Jenkins - 1

You have been asked to:
- Trigger a pipeline using Git when push on Develop branch
- Pipeline should pull git content to a folder

Launch an EC2 instance and update it using: sudo apt-get update
For proper operation we need to install 'Jenkins' and 'Java'.

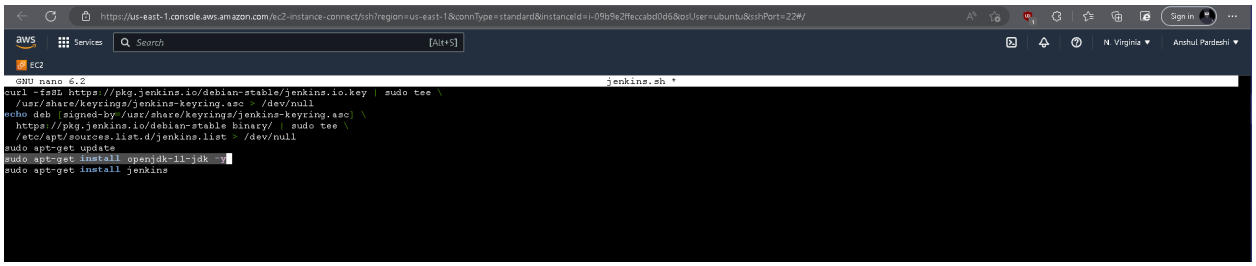
Let us create a bash file to install jenkins:

sudo nano jenkins.sh

Paste this:(commands to install jenkins are from Jenkin's official website)

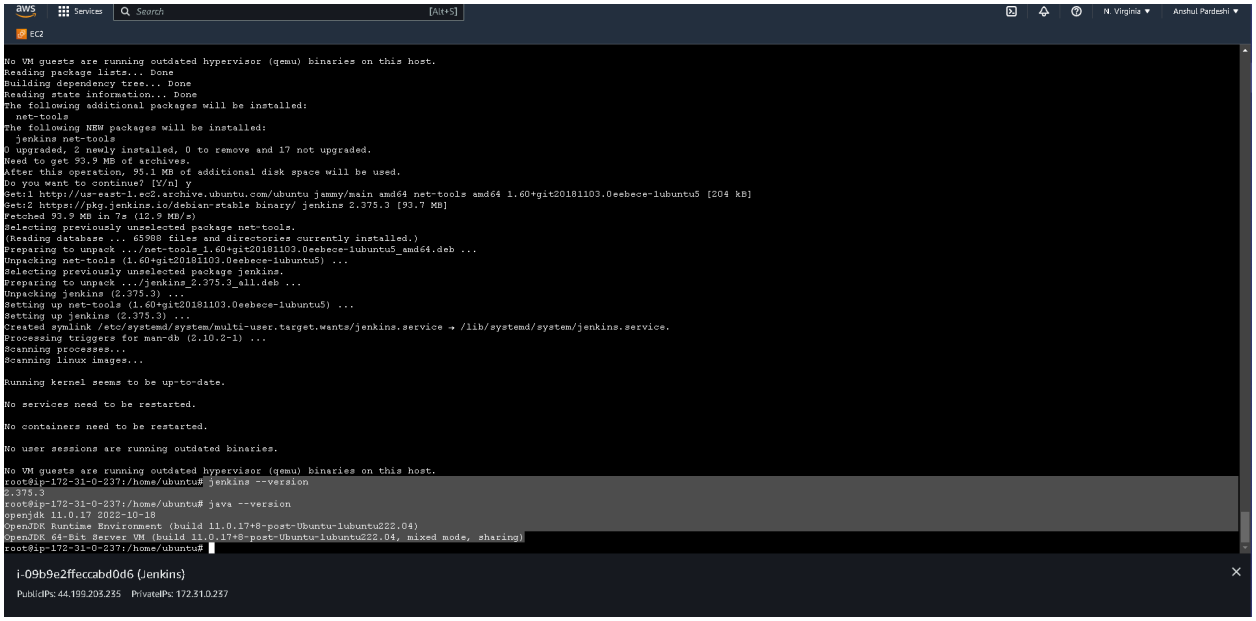
```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null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

We will add this to install java: sudo apt-get install openjdk-11-jdk -y

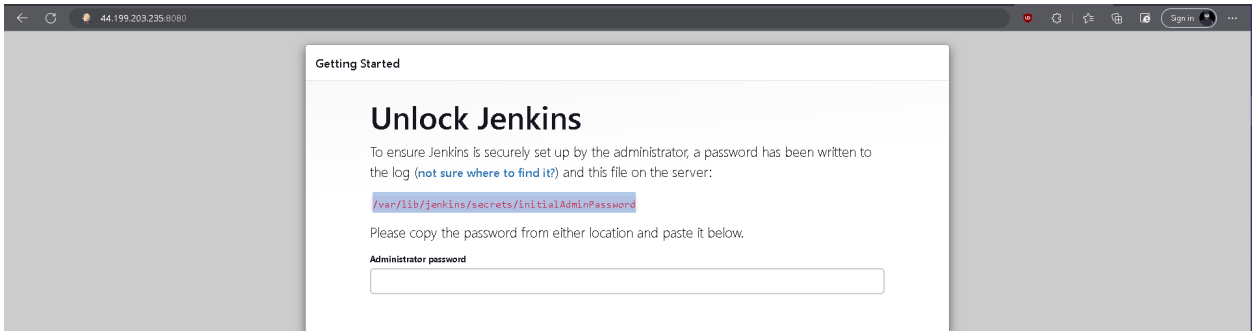


Let us run the bash file by:sudo bash jenkins.sh
All commands will be installed one by one. This procedure helps to install when multiple commands are to be used.

Commands are executed and Jenkins and Java are now successfully installed.



