Project: Capstone 1

You have been hired as a Sr. DevOps Engineer in Abode Software. They want to implement DevOps Lifecycle in their company. You have been asked to implement this lifecycle as fast as possible. Abode Software is a product-based company and their product is available on this GitHub link.

https://github.com/hshar/website.git

Following are the specifications of the lifecycle:

1. Install the necessary software on the machines using a configuration management tool

2. Git workflow has to be implemented

3. CodeBuild should automatically be triggered once a commit is made to master branch or develop branch. a. If a commit is made to master branch, test and push to prod b. If a commit is made to develop branch, just test the product, do not push to prod

4. The code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to GitHub. Use the following pre-built container for your application: hshar/webapp. The code should reside in '/var/www/html'

5. The above tasks should be defined in a Jenkins Pipeline with the following jobs:

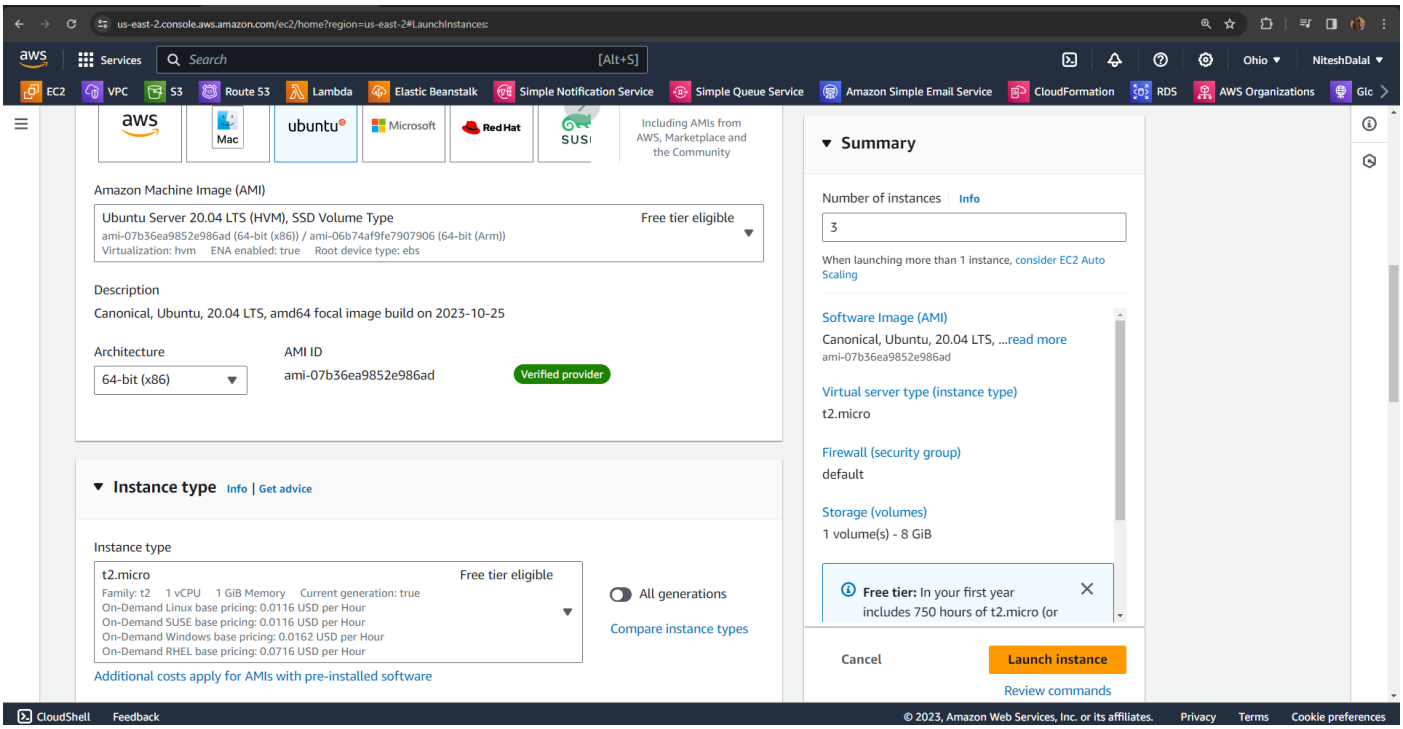
    a. Job1: Build

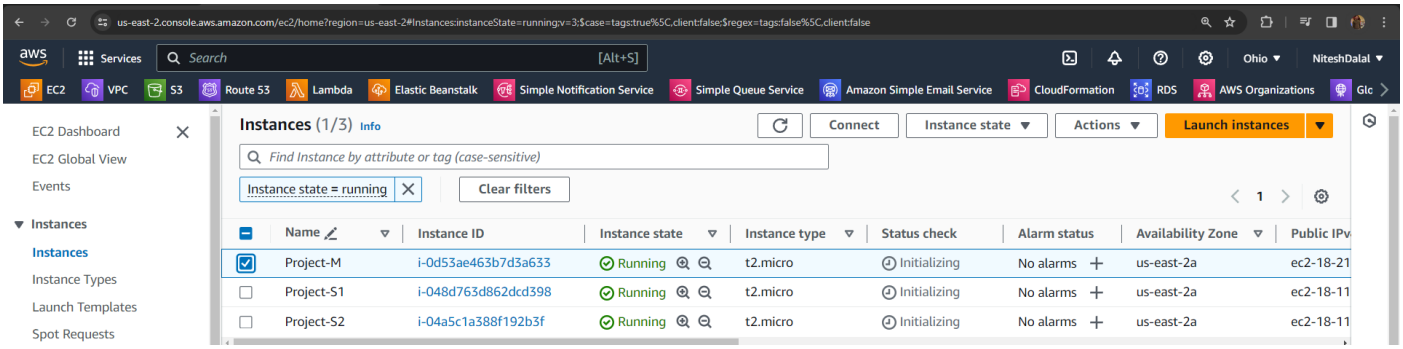
    b. Job2: Test

    c. Job3: Prod

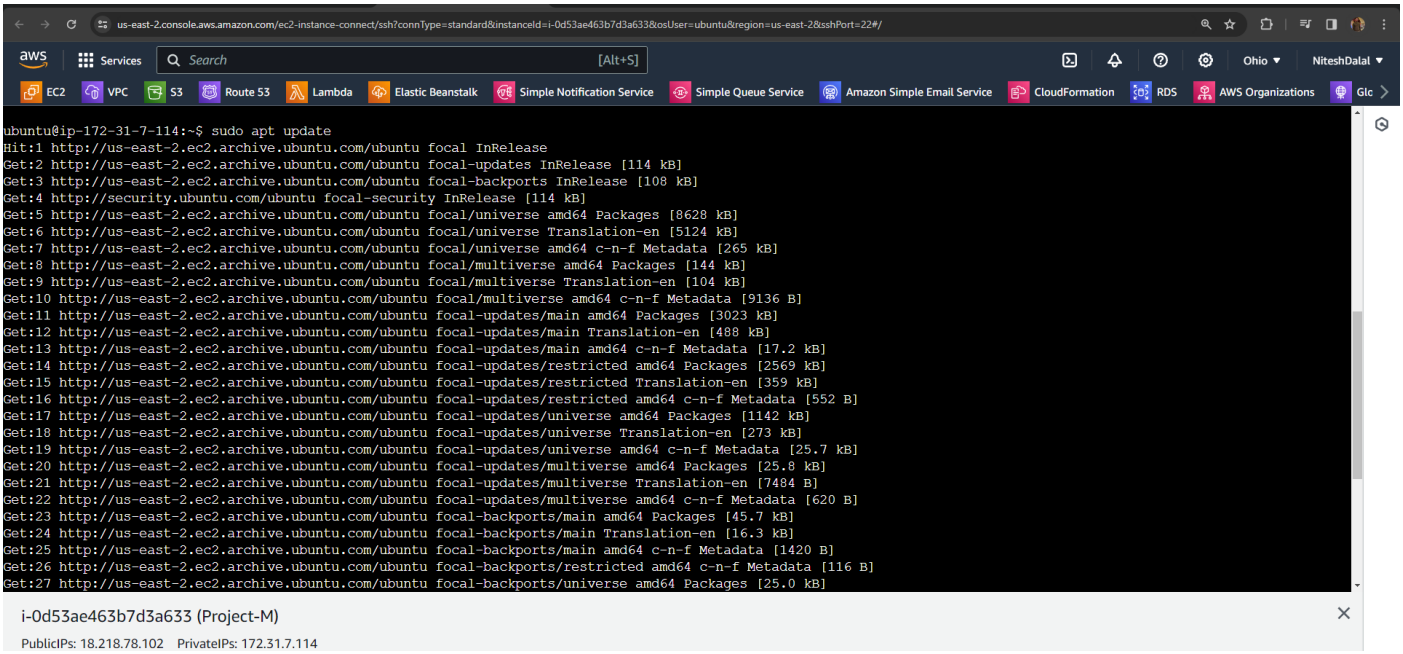
Solution:

1. As per the requirement we need to have 3 EC2 instances. Also enabling it with all traffic enabled & with Ubuntu as AMI.

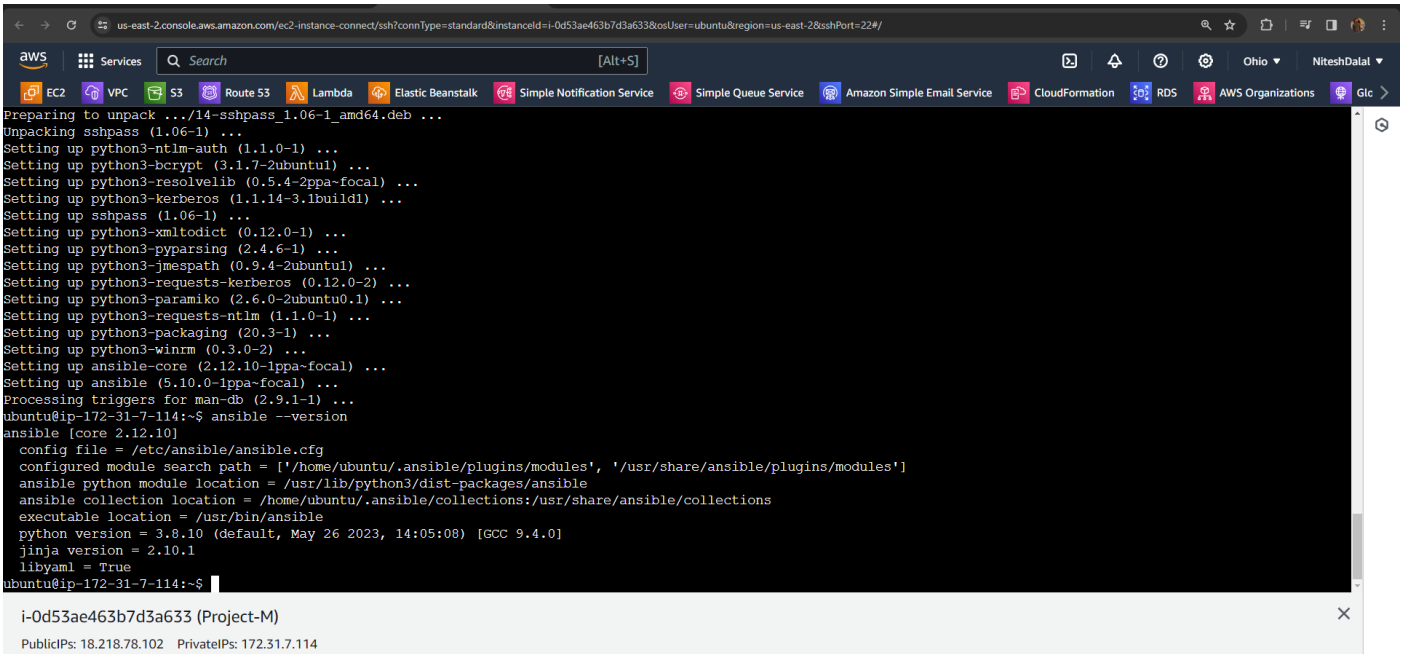




2. Now connect with master instance and first update your machine.



3. Installing the configuration management tool inside it i.e. Ansible. (from official documentation)



4. Our next step is to make configuration of ansible cluster. So, first we will make passwordless SSH connection b/w master & slave nodes. So, from master machine I will generate a key.

```
ubuntu@ip-172-31-7-114:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:sVbH9c25YSbA74/M0L9pvMq0BpxmZjFXllxpR4XpCQQ ubuntu@ip-172-31-7-114
The key's randomart image is:
+---[RSA 3072]----+
|          E+. o+O|
|           oo.oX+|
|        . . +++*=|
|         + + o* o|
|        S . B  . |
|       .   X o   |
|          = =.=  |
|           o=.=.|
|            .+o+o|
+----[SHA256]-----+
```

