Capstone Projects

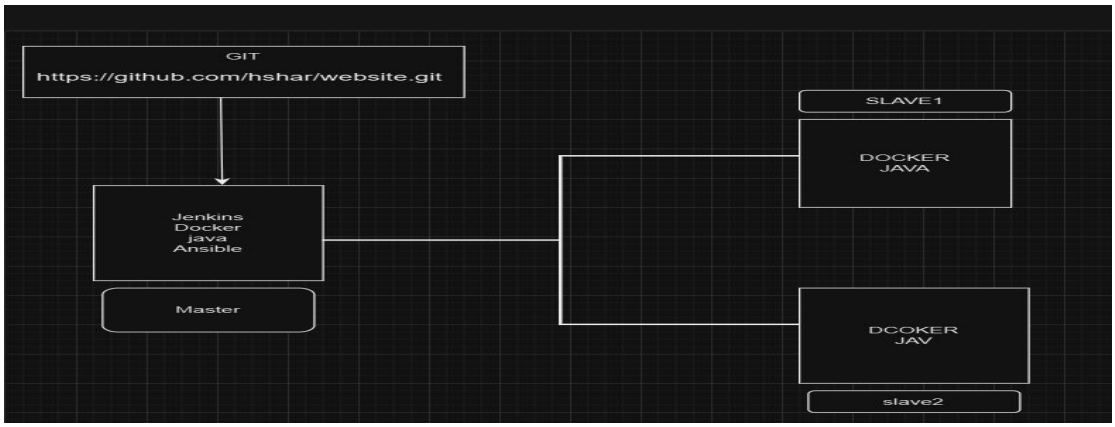
You have been Hired Sr. DevOps Engineer in Abode Software. They want to implement DevOpsLifecycle in their company. You have been asked to implement this lifecycle as fast as possible. Abode Software is a product-based company, their product is available on this GitHub link.
https://github.com/hshar/website.git
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.Code Build should automatically be triggered once commit is made to master branch or develop branch.
If commit is made to master branch, test and push to prod
If 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 Git-Hub. 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:
Job1 : build
Job2: test
Job3 : prod

Upon reading we found below things
We will create cluster of three instances, one master and two slaves, according to line 1- we need to ansible to configure tools, according to line 2 we will be using GIT , according to line 3 and 5-we need Jenkins for automation and line 4 tells us to use docker.

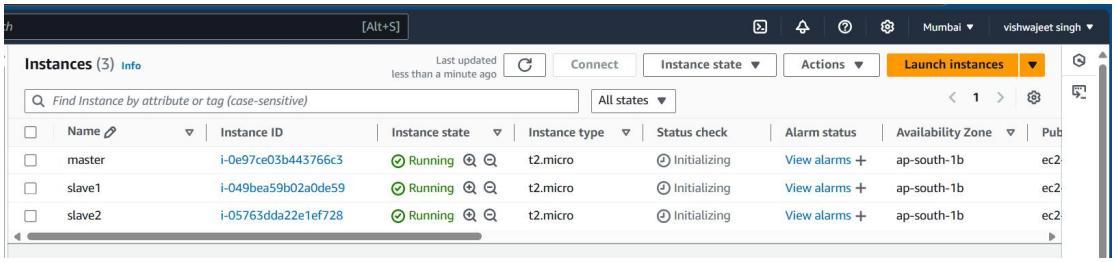
We will be using following tools

1. Ansible
2. Docker
3. GIT
4. Jenkins

Master node -----> install Ansible,docker, Jenkins,java
Slave node ----> install docker and java



Step1: lets launch ec2 instances. We will be launching 3 instances, one will act as master and other two as slaves

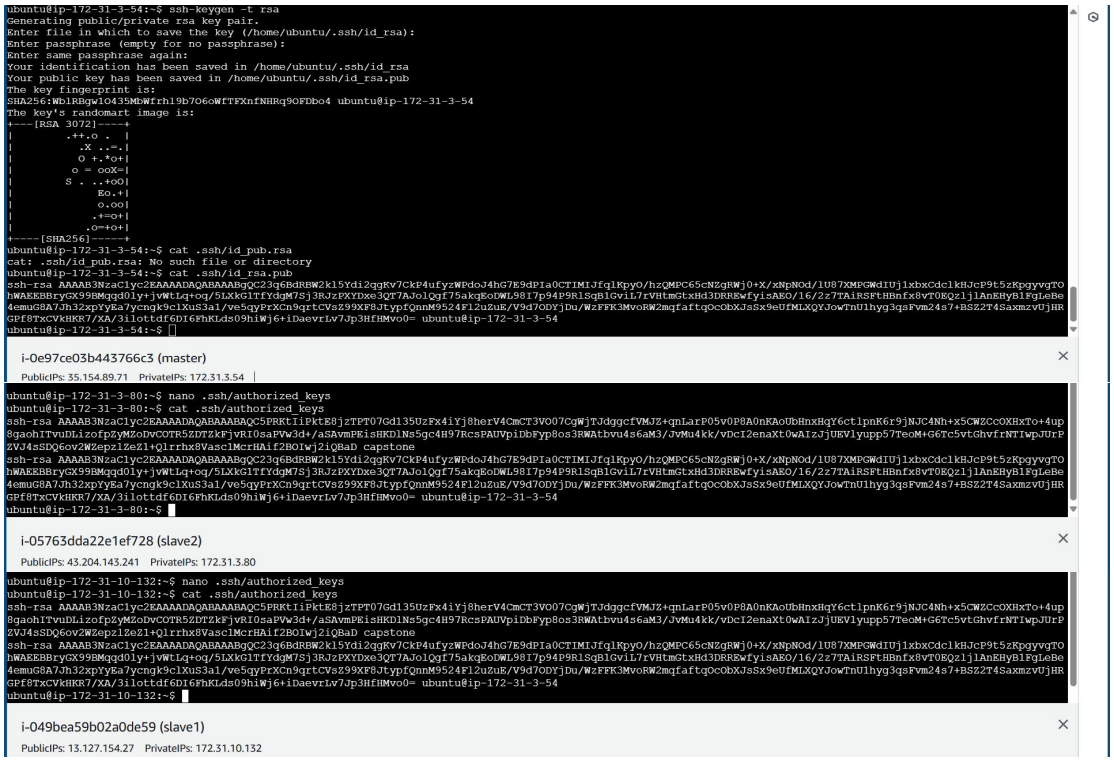


Step2: connect with master instance and install the ansible, we will use ansible to install required tools on all instances