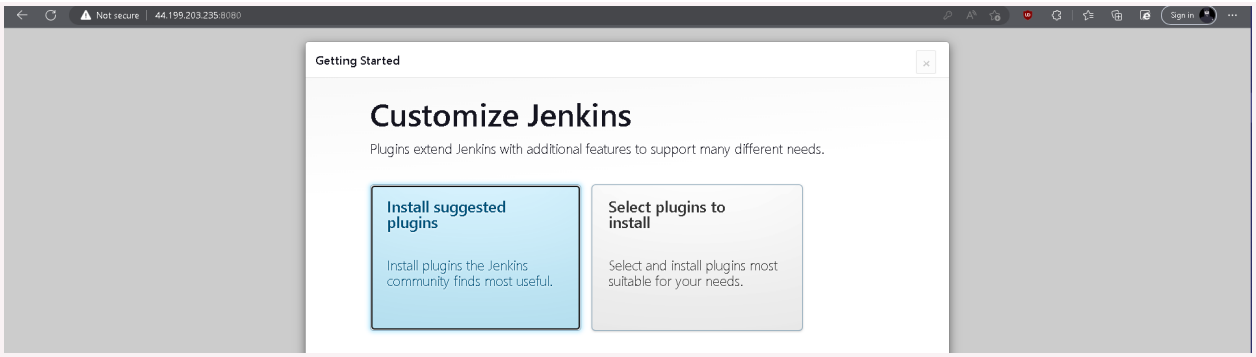
Copy paste the public_ip:8080 on your browser.
Make sure 8080 is allowed everywhere in your EC2 security group.

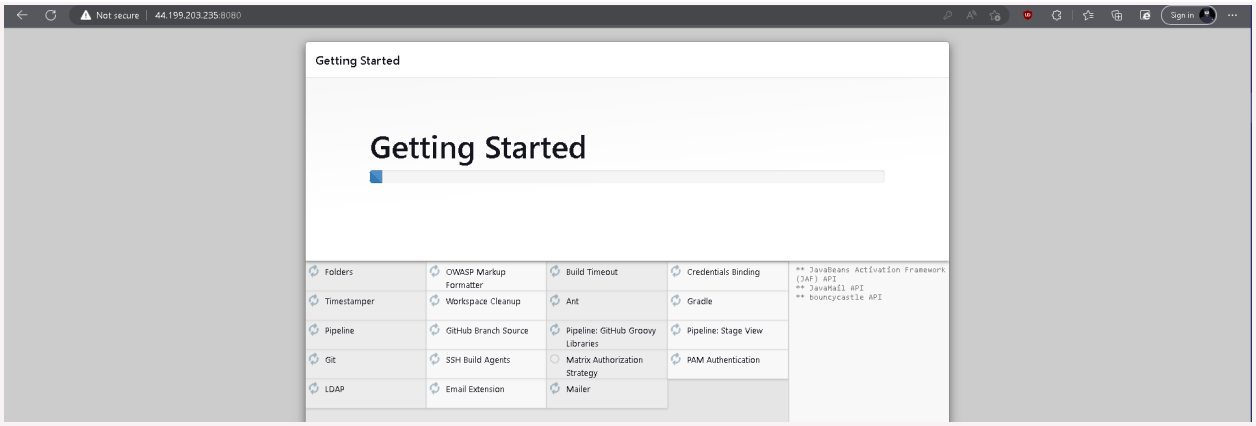


To get the password:
cat `/var/lib/jenkins/secrets/initialAdminPassword`

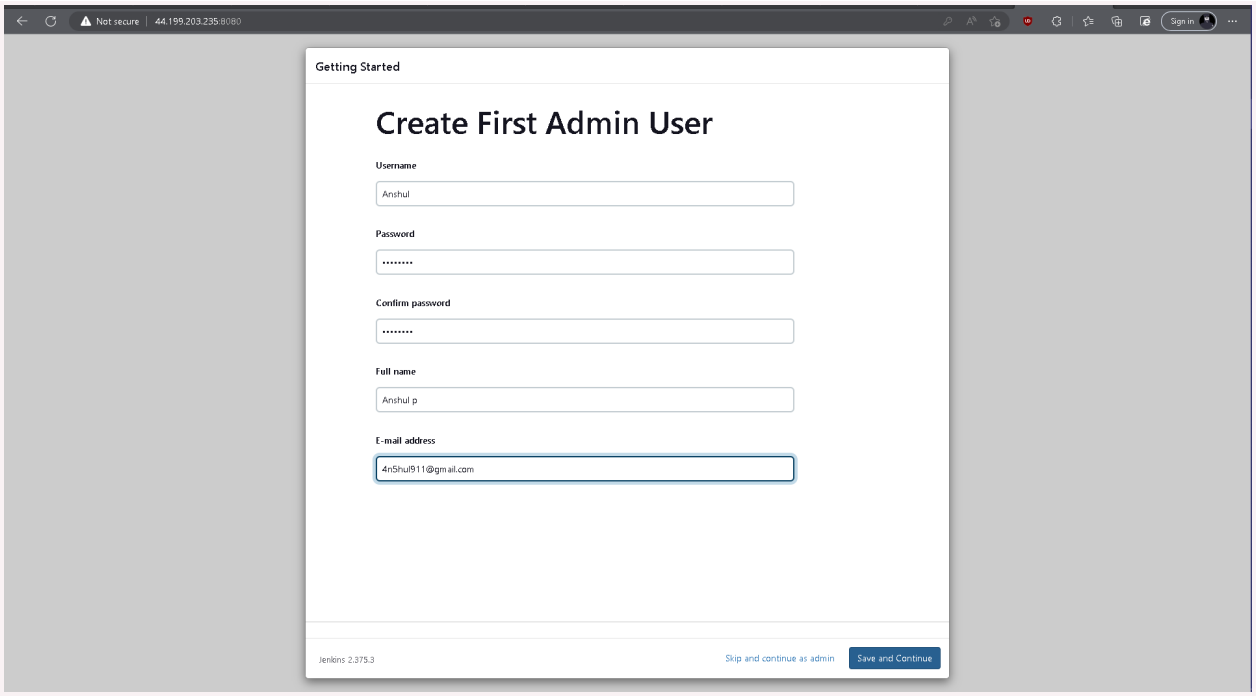
Copy paste this password on jenkin's page.

Then install suggested plugins.