5. Now from .ssh folder of master machine I will copy the id_rsa.pub file content. And then paste it to .ssh folder of slave machine in file name -> authorized keys.

```
ubuntu@ip-172-31-7-114:~$ cd .ssh
ubuntu@ip-172-31-7-114:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-7-114:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCrQA3zXQDmmAf99p3OW6d7EIxng8JffJ3Onx1S9R9clC52KOF99PwS0/owSoQ2htqj+EKoaOg/AQxqe/SaOcDJIar2qpAvExqeYuhDYANGioh1/XqbL8PtZ//gQcUzOo
ZoGLrEhcQUvgvIVtmrprtI/TPuHfS5pRBoZfk3CFexXWHsPKrlKg6/fKXfwaliRnq1znODOH4OX2S0pupAA6zmy2K0TCEKZDgfEHGpkm4X1Nf2lQU9v/d9IQpJWmZY1Q3N2MzVca7LLKjucjPUPqionyYWT14TSCTdfTg7
+KQ3xJ5ZHVtl6mOZquRj8km2dsubiQZI65EVusBK+muyO8gxHSVrk6f7yVBsYWqUlPTrIkJgwLjU/4F3hGXYBB1ibJQqOiaqJiqAVwN3oUFTjvox122crIC+oZeavrdUJj7E72v/vO27r6KAGWxhJOJ54kB8rAH51+ytIo
onURRyIu2ErPQCQWkJSBwc+LMm4TIVGt+mffYqVVufCPKwUwwZ/L0= ubuntu@ip-172-31-7-114
ubuntu@ip-172-31-7-114:~/.ssh$
```

i-0d53ae463b7d3a633 (Project-M)

PublicIPs: 18.218.78.102 PrivateIPs: 172.31.7.114

```
GNU nano 4.8                          authorized_keys                          Modified
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCqSvl+SNDJmfL8Q6tWB2nFOXgaoAiLod2N1PKAWMv1WC5iexmpZVWA9yGRrPybVckg1oggDyU3wSorWpz3nlUetTLppy7TEHQS/VG/+iMc6z2E3e1GULejsDKrt/TqR>
<AGWxhJOJ54kB8rAH51+ytIoonURRyIu2ErPQCQWkJSBwc+LMm4TIVGt+mffYqVVufCPKwUwwZ/L0= ubuntu@ip-172-31-7-114S
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket  M-Q Previous
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo  M-6 Copy Text  ^Q Where Was  M-W Next
```

i-048d763d862dcd398 (Project-S1)

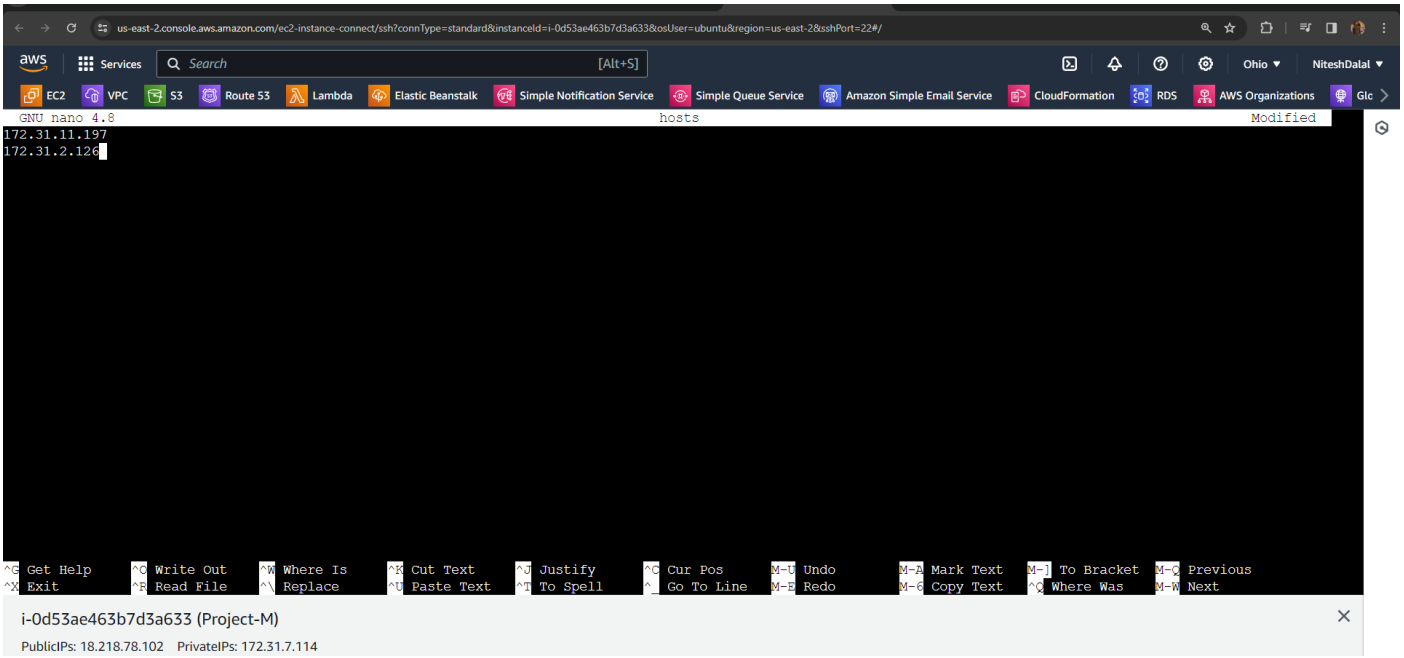
PublicIPs: 18.119.105.185 PrivateIPs: 172.31.11.197

```
GNU nano 4.8                          authorized_keys                          Modified
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCqSvl+SNDJmfL8Q6tWB2nFOXgaoAiLod2N1PKAWMv1WC5iexmpZVWA9yGRrPybVckg1oggDyU3wSorWpz3nlUetTLppy7TEHQS/VG/+iMc6z2E3e1GULejsDKrt/TqR>
<AGWxhJOJ54kB8rAH51+ytIoonURRyIu2ErPQCQWkJSBwc+LMm4TIVGt+mffYqVVufCPKwUwwZ/L0= ubuntu@ip-172-31-7-114
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos  M-U Undo  M-A Mark Text  M-] To Bracket  M-Q Previous
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo  M-6 Copy Text  ^Q Where Was  M-W Next
```

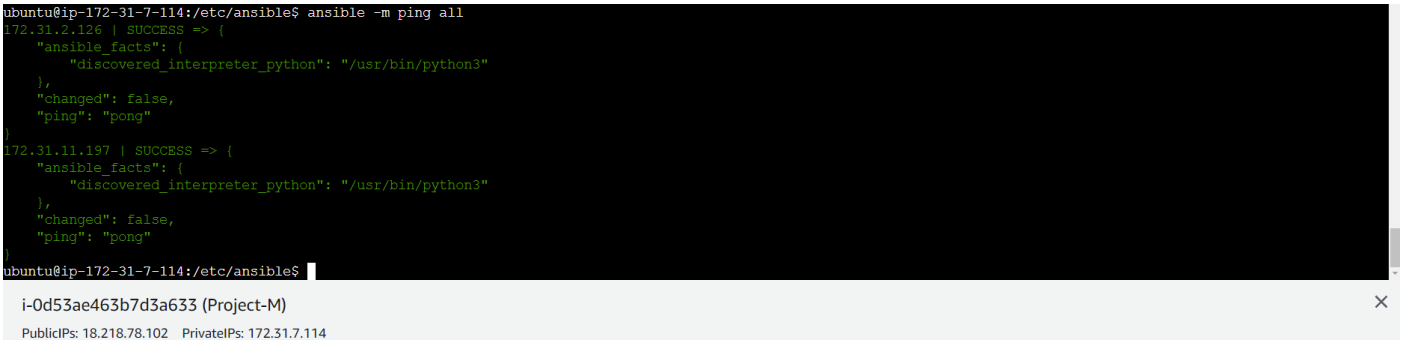
i-04a5c1a388f192b3f (Project-S2)

PublicIPs: 18.118.24.242 PrivateIPs: 172.31.2.126

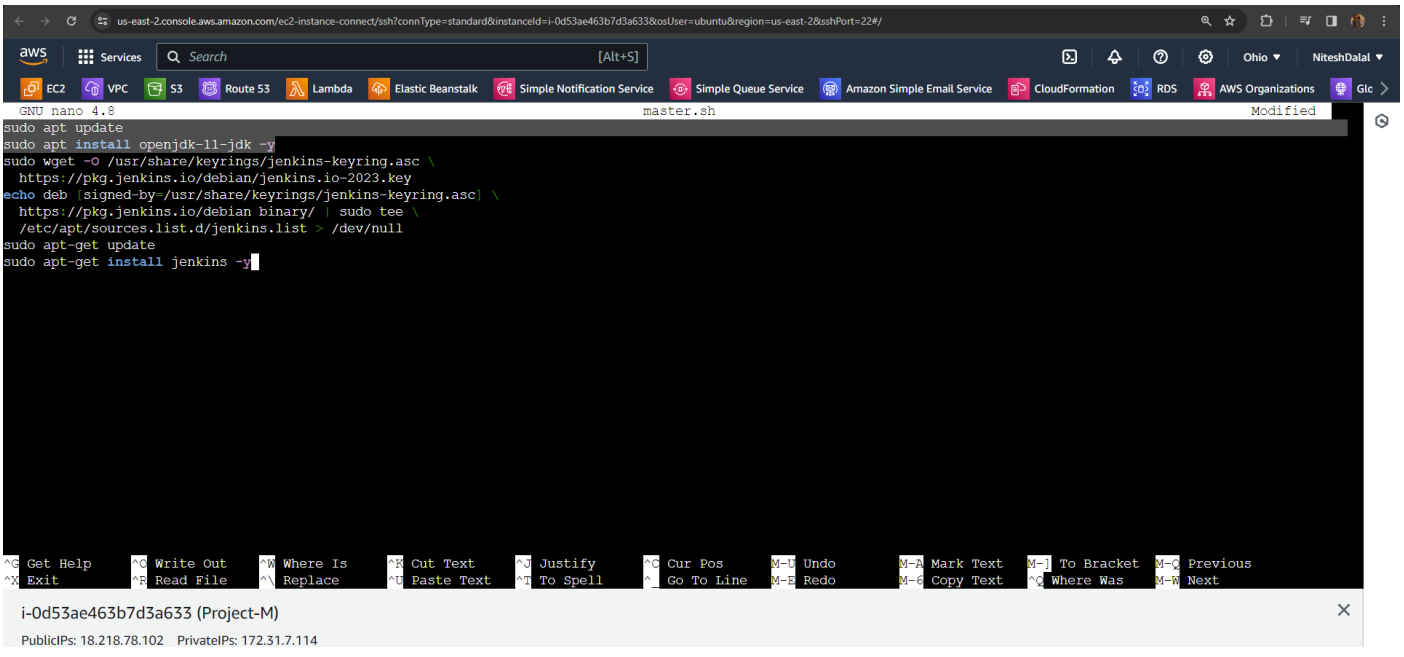
6. Now I will edit the host file for ansible slaves. Which is located in /etc/ansible/ location. Then mentioning the private IP's of both slaves in host file.

