

# 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`



```
root@ip-172-31-0-237:/home/ubuntu# sudo nano jenkins.sh
```

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

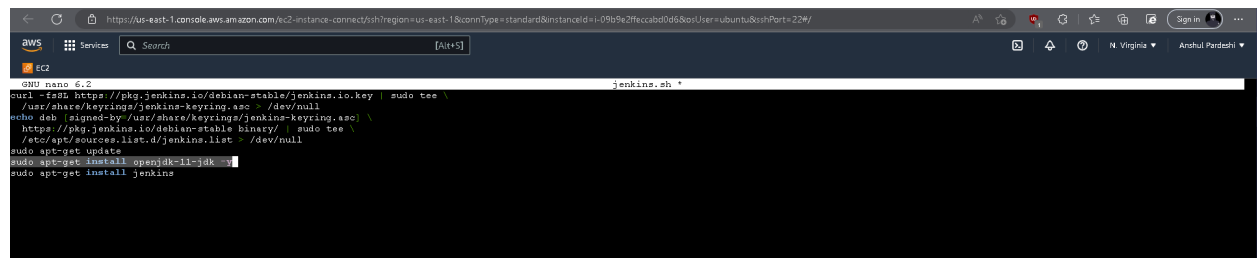
```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

```
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`



```
GNU nano 6.2 jenkins.sh
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 openjdk-11-jdk -y
sudo apt-get install jenkins
```

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.

```
aws Services Search [Alt+S]
EC2

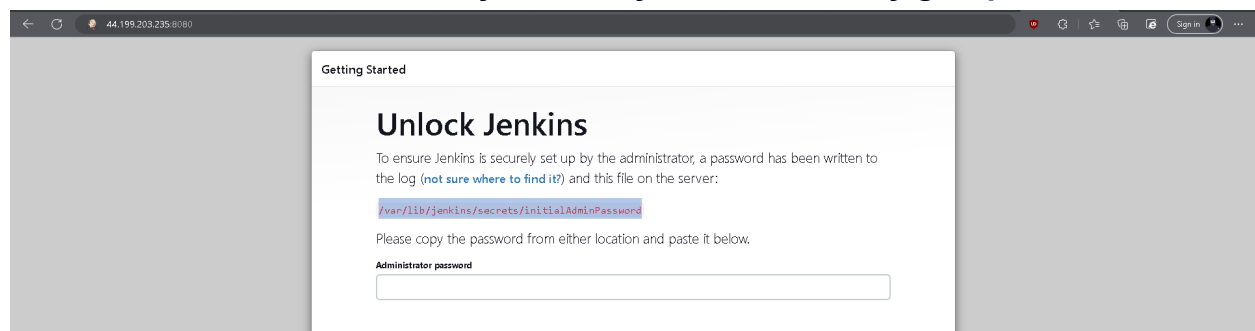
No VM guests are running outdated hypervisor (qemu) binaries on this host.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 17 not upgraded.
Need to get 90.9 MB of archives.
After this operation, 95.1 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1-ec2.archive.ubuntu.com/ubuntu/jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-lubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.375.3 [93.7 MB]
Fetched 93.9 MB in 7s (12.9 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 63908 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-lubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-lubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.375.3_all.deb ...
Unpacking jenkins (2.375.3) ...
Setting up net-tools (1.60+git20181103.0eebece-lubuntu5) ...
Setting up jenkins (2.375.3) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
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 (qemu) binaries on this host.
root@ip-172-31-0-237:/home/ubuntu# jenkins --version
2.375.3
root@ip-172-31-0-237:/home/ubuntu# java --version
openjdk 11.0.17 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-1ubuntu222.04, mixed mode, sharing)
root@ip-172-31-0-237:/home/ubuntu#

i-09b9e2ffecabd0d6 (Jenkins)
PubIdPc: 44.199.203.235 PrivatePc: 172.31.0.257
```

Copy paste the public\_ip:8080 on your browser.

Make sure 8080 is allowed everywhere in your EC2 security group.



To get the password:

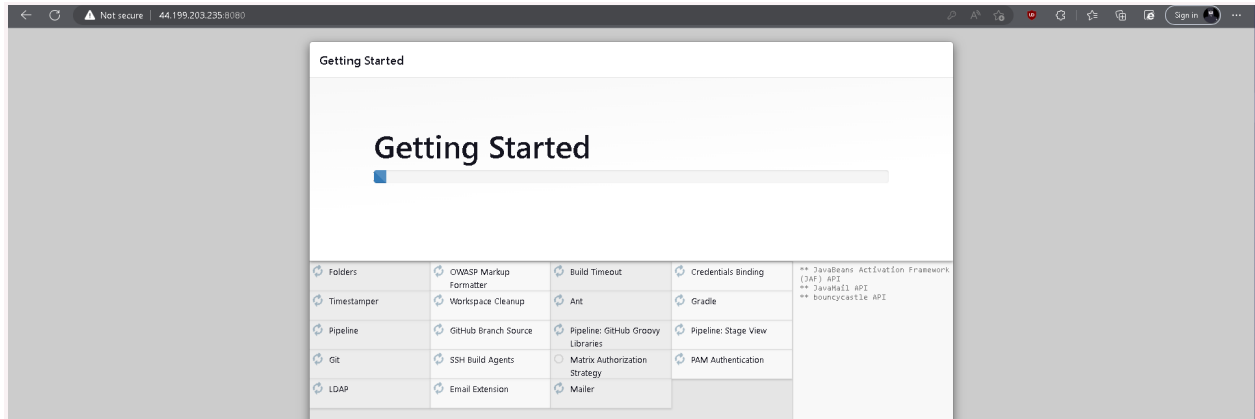
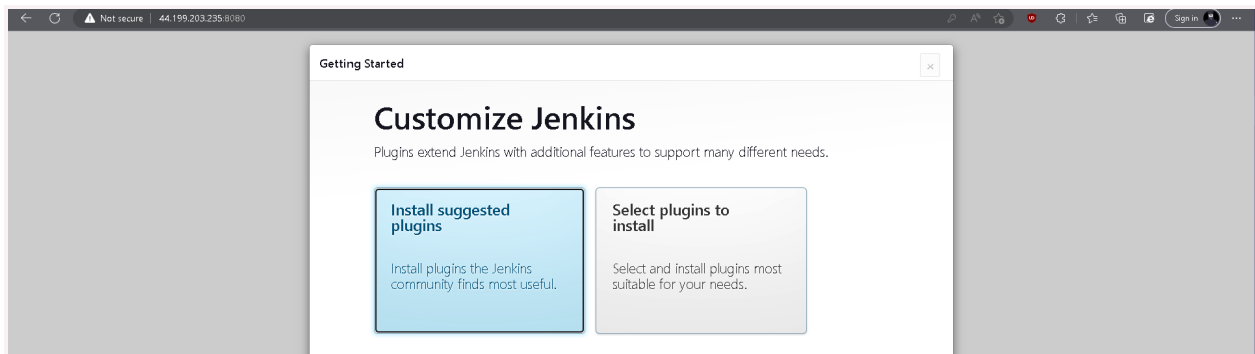
**cat /var/lib/jenkins/secrets/initialAdminPassword**

```
2.375.3
root@ip-172-31-0-237:/home/ubuntu# java --version
openjdk 11.0.17 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-1ubuntu222.04, mixed mode, sharing)
root@ip-172-31-0-237:/home/ubuntu# cat /var/lib/jenkins/secrets/initialAdminPassword
40h02e469211a498082290e4406215ea
root@ip-172-31-0-237:/home/ubuntu#

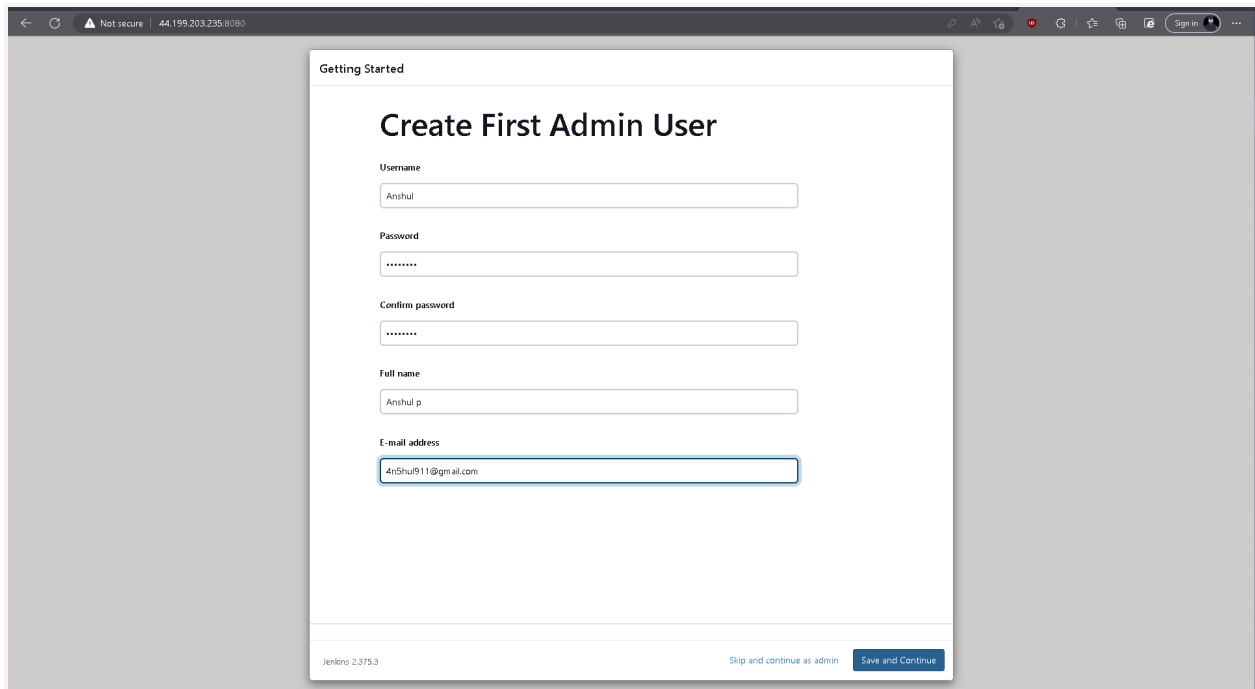
i-09b9e2ffecabd0d6 (Jenkins)
PubIdPc: 44.199.203.235 PrivatePc: 172.31.0.257
```

Copy paste this password on jenkins's page.

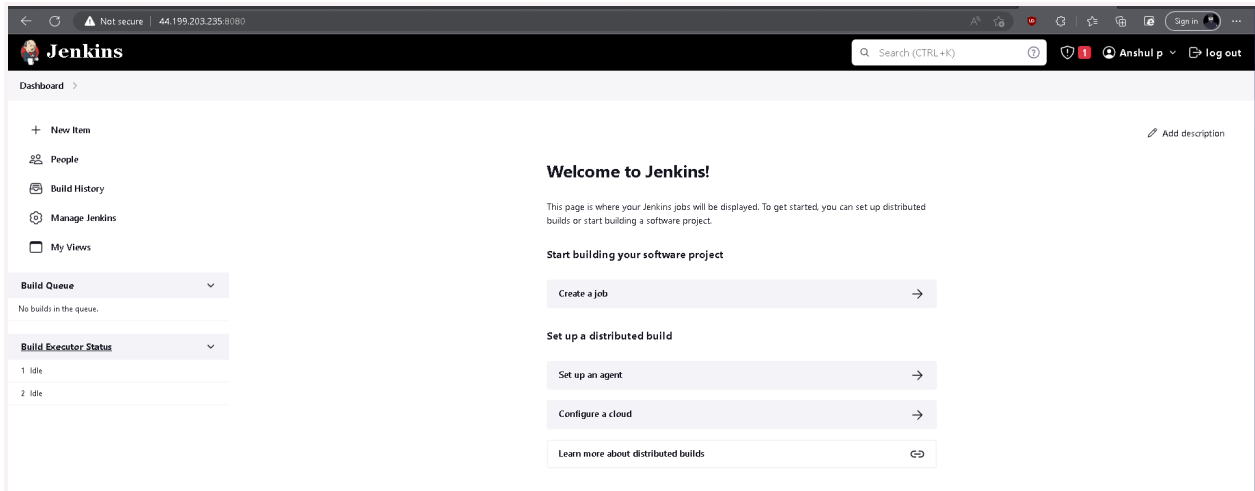
Then install suggested plugins.