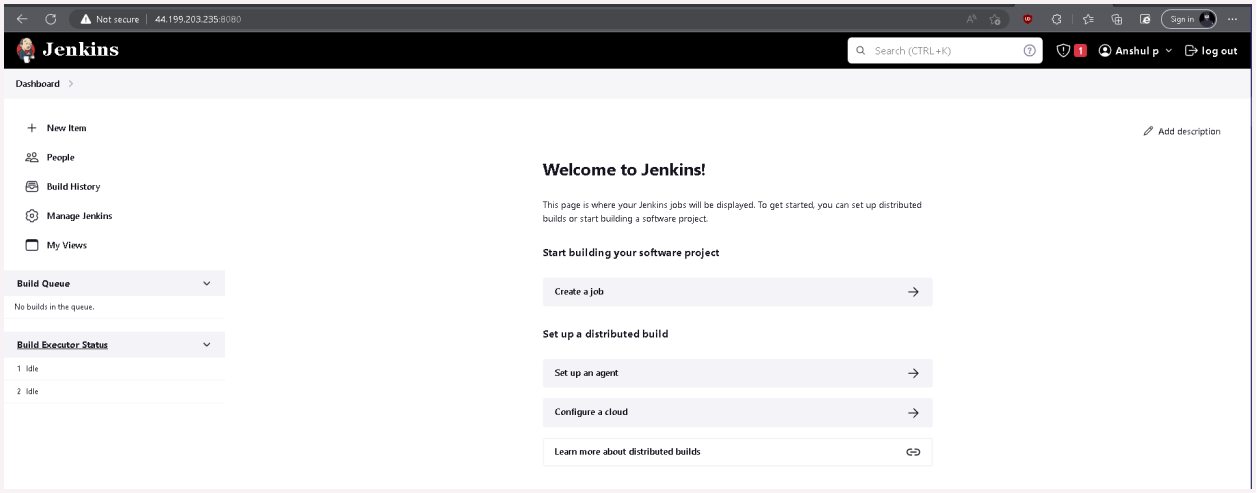




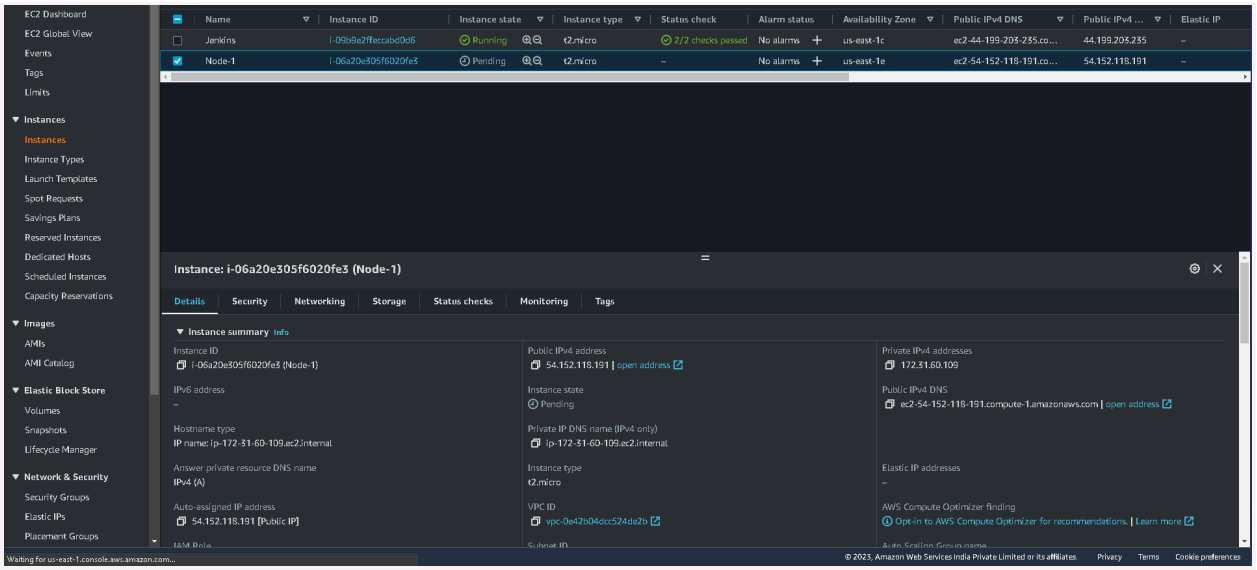
Fill in the details appropriately.



Jenkins has been installed and ready to create a job.

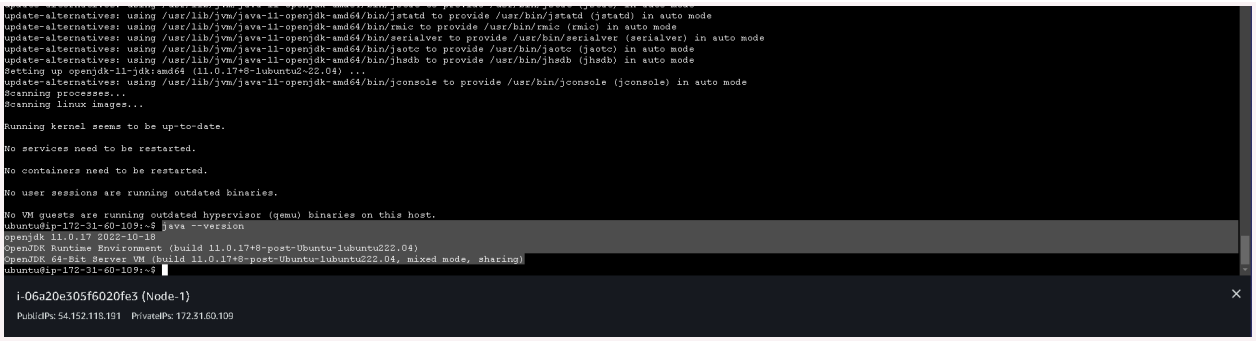


Now, let us add node/slave. We'll create a new instance for that.