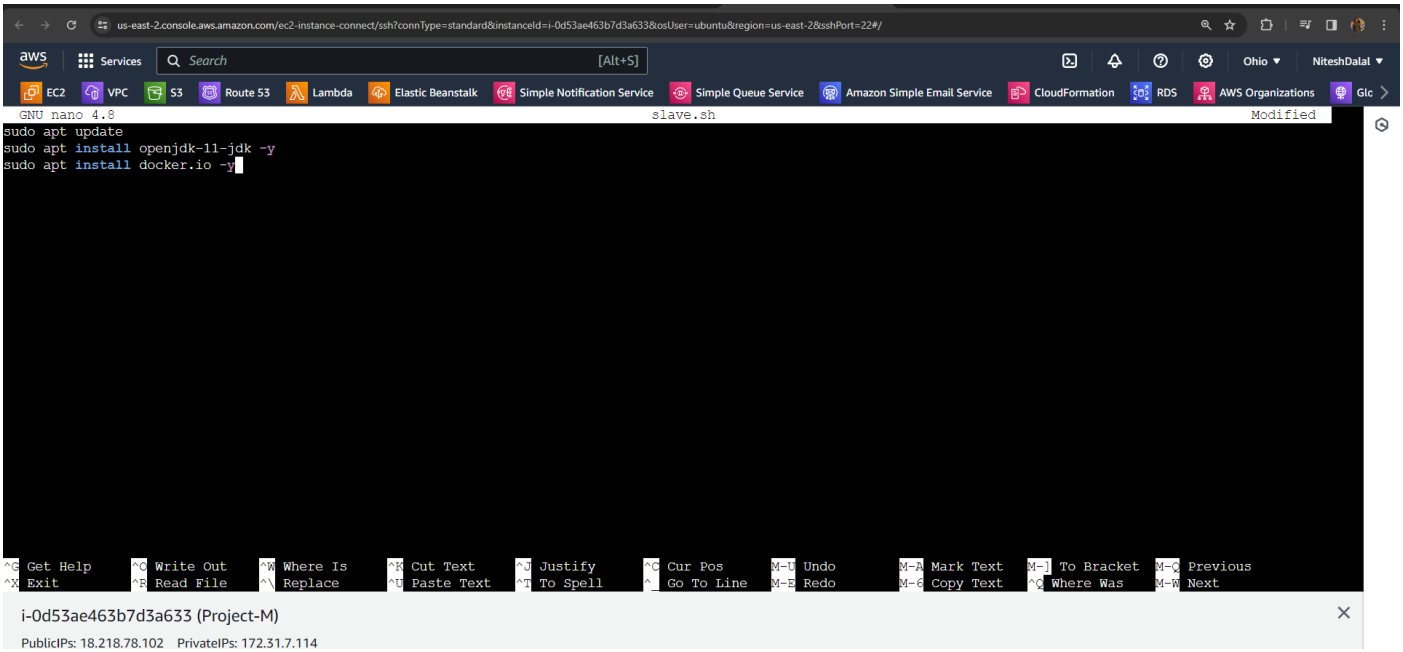


7. Test the connection with hosts.



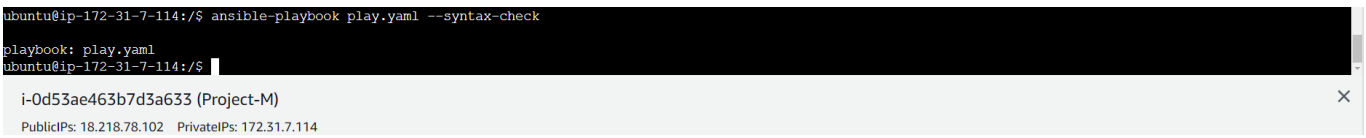
8. Now let's create a playbook file for installation of required tools. For master I will create -> master.sh and for slave instances I will create slave.sh





9. Now I will create a playbook file for installation.

10. Let’s do a syntax check.



11. Now doing a dry run of the code.

```
ubuntu@ip-172-31-7-114:/$ ansible-playbook play.yaml --check

PLAY [running tasks on localhost] ***********************************************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************************************************
ok: [localhost]

TASK [executing script on master machine] ***************************************************************************************************************************************
skipping: [localhost]

PLAY [running tasks on slaves] **************************************************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************************************************
ok: [172.31.11.197]
ok: [172.31.2.126]

TASK [executing script on slave machine] ****************************************************************************************************************************************
skipping: [172.31.11.197]
skipping: [172.31.2.126]

PLAY RECAP **********************************************************************************************************************************************************************
172.31.11.197              : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
172.31.2.126               : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

ubuntu@ip-172-31-7-114:/$
```

i-0d53ae463b7d3a633 (Project-M)

PublicIPs: 18.218.78.102 PrivateIPs: 172.31.7.114

12. Finally executing the playbook file.

```
ubuntu@ip-172-31-25-104:~$ ansible-playbook play.yaml

PLAY [running tasks on localhost] ***********************************************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************************************************
ok: [localhost]

TASK [executing script on master machine] ***************************************************************************************************************************************
changed: [localhost]

PLAY [running tasks on slaves] **************************************************************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************************************************************
ok: [172.31.17.205]
ok: [172.31.30.176]

TASK [executing script on slave machine] ****************************************************************************************************************************************
changed: [172.31.17.205]
changed: [172.31.30.176]

PLAY RECAP **********************************************************************************************************************************************************************
172.31.17.205              : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.176              : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

ubuntu@ip-172-31-25-104:~$
```

i-068ab839ae8fd3689 (Project-M)

PublicIPs: 18.60.217.76 PrivateIPs: 172.31.25.104

13. Let’s test it from slave machine.

```
ubuntu@ip-172-31-17-205:~$ java --version
openjdk 11.0.21 2023-10-17
OpenJDK Runtime Environment (build 11.0.21+9-post-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 11.0.21+9-post-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-17-205:~$ docker --version
Docker version 24.0.5, build 24.0.5-0ubuntu1~22.04.1
ubuntu@ip-172-31-17-205:~$
```

i-08f2573b44790c0d6 (Project-S1)

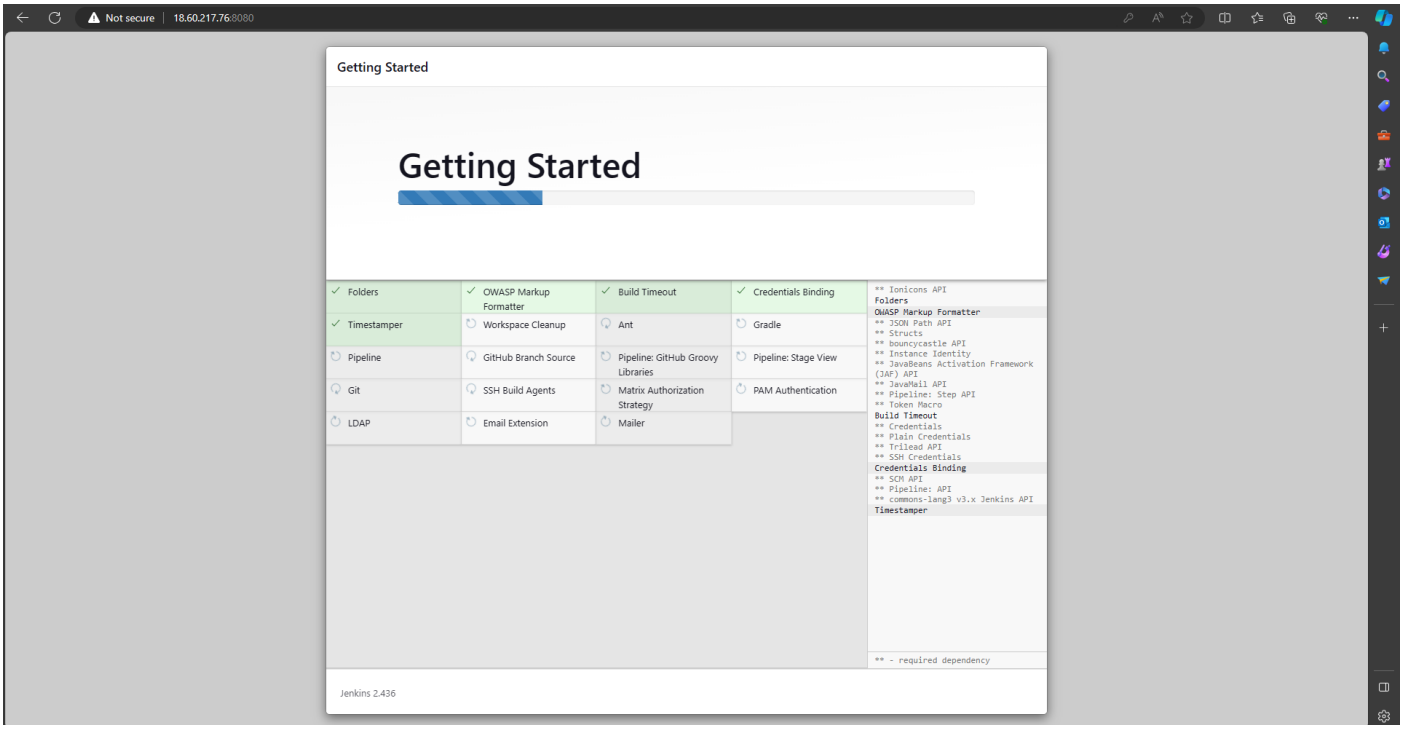
PublicIPs: 18.60.62.204 PrivateIPs: 172.31.17.205

14. Now next task is to setup Jenkins dashboard from master machine public IP. Pasted initial password & installed suggested plugins.

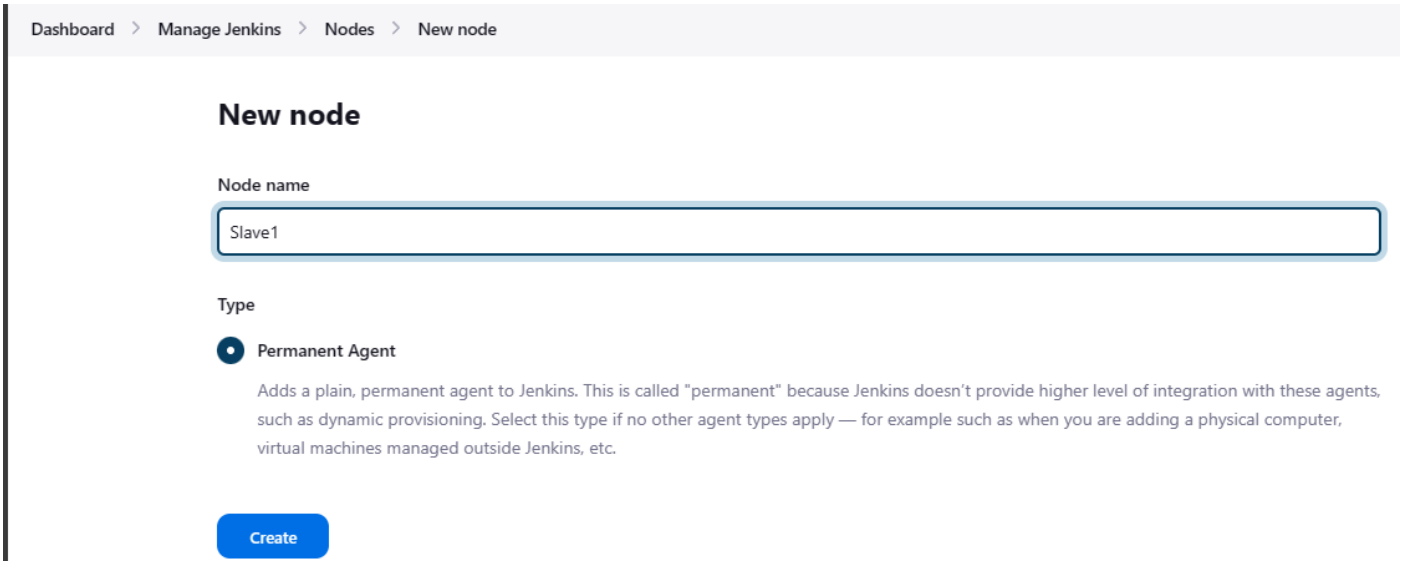
```
ubuntu@ip-172-31-25-104:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
a34952428b4e41f6884b9cadfed6f6d8
ubuntu@ip-172-31-25-104:~$ ^C
ubuntu@ip-172-31-25-104:~$
```

i-068ab839ae8fd3689 (Project-M)

