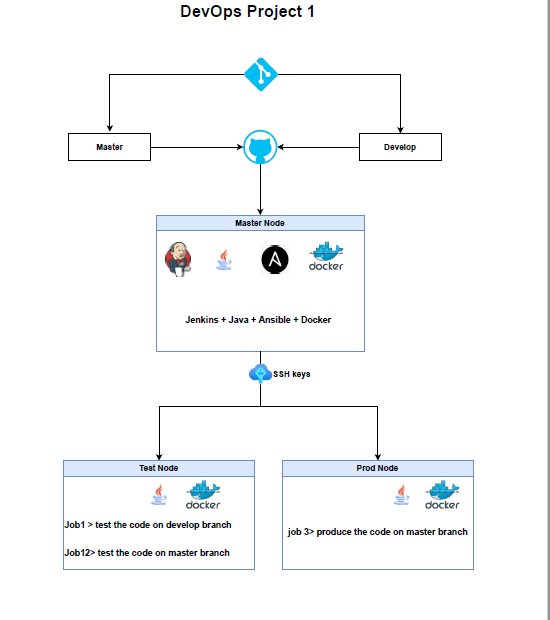
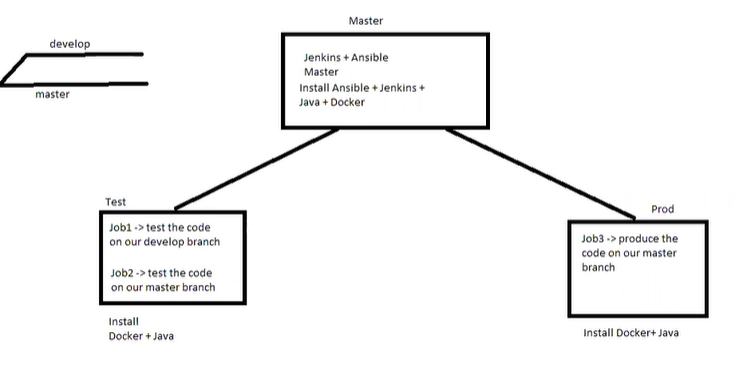
nghtf