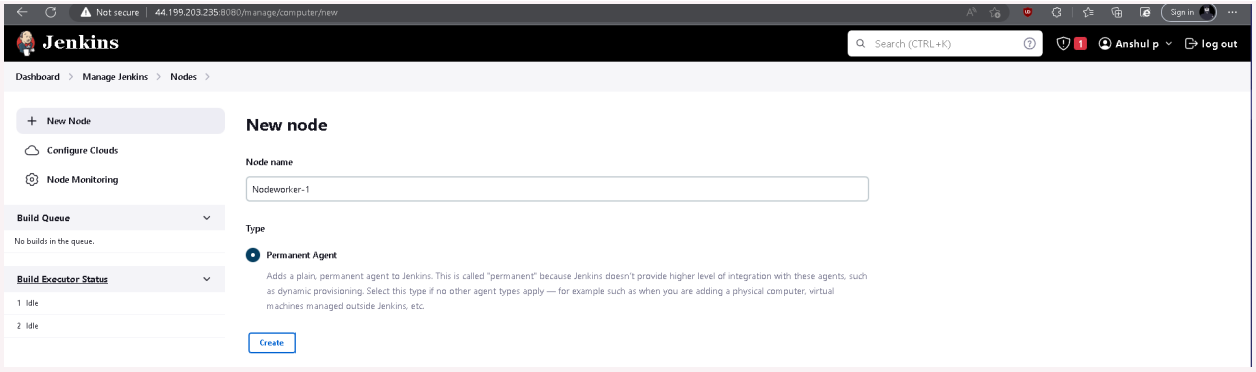


sudo apt-get install openjdk-11-jdk -y

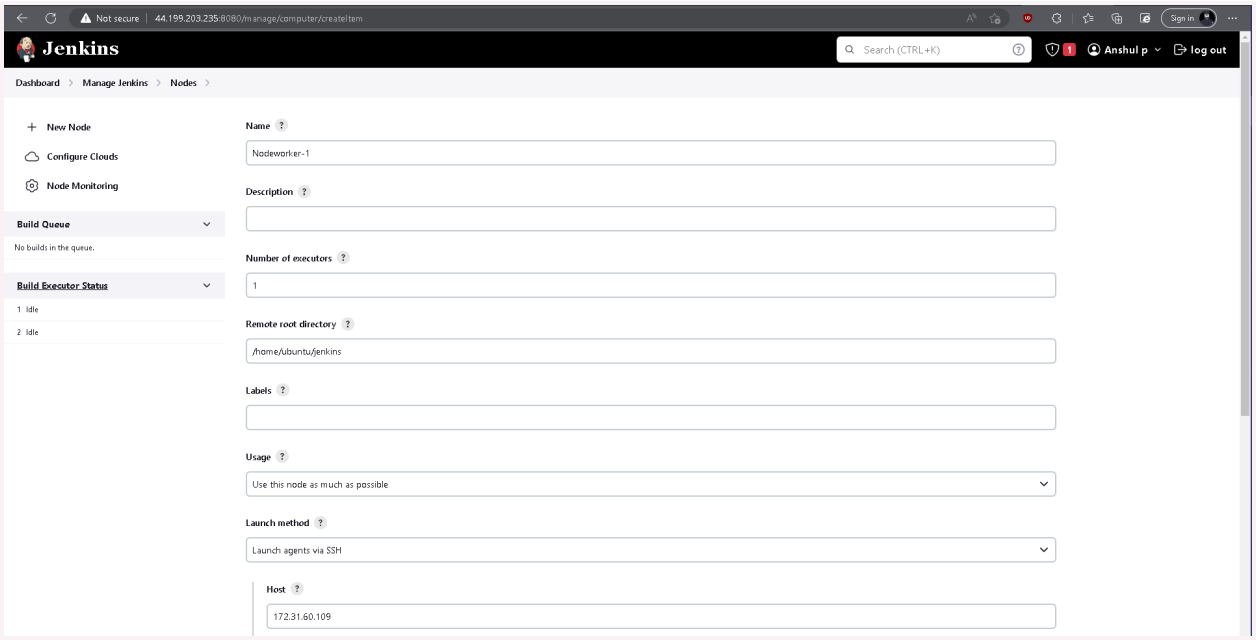
Java is installed



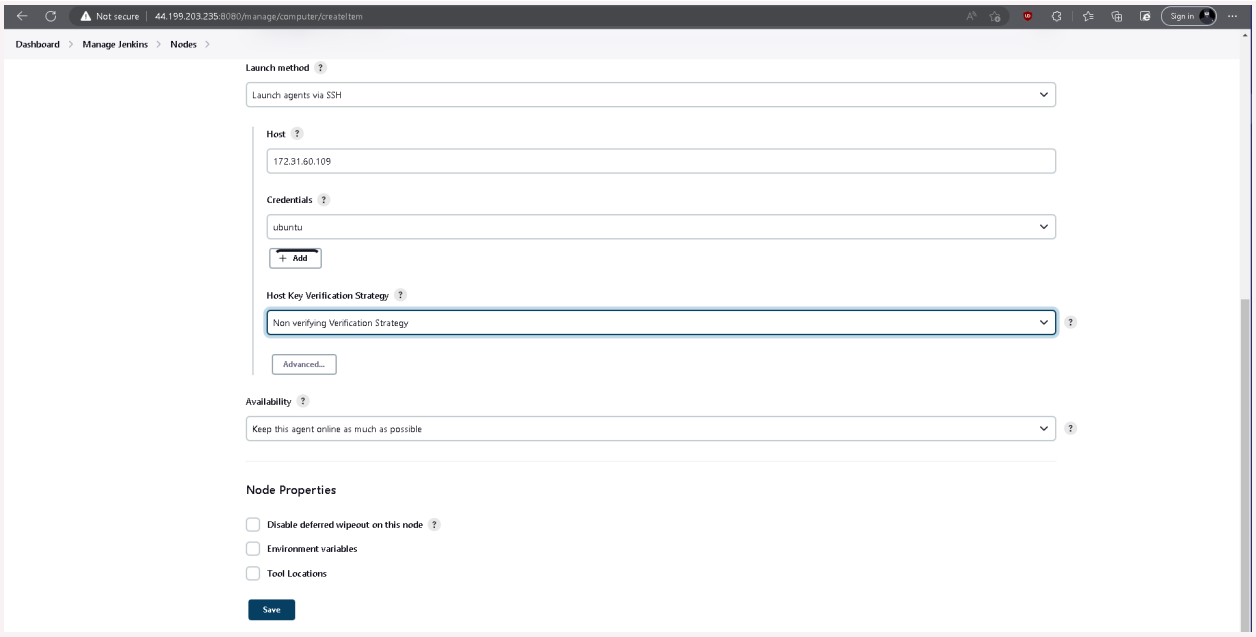
Next, we will add this node to jenkins from the jenkins site.