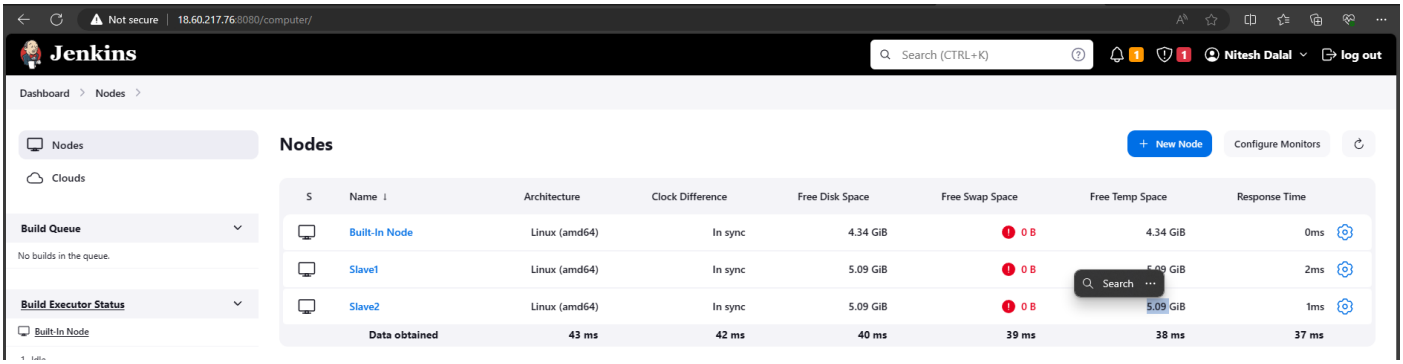
PublicIPs: 18.60.217.76 PrivateIPs: 172.31.25.104



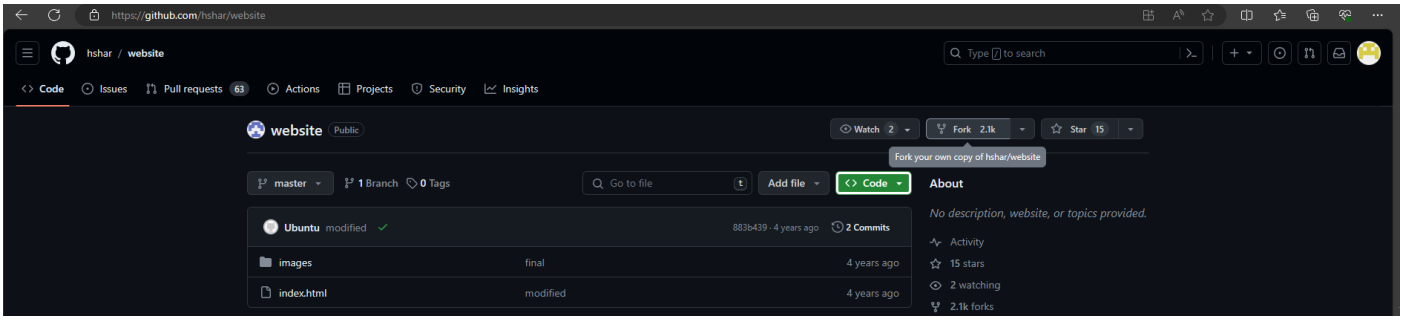
15. Now next task is to add our slave machines as two nodes of Jenkins.



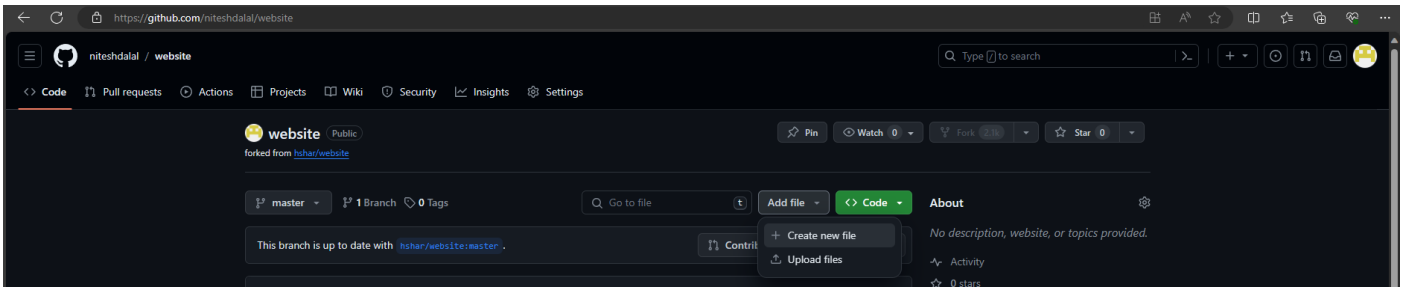
16. Repeat the same for slave2.



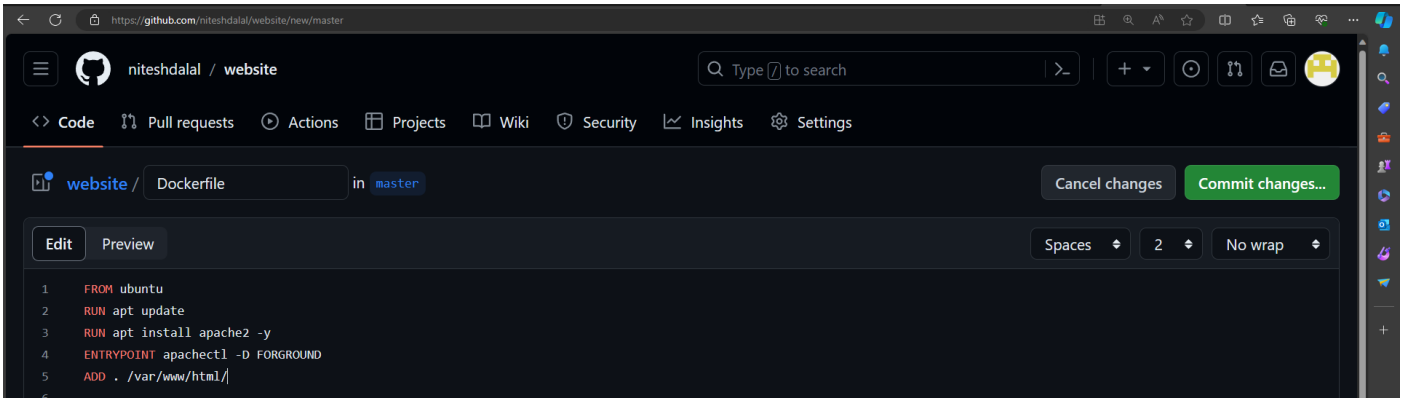
17. Now next we are going to create a repository on github.com by forking it from the project question url.



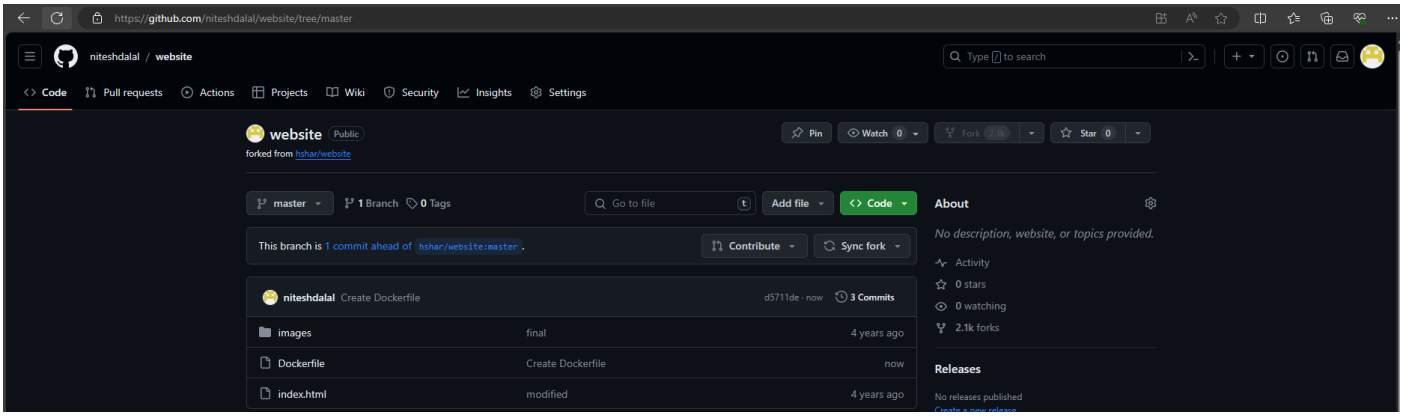
18. Now we will create a docker file on github.com from here.



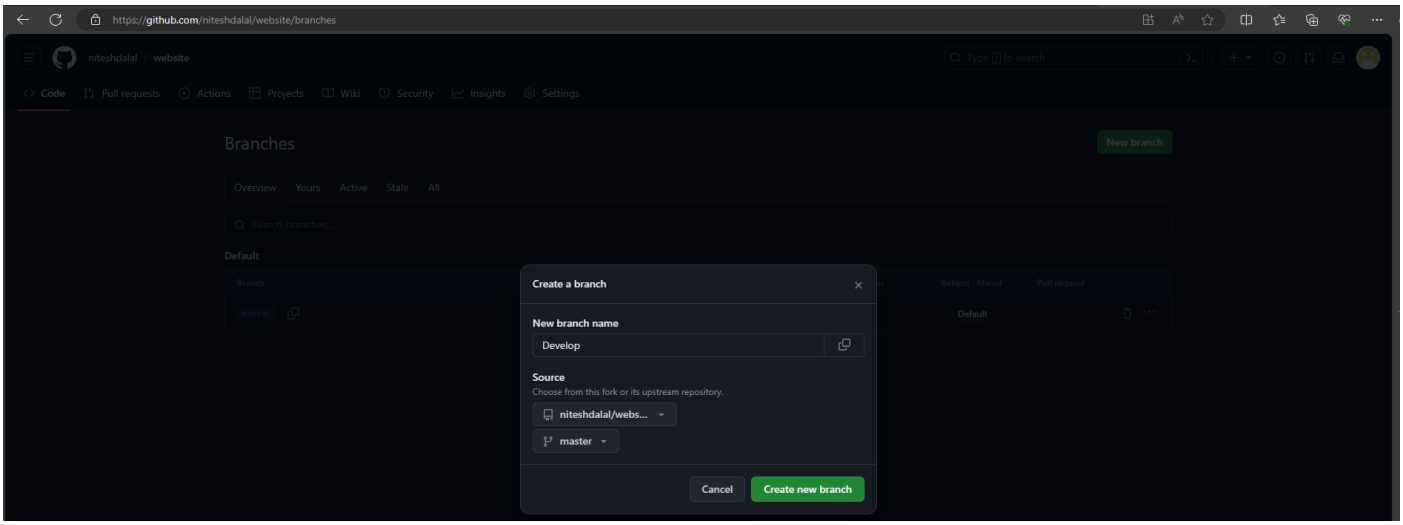
19. Write the code to install everything and copy the content in that specific folder.



