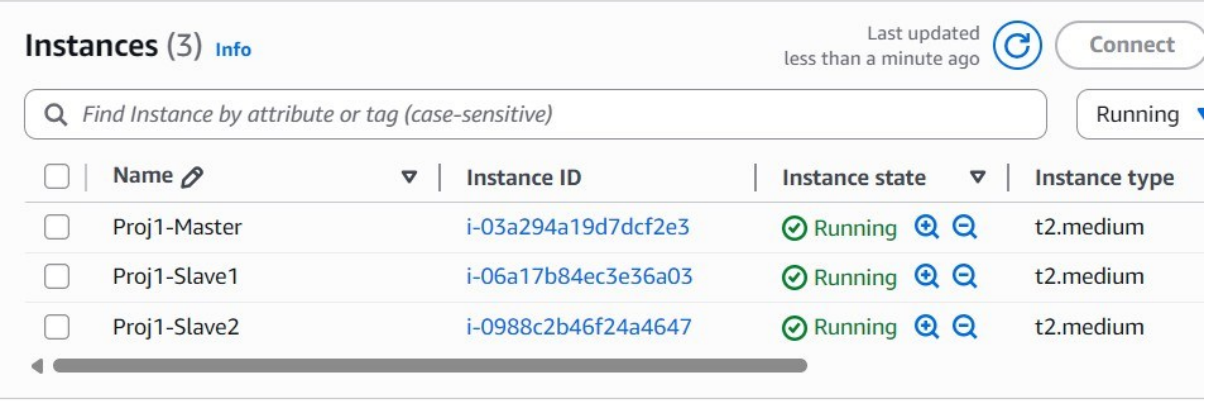
Capstone Project 1:-

Step1:- Launch 3 EC2 Instance – Proj1-Slave1 , Proj1-Slave2, Proj1-Master

Version – Ubuntu 22.0 LTS

t2.medium – default security group



Step2:- Connect all three nodes and update it

Sudo apt update -y

Step3:- Install Ansible in master node using ansible documentation.

Create **install.sh** script file and paste commands in script file

Implement the script file using **Bash install.sh**

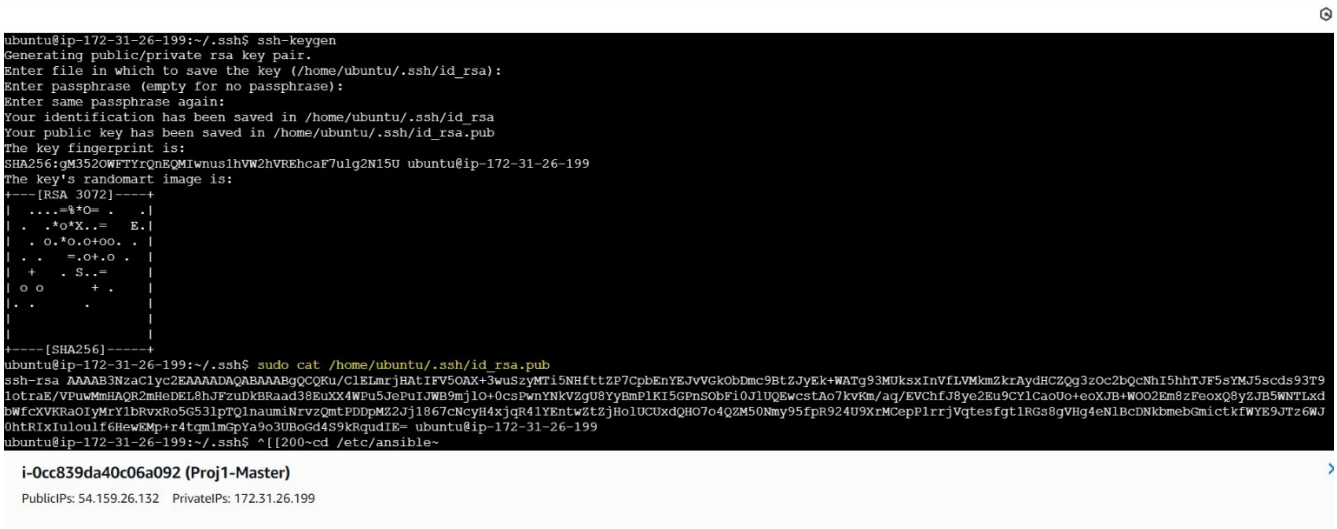
Step4:- Go in ssh folder – **cd .ssh**

There is one file authorized_keys

Generate key - **ssh-keygen**

sudo cat /home/ubuntu/.ssh/id_rsa.pub

Copy the key contain.