For this in jenkins website goto: manage jenkins>>manage nodes>>add nodes



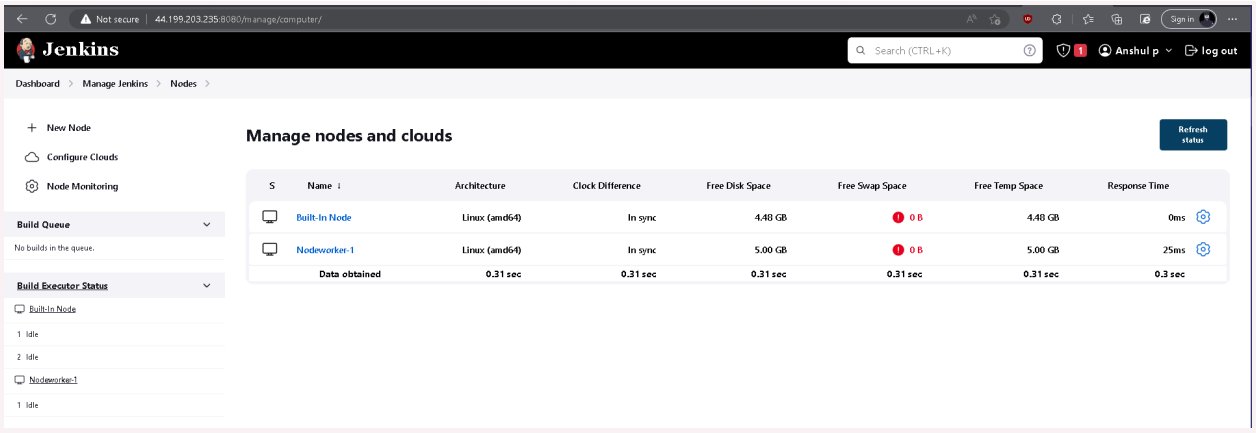
Give your node a name and move forward.
As our OS is ubuntu, remote root directory will be: /home/ubuntu/jenkins/
In launch method choose: Launch via SSH
In host 1 we need to give public/private ip of worker1.
Why private ip? Because if instance is restarted the public ip will change then you need to change the ip every time the instance restarts. Private ip will be the same.



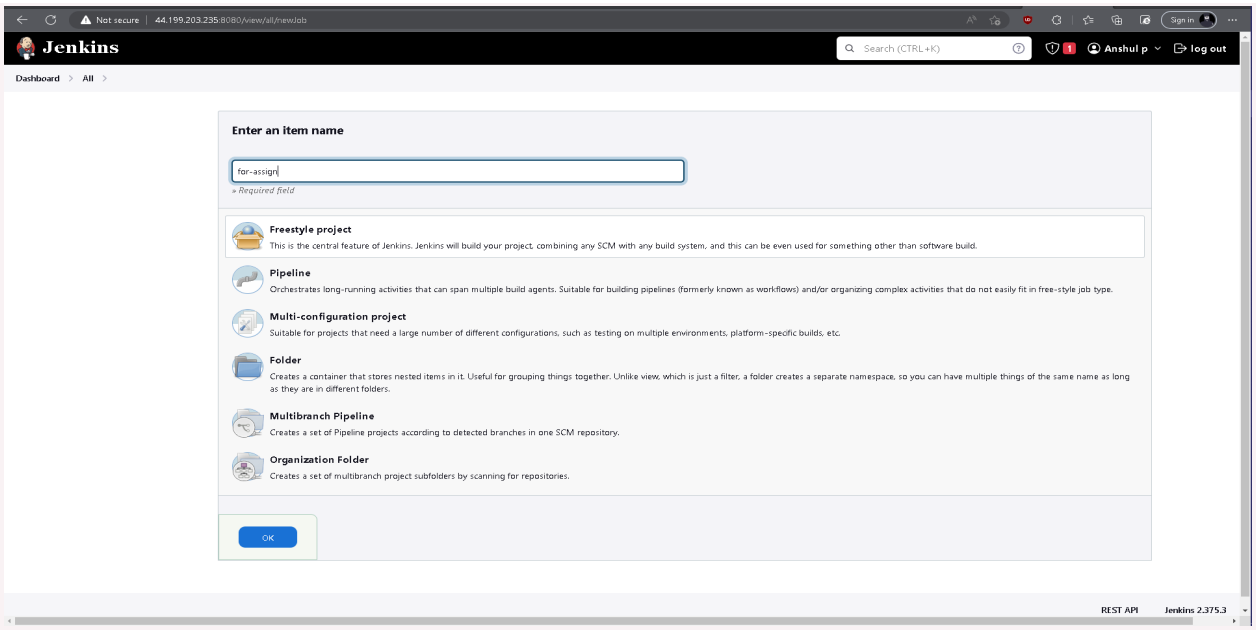
Now we need to add credentials.
We will be going with ‘ssh username with private key’.
In username: ubuntu
And in private key enter it directly. Copy paste the contents of your pem file keypair of instance.
Add the user and save.



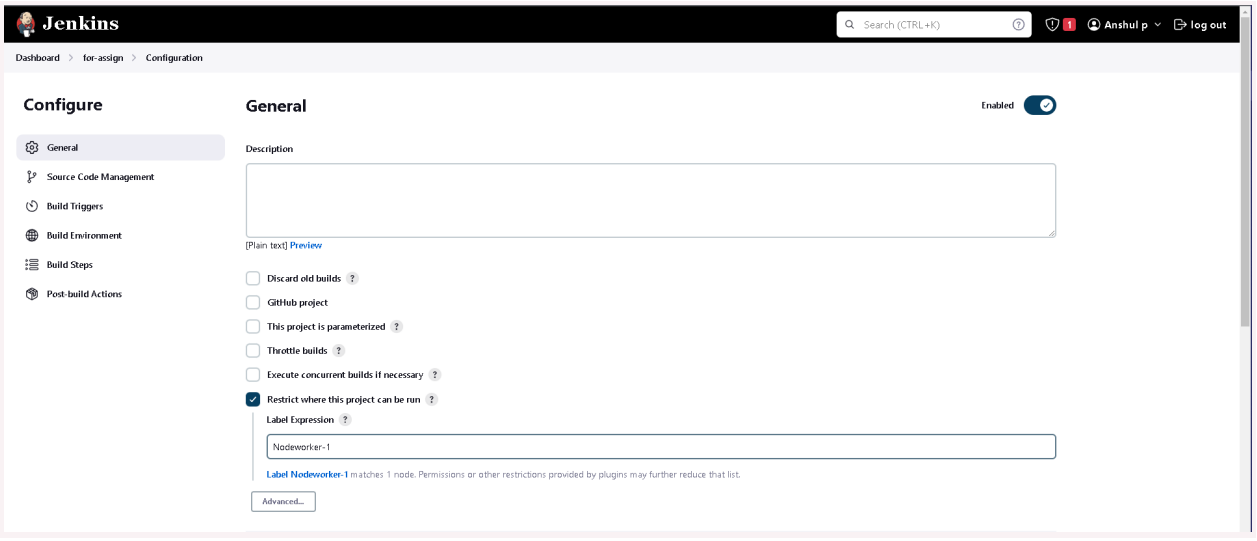
Refresh the status and you can see node is connected.