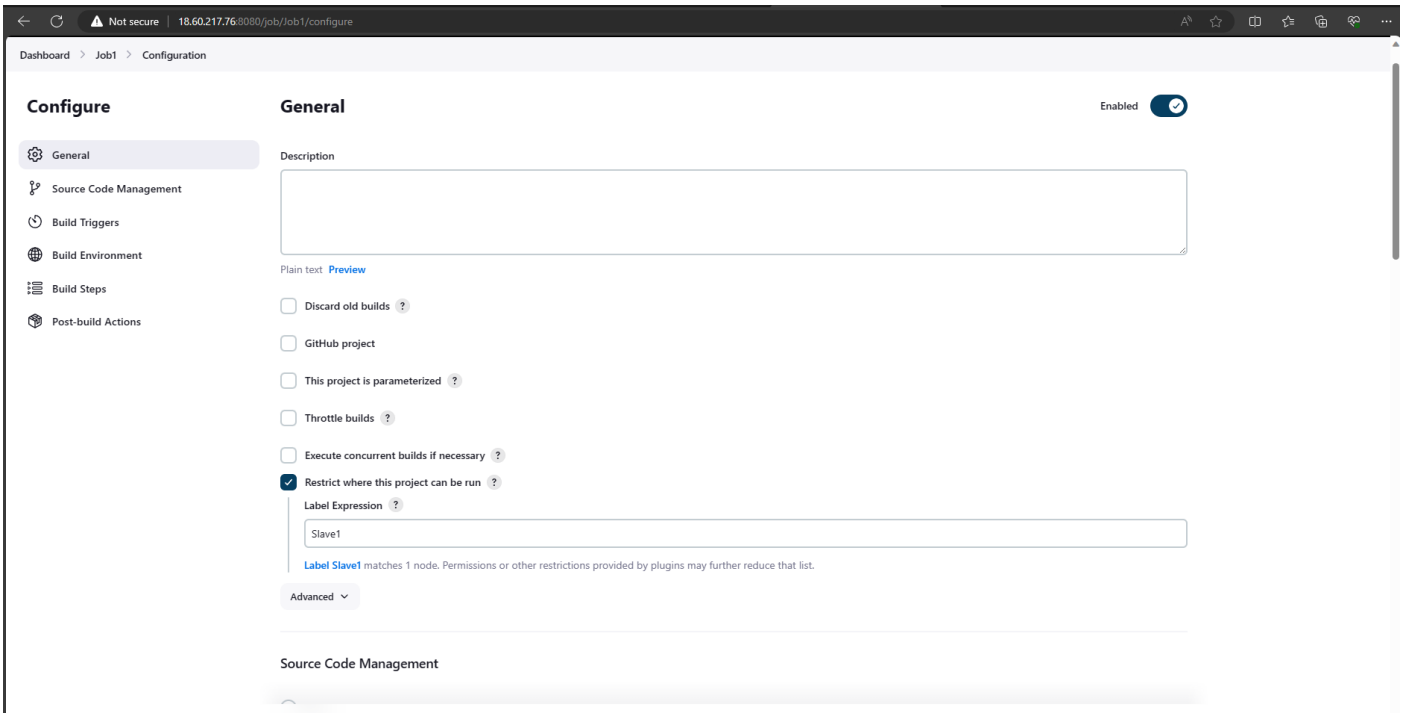
20. Commit the changes and it will create a docker file.

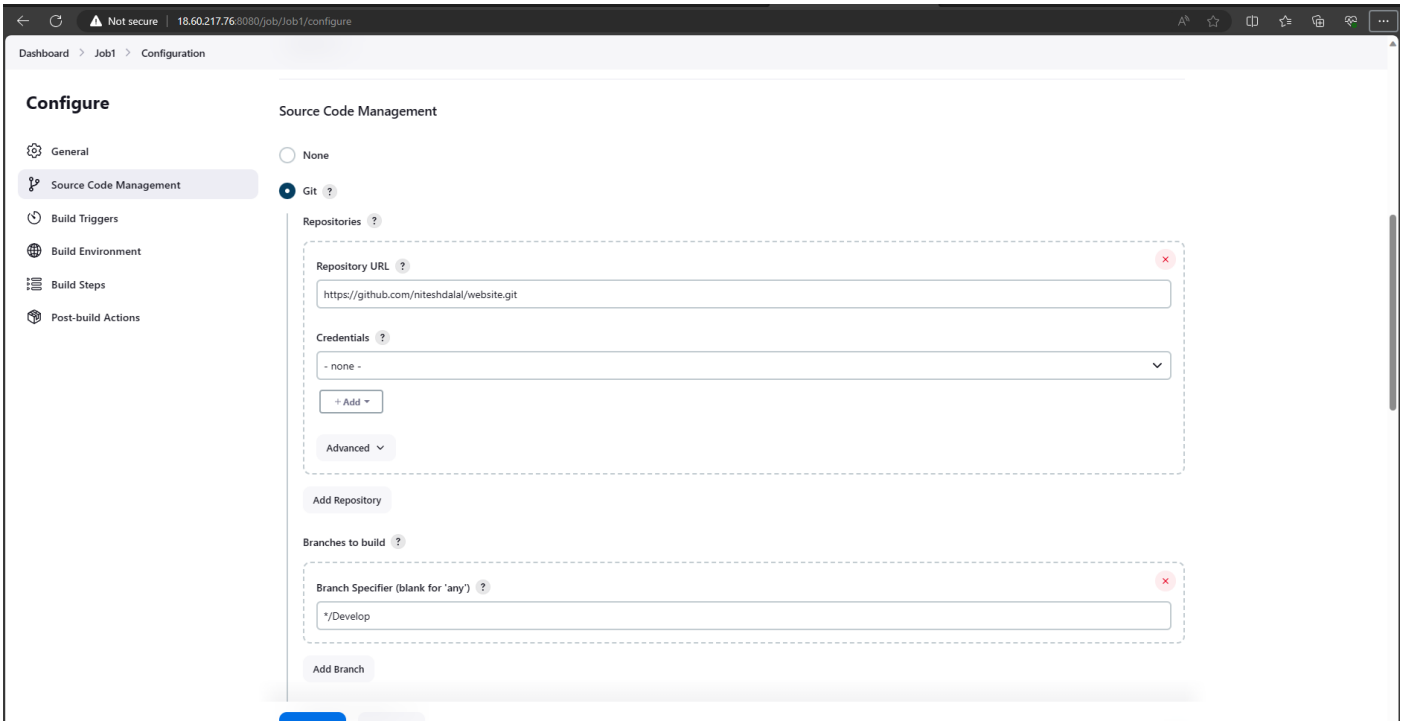


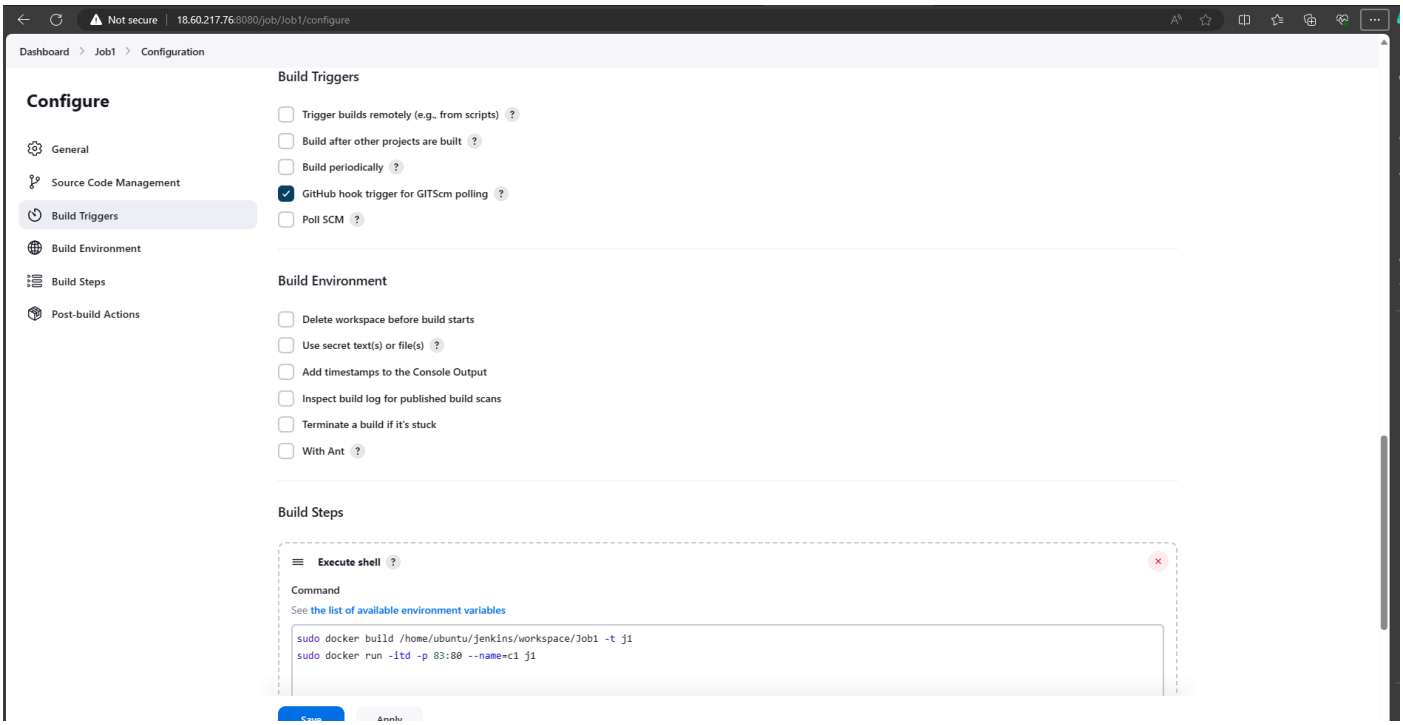
21. Now we need to create an additional branch as per question i.e. Develop.