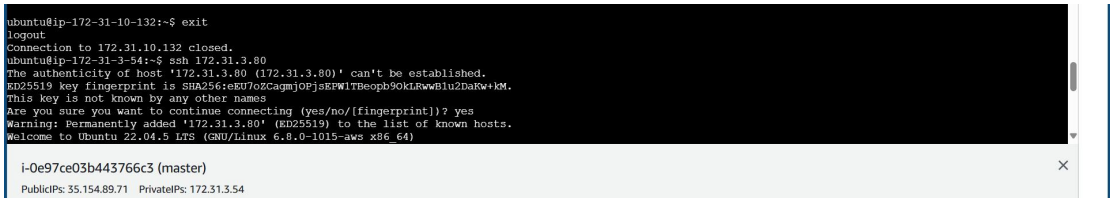
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible

Run above command one by one and ansible will be installed.

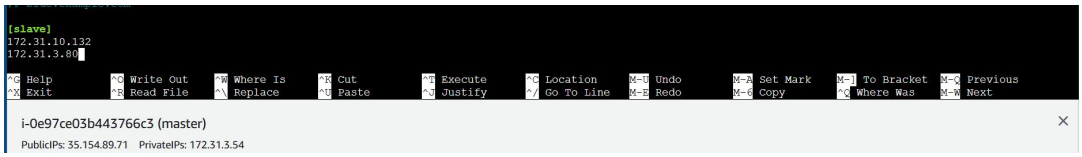
Next we will setup ssh password-less connection between slave and master, also we will add the private ip in the hosts file of ansible directory.
- use command ssh-keygen -t rsa to generate public key, this public key needs to be pasted in the authorized_keys file.



Check if connection has been established or not



Now we will add the private ip in hosts file, run command sudo nano /etc/ansible/hosts



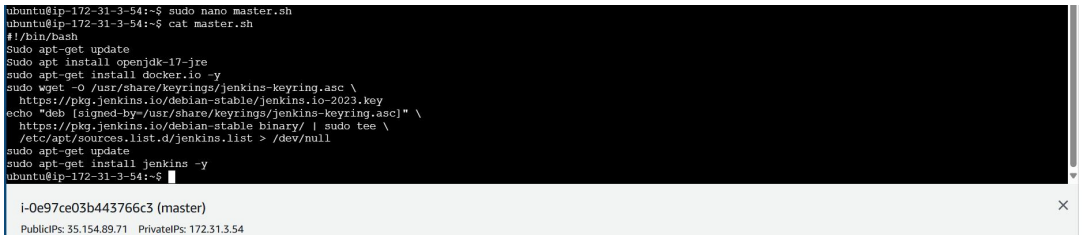
now we will create yaml file to install the tools, which is

On master - Java ,Docker, Jenkins
On slave- Java, Docker

We will do this with the help of script, so lets create two script as per above requirement

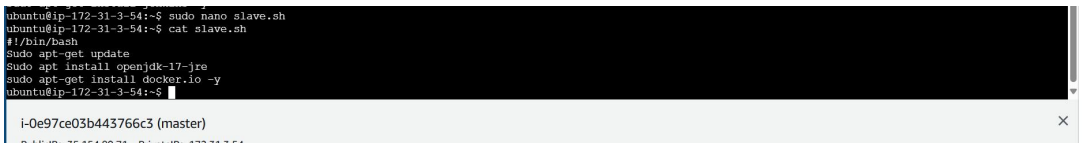
Master.sh(check official documenttion for below commands)

!/bin/bash
Sudo apt update
sudo apt install fontconfig openjdk-17-jre -y
sudo apt install docker.io -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
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 update
sudo apt install jenkins -y



Slave.sh

#!/bin/bash
sudo apt update
sudo apt install fontconfig openjdk-17-jre -y
sudo apt-get install docker.io -y



Lets write the playbook which will be in yaml format

```
---
- name: Installation of Java, Docker, and Jenkins
  hosts: localhost
  become: yes
  tasks:
    - name: Run the master.sh script
      script: master.sh

- name: Installation of Java and Docker
  hosts: slave
  become: yes
  tasks:
    - name: Run the slave.sh script
      script: slave.sh
```