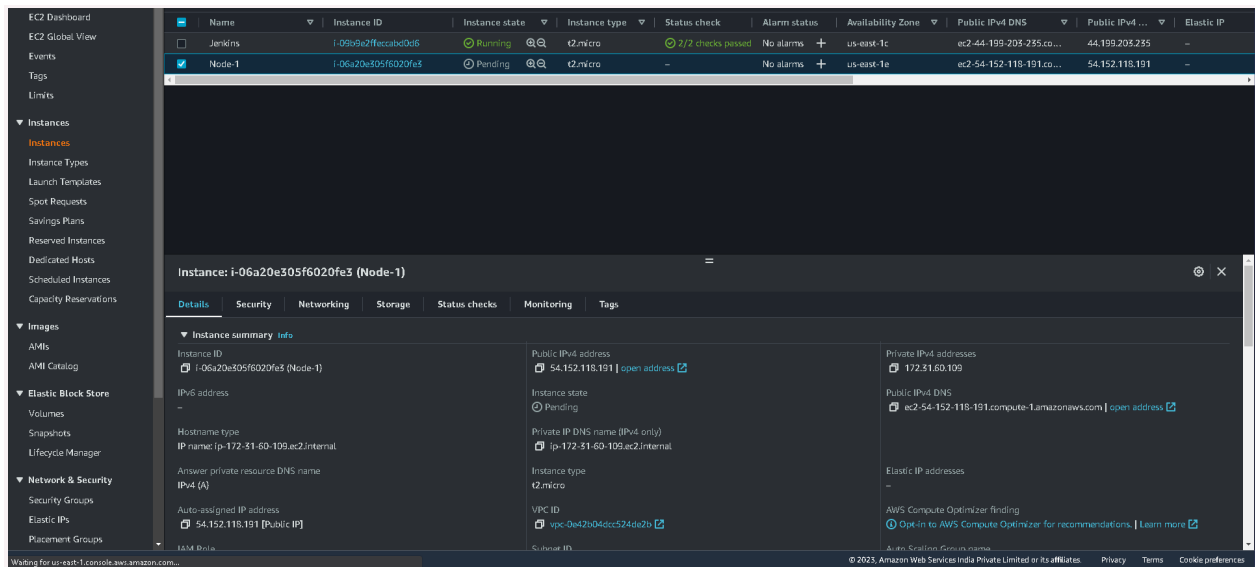
Fill in the details appropriately.



## Jenkins has been installed and ready to create a job.



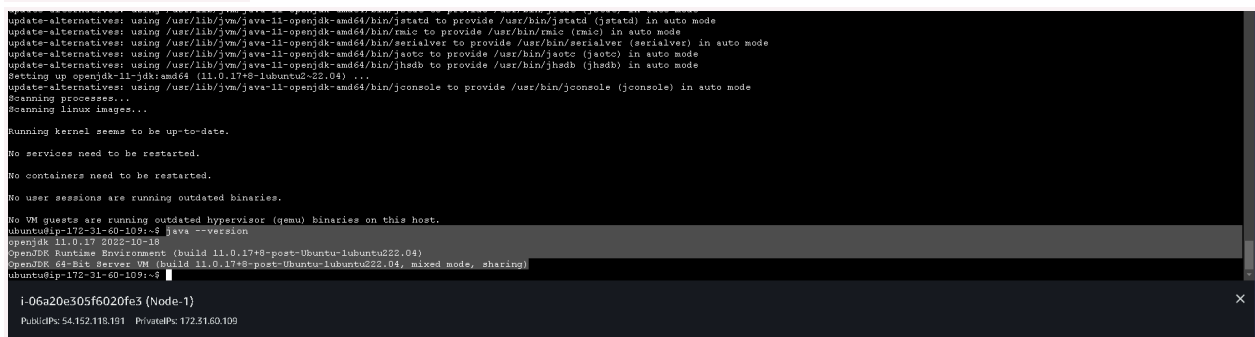
## Now, let us add node/slave. We'll create a new instance for that.



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

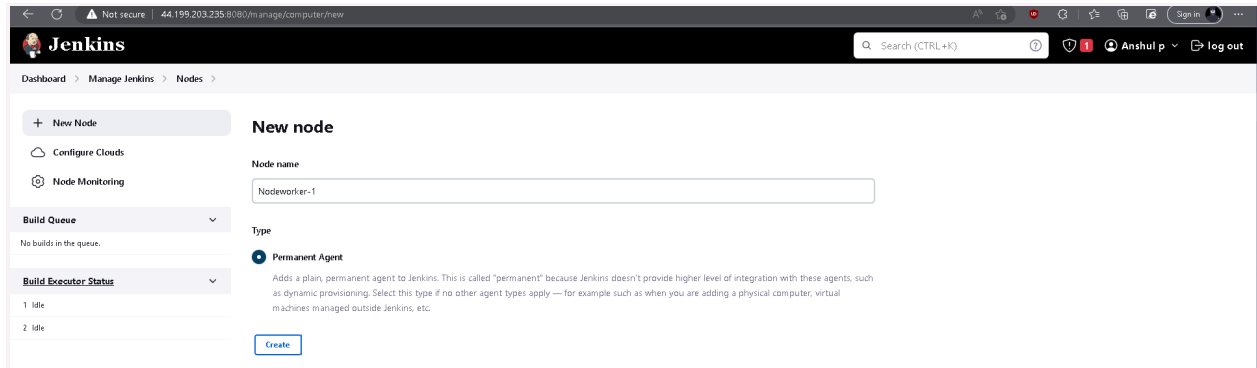


## Java is installed



Next, we will add this node to jenkins from the jenkins site.

