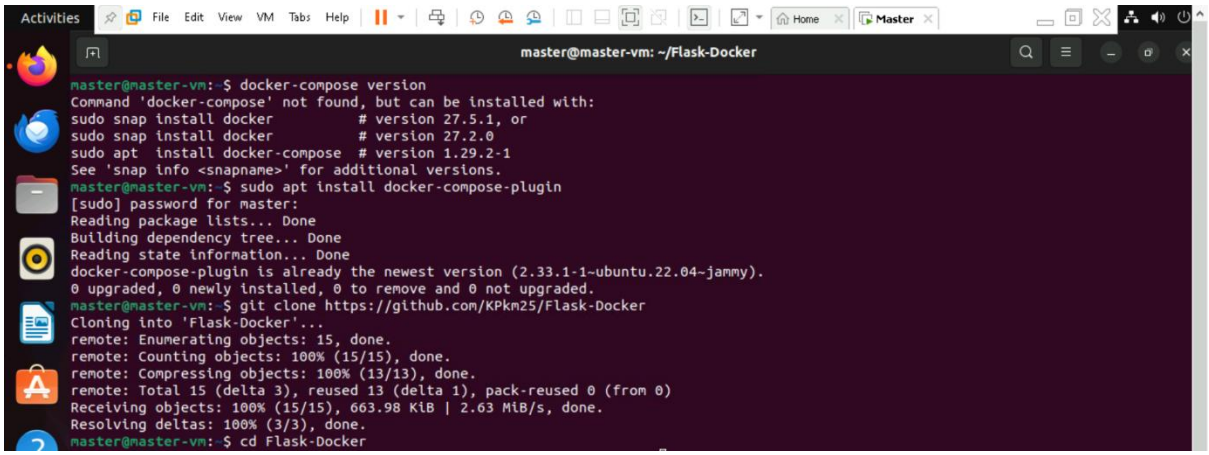
1. **Multi-Container Flask Application with PostgreSQL Using Docker Compose**

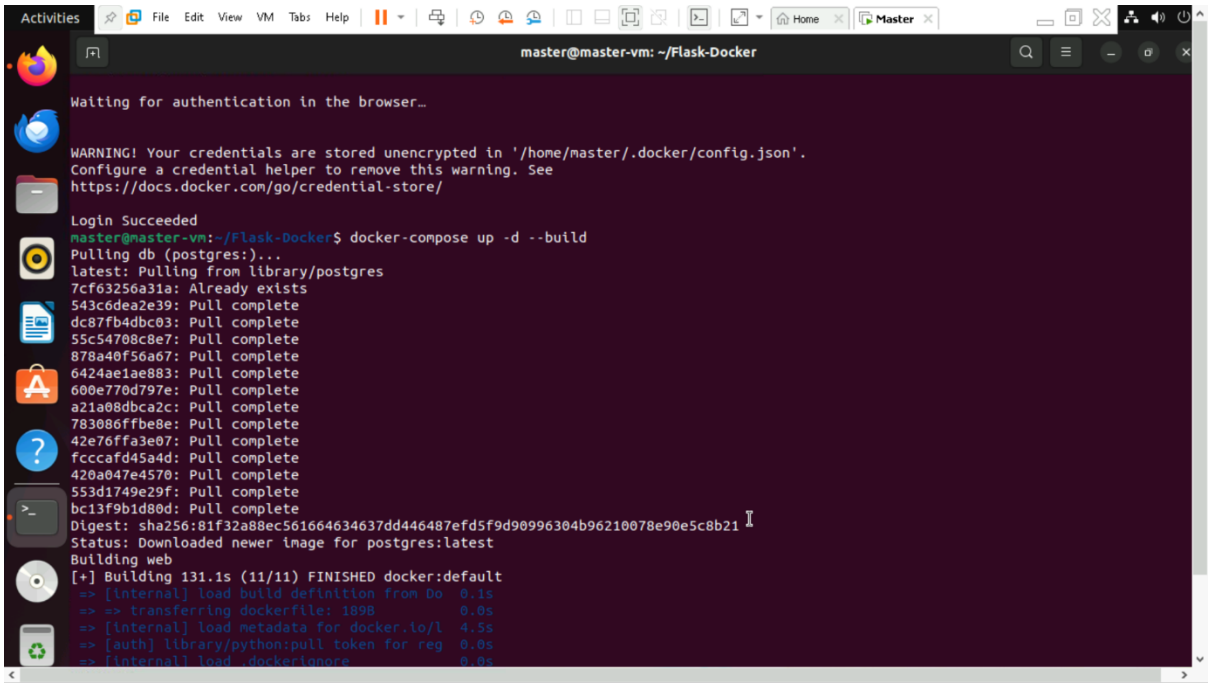
**Step 1: Download Docker .**

***Sudo apt install docker-compose-plugin***

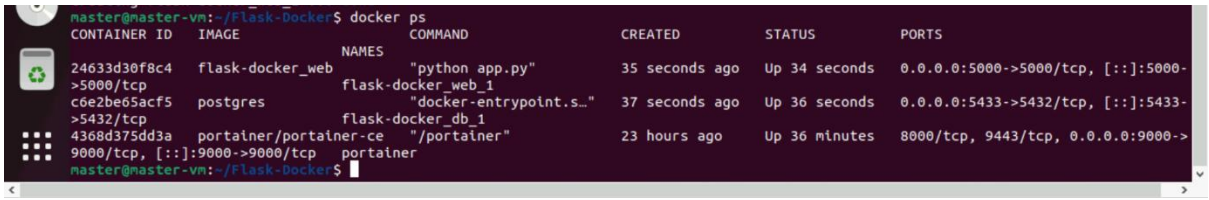
**Step 2: Clone the git repository and move to directory cd Flask-Docker**




