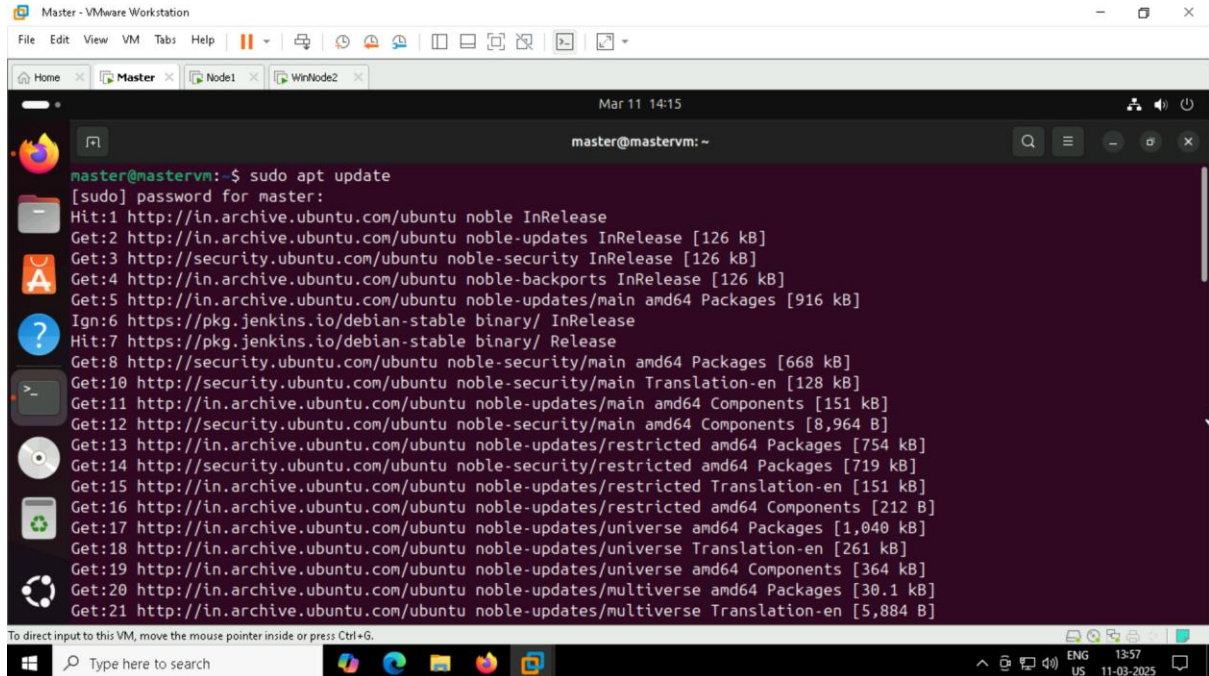


# DOCKER PROJECT-3

## Dockerized CI/CD Pipeline using Kubernetes

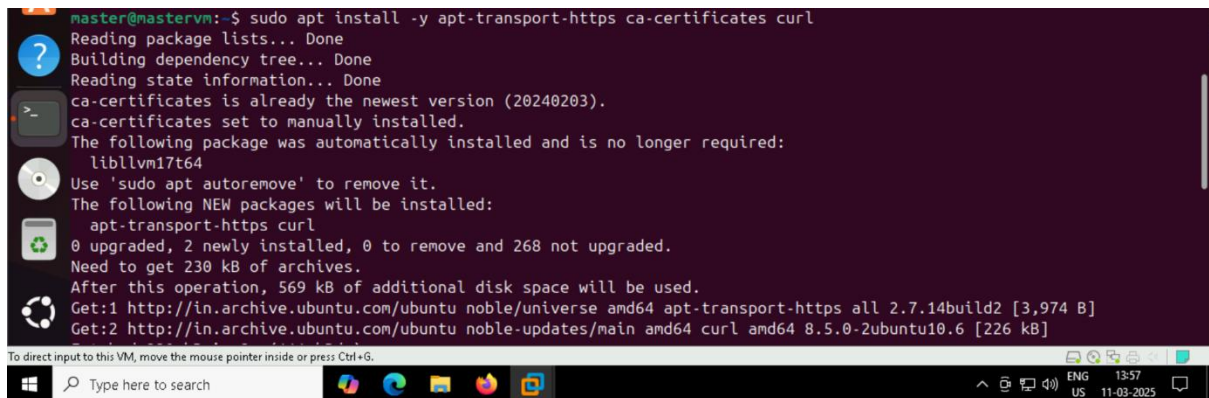
### Step 1: Install Kubernetes on Ubuntu