22. Next we will create jobs in Jenkins dashboard.

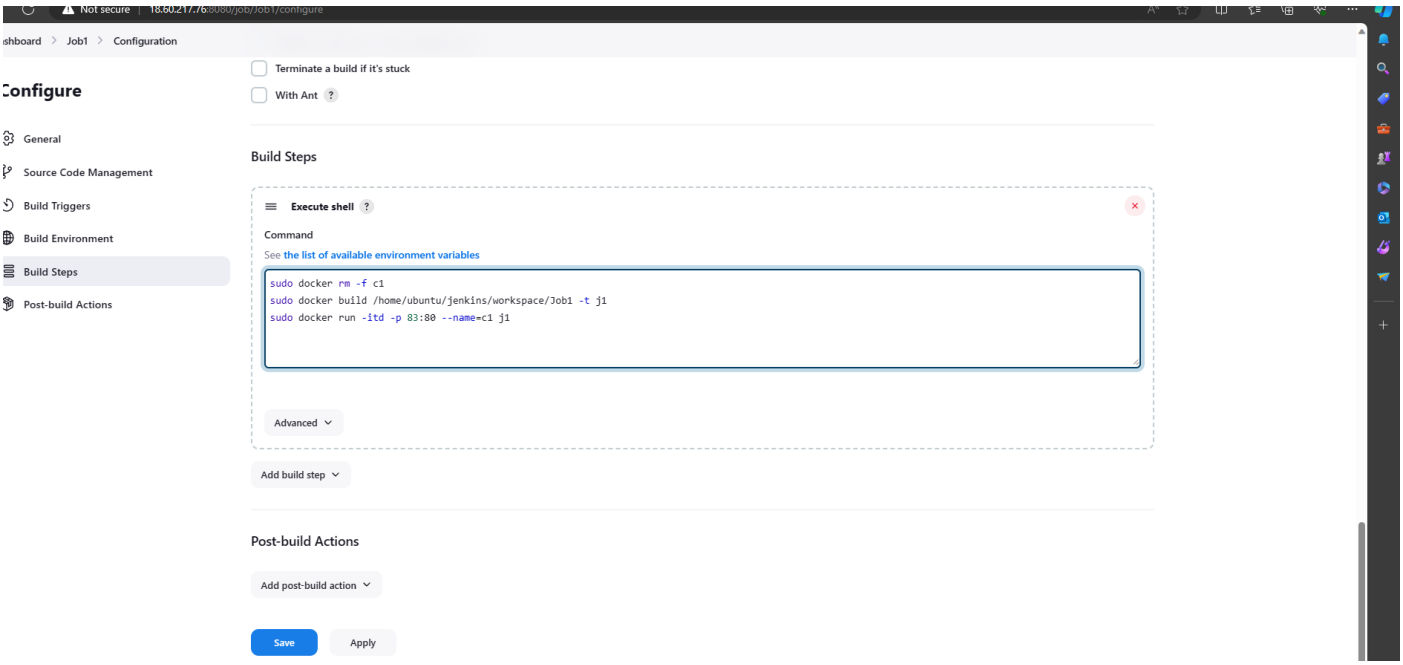


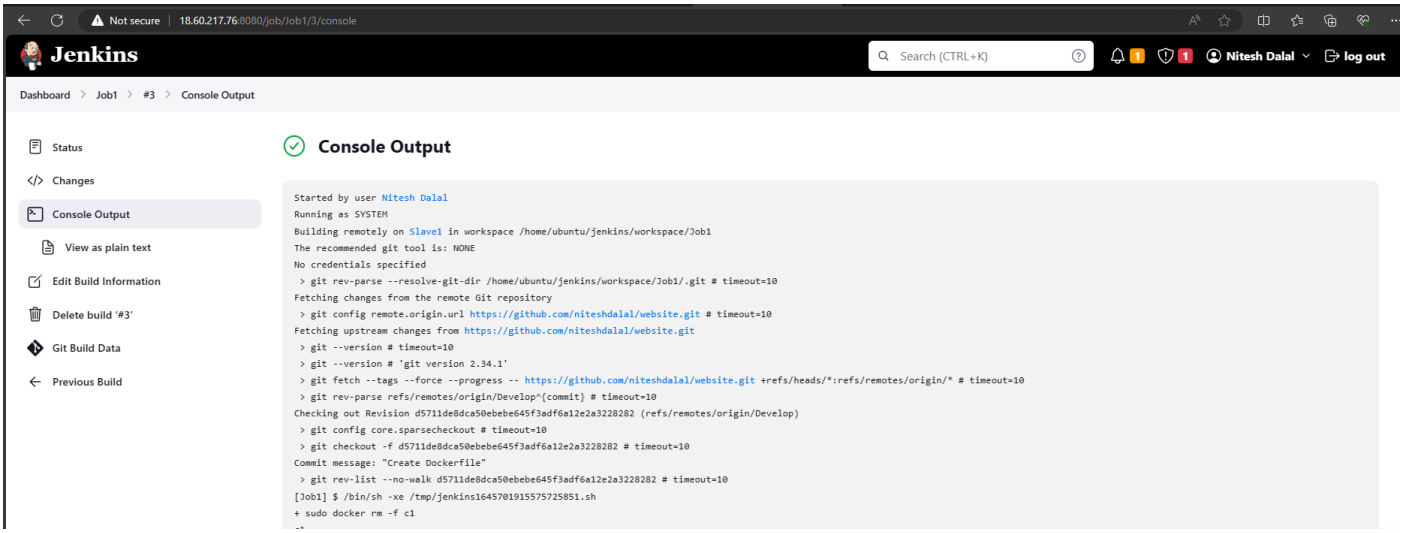




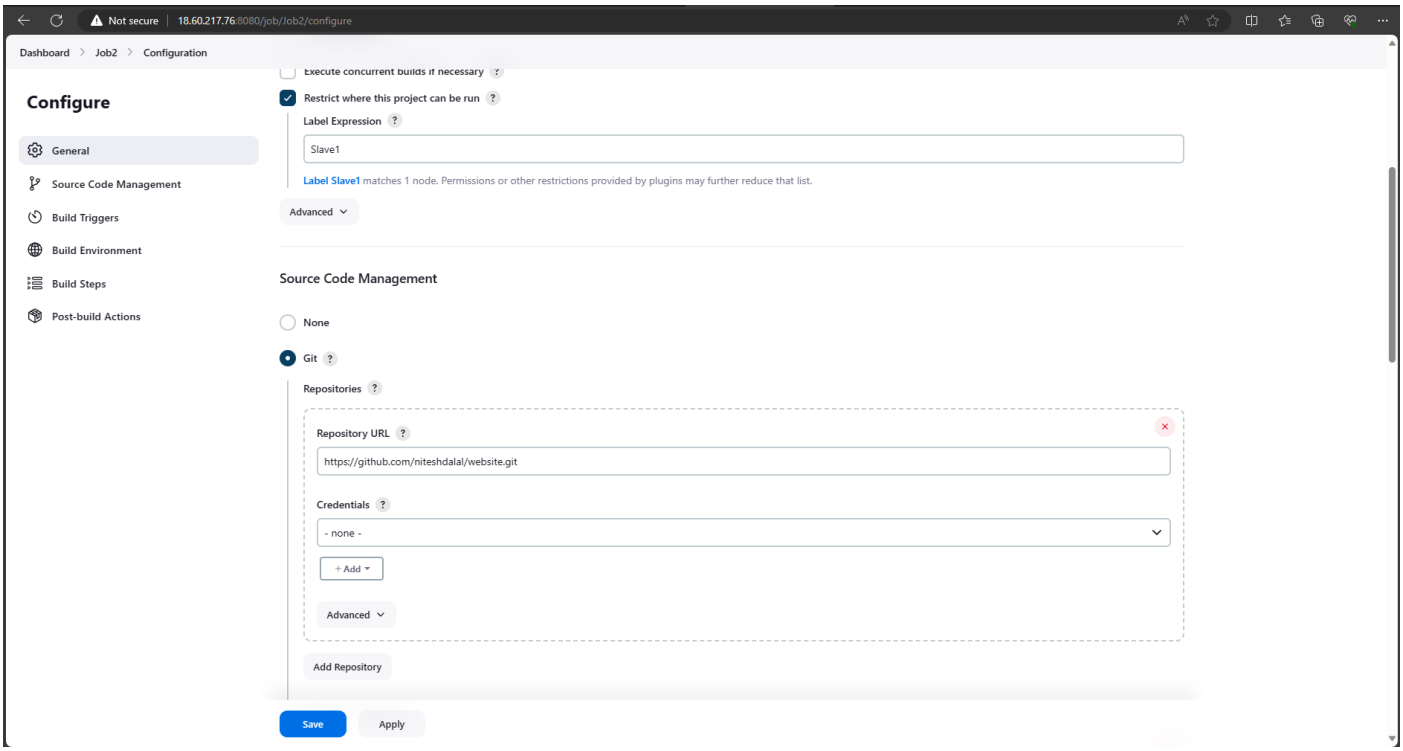
23. Now build the first job with Build now button of Jenkins dashboard and test it through the Slave1 public IP. As it is your testing environment.

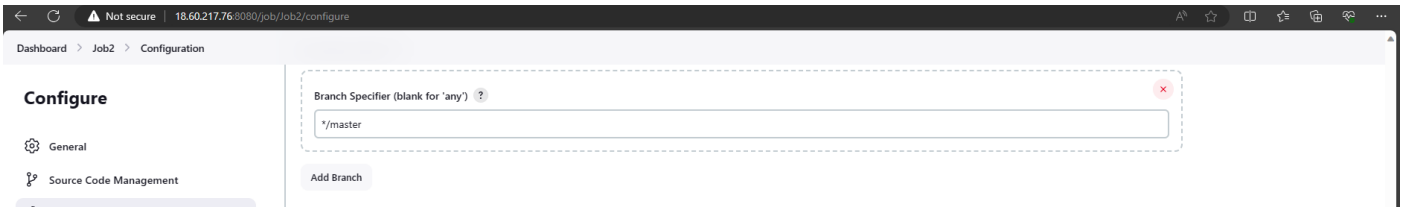
24. Once the first job is successfully build we need to re-configure the job in such a way to delete the existing container. To avoid the error on our next jobs creation.