step3: lets run playbook

Run the command ansible-playbook play1.yml

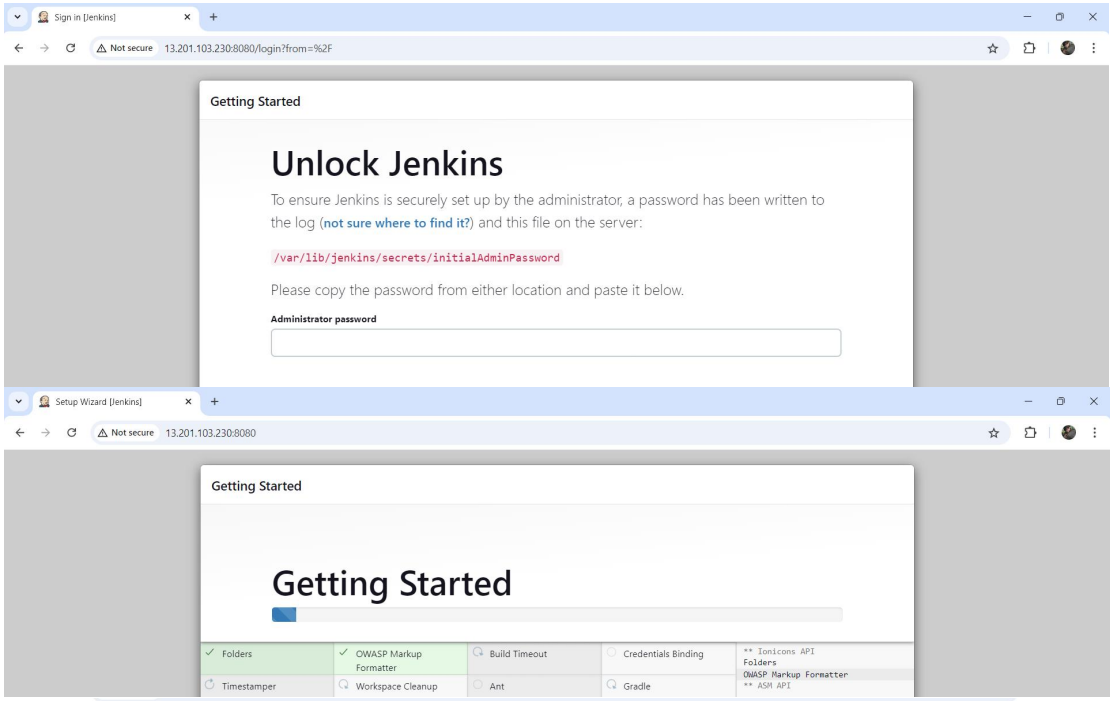
Check if all tools has been installed or not

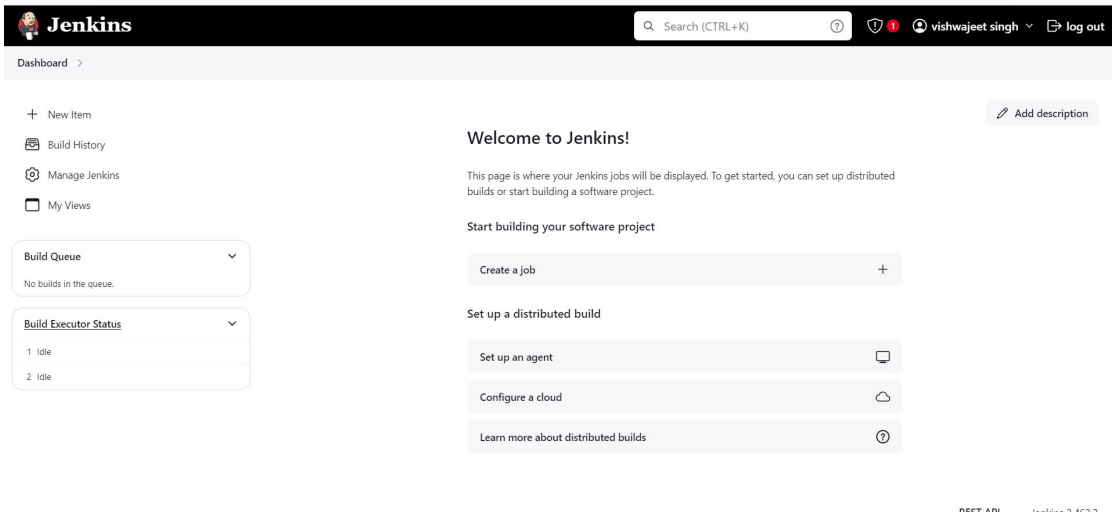
Run command on master
Java --version
Docker --version
Jenkins --version

Required tools have been installed. Now lets do the assigned task, we will establish Jenkins and clone the repo to execute the task. So lets configure the Jenkins first

Step 4: configure jenkins

Copy public ip of master and add :8080 after it to get below page on browser
Now lets do all the setup





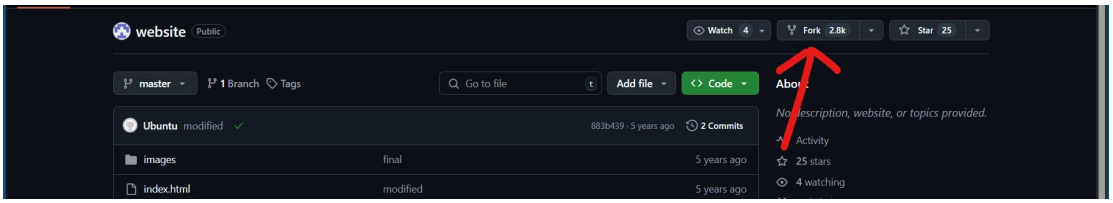
Step 5: now we will add the slave nodes on jenkins