For this in jenkins website goto: manage jenkins>>manage nodes>>add nodes



The screenshot shows the Jenkins 'New node' page. On the left sidebar, there are links for 'New Node', 'Configure Clouds', 'Node Monitoring', 'Build Queue', and 'Build Executor Status'. The 'Build Queue' shows 'No builds in the queue.' and the 'Build Executor Status' shows two idle executors. The main form is titled 'New node' and contains the following fields:

- Node name:** Nodeworker-1
- Type:** Permanent Agent (selected). A description below states: 'Adds a plain, 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.'
- Create:** A blue button at the bottom.

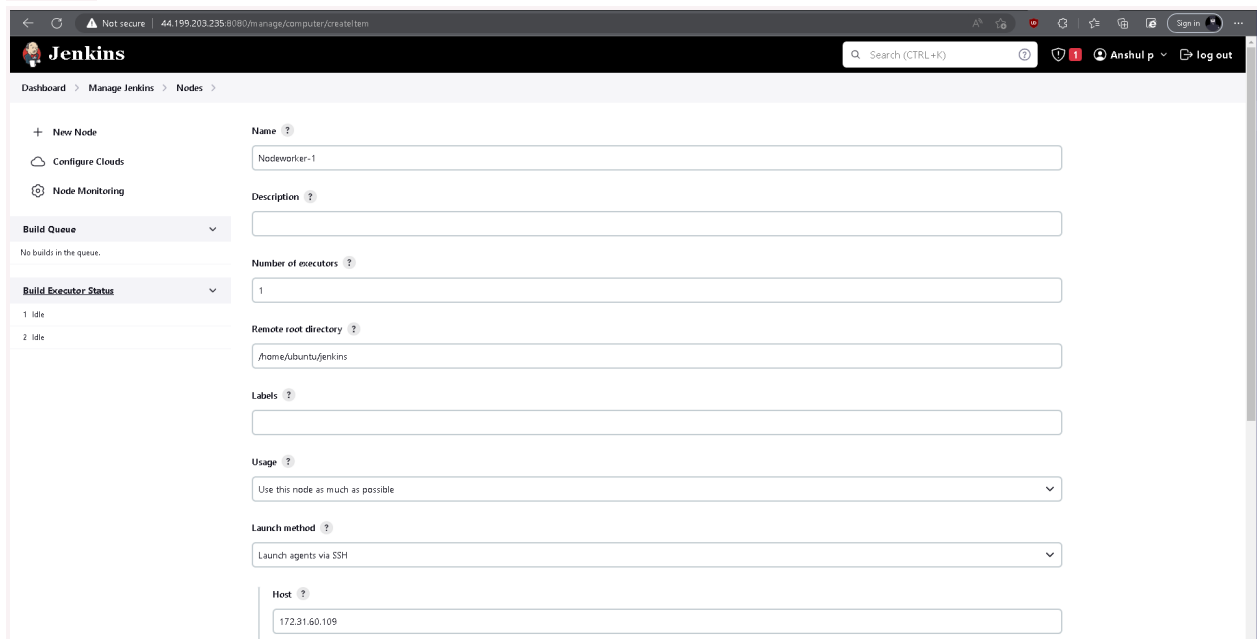
Give your node a name and move forward.

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.



The screenshot shows the Jenkins 'New node' page with the configuration form filled out. The fields are as follows:

- Name:** Nodeworker-1
- Description:** (empty)
- Number of executors:** 1
- Remote root directory:** /home/ubuntu/jenkins
- Labels:** (empty)
- Usage:** Use this node as much as possible
- Launch method:** Launch agents via SSH
- Host:** 172.31.60.109

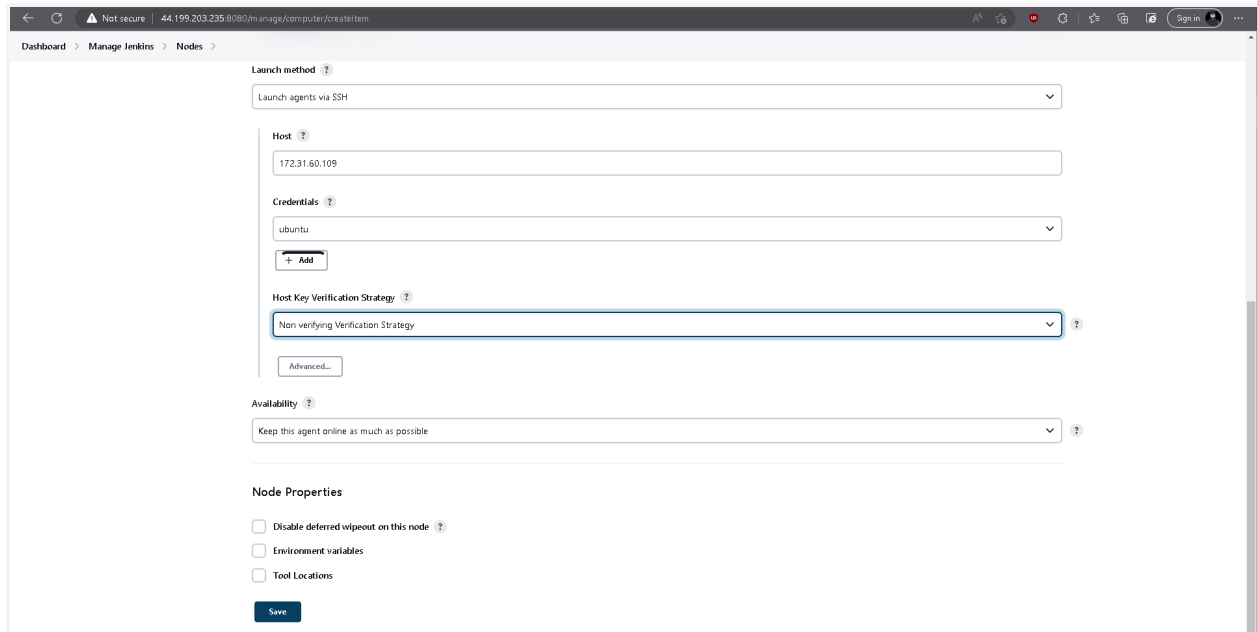
Now we need to add credentials.