Step5:- Go in Slave1 Node and go inside ssh folder and in authorized_keys folder – **sudo nano authorized_keys**

Paste the key contain in it save and exite.

Step6:- Same steps with Slave 2

Step7:- Master Node – **cd /etc/ansible/** path and do **ls** and check available file init.

There is **hosts** file , Copy Slave1 and Slave2 PrivateIPs and

paste in host file.

Step8:-Now create a playbook for installing other things like java and Jenkins.

sudo nano play.yaml create and write a code inside the file.

- name: installing java and jenkins inside the machine where

ansible is installed

hosts: master

become: yes

tasks:

- name: install java and jenkins

script: master.sh

- name: installing java and docker inside the other 2 machines

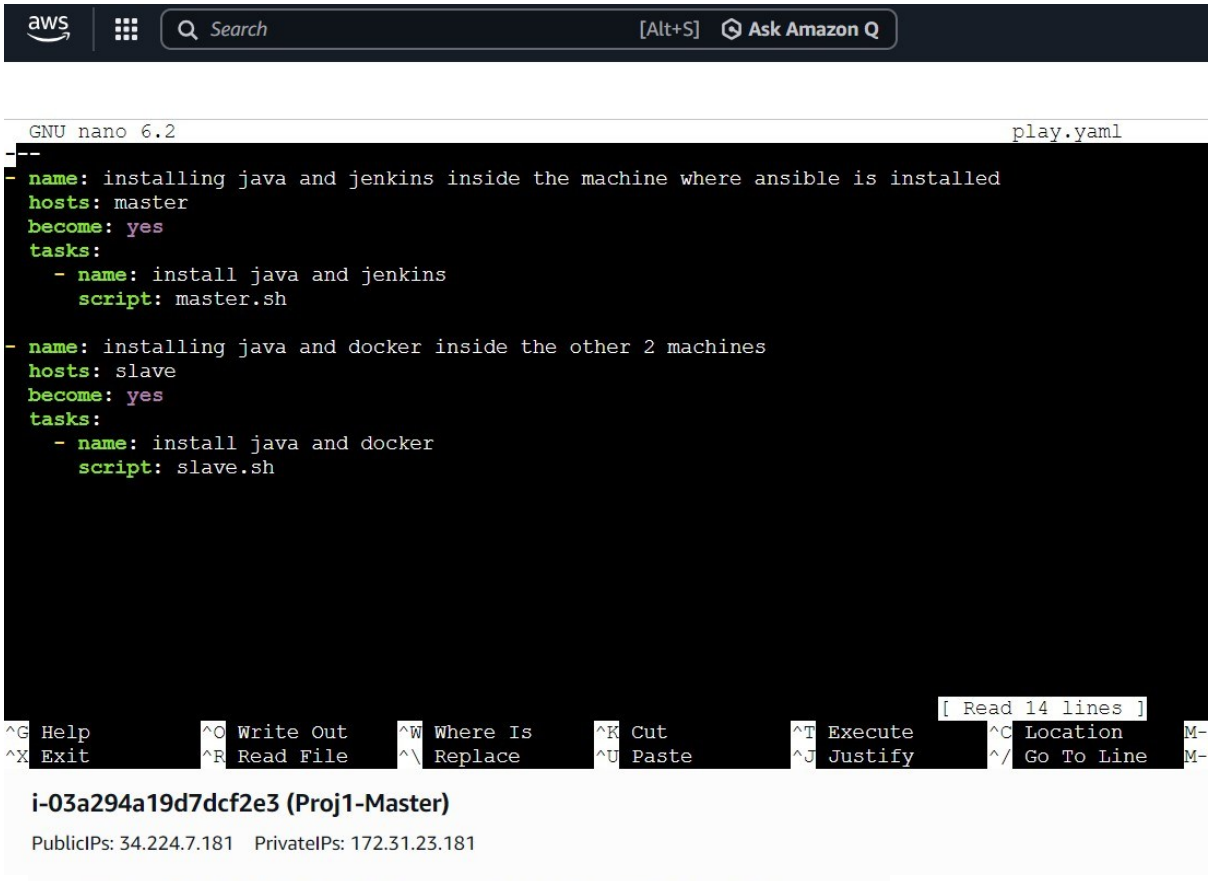
hosts: all

become: yes

tasks:

- name: install java and docker

script: slave.sh



Step9:-Create **master.sh** scrip file **sudo nano master.sh**

```
sudo apt update
sudo apt install openjdk-17-jdk  -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-get update
sudo apt-get install fontconfig openjdk-17-jre -y
sudo apt-get install jenkins -y
```

Step 10:-Create **slave.sh** file **sudo nano slave.sh**

```
Sudo apt update
Sudo apt install openjdk-17-jdk -y
Sudo apt install docker.io -y
```

```
  GNU nano 6.2
sudo apt update
sudo apt install openjdk-17-jre -y
sudo apt install docker.io -y

^G Help          ^O Write Out     ^W Where Is
^X Exit          ^R Read File     ^\ Replace
```

i-0cc839da40c06a092 (Proj1-Master)

PublicIPs: 54.159.26.132 PrivateIPs: 172.31.26.199

Step 11:-Execute the playbook – **ansible-playbook play.yaml**

```
ok: [localhost]

TASK [install java and jenkins] ***********************************************************************************************************************************
changed: [localhost]

PLAY [installing java and docker inside the other 2 machines] *****************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************************************************
[WARNING]: Platform linux on host 172.31.17.97 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.17.97]
[WARNING]: Platform linux on host 172.31.30.94 is using the discovered Python interpreter at /usr/bin/python3.10, but future installation of another Python
interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.17/reference_appendices/interpreter_discovery.html for more
information.
ok: [172.31.30.94]

TASK [install java and docker] ************************************************************************************************************************************
changed: [172.31.30.94]
changed: [172.31.17.97]

PLAY RECAP ********************************************************************************************************************************************************
172.31.17.97               : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
172.31.30.94               : 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-172-31-26-199:/etc/ansible$
```

i-0cc839da40c06a092 (Proj1-Master)