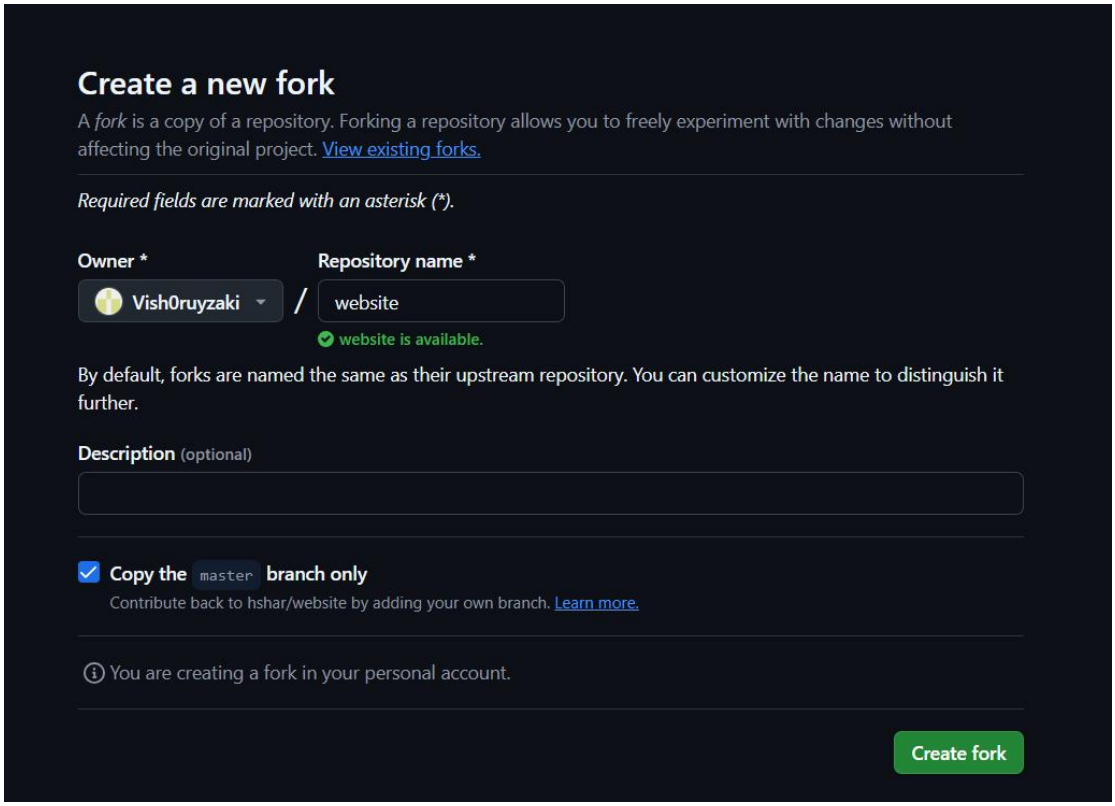
We have added the both nodes. Now we clone the repo which is given in this assigment