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.



Dashboard > Manage Jenkins > Nodes >

Launch method ?  
Launch agents via SSH

Host ?  
172.31.60.109

Credentials ?  
ubuntu  
+ Add

Host Key Verification Strategy ?  
Non verifying Verification Strategy ?  
Advanced...

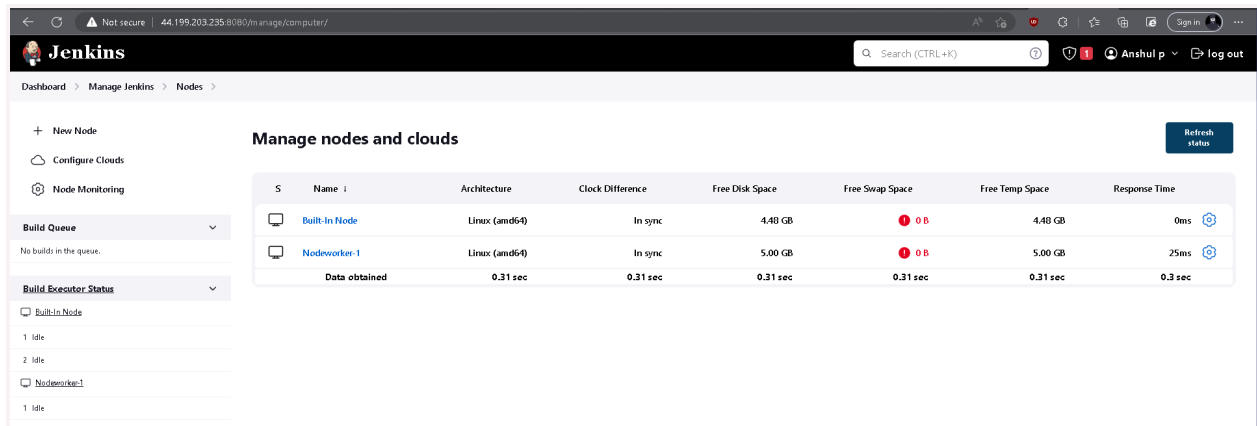
Availability ?  
Keep this agent online as much as possible

Node Properties

- ☐ Disable deferred wipeout on this node ?
- ☐ Environment variables
- ☐ Tool Locations

Save

Refresh the status and you can see node is connected.



Jenkins

Dashboard > Manage Jenkins > Nodes >

+ New Node  
Configure Clouds  
Node Monitoring

Build Queue  
No builds in the queue.

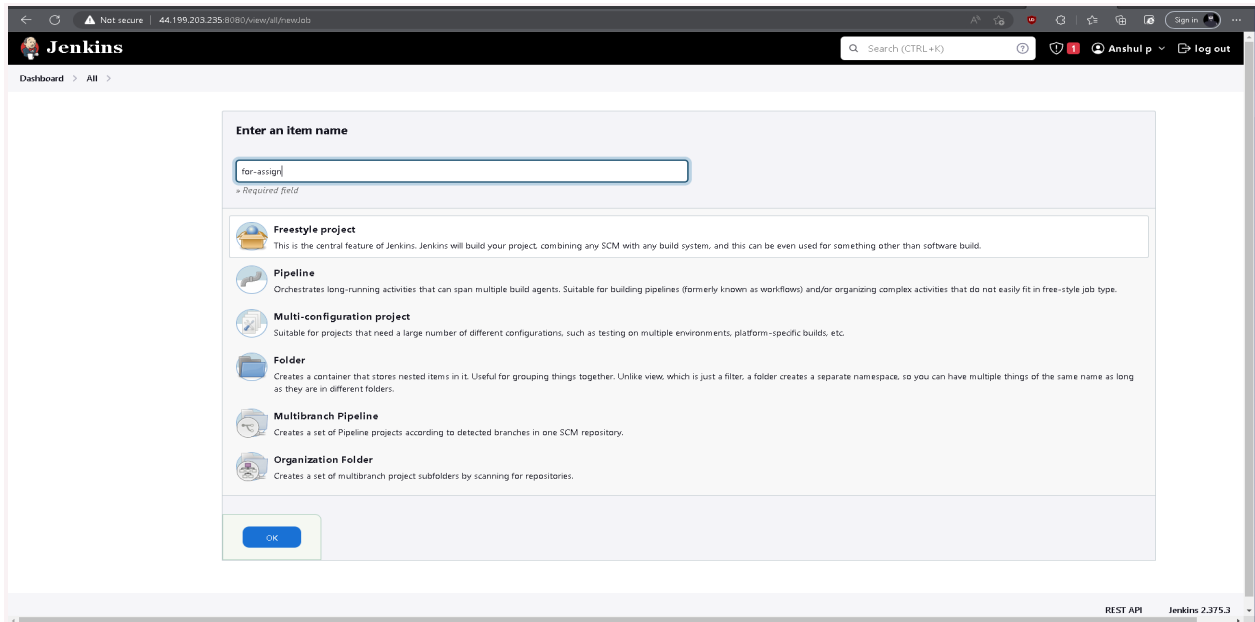
Build Executor Status  
Built-In Node  
1 Idle  
2 Idle  
Nodeworker-1  
1 Idle

