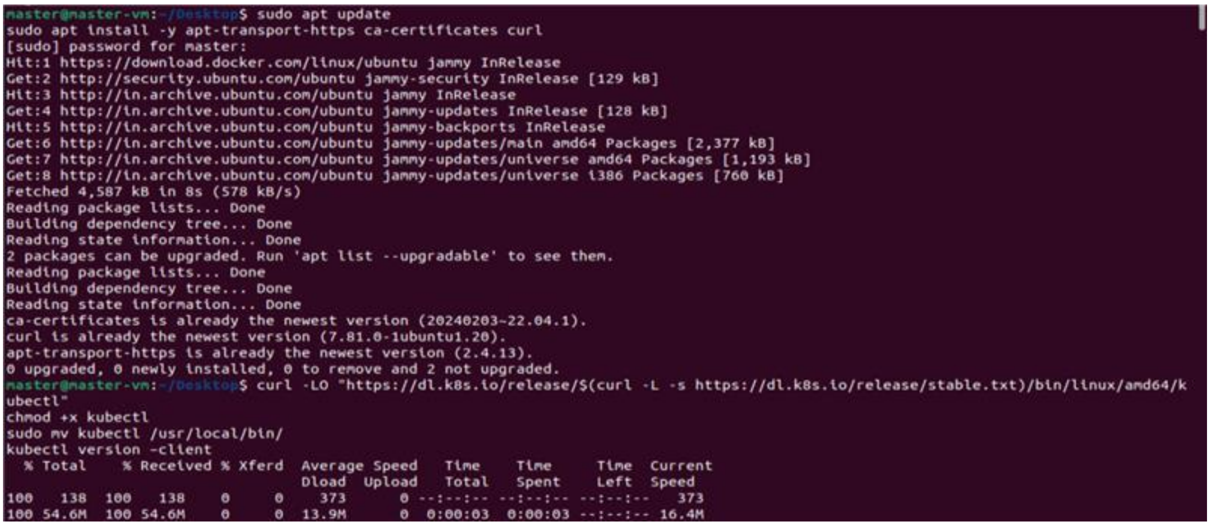
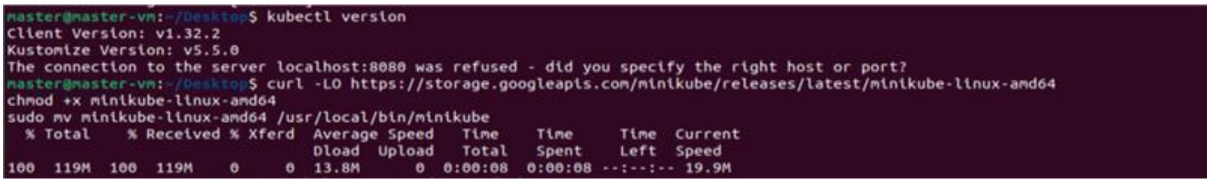
Docker project 3

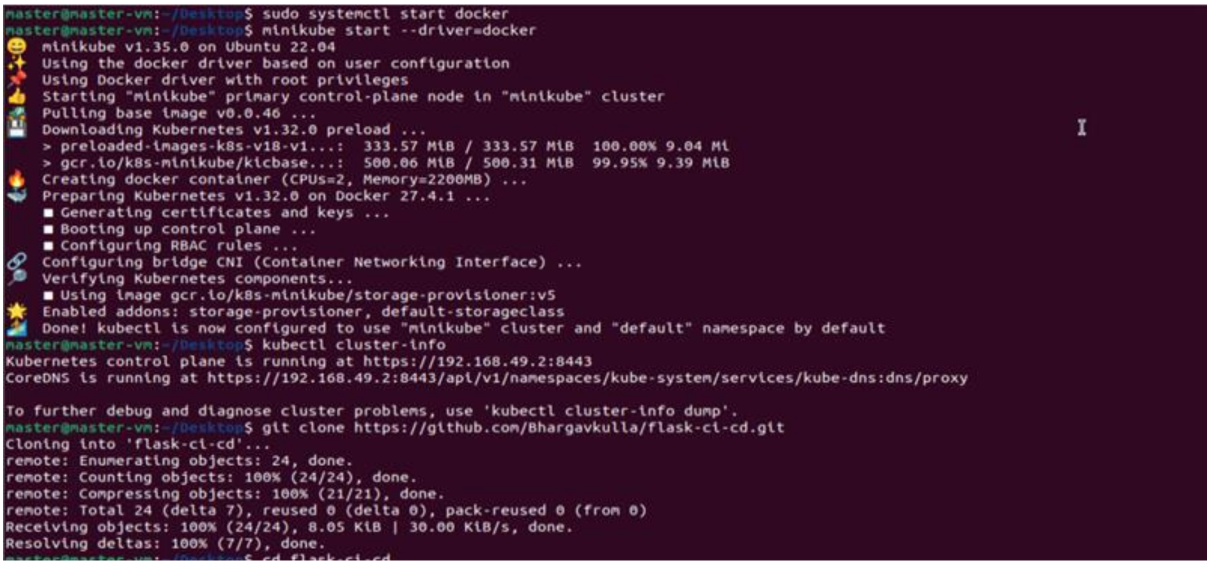
Dockerized CI/CD pipeline using Jenkins, Dockers, Kubernetes



```
master@master-vm:~/Desktop$ sudo apt update
sudo apt install -y apt-transport-https ca-certificates curl
[sudo] password for master:
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Get:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2,377 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,193 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates/universe i386 Packages [760 kB]
Fetched 4,587 kB in 8s (578 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
2 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20240203~22.04.1).
curl is already the newest version (7.81.0-1ubuntu1.20).
apt-transport-https is already the newest version (2.4.13).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
master@master-vm:~/Desktop$ curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/k
ubectl"
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
100   138  100   138    0     0    373      0 --:--:-- --:--:-- --:--:--   373
100 54.6M  100 54.6M    0     0  13.9M      0  0:00:03  0:00:03 --:--:-- 16.4M
```