Create the three instances

Named

1.master

2.slave1

3.slave2

For connecting master with slaves

We can access three server using multiexecequte mode in aws

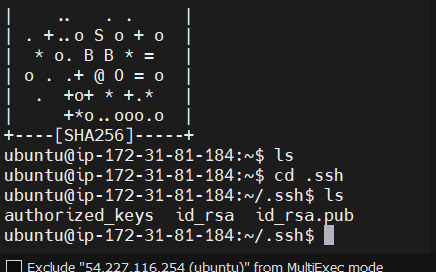
In master

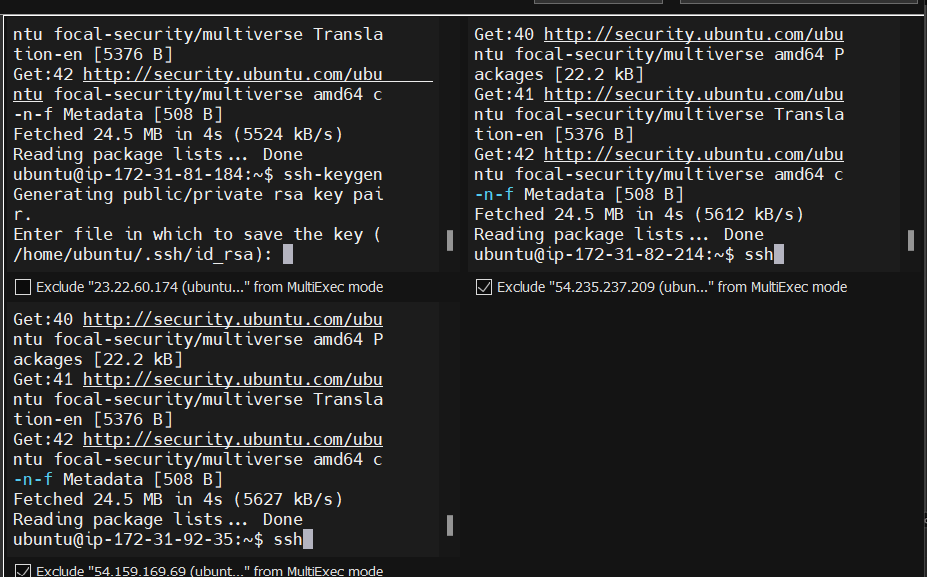
we

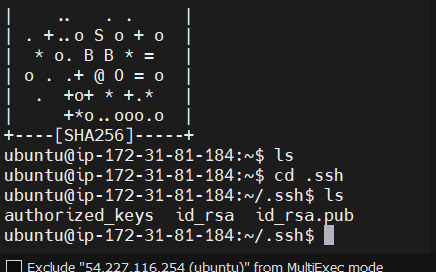
Create the ssh key

$ssh-keygen

3 files will be created







Copy the key from the id\_rsa file

cat id\_rsa

copy the key go to slave instances

cd .ssh

ls

authorized\_keys

sudo nano authorized\_keys

copy the key value of id\_rsa.pub key to the file

this process is useful to the master with the slave

go the master

install ansible



After installing ansible

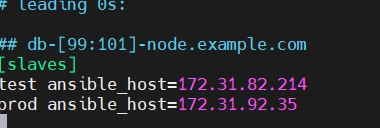
Go to

$ cd /etc/ansible/

In that go to

$ sudo nano hosts

Create the slave private ip address

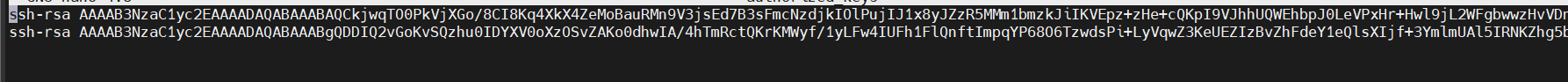


Save and exit

Go to slave

Copy and paste the id\_rsa.pub key into the slave authorized\_keys

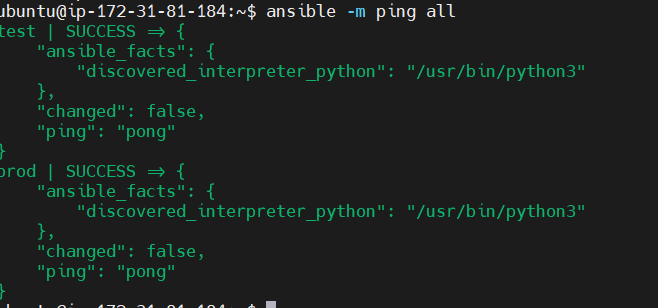
Sudo nano authorized \_keys



Save and exit

Do the same in slave2 aloso

Go to the master check the connectivity



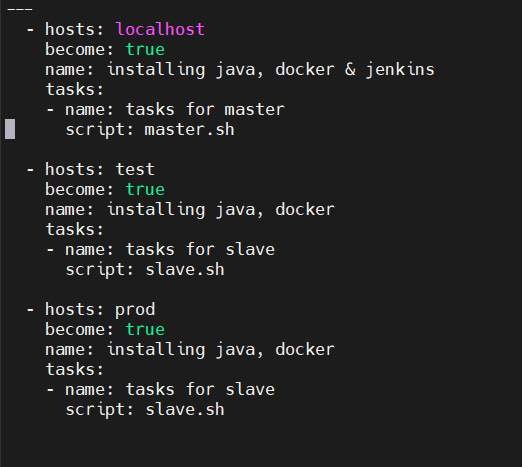
It is connected well

Create an Ansible playbook for installing necessary software in master and slave

Create the play. yaml in master

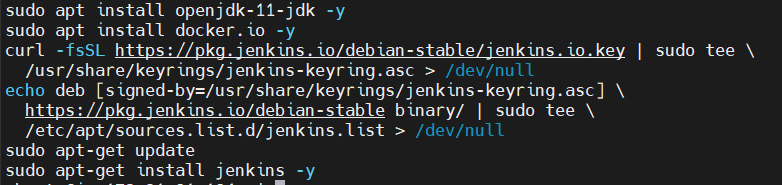
Install java, docker, and Jenkin in master