PublicIPs: 54.159.26.132 PrivateIPs: 172.31.26.199

Step 12:- Go to given git link

Create a new fork – **Dockerfile**

FROM ubuntu

RUN apt-get update

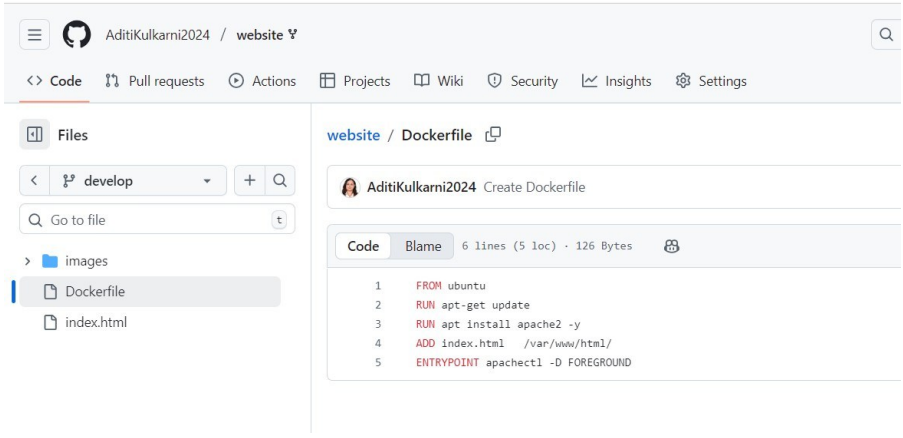
RUN apt install apache2 -y

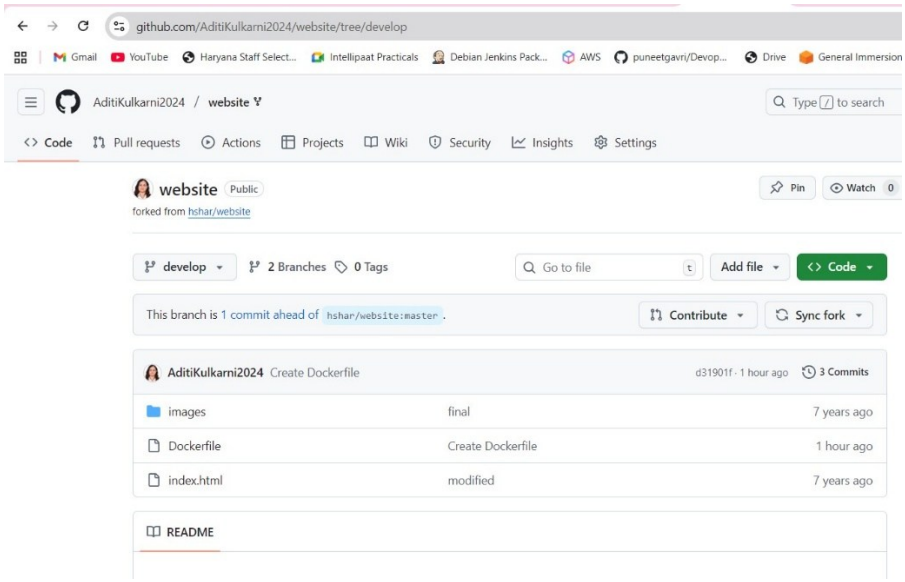
ADD index.html /var/www/html/

ENTRYPOINT apachectl -D FOREGROUND

Commit the changes and

Create one branch **develop**

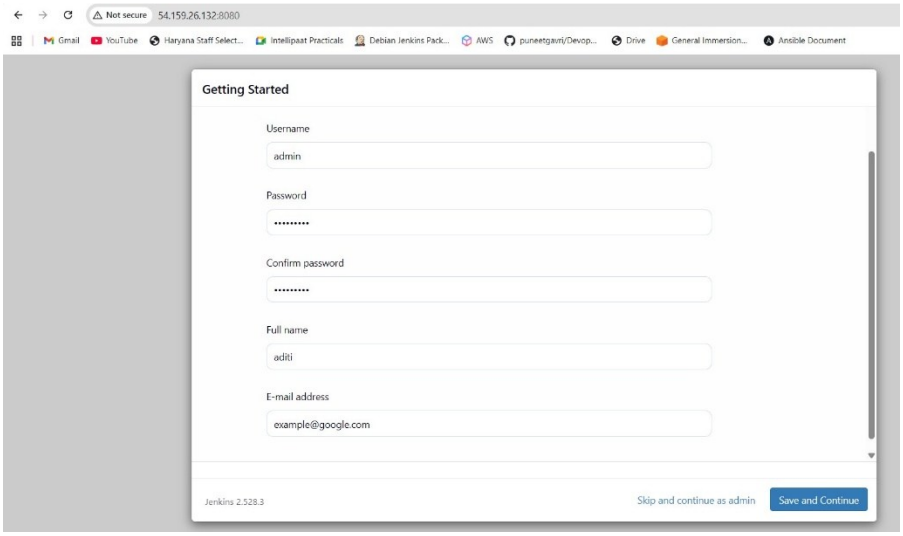




Step 13:- Go back to Master node and check Jenkins version.

Copy the PublicIP of master node and open Jenkins and

setup the Jenkins.