Step6: clone the repo

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

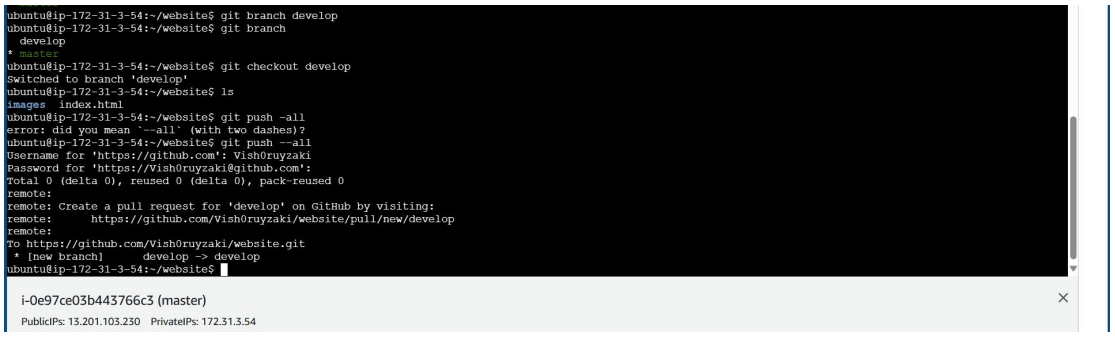
Now fork the repo in your repo



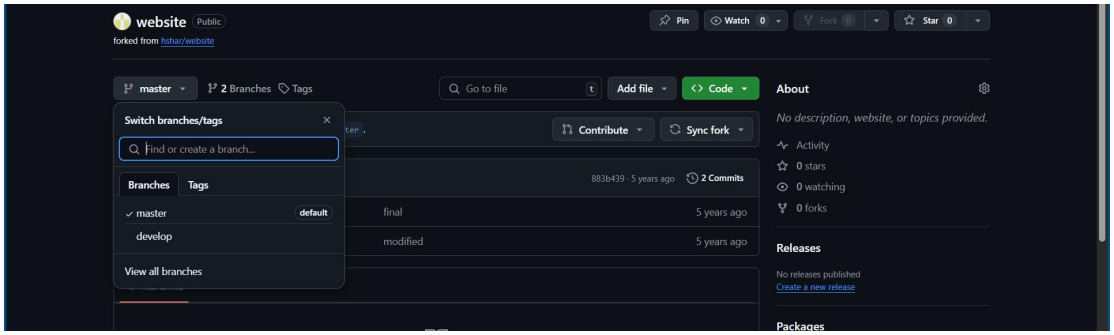


Now we will run command git clone “link” to clone the repo

Step7: lets create develop branch and then push it to github

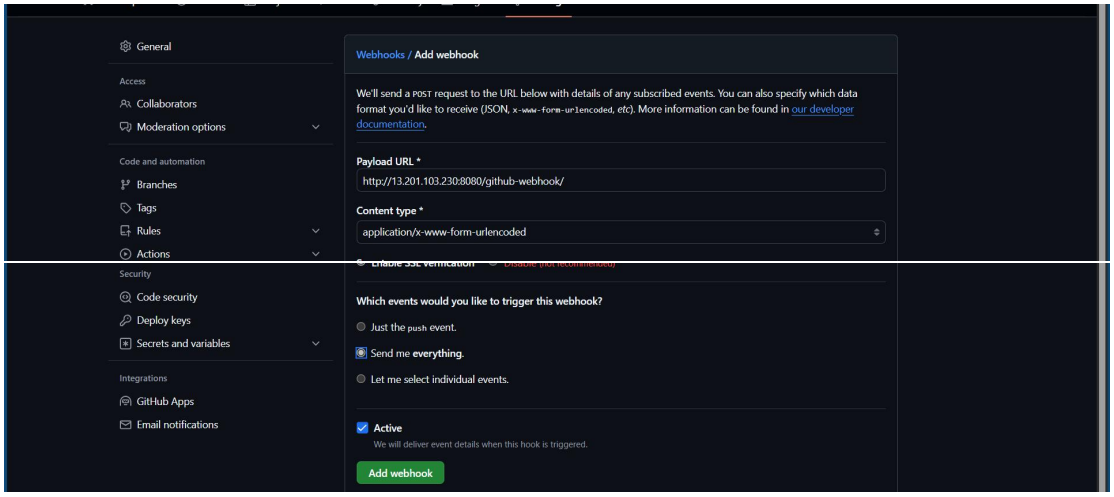


Branch is created and its pushed to github, lets check

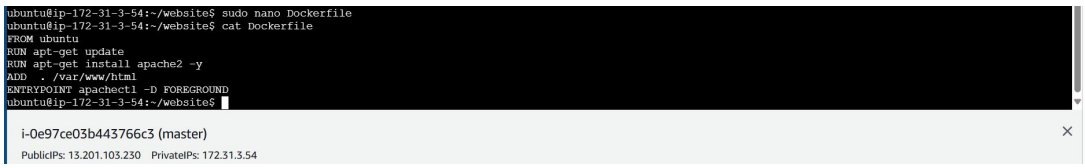


The branches has been pushed succesfully

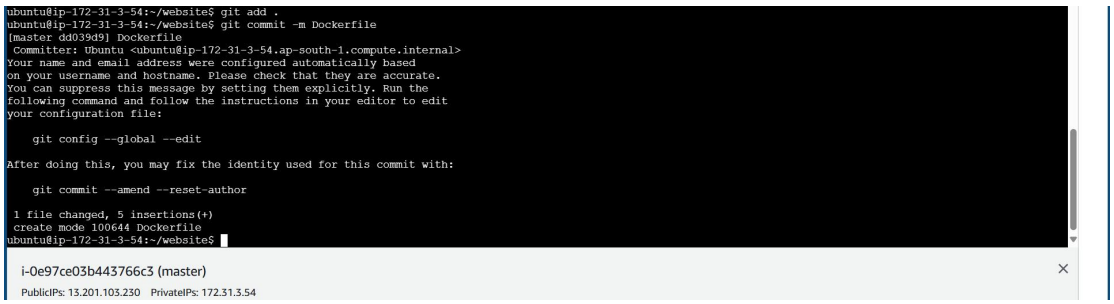
Step8: lets set webhook with our repo



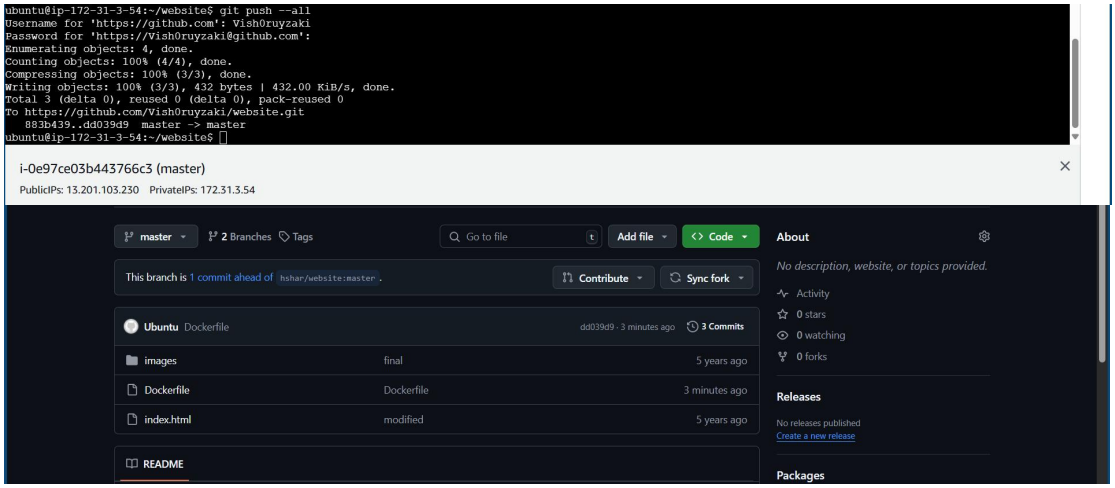
Step 9: we will create dockerfile and create image out of it. This image will be pushed to repo