Manage nodes and clouds Refresh status

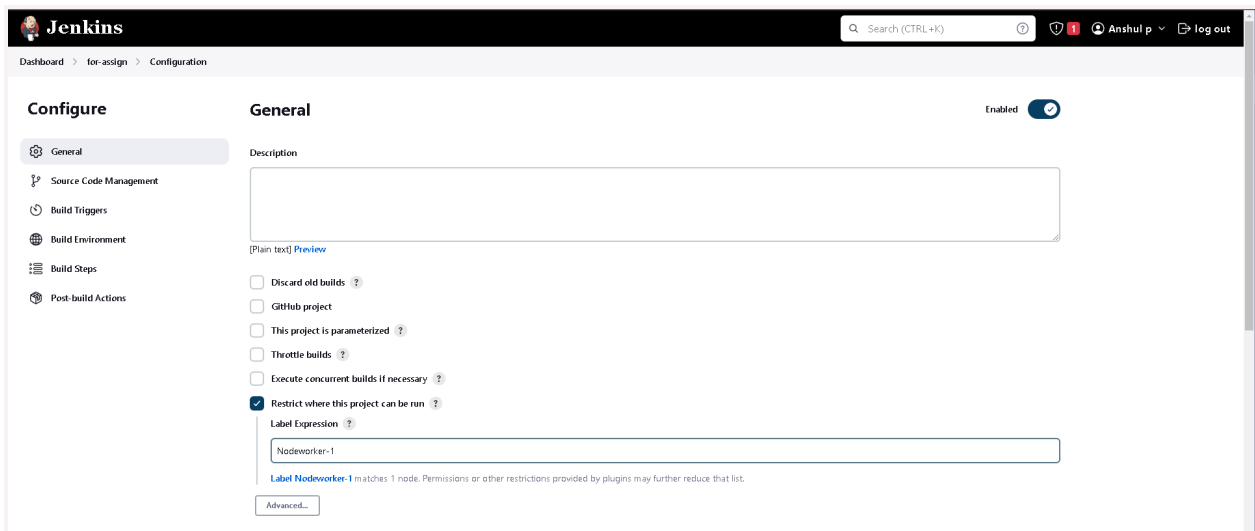
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.48 GB	0 B	4.48 GB	0ms
	Nodeworker-1	Linux (amd64)	In sync	5.00 GB	0 B	5.00 GB	25ms
	Data obtained	0.31 sec	0.31 sec	0.31 sec	0.31 sec	0.31 sec	0.3 sec

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**

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-60-109:~$ java --version
openjdk 11.0.17 2022-10-19
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-1ubuntu222.04; mixed mode, sharing)
ubuntu@ip-172-31-60-109:~$ mkdir test
ubuntu@ip-172-31-60-109:~$ cd test
ubuntu@ip-172-31-60-109:~/test$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:     git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:     git branch -m <name>
Initialized empty Git repository in /home/ubuntu/test/.git/
ubuntu@ip-172-31-60-109:~/test$
```

**Create a file and commit it to create a master branch.**

**touch file1.txt**

**Git add .**

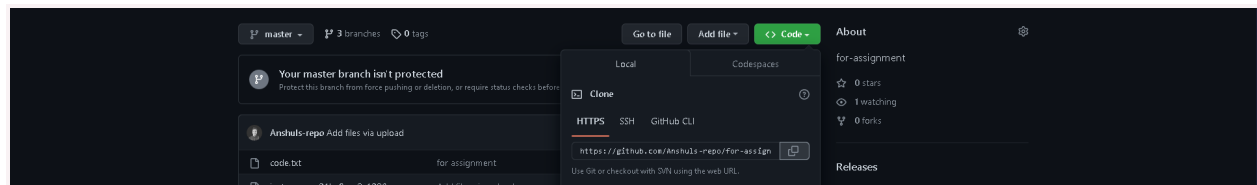
**Git commit -m “<any msg>”**

```
hint:
hint:     git branch -m <name>
Initialized empty Git repository in /home/ubuntu/test/.git/
ubuntu@ip-172-31-60-109:~/test$ touch file.txt
ubuntu@ip-172-31-60-109:~/test$ git add file.txt
ubuntu@ip-172-31-60-109:~/test$ git commit -m "for assign"
[master (root-commit) 53dfbad] for assign
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file.txt
ubuntu@ip-172-31-60-109:~/test$
```

**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.**

```
ubuntu@ip-172-31-60-109:~/test$ git push --help
ubuntu@ip-172-31-60-109:~/test$ git remote set-url origin https://github.com/Anshuls-repo/for-assignment2.git
ubuntu@ip-172-31-60-109:~/test$ git push --all
Username for 'https://github.com': Anshuls-repo
Password for 'https://Anshuls-repo@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 221 bytes | 221.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Anshuls-repo/for-assignment2.git
 * [new branch]      master -> master
ubuntu@ip-172-31-60-109:~/test$
```



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