**Step 3: Before build the containers login to docker hub using command *"docker login"*. Build and start the containers using *docker-compose up -d –build.***

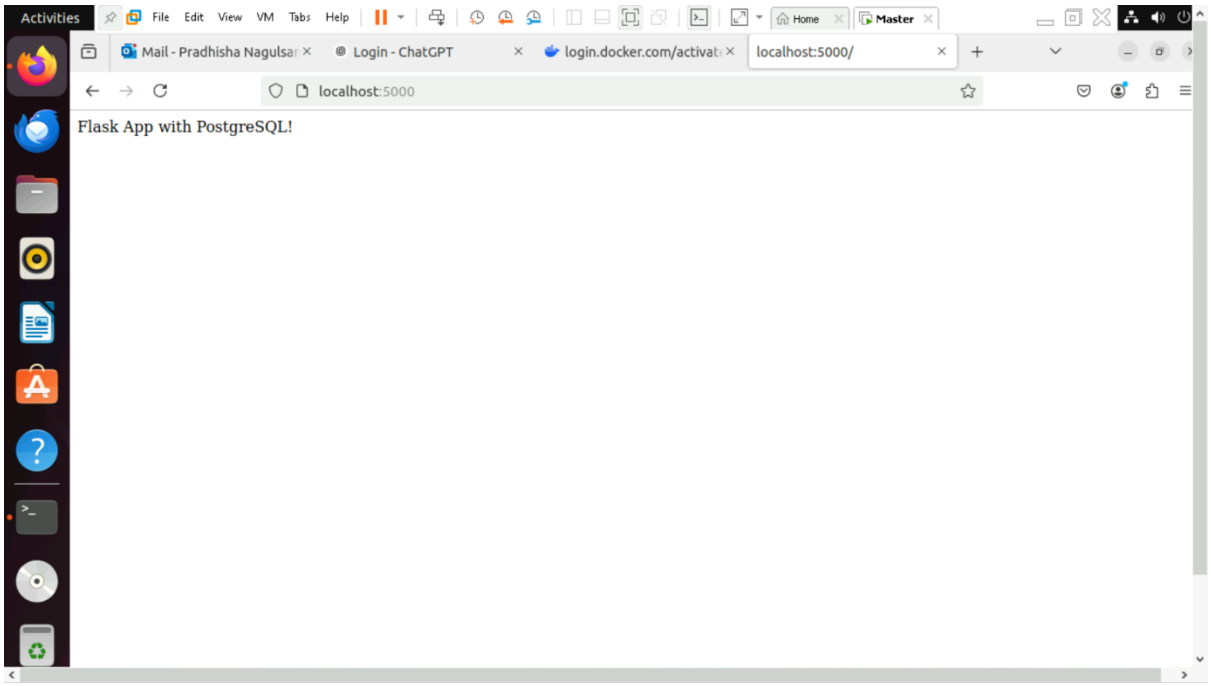



**Step 4: Verify the running containers using *"docker ps"*. It used to verify whether web and db is there or not.**

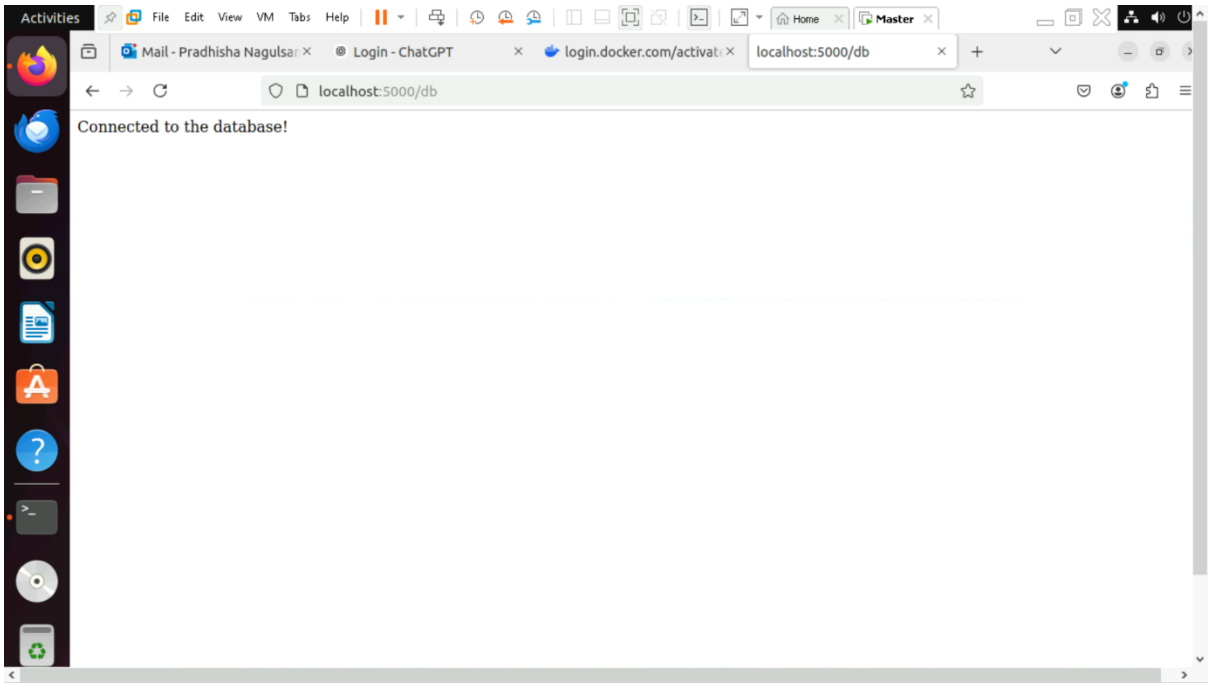


Step 5: Test the application.

http://localhost:5000/ → Should return "Flask App with PostgreSQL!"




http://localhost:5000/db → Should confirm database connection.



## 2. **Jenkins + Docker Pipeline Project Documentation**

**Step 1: Install Docker on Jenkins Server**

1. **Update system packages and install Docker:**

   sudo apt update

   sudo apt install docker.io -y

2. **Start and enable Docker:**

   sudo systemctl start docker

   sudo systemctl enable docker

3. **Add Jenkins user to Docker group (to allow Jenkins to run Docker commands):**

   sudo usermod -aG docker jenkins

4. **Restart Jenkins to apply changes:**

   sudo systemctl restart jenkins

5. **Verify Docker installation:**

   docker –version

**Step 2: Enable Password Authentication (If Needed)**

If SSH key authentication is not set up, enable password login:

1. **Connect to the remote server and edit the SSH configuration file:**

   sudo nano /etc/ssh/sshd_config

2. **Modify these lines:**

   PasswordAuthentication yes

   PermitRootLogin yes

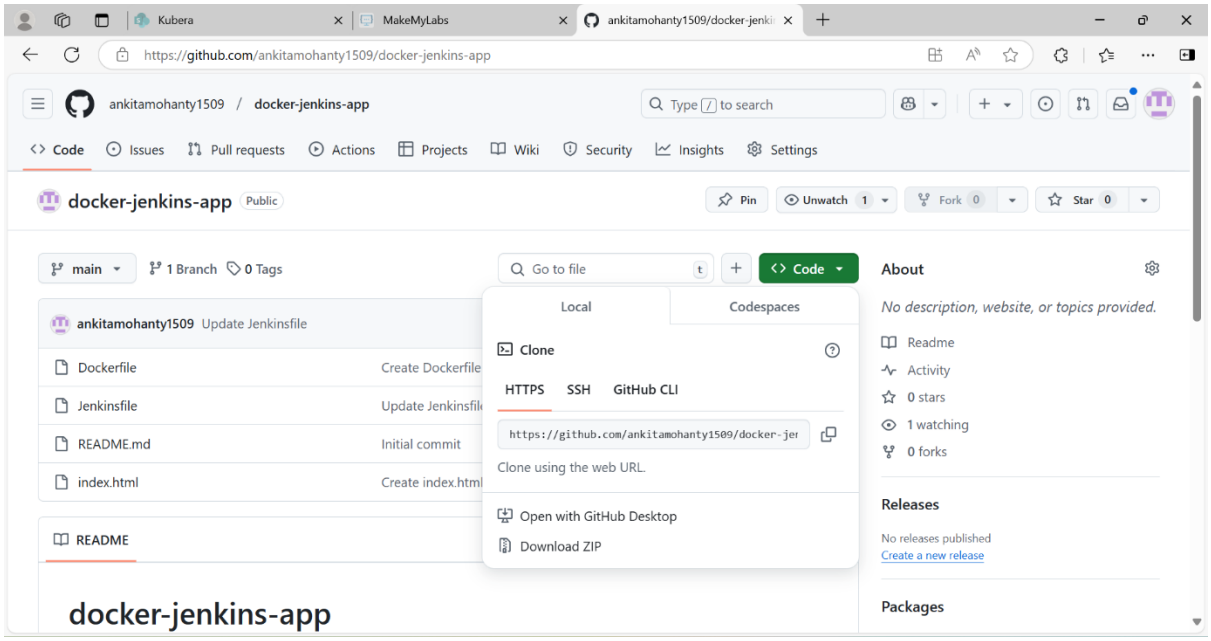
3. **Save the file and restart SSH:**

   sudo systemctl restart ssh

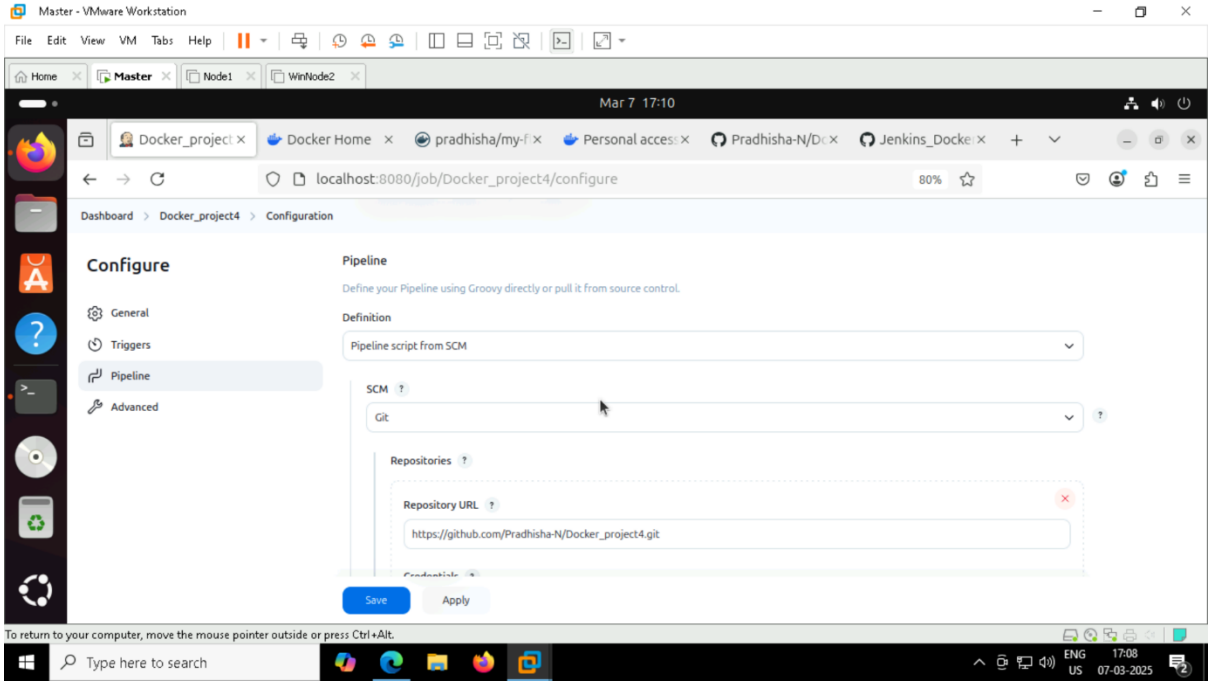
4. **Test SSH login:**

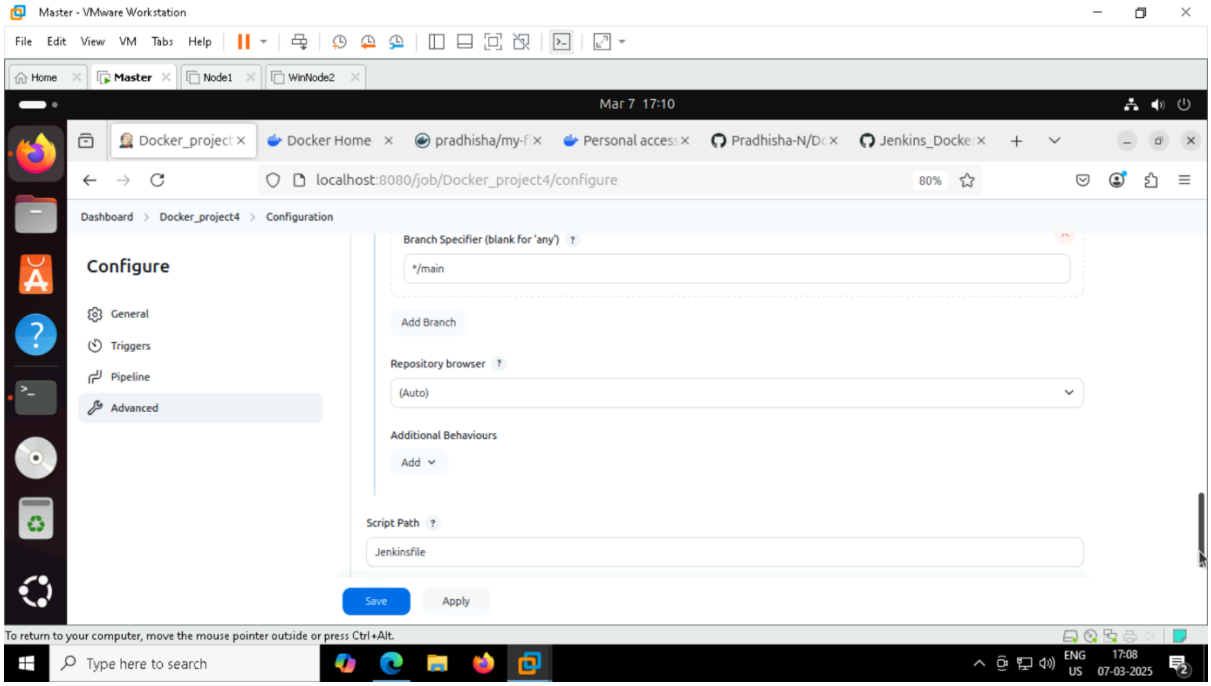
   ssh master@192.168.203.128

## Step 3: Create a Simple Web Application

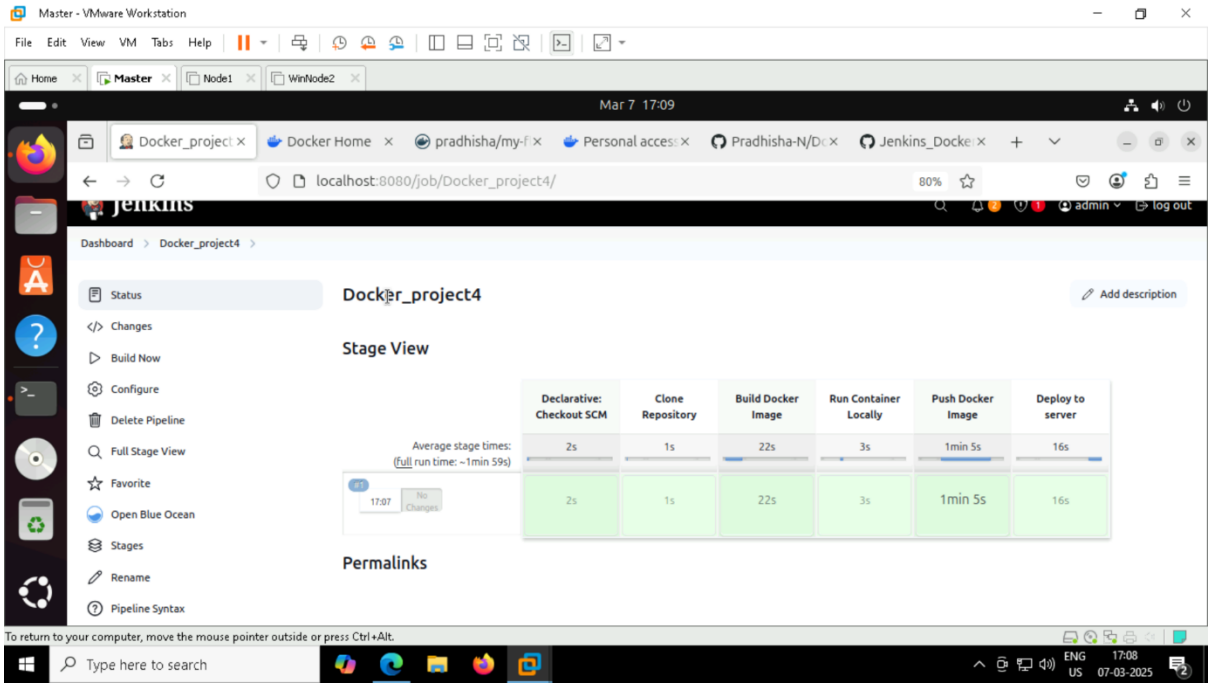


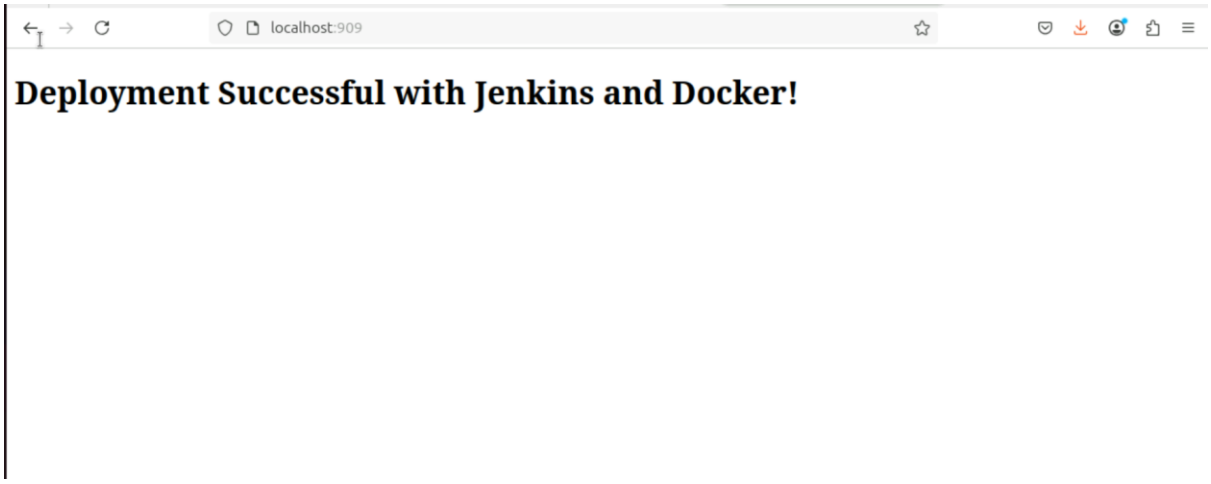

## Step 4: Create a Jenkins Pipeline






## Step 5: Run the Jenkins Pipeline

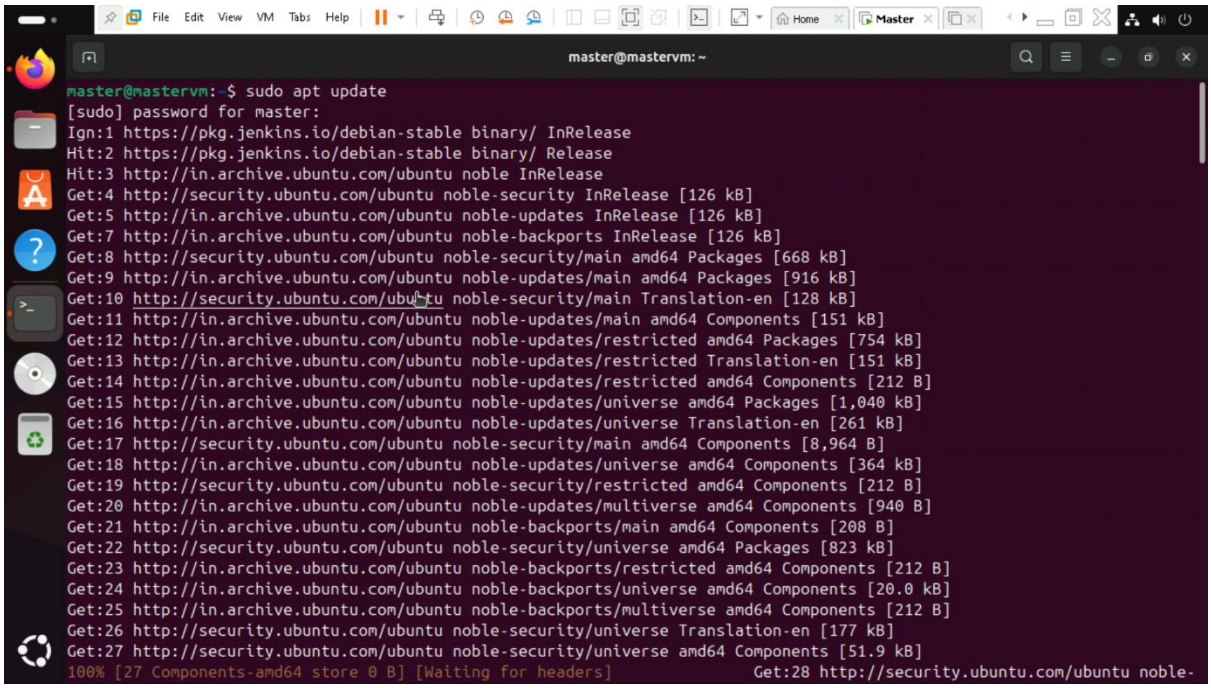





## 3. Kubectl setup

### 1. Install Kubernetes on Ubuntu

a. Install Dependencies



```