Step 14:-Manage Jenkins – Nodes – create new Node – **test-S1**

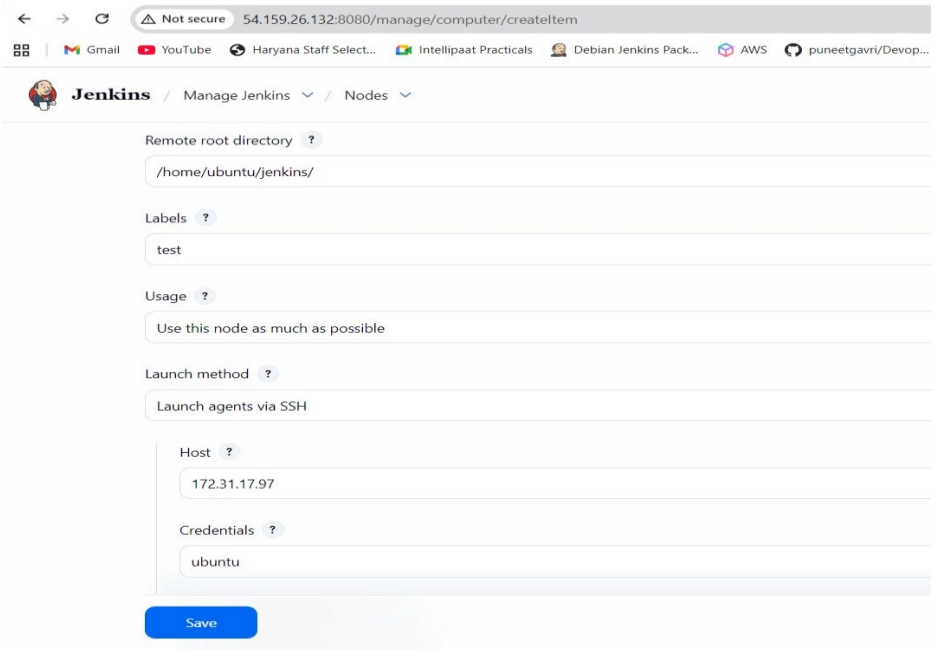
Remote root directory **-/home/ubuntu/Jenkins**

Labels **– test**

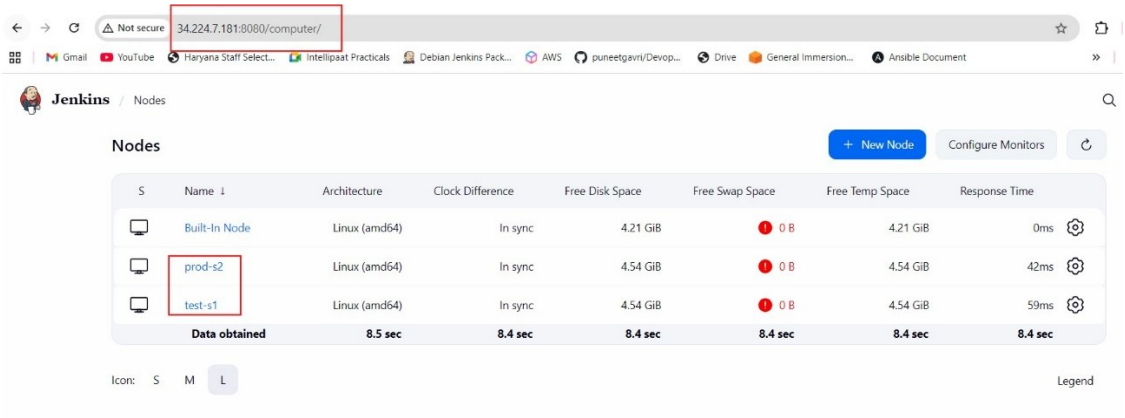
Copy Slave1 Private IP and paste in Host column

Credentials – Add **username - ubuntu** and **key-pem** file in secret key.

Save and create Node.