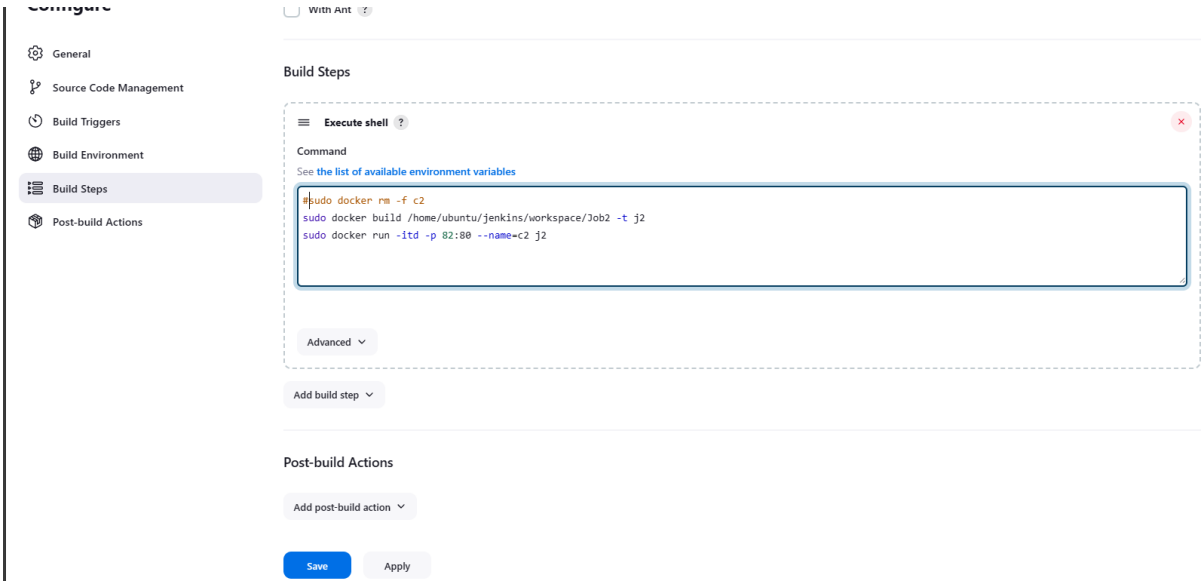




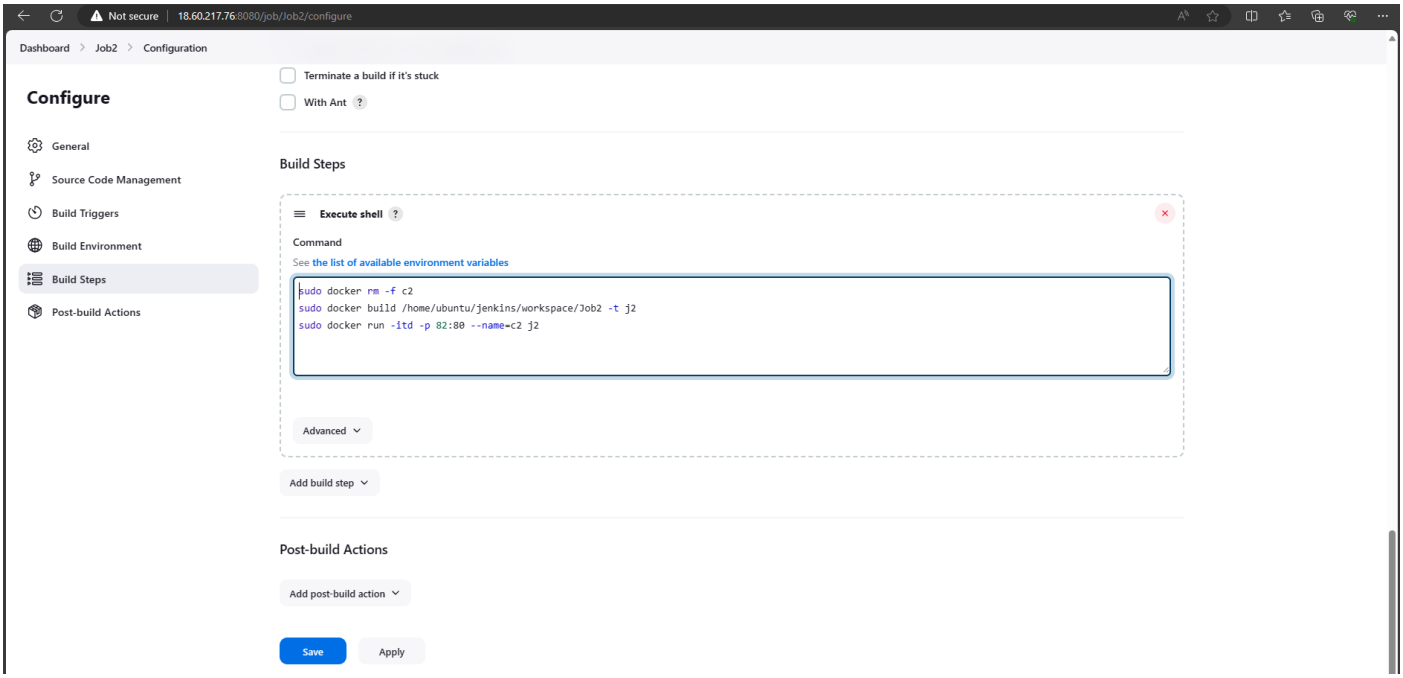
25. Now next we will create Job for Slave1 but this time for master branch. Also we need to edit the image & container names.



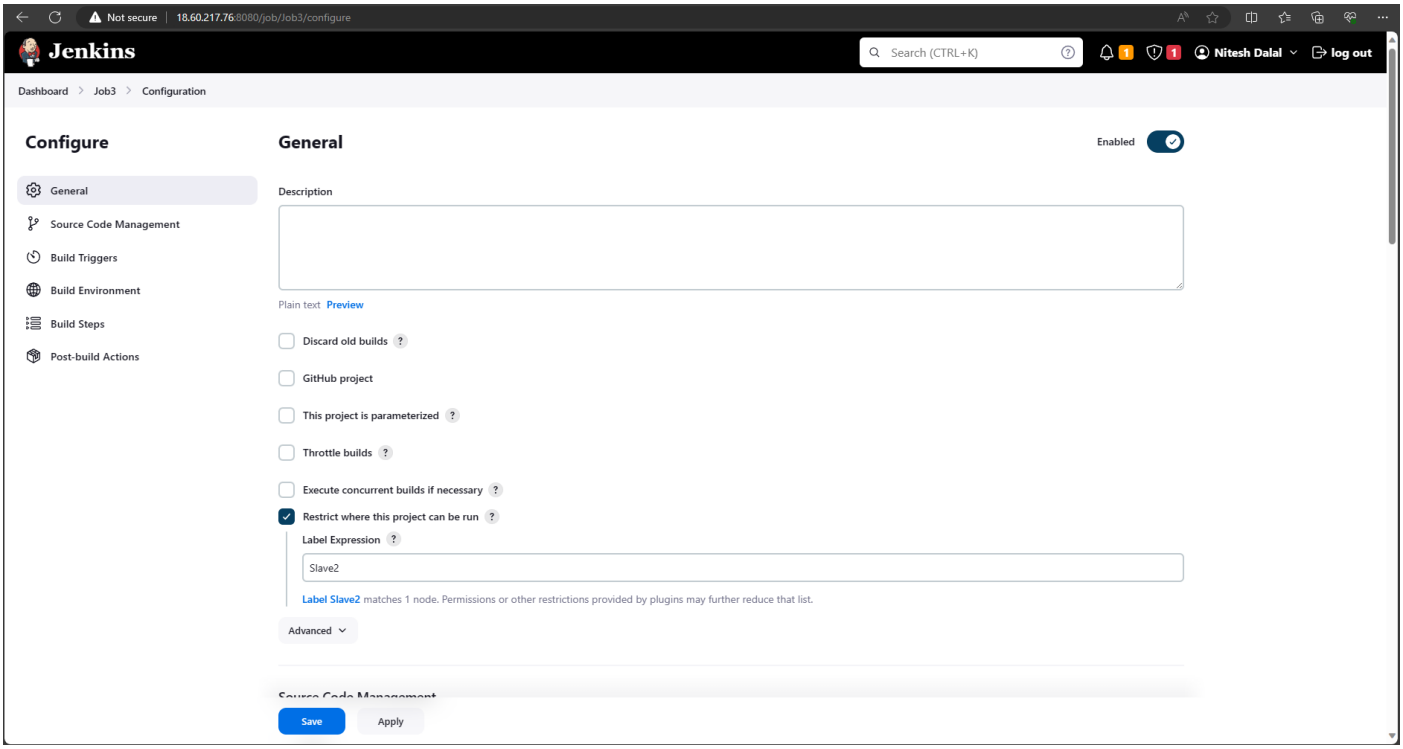


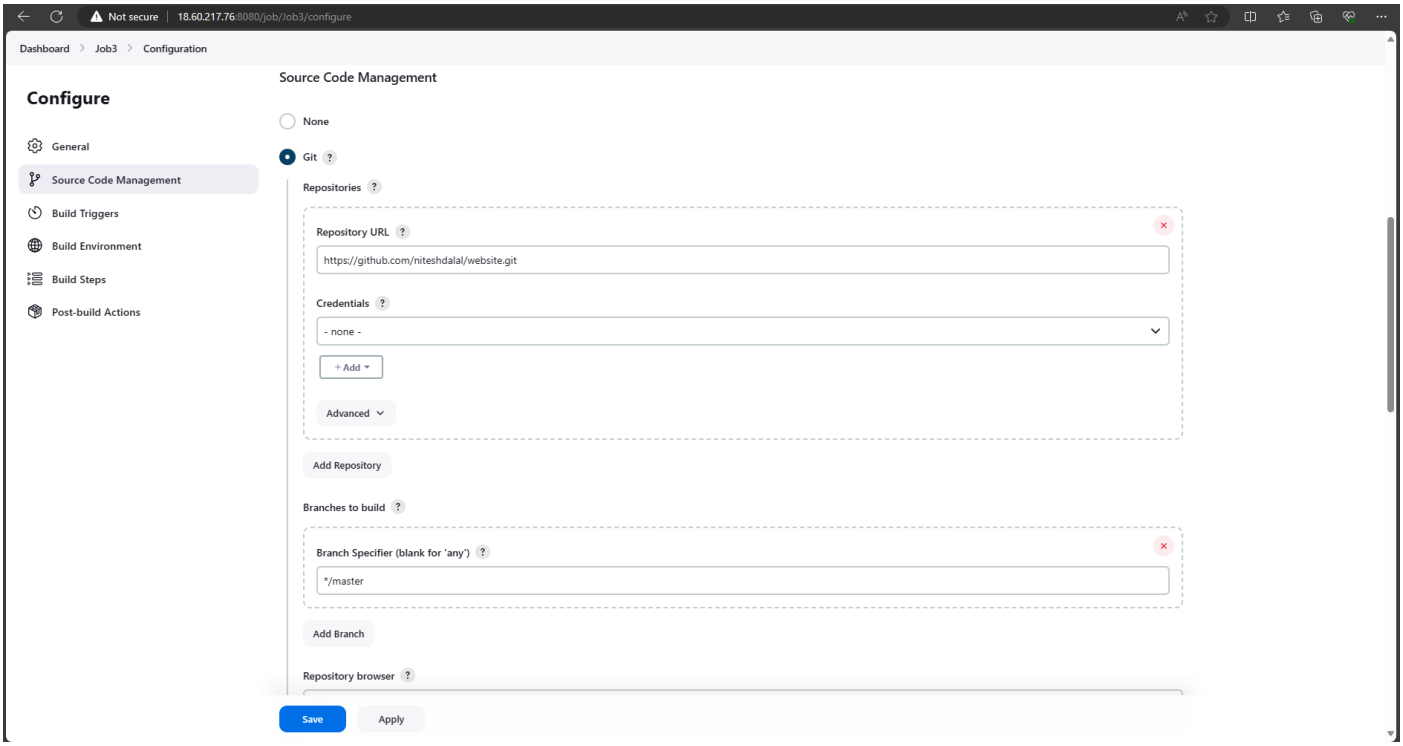


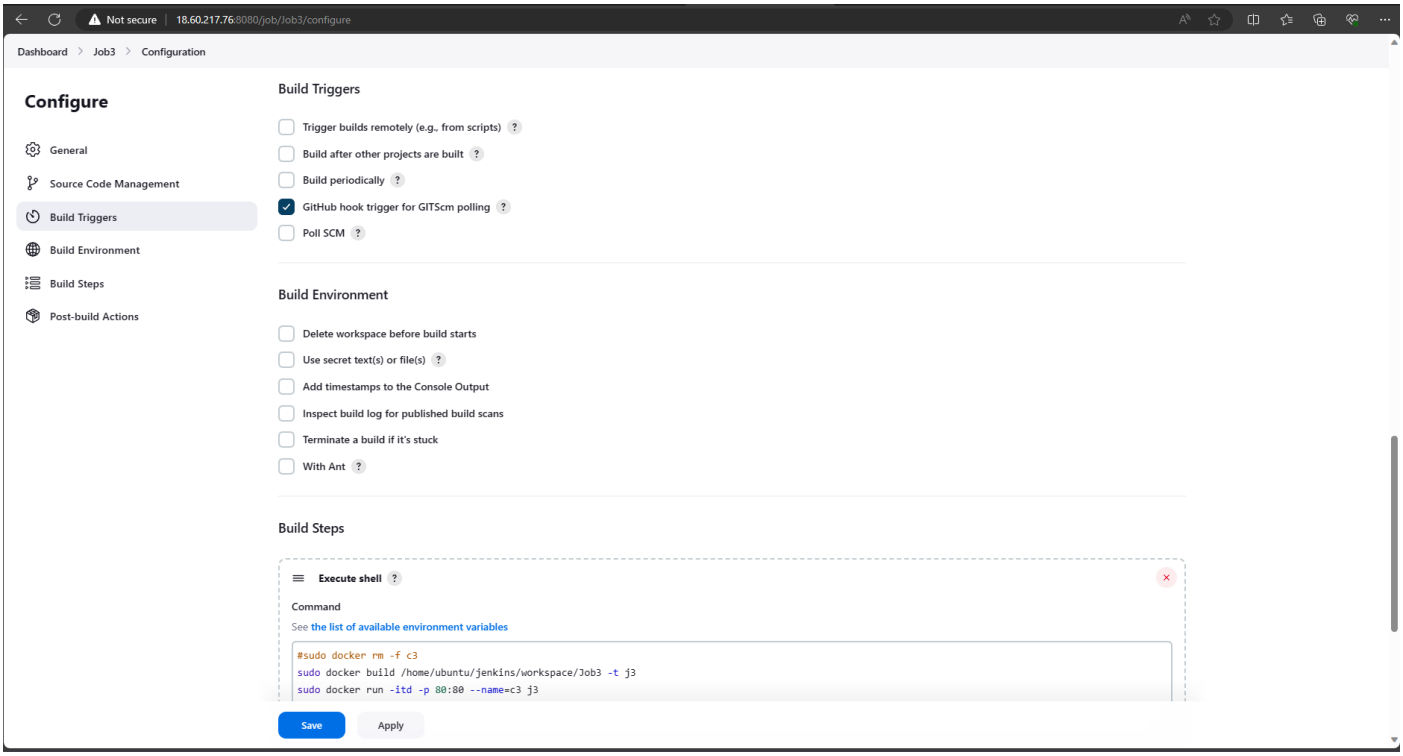
26. After the first successful build we need to re configure the Job2 with execute shell commands as follow.



