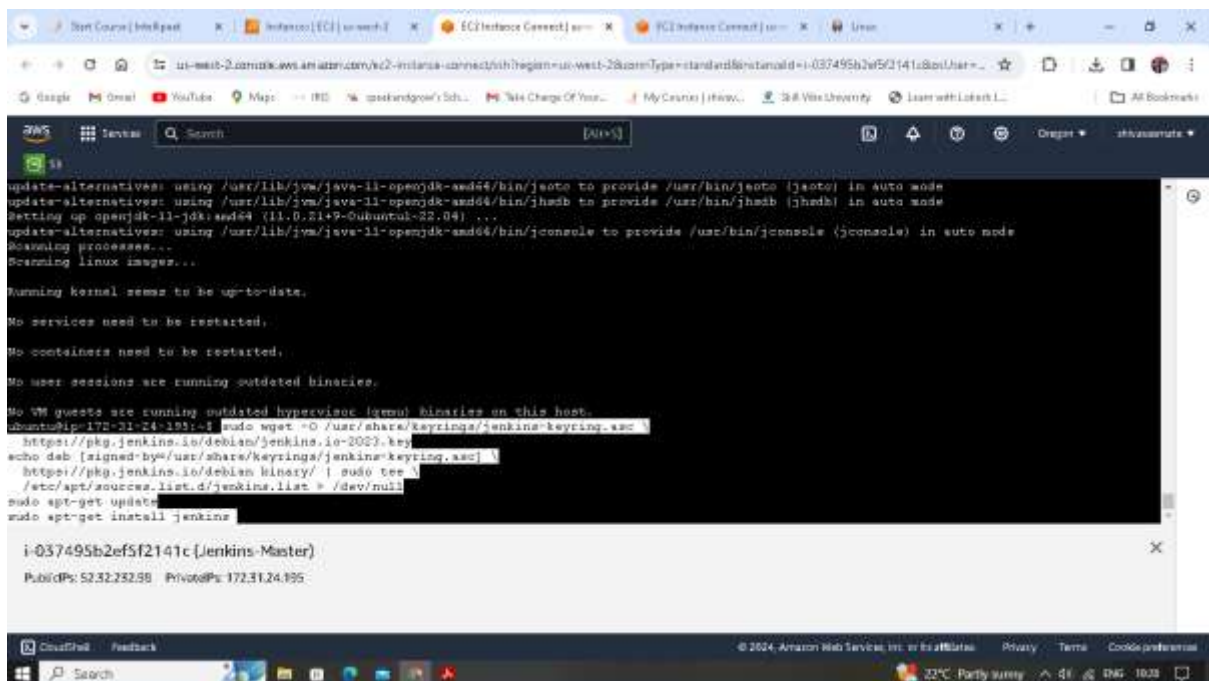
```
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
```

```
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 within an AWS CloudShell environment. The terminal output displays the execution of the commands provided in the previous blocks. It shows the download of the Jenkins keyring, the addition of the Jenkins repository to the sources list, and the successful installation of Jenkins. The terminal also shows system updates and package management status. The instance is identified as 'i-037495b2ef5f2141c (Jenkins-Master)' with public IP 52.82.282.98 and private IP 172.31.24.195.

```
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jre to provide /usr/bin/jre (jre) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jhsdb to provide /usr/bin/jhsdb (jhsdb) in auto mode
Setting up openjdk-11-jdk:amd64 (11.0.21+9-Substnuk-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...
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-172-31-24-195:~$ 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)
PublicIPs: 52.82.282.98 PrivateIPs: 172.31.24.195
```

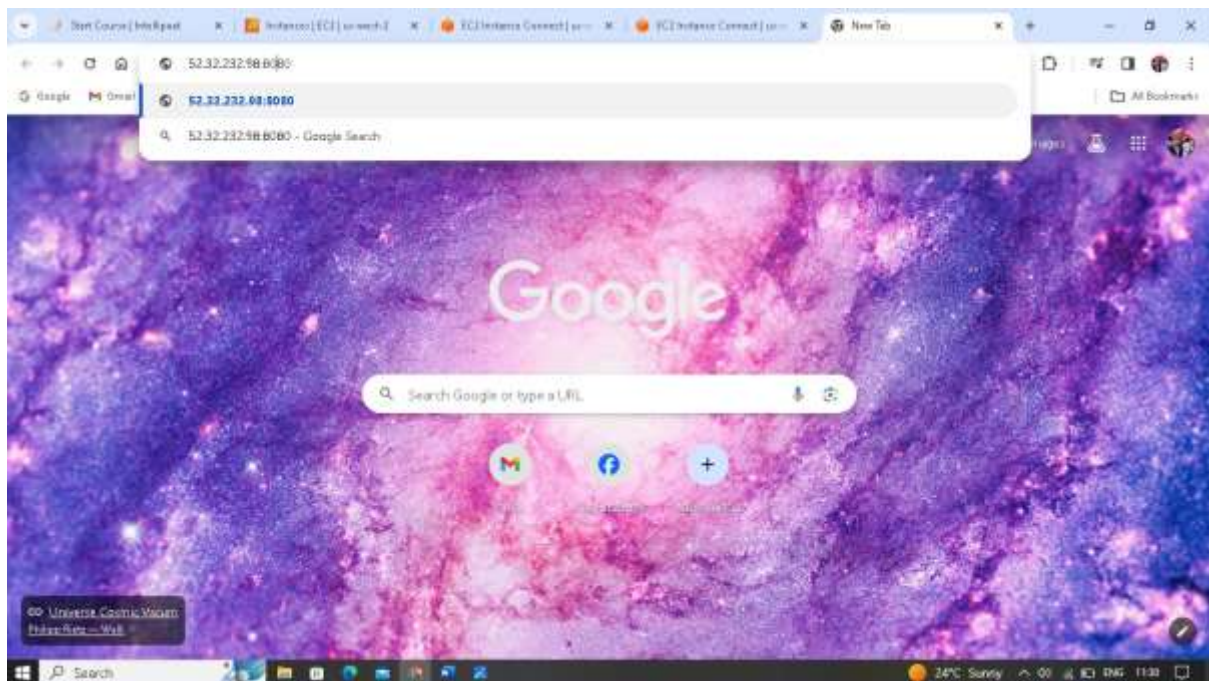


The screenshot shows a terminal window with a context menu open over a URL. The menu options include 'Copy', 'Copy text to highlight', 'Go to 52.82.282.98', 'Paste', 'Translate selection to English', and 'Open reading mode NOW'. The terminal output shows the installation of Jenkins and the execution of the commands provided in the previous blocks.

```
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)
PublicIPs: 52.82.282.98 PrivateIPs: 172.31.24.195
```

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

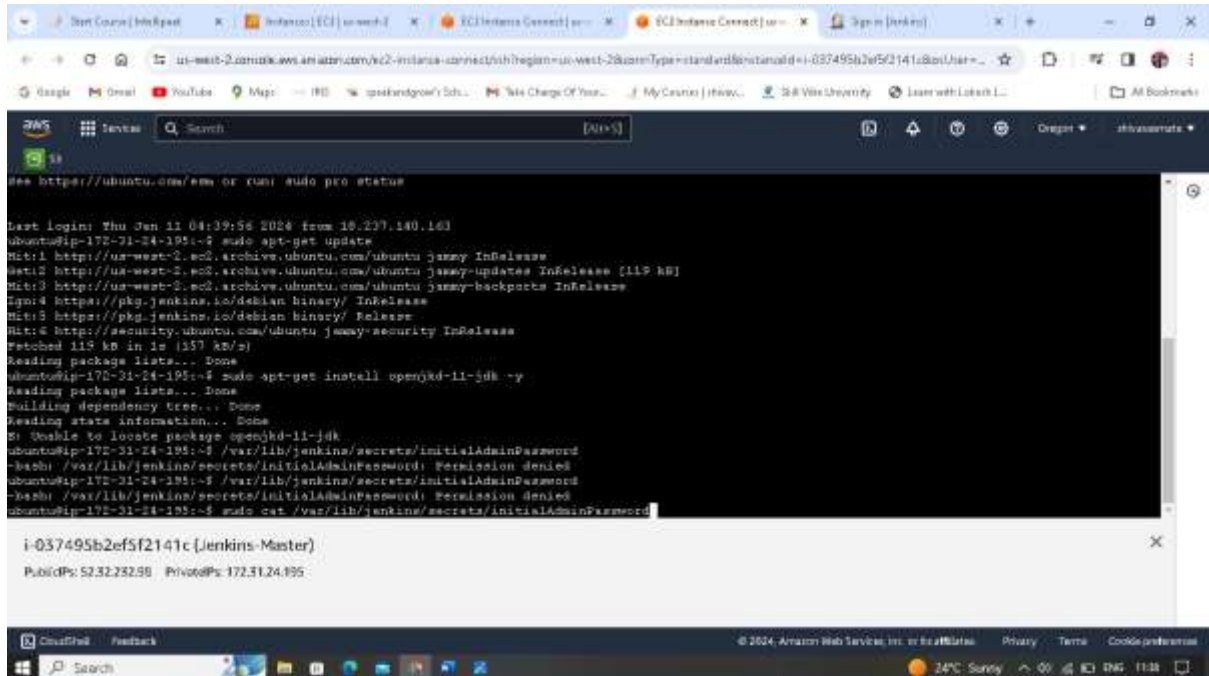


11. Jenkins sign-in page opens.



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

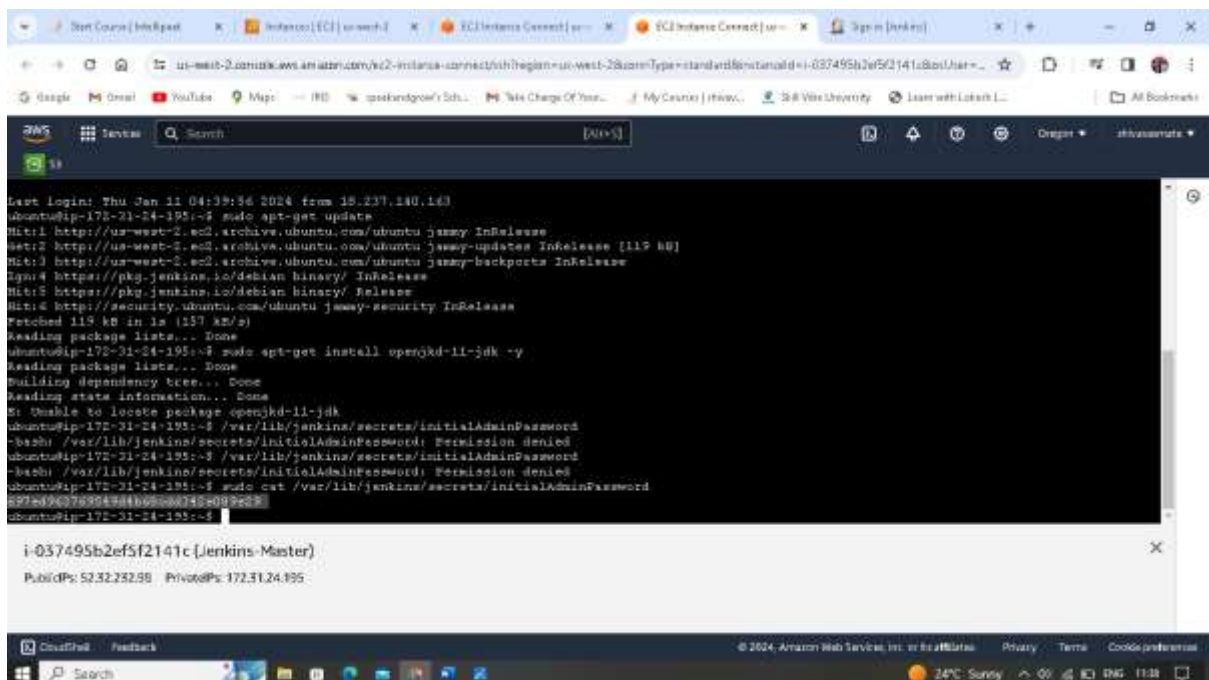
Go to Jenkins-Master using command `sudo cat` paste it.



```
See https://ubuntu.com/sem or run: sudo pro status

Last login: Thu Jan 11 04:39:56 2024 from 18.237.140.143
ubuntu@ip-172-31-24-195:~$ sudo apt-get update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-updates InRelease [119 kB]
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports InRelease
Hit:4 https://pkg.jenkins.io/debian binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian binary/ Release
Hit:6 https://security.ubuntu.com/ubuntu jessy-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
S: Unable to locate package openjdk-11-jdk
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
i-037495b2ef5f2141c (Jenkins-Master)
PublicPs: 52.32.252.58 PrivatePs: 172.31.24.195
```

13. Hit enter to get password copy it.



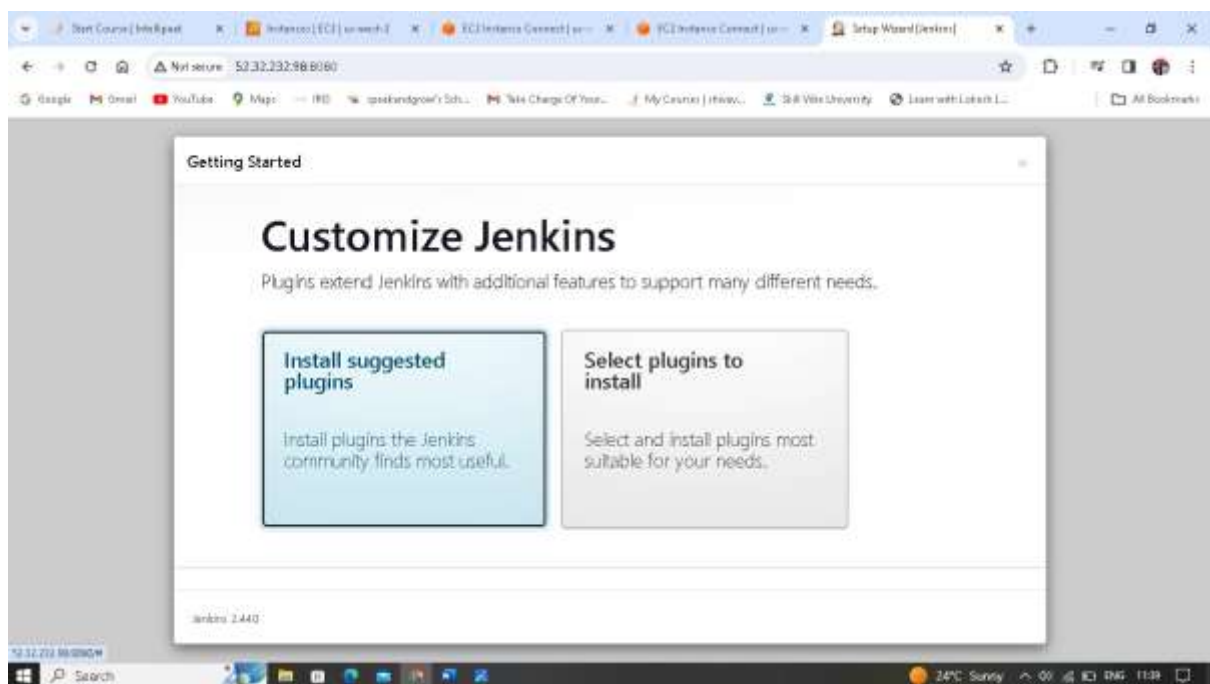
```
Last login: Thu Jan 11 04:39:56 2024 from 18.237.140.143
ubuntu@ip-172-31-24-195:~$ sudo apt-get update
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy InRelease
Hit:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-updates InRelease [119 kB]
Hit:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports InRelease
Hit:4 https://pkg.jenkins.io/debian binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian binary/ Release
Hit:6 https://security.ubuntu.com/ubuntu jessy-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
S: Unable to locate package openjdk-11-jdk
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ /var/lib/jenkins/secrets/initialAdminPassword
-bash: /var/lib/jenkins/secrets/initialAdminPassword: Permission denied
ubuntu@ip-172-31-24-195:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
i-037495b2ef5f2141c (Jenkins-Master)
PublicPs: 52.32.252.58 PrivatePs: 172.31.24.195
```



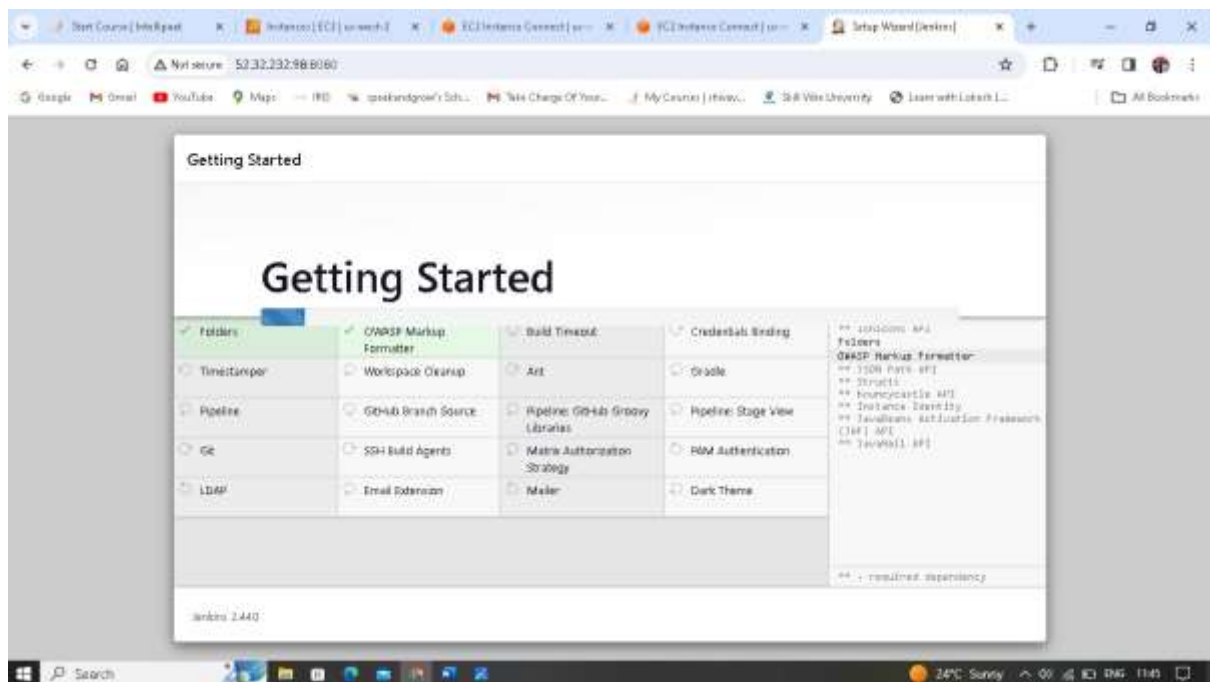
14. Paste the password and click on continue



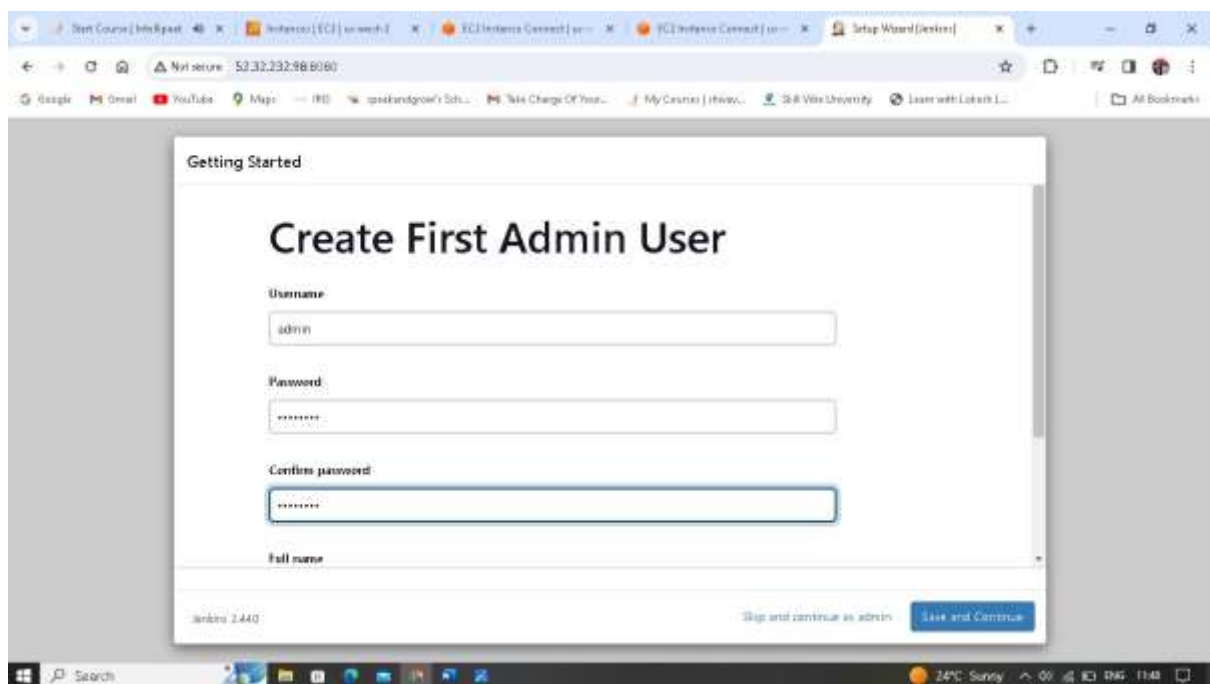
15. Click on install suggested plugins



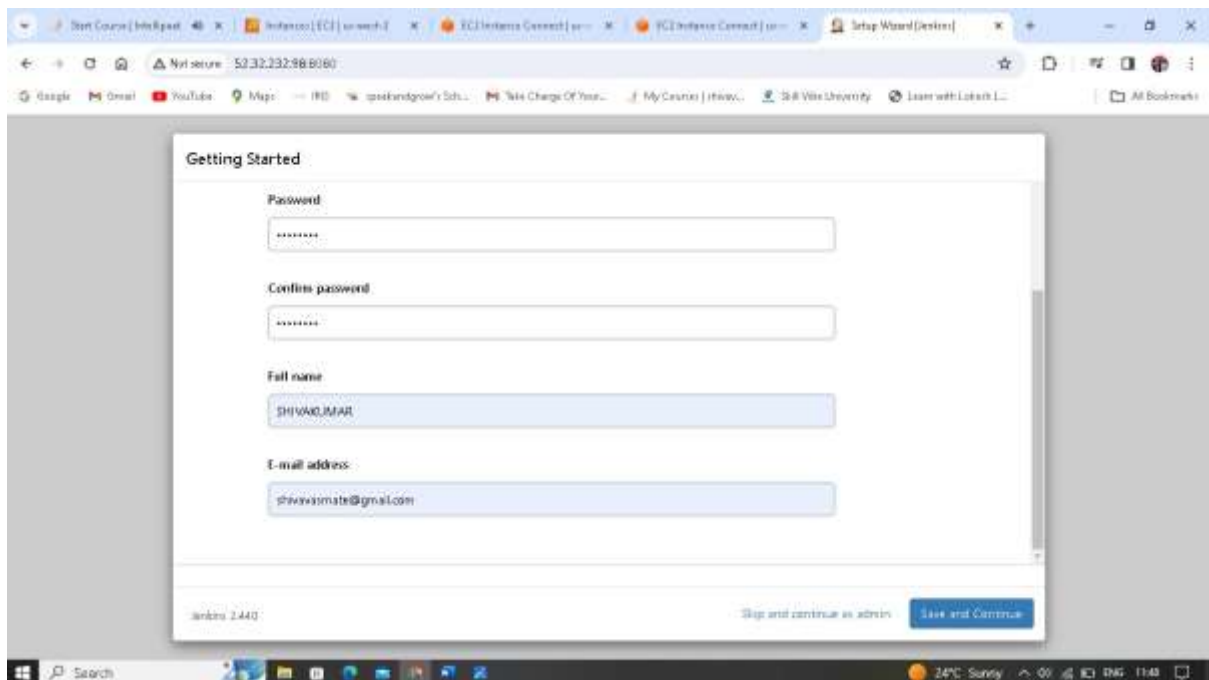
16. Just wait for all the plugins to install



17. Now create first admin user.



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

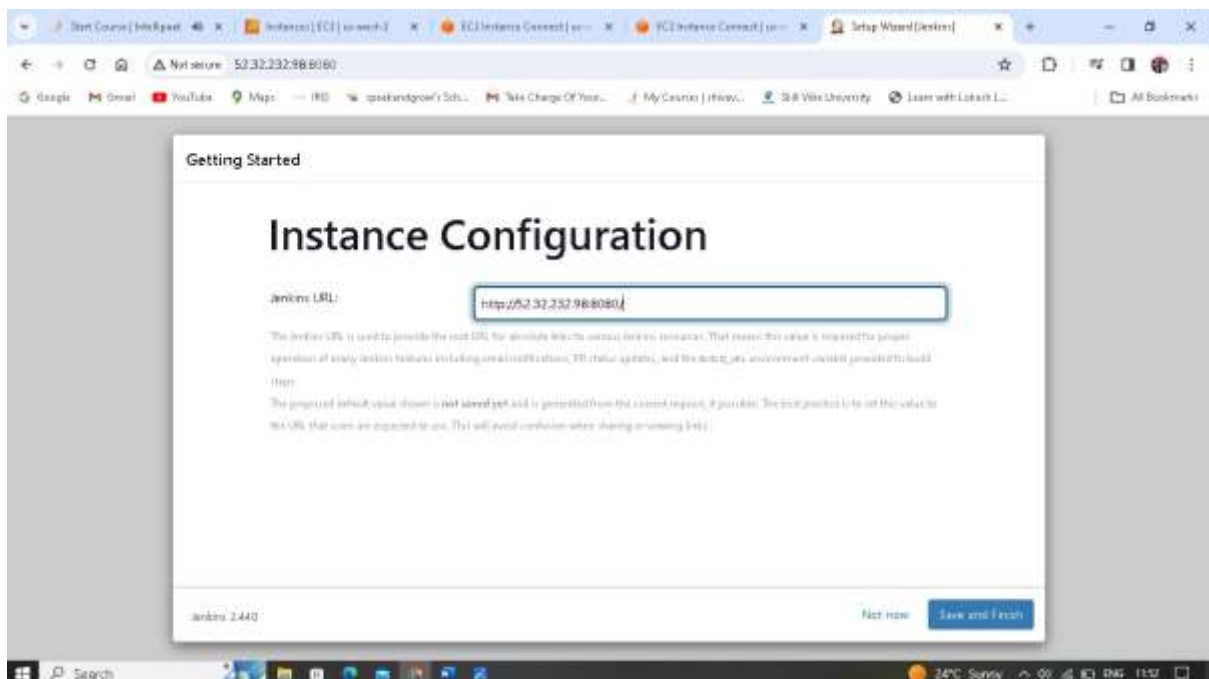


The screenshot shows the 'Getting Started' screen of the Jenkins Setup Wizard. The browser address bar shows 'http://52.32.232.98:8080/'. The form contains the following fields:

- Password:** A text box with a masked password (dots).
- Confirm password:** A text box with a masked password (dots).
- Full name:** A text box containing 'SHIVAM JARAT'.
- E-mail address:** A text box containing 'shivamjarat@gmail.com'.

At the bottom right, there is a 'Save and Continue' button. The bottom of the window shows a Windows taskbar with the date and time '24°C Sunny' and '11:48'.

19 clicks save and finish

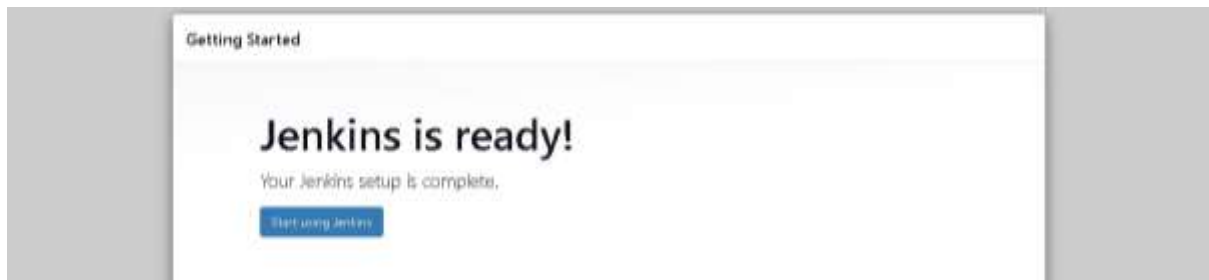


The screenshot shows the 'Instance Configuration' screen of the Jenkins Setup Wizard. The browser address bar shows 'http://52.32.232.98:8080/'. The form contains the following fields:

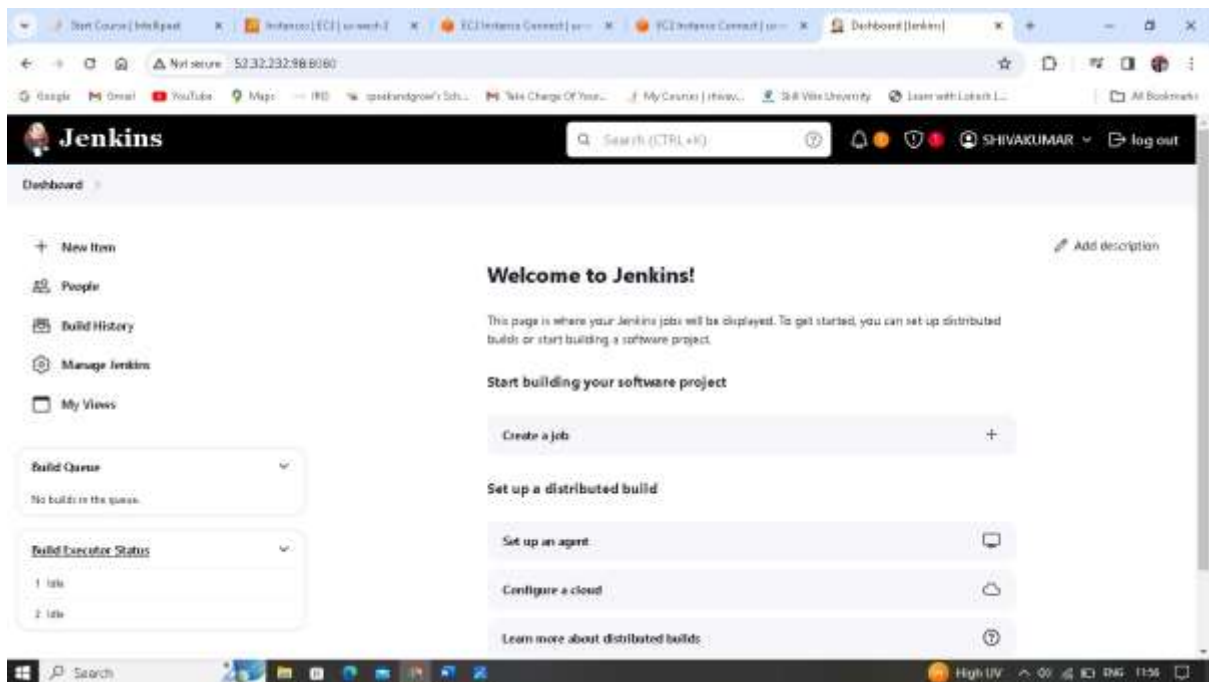
- Jenkins URL:** A text box containing 'http://52.32.232.98:8080/'.

Below the text box, there is a note: 'The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the default Jenkins environment variable provided to build jobs. The proposed default value shown is not ideal yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.'

At the bottom right, there is a 'Save and Finish' button. The bottom of the window shows a Windows taskbar with the date and time '24°C Sunny' and '11:50'.



20. With this Jenkins dashboard is setup.

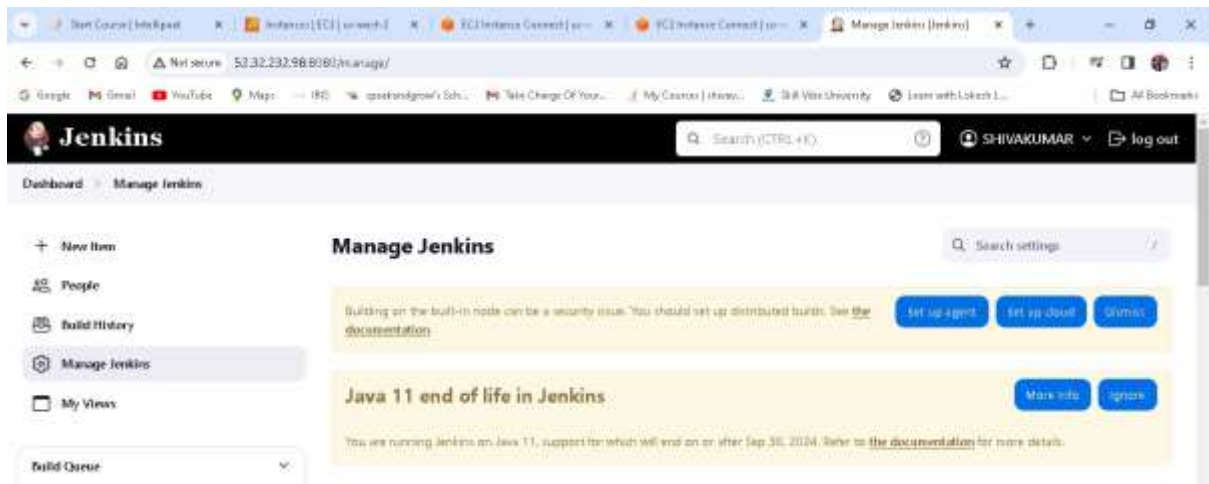


21. So far command line used.

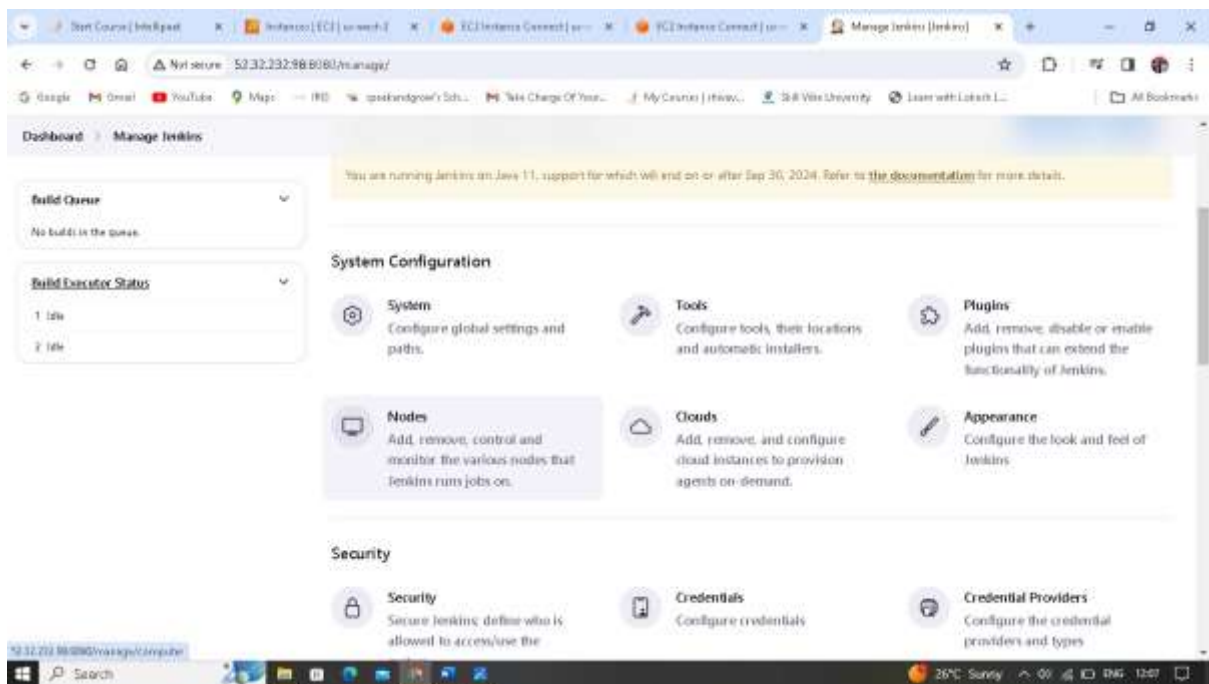


22. Now connect Jenkins-Node1. Click on Manage Jenkins

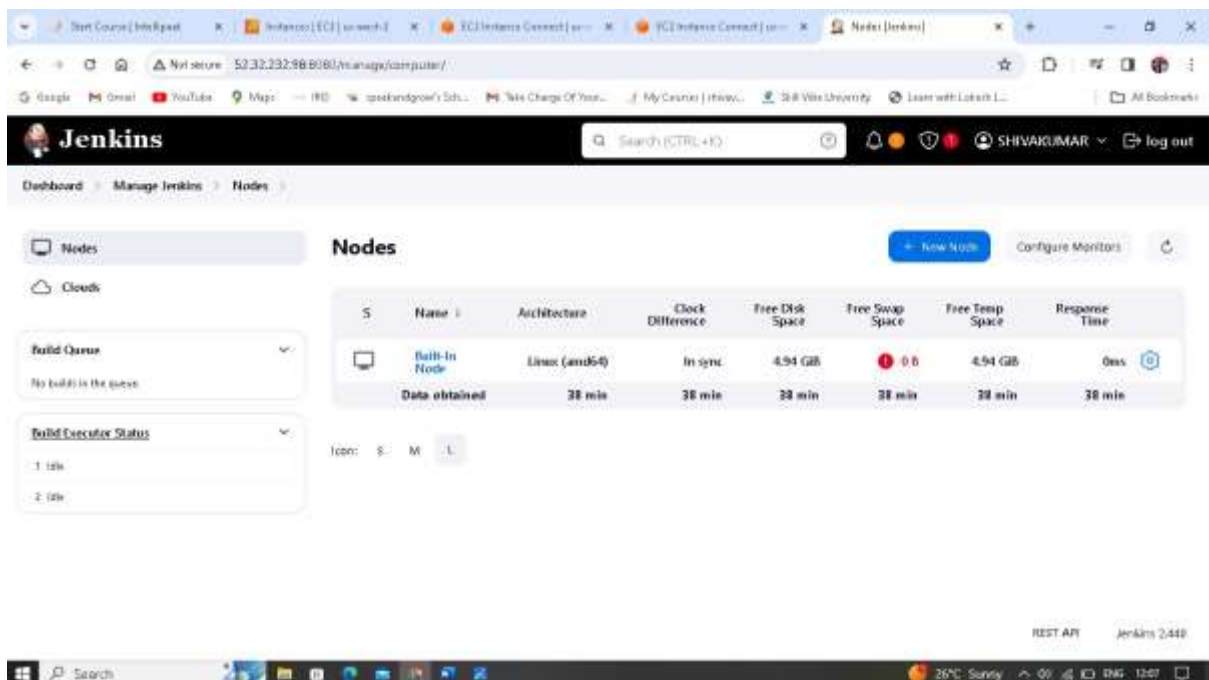




23. Go to system configuration and click on Nodes.



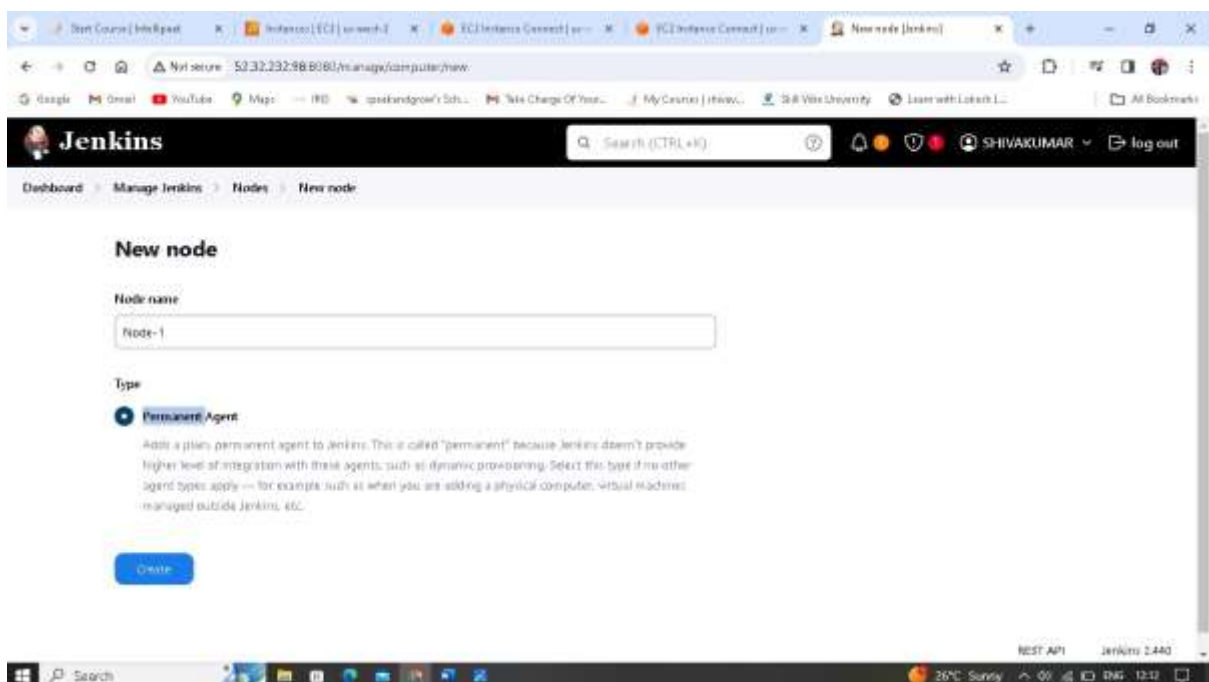
## 24. Click on +New Node



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for SHIVAKUMAR. The main content area is titled 'Nodes' and features a '+ New Node' button. Below this, there is a table listing the current nodes. The table has columns for Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The only node listed is 'Built-In Node' with a Linux (amd64) architecture. To the left of the table, there are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing two idle executors).

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.94 GB	0.6	4.94 GB	0ms
	Data obtained	38 min	38 min	38 min	38 min	38 min	38 min

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



The screenshot shows the 'New node' configuration page in Jenkins. The 'Node name' field is filled with 'Node-1'. Under the 'Type' section, 'Permanent Agent' is selected. A description explains that a permanent agent is a long-lived agent that Jenkins doesn't provide a higher level of integration with. At the bottom, there is a 'Create' button.

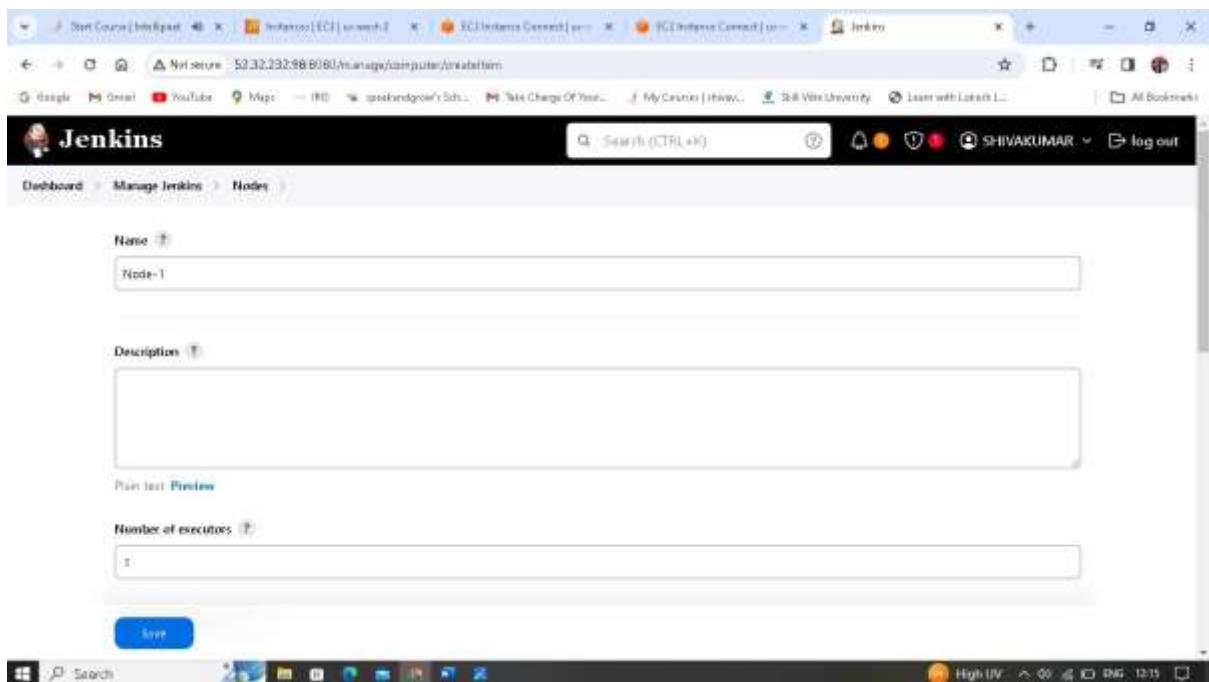
**New node**

Node name  
Node-1

Type  
☒ Permanent Agent

Create

26. Now fill



The screenshot shows the Jenkins web interface for configuring a new node. The browser tabs include 'Start Course | Intellipaat', 'Instances | EC2 | us-west-1', 'EC2 Instance Connect | us-west-1', 'Jenkins', and 'Nyl secure: 52.32.232.98:8080/instance/connect/instanceid'. The Jenkins header shows the user 'SHIVAKUMAR' and a 'log out' link. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes'. The configuration form has the following fields:

- Name:** Node-1
- Description:** (empty text area)
- Number of executors:** 1

A 'Save' button is located at the bottom of the form.



This screenshot shows the lower portion of the Jenkins node configuration form. The 'Remote root directory' field is empty, and a red error message below it states 'Remote directory is mandatory'. The 'Labels' field is empty. The 'Usage' dropdown menu is set to 'Use this node as much as possible'. The 'Launch method' dropdown menu is set to 'Launch agent by connecting it to the controller'. A 'Save' button is at the bottom.

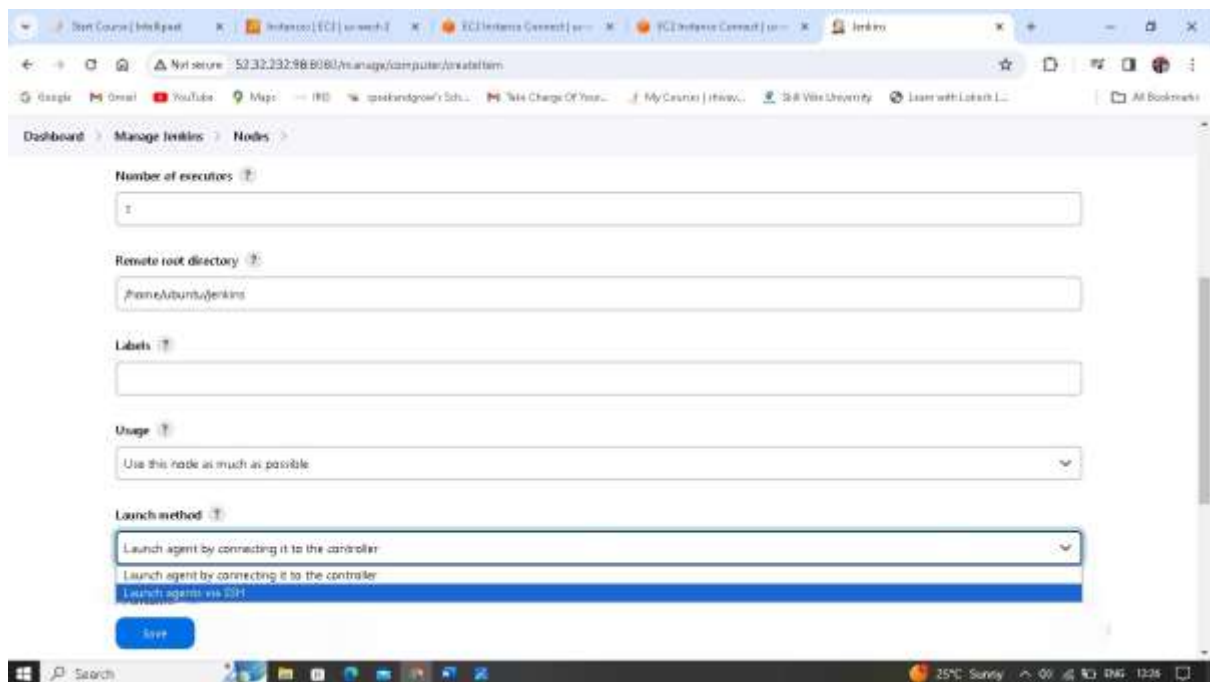
27. First of all give the Root directory. As to connect Jenkins-Node1 see the pwd (/home/ubuntu)



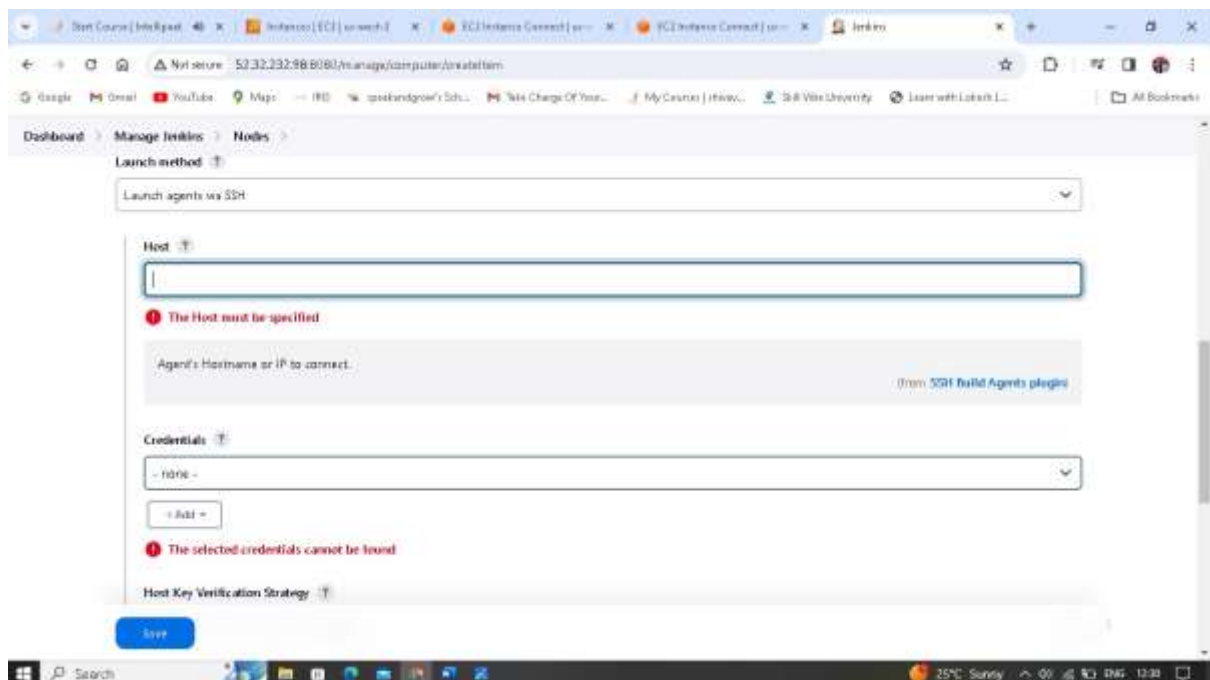
```
root@ubuntu:~# pwd
/home/ubuntu
```

i-0c134b10a01ddc44c (Jenkins- Node1)  
Public IP: 52.30.59.80 Private IP: 172.31.25.111

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



29. In Launch method select agents via SSH.





The screenshot shows a terminal window with the following output:

```

update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jshell to provide /usr/bin/jshell (jshell) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstack to provide /usr/bin/jstack (jstack) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstat to provide /usr/bin/jstat (jstat) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jstatd to provide /usr/bin/jstatd (jstatd) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/java to provide /usr/bin/java (java) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javac to provide /usr/bin/javac (javac) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/javadoc to provide /usr/bin/javadoc (javadoc) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jdb to provide /usr/bin/jdb (jdb) in auto mode
update-alternatives: using /usr/lib/jvm/java-11-openjdk-amd64/bin/jconsole to provide /usr/bin/jconsole (jconsole) in auto mode
Setting up openjdk-11-jdk-amd64 (11.0.21+9-0ubuntu1~22.04) ...
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 outside of systemd.

No VM guests are running outside of vmtoolsd.
amazon-172-31-18-21-1 [1]

```

The context menu is open over the terminal, showing the following options:

- Copy
- Copy link to highlight
- Go to 172.31.18.21
- Print...
- Translate selection to English
- Open in reading mode
- Get image description from Google
- Inspect

The terminal output shows the Jenkins master is running on port 8080.

Dashboard > Manage Jenkins > Nodes >

Launch method

Launch agents via SSH

Host

172.31.18.21

Agent's Hostname or IP to connect. [\(from SSH Build Agents plugin\)](#)

Credentials

- none -

Jenkins Credentials Provider

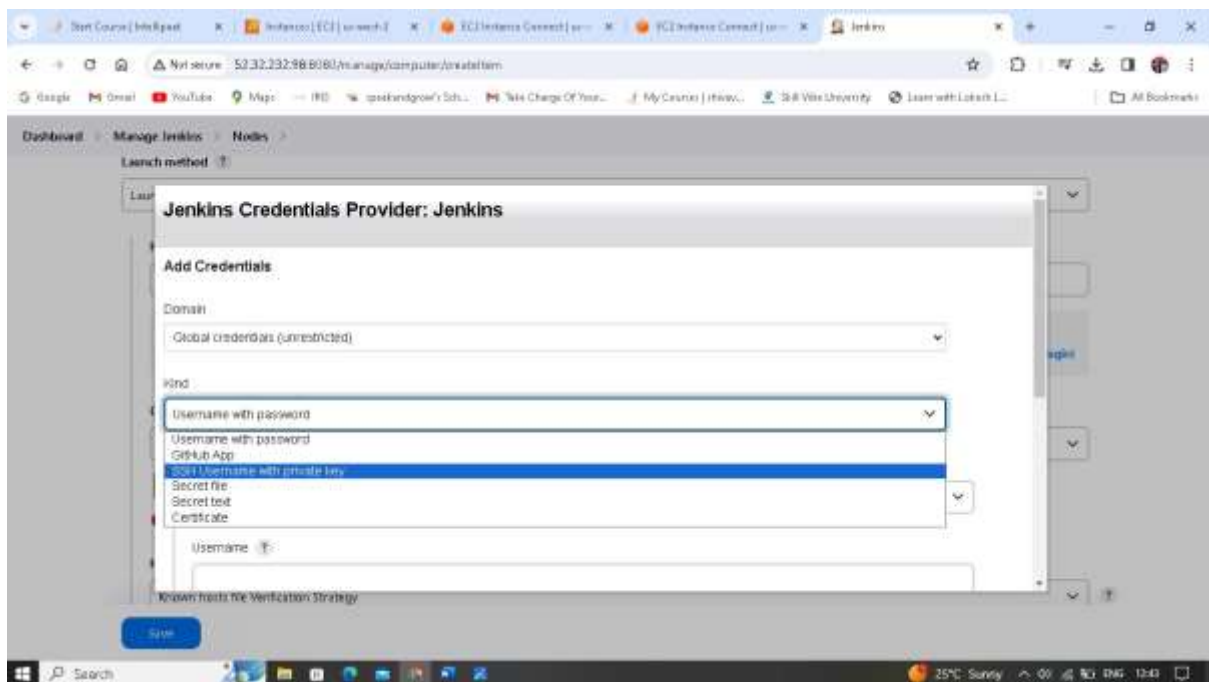
Credentials cannot be found

Host Key Verification Strategy

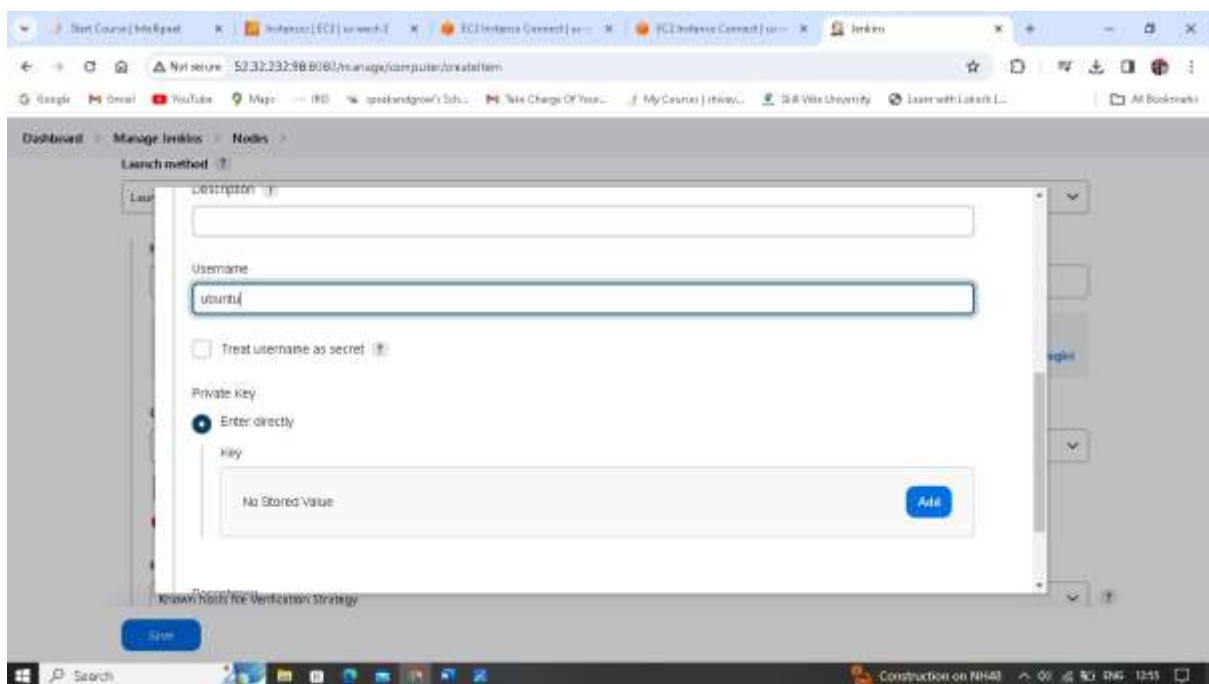
Known hosts file Verification Strategy

Save

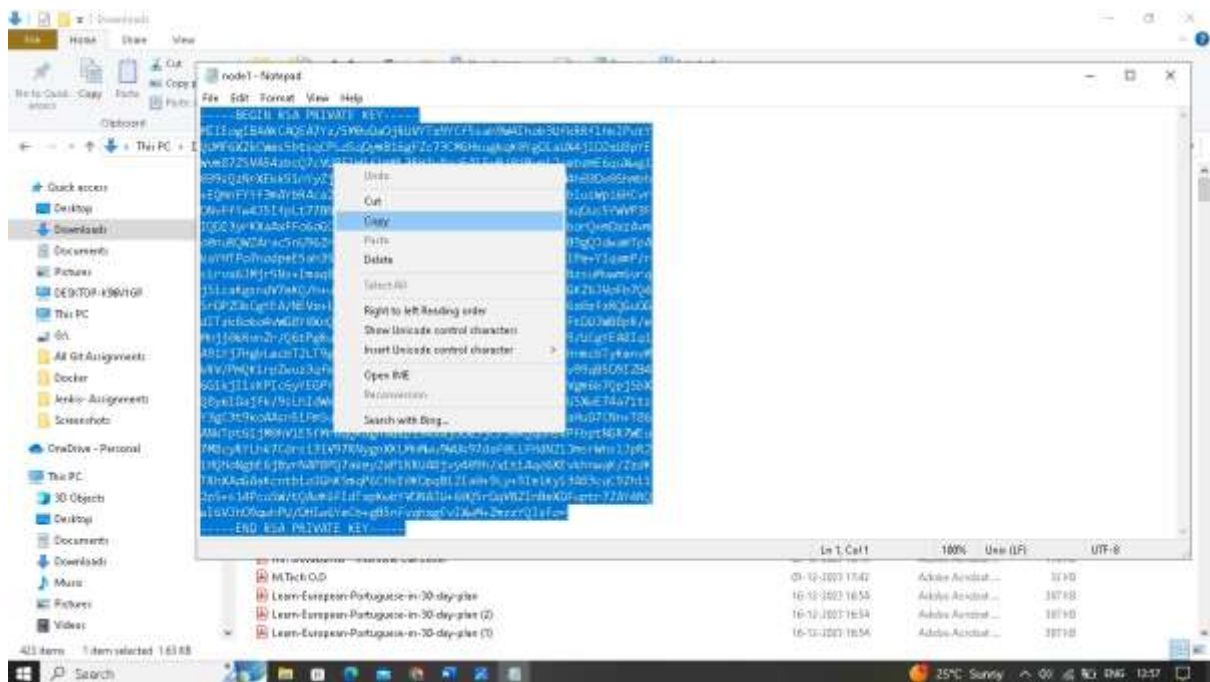
32. In kind select SSH Username with private key



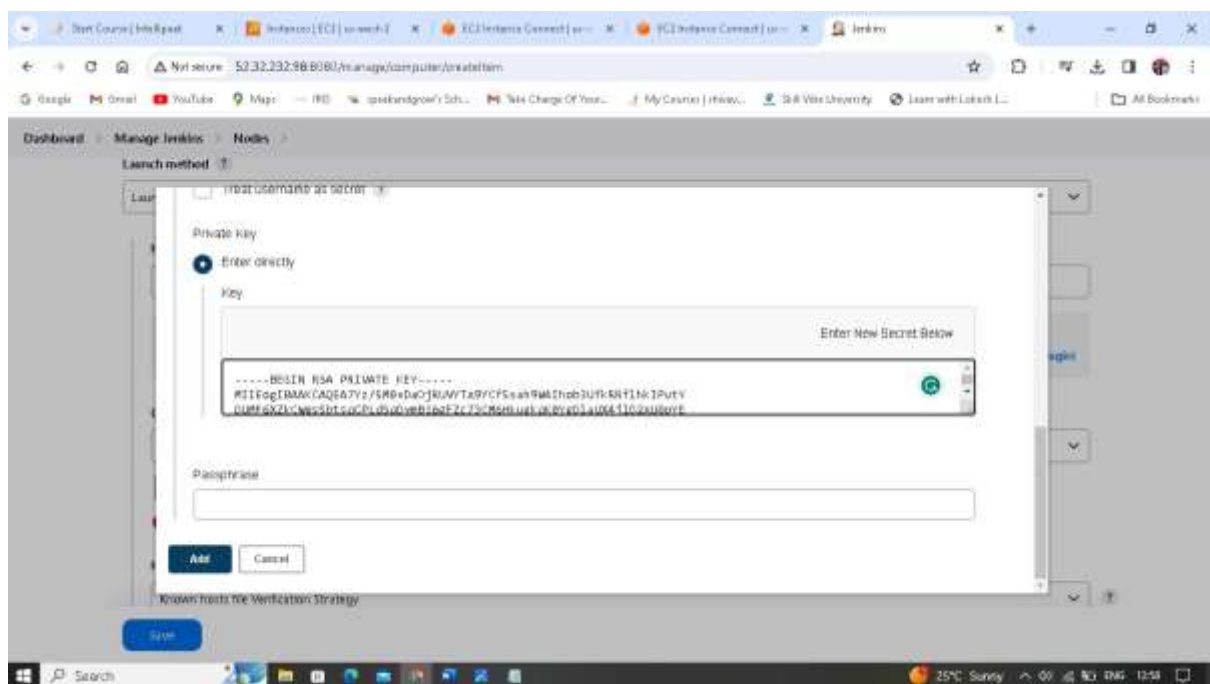
33. Give username as ubuntu click on enter directory and add



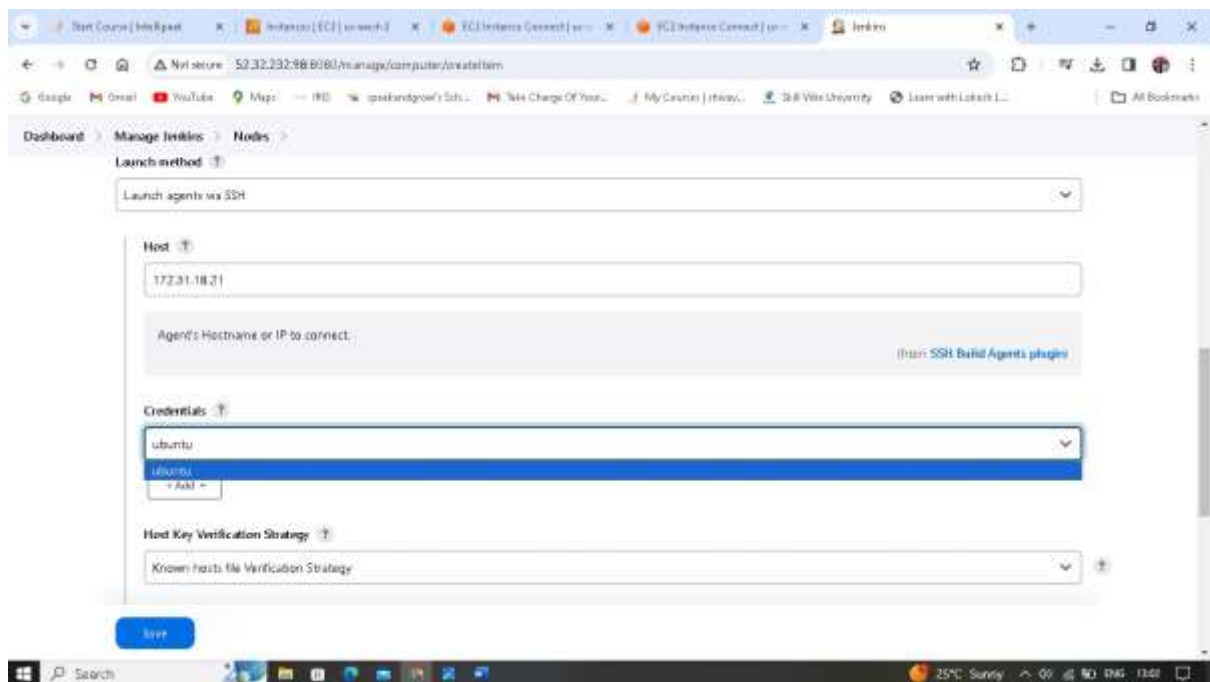
### 34. Copy the Jenkins-Node1 .pem file



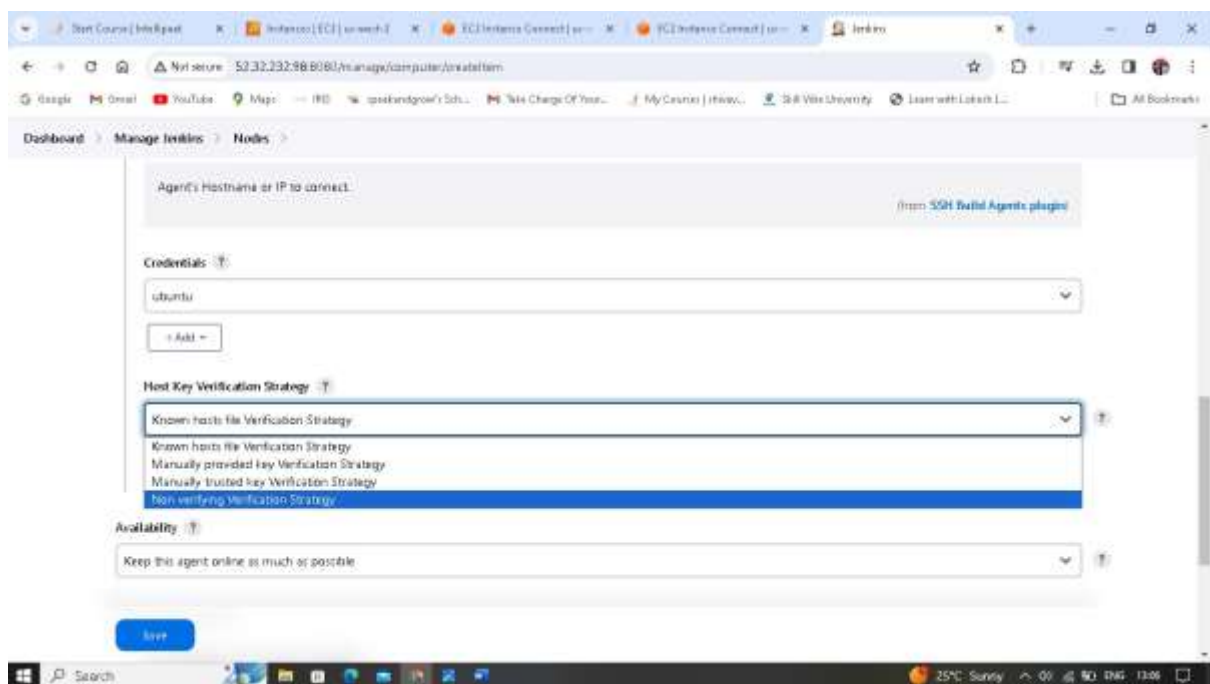
### 35. Paste the private key and click on add



### 36. Now in the credentials select ubuntu

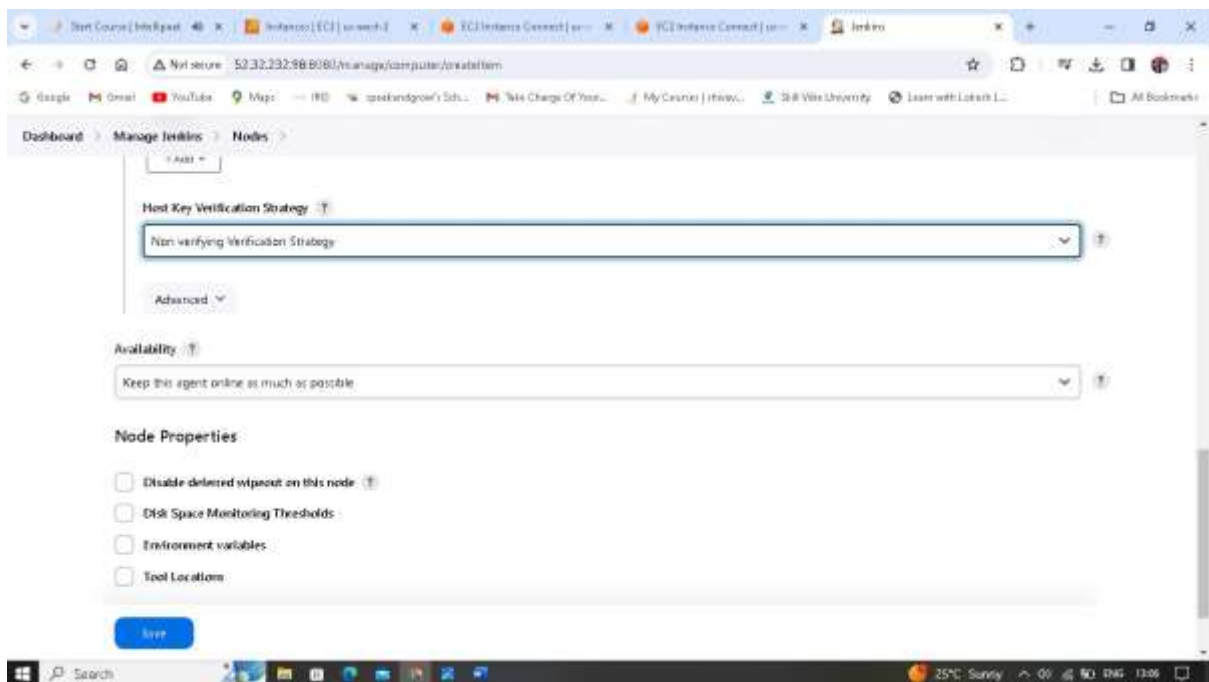


### 37. Select Non-verification strategy

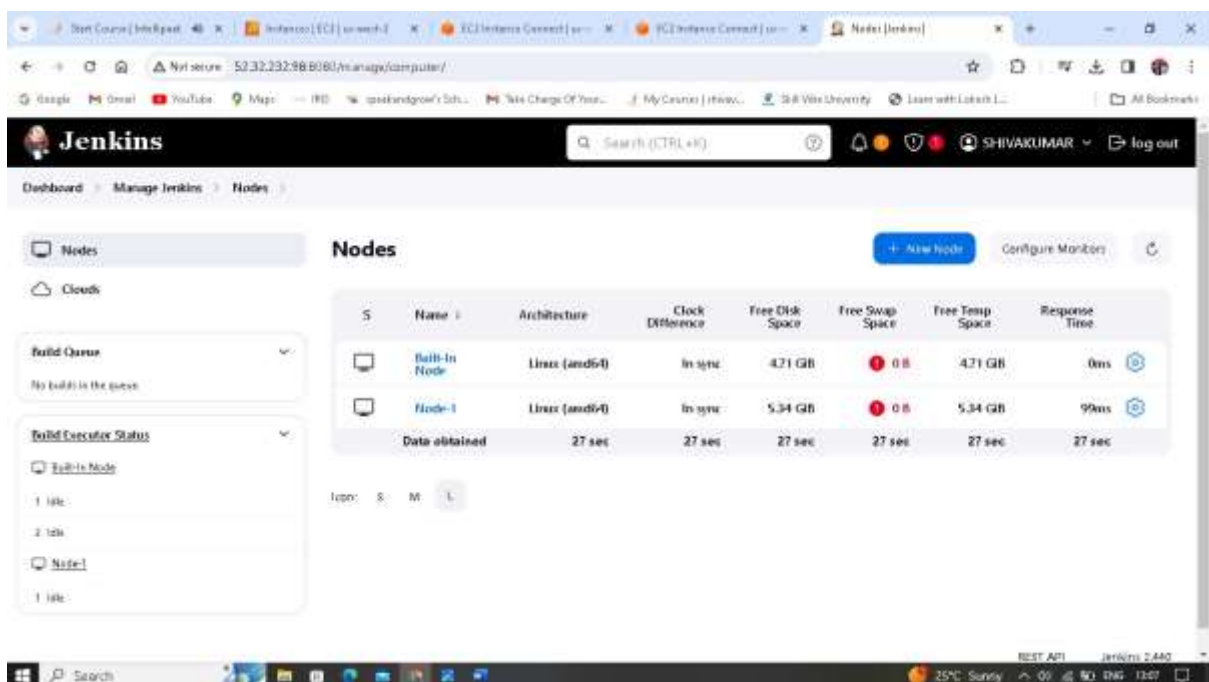




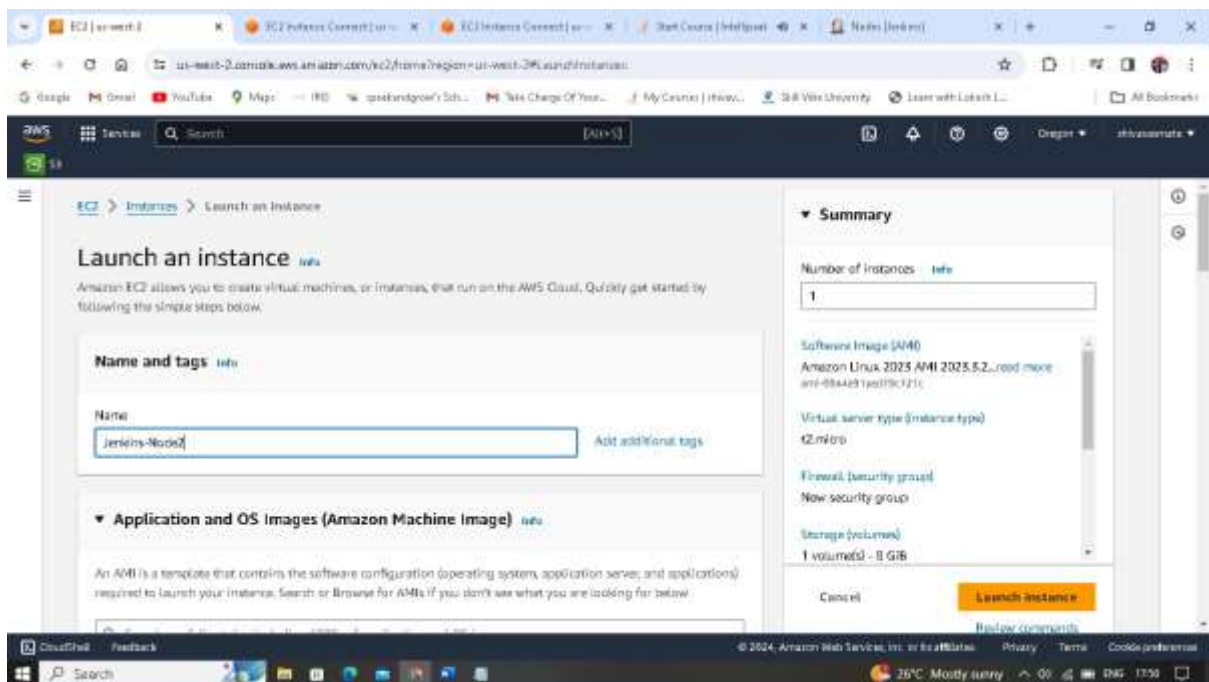
38. Click on save



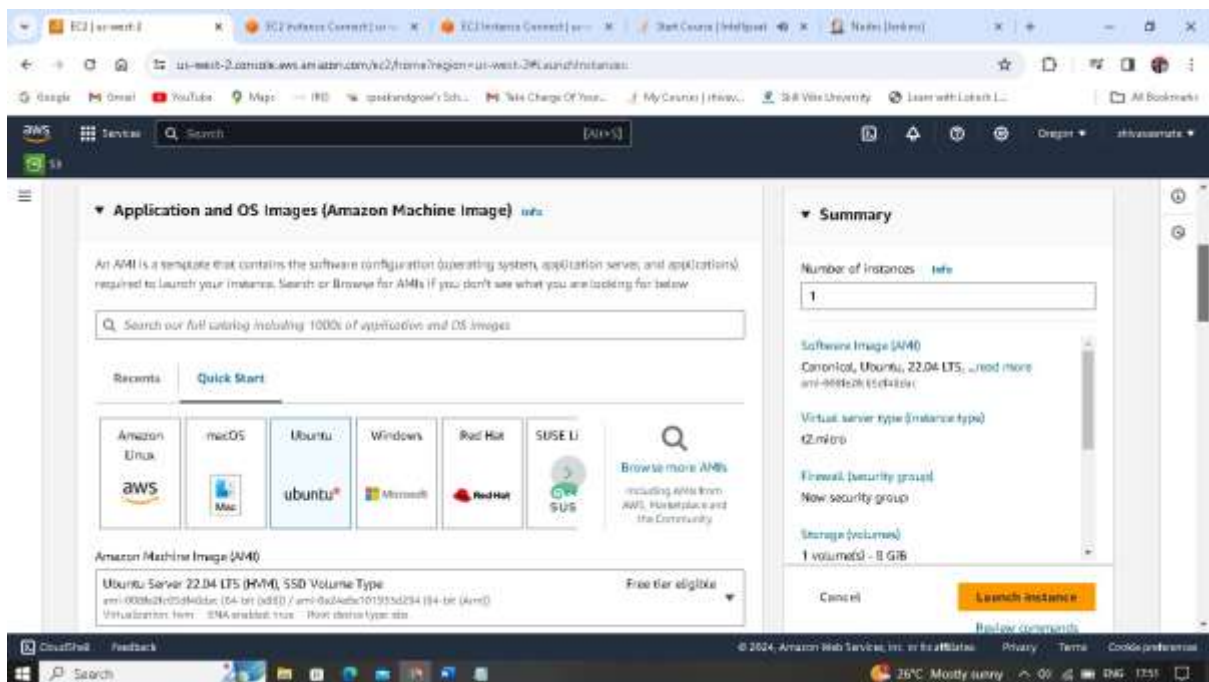
39. Successfully added Node-1



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



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



EC2 | us-west-2

us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#LaunchInstances

Instance type

Instance type

t2.micro

Family: t2 - 1 vCPU - 1 GB Memory - Current generation Linux

On-Demand Linux base pricing: 0.0116 USD per hour

On-Demand Linux base pricing: 0.0116 USD per hour

On-Demand Windows base pricing: 0.0182 USD per hour

On-Demand RHEL base pricing: 0.0316 USD per hour

Free tier 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

Summary

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-00000000 (5443434)

Virtual server type (instance type)

t2.micro

Firewall (security group)

Now security group

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

Launch instance

Review components

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

25°C Mostly sunny 17:51

EC2 | us-west-2

us-west-2.console.aws.amazon.com/ec2/home?region=us-west-2#LaunchInstances

Network settings

Network

vpc-0a610512c18573ee4

Subnet

No preference (Default subnet in any availability zone)

Auto-assign public IP

Enable

Firewall (security group)

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

Create security group

Select existing security group

Content security groups

Select security groups

default sg-00c8b9000c1a145da

VPC: vpc-0a610512c18573ee4

Compare security group rules

Summary

Number of instances

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-00000000 (5443434)

Virtual server type (instance type)

t2.micro

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 8 GiB

Cancel

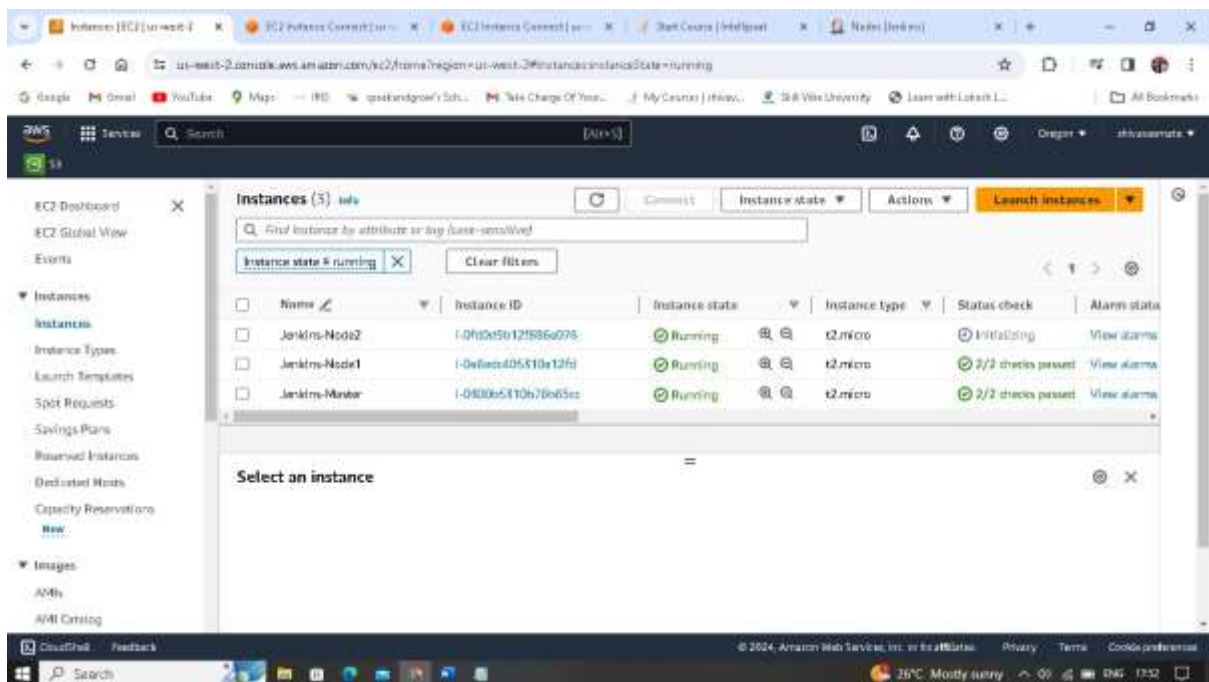
Launch instance

Review components

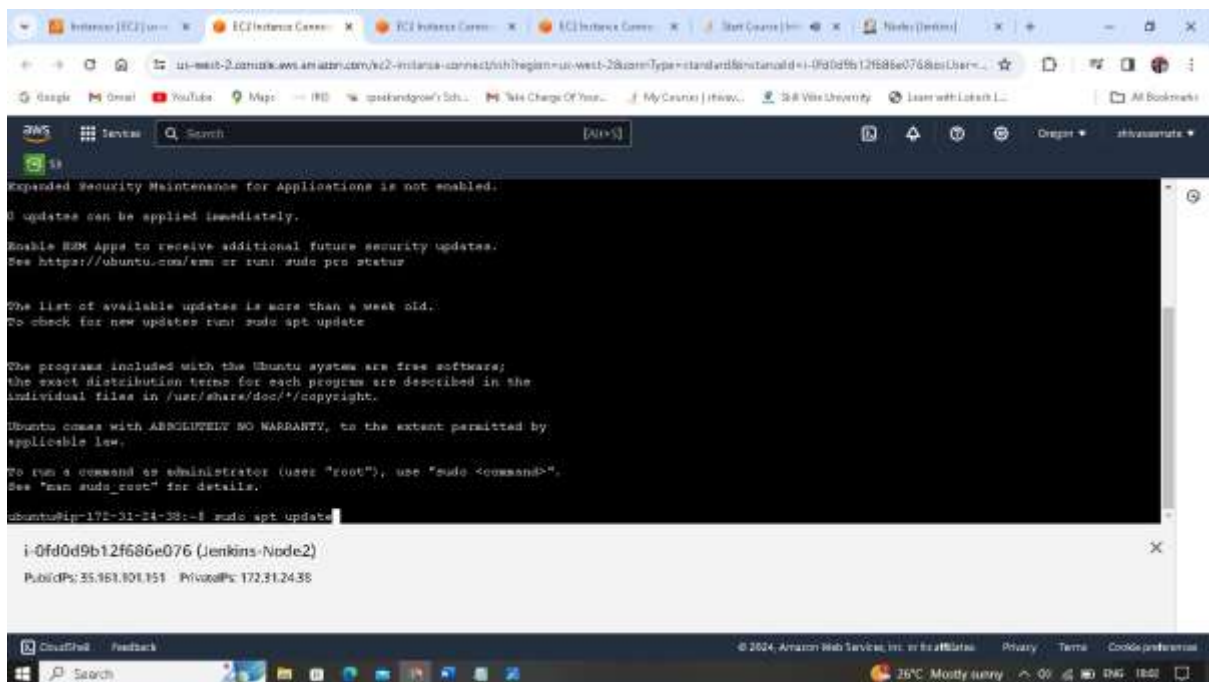
© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

25°C Mostly sunny 17:51

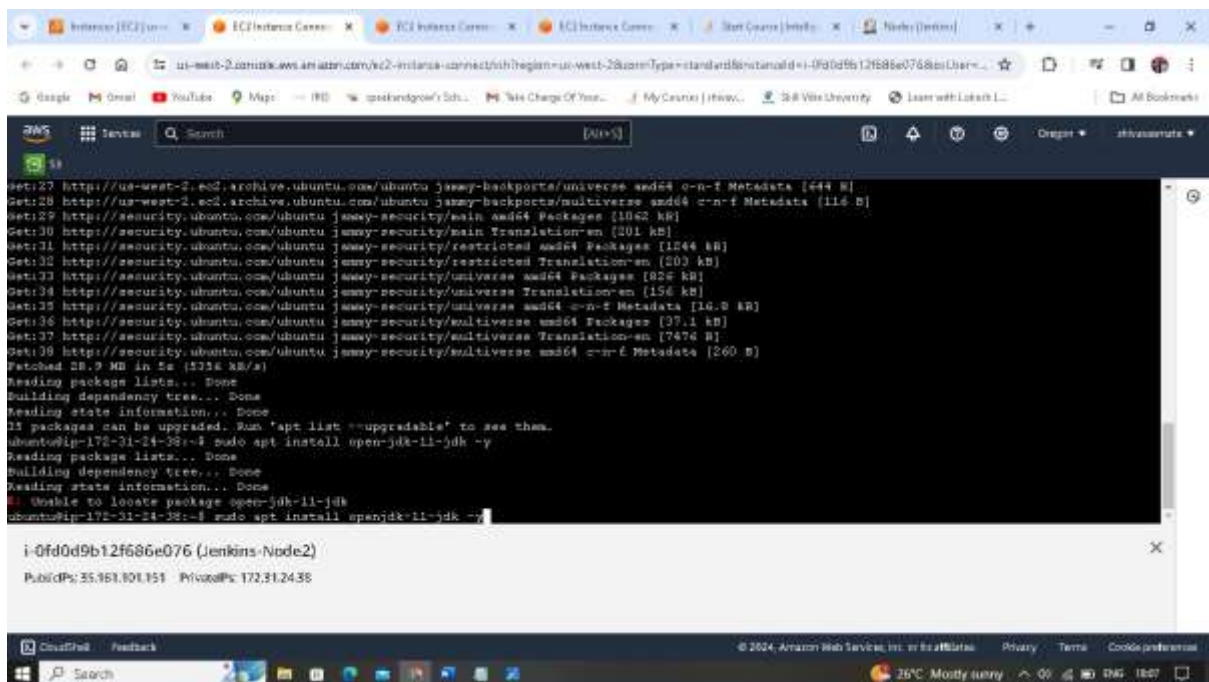
42. Now Jenkins-Node2 is also running status.



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

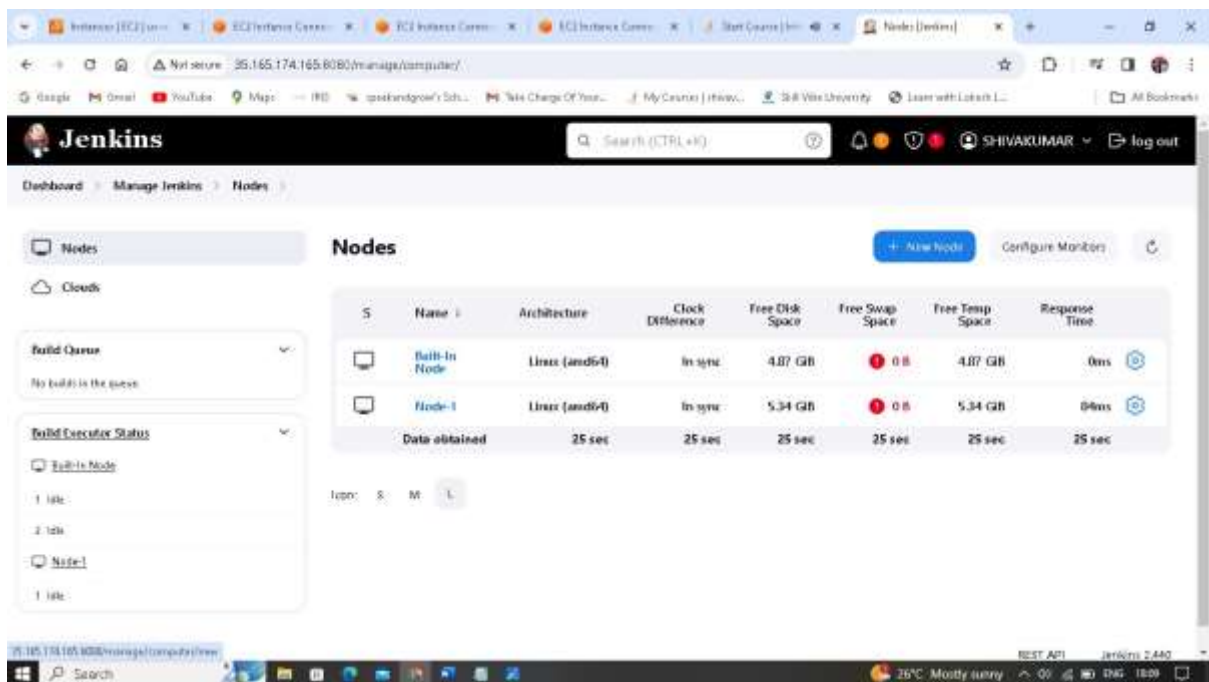


#### 44. Install java by using `sudo apt install openjdk-11-jdk -y`



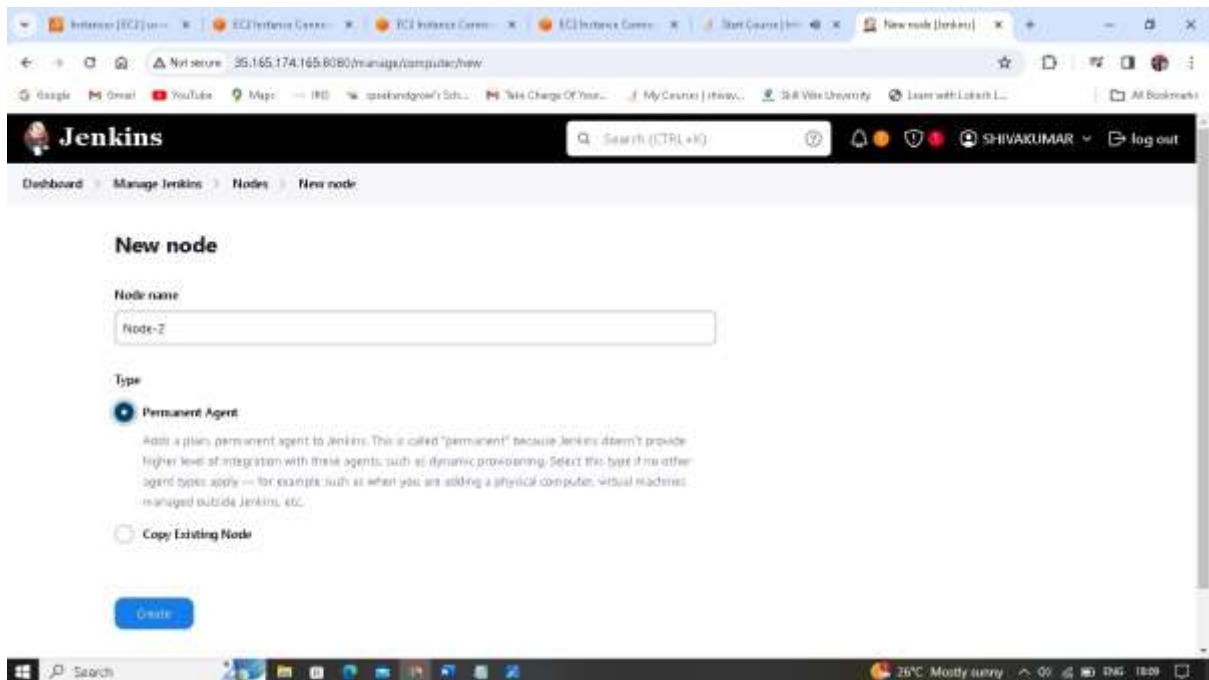
```
Get:27 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/universe amd64 o-n-f Metadata [444 B]
Get:28 http://us-west-2.ec2.archive.ubuntu.com/ubuntu jessy-backports/multiverse amd64 o-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu jessy-security/main amd64 Packages [1062 kB]
Get:30 http://security.ubuntu.com/ubuntu jessy-security/main Translation-en [201 kB]
Get:31 http://security.ubuntu.com/ubuntu jessy-security/restricted 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 [928 kB]
Get:34 http://security.ubuntu.com/ubuntu jessy-security/universe Translation-en [156 kB]
Get:35 http://security.ubuntu.com/ubuntu jessy-security/universe amd64 o-n-f Metadata [16.0 kB]
Get:36 http://security.ubuntu.com/ubuntu jessy-security/multiverse amd64 Packages [39.1 kB]
Get:37 http://security.ubuntu.com/ubuntu jessy-security/multiverse Translation-en [7474 B]
Get:38 http://security.ubuntu.com/ubuntu jessy-security/multiverse amd64 o-n-f Metadata [260 B]
Fetched 28.9 MB in 5s (5336 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
11 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
E: Unable to locate package openjdk-11-jdk
ubuntu@ip-172-31-24-38:~$ 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.