Install Java and docker in slaves



Create a script file as

sudo nano master. sh to install java, docker and Jenkins



Create slave. sh as a script file to install Java and docker

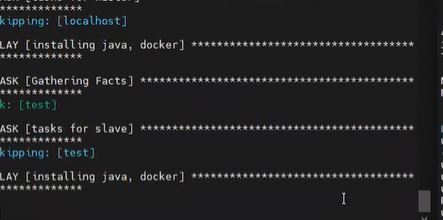
Sudo nano slave. sh



To check the playbook play.yaml for dryrun

ansible-playbook play.yaml –check for dry the file





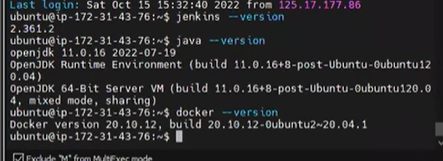
A dry run is successful

To check the syntax error

Ansible-playbook play.yaml --syntax -check



To check the installation of software



Check the process in slave

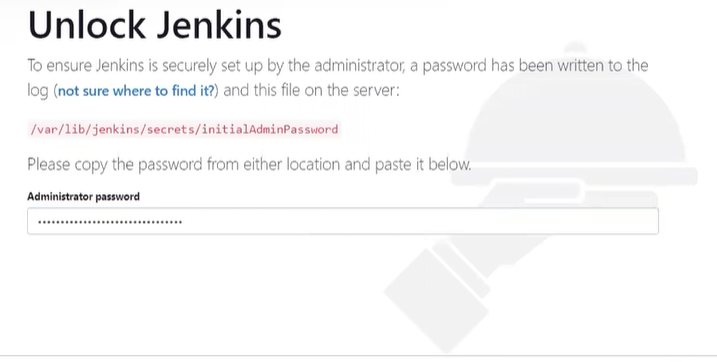
Open the Jenkin dashboard

Copy the password link in to master

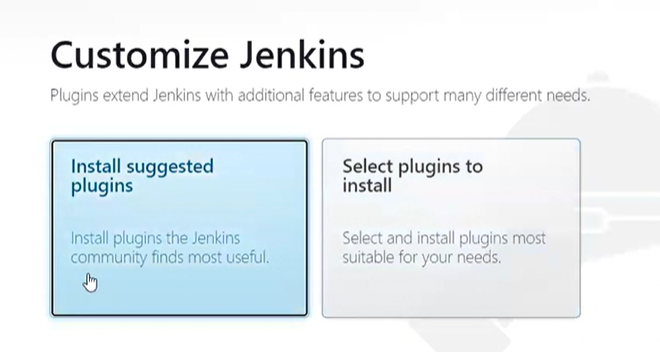
As below

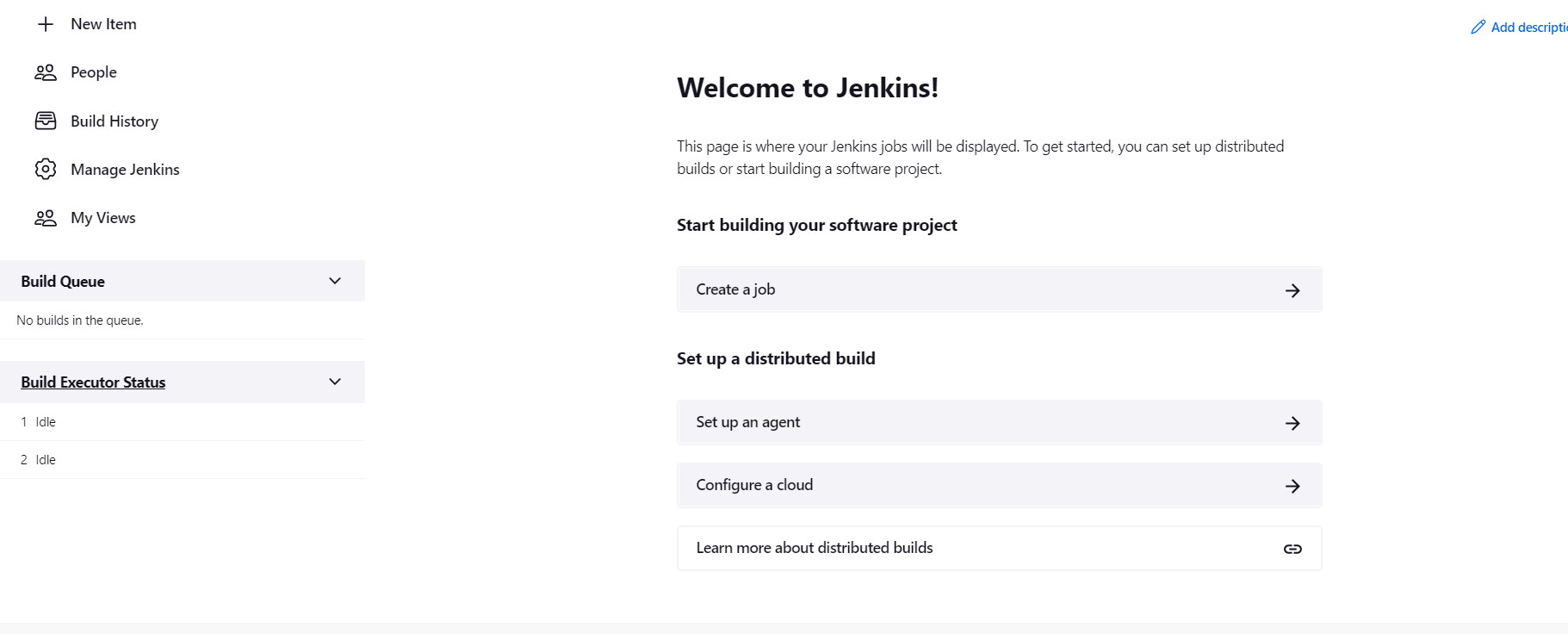


Enter the pass word to the Jenkins as administrator password



Select the install suggested plugin





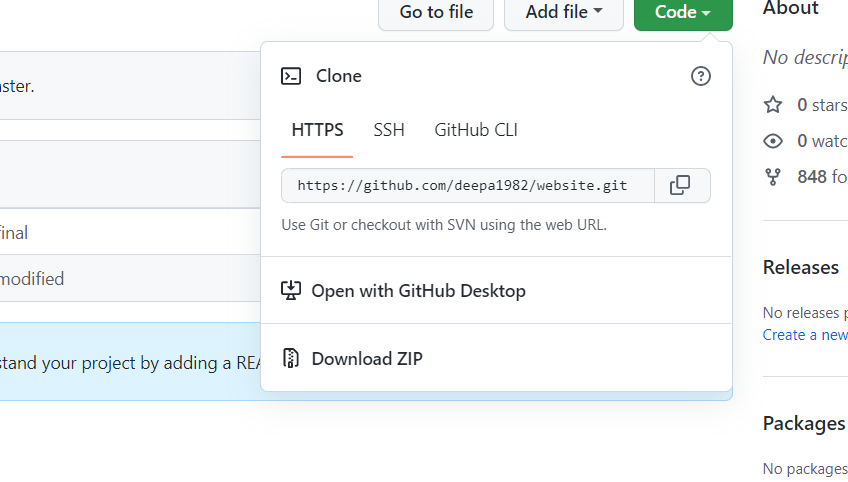
Jenkin dashboard has been created

Click the GitHub link provided in the assignment

It will take to GitHub repository

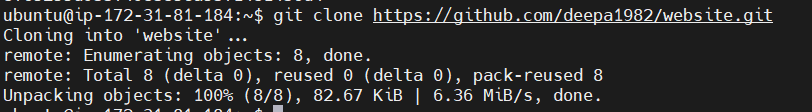
Create a new fork

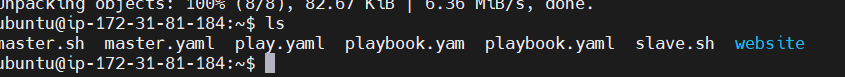
Copy the code link of created fork

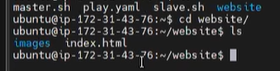


Copy the code link of created fork

The file website was created using the git clone of copied link

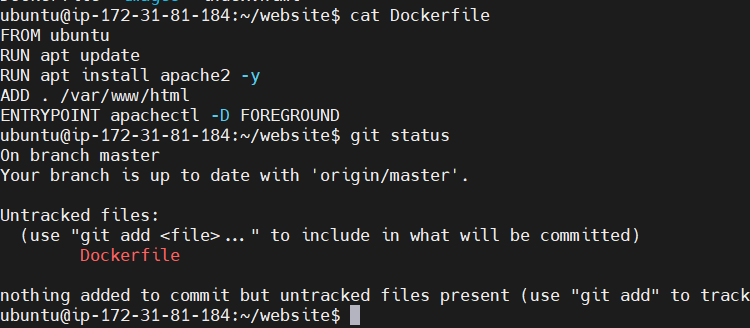




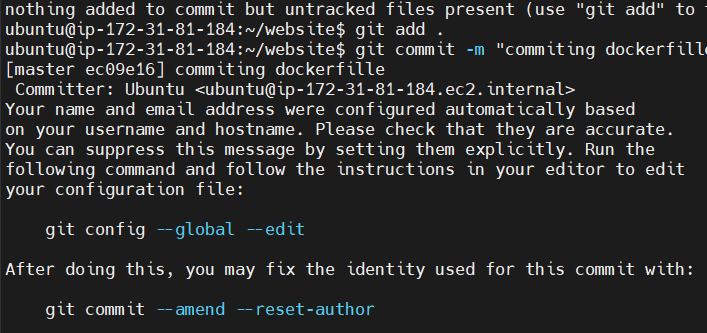