Lets commit this change



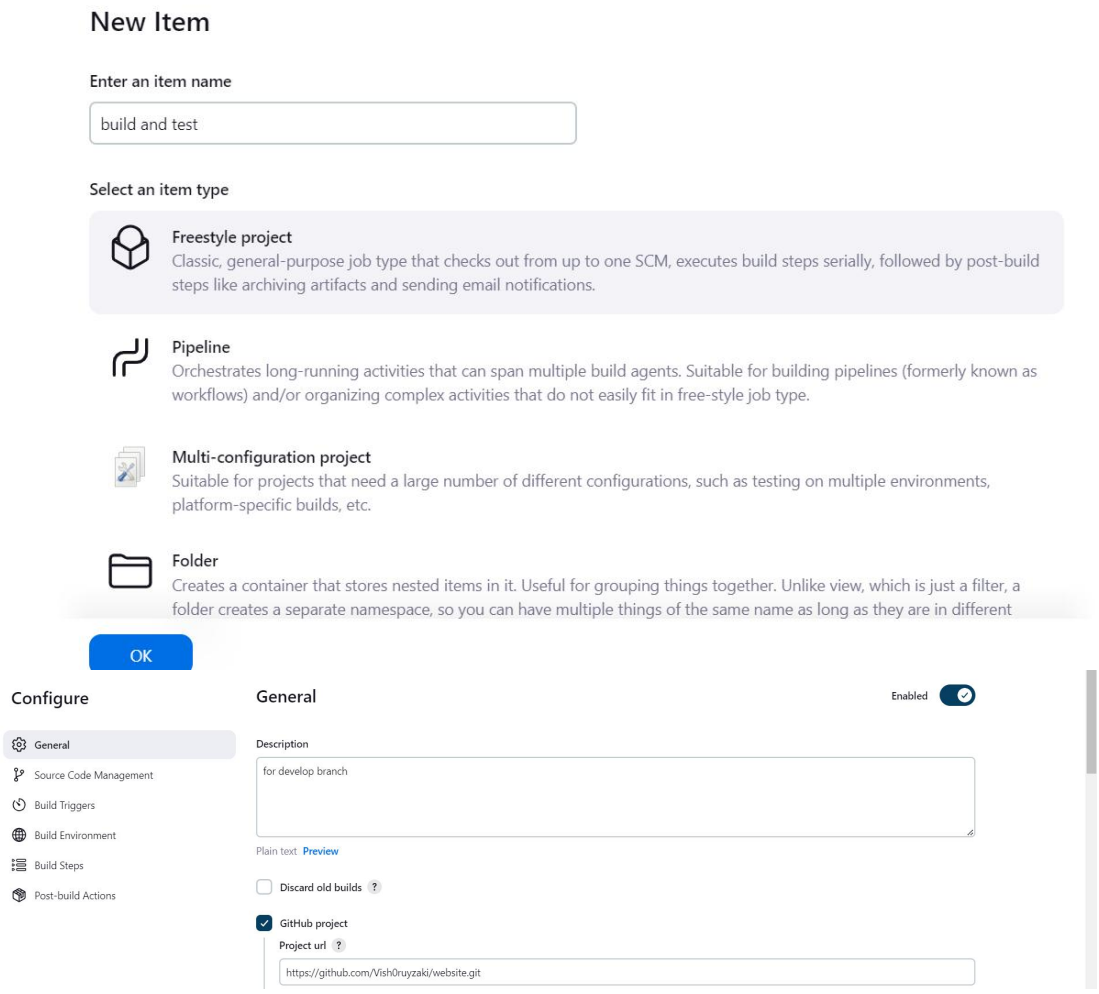
Push the changes back to github

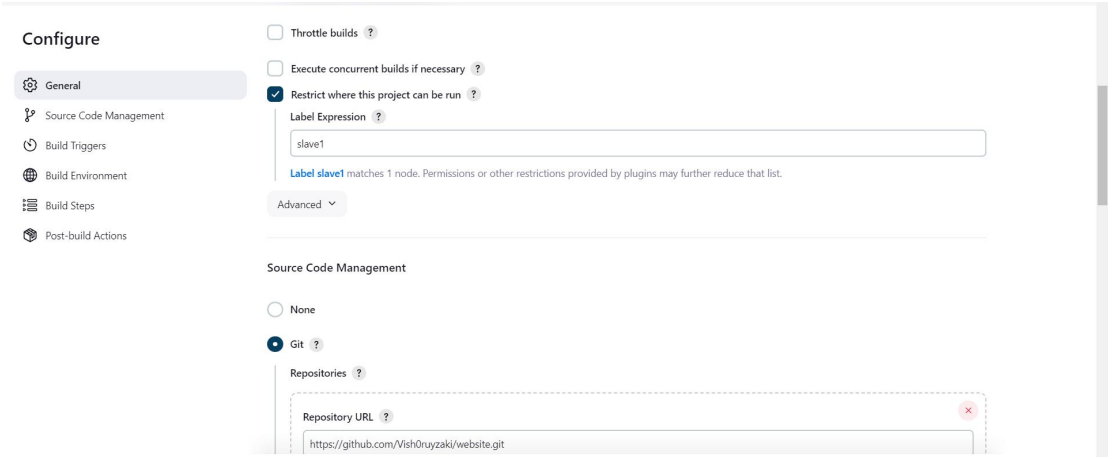


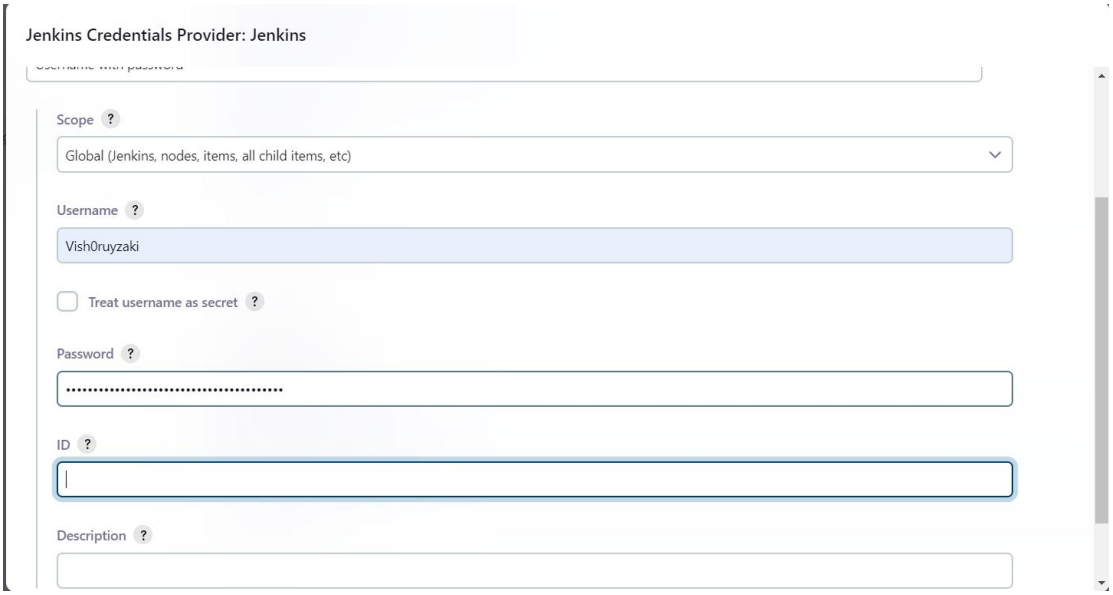
Step 10: lets create the below pipeline on jenkins to run the task