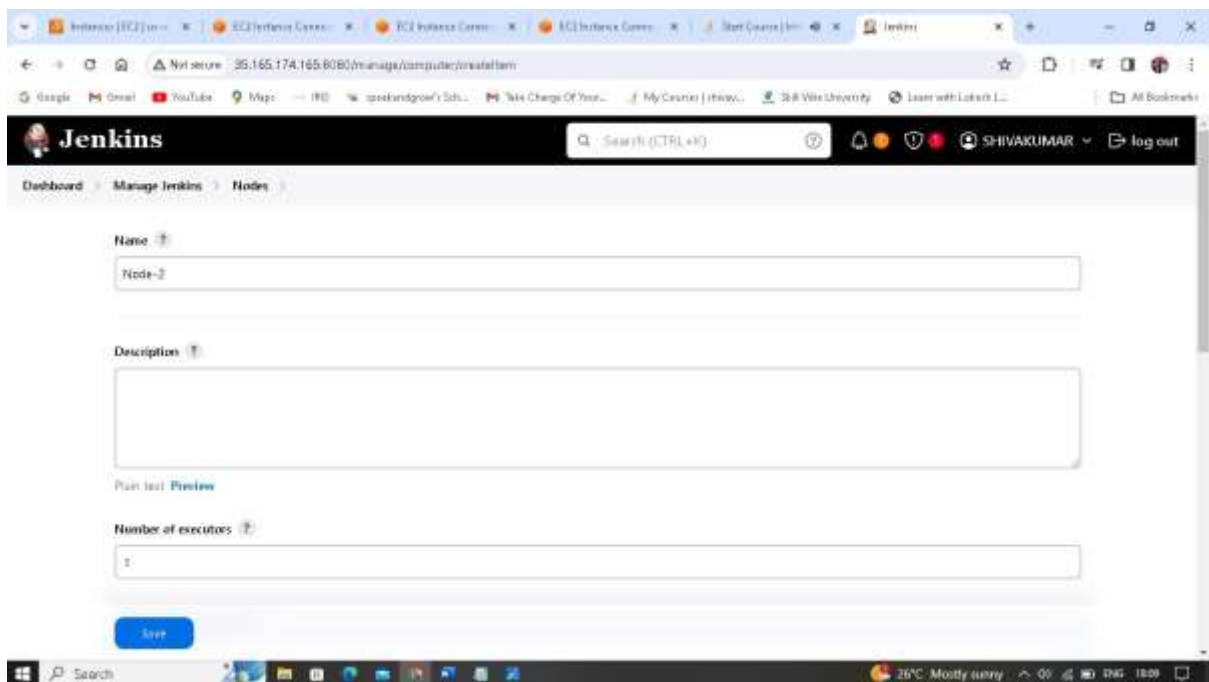


The screenshot shows the Jenkins 'New node' configuration page. The browser's address bar displays 'Nxt secure: 35.165.174.165:8080/manage/computer/new'. The Jenkins header includes a search bar and the user 'SHIVAKUMAR' with a 'log out' link. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes > New node'. The form is titled 'New node' and contains the following fields:

- Node name:** A text input field containing 'Node-2'.
- Type:** Two radio button options:
  - Permanent Agent:** Selected. Below it, a description reads: 'Add a jenkins permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply --- for example such as when you are adding a physical computer, Virtual Machines managed outside Jenkins, etc.'
  - Copy Existing Node:** Unselected.

A blue 'Create' button is located at the bottom of the form.

47. fill it

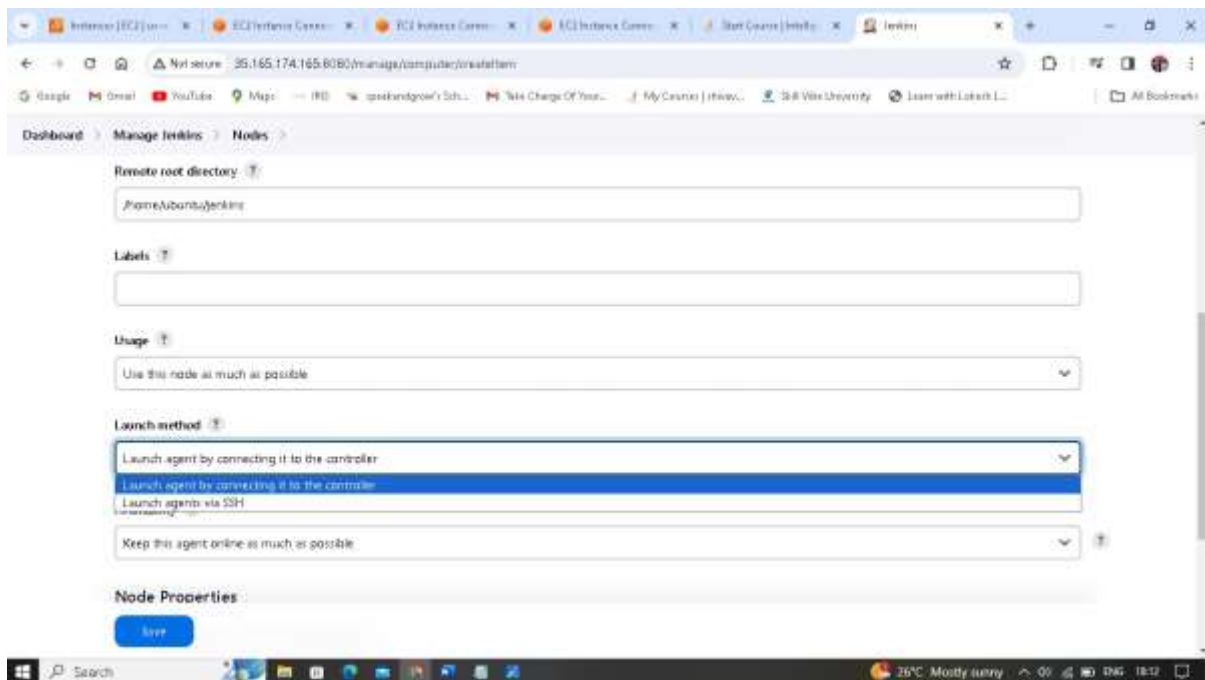


The screenshot shows the Jenkins 'Node configuration' page for 'Node-2'. The browser's address bar displays 'Nxt secure: 35.165.174.165:8080/manage/computer/new/edit'. The Jenkins header is the same as in the previous screenshot. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes'. The form contains the following fields:

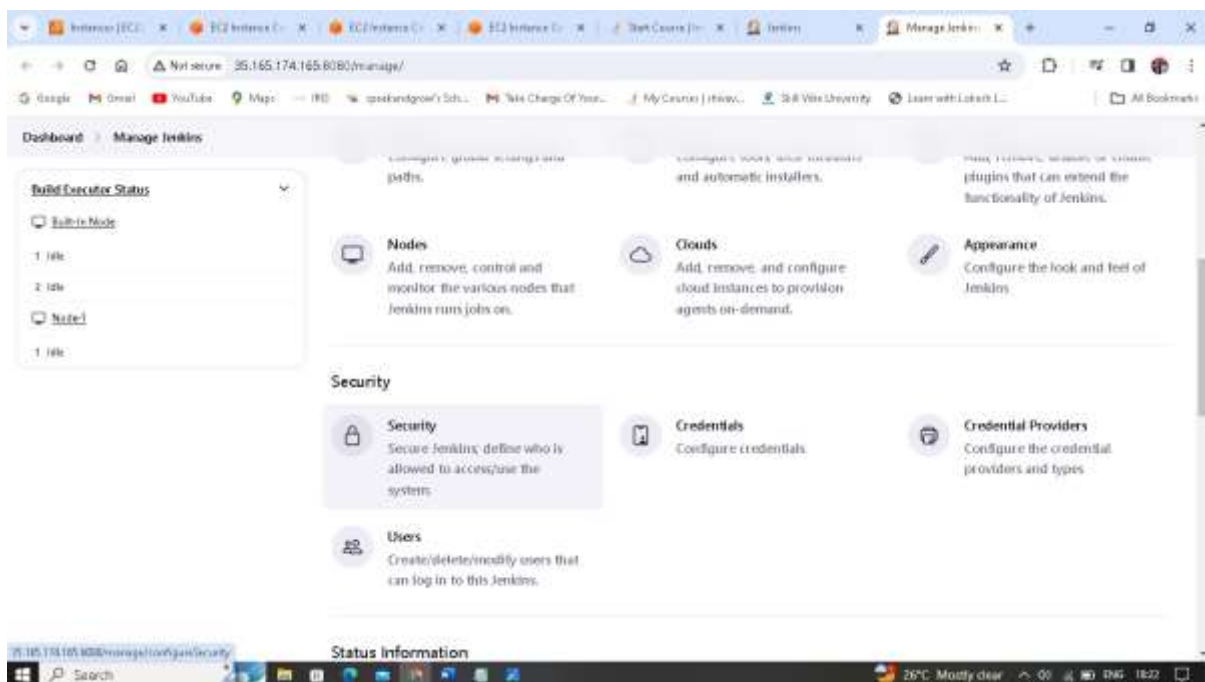
- Name:** A text input field containing 'Node-2'.
- Description:** A large text area, currently empty.
- Number of executors:** A text input field containing '1'.

Below the 'Description' field, there are links for 'Plain text' and 'Preview'. A blue 'Save' button is at the bottom of the form.

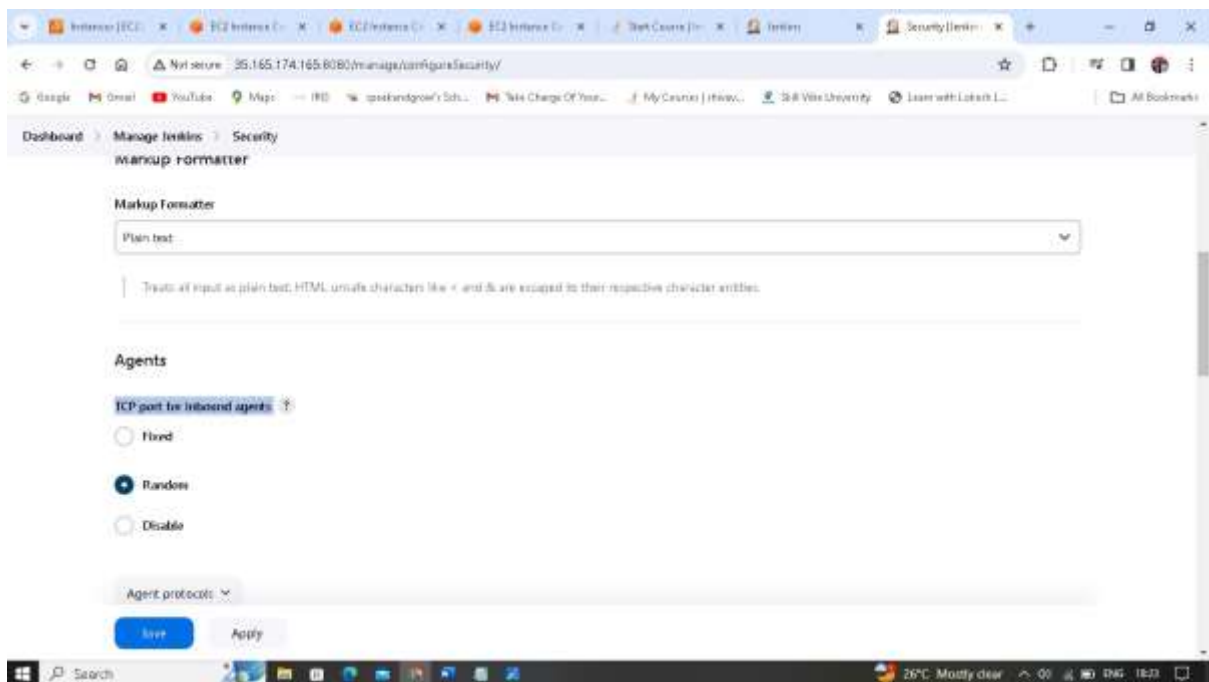
48. For root directory give /home/ubuntu/jenkins and In Launch method select Launch agent by connecting it to the controller.



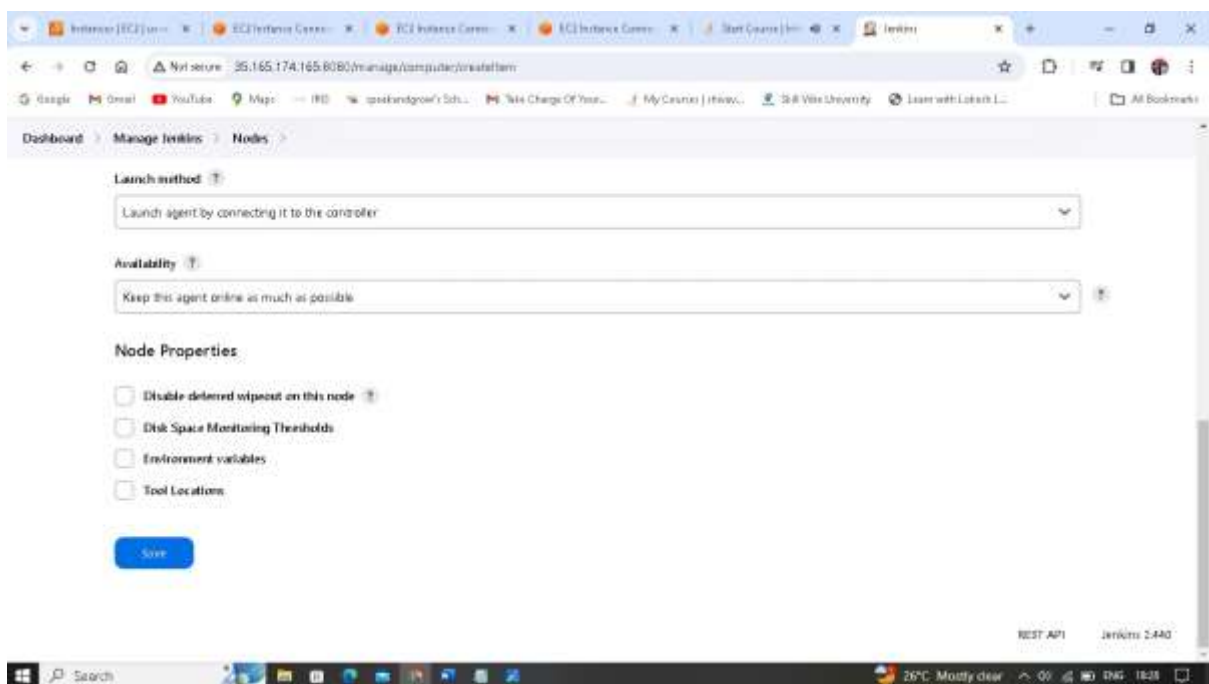
49. Now go to Manage Jenkins click on security



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

The screenshot shows the Jenkins 'Nodes' page. On the left sidebar, there are sections for 'Nodes', 'Clouds', 'Build Queue' (showing 'No builds in the queue'), and 'Build Executor Status' (showing 'Built-In Node' with 1 idle and 2 total executors, and 'Node-1' with 1 idle and 1 total executor). The main area is titled 'Nodes' and contains a table with the following data:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.87 GB	0 B	4.87 GB	0ms
	Node-1	Linux (amd64)	In sync	5.34 GB	0 B	5.34 GB	46ms
	Node-2		N/A	N/A	N/A	N/A	N/A
Data obtained		43 min	43 min	43 min	43 min	43 min	43 min

Below the table, there is a legend for icons: S (Success), M (Maintenance), and L (Offline). Node-2 is marked with a red 'X' icon, indicating it is offline.

53. Copy the command

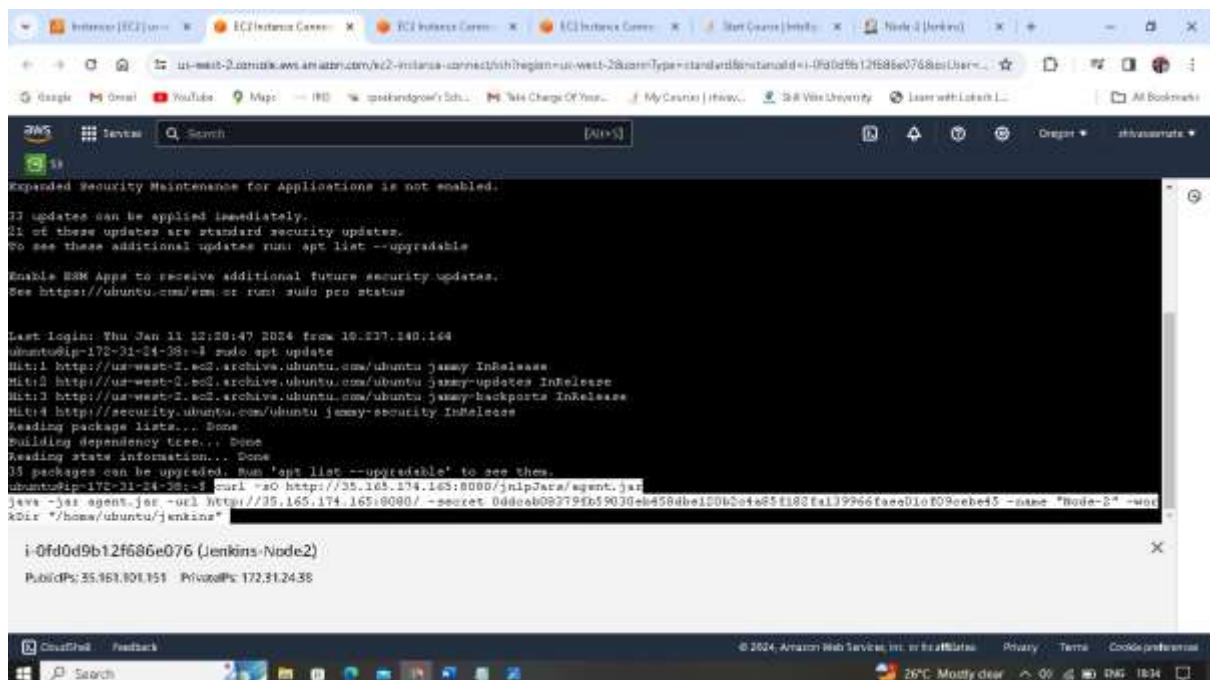
The screenshot shows the Jenkins 'Agent Node-2' configuration page. On the left sidebar, there are sections for 'States', 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', and 'Log'. The main area is titled 'Agent Node-2' and contains a section 'Run from agent command line: (Unix)' with the following command:

```
curl -XO https://35.165.174.165:8080/jenkins/agent.jar  
java -jar agent.jar -url http://35.165.174.165:8080/ -secret B55a6b83797b6982b9a168be12b2c4a8f7162fa13956fawad1c79cbe6d5 -name "Node-2" -workDir "/home/ubuntu/jenkins"
```

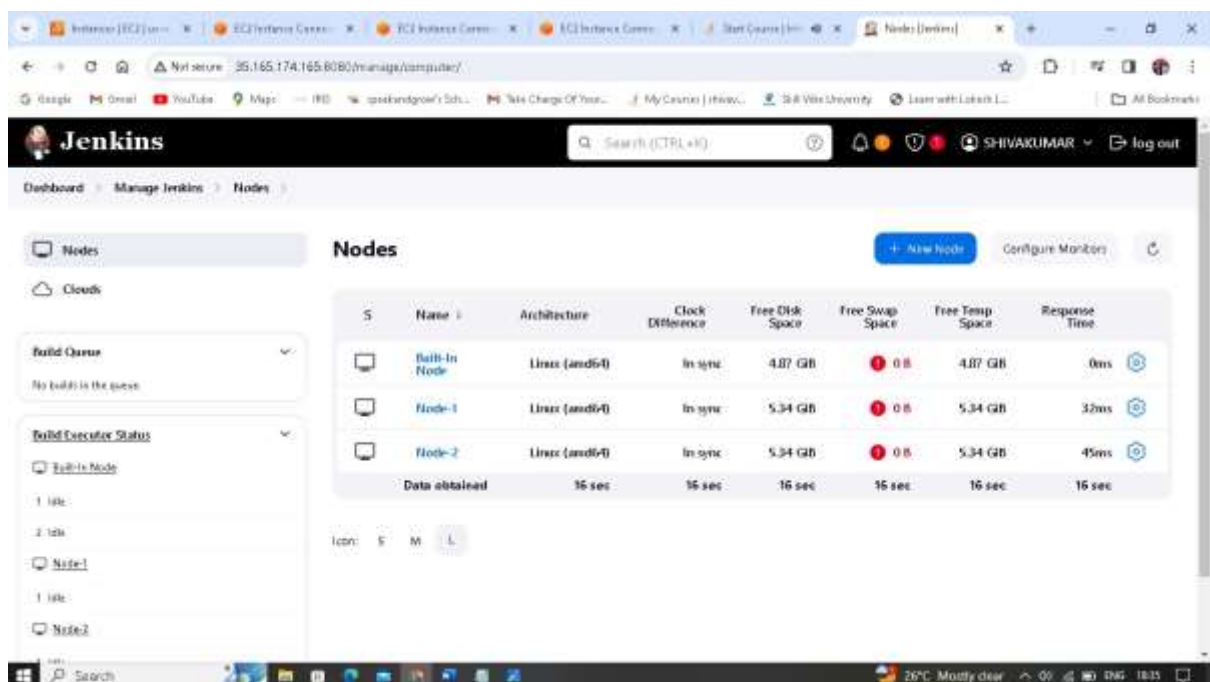
Below this, there is a section 'Run from agent command line: (Windows)' with the following command:

```
curl.exe -XO https://35.165.174.165:8080/jenkins/agent.jar  
java -jar agent.jar -url http://35.165.174.165:8080/ -secret B55a6b83797b6982b9a168be12b2c4a8f7162fa13956fawad1c79cbe6d5 -name "Node-2" -workDir "%home/ubuntu/jenkins"
```

## 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).

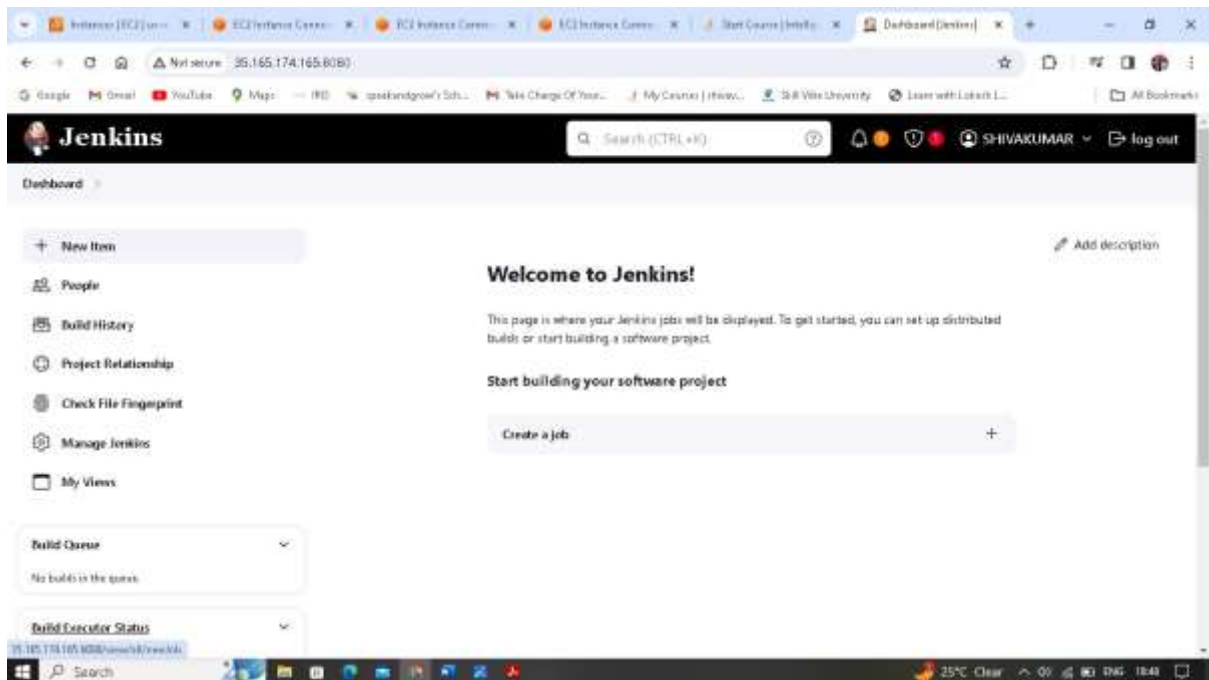
Job2 → master branch → Node-2 (prod)

Job3 → Job1-Job2

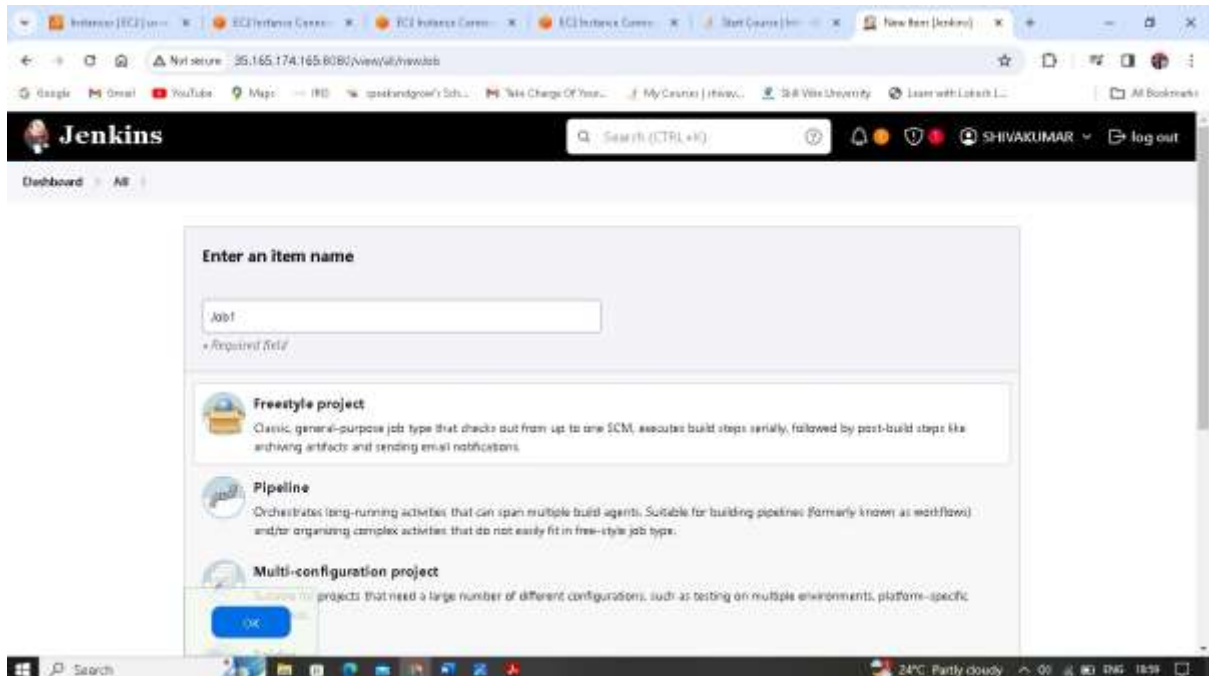


## 2. FREESYSTYLE PROJECT

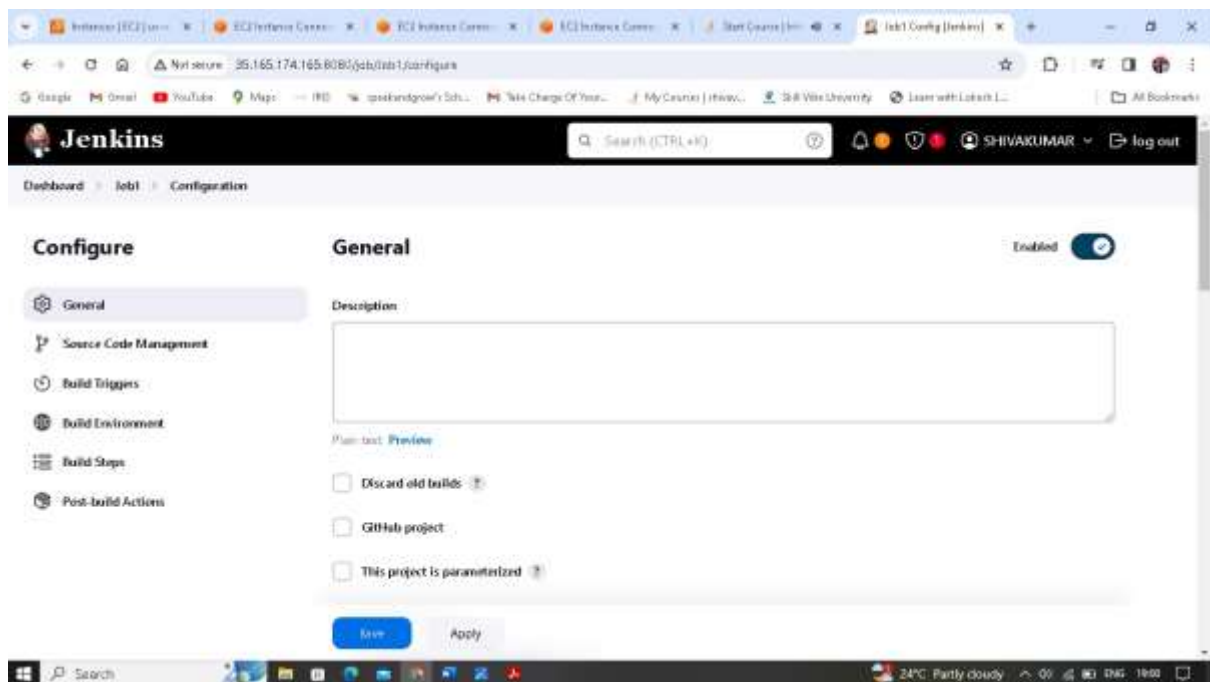
1. First go to Jenkins dashboard and click on New item



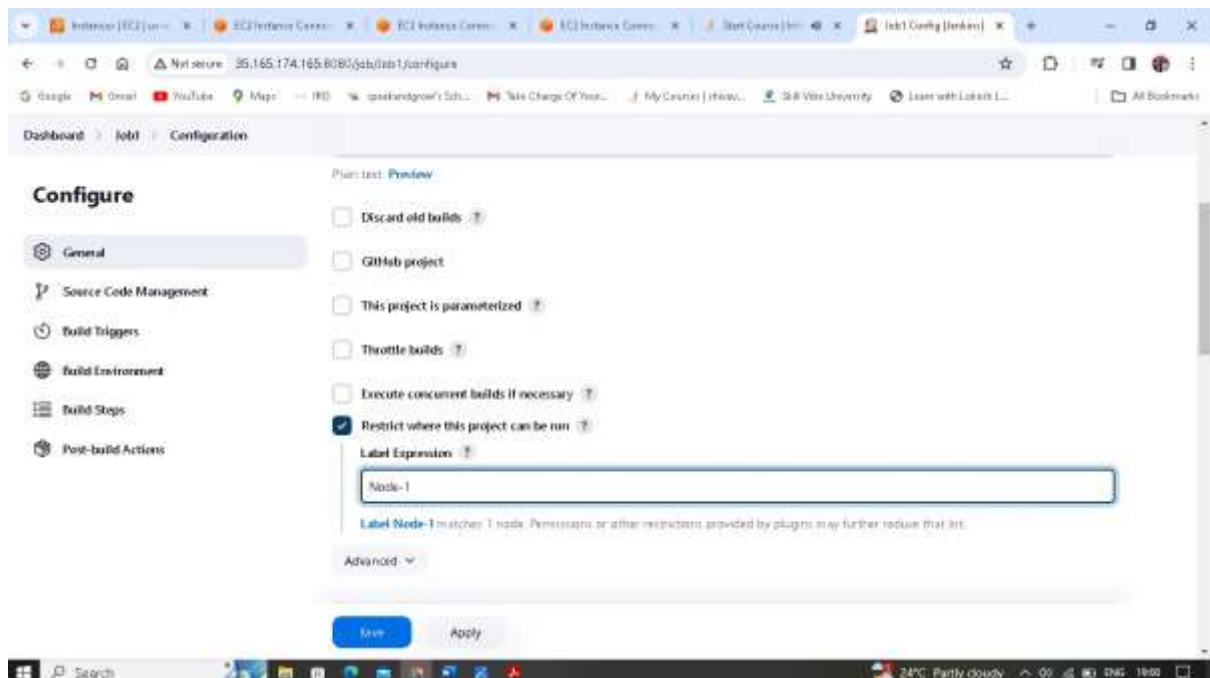
2. Enter an item name as Job1 choose Freestyle project and click on ok.



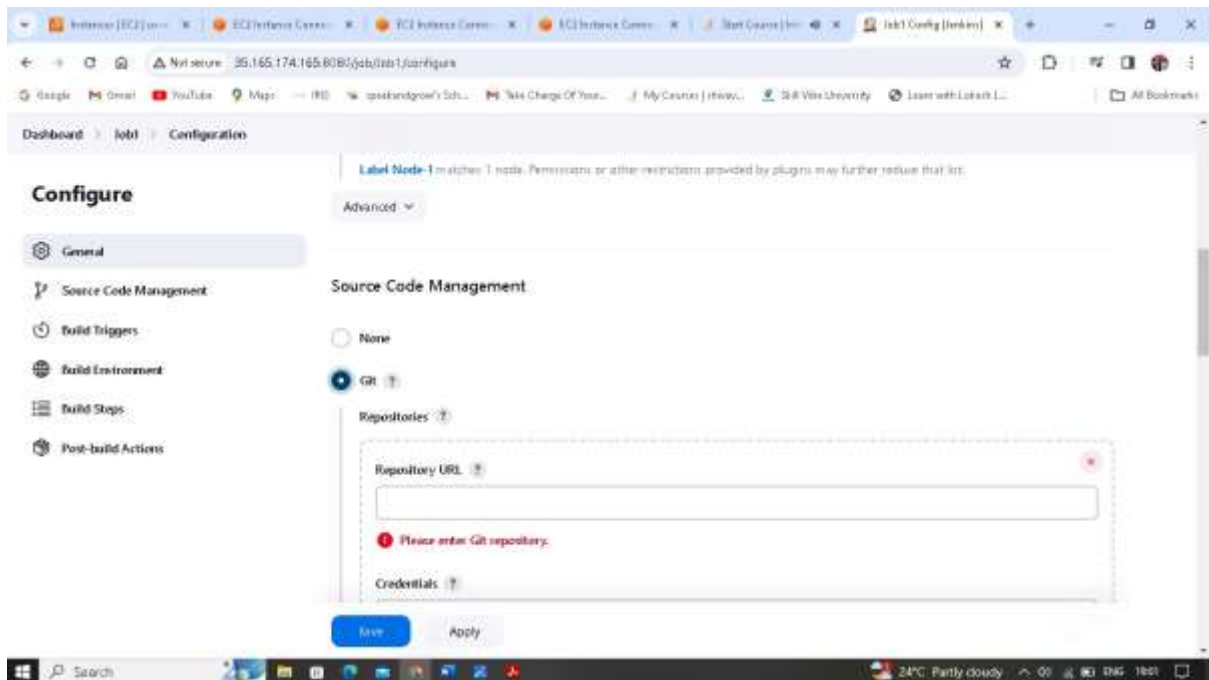
3. check it one by one and fill it



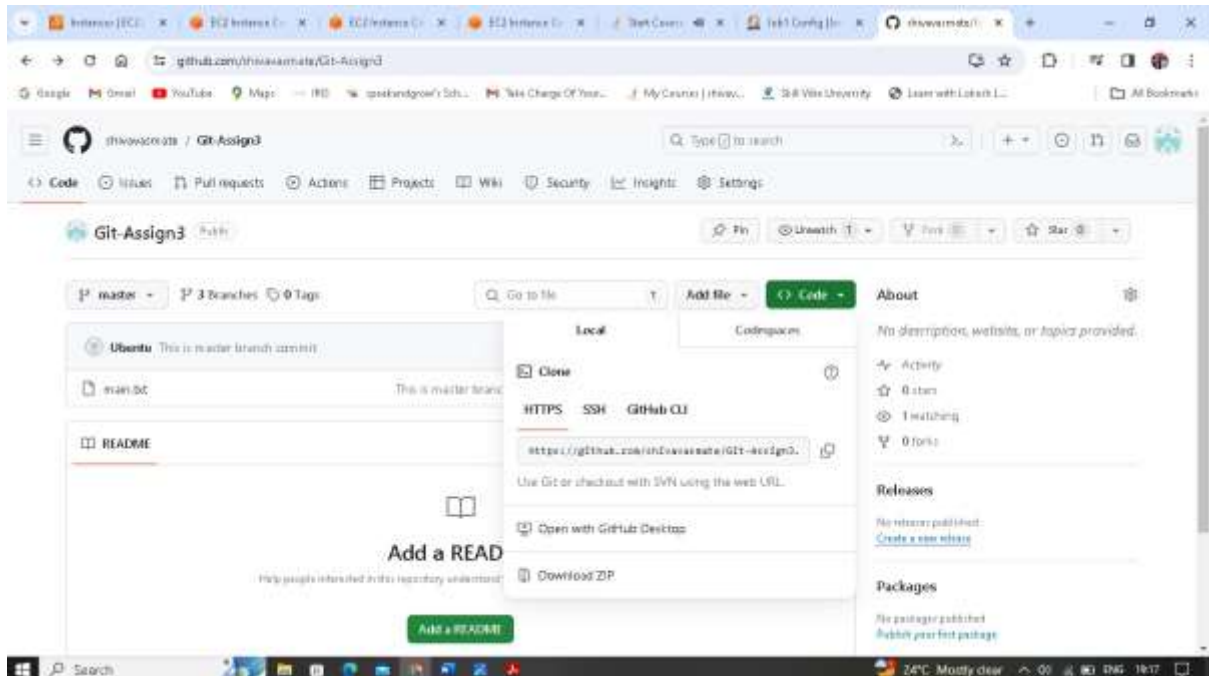
4. First of all check in Restrict where this project can be run. Label Expression as Node-1



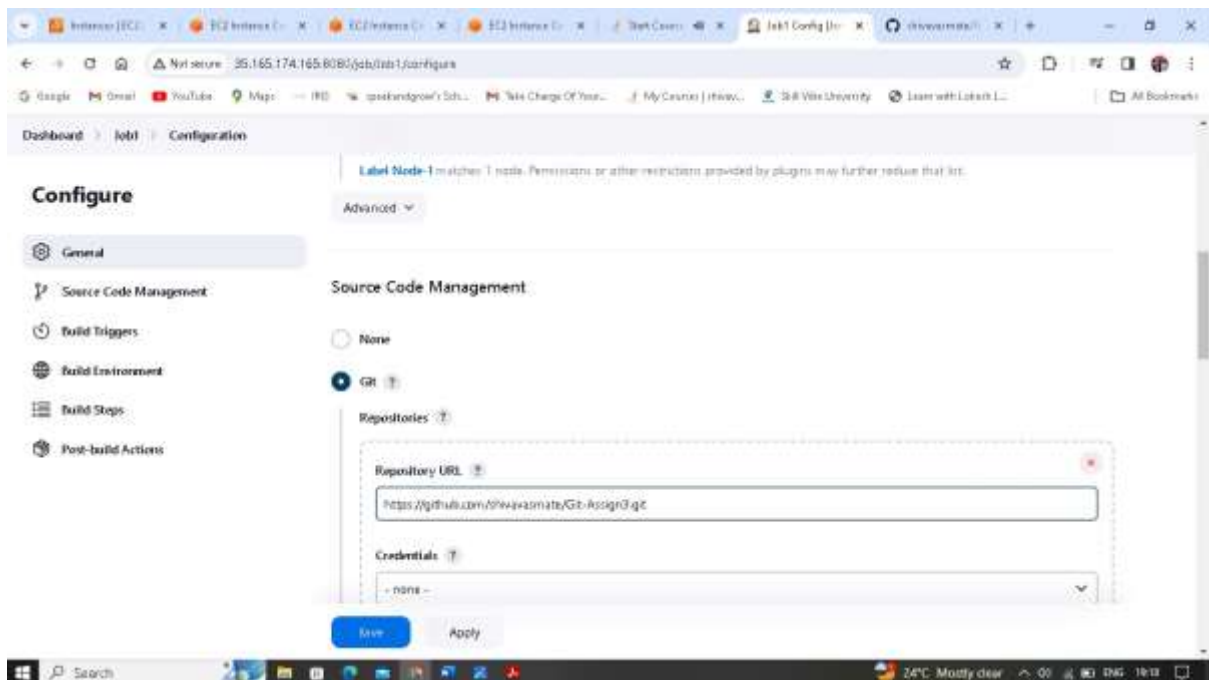
## 5. Under Source Code Management Check Git



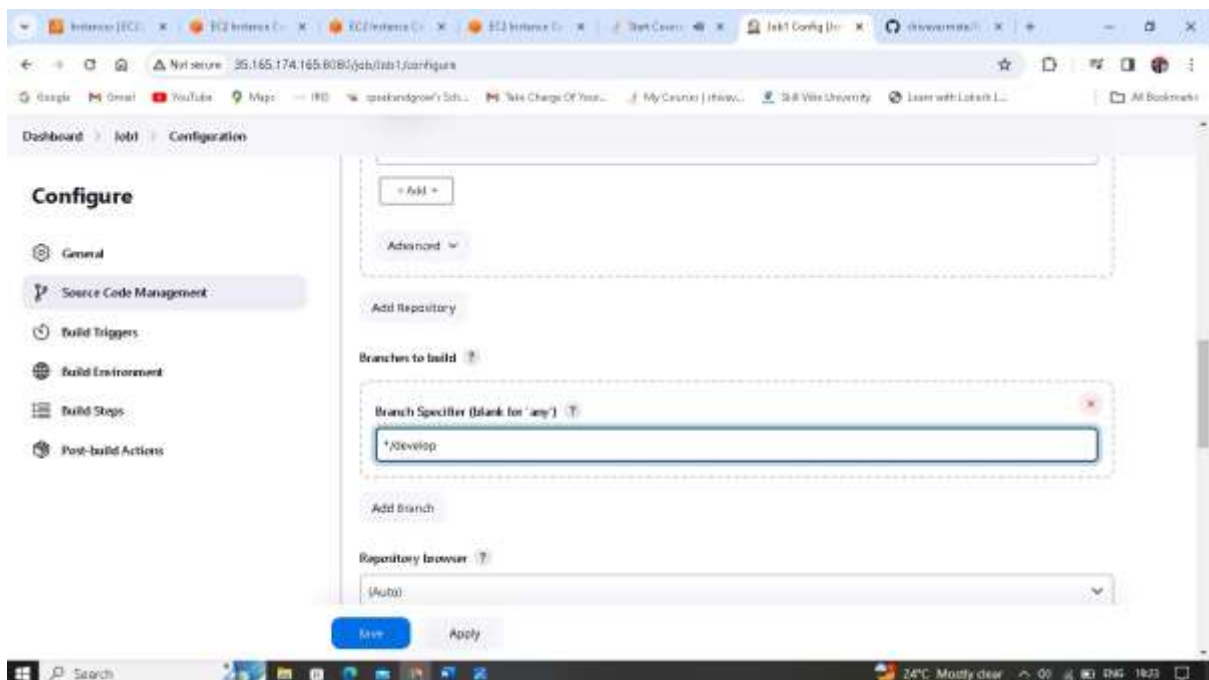
## 6. Now log in to your GitHub in another browser.