There are two files inside the website directory named images and index.html

Create a docker file as the below content



Add and commit the docker file



git branch

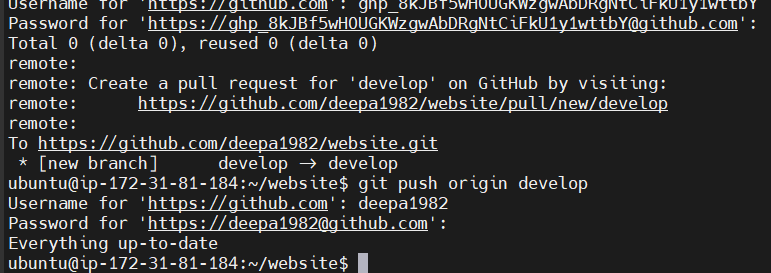
the master branch has been created

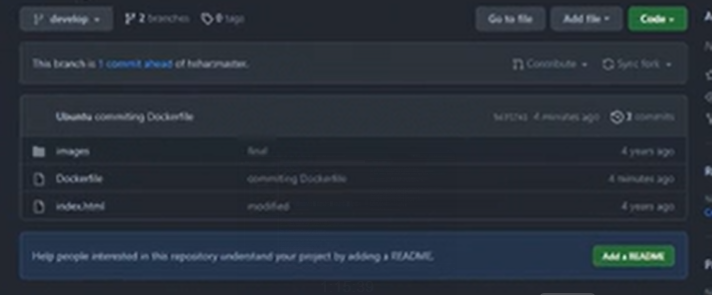
create a develop branch

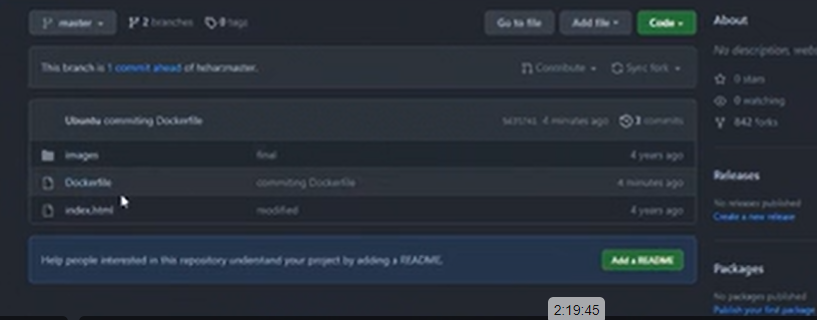
git branch develop

git push origin master

git push origin develop





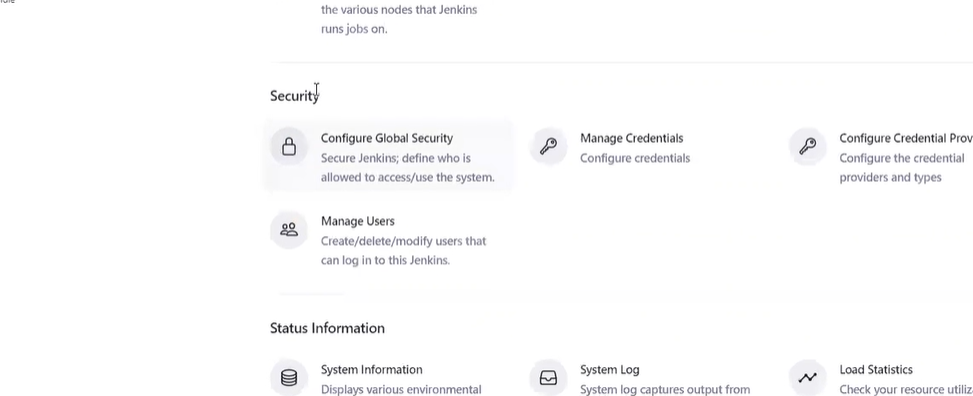


Above files are created in develop branch and master branch and push to the GitHub repository

Now we have to create our nodes in the Jenkins dashboard

Go to Jenkins dashboard

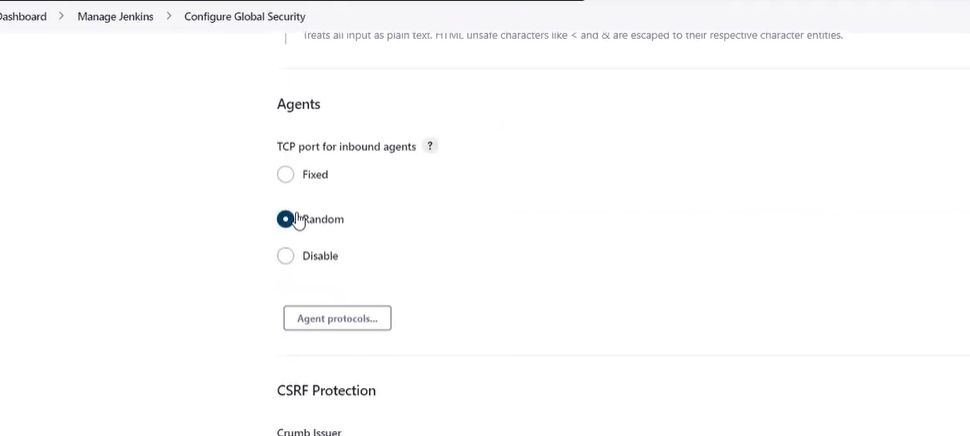
Select the manage, Jenkins



Select the manage Jenkins

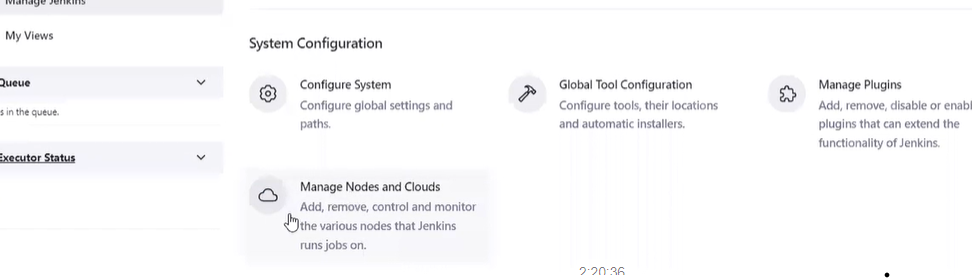
Select in security

Select Configure global security



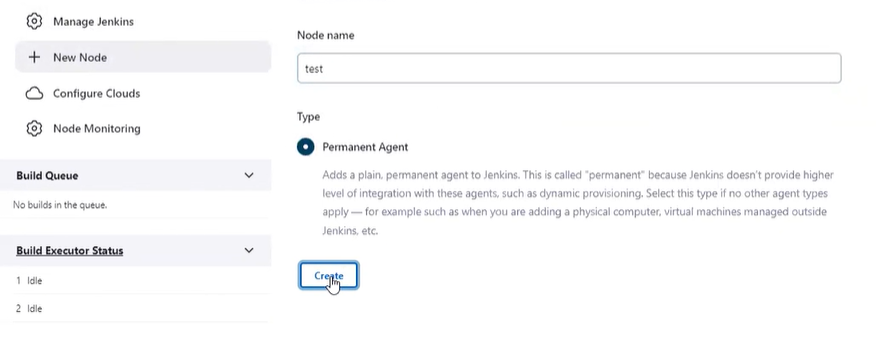
Select the random in the agents

And save it.



Now we have to create our nodes

Select manage nodes and clouds

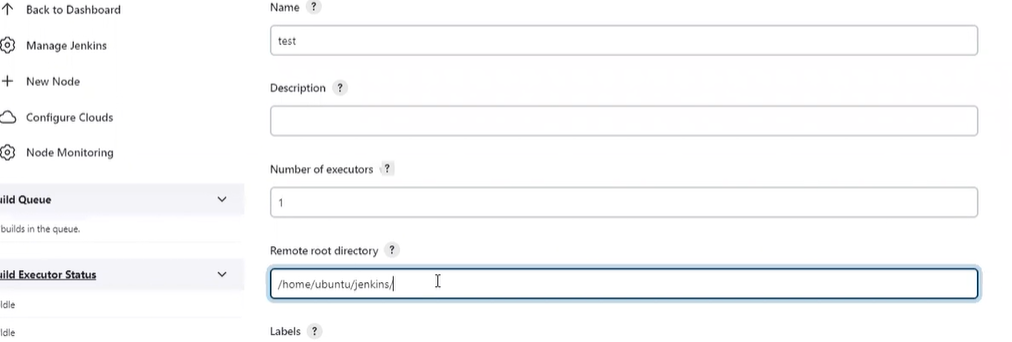


Select the new node

Enter the nodename as test

Select the permanent agent

Click create



Enter the remote directory

To create the workspace for cloned or pulled directories from the GitHub

As /home/ubuntu/Jenkins/



Launch method select launch agents via SSH



Enter the slave1 private ip address in the host

Select the credential as Jenkins

Enter the credential details.