master@mastervm: ~
Reading state information... Done
273 packages can be upgraded. Run 'apt list --upgradable' to see them.
master@mastervm:~$ sudo apt install -y apt-transport-https ca-certificates curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203).
ca-certificates set to manually installed.
The following package was automatically installed and is no longer required:
  libllvm17t64
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  apt-transport-https curl
0 upgraded, 2 newly installed, 0 to remove and 273 not upgraded.
Need to get 230 kB of archives.
After this operation, 569 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-https all 2.7.14build2 [3,974 B]
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 curl amd64 8.5.0-2ubuntu10.6 [226 kB]
Fetched 230 kB in 1s (168 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (Dialog frontend requires a screen at least 13 lines tall and 31 columns wide.)
debconf: falling back to frontend: Readline
Selecting previously unselected package apt-transport-htt
ps.
(Reading database ... 159871 files and directories curren
tly installed.)
Preparing to unpack .../apt-transport-https_2.7.14build2_
all.deb ...
Unpacking apt-transport-https (2.7.14build2) ...
```

b. Install kubectl

```
master@mastervm: ~
master@mastervm:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd6
4/kubectl"

chmod +x kubectl

sudo mv kubectl /usr/local/bin/

kubectl version -client
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --100   138  100   138    0     0    395      0 --:--:-- --:--:--
 --:--:--   396
  9 54.6M    9 5197k    0     0  6188k      0  0:00:09 -- 39 54.6M   39 21.7M    0     0  11.8M      0  0:00:04  0 77 54
.6M   77 42.5M    0     0  14.9M      0  0:00:03  0100 54.6M  100 54.6M    0     0  16.2M      0  0:00:03  0:00:03 --:--
:-- 19.6M
error: extra arguments: [-client]
master@mastervm:~$ kubectl version
Client Version: v1.32.2
Kustomize Version: v5.5.0
Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fversion%3Ftimeout%3D
32s'/><script id='redirect' data-redirect-url='/login?from=%2Fversion%3Ftimeout%3D32s' src='/static/4cd96940/scripts/red
irect.js'></script></head><body style='background-color:white; color:white;'>
Authentication required
<!--
-->

</body></html>
master@mastervm:~$ kubectl version v1.32.2
error: extra arguments: [v1.32.2]
```