Now we need to trigger a pipeline.
Goto: dashboard>>new job>>freestyle, name it and save.

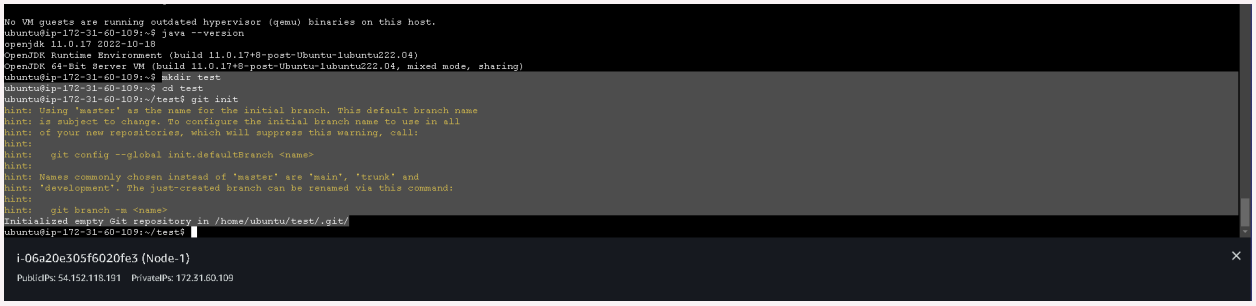


We will restrict this project to run on the worker node that we created.



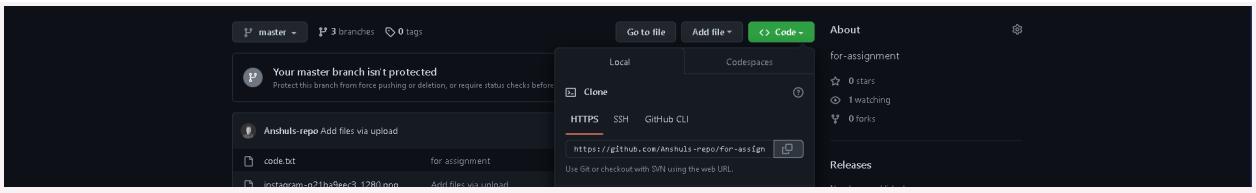
For ‘source code management we need to give github link.
Let us initialise git in worker node to get the link.

mkdir <directory name>
Cd <same directory name>
git init

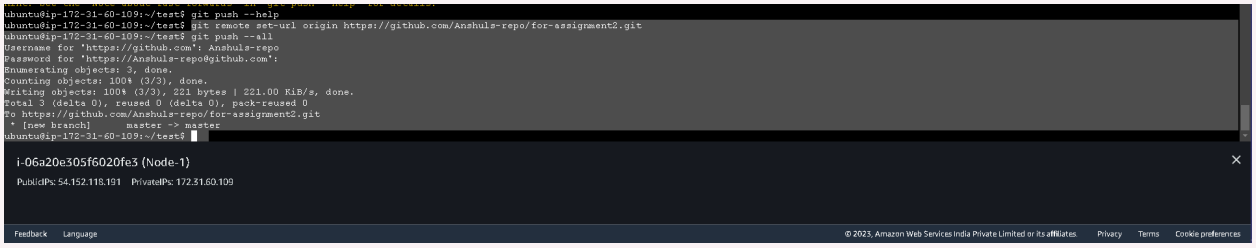


Create a file and commit it to create a master branch.
touch file1.txt
Git add .
Git commit -m "<any msg>"

Now push this to a repo.
For this add a origin first.
Copy https link of you new repository in github account.Then use command:

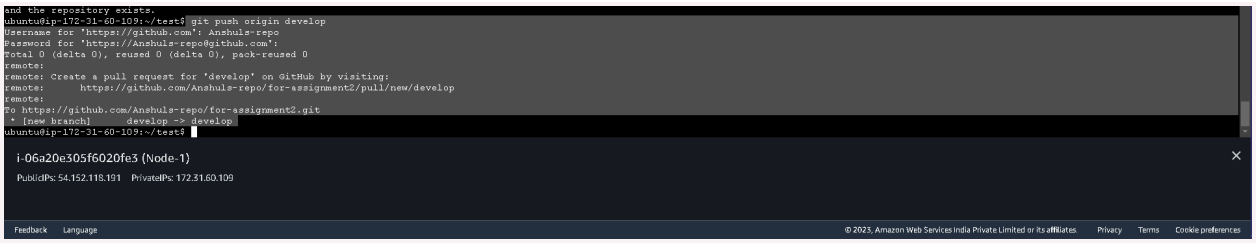


git remote add origin <https github link>
Git push --all
Enter id and your github token as your password.

