1. **We have to launch 3 instance**

- 1 instance as master
- 2 instance as test
- 3 instance as prod

2. **Update machine:**

sudo apt update

3. **Go to the documentation of ansible:**

https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html#installing-ansible-on-ubuntu

Run the command on **master**

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

4. **Generate a SSH KEY ON MASTER**

Ssh-keygen

Hit enter 3 times

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

Copy SSH key from master

5. **Go to test machine**

Cd .ssh>ls>sudo nano authorized_keys

Paste the ssh key >Ctrl +s >Ctrl+X

6. **Go to prod machine**

Cd .ssh>ls>sudo nano authorized_keys

Paste the ssh key >Ctrl +s >Ctrl+X

7. **Go to master machine**

Cd /etc/ansible

Ls

Sudo nano hosts

(Paste the private ip of both slaves)

Slave1 ansible_host=172.31.9.93

Slave2 ansible_host=172.31.0.169

Ctrl+s and ctrl+x

8. ansible -m ping all
9. Sudo nano play.yaml

   (Inside the file)

```
---
- name: task for master
  hosts: localhost
  become: true
  tasks:
  - name: executing script master
    script: master.sh


- name: task for slave
  hosts: all
  become: true
  tasks:
  - name: executing script slaves
    script: slave.sh
```

10. Sudo nano master.sh

    (Inside file)

    sudo apt update

sudo apt install openjdk-11-jdk -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y

11. Sudo nano slave.sh
    sudo apt update
    sudo apt install openjdk-11-jdk -y
    sudo apt install docker.io -y
12. ansible-playbook play.yaml --syntax-check
13. ansible-playbook play.yaml --check
14. ansible-playbook play.yaml
15. Check jenkins and java on master
    Java –version
16. Check java and docker on prod and test
    Java –version
    Docker –version
17. Copy the public ip of master go on a new tab run it with port 8080 (Public ip:8080)
18. Copy the path of jenkins where the password is given (/var/lib/jenkins/secrets/initialAdminPassword)
19. Come back to your master
    Sudo cat /var/lib/jenkins/secrets/initialAdminPassword
20. Copy the password

21. Come back to jenkins> paste password>continue>install suggested plugins>write user name password> continue and finish.
22. Go on Manage Jnekins> System Configuration>Nodes> Create Node
    Node Name: slave1> permanent agent
    Remote root directory:/home/ubuntu/jenkins/
    Launch method: via ssh
    Host: paste the private ip of test
    credentials:> add> jenkins>
    Kind:SSH username with private key
    Username: ubuntu
    Private key>eneter directly:paste the pem file>add
    Hoat key Verification: Non verifying strategy
    Save > Refresh

Go on Manage Jnekins> System Configuration>Nodes> Create Node
    Node Name: slave1> permanent agent
    Remote root directory:/home/ubuntu/jenkins/
    Launch method: via ssh
    Host: paste the private ip of prod
    credentials:> add> jenkins>
    Kind:SSH username with private key
    Username: ubuntu
    Private key>eneter directly:paste the pem file>add
    Hoat key Verification: Non verifying strategy
    Save > Refresh

23. Click on the link given on project >fork it

24. Click on ADD file on github
25. Create new file

    name= Dockerfile

    FROM ubuntu

    RUN apt update

    RUN apt install apache2 -y

    ADD . /var/www/html/

    ENTRYPOINT apachectl -D FOREGROUND

    Commit changes
26. Create branch name =Develop
27. Go to dashboard> Create Job1

    Name: Job1

    Freestyle project

    Create

    - GIt hub project: add repo
    - Click on Restrict where the project run: Slave1
    - Source Code Management: Git> paste URL
    - Specify branches: develop
    - Click on Git scm polling > Copy the main Url > go to github>settings>webhook>add webhook> paste the url(url/github-webhook/) > add > click the link> recent deliveries
    - Come back to dashboard>
    - Save >Build

28. Go on test

    Ls

    Cd jenkins

Ls

Cd workspace

Ls

Job1

29.Go back to your dashboard

Job1> Configure> Build Step>Execute shell

sudo docker build /home/ubuntu/jenkins/workspace/Job1/ -t imagejob1

sudo docker run -itd -p 85:80 –name c1 imagejob1

Add >BUild now

Copy the public ip of test and run it with port 85

30 Similarly create job2 and job3