The screenshot shows a terminal window titled 'master@mastervm: ~' with the command 'sudo apt update' being executed. The output lists various updates from the Ubuntu noble repository, including InRelease files, Packages, Components, and Translation-en files. The terminal is running inside a VMware Workstation environment, with the window title 'Master - VMware Workstation' visible at the top.

```
master@mastervm:~$ sudo apt update
[sudo] password for master:
Hit:1 http://in.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:5 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [916 kB]
Ign:6 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:7 https://pkg.jenkins.io/debian-stable binary/ Release
Get:8 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [668 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [128 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8,964 B]
Get:13 http://in.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [754 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [719 kB]
Get:15 http://in.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [151 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:17 http://in.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1,040 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [261 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [364 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [30.1 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [5,884 B]
```

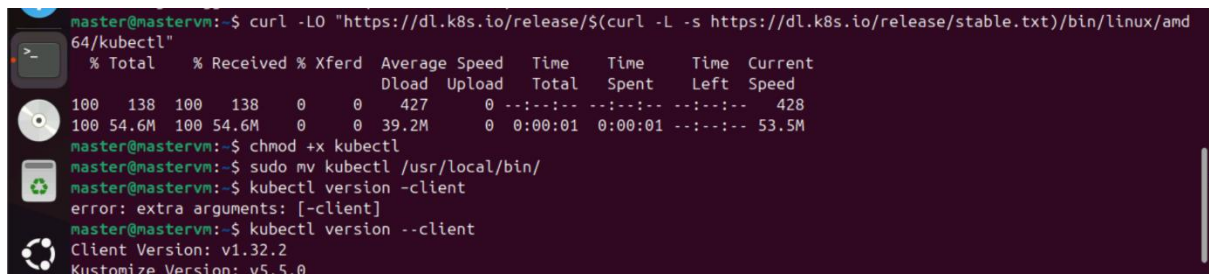
### Step 2: Install Dependencies



The screenshot shows a terminal window titled 'master@mastervm: ~' with the command 'sudo apt install -y apt-transport-https ca-certificates curl' being executed. The output shows that 'ca-certificates' is already the newest version, and 'apt-transport-https' and 'curl' are being installed. The terminal is running inside a VMware Workstation environment, with the window title 'Master - VMware Workstation' visible at the top.

```
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 268 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]
```

### Step 3: Install kubectl



The screenshot shows a terminal window titled 'master@mastervm: ~' with the command 'curl -LO "https://dl.k8s.io/release/\$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"' being executed. The output shows the download progress of kubectl. The terminal is running inside a VMware Workstation environment, with the window title 'Master - VMware Workstation' visible at the top.

```
master@mastervm:~$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 138 100 138 0 0 427 0 --:--:-- --:--:-- --:--:-- 428
100 54.6M 100 54.6M 0 0 39.2M 0 0:00:01 0:00:01 --:--:-- 53.5M
master@mastervm:~$ chmod +x kubectl
master@mastervm:~$ sudo mv kubectl /usr/local/bin/
master@mastervm:~$ kubectl version -client
error: extra arguments: [-client]
master@mastervm:~$ kubectl version --client
Client Version: v1.32.2
Kustomize Version: v5.5.0
```

## Step 4: Install Minikube

```
master@mastervm:~$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 119M 100 119M 0 0 14.1M 0 0:00:08 0:00:08 --:--:-- 18.3M
master@mastervm:~$ chmod +x minikube-linux-amd64
master@mastervm:~$ sudo mv minikube-linux-amd64 /usr/local/bin/minikube
```