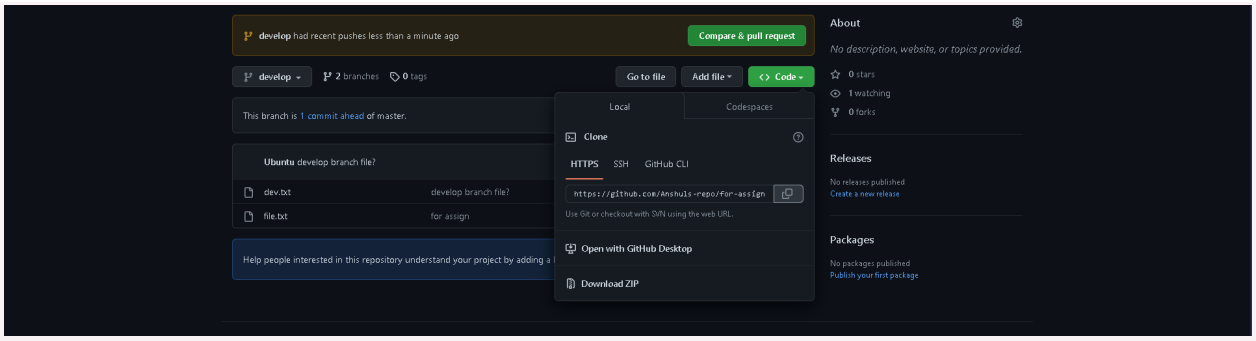


We need to trigger a pipeline when push on the Develop branch. We will create new branch: git branch develop
Then checkout to new branch: git checkout develop

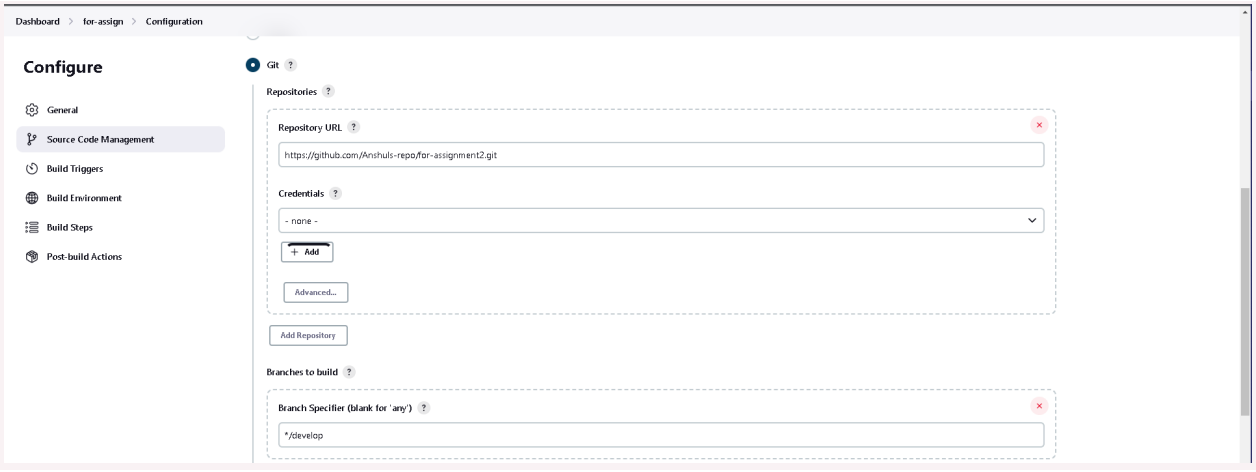
Now git push new branch too: git push origin develop, then provide username and token.



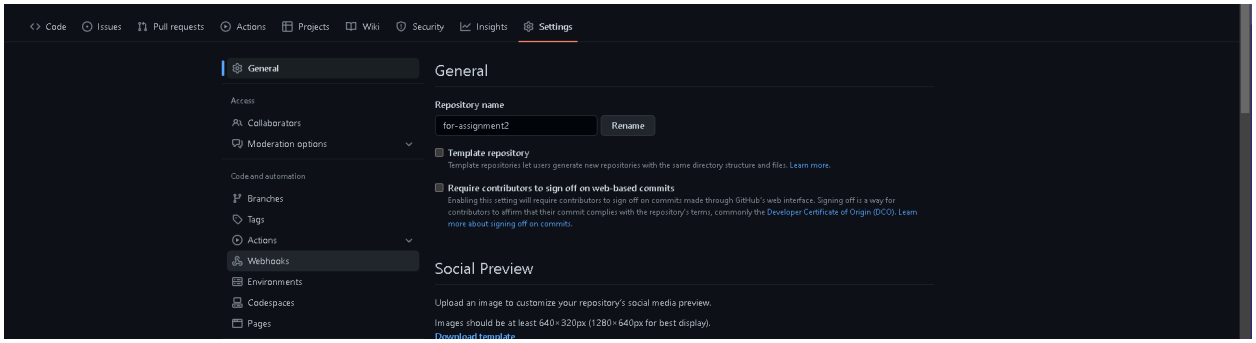
Create and push a file in develop branch: touch dev.txt
Git add.
Git push --all
Copy the https link.



Paste the link in git section in source code management. Also change the branch name to “develop”.

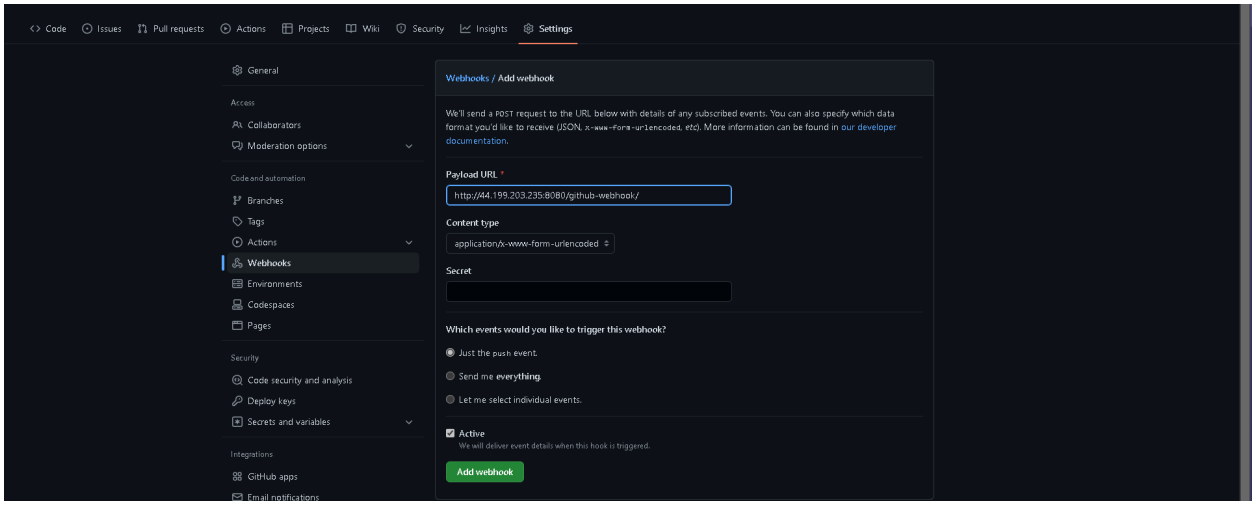


For automatic trigger, in ‘Build trigger’ choose ‘GitHub hook trigger for GITScm polling’.