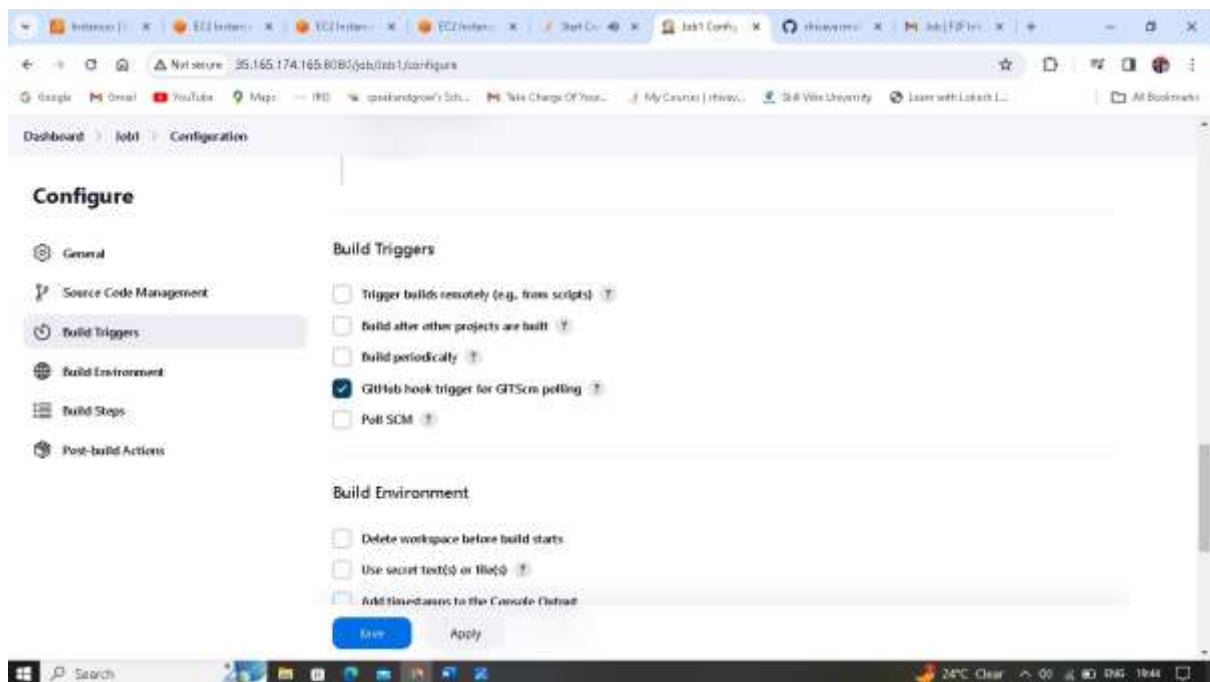
7. For Repository URL copy from code and choose the repo which has develop branch in it(Git-Assign3).



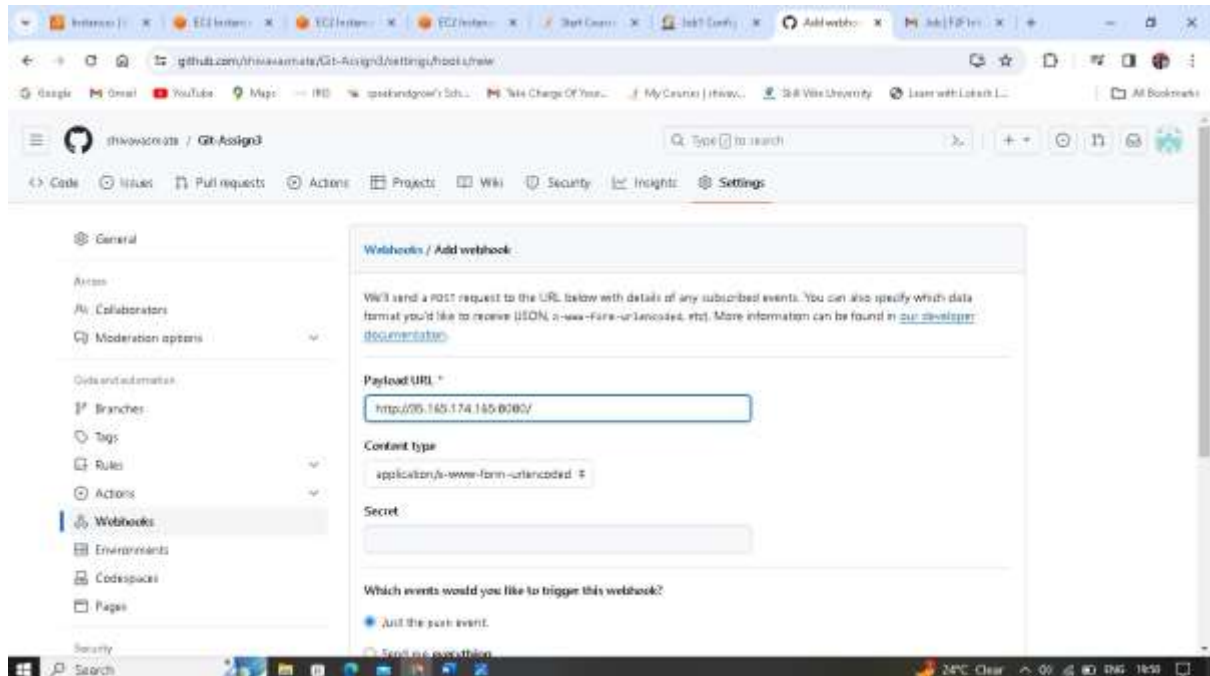
8. In Branches to build and Branch Specifier give \*/develop



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

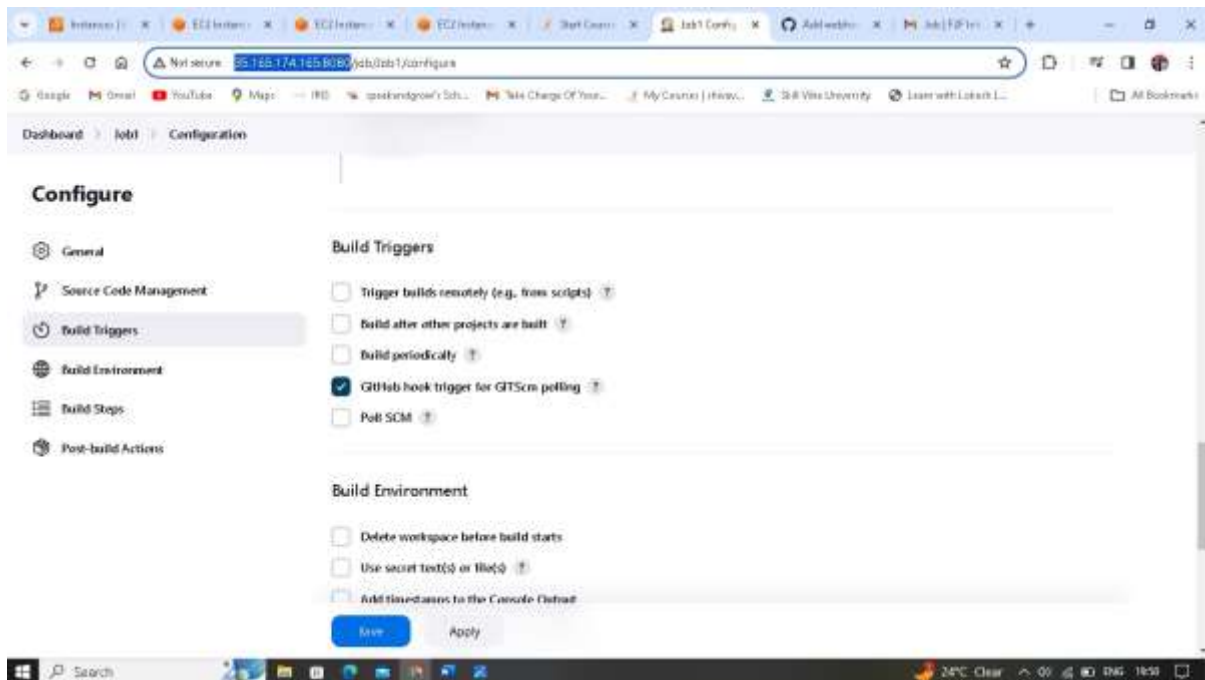


10. Now go to GitHub under repo settings look for Webhooks on left side

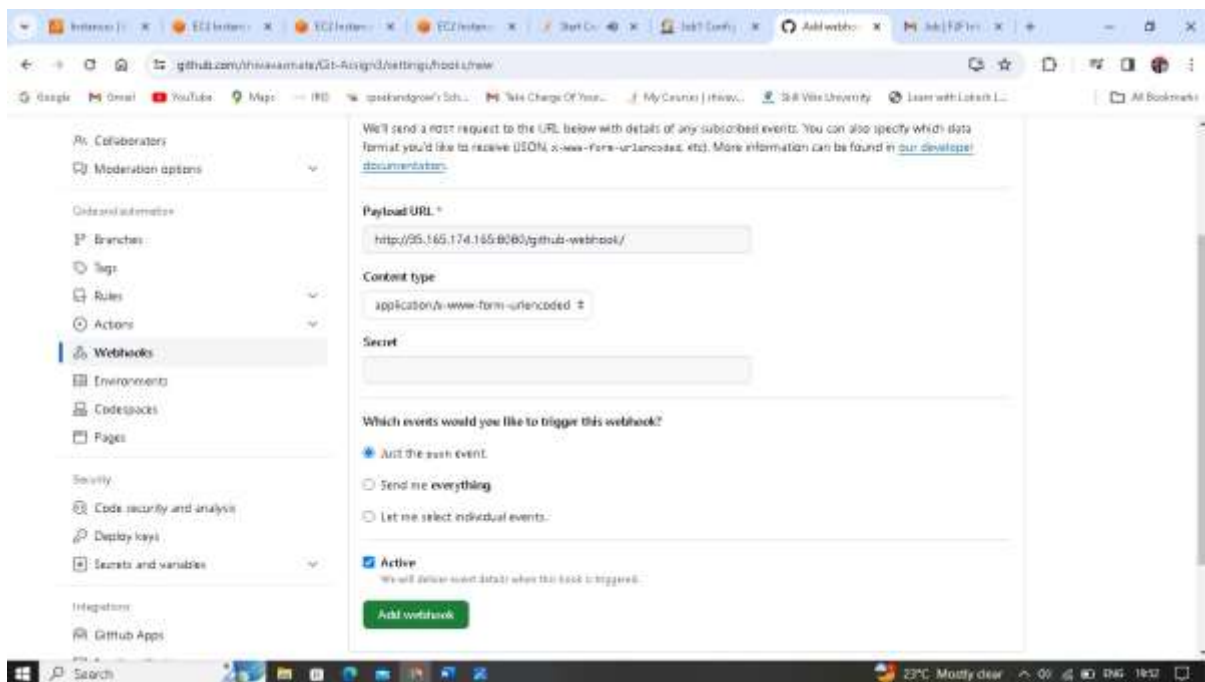




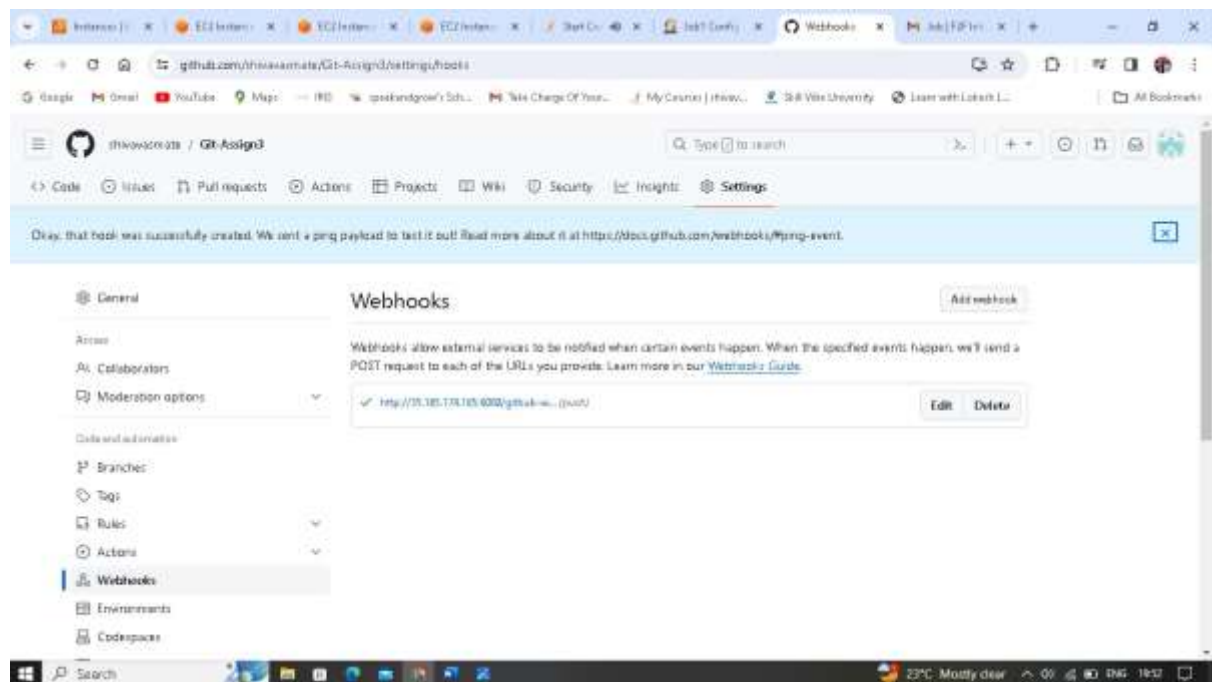
11. For Payload URL copy the `http:// 35.165.174.165:8080/`



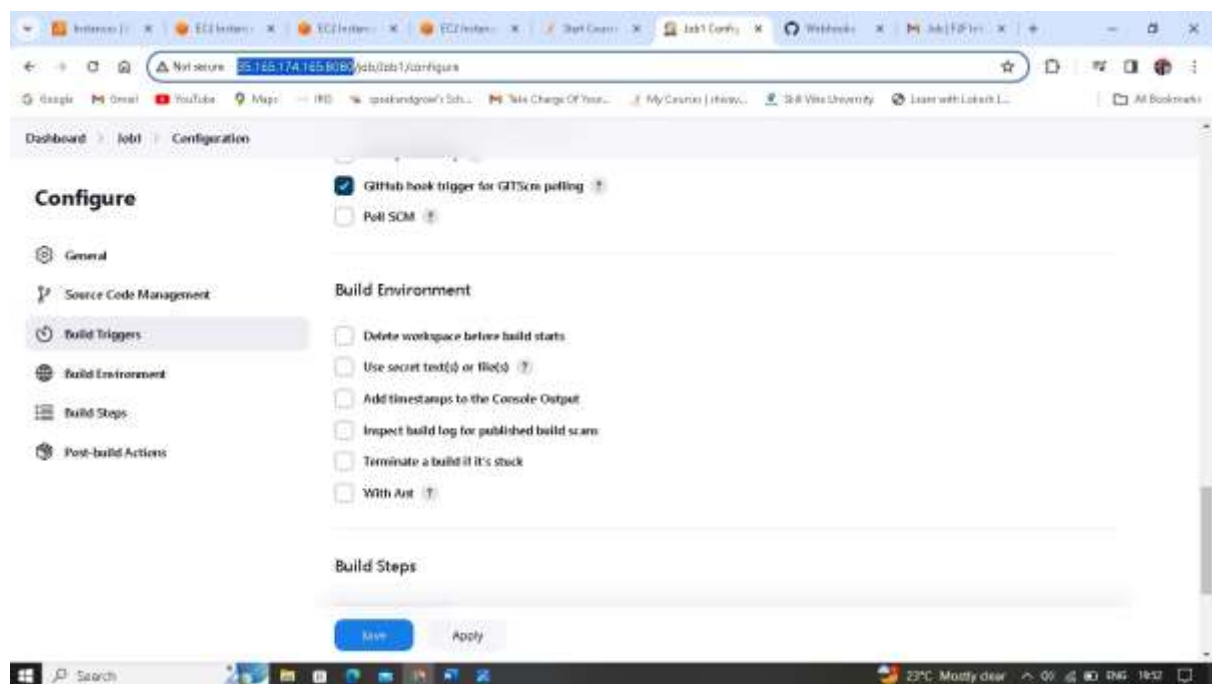
12. Give as `http://35.165.174.165:8080/github-webhook/`



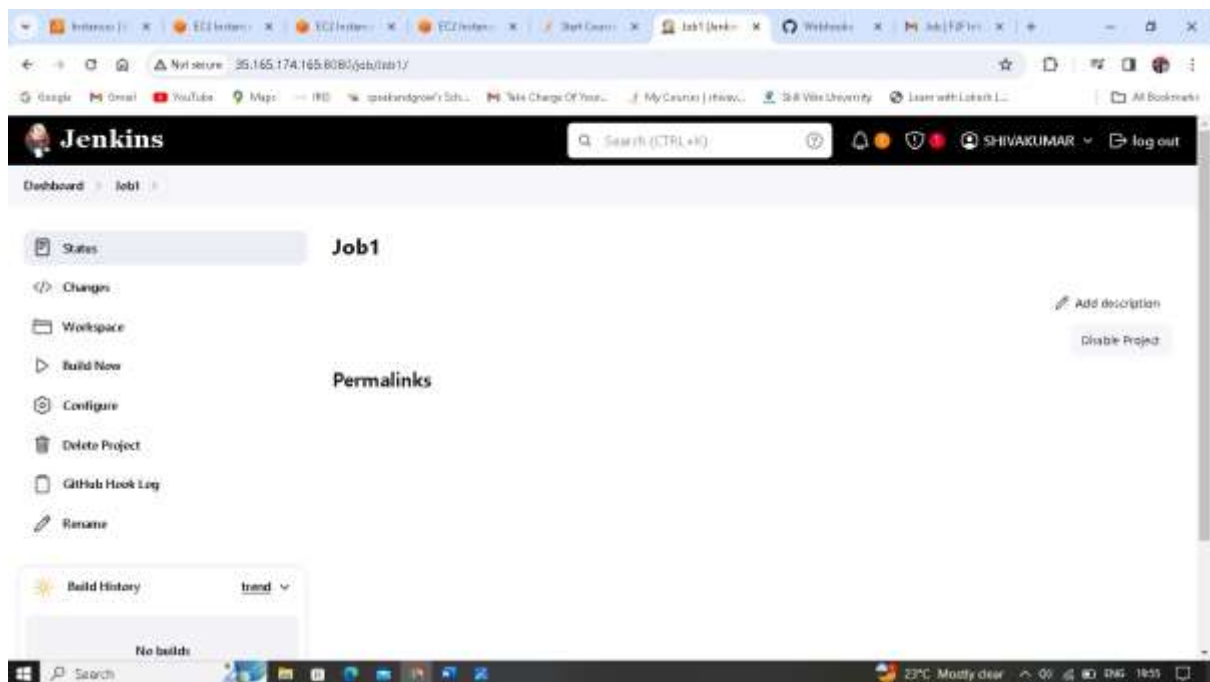
### 13. Check for right mark



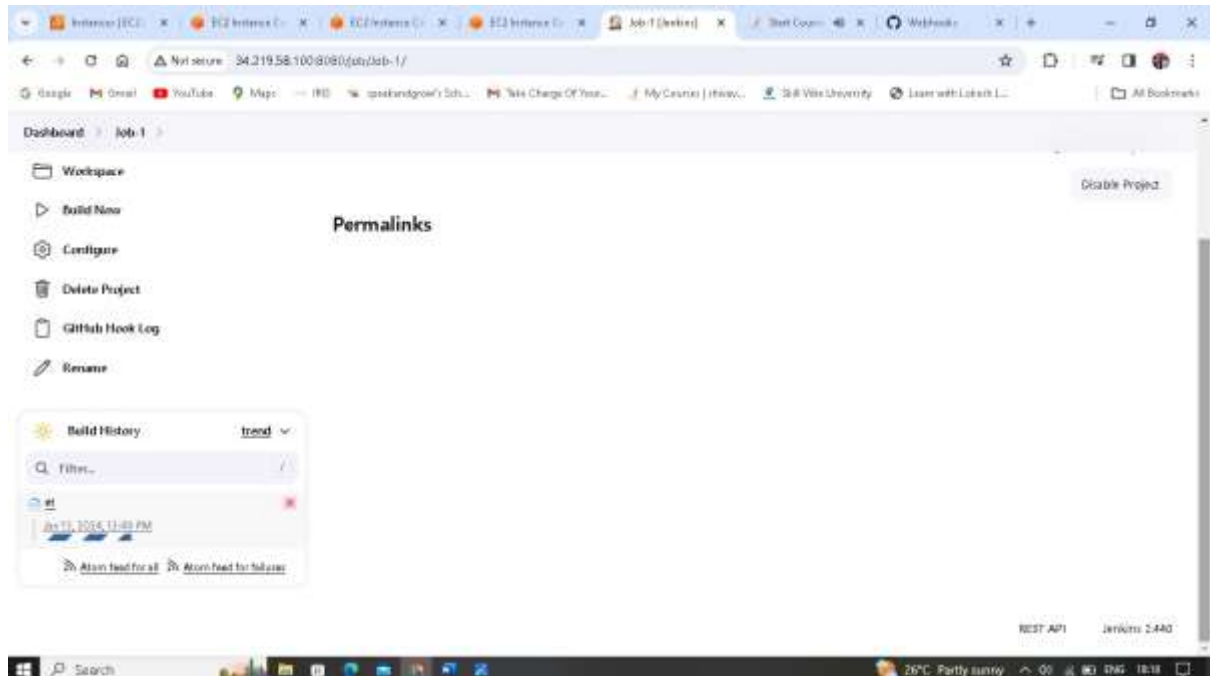
### 14. Go to Jenkins and click on save.



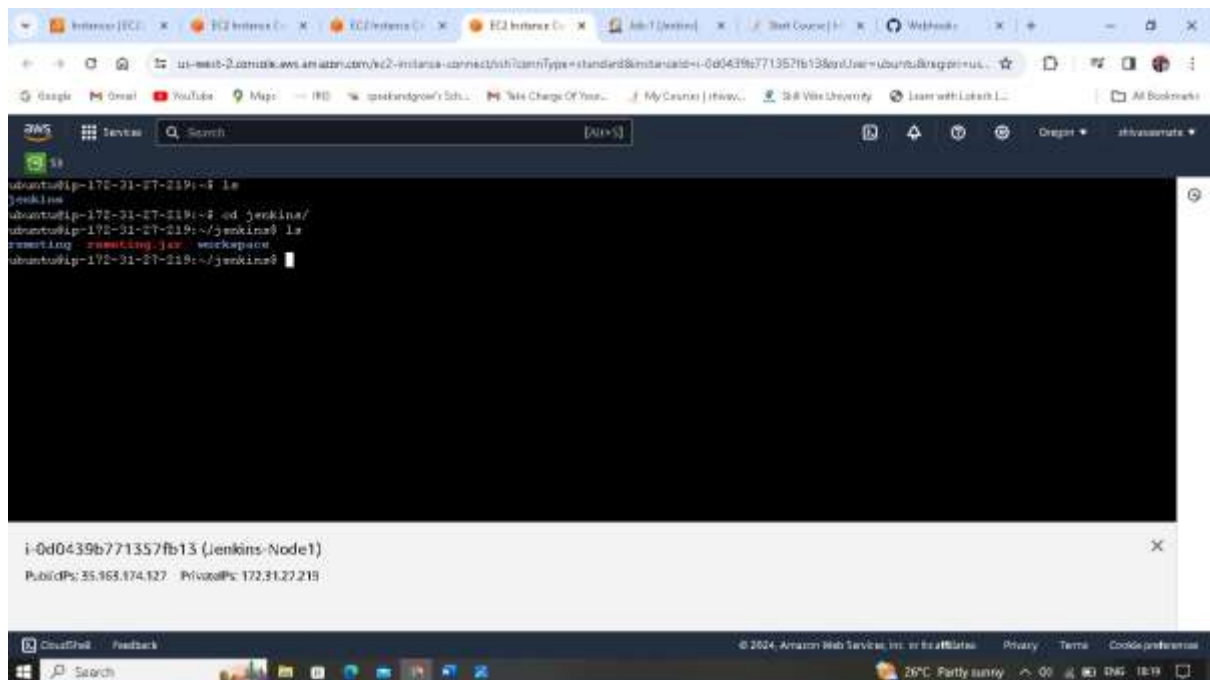
## 15. Job1 is create



## 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)

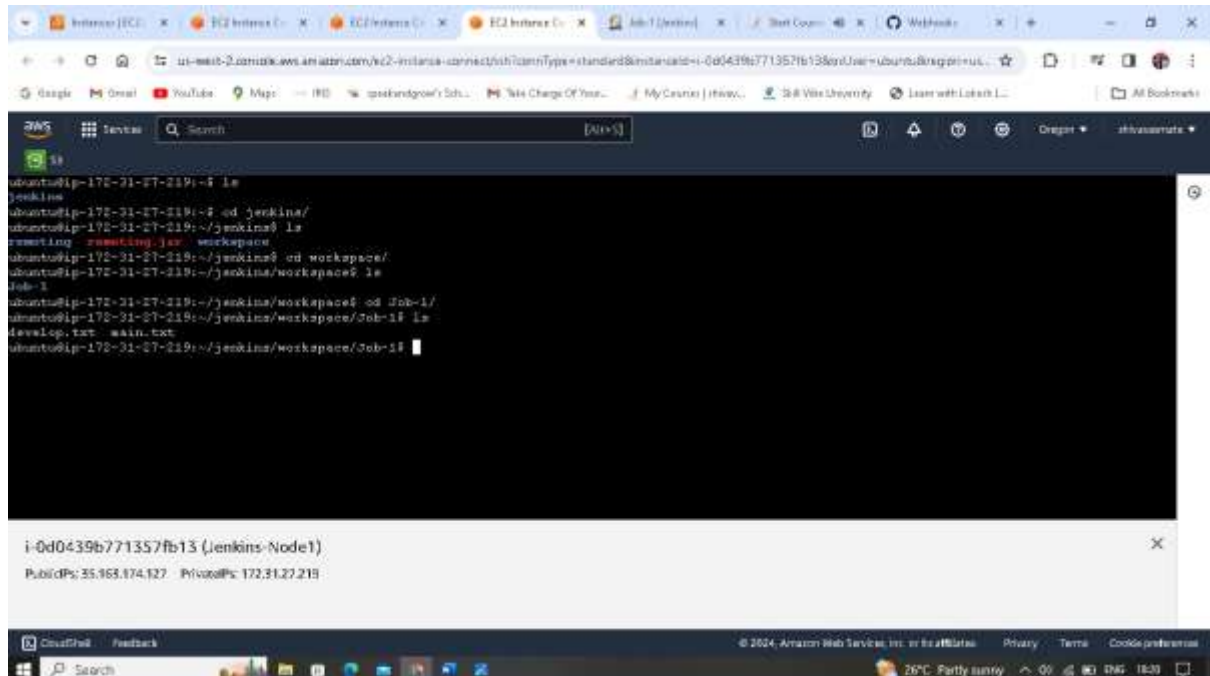


The screenshot shows a terminal window within the AWS CloudShell interface. The terminal prompt is `ubuntu@ip-172-31-27-219:~$`. The user has navigated to the `jenkins` directory and executed `ls`. The output shows the following files and directories: `remoting`, `remoting.jar`, and `workspace`. The `workspace` directory is newly created. The terminal output is as follows:

```
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$
```

Below the terminal window, the instance details for `i-0d0439b771357fb13 (Jenkins-Node1)` are visible, showing the Public IP as `35.163.174.127` and the Private IP as `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.

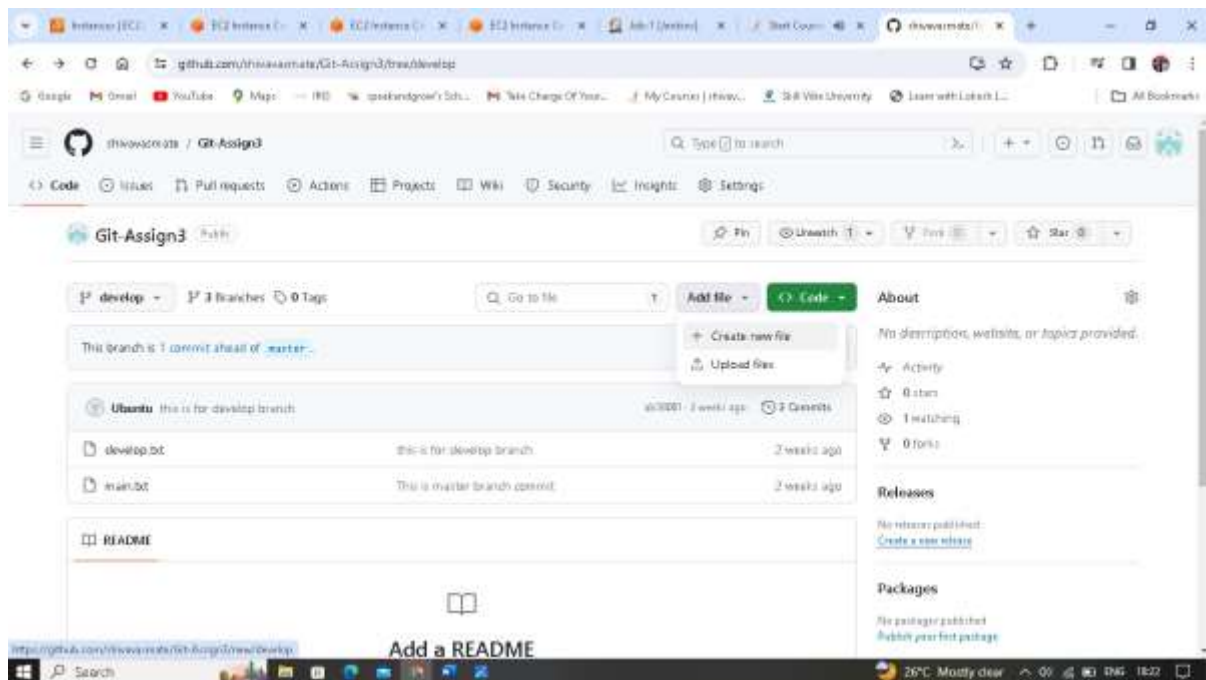


The screenshot shows a terminal window within the AWS CloudShell interface. The terminal prompt is `ubuntu@ip-172-31-27-219:~$`. The user has navigated to the `workspace` directory and executed `ls`. The output shows the following files and directories: `Job-1`, `develop.txt`, and `main.txt`. The `Job-1` directory is newly created. The terminal output is as follows:

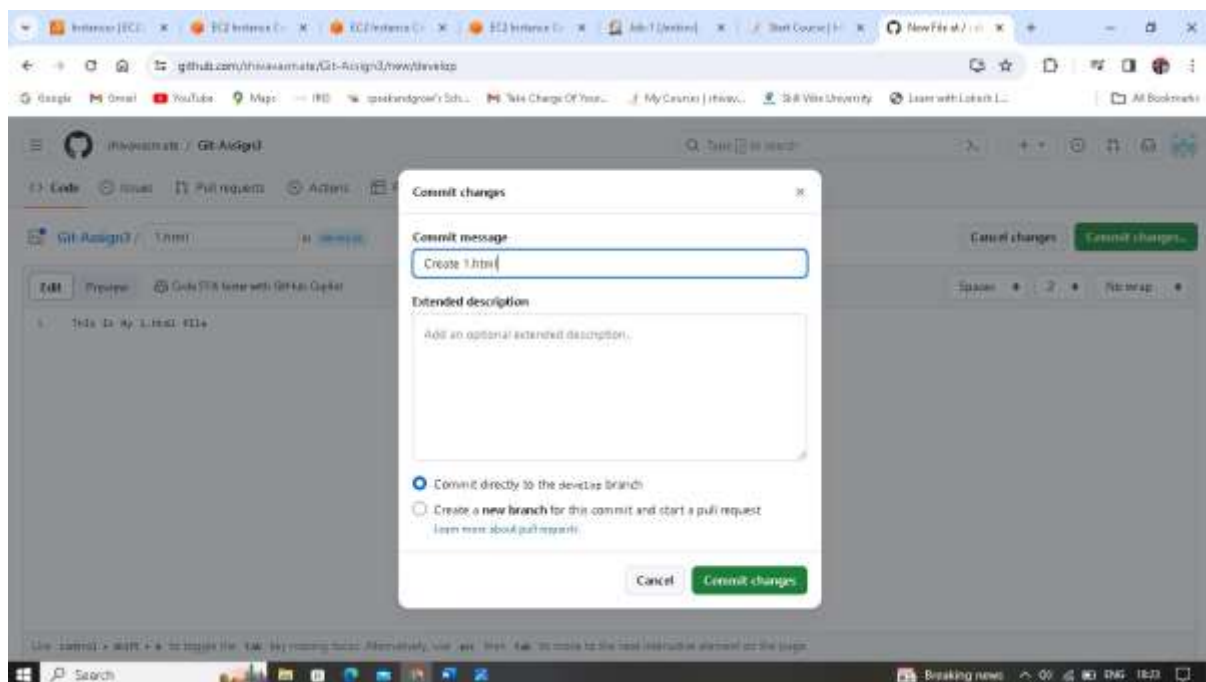
```
ubuntu@ip-172-31-27-219:~$ ls
jenkins
ubuntu@ip-172-31-27-219:~$ cd jenkins/
ubuntu@ip-172-31-27-219:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-27-219:~/jenkins$ cd workspace/
ubuntu@ip-172-31-27-219:~/jenkins/workspace$ ls
Job-1
ubuntu@ip-172-31-27-219:~/jenkins/workspace$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

Below the terminal window, the instance details for `i-0d0439b771357fb13 (Jenkins-Node1)` are visible, showing the Public IP as `35.163.174.127` and the Private IP as `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

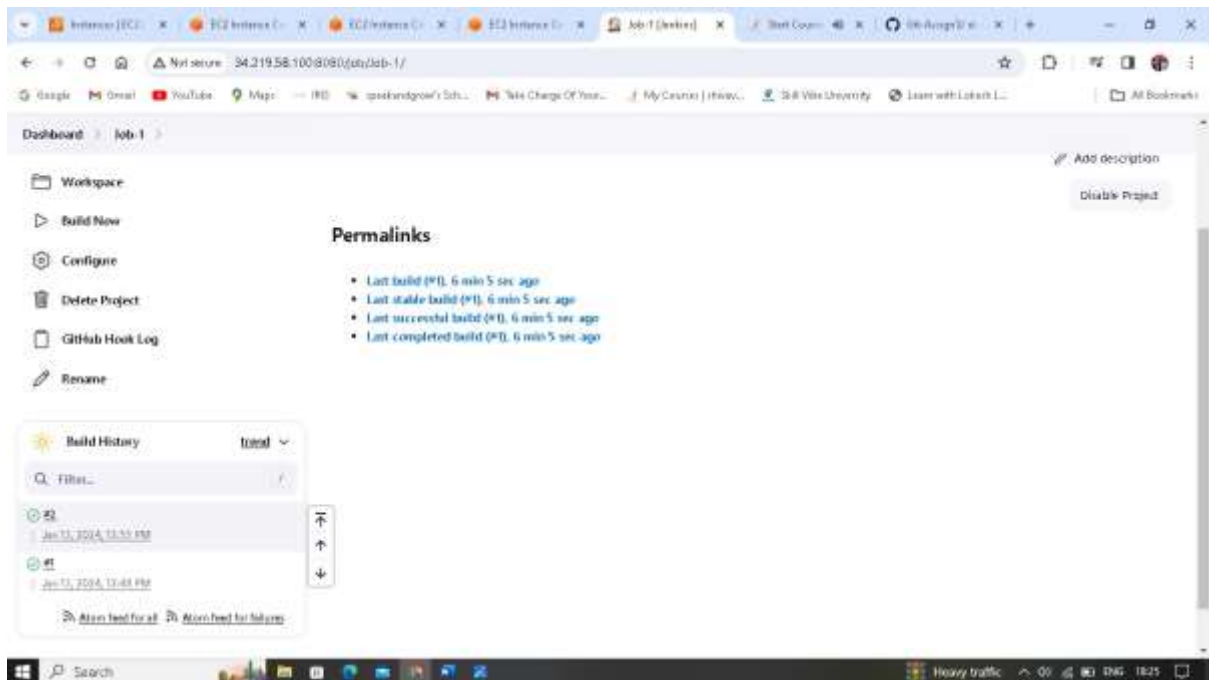


20. Now click on commit changes

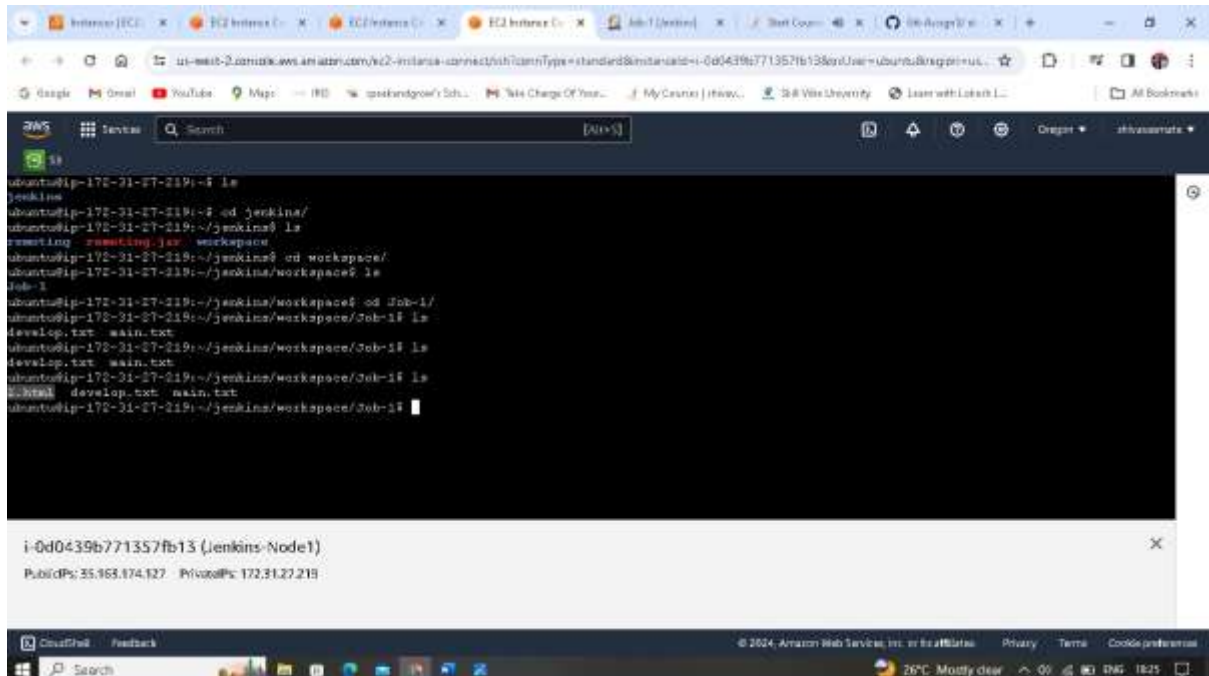




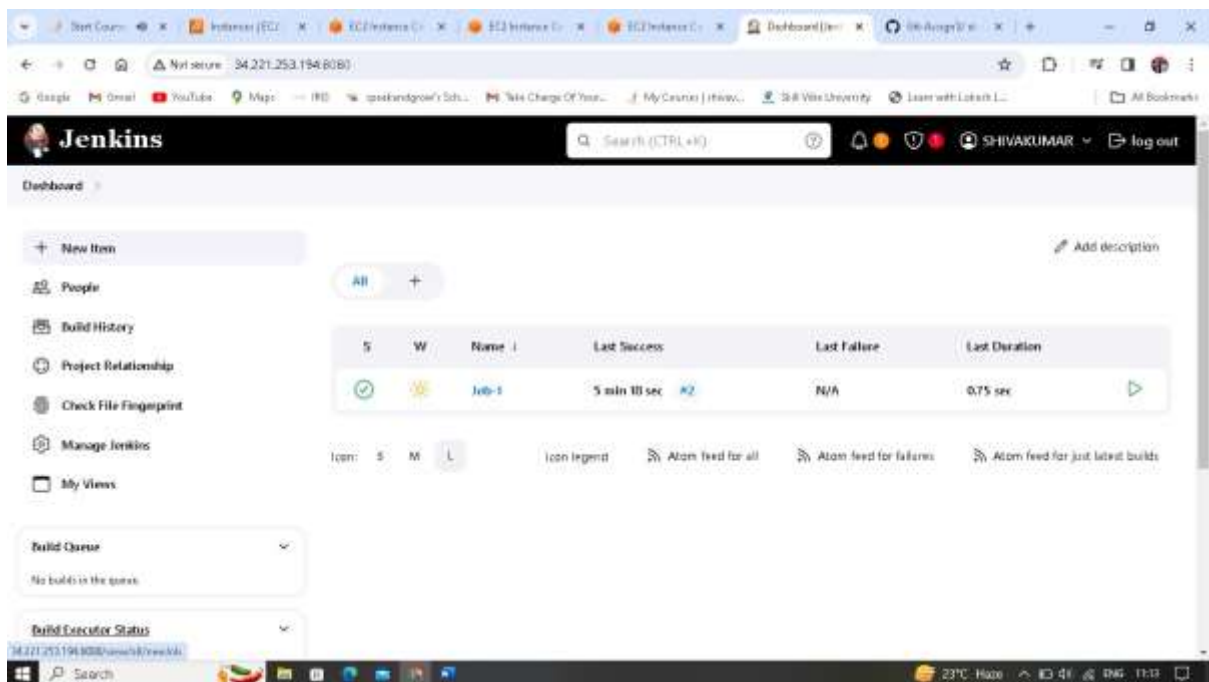
21. Automatic #2 build is created



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



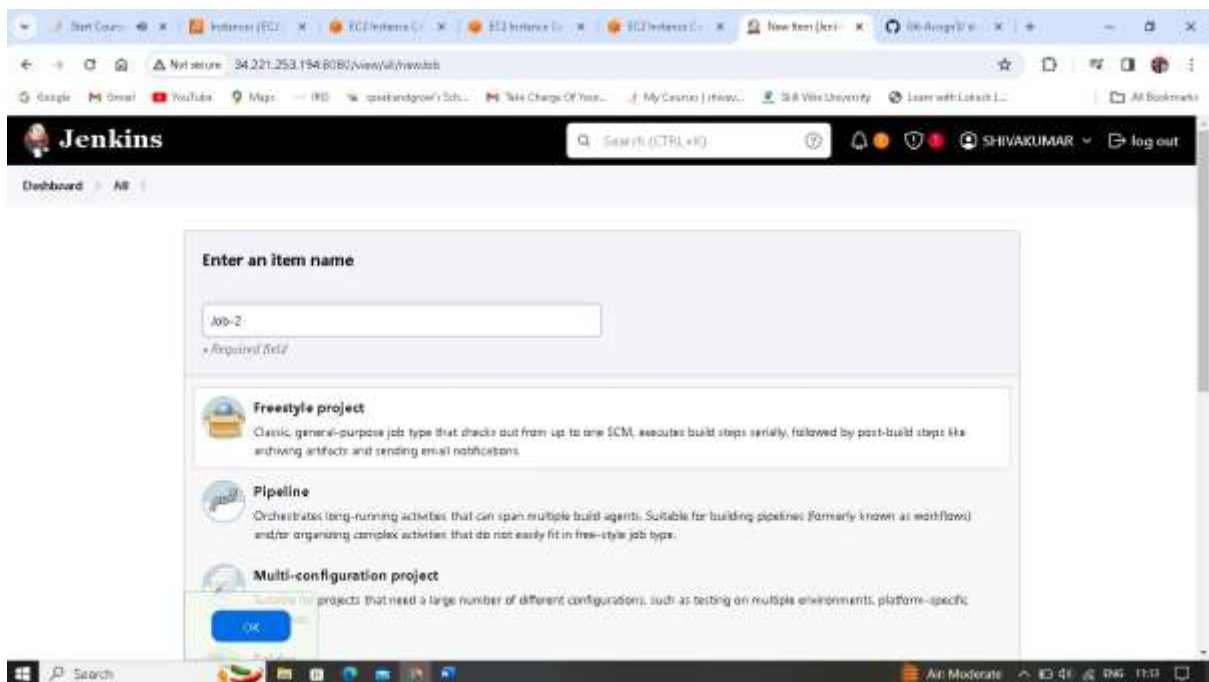
1. Go back to Dashboard and click on New item.



The screenshot shows the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for SHIVAKUMAR. The main content area features a sidebar on the left with options like 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The central part of the dashboard displays a table of jobs. The table has columns for 'S', 'W', 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. A single job named 'Job-1' is listed with a success status and a duration of 0.75 sec. Below the table, there are filters for 'Icon', 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. On the left, there are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	✓	Job-1	5 min 10 sec #2	N/A	0.75 sec

2. Enter an item name as Job-2



The screenshot shows the 'New Item' dialog box in Jenkins. The 'Enter an item name' field contains 'Job-2'. Below this, there are three options for the item type: 'Freestyle project', 'Pipeline', and 'Multi-configuration project'. The 'Freestyle project' option is selected. The 'OK' button is highlighted in blue.

Enter an item name

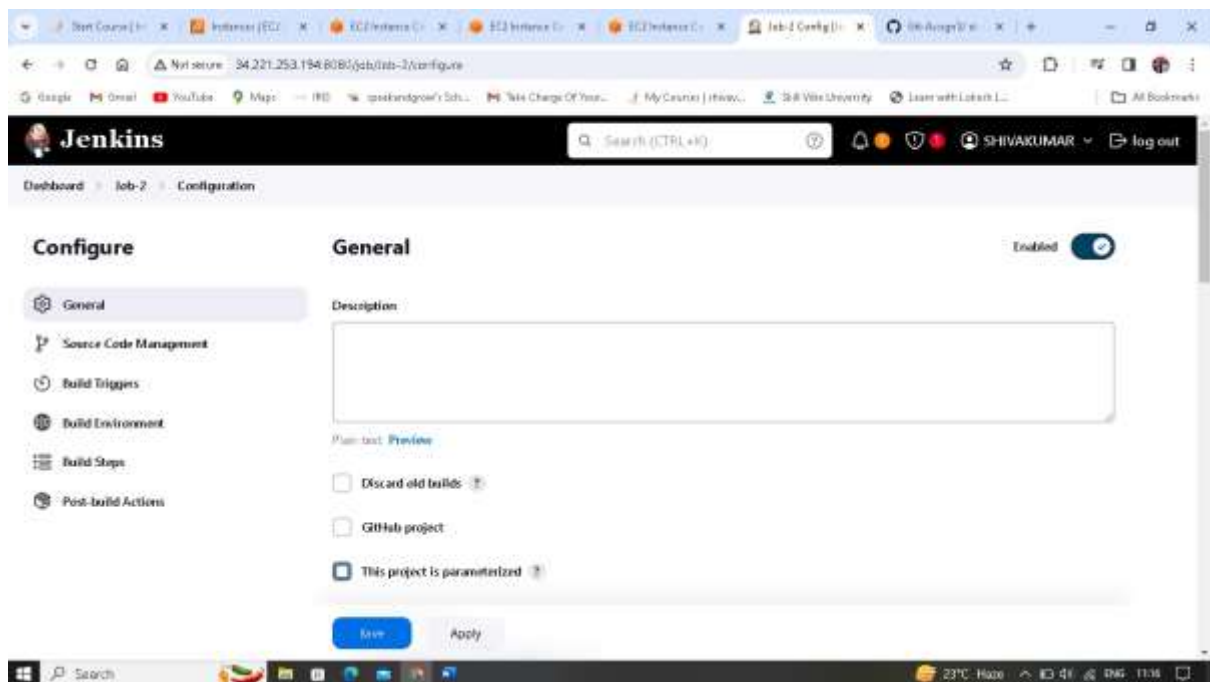
Job-2

Required field

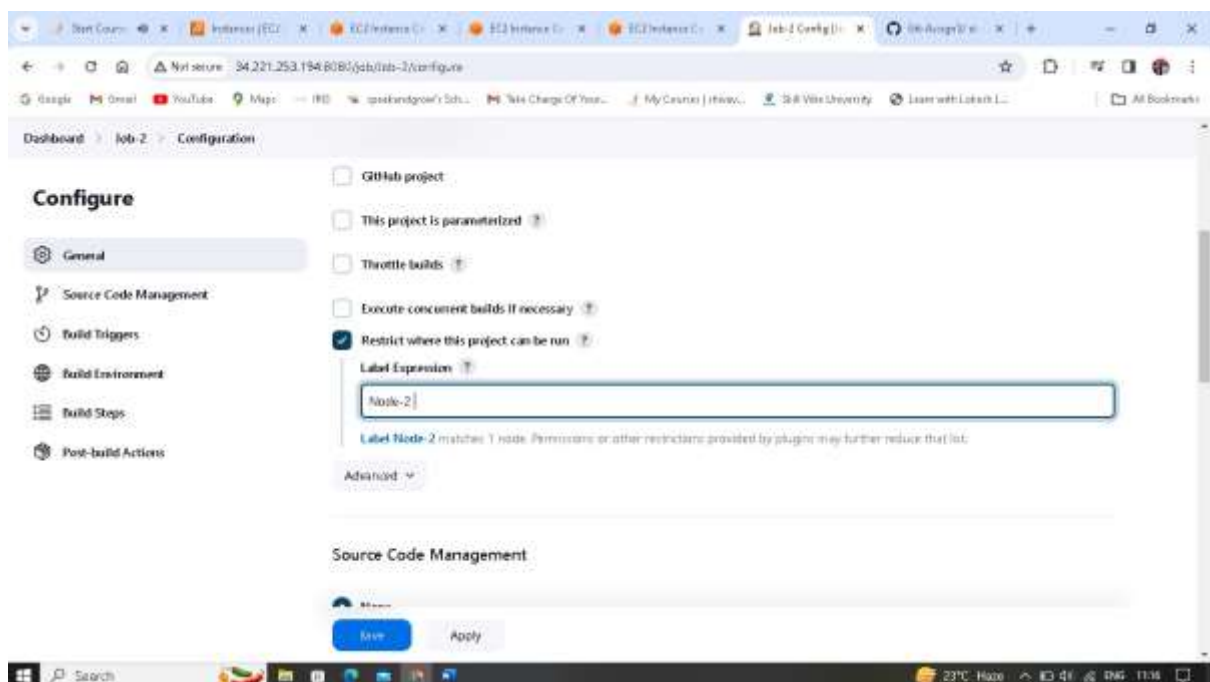
- Freestyle project**  
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific

OK

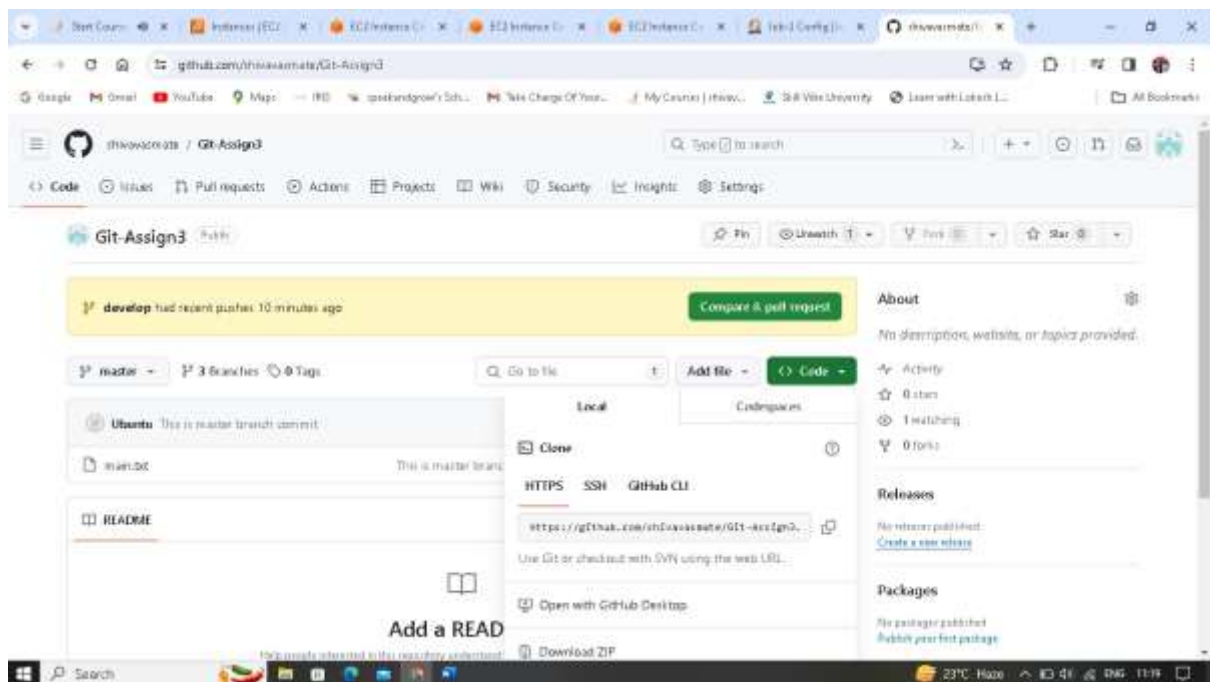
3. fill it



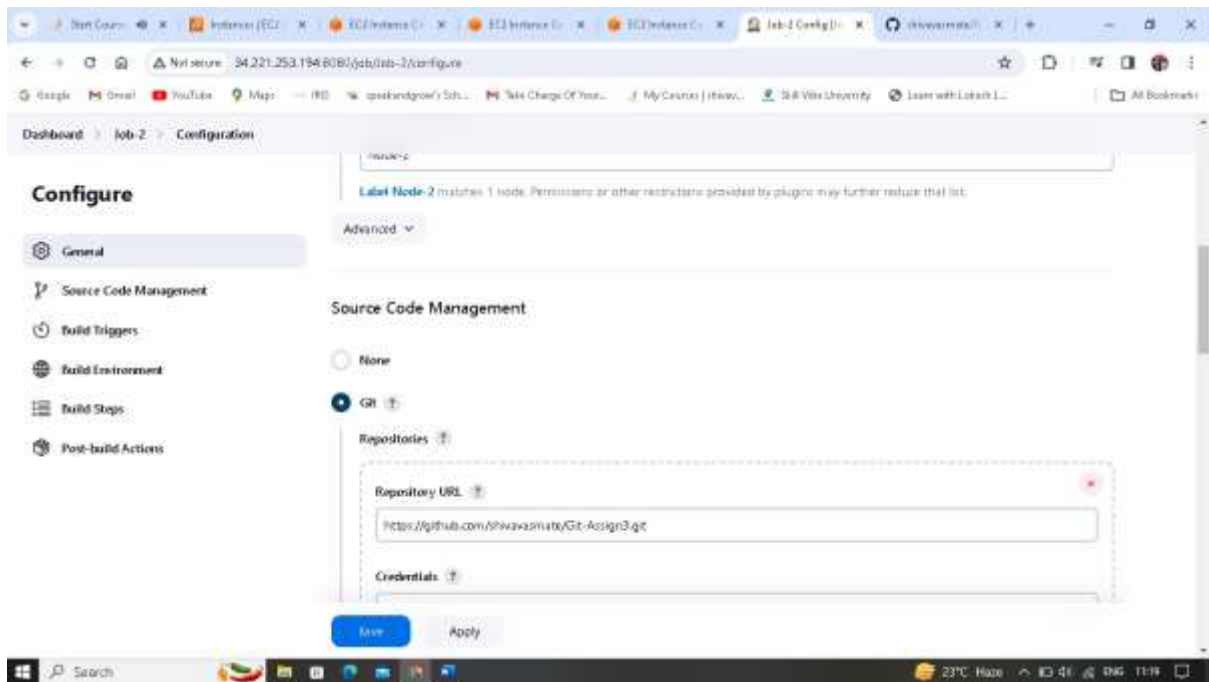
4. In Label Expression as Node-2



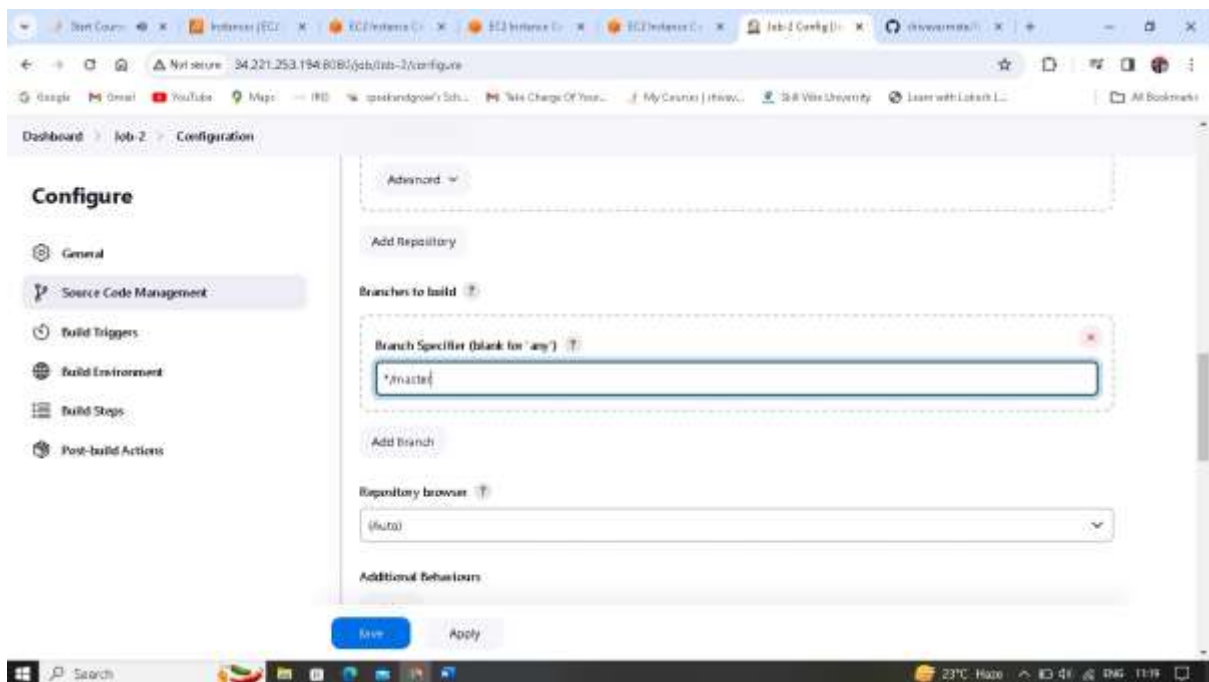
5. copy the URL for Github



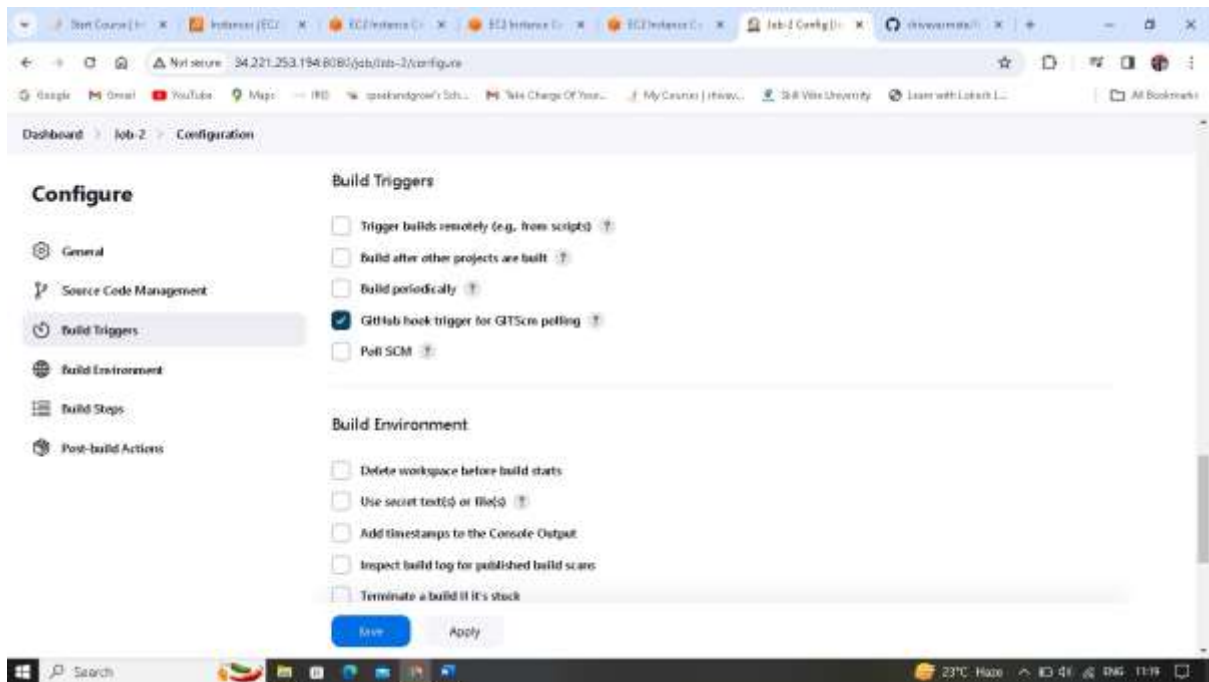
6. Paste in the Repository URL and choose the repo which has master branch in it (Git-Assign3).



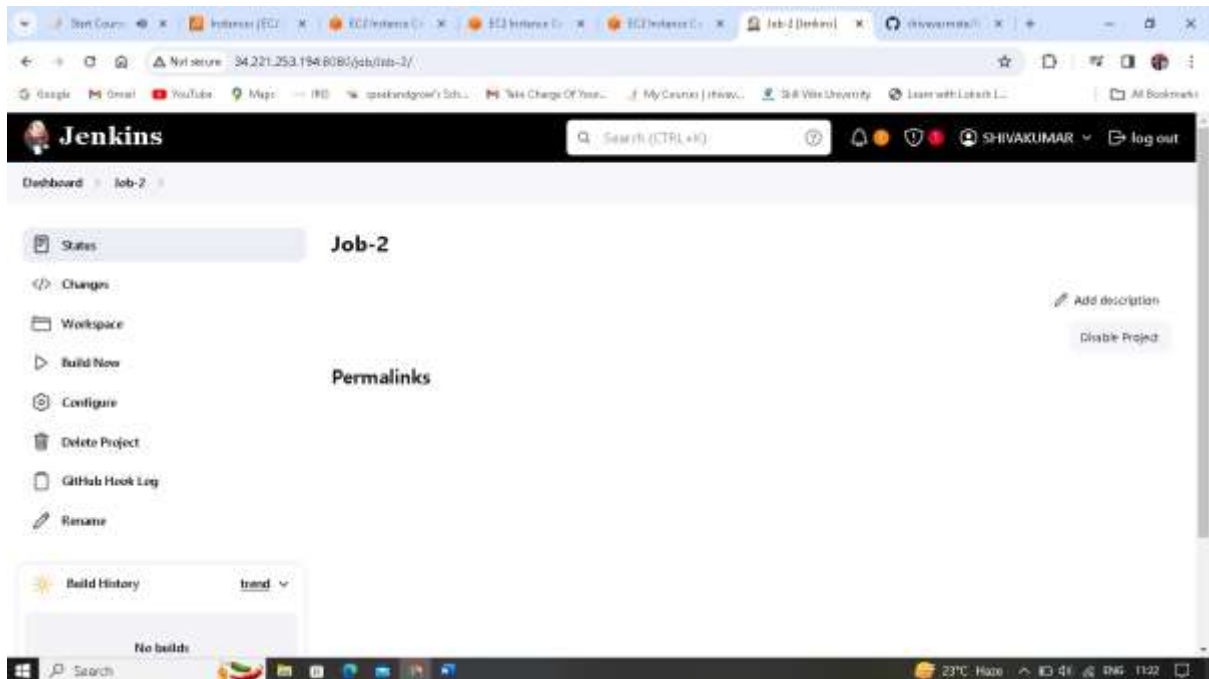
7. In Branches to build and Branch Specifier give \*/master



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

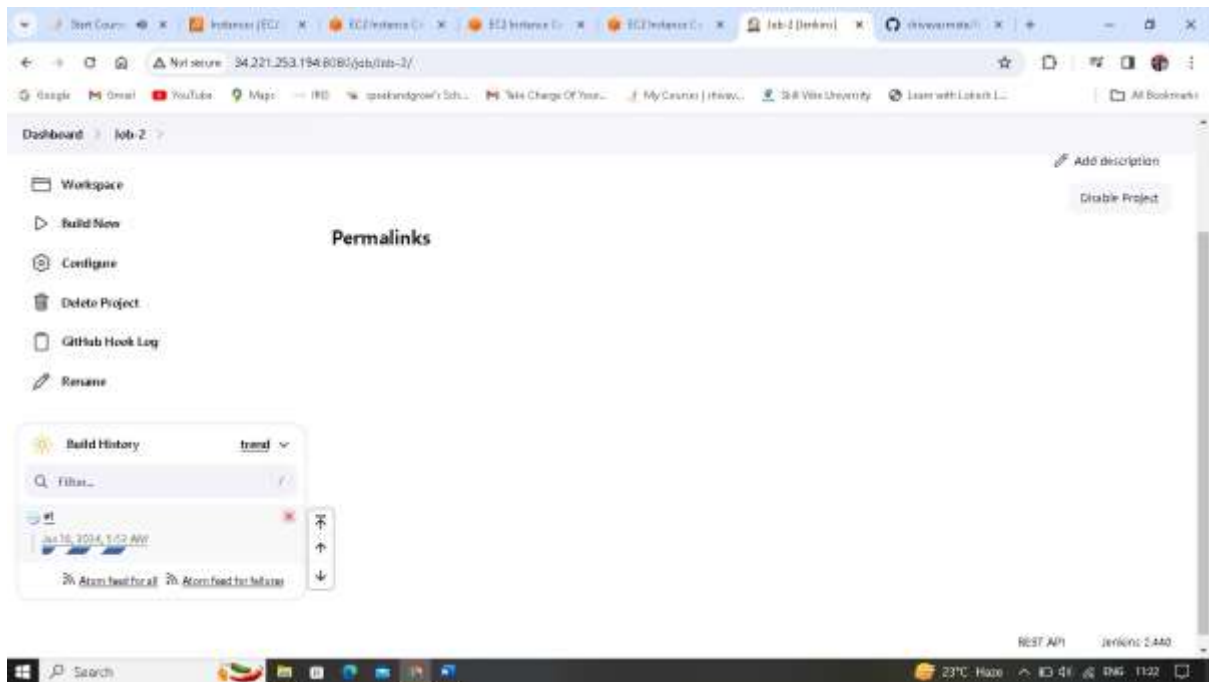


9. Job-2 is created

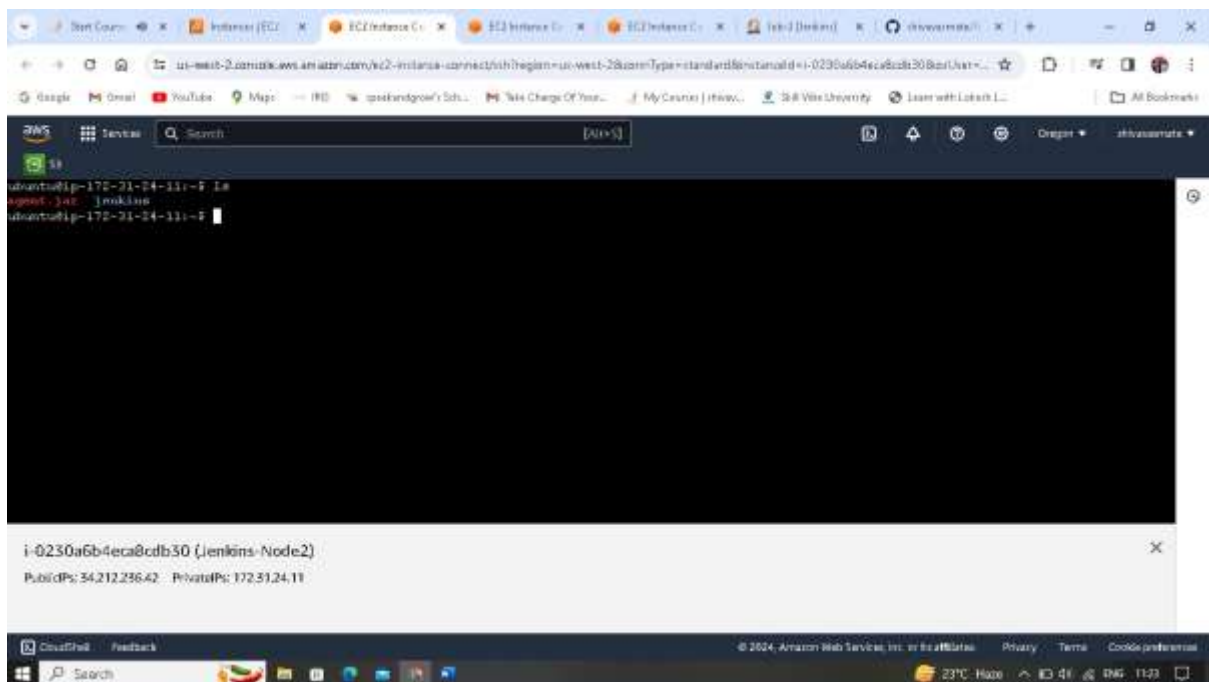


10. Click on Build Now.

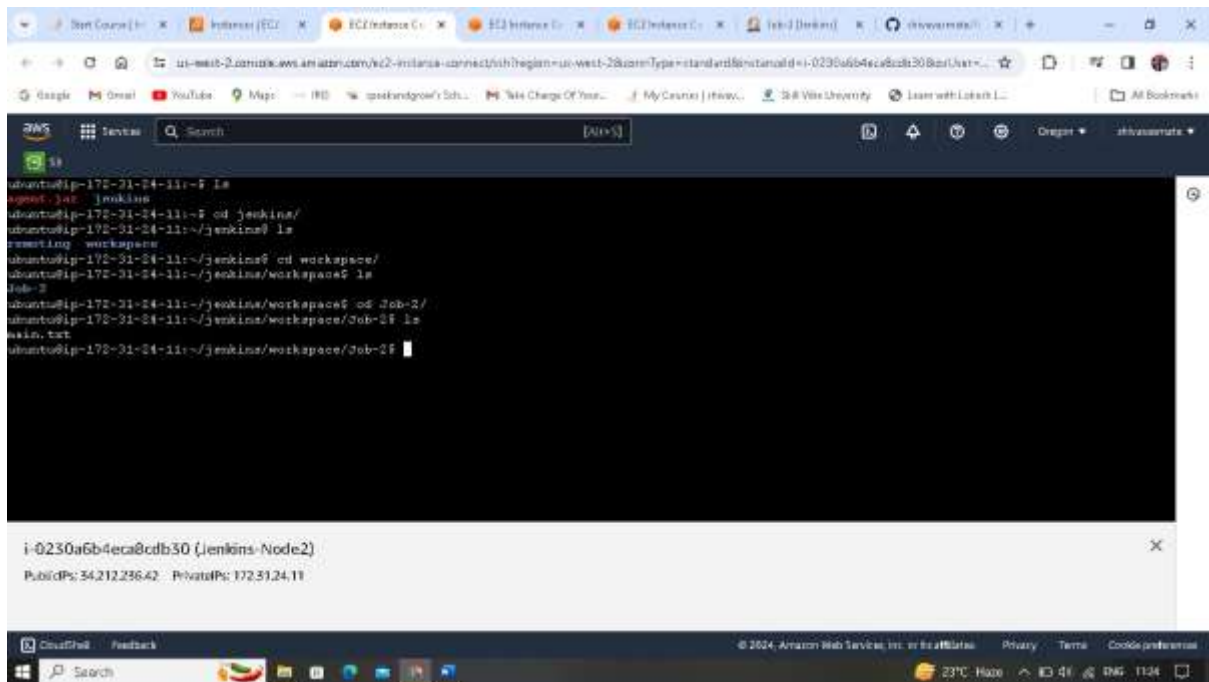




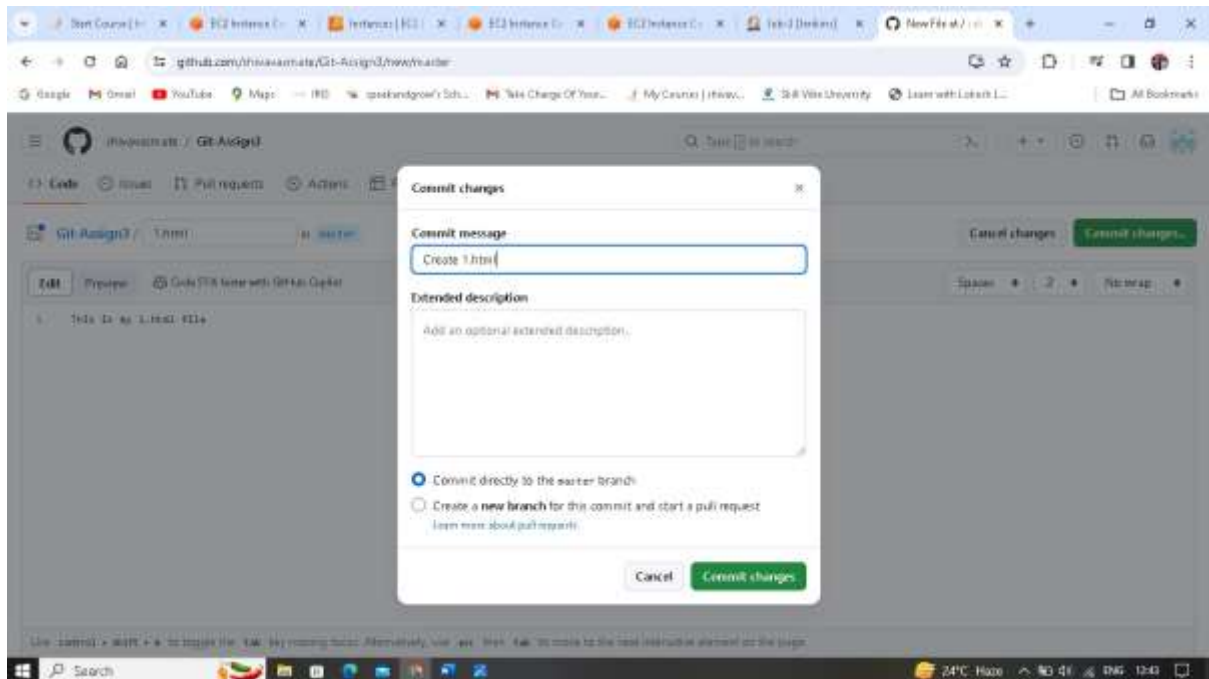
11. Go to Node-2 machine do ls, agent.jar and jenkins folders are present.



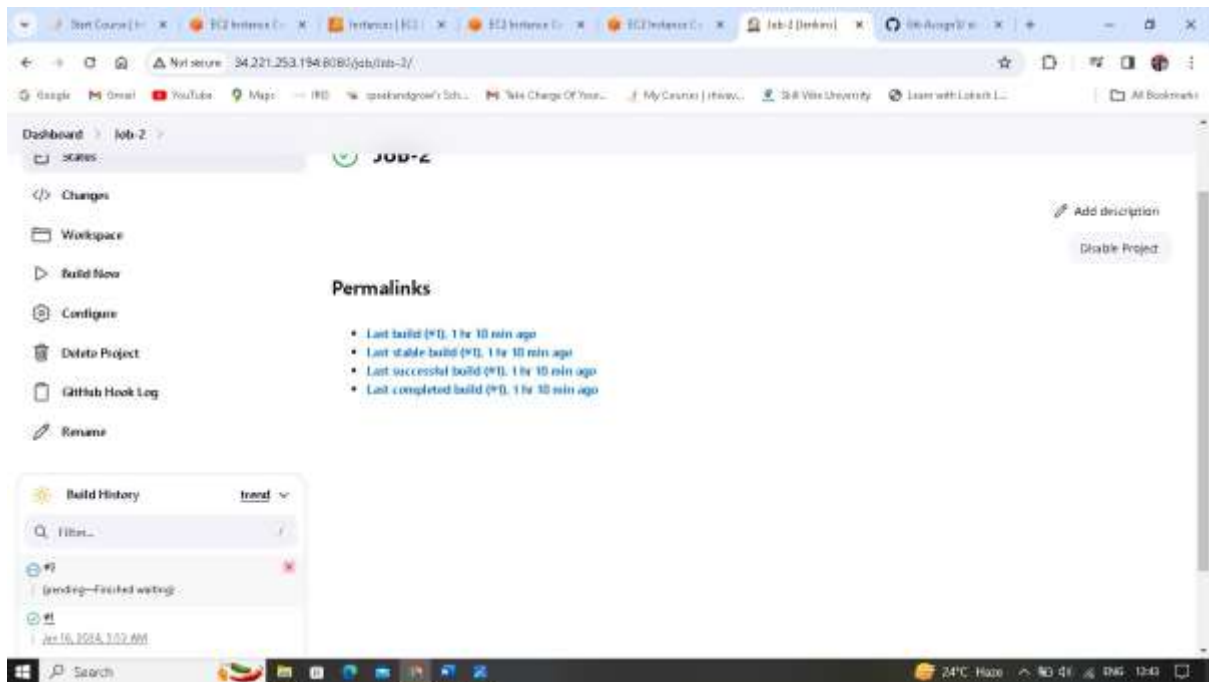
12. Finally Job-2 have main.txt from main branch of github.



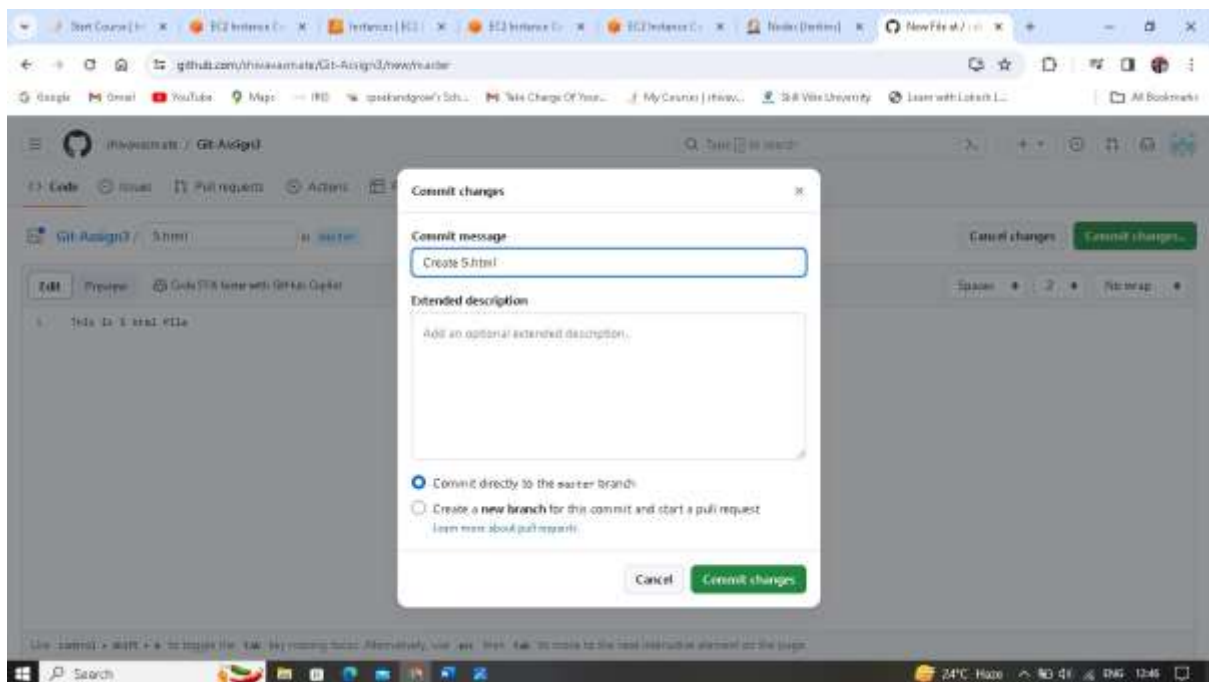
13. Now add a new file 1.html and click on Commit changes.



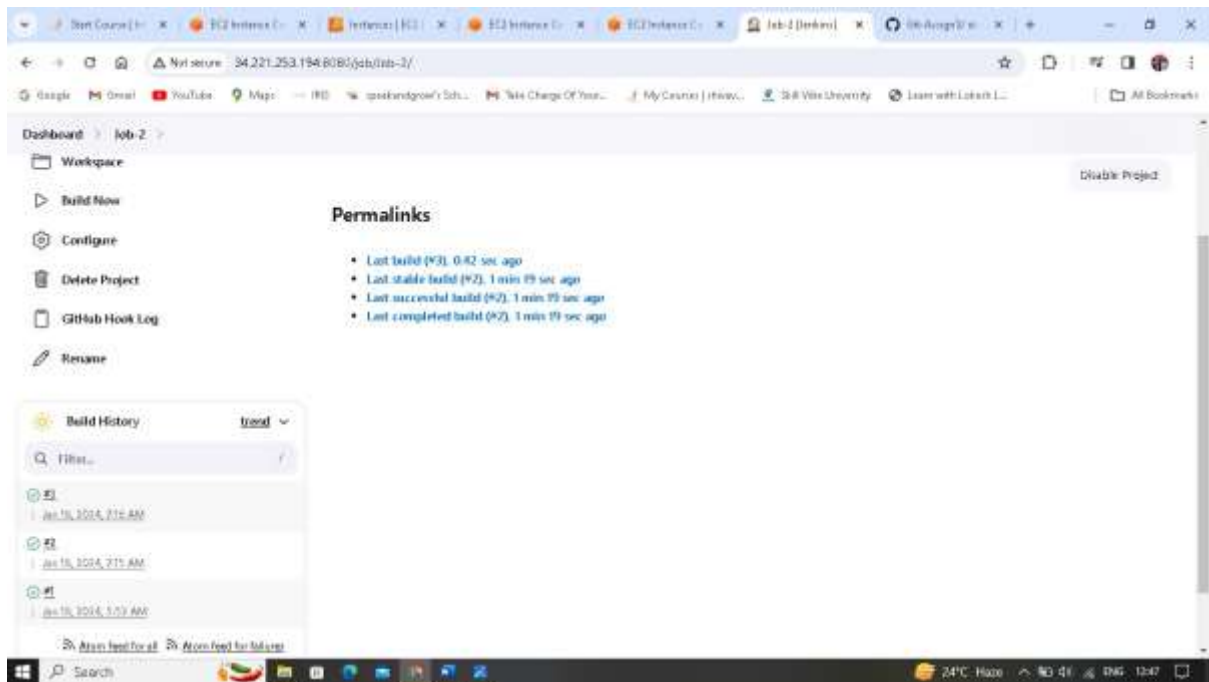
14.#2 build is created automatically



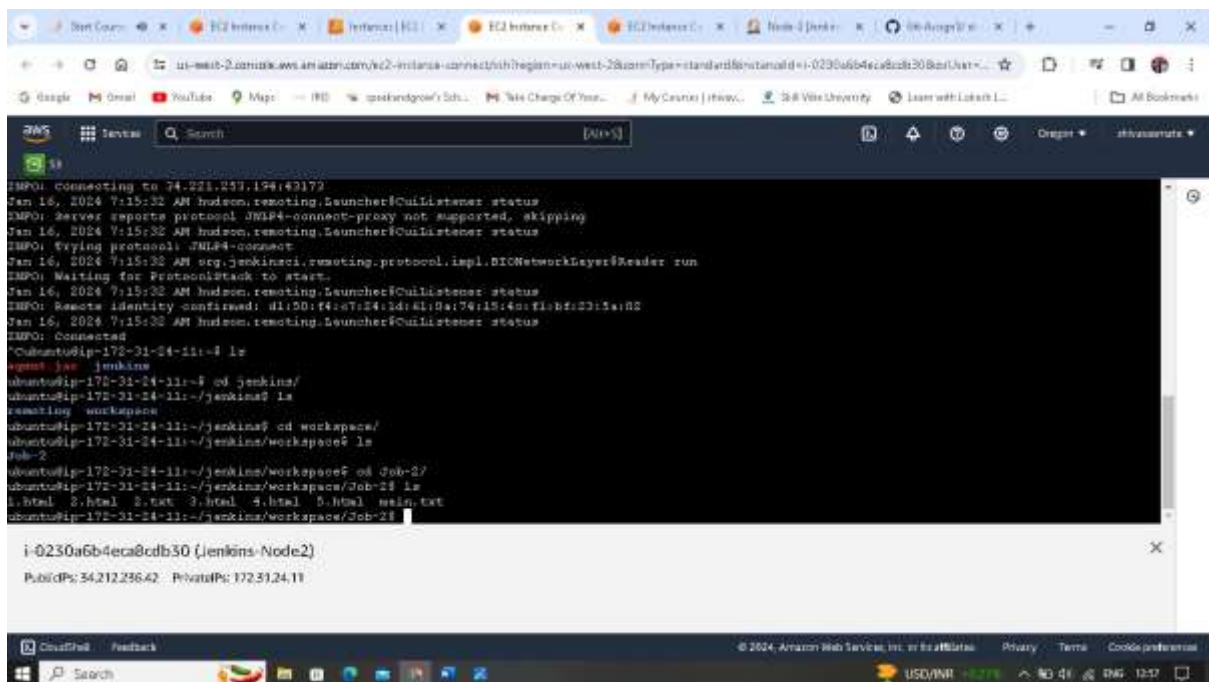
15. Similarly For 5.html file



16.#3 build also created automatically

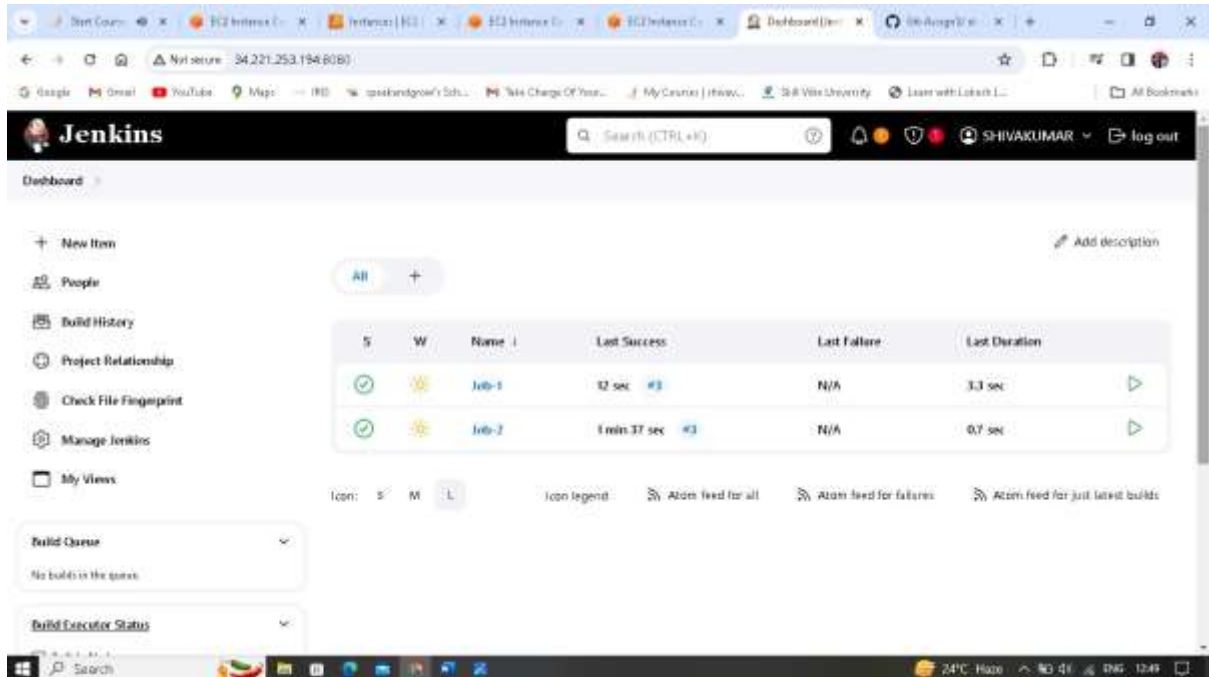


17. Go to Jenkins-Node 2 check for 1.html and 5.html files in the Job-2 folder.



Job3 → Job1-Job2 (combine Jobs)

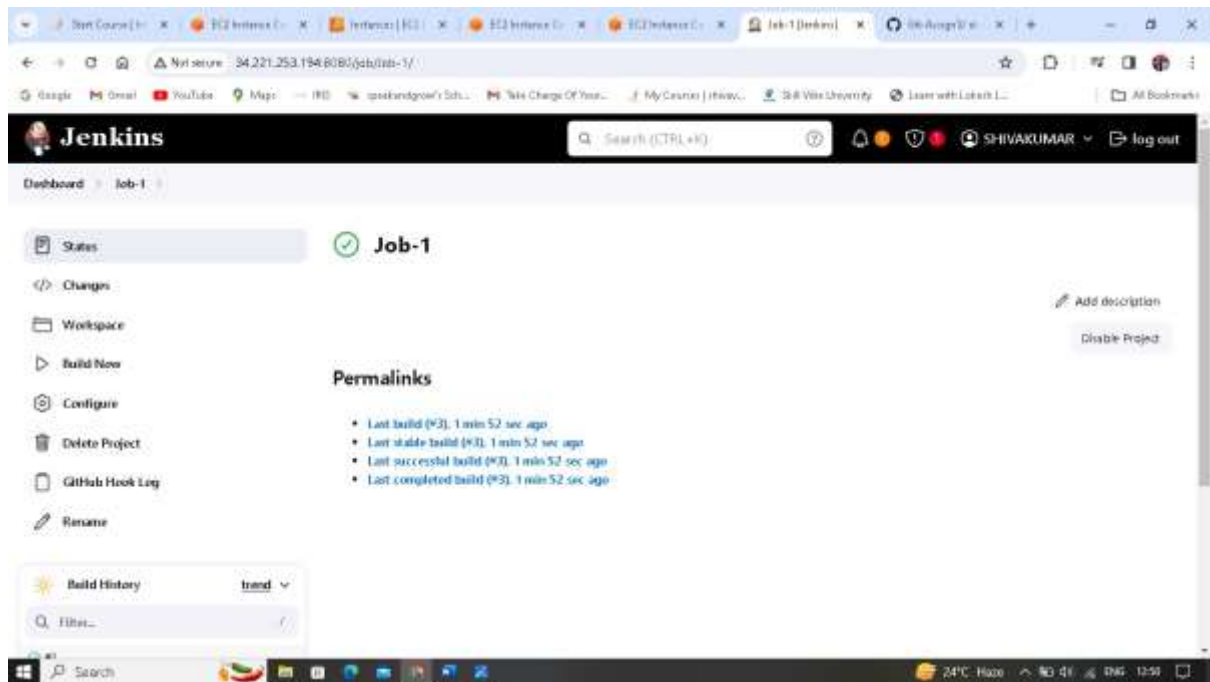
1. Go back to Dashboard and click on Job-1



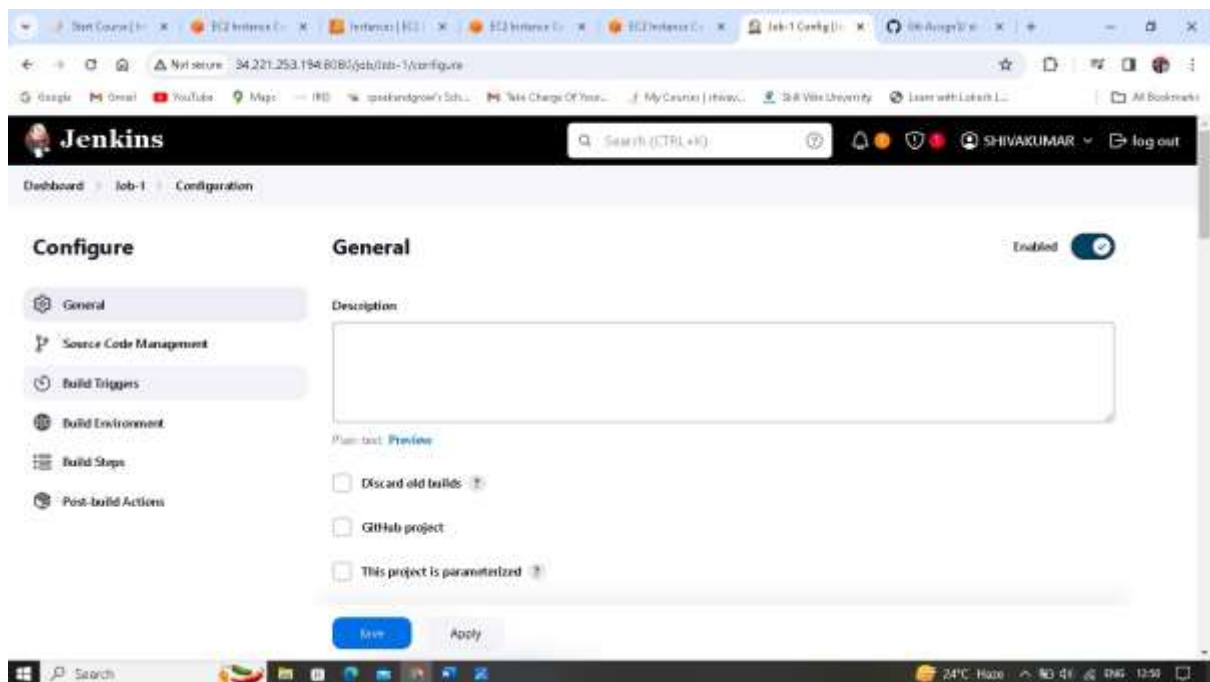
The screenshot shows the Jenkins Dashboard in a web browser. The dashboard has a sidebar on the left with links to 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of jobs. The table has columns for 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. There are two jobs listed: 'Job-1' and 'Job-2'. 'Job-1' has a last success of '12 sec' and a last failure of 'N/A'. 'Job-2' has a last success of '1 min 37 sec' and a last failure of 'N/A'. Below the table, there are filters for 'Icon: S M L', 'Icon legend', and 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. The bottom of the dashboard shows 'Build Queue' and 'Build Executor Status' sections.

Name	Last Success	Last Failure	Last Duration
Job-1	12 sec	N/A	3.3 sec
Job-2	1 min 37 sec	N/A	0.7 sec

2. Now click on configure

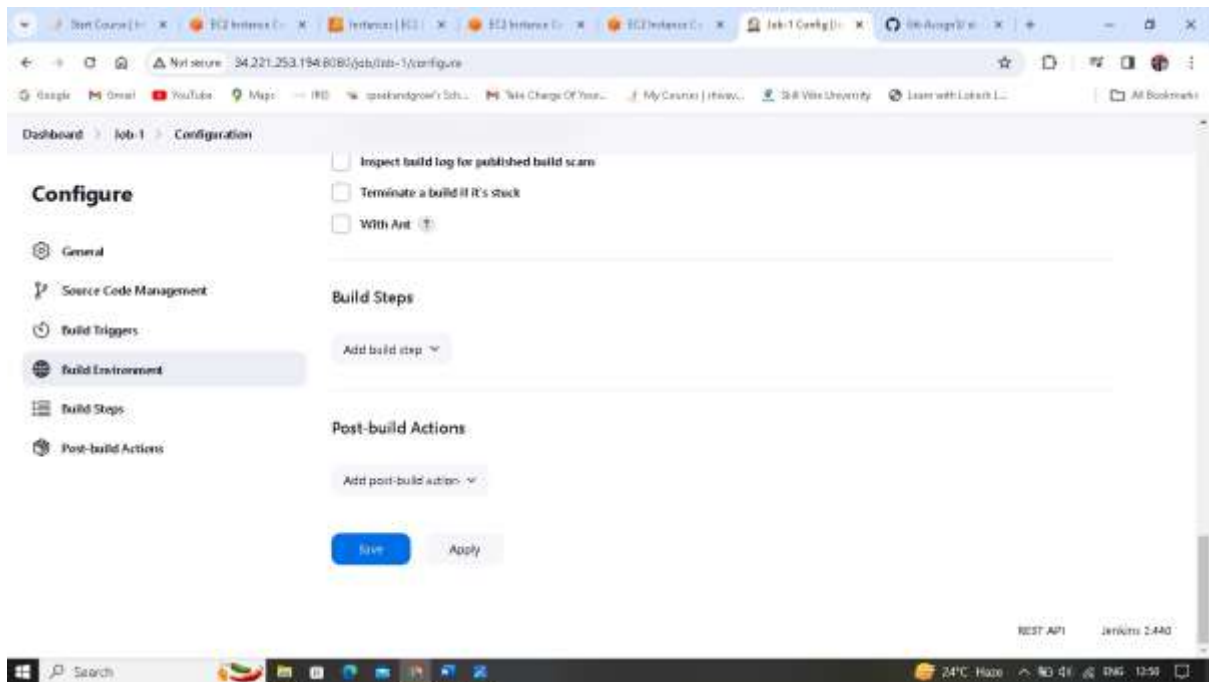


3. Go at the end

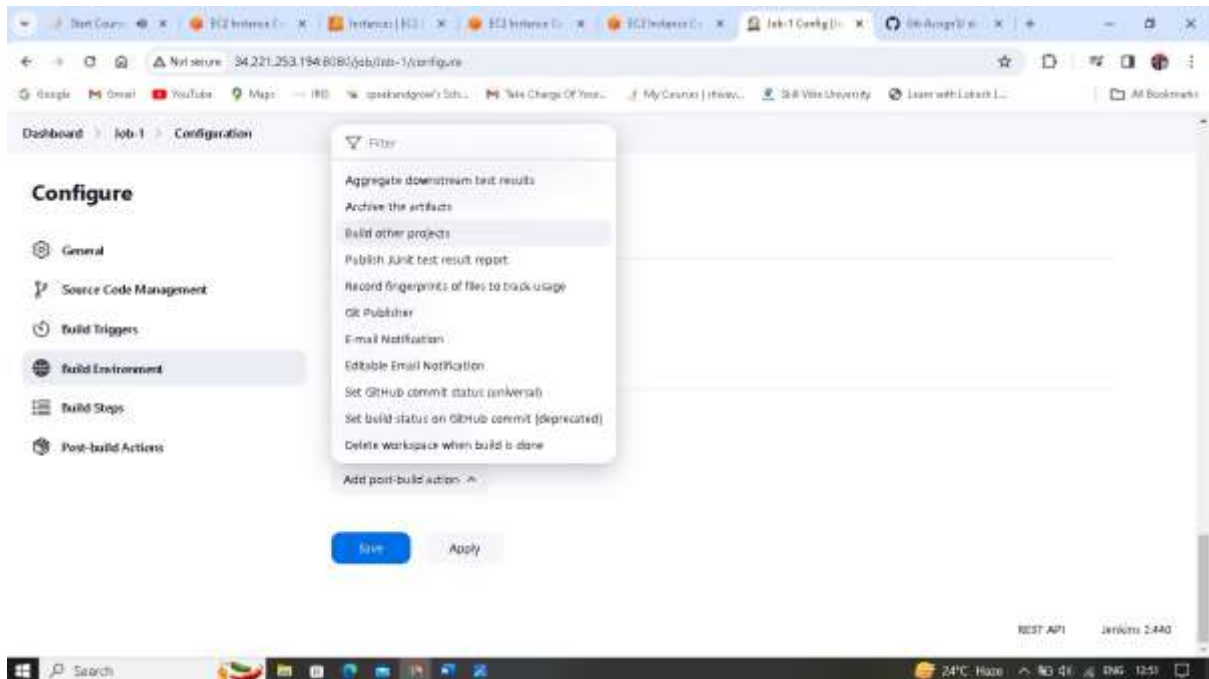


4. Under Post-build Actions

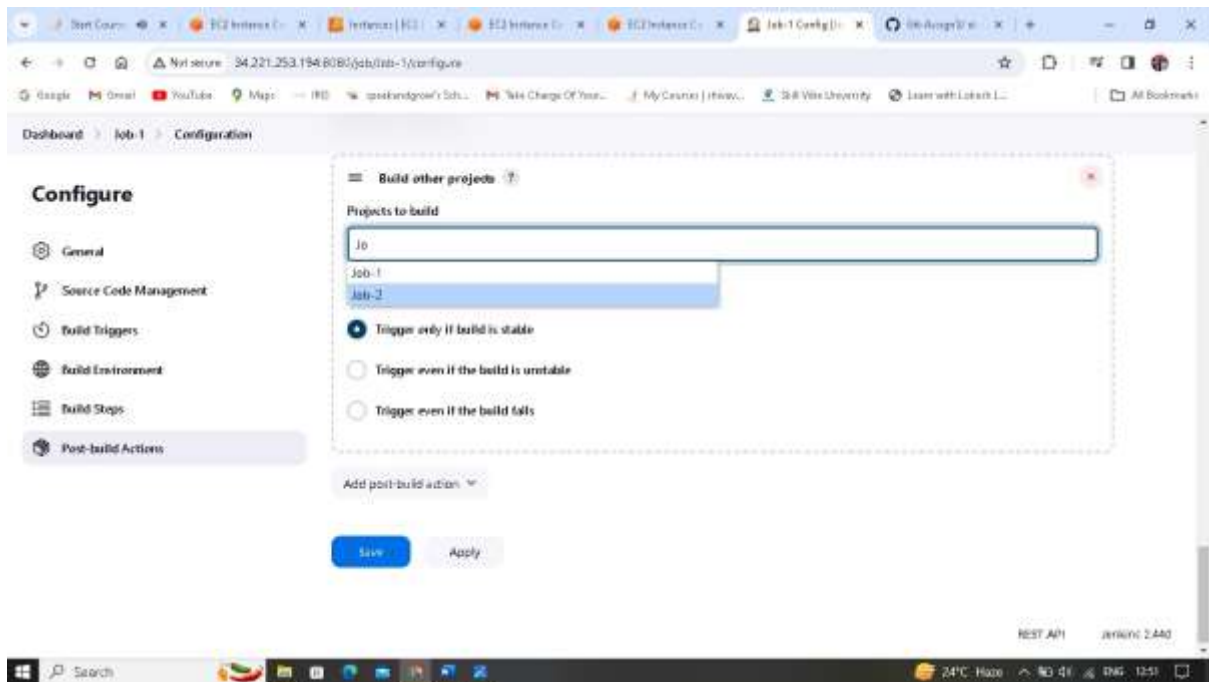




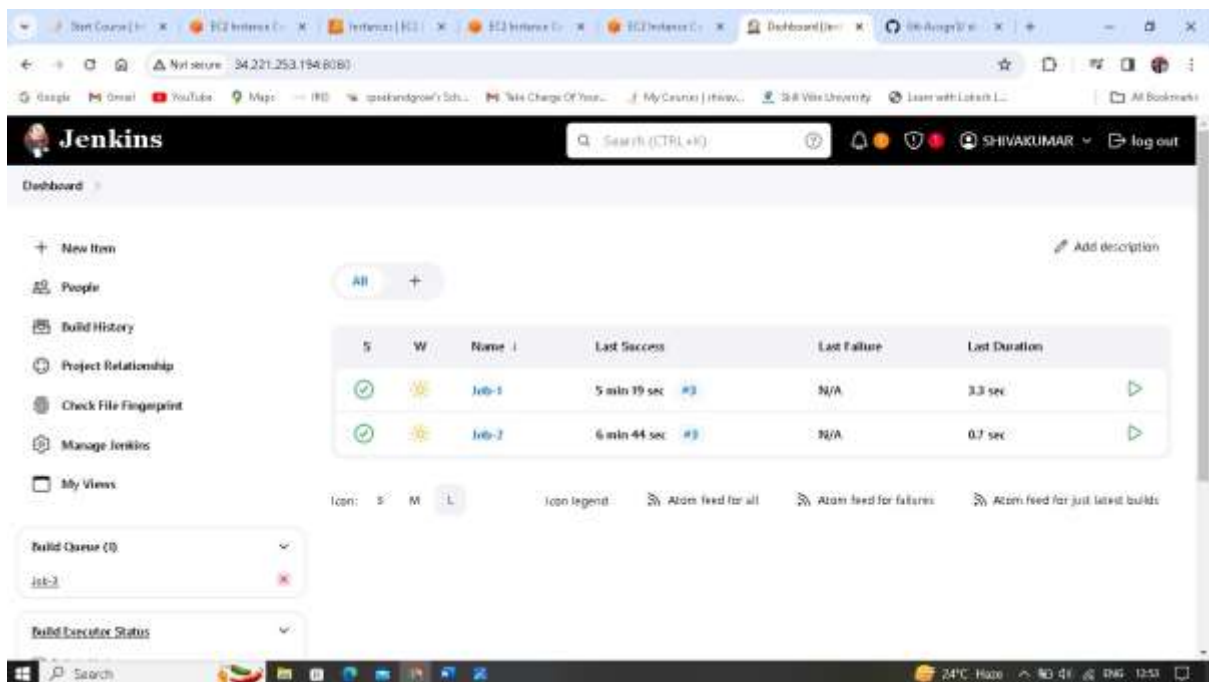
5. Choose Build other project (3<sup>rd</sup> )



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.

Dashboard [Jenkins]

SHIVAKUMAR

log out

Dashboard

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

Build Executor Status

All

S	W	Name	Last Success	Last Failure	Last Duration	
✓	☀	Job-1	0.7 sec #5	N/A	0.40 sec	▶
✓	☀	Job-2	0.3 sec #5	N/A	0.57 sec	▶

Icon: S W L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

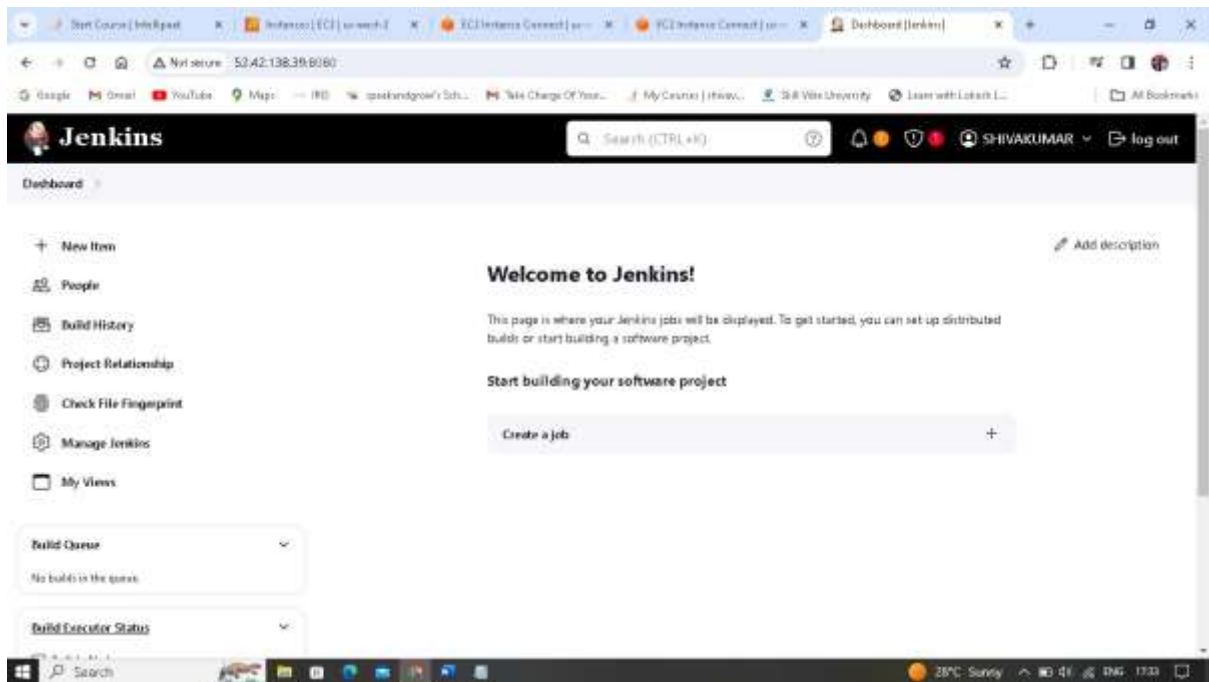
25°C: Haze

10:41

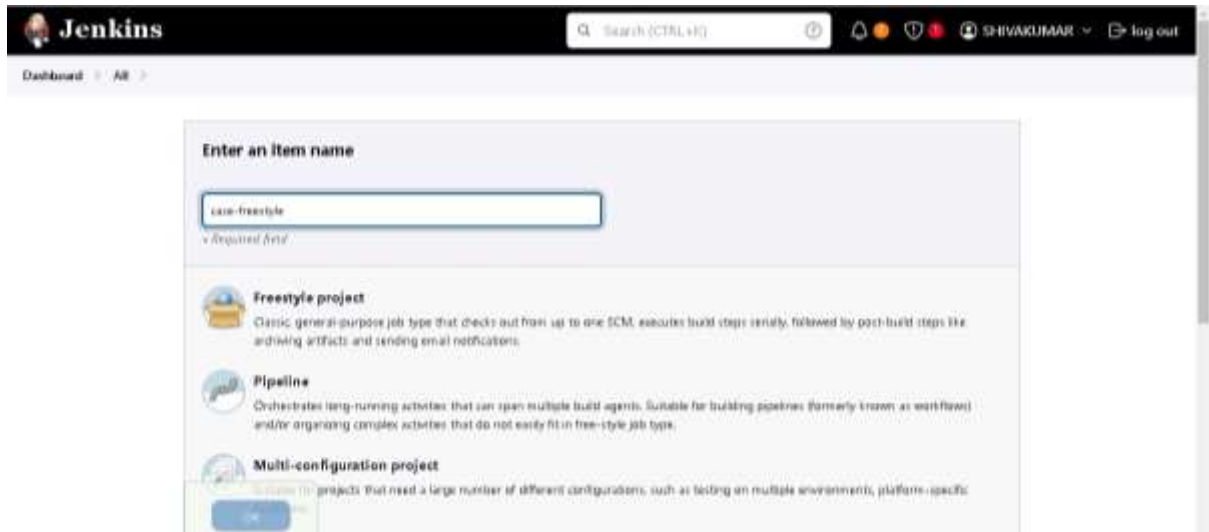
THU 10-11

# I. FOR FREESTYLE

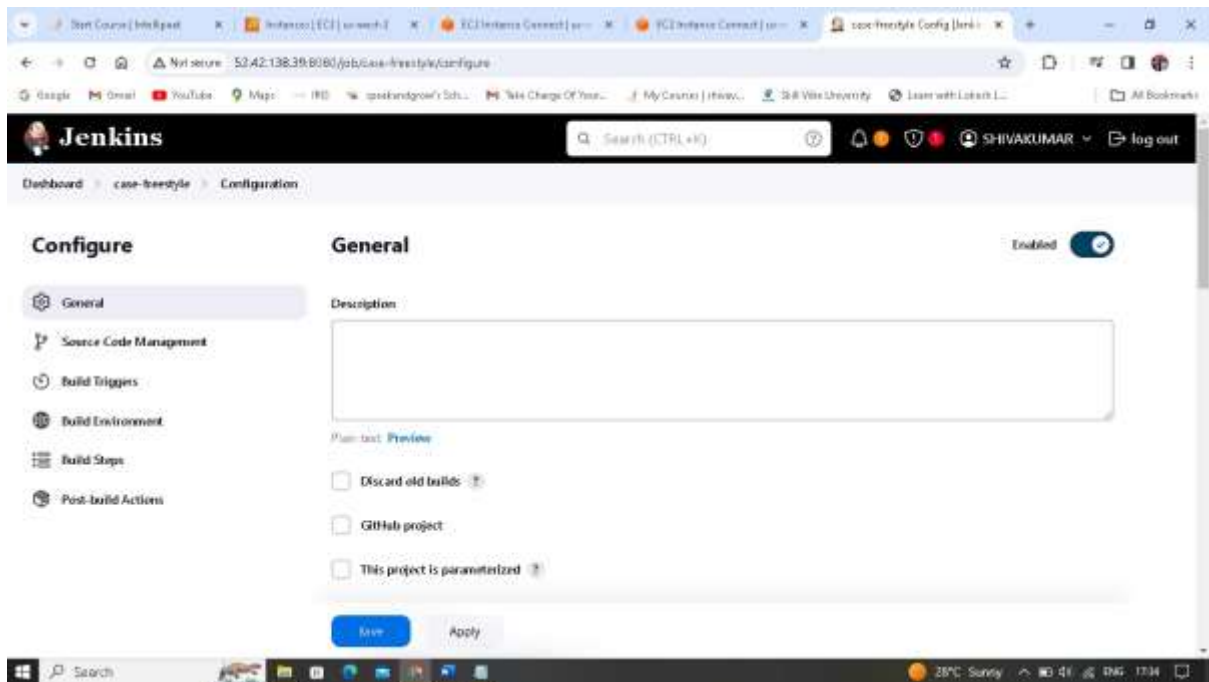
1. Go to Dashboard click on New Item.



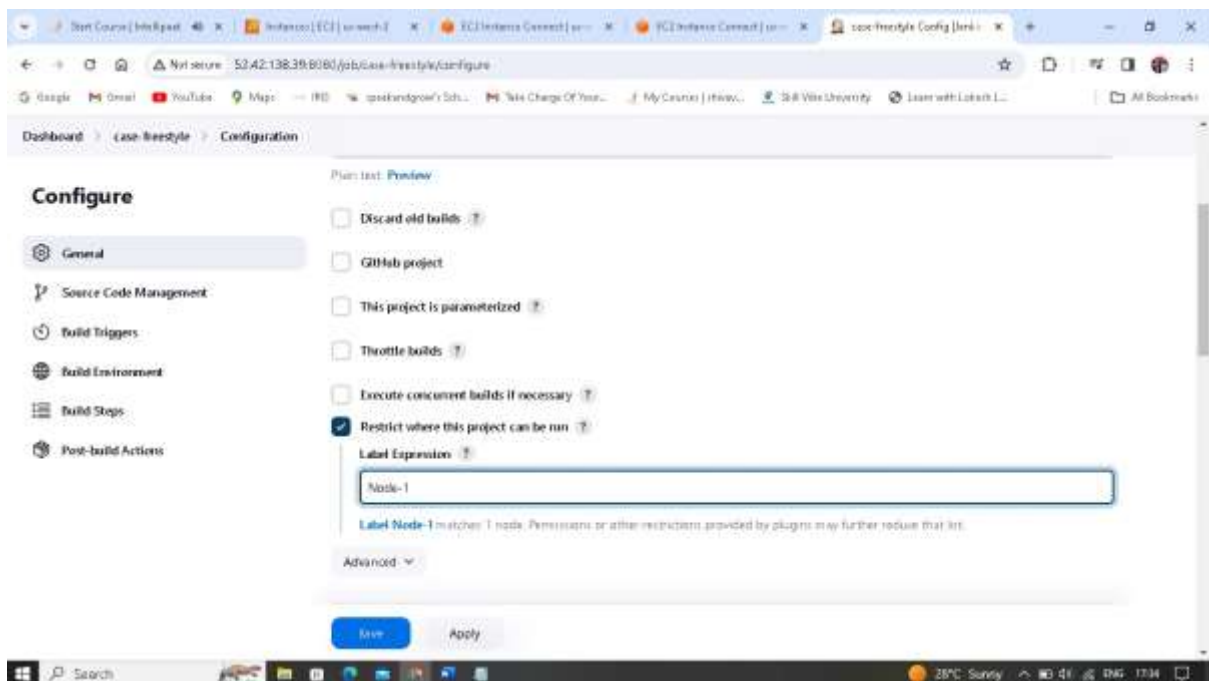
2. Enter an item name as “case-freestyle” and click on ok



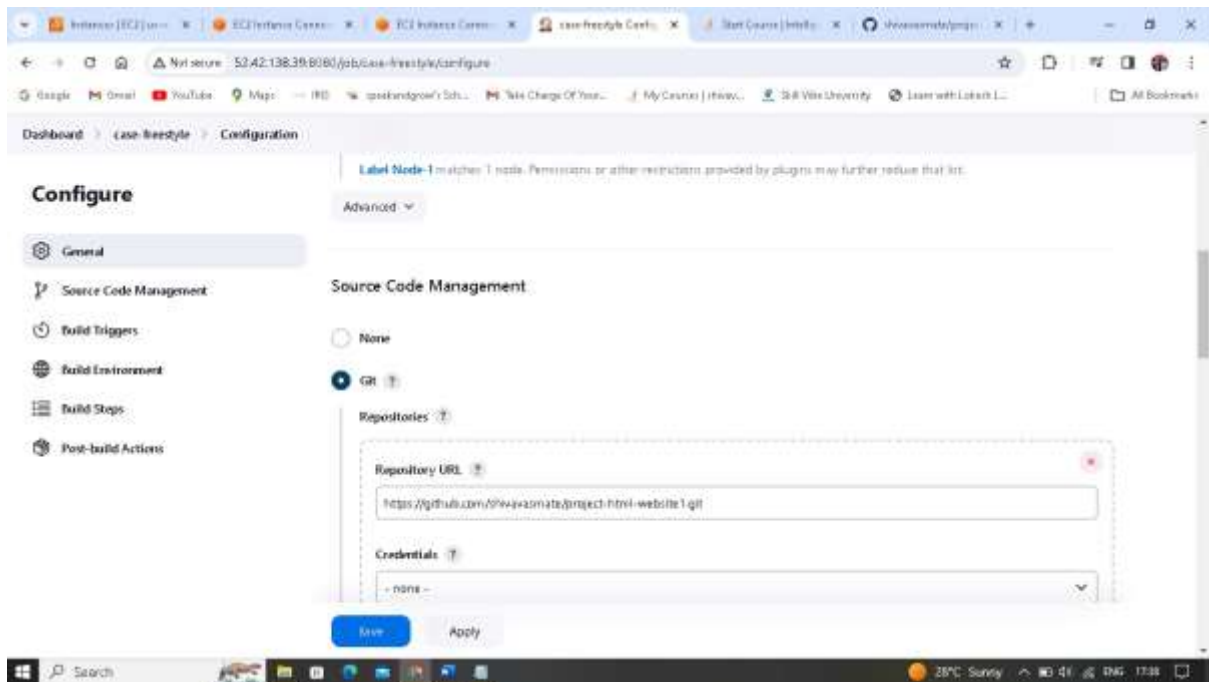
3. Fill it



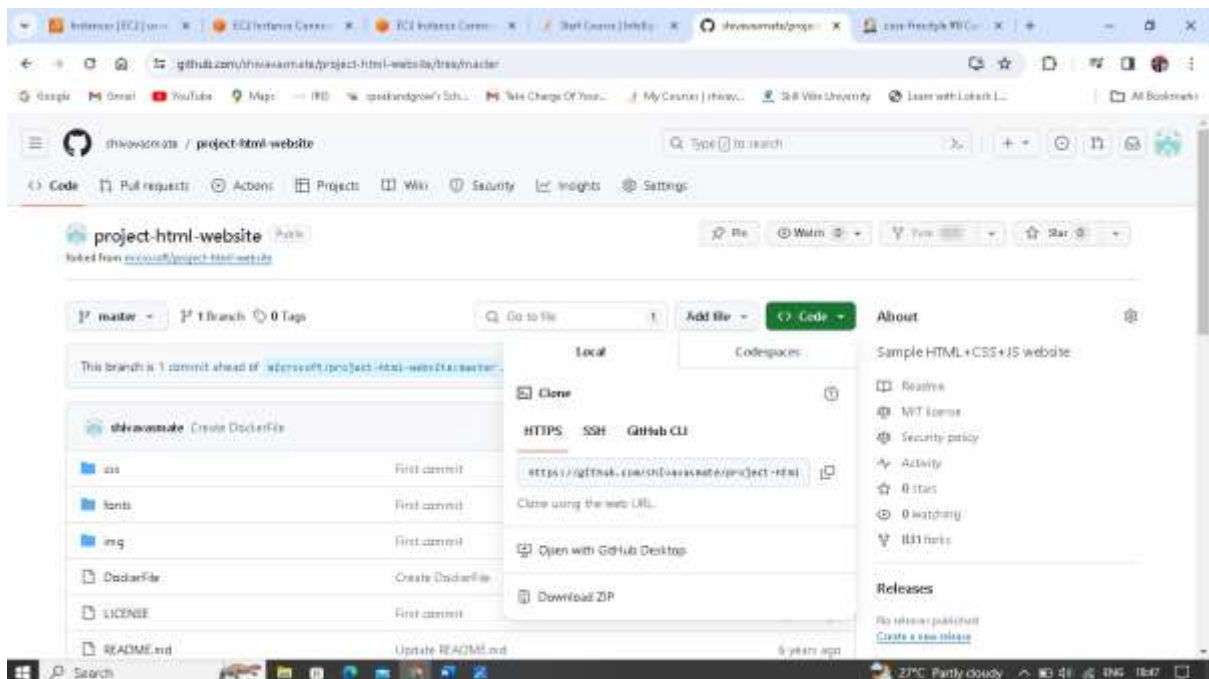
4. First of all check in Restrict where this project can be run. Label Expression as Node-1



5. Under Source Code Management Check Git



6. Now log in to your GitHub in another browser

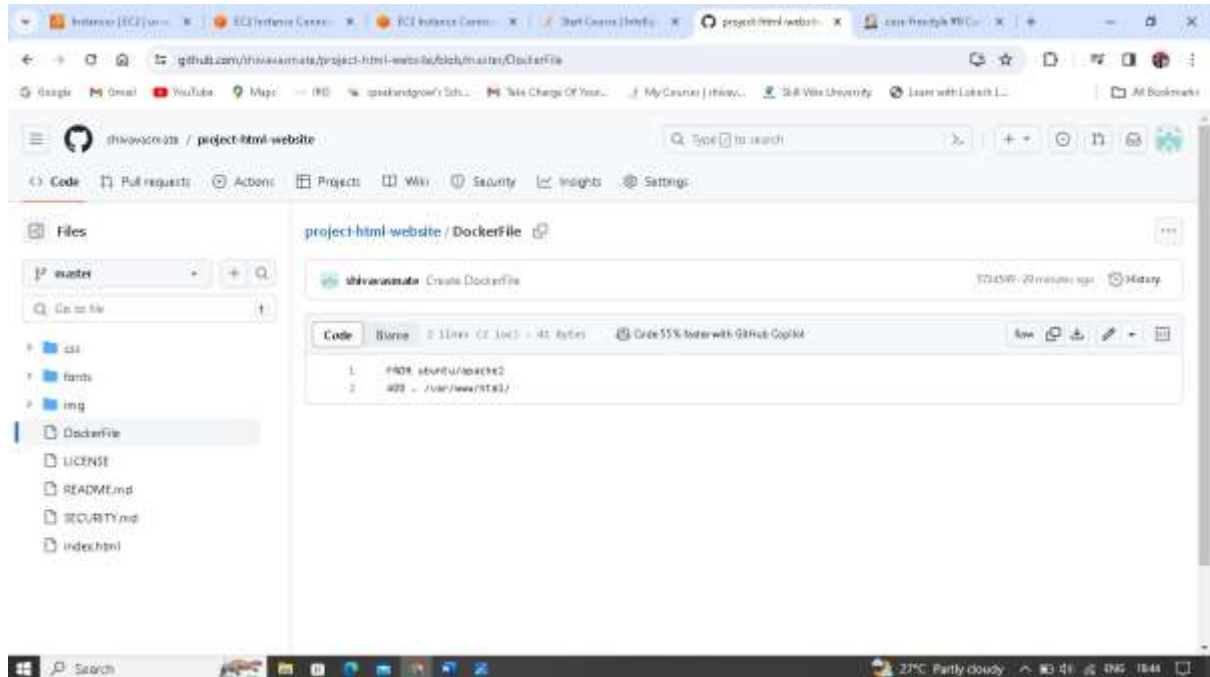




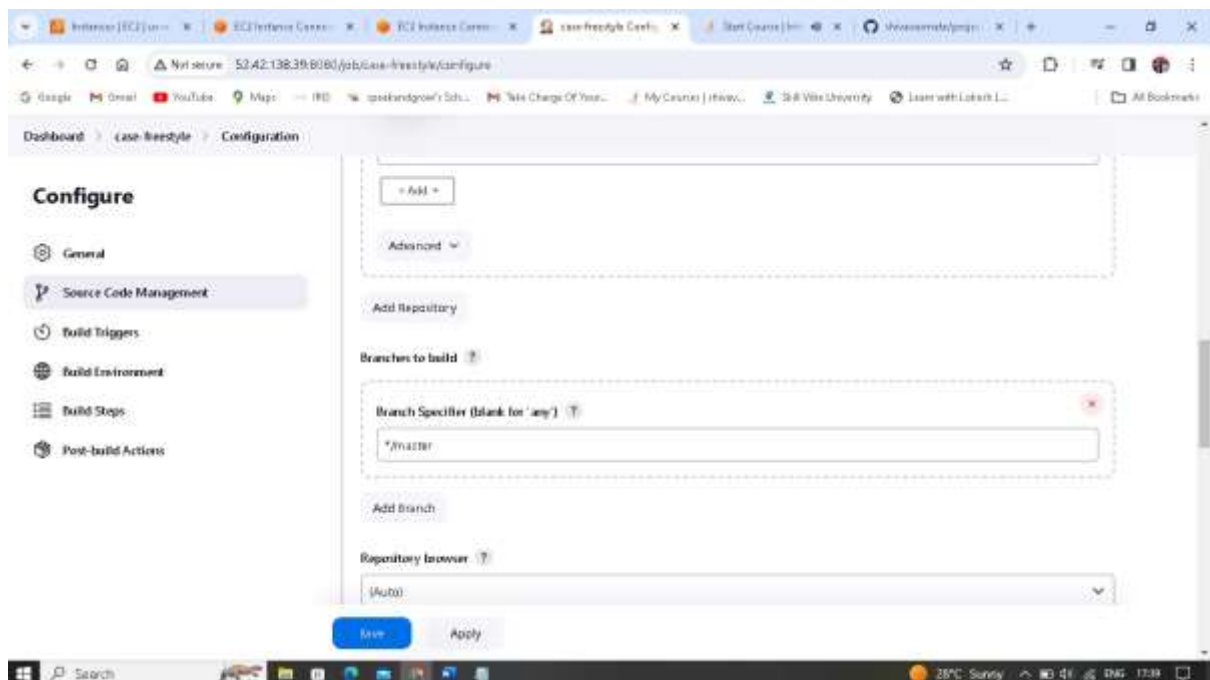
7. Create DockerFile with command

FROM ubuntu/apache2

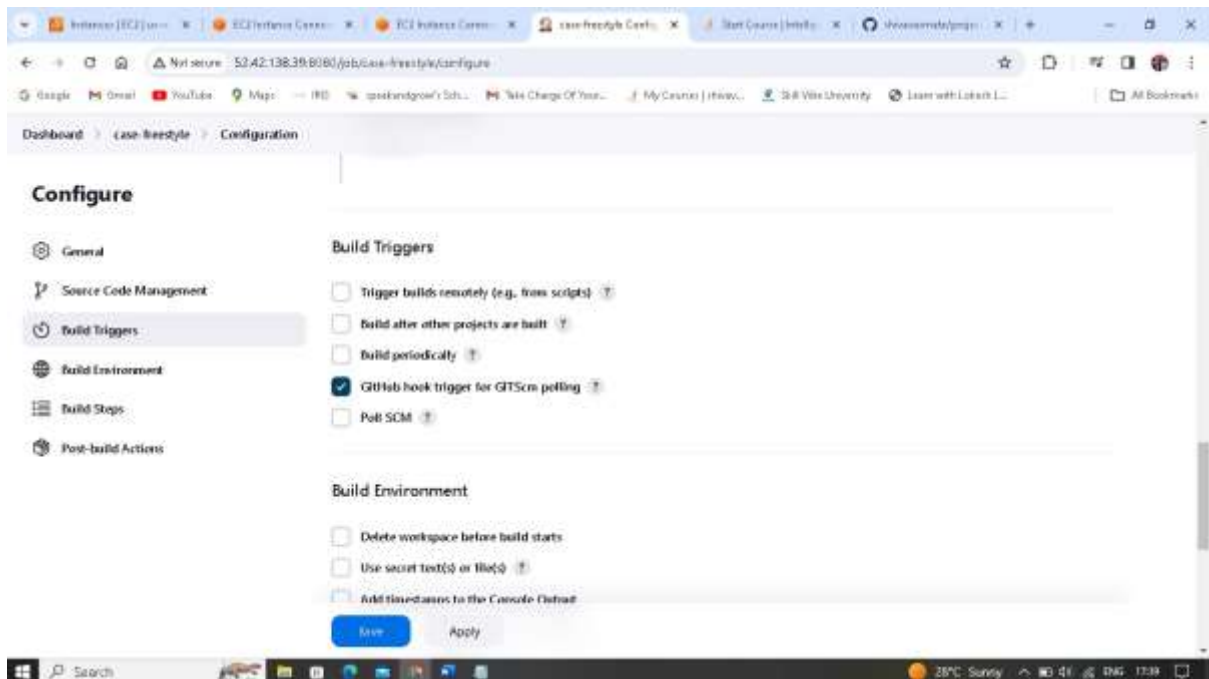
ADD ./var/www/html/ command changes.



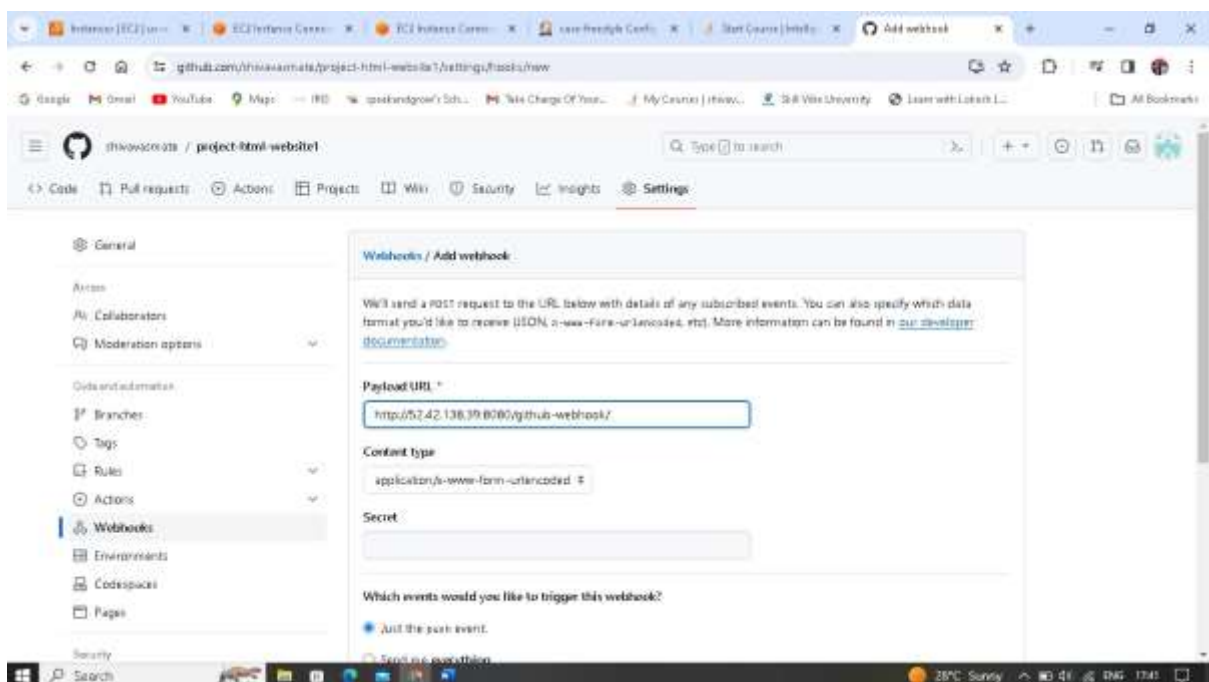
8. In Branches to build and Branch Specifier give \*/master



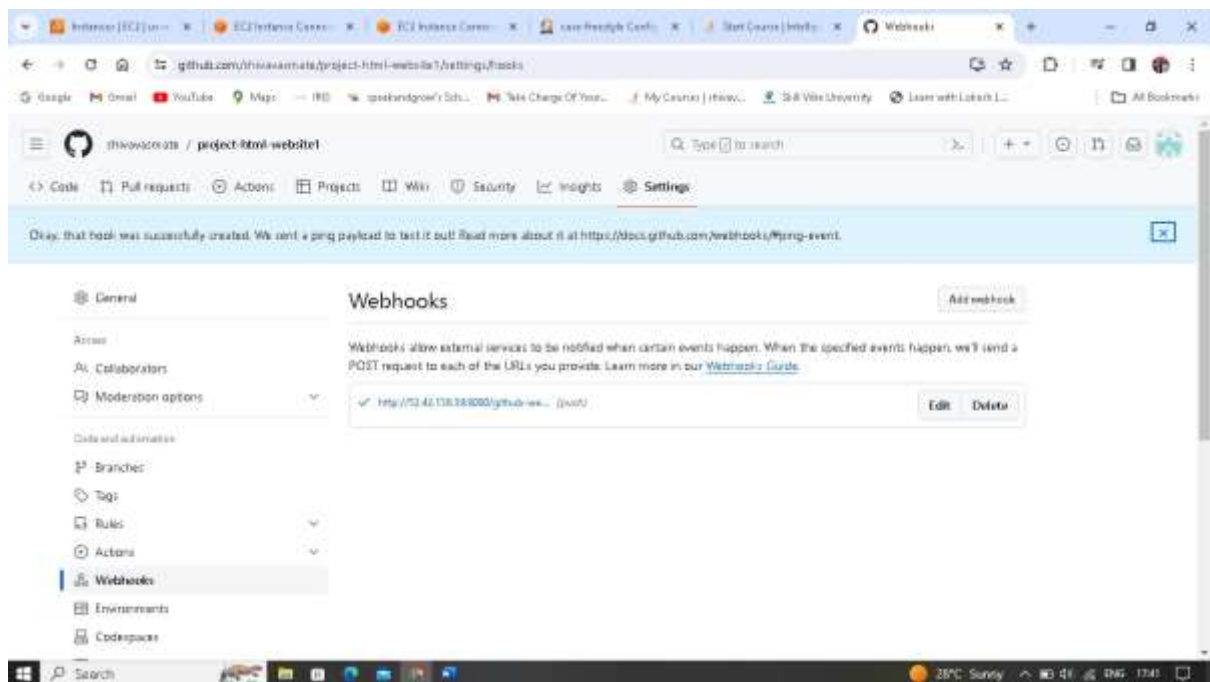
9. Under Build Triggers choose GitHub hook trigger for GIT Scm polling.



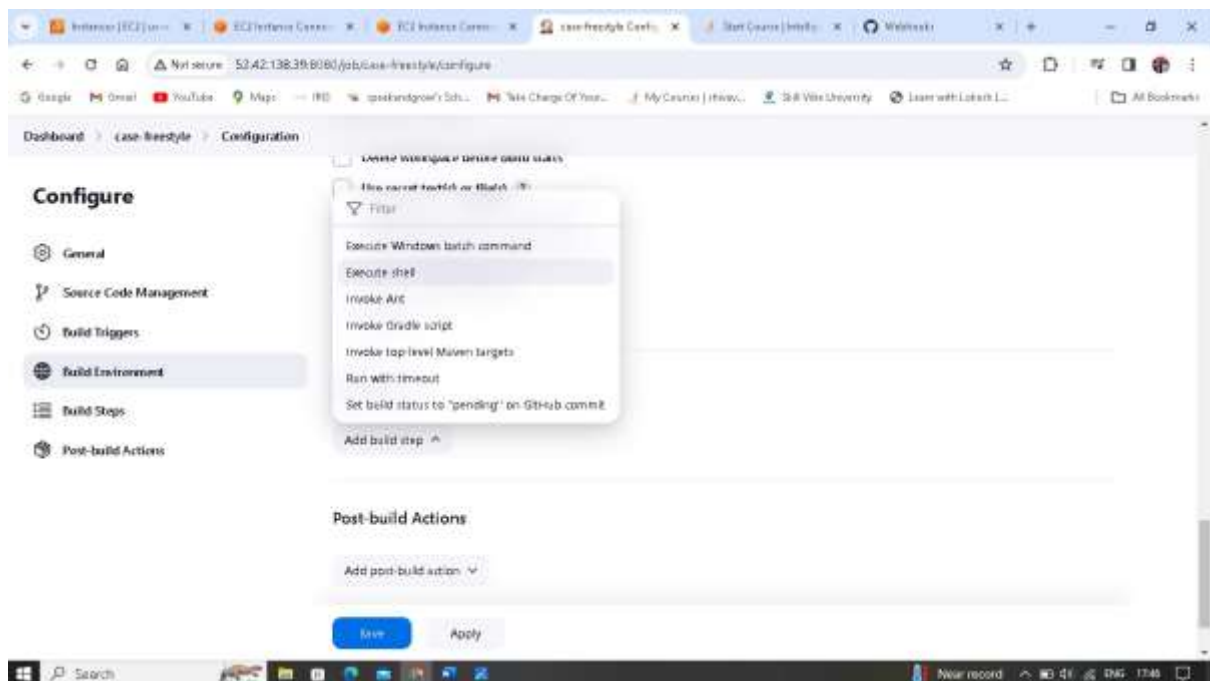
10. Now go to GitHub under repo settings look for Webhooks on left side



## 11. Check for right mark



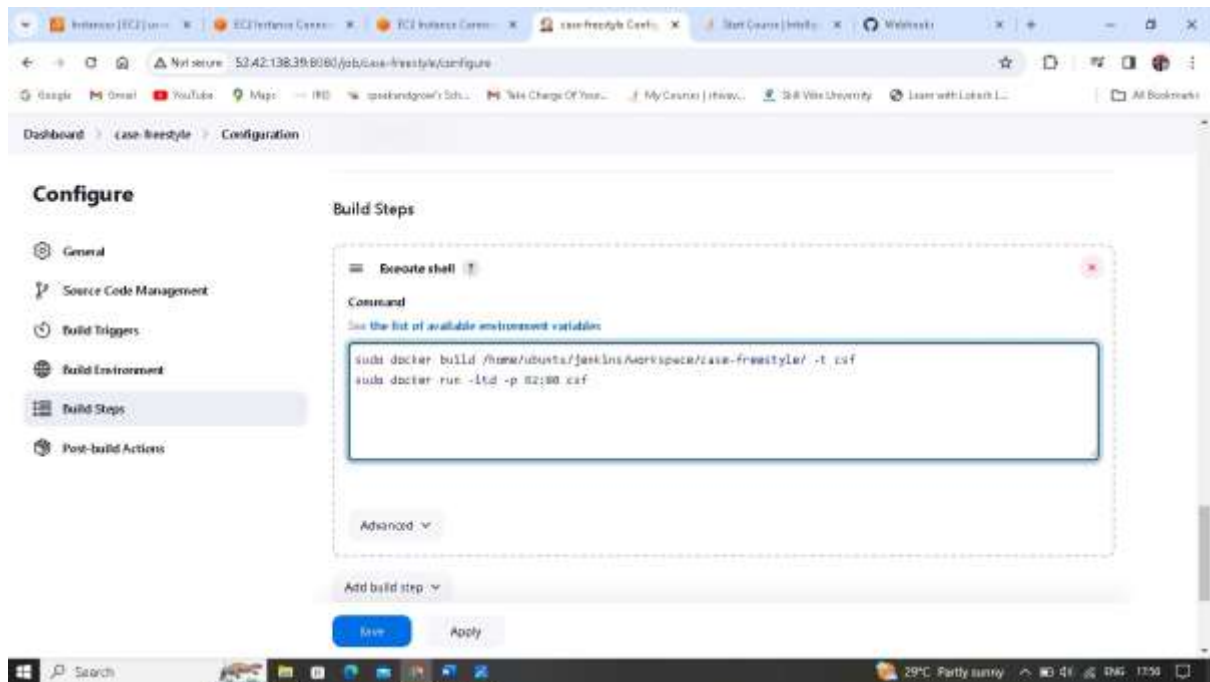
## 12. Under Build Steps click on Execute shell



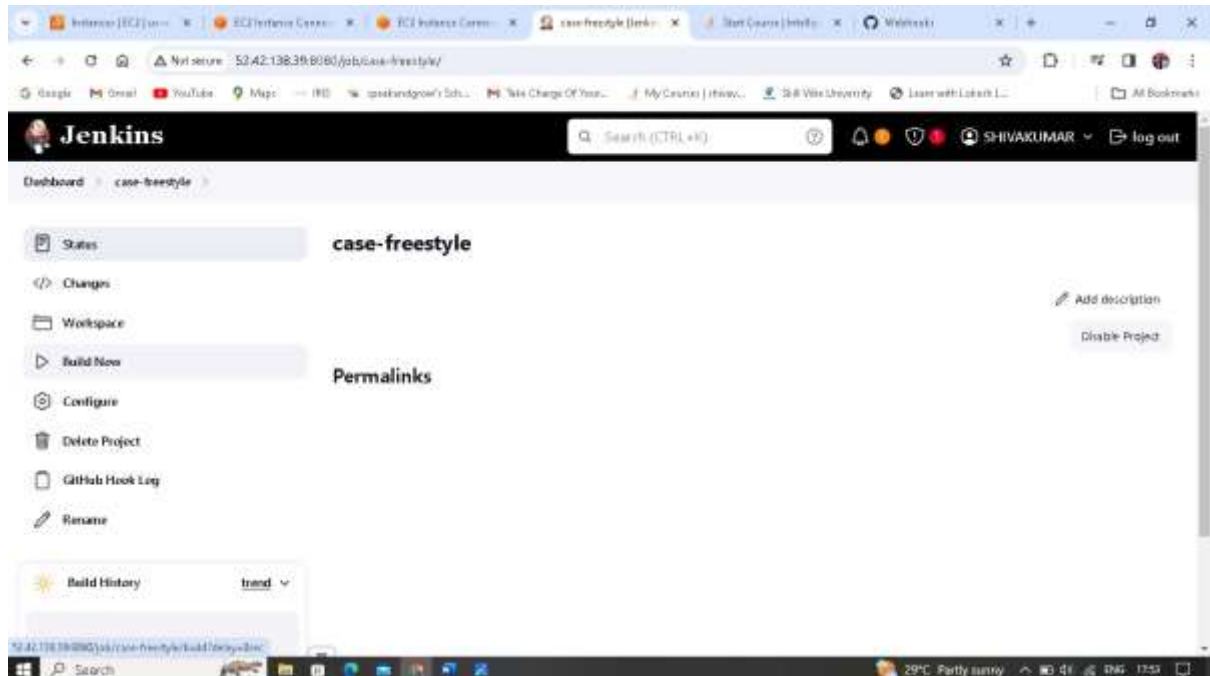
## 13. Under Command type

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

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



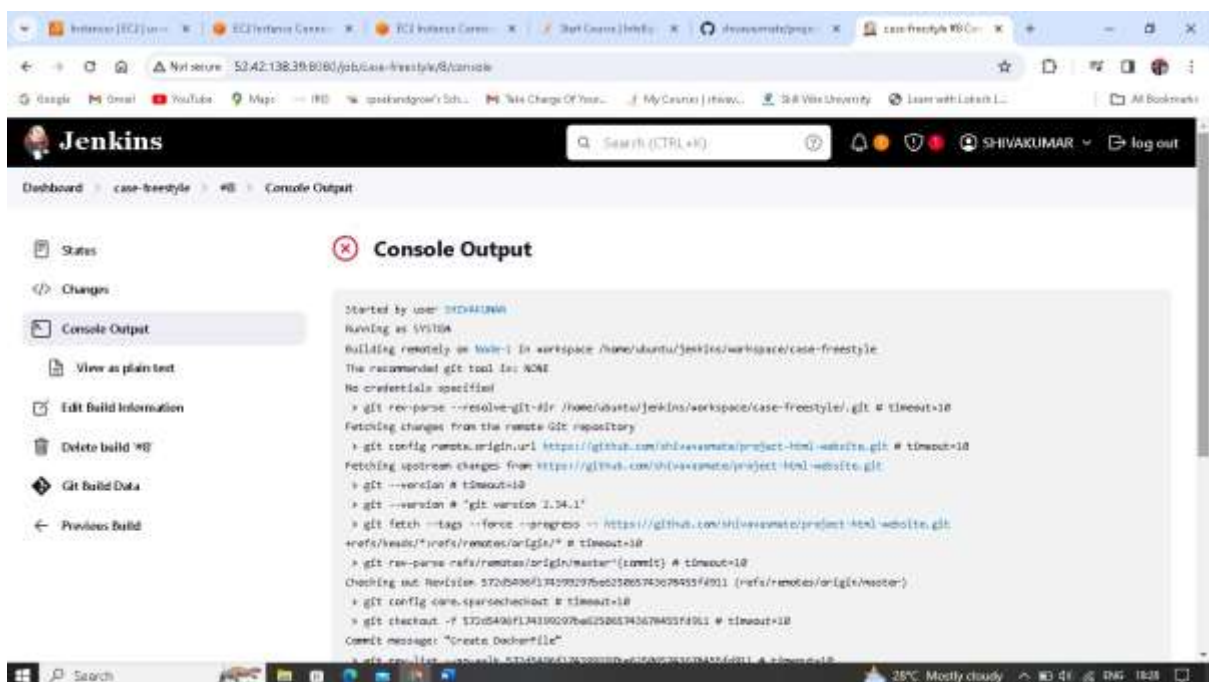
14.The case-freestyle is created



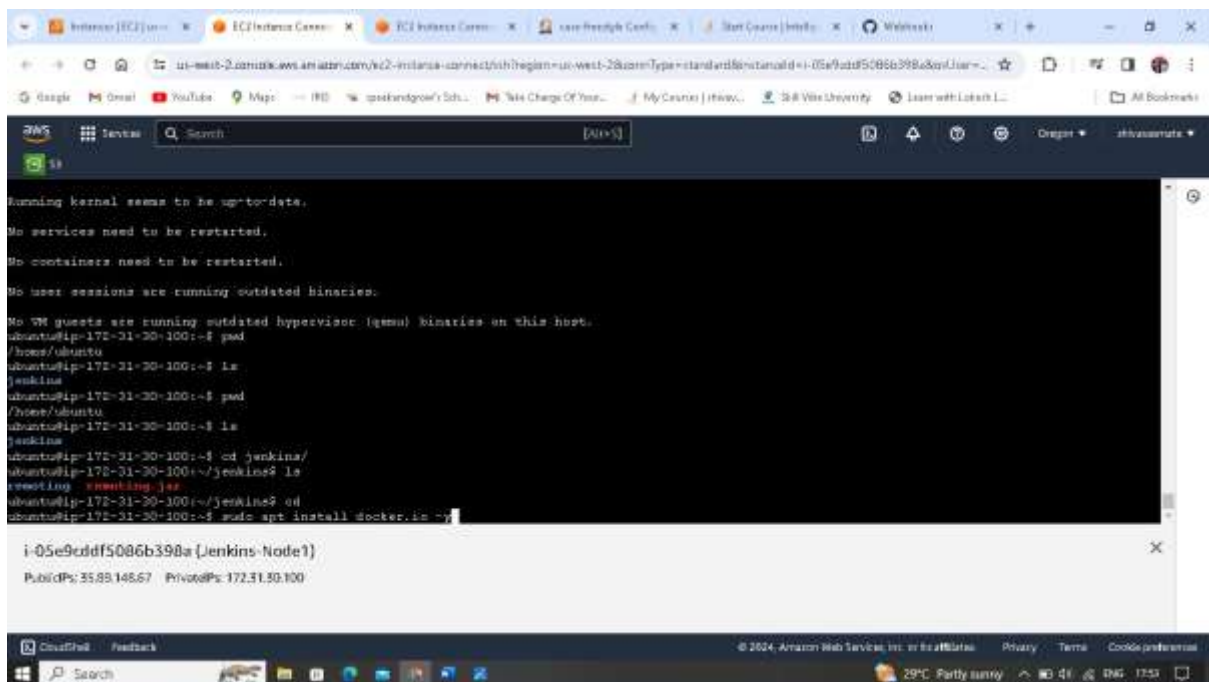
## 15. Click on Build Now



## 16. Check for error



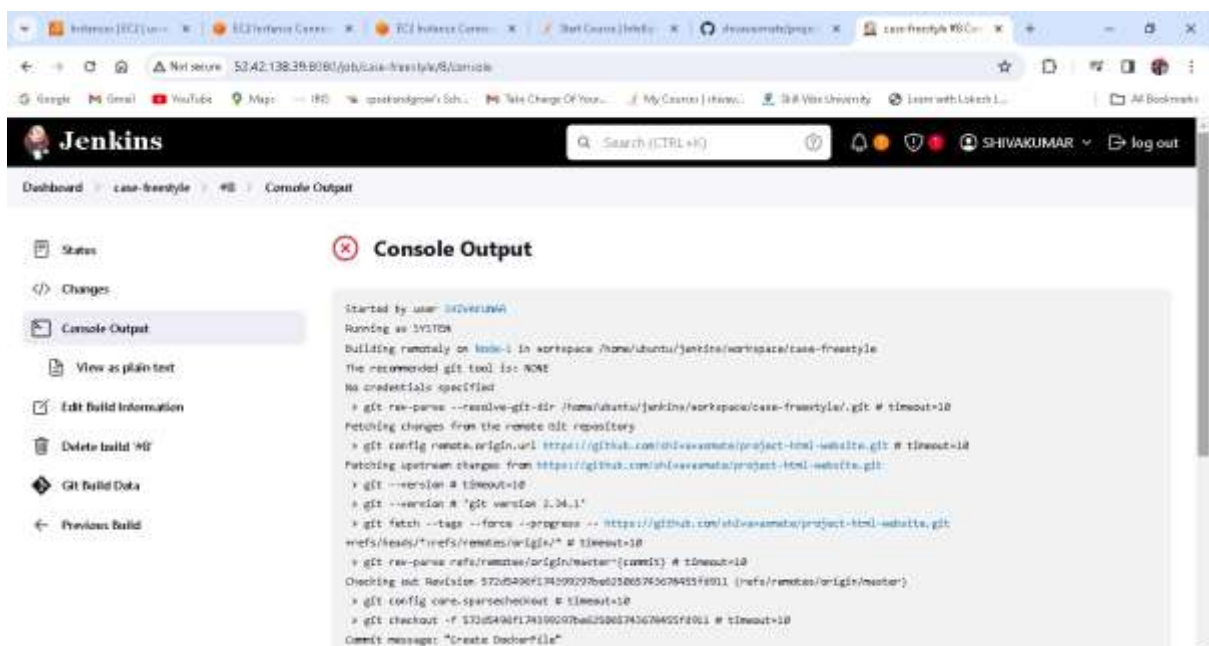
## 17. Install docker in Jenkins-Node1 using the command sudo apt install docker.io -y



The screenshot shows a terminal window on an AWS EC2 instance. The terminal output indicates that the kernel is up-to-date, no services need to be restarted, and no containers are running. The user then runs the command `sudo apt install docker.io -y` to install Docker. The installation is successful, and the user is prompted to press `ENTER` to continue. The terminal also shows the instance ID `i-05e9cddf5086b398a` and the public IP `35.89.148.67`.