## Step 5: Start Minikube and check the status

```
master@mastervm:~$ sudo usermod -aG docker $USER && newgrp docker
master@mastervm:~$ minikube start --driver=docker
minikube v1.35.0 on Ubuntu 24.04
Using the docker driver based on user configuration
Using Docker driver with root privileges
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Downloading Kubernetes v1.32.0 preload ...
> gcr.io/k8s-minikube/kicbase...: 500.31 MiB / 500.31 MiB 100.00% 16.39 M
> preloaded-images-k8s-v18-v1...: 31.91 MiB / 333.57 MiB 9.56% 74.28 KiB
Creating docker container (CPUs=2, Memory=2200MB) ...
> kubectrl.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubelet.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubeadm.sha256: 64 B / 64 B [-----] 100.00% ? p/s 0s
> kubectrl: 54.67 MiB / 54.67 MiB [-----] 100.00% 23.38 MiB p/s 2.5s
> kubeadm: 67.66 MiB / 67.66 MiB [-----] 100.00% 22.10 MiB p/s 3.3s
> kubelet: 73.81 MiB / 73.81 MiB [-----] 100.00% 17.65 MiB p/s 4.4s

■ Generating certificates and keys ...
■ Booting up control plane ...
■ Configuring RBAC rules ...
Configuring bridge CNI (Container Networking Interface) ...
Verifying Kubernetes components...
■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
Enabled addons: default-storageclass, storage-provisioner
Done! kubectrl is now configured to use "minikube" cluster and "default" namespace by default
master@mastervm:~$ kubectrl cluster-info
Kubernetes control plane is running at https://192.168.49.2:8443
CoreDNS is running at https://192.168.49.2:8443/api/v1/namespaces/kube-system/services/kube-dns/dns/proxy
```

## Step 6: Create folder and create a app.py flask app, dockerfile, requirements.txt

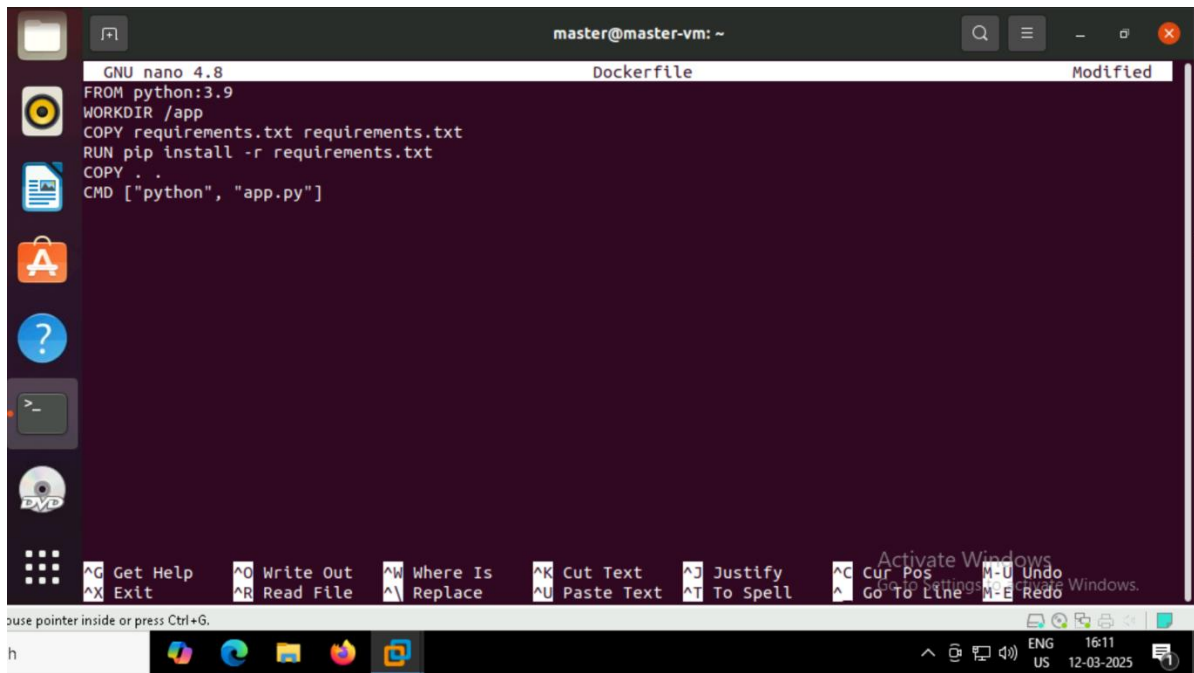
```
master@mastervm:~$ mkdir flask-ci-cd
master@mastervm:~$ cd flask-ci-cd
master@mastervm:~/flask-ci-cd$ nano app.py
master@mastervm:~/flask-ci-cd$ nano Dockerfile
master@mastervm:~/flask-ci-cd$ nano requirements.txt
```

```
GNU nano 4.8 app.py Modified
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "Hello, World!"

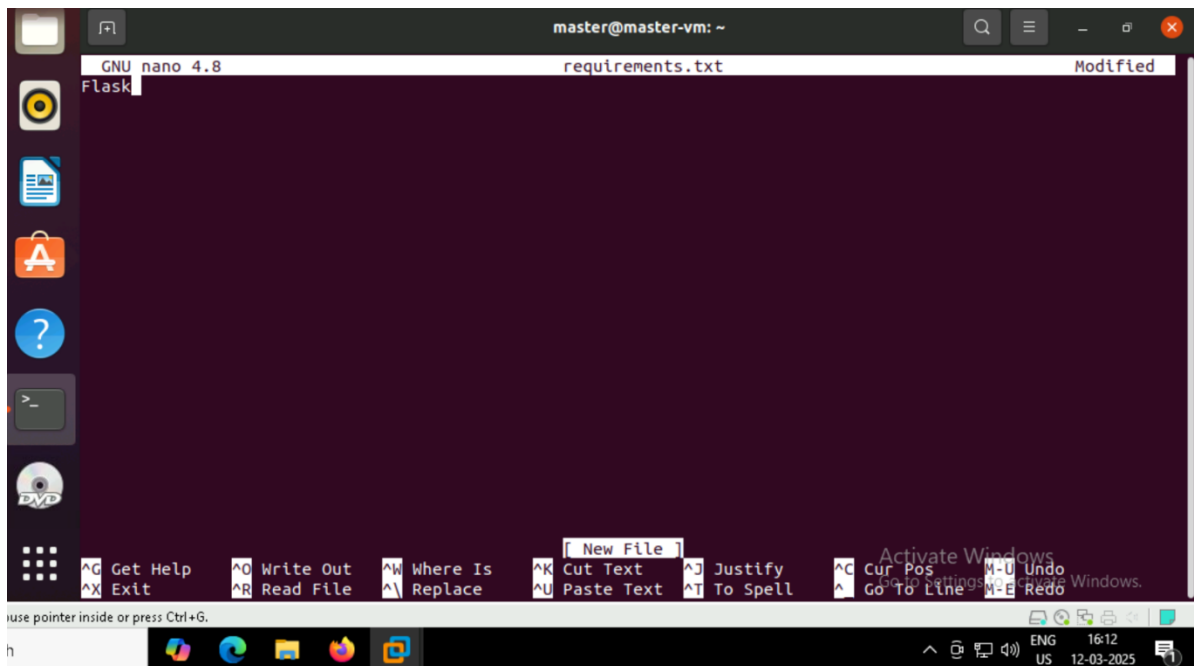
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=5000)
```