Step 14:- Create Node 2 name- **prod-S2** using same steps



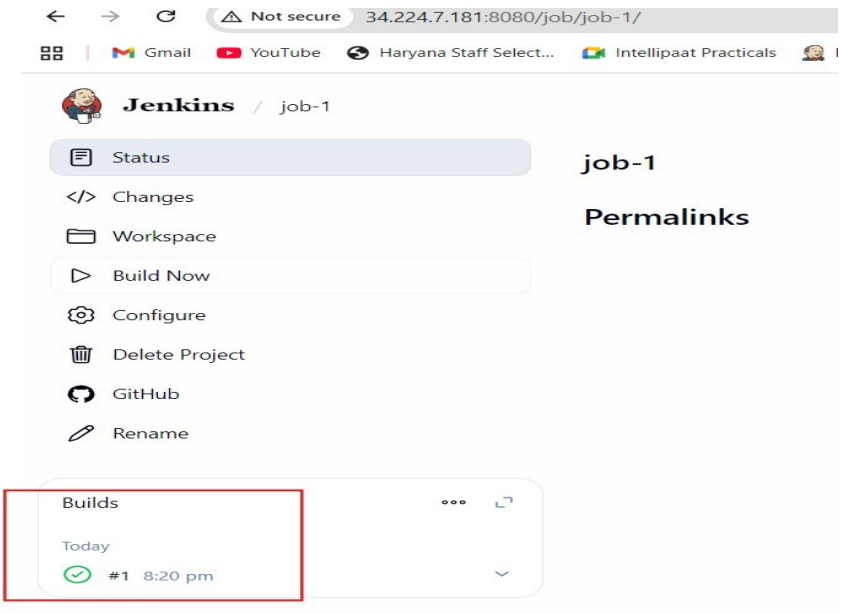
Step 15:-Create a Job – **Job-1** – freestyle project

GitHub project column – paste githup repo url

Label expression – test

Source code Management – same url

Branch Specifire - */develop - save and build the job.



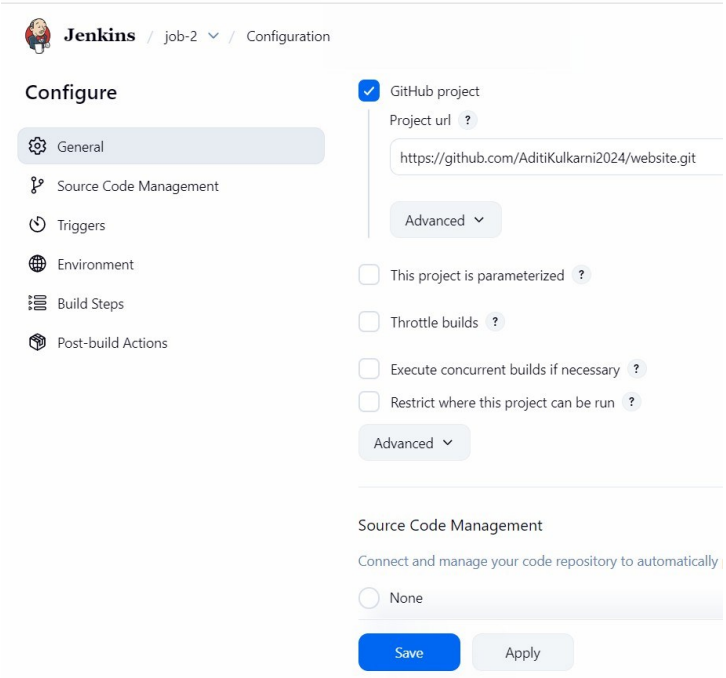
Step 17:-Create a **Job-2** in Jenkins – freestyle