```
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
ubuntu@ip-172-31-30-100:~$ ls
jenkins
ubuntu@ip-172-31-30-100:~$ ps
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...creating.jar
ubuntu@ip-172-31-30-100:~/jenkins$ cd
ubuntu@ip-172-31-30-100:~$ sudo apt install docker.io -y
i-05e9cddf5086b398a (Jenkins-Node1)
PublicIPs: 35.89.148.67 PrivateIPs: 172.31.30.100
```

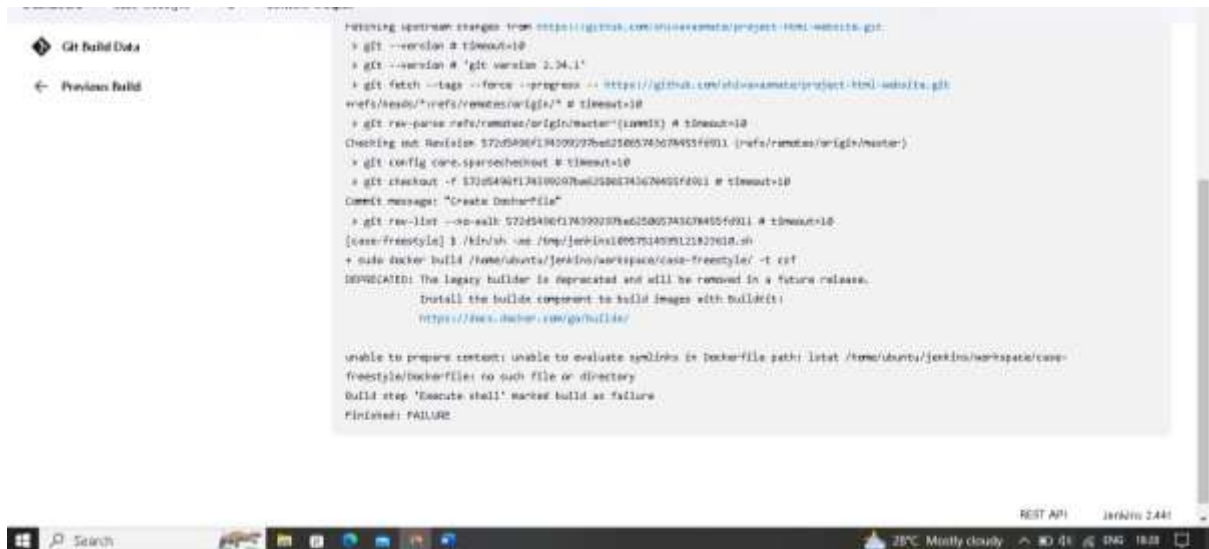
## 18. Build now again and check for error



The screenshot shows the Jenkins web interface. The 'Console Output' tab is selected, displaying the output of a build. The output shows that the build was started by user 'shivakumar' and is running as 'SVN'. It then shows the steps of the build, including fetching changes from the remote git repository, checking out the code, and creating a Dockerfile. The build is currently in a 'Building' state.