```
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 221 bytes | 221.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Anahuls-repo/for-assignment2.git
* [new branch]      master -> master
ubuntu@ip-172-31-60-109:~/test$ git branch develop
ubuntu@ip-172-31-60-109:~/test$ git checkout develop
* develop
Switched to branch 'develop'
ubuntu@ip-172-31-60-109:~/test$ git branch
* develop
master
ubuntu@ip-172-31-60-109:~/test$
```

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

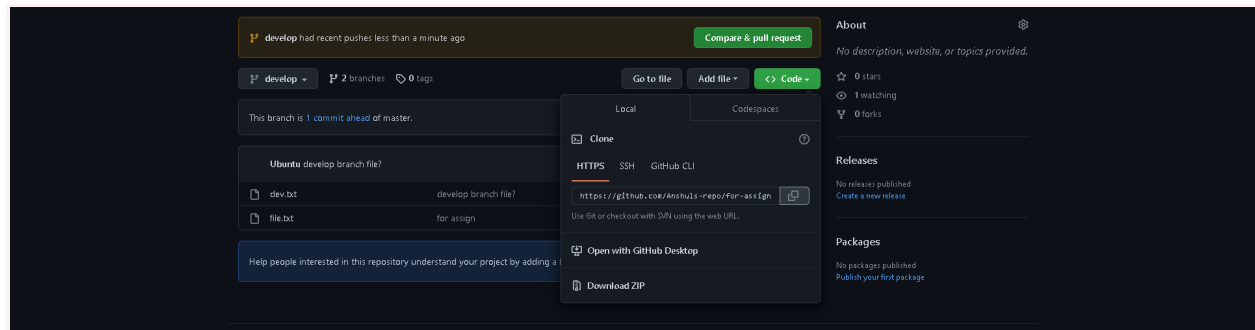
```
and the repository exists.
ubuntu@ip-172-31-60-109:~/test$ git push origin develop
Username for 'https://github.com': Anahuls-repo
Password for 'https://Anahuls-repo@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:   https://github.com/Anahuls-repo/for-assignment2/pull/new/develop
remote:
To https://github.com/Anahuls-repo/for-assignment2.git
* [new branch]      develop -> develop
ubuntu@ip-172-31-60-109:~/test$
```

Create and push a file in develop branch: touch dev.txt

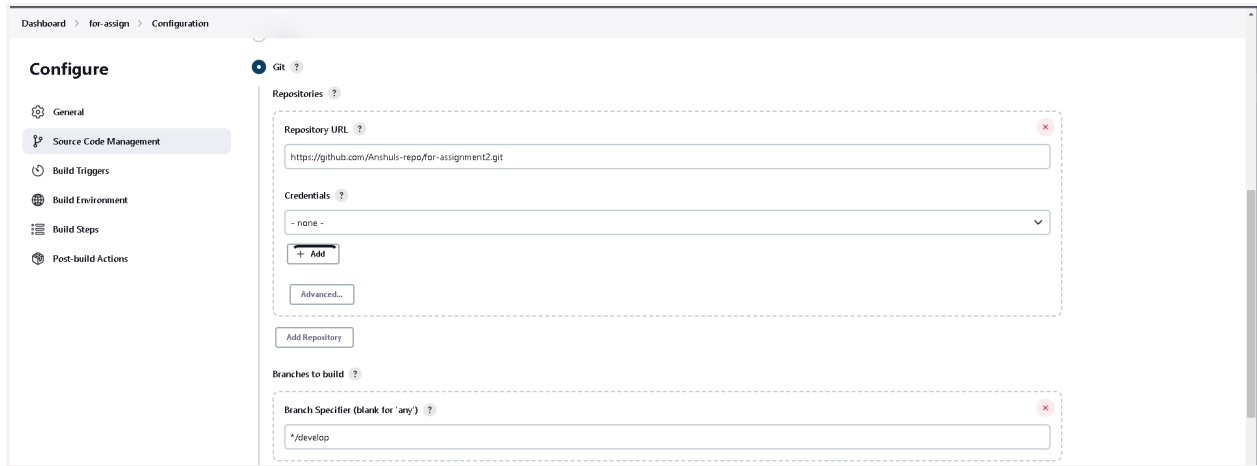
Git add.

Git push --all

Copy the https link.



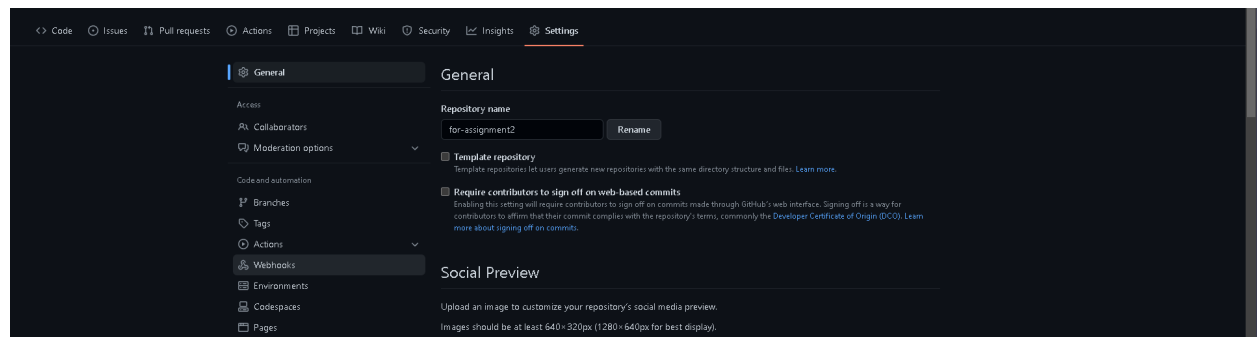
**Paste the link in git section in source code management. Also change the branch name to “develop”.**



The screenshot shows the Jenkins 'Configure' page for a job named 'for-assign'. The 'Source Code Management' section is selected in the left sidebar. Under the 'Git' configuration, the 'Repository URL' is set to 'https://github.com/Anshuls-repo/for-assignment2.git'. The 'Credentials' dropdown is set to '- none -'. Below this, there is an '+ Add' button and an 'Advanced...' link. An 'Add Repository' button is also present. In the 'Branches to build' section, the 'Branch Specifier (blank for \'any\')' is set to '\*/develop'.