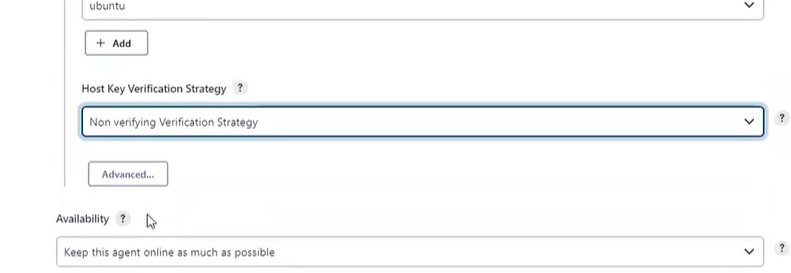
Select the kind as SSH username with private key



Select the username as ubuntu



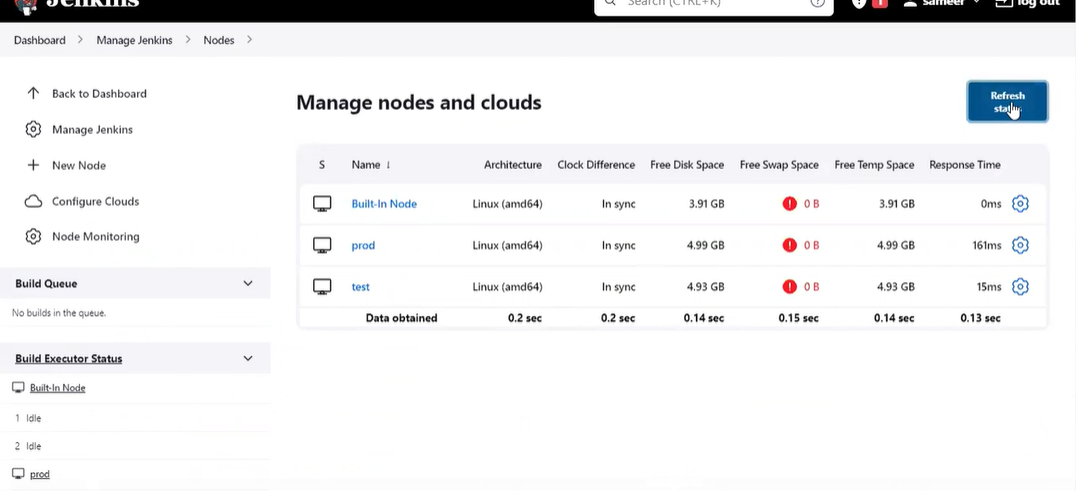
Copy and paste the pem key file of that slave1 instance



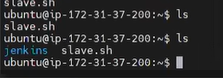
Select host key verification strategy

Non verifying verification strategy

Then



The test node is created



Jenkin file has been created in the slave1



There will be 2 files inside the Jenkin directory

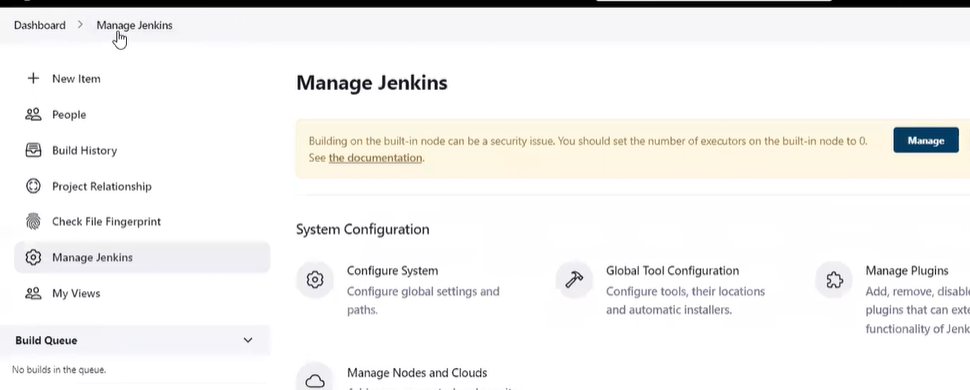
Remoting and remoting.jar

Create another node for the prod

Follow the same procedure as above.



Another node for prod has been created.



We have to create the job

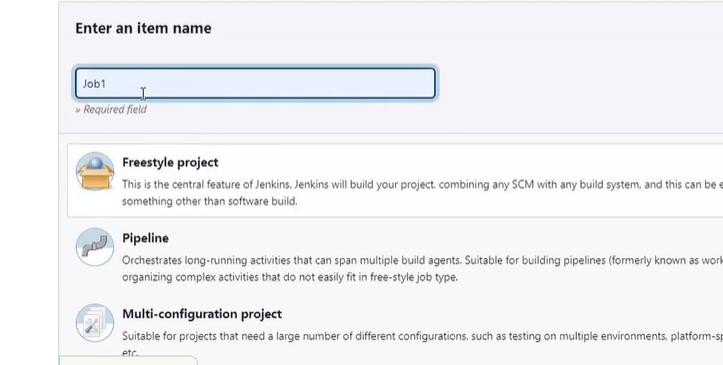
We have to create 3 jobs

2 jobs for test server

1 job for prod server

Select manage Jenkins

Select new item.