c. Install minikube

```
master@mastervm:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

chmod +x minikube-linux-amd64

sudo mv minikube-linux-amd64 /usr/local/bin/minikube
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --  0     0    0     0    0     0      0      0 --:--:-- --  0  1
19M    0  110k    0     0    98k      0  0:20:46  0  1  119M    1 2368k    0     0  1118k      0  0:01:49  0 16  119M
16 19.6M    0     0  6455k      0  0:00:18  0 32  119M   32 38.9M    0     0  9677k      0  0:00:12  0 48  119M   48 57.
9M    0     0  11.1M      0  0:00:10  0 63  119M   63 76.2M    0     0  12.3M      0  0:00:09  0 79  119M   79 94.7M
0     0  13.1M      0  0:00:09  0 91  119M   91  109M    0     0  13.4M      0  0:00:08  0100  119M  100  119M    0
0  13.5M      0  0:00:08  0:00:08 --:--:-- 17.1M
```

d. Start Minikube

```
master@mastervm:~$ minikube start --driver=docker
😄  minikube v1.35.0 on Ubuntu 24.04
✨  Using the docker driver based on user configuration

💣  Exiting due to PROVIDER_DOCKER_NEWGRP: "docker version --format <no value>-<no value>:<no value>" exit status 1: per
mission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2F
run%2Fdocker.sock/v1.45/version": dial unix /var/run/docker.sock: connect: permission denied
💡  Suggestion: Add your user to the 'docker' group: 'sudo usermod -aG docker $USER && newgrp docker'
📘  Documentation: https://docs.docker.com/engine/install/linux-postinstall/
```

e. Check status

```
master@mastervm:~$ kubectl cluster-info
E0311 13:12:52.589497    6750 memcache.go:265] "Unhandled Error" err=<
        couldn't get current server API group list: <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2
Fapi%3Ftimeout%3D32s'/><script id='redirect' data-redirect-url='/login?from=%2Fapi%3Ftimeout%3D32s' src='/static/4cd9694
0/scripts/redirect.js'></script></head><body style='background-color:white; color:white;'>
```

## 2. Create project

a. Create folder

```
Help
Home   Master   Node1
Mar 11 15:17
master@mastervm: ~/flask-ci-cd
master@mastervm:~$ mkdir flask-ci-cd

cd flask-ci-cd
```

b. Create app.py

```
master@mastervm:~/flask-ci-cd$ nano app.py
```

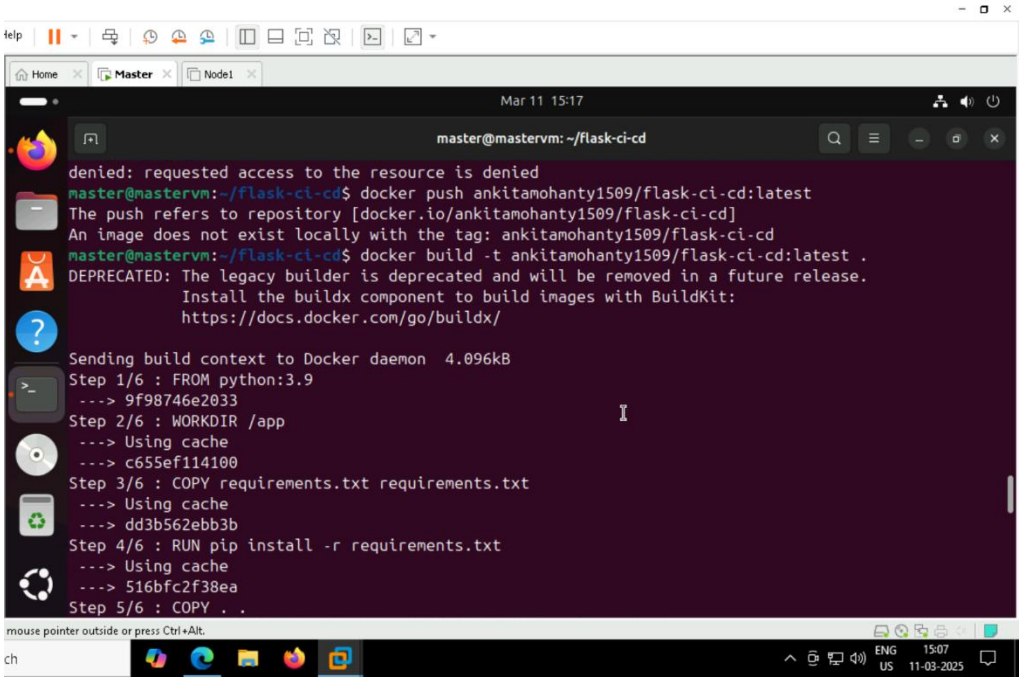
c. Create Dockerfile

```
master@mastervm:~/flask-ci-cd$ nano Dockerfile
```