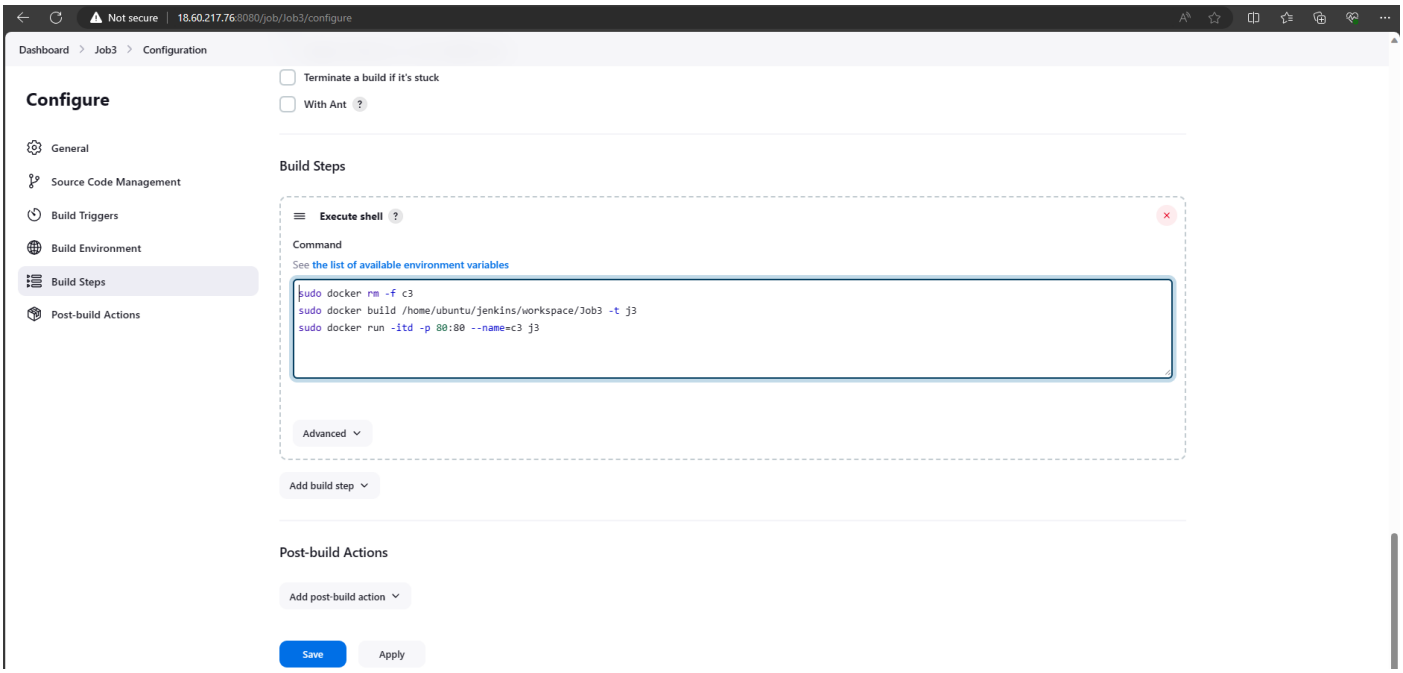
27. Let's create the final Job3 which will execute on production server i.e. Slave2 with master branch restriction.


Not secure 18.60.217.76:8080/job/Job3/configure
Jenkins
Search (CTRL+K)
Nitesh Dalal
log out
Dashboard > Job3 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
General
Enabled
Description
Plain text Preview
Discard old builds
GitHub project
This project is parameterized
Throttle builds
Execute concurrent builds if necessary
Restrict where this project can be run
Label Expression
Slave2
Label Slave2 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.
Advanced
Save
Apply


Not secure 18.60.217.76:8080/job/Job3/configure
Dashboard > Job3 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Source Code Management
None
Git
Repositories
Repository URL
https://github.com/niteshdalal/website.git
Credentials
- none -
+ Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/master
Add Branch
Repository browser
Save
Apply

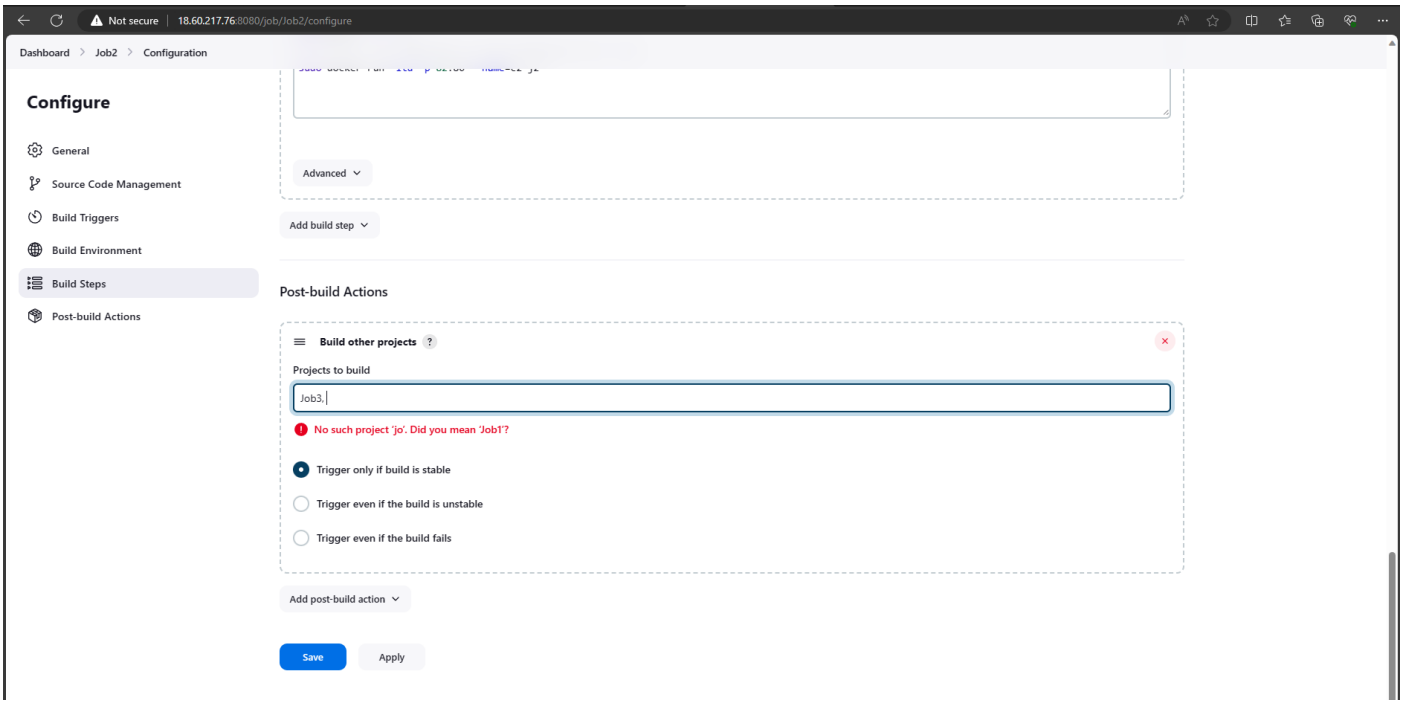


28. Reconfiguring the Job3 after first build.

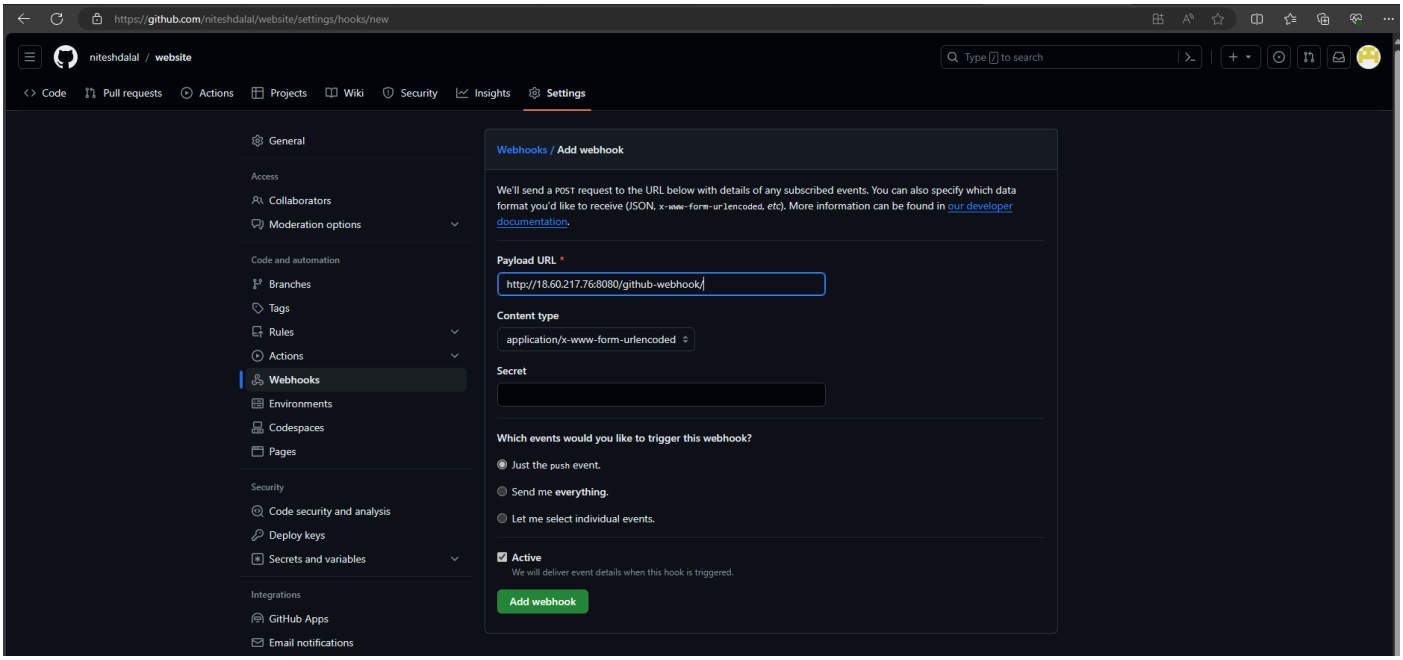


29. See on slave2 machine it is running on port 80 correctly.

30. Now we need to re configure Job3 in a manner so that if Job2 is successful then Job3 should be automatically triggered. So, we need to enable post build actions.



31. Last thing remaining is to add webhook of github for Jenkins jobs.


https://github.com/niteshdalal/website/settings/hooks/new
niteshdalal / website
Type / to search
Code
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Environments
Codespaces
Pages
Security
Code security and analysis
Deploy keys
Secrets and variables
Integrations
GitHub Apps
Email notifications
Webhooks / Add webhook
We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in our developer documentation.
Payload URL *
http://18.60.217.76:8080/github-webhook/
Content type
application/x-www-form-urlencoded
Secret
Which events would you like to trigger this webhook?
Just the push event.
Send me everything.
Let me select individual events.
Active
We will deliver event details when this hook is triggered.
Add webhook

32. That's all.