```
Started by user shivakumar
Running as SVN
Building remotely on node1 in workspace /home/ubuntu/jenkins/workspace/case-freestyle
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-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 --tags --force --progress -- https://github.com/shivakumar/project-html-website.git
refs/heads/*>refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master:{commit} # timeout=10
Checking out Revision: 572d5496f174599277b0d5065745678455f8011 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 572d5496f174599277b0d5065745678455f8011 # timeout=10
Commit message: "Create Dockerfile"
```





```
Fetching upstream changes from https://github.com/shivaraman/project-html-website.git:
+ git --version # timeout=10
+ git --version # 'git version 2.34.1'
+ git fetch --tags --force --progress -- https://github.com/shivaraman/project-html-website.git
refs/heads/*refs/remotes/origin/* # timeout=10
+ git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision: 572d5496f19400077b6d2505743c78455f9011 (refs/remotes/origin/master)
+ git config core.sparsecheckout # timeout=10
+ git checkout -f 572d5496f19400077b6d2505743c78455f9011 # timeout=10
Commit message: "Create Dockerfile"
+ git rev-list --topo-sort --no-walk 572d5496f19400077b6d2505743c78455f9011 # timeout=10
[case-freestyle] $ /bin/sh -m /tmp/jenkins16057514029121927618.sh
+ sudo docker build /home/ubuntu/jenkins/workspace/case-freestyle/ -t rst
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Docker the buildx component to build images with buildx:
https://docs.docker.com/buildx/

unable to prepare context: unable to evaluate symlinks in Dockerfile path: lsat /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: lsat  
/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

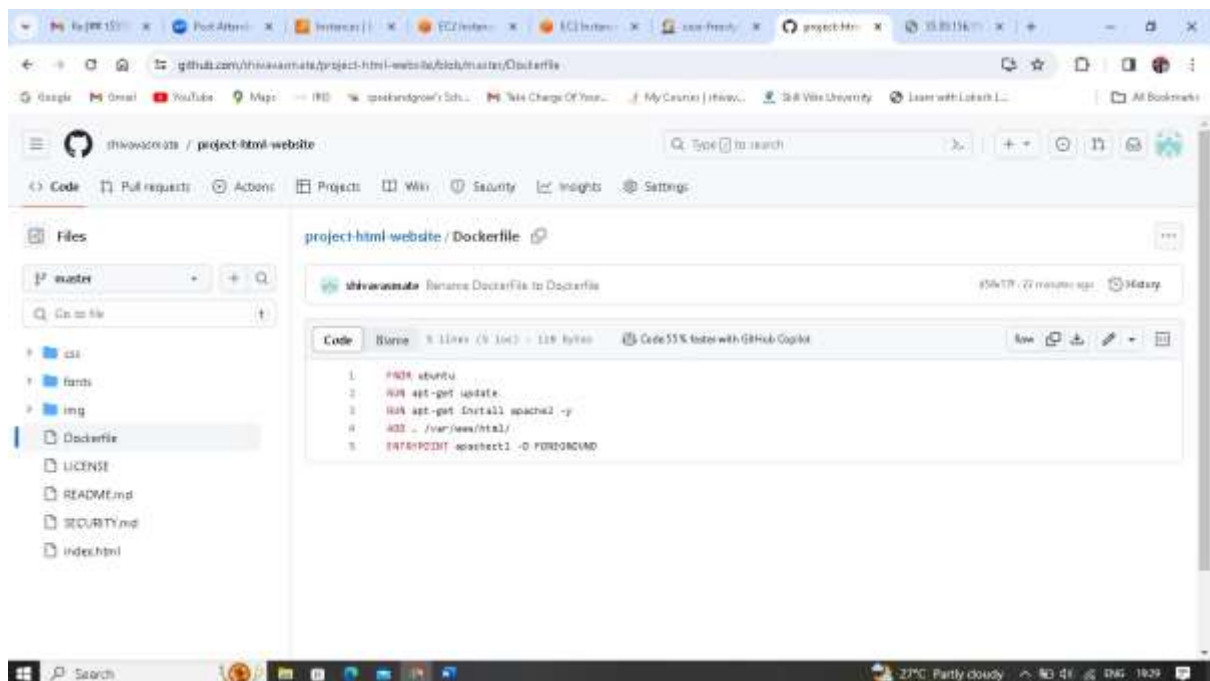
b. Update Dockerfile as  
FROM ubuntu

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
```

```
cache  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 apache2l -D FOREGROUND
```

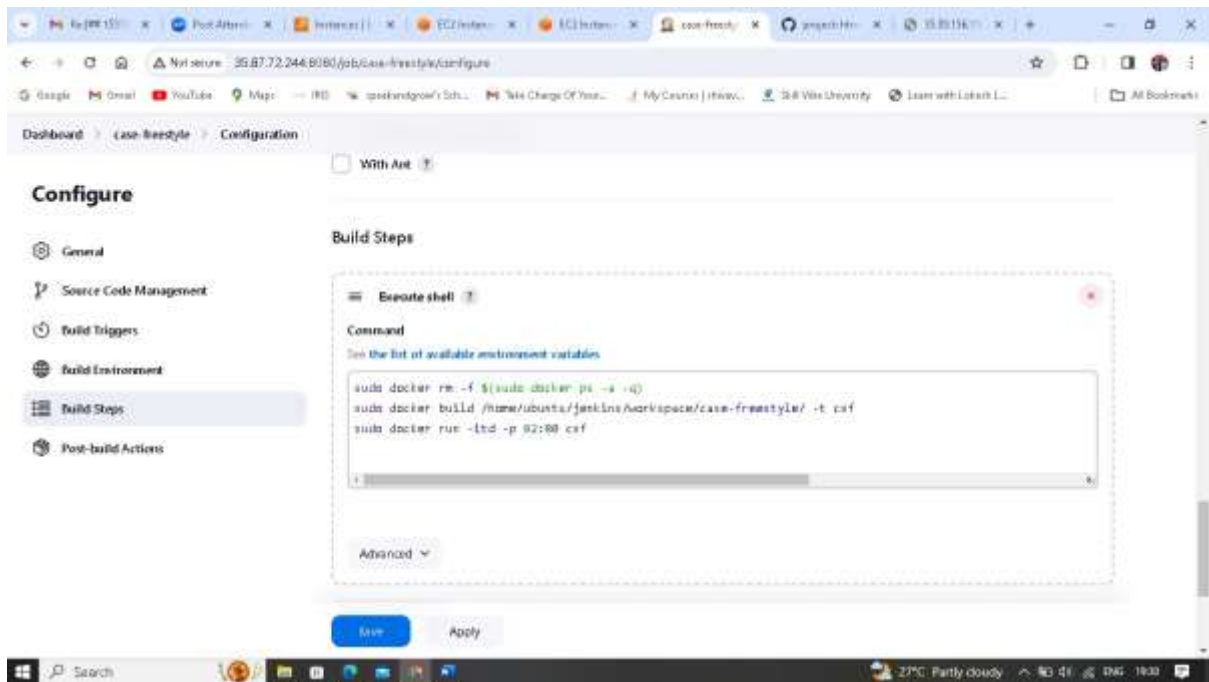
22. Update with command

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

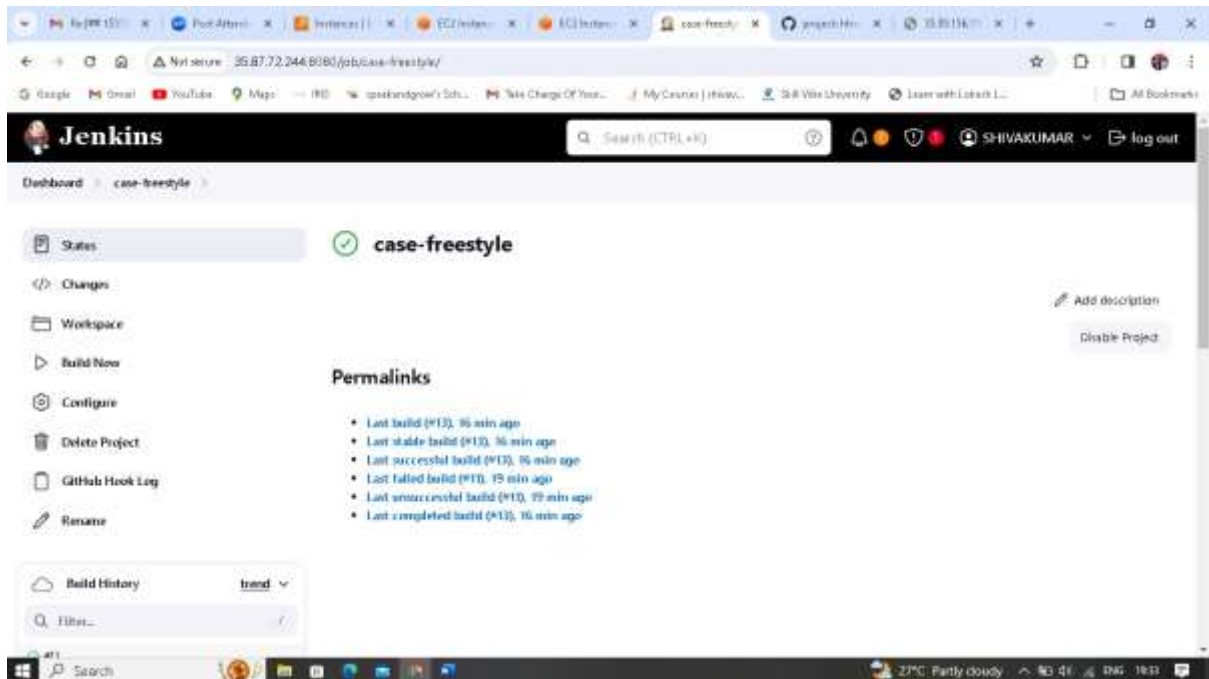
```
sudo docker build /home/ubuntu/jenkins/workspace/case-freestyle/ -t csf
```