The screenshot shows a terminal window titled "master@master-vm: ~" with the nano 4.8 editor open to a file named "Dockerfile". The file content is as follows:

```
FROM python:3.9
WORKDIR /app
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
COPY . .
CMD ["python", "app.py"]
```

The terminal window includes a sidebar with application icons and a bottom status bar with system information and a Windows activation watermark.

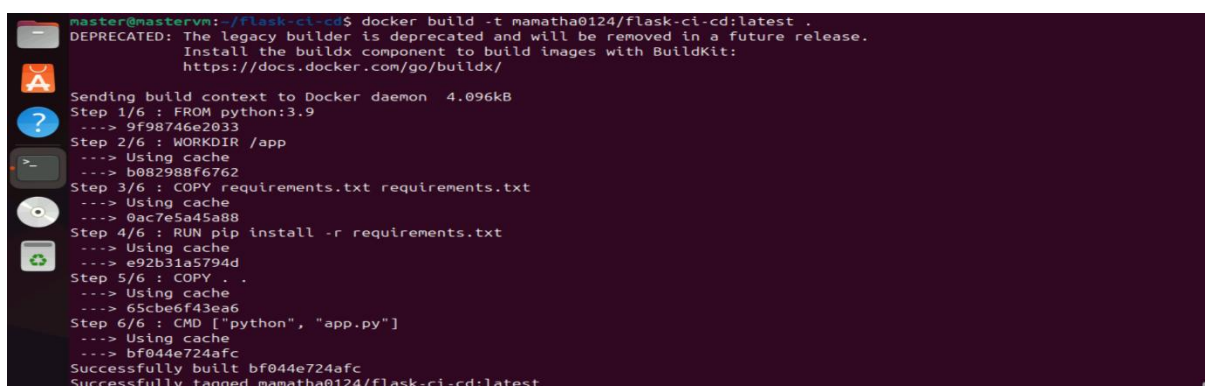


The screenshot shows the same terminal window with the nano 4.8 editor now open to a file named "requirements.txt". The file content is:

```
Flask
```

The terminal window layout is consistent with the previous screenshot, showing the same sidebar and status bar.