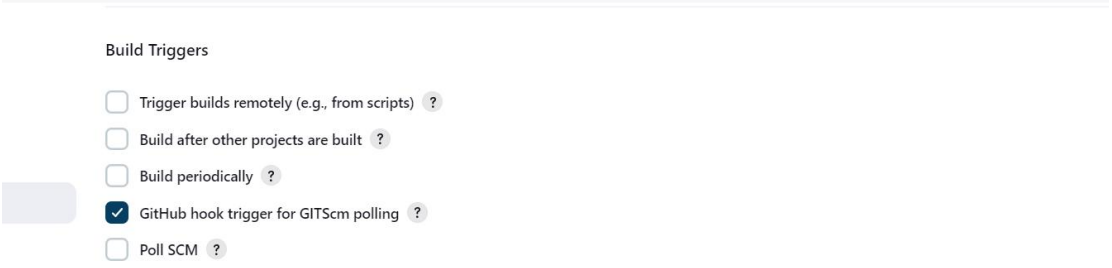
Job1 : build and test (mergeing two task together)

Job2 : prod



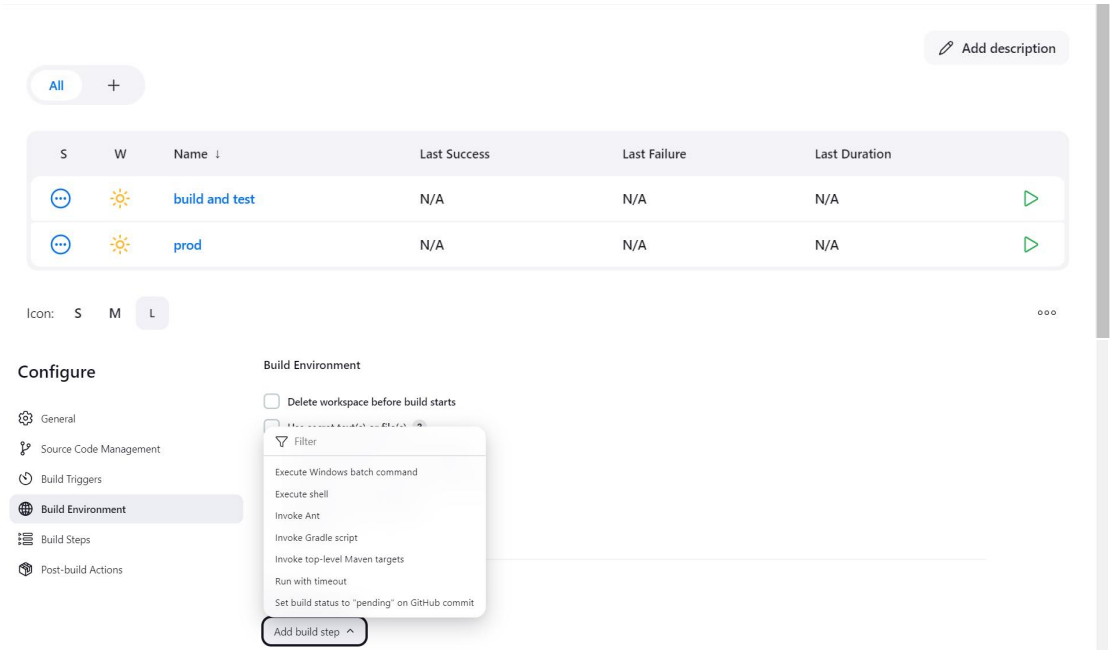




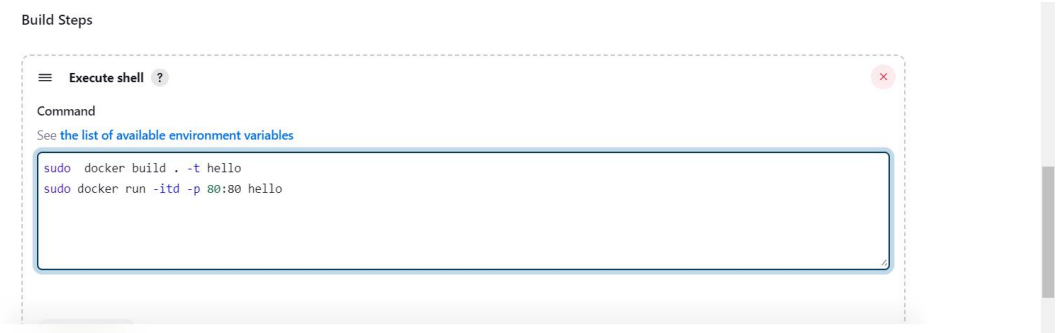


Similarty create job which will run on slave2 and will be restricted on master branch

One prod job is created we will add additional step which build after and add script command to run