**For automatic trigger, in ‘Build trigger’ choose ‘GitHub hook trigger for GITScm polling’.**

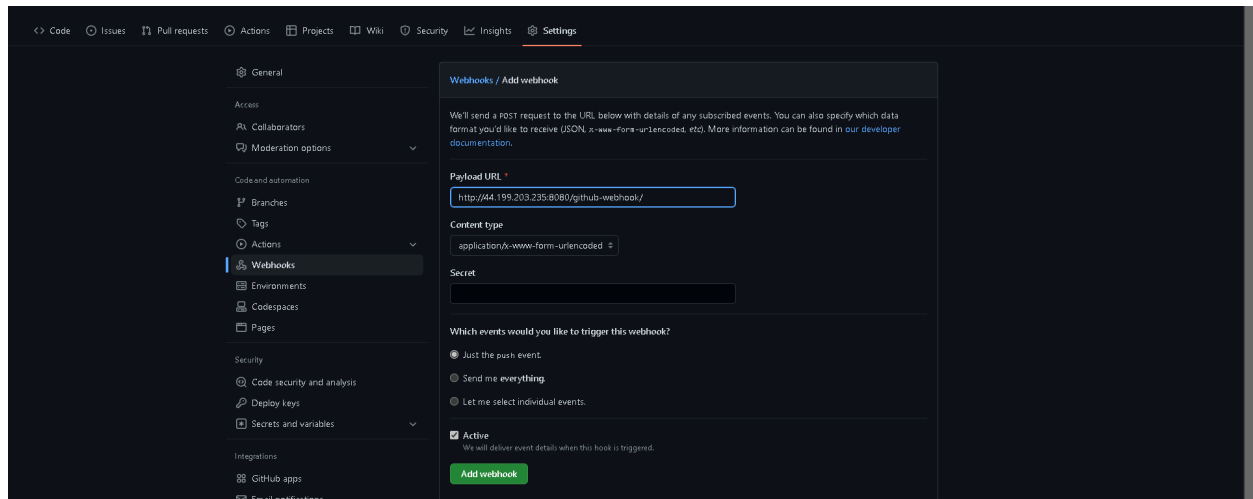
**Also we need to create a webhook. Goto your github then choose: settings>>webhook.**



The screenshot shows the GitHub 'Settings' page for a repository named 'for-assignment2'. The 'General' tab is selected in the left sidebar. The 'Repository name' is 'for-assignment2'. There are checkboxes for 'Template repository' and 'Require contributors to sign off on web-based commits'. The 'Social Preview' section is also visible, with instructions to upload an image for the repository's social media preview.

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

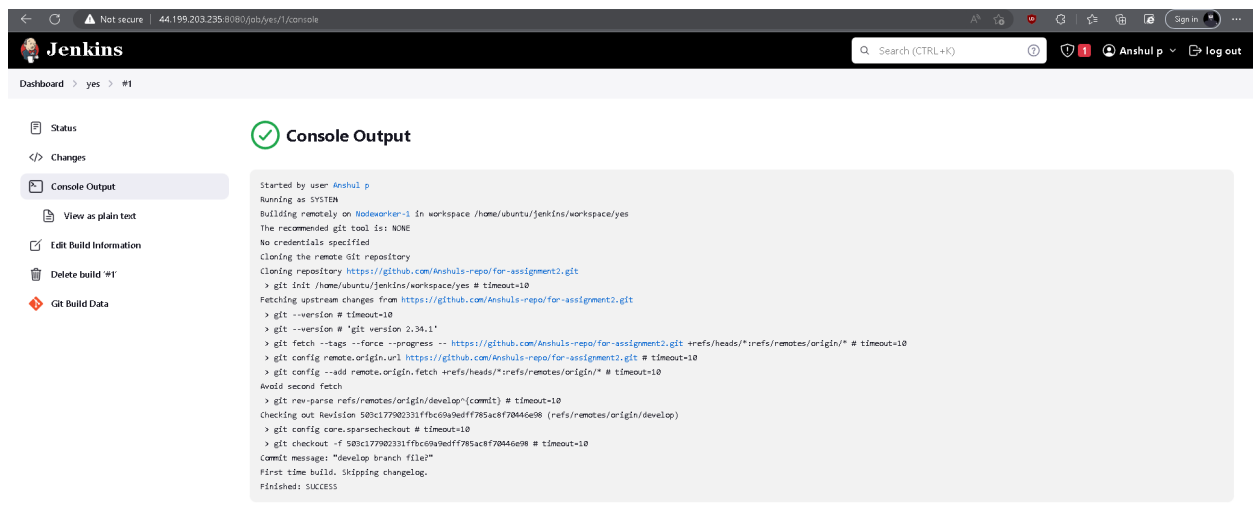
## jenkins\_ip/github-webhook/



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.



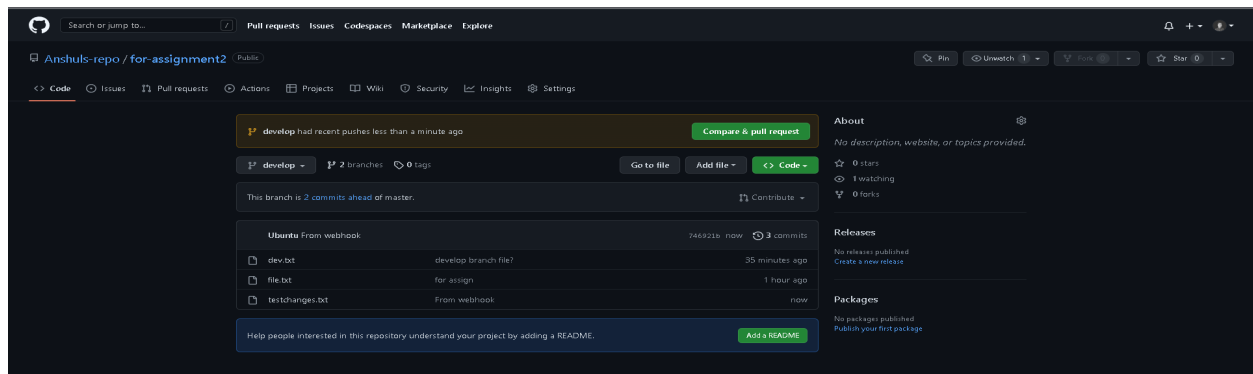
Now, if we make any changes in git repo in dev branch they will be reflected here.

Let us go to git folder and make some changes.

cd

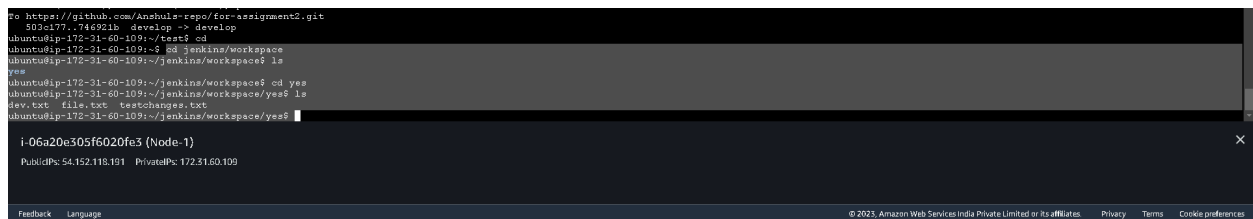
```
cd test
touch testchanges.txt
Git add .
Git commit -m "<message>"
Git push --all
Give your credentials and token.
```

New file has been added to repo.



Let us check if changes has been reflected in jenkins workspace.

```
cd
cd jenkins/workspace/yes
ls
```



**Conclusion:** Whenever changes are made to develop branch in git repo, they will reflect to the 'workspace' directory in the worker node.