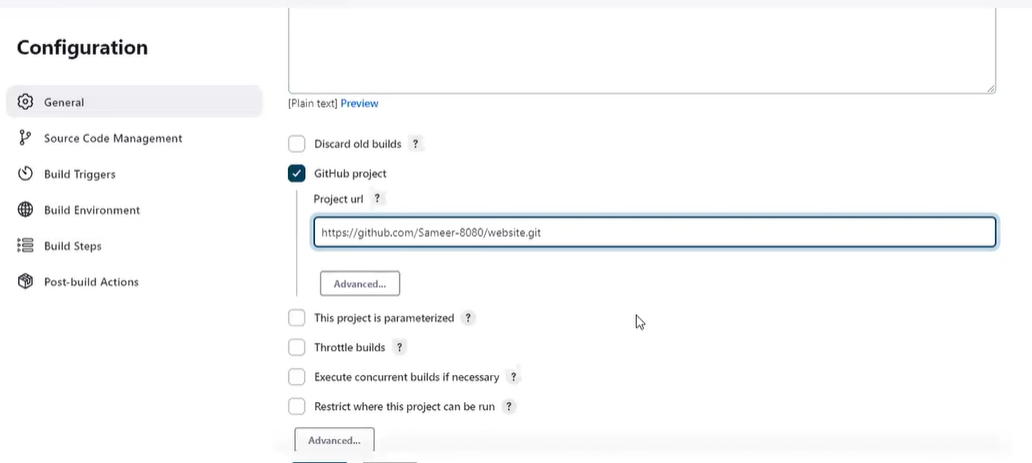


Enter the name of the job

And select a freestyle project

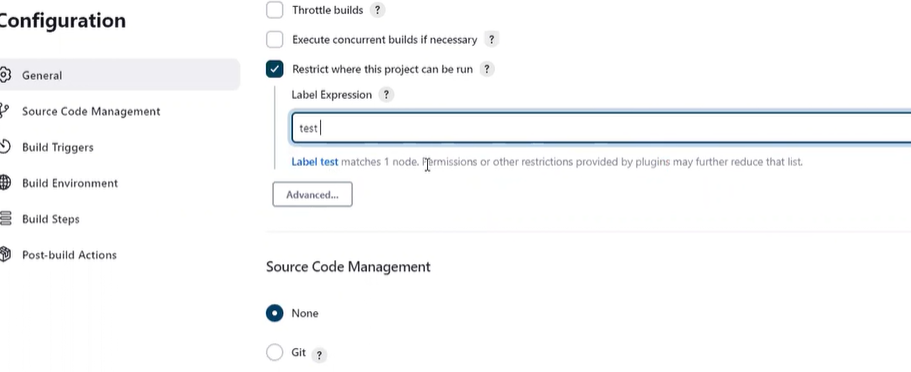
And click ok

And configuration page of the job will display



Select the GitHub project

Give the link of the source code of GitHub repository



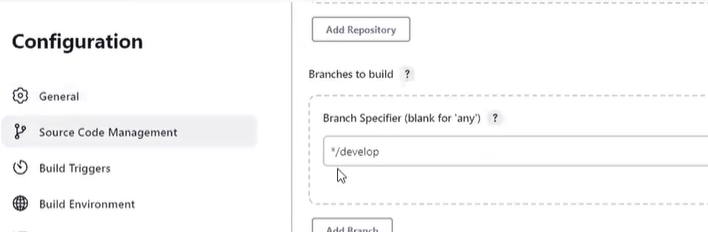
Select the restriction where this project can be run

And give the name of the node while we created during the nodes

Give the name as test



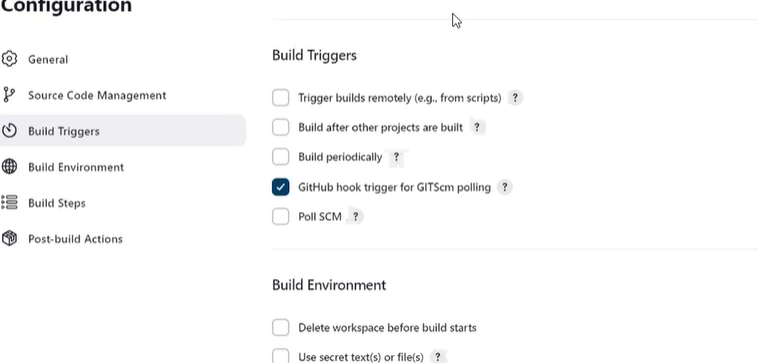
Select the source code as git and give the and enter the URL of GitHub repository



Enter the branch to build as

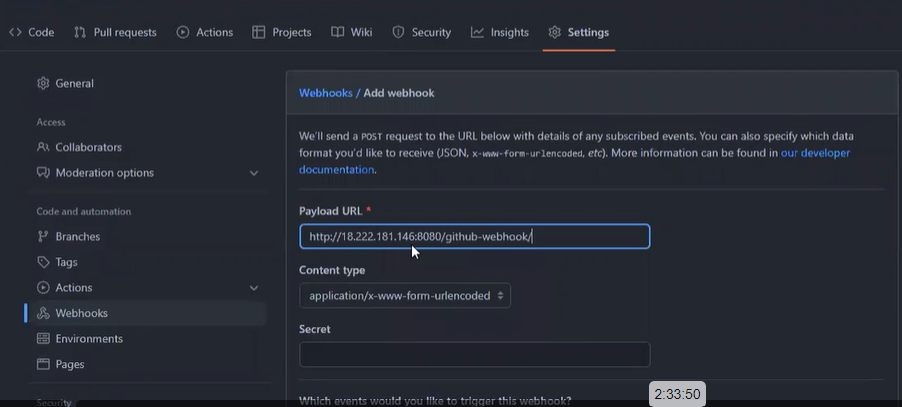
Develop

Since our first job has to be pushed from develop branch to test server



Select the GitHub hook trigger for GITcm polling

This is used to give permission to copy the content of the GitHub repository to Jenkins