GitHub – Git URL

Label – **prod**

Branch Specifire - */master

Save and create

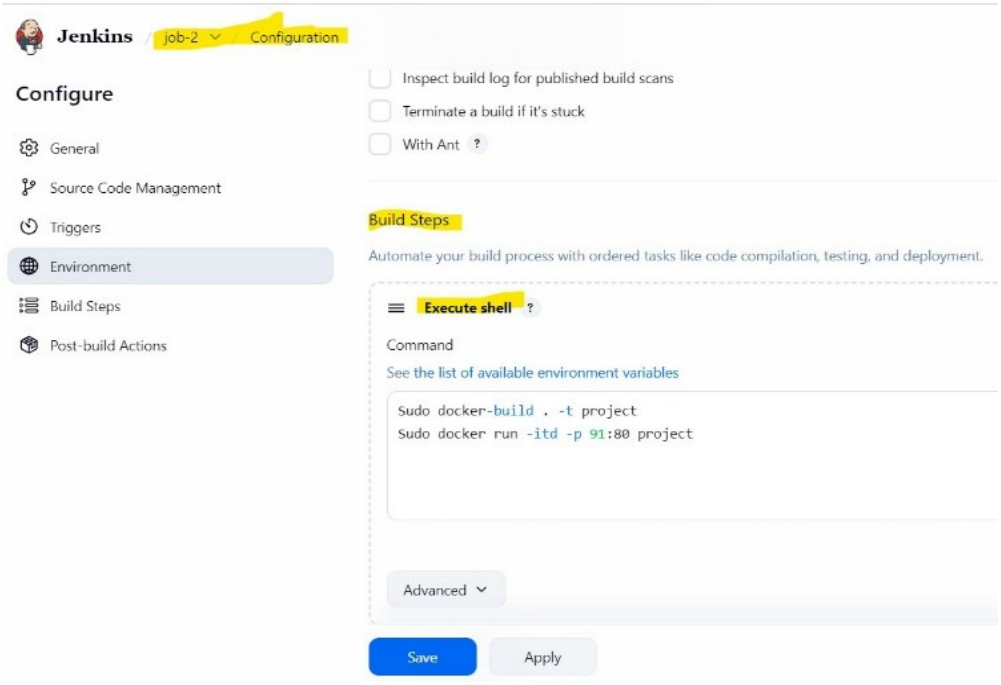


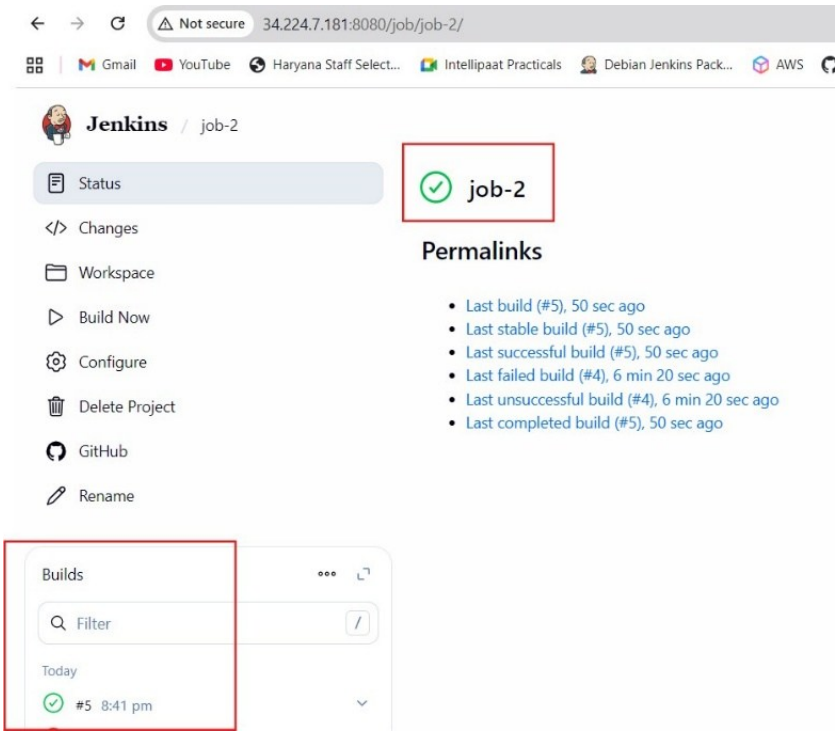
Again go to Job-2 -and configure and go to **Build Steps**

In Build Steps – Execute shell

Sudo docker-build . -t project

Sudo docker run -itd -p 91:80 project





Save and Build – go to Slave2 copy PublichIp and check on browser.