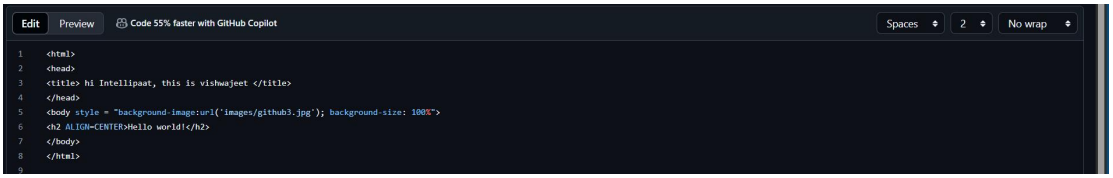


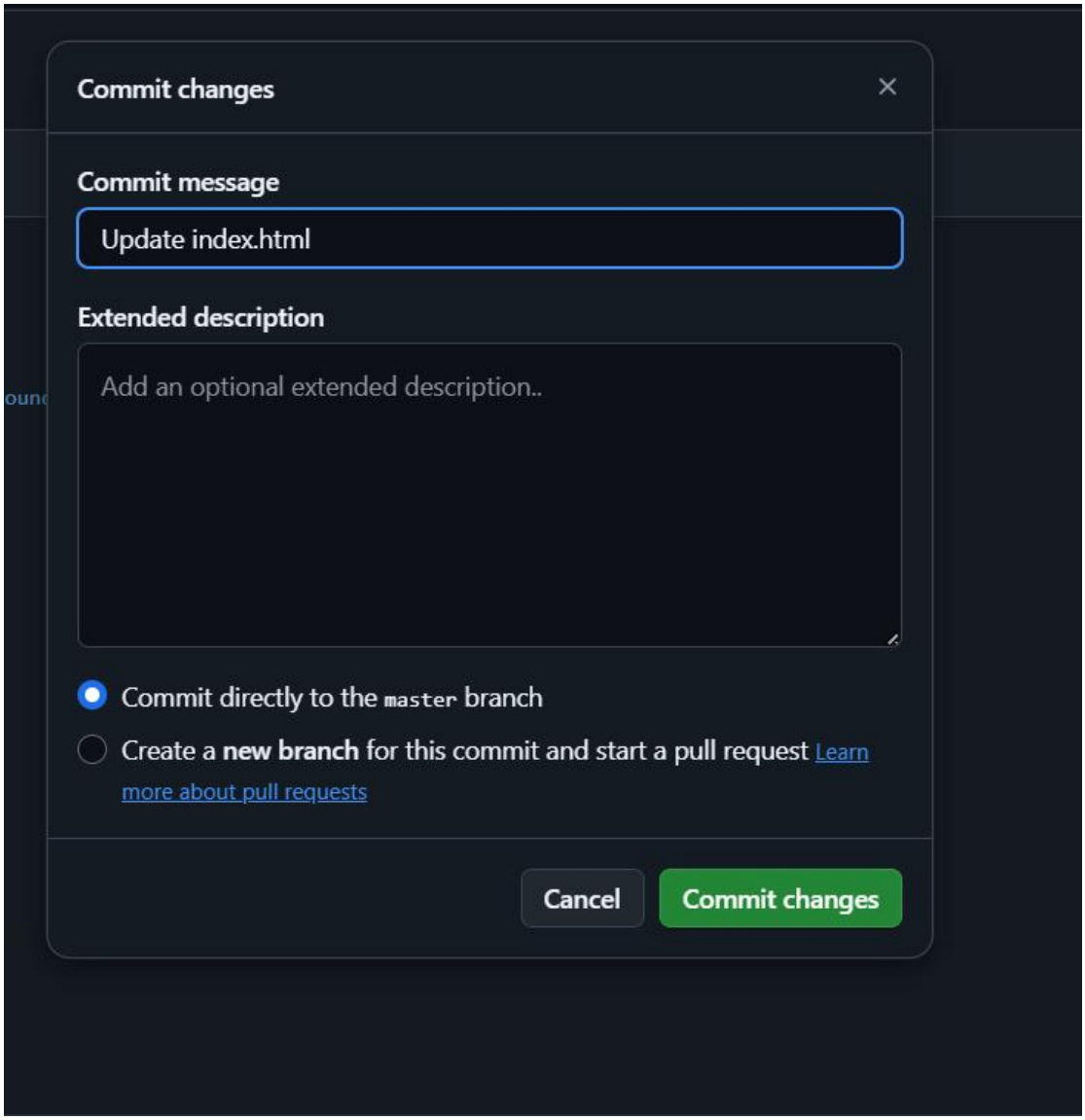
sudo docker build . -t hello
sudo docker run -itd -p 80:80 hello



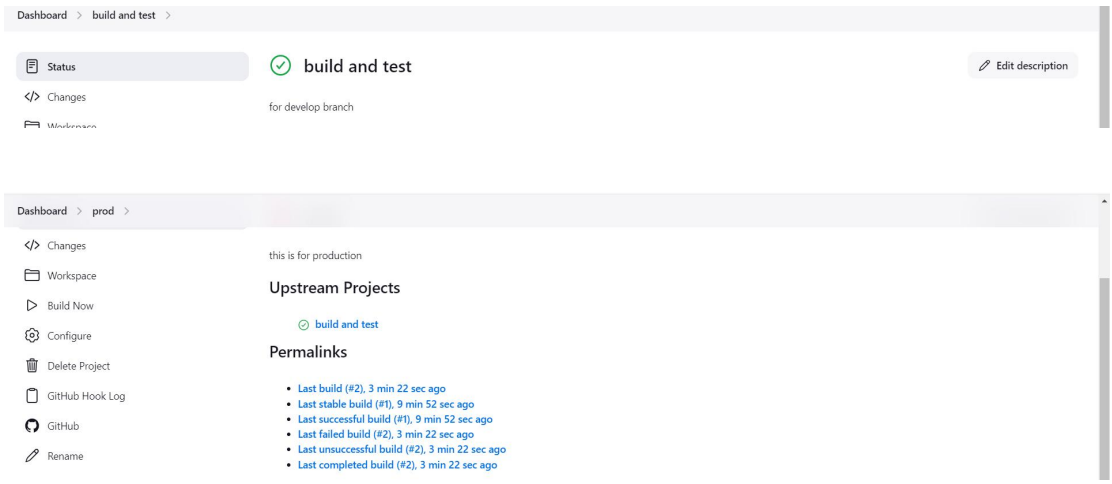
Step11: if commit is done on master, then test and prod job should run automatically

Made change in the html file now lets commit it





We can see the job run automatically once we made change,



We can see the webpage on browser