Go to GitHub repository

Select setting

Select webhook

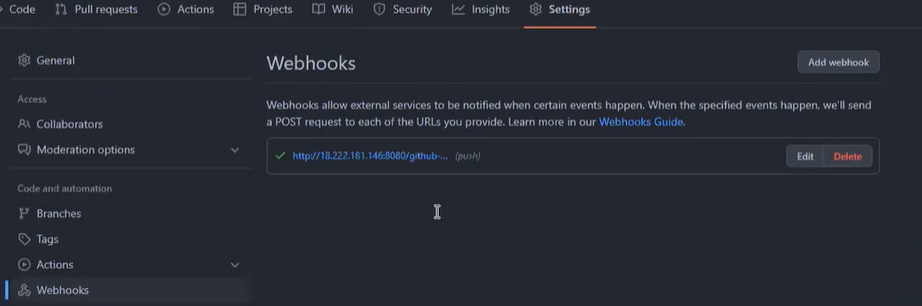
In payload URL

Give the URL of Jenkins the job created

And add /GitHub-webhook/ with URL

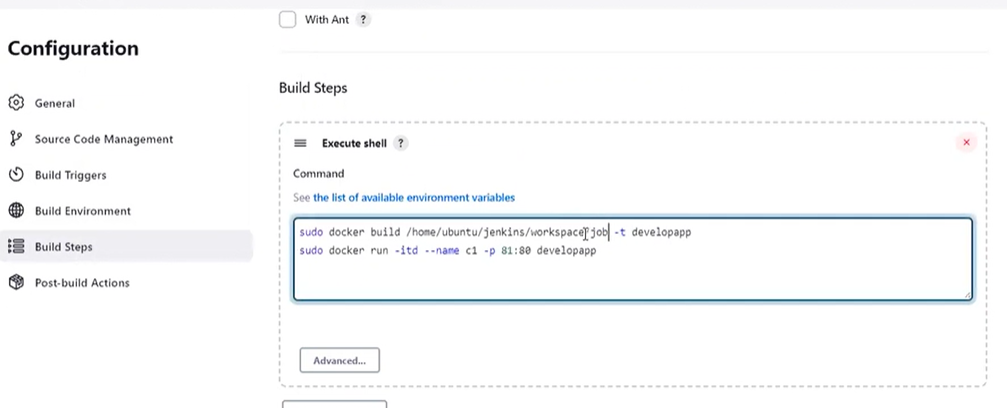
Select the just push event

Select add webhook

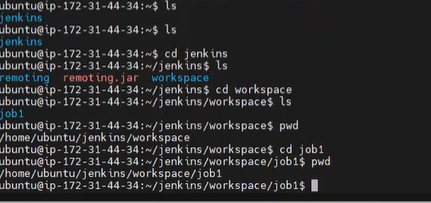


Webhook has been created

Got to jenkin configuration for build the first job



The above execute shell command is useful to create the container of docker file,



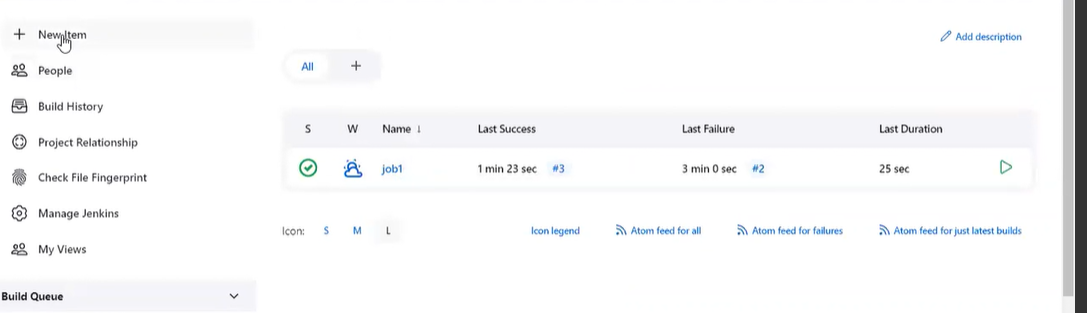
Job1 has been created in our workspace dir

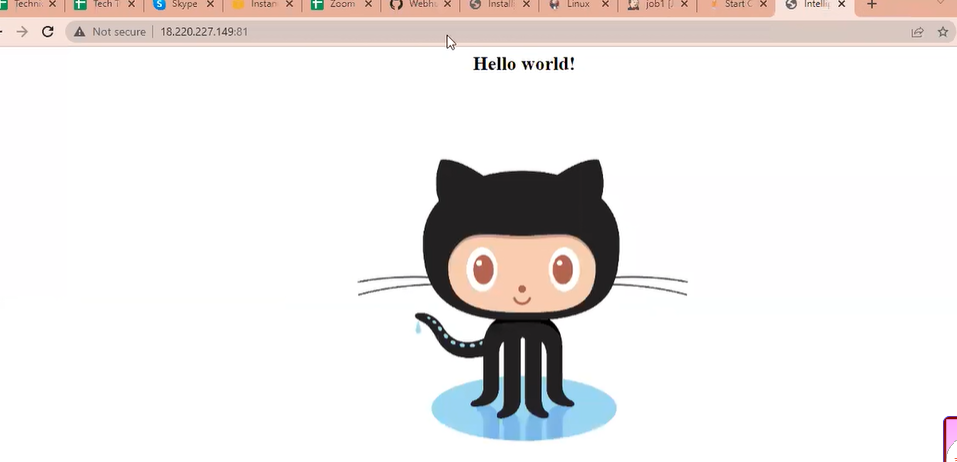
Apply and save it

Click build now

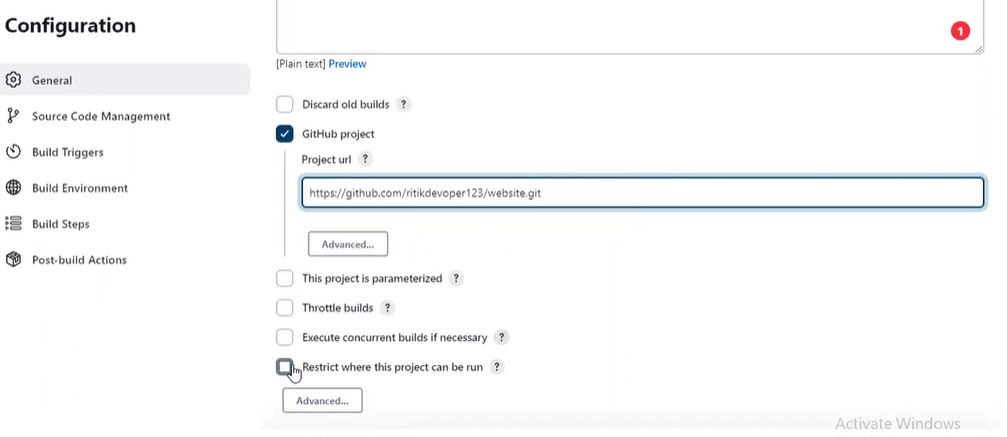
Our job1 has been created.

Copy the public IP address of the test instance.