Also we need to create a webhook. Goto your github then choose: settings>>webhook.

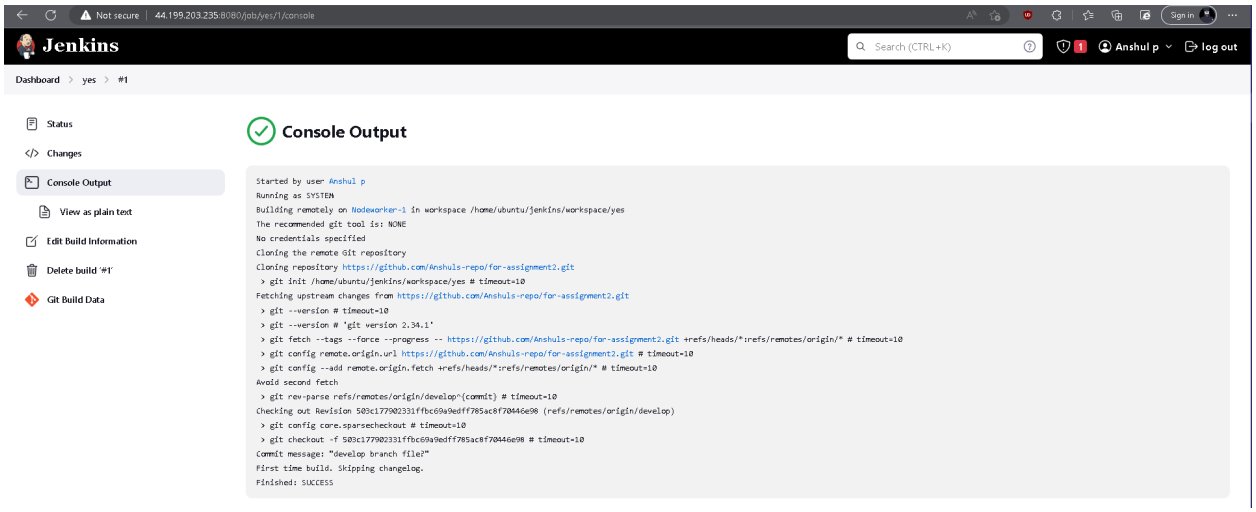


While creating webhook, copy paste your jenkins ip paste it. Write it in this format:

jenkins_ip/github-webhook/



Apply the settings on jenkins side.
Now click on build now to create a job
Before that observe that there is no workspace folder in jenkins directory on worker node.

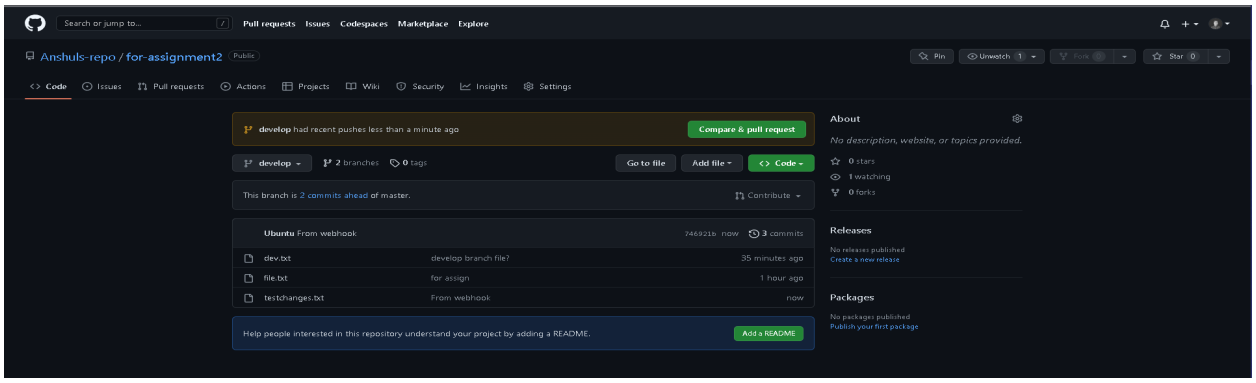


Now you can see the files on the develop branch in the worker node.

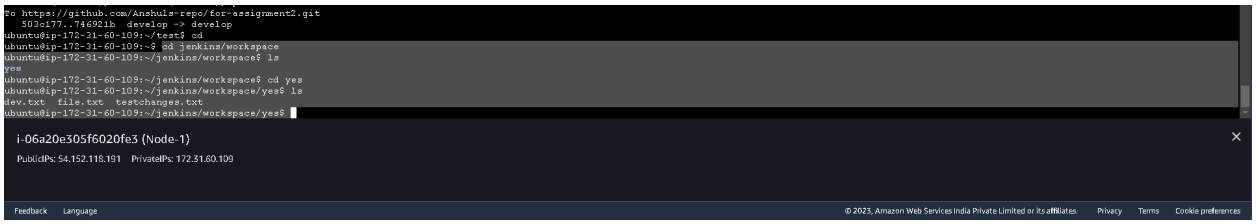
Now, if we make any changes in git repo in dev branch they will be reflected here.
Let us go to git folder and make some changes.
cd

cd test
touch testchanges.txt
Git add .
Git commit -m “<message>”
Git push –all
Give your credentials and token.

New file has been added to repo.



Let us check if changes has been reflected in jenkins workspace.
cd
cd jenkins/workspace/yes
ls



<u>Conclusion:</u> Whenever changes are made to develop branch in git repo, they will reflect to the ‘workspace’ directory in the worker node.