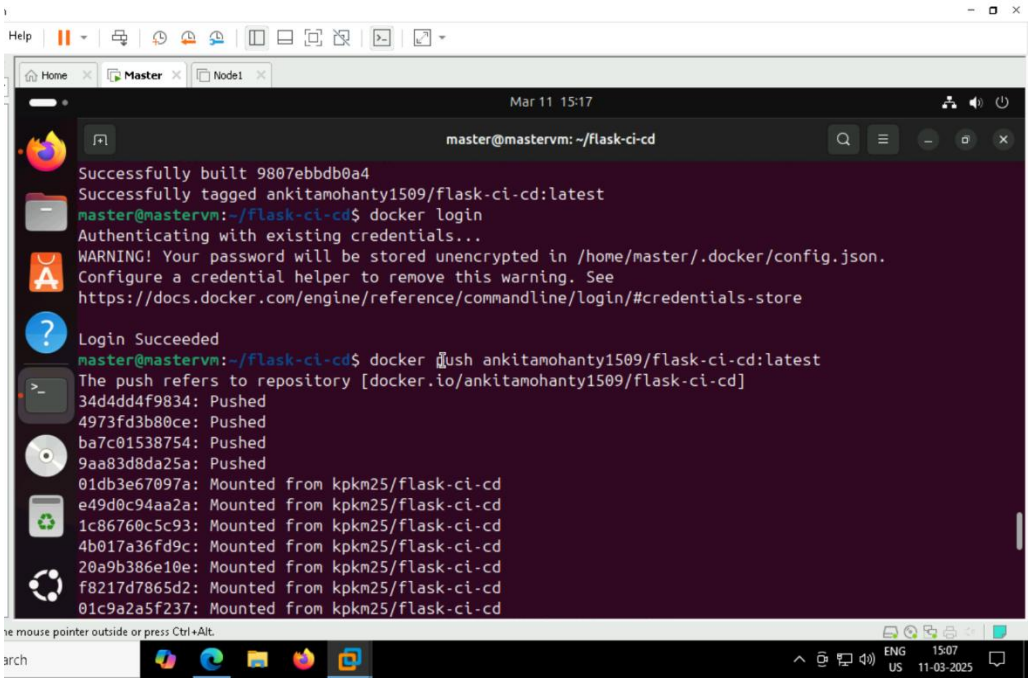
d. Create Create requirements.txt

```
master@mastervm:~/flask-ci-cd$ nano requirements.txt
master@mastervm:~/flask-ci-cd$ docker build -t kpkm25/flask-ci-cd:latest .

docker login

docker push kpkm25/flask-ci-cd:latest
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  4.096kB
Step 1/6 : FROM python:3.9
3.9: Pulling from library/python
155ad54a8b28: Pulling fs layer
8031108f3cda: Pull complete
1d281e50d3e4: Pull complete
447713e77b4f: Pull complete
```

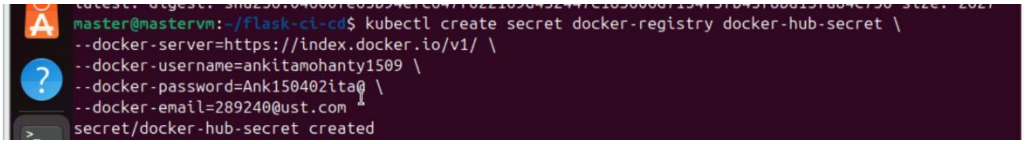
## 3. Build Docker Image




a. Push Docker Image




## 4. Connect Kubernetes to Docker




## 5. Kubernetes Deployment

a. Create k8s-deployment.yaml

```
master@mastervm:~/flask-ci-cd$ nano k8s-deployment.yaml
master@mastervm:~/flask-ci-cd$ kubectl apply -f k8s-deployment.yaml
error when retrieving current configuration of:
Resource: "apps/v1, Resource=deployments", GroupVersionKind: "apps/v1, Kind=Deployment"
Name: "", Namespace: "default"
from server for: "k8s-deployment.yaml": resource name may not be empty
error when retrieving current configuration of:
Resource: "/v1, Resource=services", GroupVersionKind: "/v1, Kind=Service"
Name: "", Namespace: "default"
```

## 6. Apply the deployment

```
master@mastervm:~/flask-ci-cd$ kubectl apply -f k8s-deployment.yaml
deployment.apps/flask-app created
service/flask-service created
```

## 7. Check if the pods are running

```
master@mastervm:~/flask-ci-cd$ kubectl get pods
NAME                          READY   STATUS    RESTARTS   AGE
flask-app-58b8cc8758-kj5v5    1/1     Running   0          3m48s
flask-app-58b8cc8758-wvl9m    1/1     Running   0          3m48s
```