Copy the public IP address of the test instance. with the port number

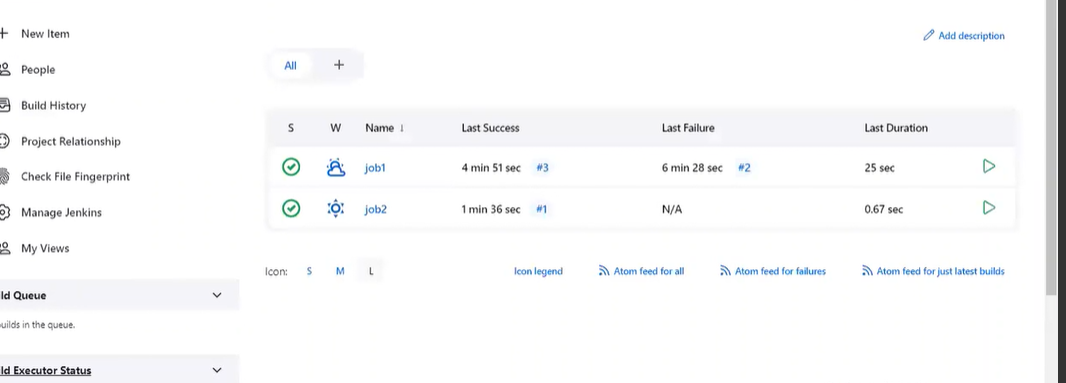


For the job2 goto configuration

Give the GitHub fork link

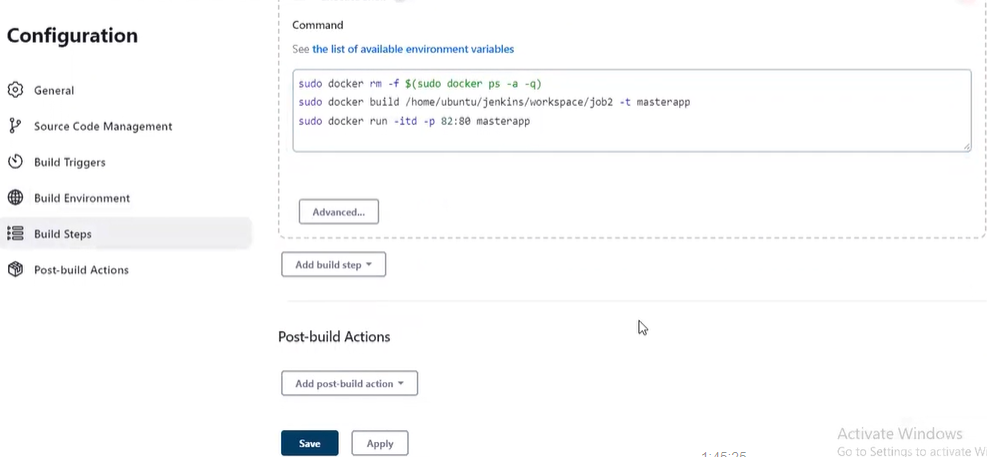
Do the process for job2

Build now



Job2 has been created,

Create the container for job2

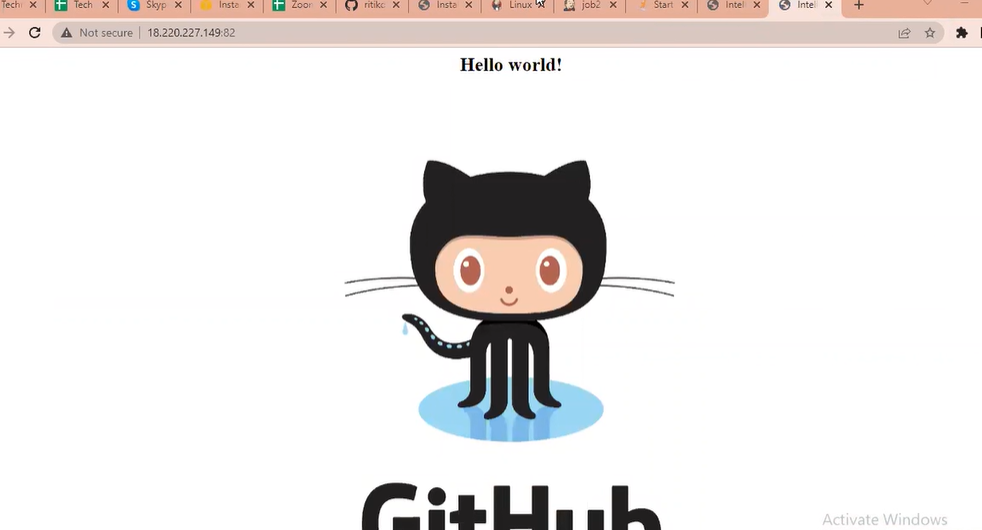


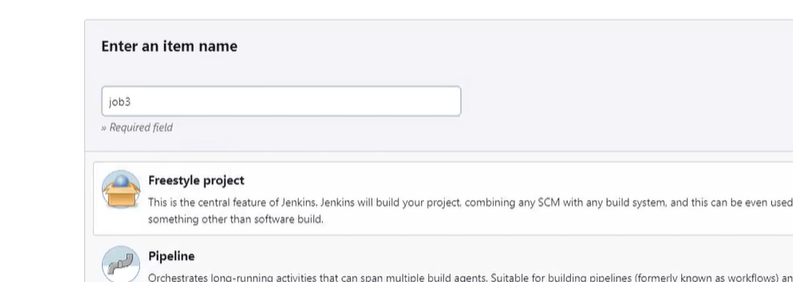
Enter the above command in the executing shell to create the container for the docker file

Apply and save it

Copy the test server’s Public IP address and paste it with the port number 82

Our job2 is created.





For the job3 select a new item,

Select freestyle project job3

Follow the step as job2

In the label expression enter it as a prod

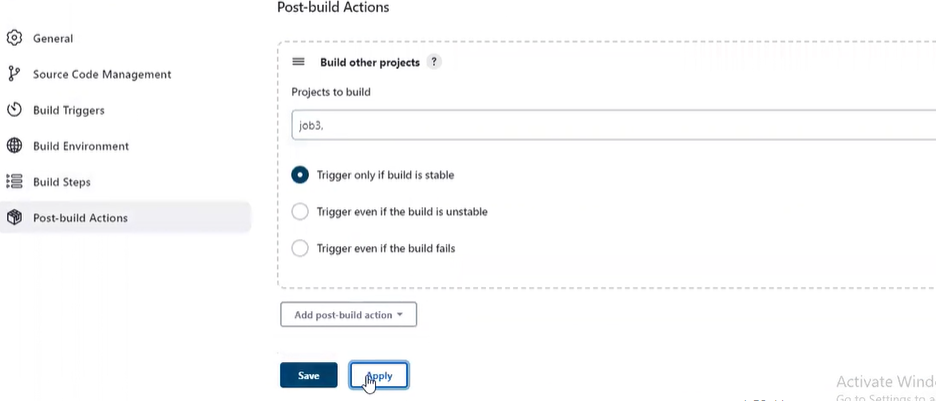
It is going to commit in the prod server



Create the execute shell to create the container for prod.



Give the above command to create the container



Select the job2 to trigger

Select post-build actions.

Give a job3 apply and save it

The final tested code from the master branch is deployed in the prod server.

Three jobs have been created created through Jenkins pipeline