```
sudo docker run -itd -p 82:80 csf
```

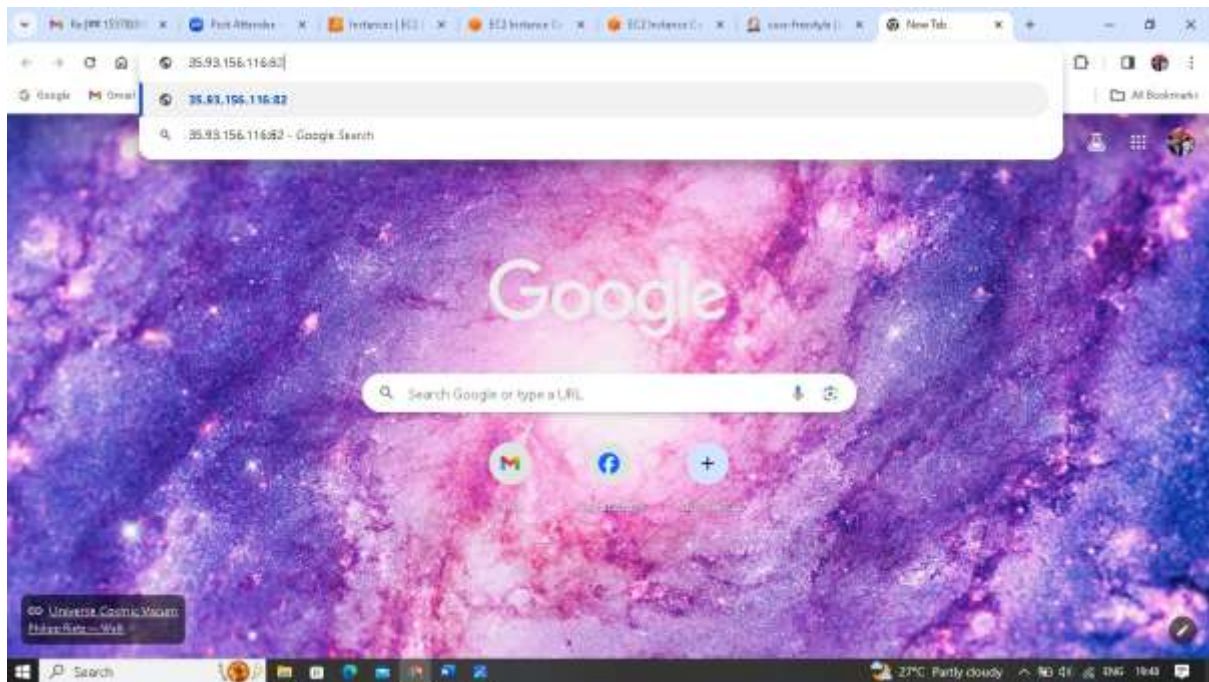
Click on apply and save.



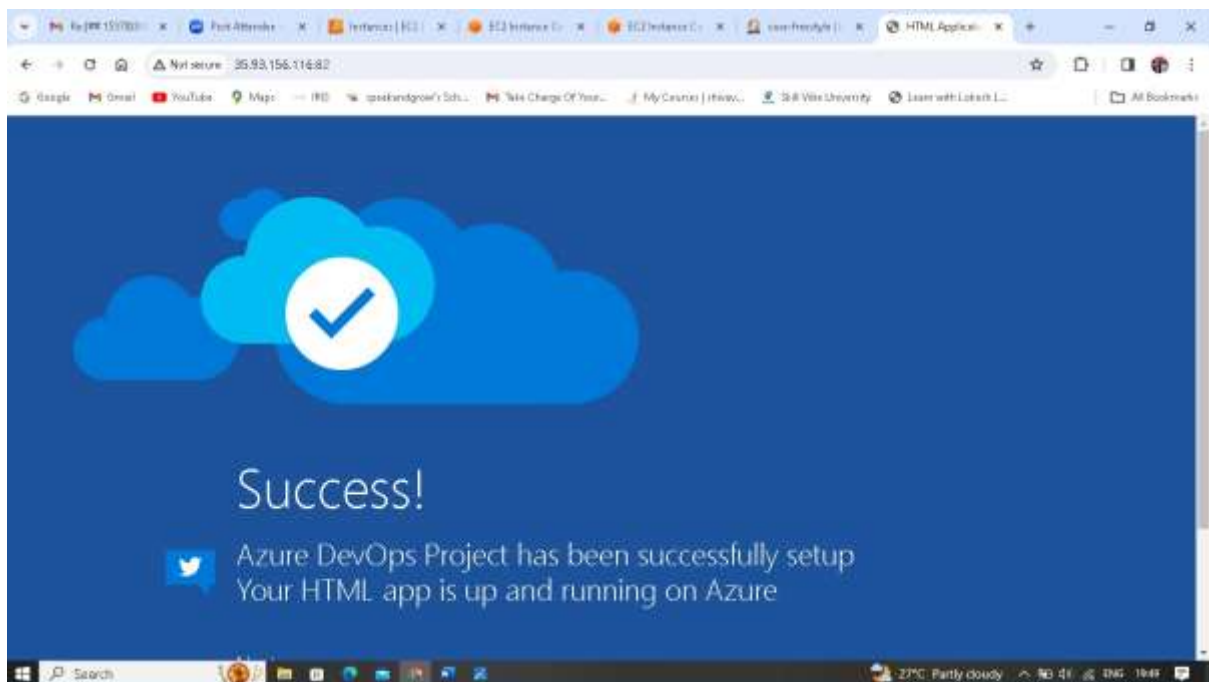
23. Click on Build Now. This time it is successful



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

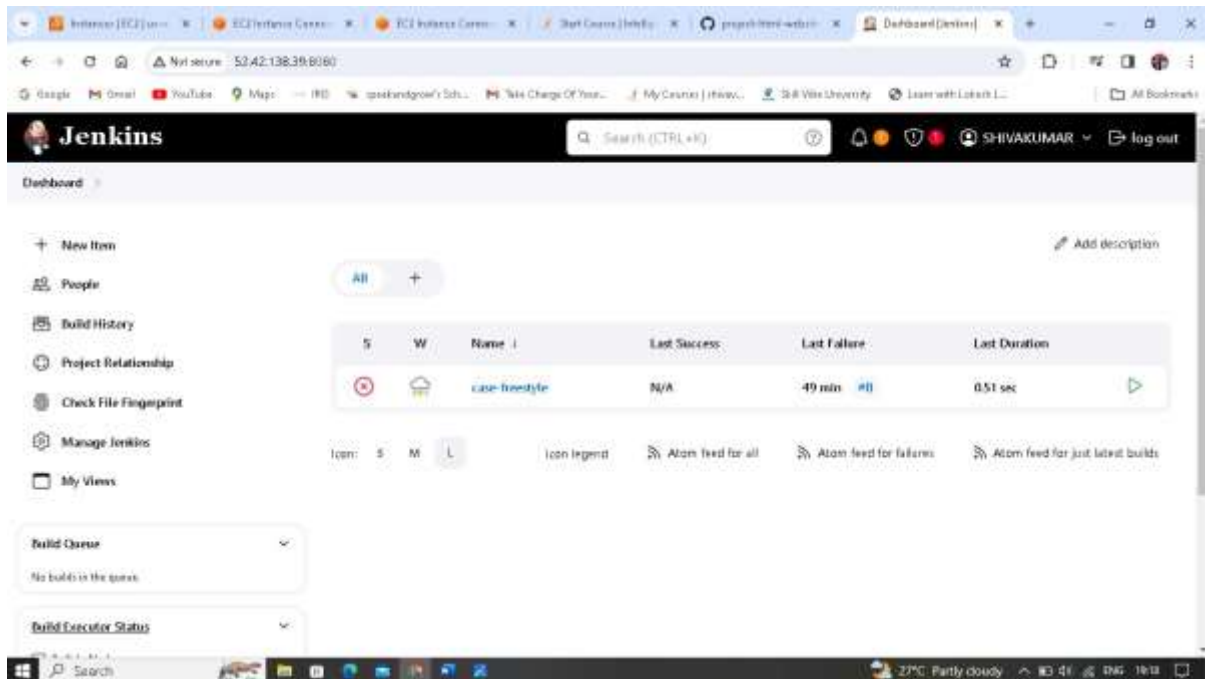


25. Now HTML Application opens.

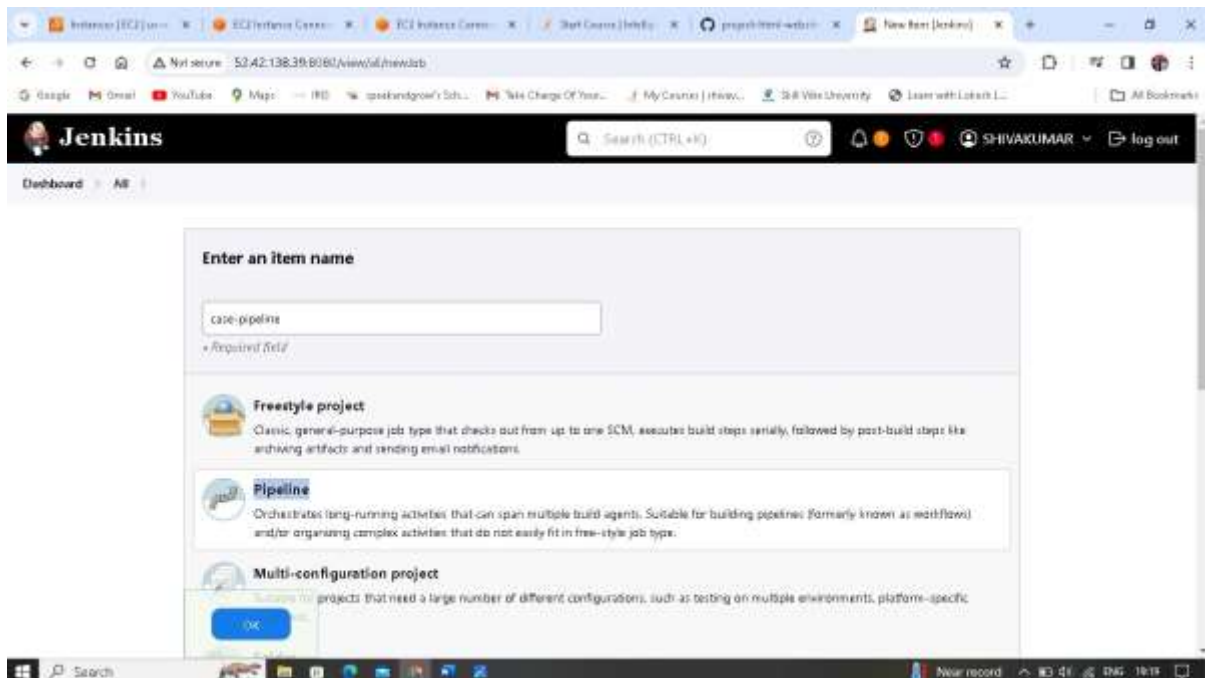


### 3. FOR PIPELINE

1. Go to Dashboard click on New Item.

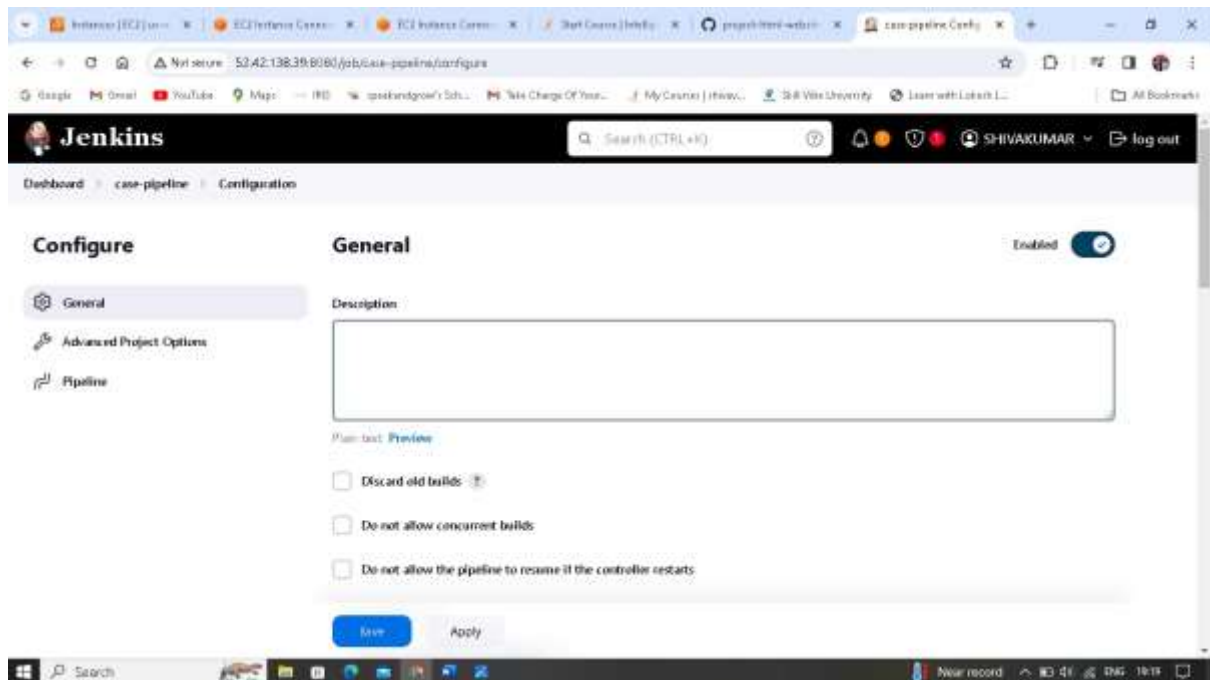


2. Enter an item name as “case-pipeline” click on pipeline and ok



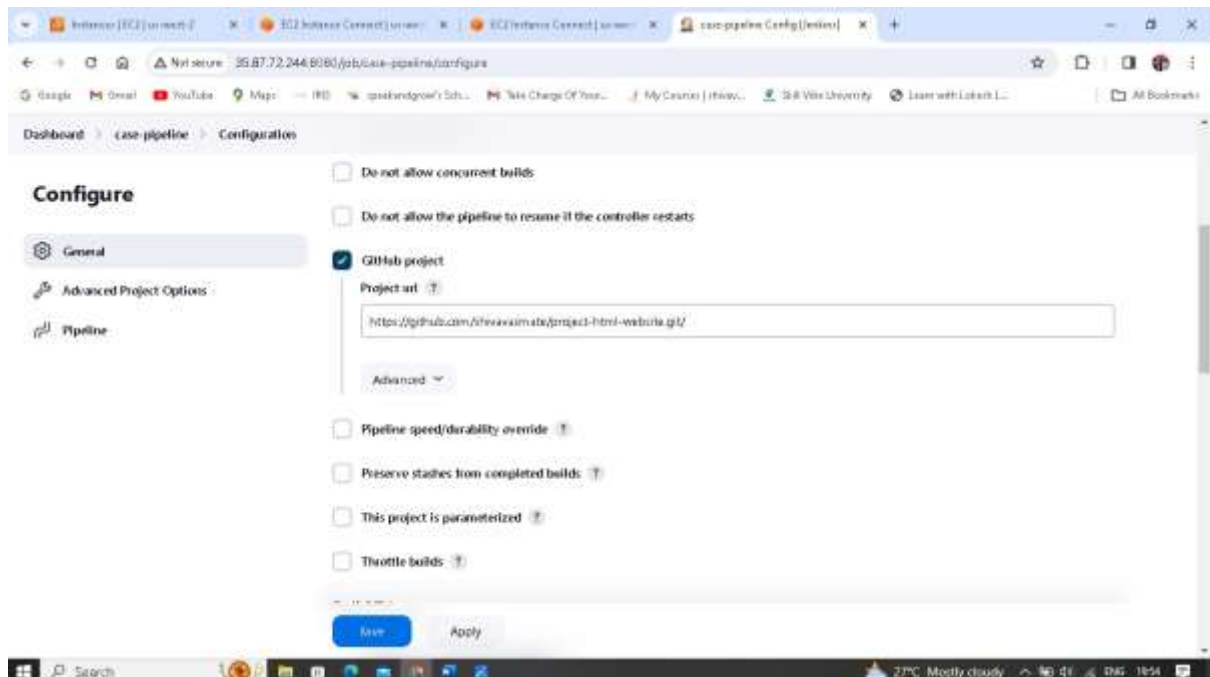


### 3. Fill it

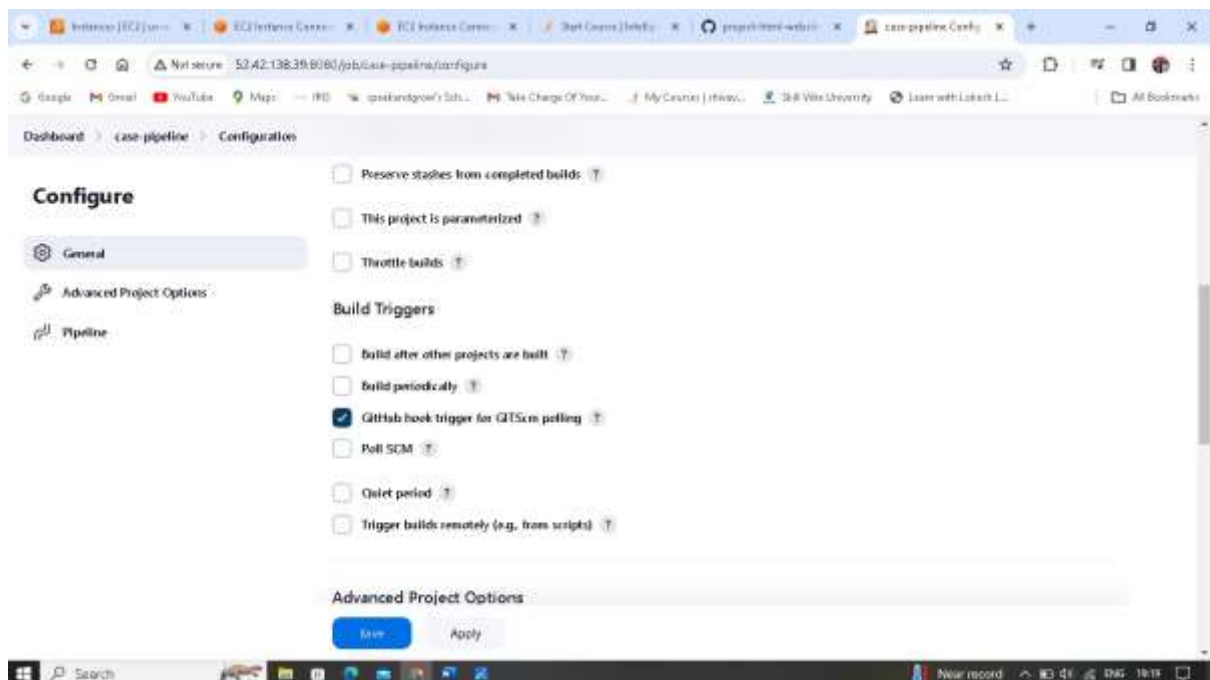


### 4. Check in GitHub Project

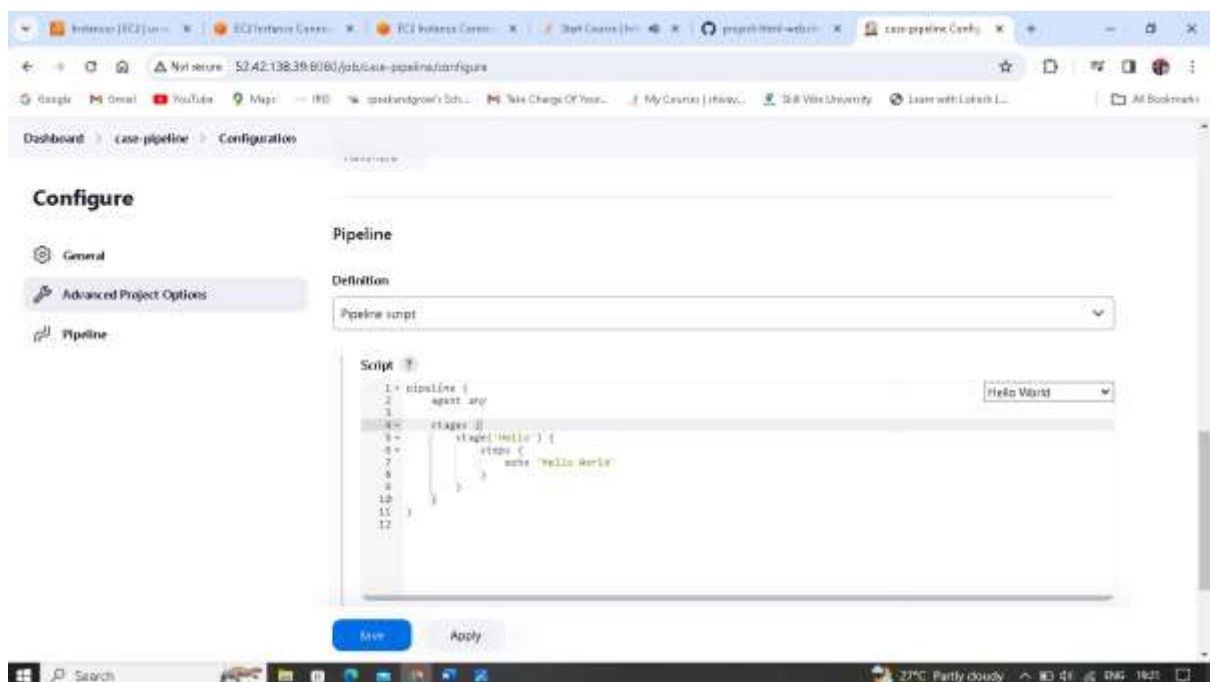
Project url <https://github.com/shivavasmate/project-html-website.git/>



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”

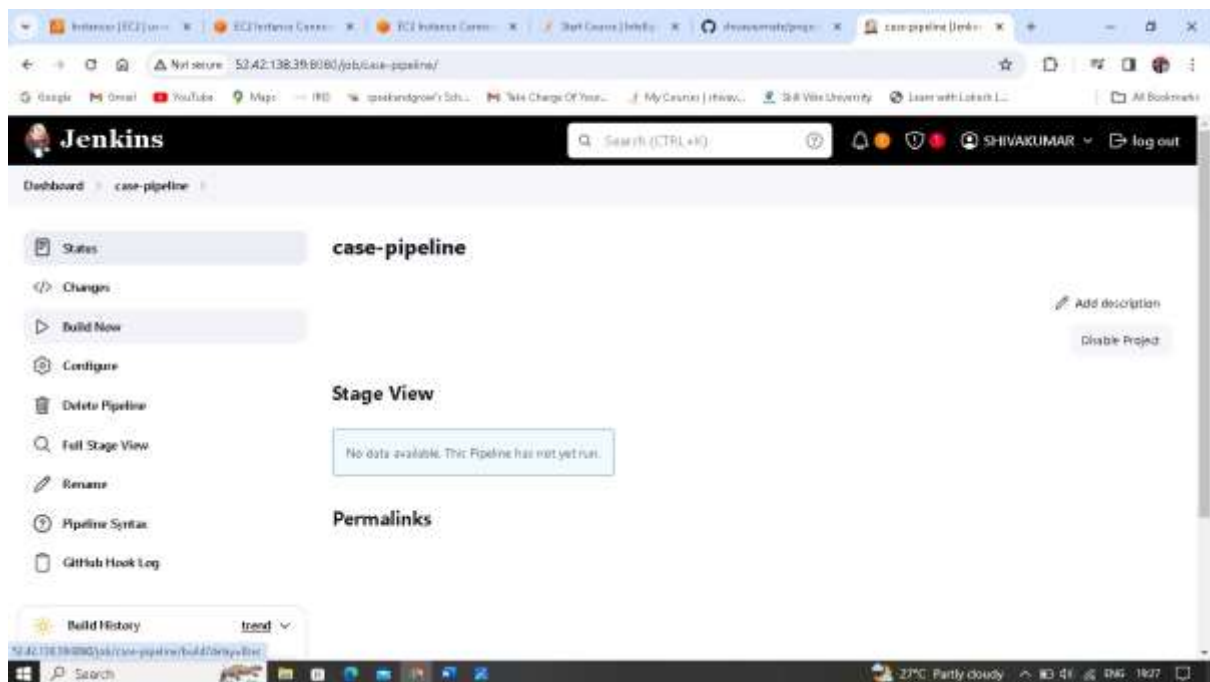


7. Edit the script as

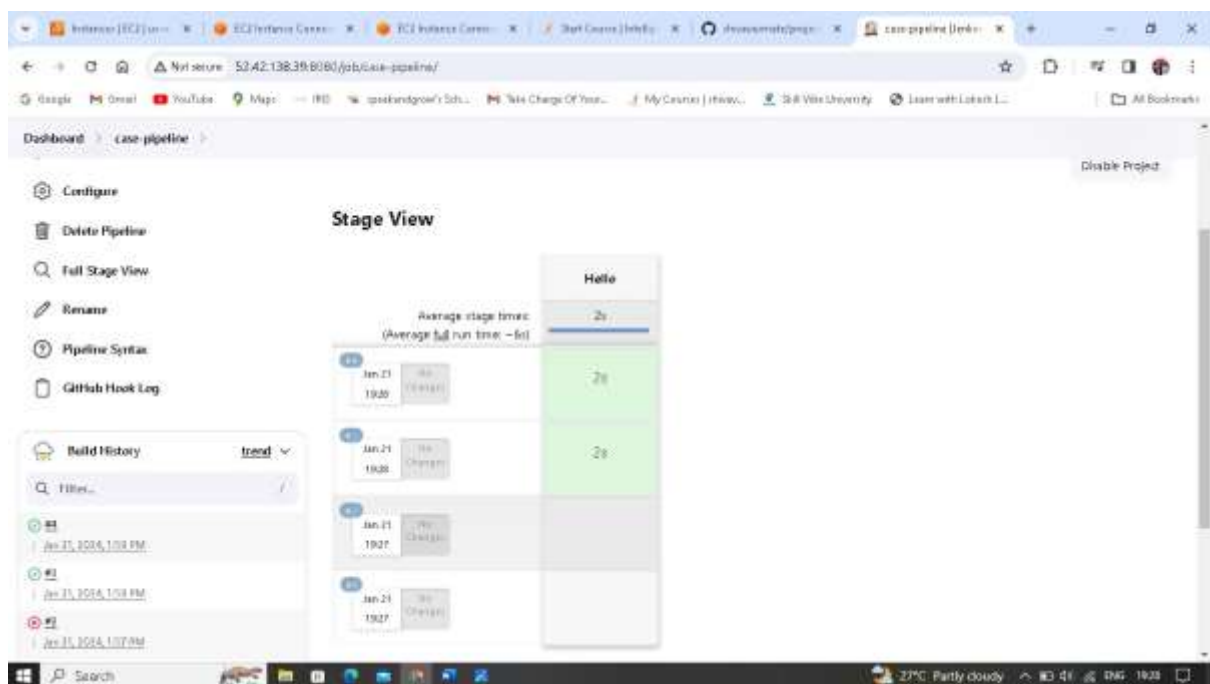
```
pipeline {  
    agent none  
  
    stages {  
        stage('Hello') {  
            agent {  
                label "Node-1"  
            }  
        }  
    }  
    steps {  
        echo 'Hello World'  
        git 'https://github.com/shivavasmate/project-html-website.git'  
    }  
}  
}
```

and click on save.

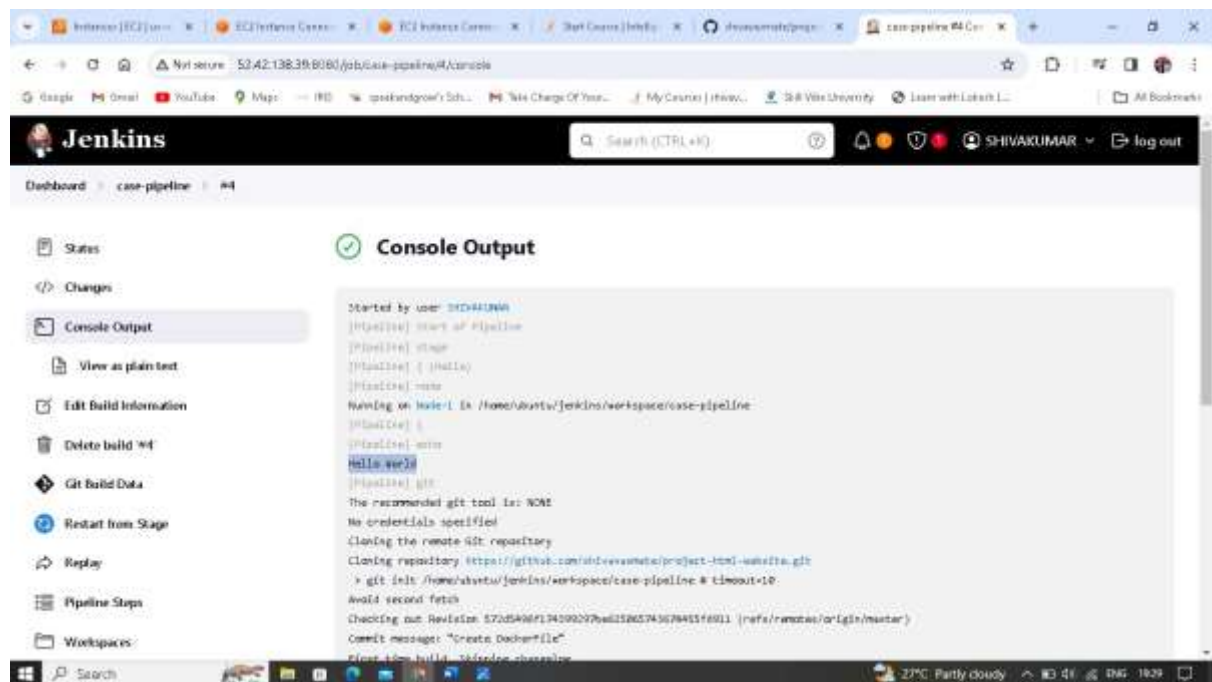
8. The case-pipeline is created and click on Build Now.



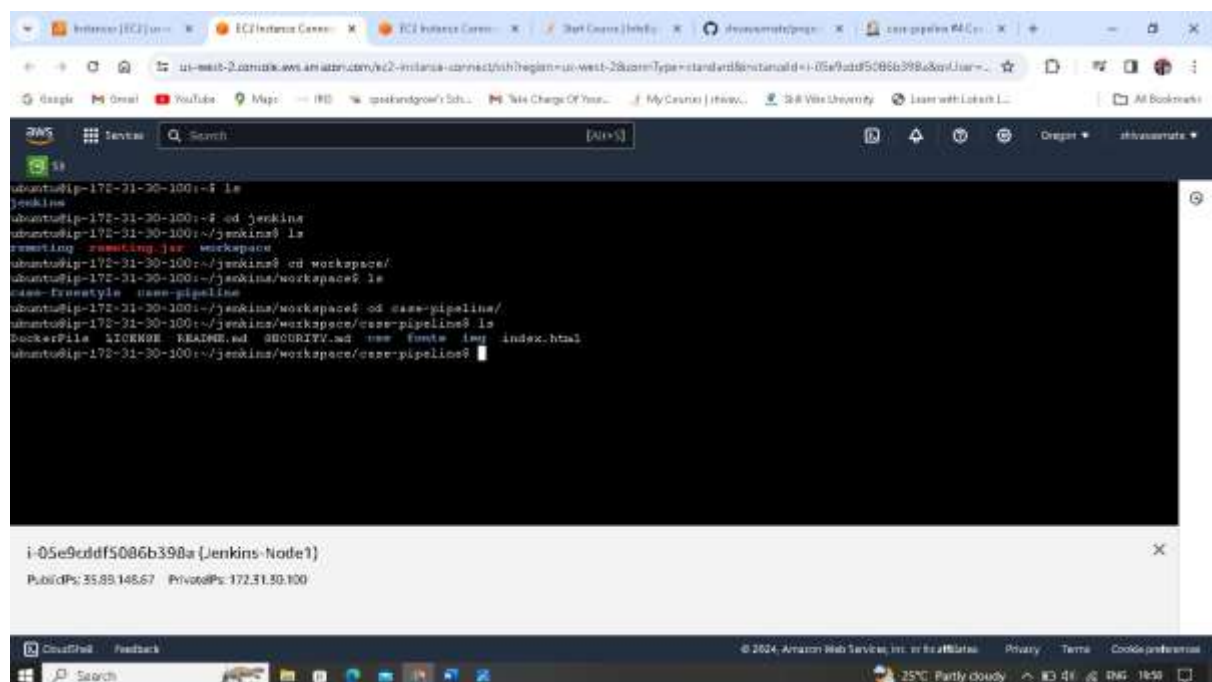
9. Build is successful.



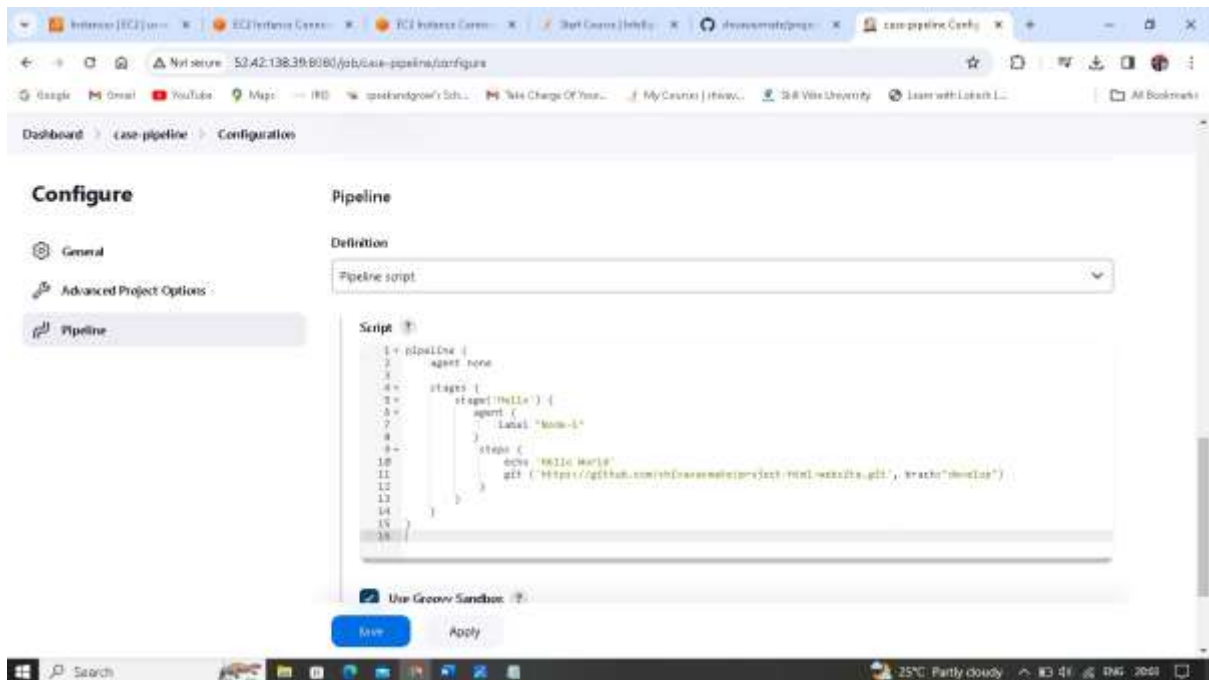
10. Check for it.



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'

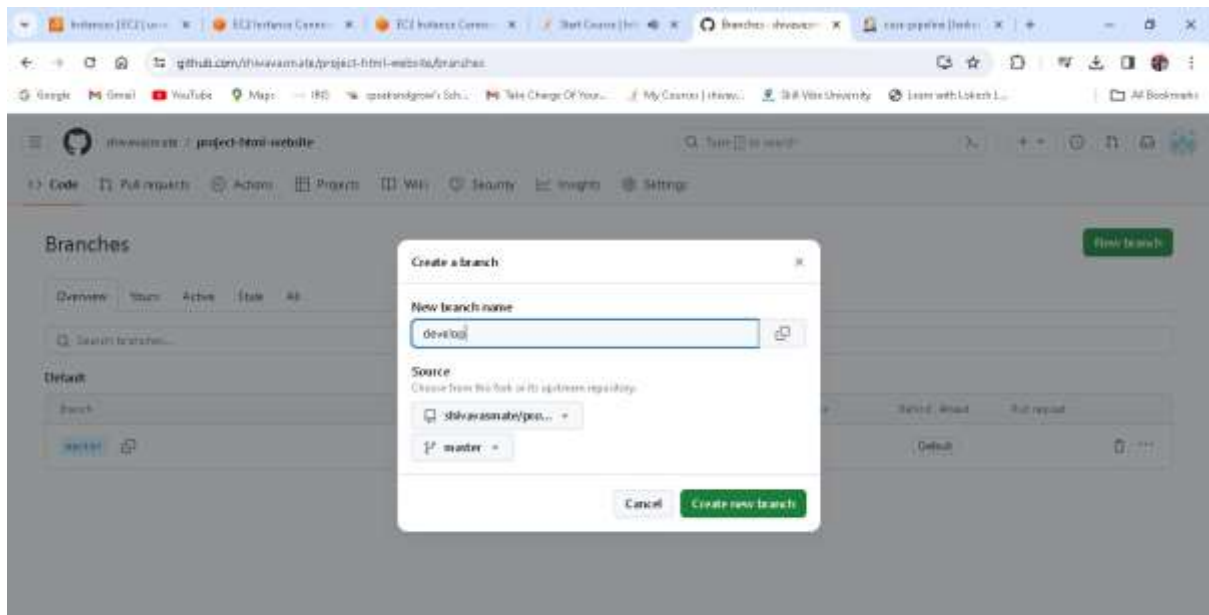
                git (url: 'https://github.com/shivavasmate/project-html-website.git' branch:"develop")

            }
        }
    }
}
```

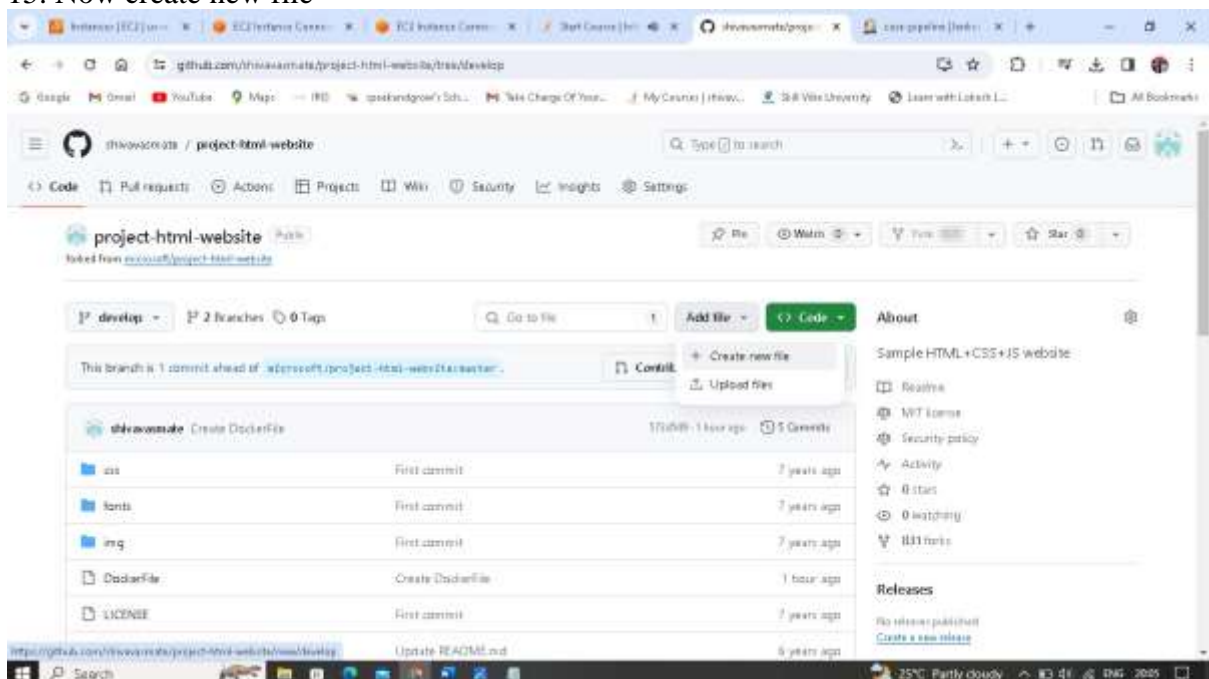


}

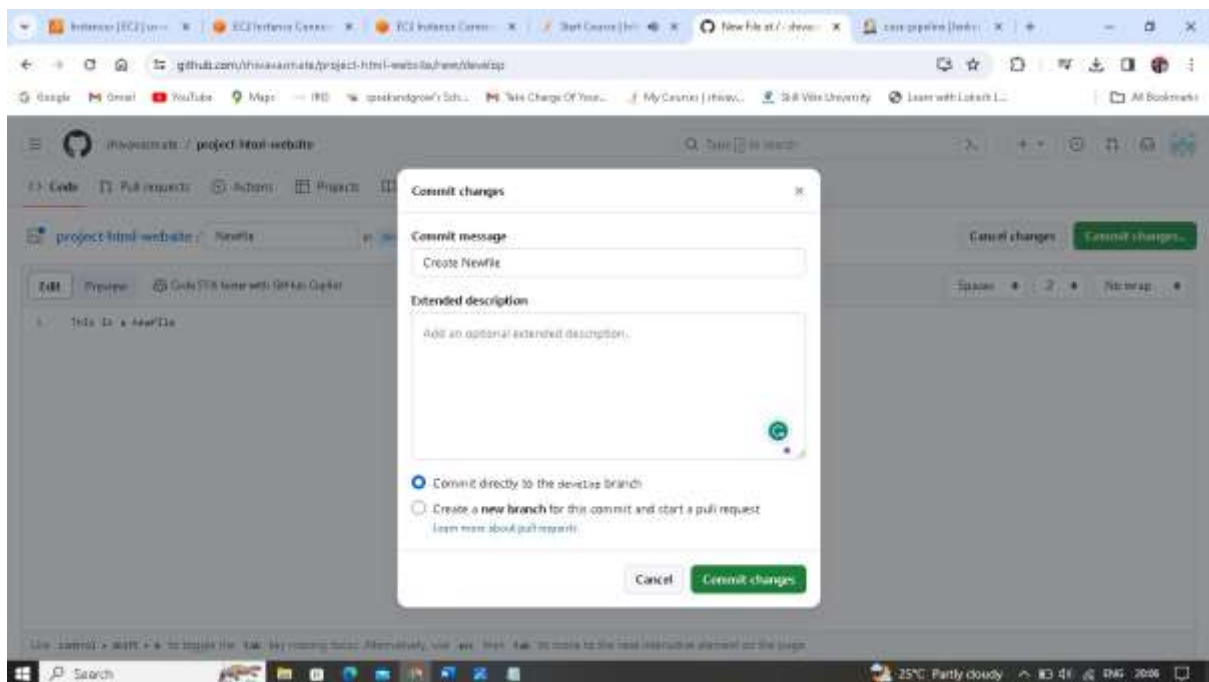
14. Create develop branch in the repo.



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'

    }

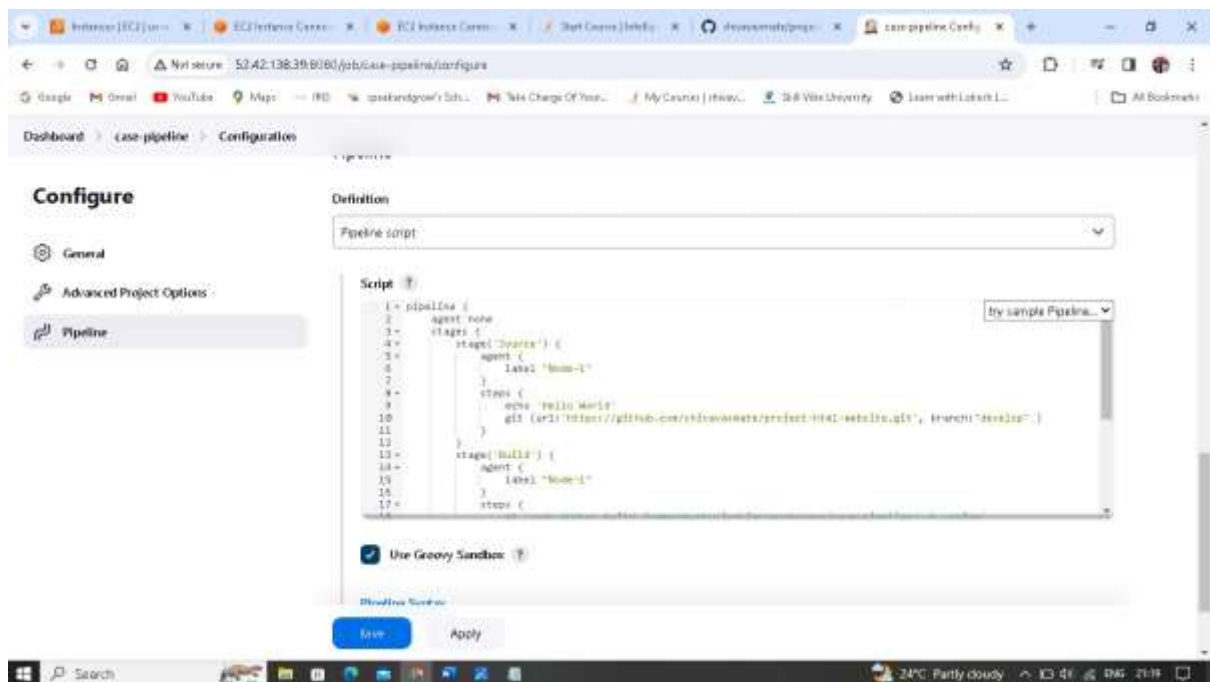
}

}

}

```

19. Click on save



## 20. Click Build Now

The screenshot shows the Jenkins dashboard for a project named 'case-pipeline'. The 'Status' tab is selected, showing a green checkmark and the text 'case-pipeline'. The 'Build View' section displays a single build with a duration of 2s. The 'Build History' section shows a list of builds, with the most recent one being a successful build.

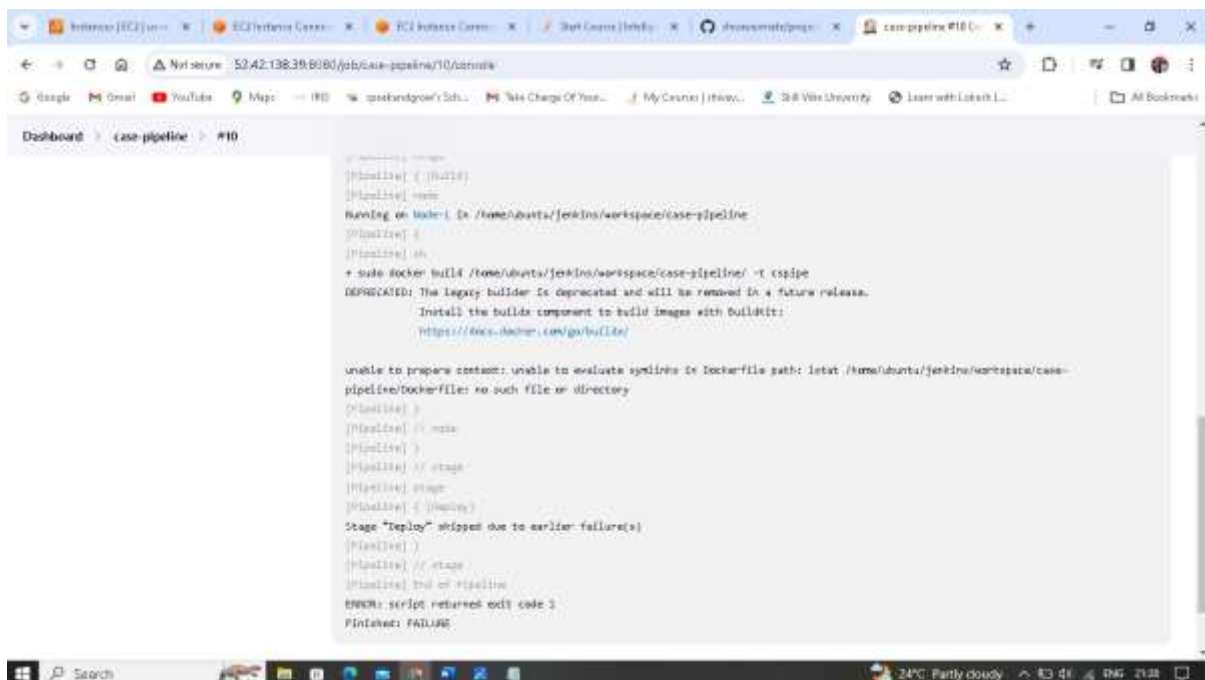
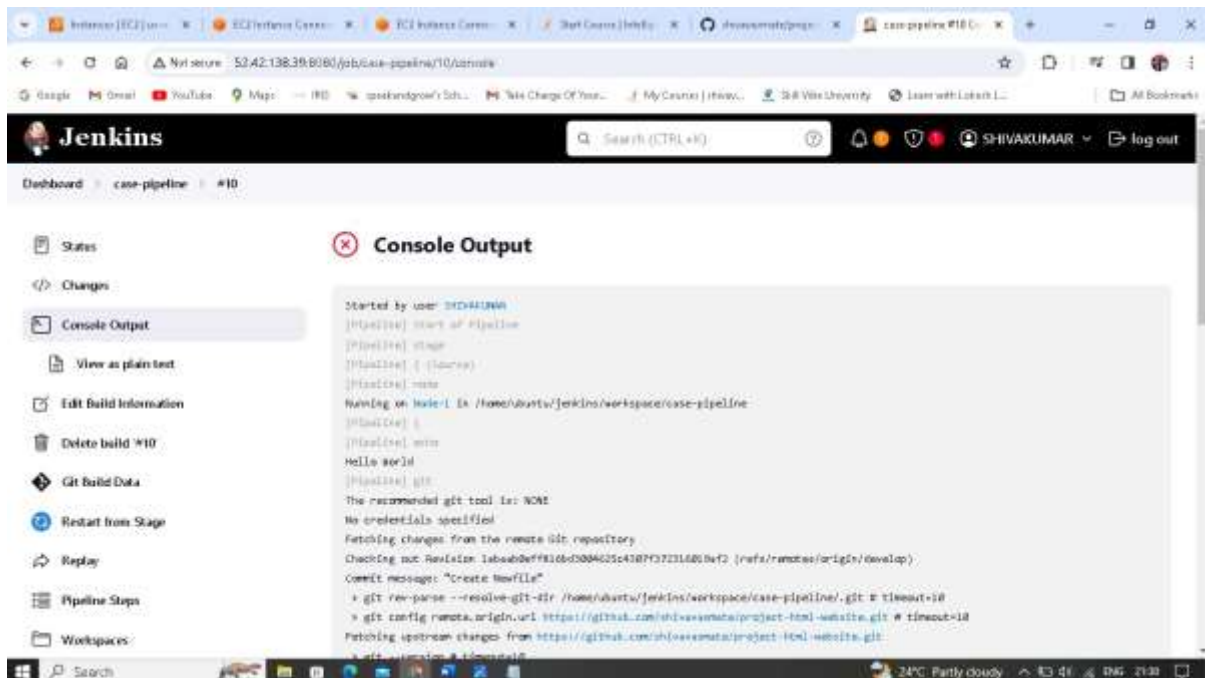
Build	Duration	Status
1	2s	Success

## 21. Error in Build and Deploy stages

The screenshot shows the Jenkins dashboard for a project named 'case-pipeline'. The 'Status' tab is selected, showing a green checkmark and the text 'case-pipeline'. The 'Build View' section displays a table of builds with columns for 'Source', 'Build', and 'Deploy'. The 'Build' and 'Deploy' columns show 'Failed' status for the most recent builds. The 'Build History' section shows a list of builds, with the most recent one being a failed build.

Build	Source	Build	Deploy
1	1s	1s	112ms
2	1s	1s	79ms
3	905ms	1s	186ms
4	2s	1s	63ms

## 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

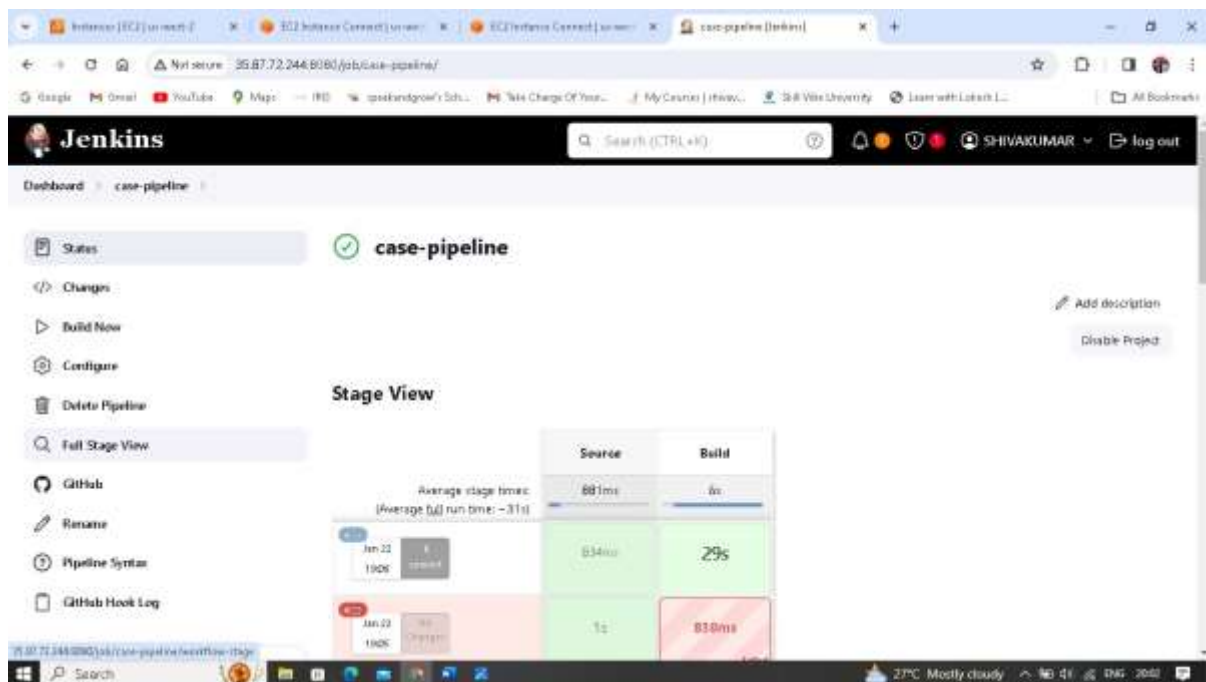
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

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.

The screenshot shows the Jenkins web interface for a pipeline named 'case-pipeline'. The left sidebar contains navigation links: Dashboard, Case-pipeline, Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, GitHub, Rename, Pipeline Syntax, and GitHub Hook Log. The main area displays the 'Stage View' for the 'case-pipeline' project. It shows a table of stages: Source, Build, and Deploy. The Source stage is highlighted with a 'Success' status and a 'Log' button. The Build stage shows a duration of 76.0ms, and the Deploy stage shows a duration of 1s. The table also includes columns for 'Average stage time' and 'Average full run time' (12s). The bottom of the screen shows the Windows taskbar with the date and time as 27°C Mostly cloudy, 10:41, 24/01/2024.

Stage	Source	Build	Deploy
Source	Success	76.0ms	1s
Build	76.0ms	772ms	
Deploy	634ms	29s	

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

The screenshot shows a web browser window with the Google homepage. The address bar displays the public IP address 35.93.156.116:84. The search bar is visible, and the Google logo is prominently displayed. The background of the page is a colorful, abstract image. The bottom of the screen shows the Windows taskbar with the date and time as 27°C Mostly cloudy, 10:41, 24/01/2024.

27. Now HTML Application is running.