## Step 7: Build and Push Docker Image



The screenshot shows a terminal window where the command `docker build -t mamatha0124/flask-ci-cd:latest .` has been executed. The output shows the build steps and their results:

```
master@mastervm:~/flask-ci-cd$ docker build -t mamatha0124/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
--> 9f98746e2033
Step 2/6 : WORKDIR /app
--> Using cache
--> b082988f6762
Step 3/6 : COPY requirements.txt requirements.txt
--> Using cache
--> 0ac7e5a45a88
Step 4/6 : RUN pip install -r requirements.txt
--> Using cache
--> e92b31a5794d
Step 5/6 : COPY . .
--> Using cache
--> 65cbe6f43ea6
Step 6/6 : CMD ["python", "app.py"]
--> Using cache
--> bf044e724afc
Successfully built bf044e724afc
Successfully tagged mamatha0124/flask-ci-cd:latest
```

The terminal window shows the same sidebar and status bar as the previous screenshots.



```
master@mastervm:~/flask-ci-cd$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/master/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
master@mastervm:~/flask-ci-cd$ docker push mamatha0124/flask-ci-cd:latest
The push refers to repository [docker.io/mamatha0124/flask-ci-cd]
41ea8da4070b: Pushed
84667c24123c: Pushed
c7695642d7ef: Pushed
fe40e043d454: Pushed
01db3e67097a: Mounted from kpk25/flask-ci-cd
e49d0c94aa2a: Mounted from kpk25/flask-ci-cd
1c86760c5c93: Mounted from kpk25/flask-ci-cd
4b017a36fd9c: Mounted from kpk25/flask-ci-cd
20a9b386e10e: Mounted from kpk25/flask-ci-cd
f8217d7865d2: Mounted from kpk25/flask-ci-cd
01c9a2a5f237: Mounted from kpk25/flask-ci-cd
latest: digest: sha256:a6d24844a6d13d3007e520671e314cca996716abc16d5c6b5185ac630584434 size: 2627
```

## Step 8: Connect Kubernetes to Docker

```
master@mastervm:~/flask-ci-cd$ kubectl create secret docker-registry docker-hub-secret \
--docker-server=https://index.docker.io/v1/ \
--docker-username=mamatha0124 \
--docker-password='PinkyManu@24' \
--docker-email=289247@ust.com
```

## Step 9: Create k8s-deployment.yaml

```
GNU nano 7.2 k8s-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-app
spec:
  replicas: 2
  selector:
    matchLabels:
      app: flask-app
  template:
    metadata:
      labels:
        app: flask-app
    spec:
      imagePullSecrets:
        - name: docker-hub-secret
      containers:
        - name: flask-app
          image: mamatha0124/flask-ci-cd:latest # Updated to match your correct Docker Hub username
          ports:
            - containerPort: 5000
---
apiVersion: v1
kind: Service
metadata:
```

## Step 10 : Apply the deployment and Check if the pods are running

```
master@mastervm:~/flask-ci-cd$ kubectl apply -f k8s-deployment.yaml
deployment.apps/flask-app created
service/flask-service created
master@mastervm:~/flask-ci-cd$ kubectl get pods
NAME READY STATUS RESTARTS AGE
flask-app-58b8cc8758-bbct4 0/1 ContainerCreating 0 22s
flask-app-58b8cc8758-jdv4w 0/1 ContainerCreating 0 22s
master@mastervm:~/flask-ci-cd$ docker pull flask-ci-cd:latest
Error response from daemon: pull access denied for flask-ci-cd, repository does not exist or may require 'docker login':
denied: requested access to the resource is denied
master@mastervm:~/flask-ci-cd$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/master/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
master@mastervm:~/flask-ci-cd$ docker pull flask-ci-cd:latest
Error response from daemon: pull access denied for flask-ci-cd, repository does not exist or may require 'docker login':
denied: requested access to the resource is denied
master@mastervm:~/flask-ci-cd$ kubectl get pods
NAME READY STATUS RESTARTS AGE
flask-app-58b8cc8758-bbct4 1/1 Running 0 114s
flask-app-58b8cc8758-jdv4w 1/1 Running 0 114s
```