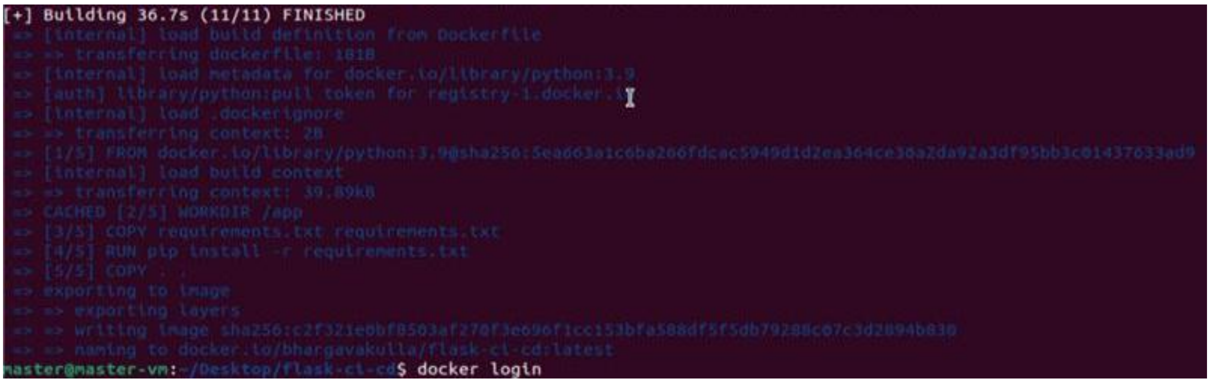


```
master@master-vm:~/Desktop$ kubectl version
Client Version: v1.32.2
Kustomize Version: v5.5.0
The connection to the server localhost:8080 was refused - did you specify the right host or port?
master@master-vm:~/Desktop$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
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
100  119M  100  119M    0     0  13.8M      0  0:00:08  0:00:08 --:--:-- 19.9M
```



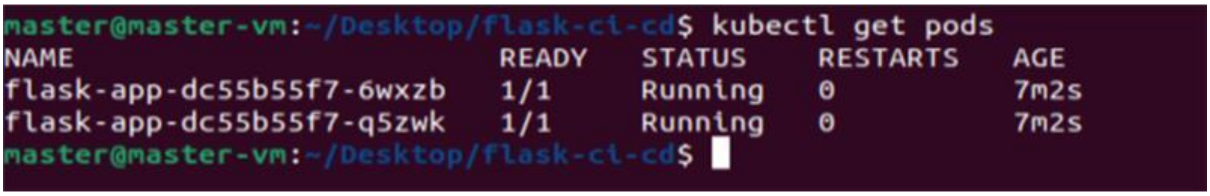
```
master@master-vm:~/Desktop$ sudo systemctl start docker
master@master-vm:~/Desktop$ minikube start --driver=docker
😄  minikube v1.35.0 on Ubuntu 22.04
✨  Using the docker driver based on user configuration
📌  Using Docker driver with root privileges
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.46 ...
💾  Downloading Kubernetes v1.32.0 preload ...
    > preloaded-images-k8s-v18-v1...:  333.57 MiB / 333.57 MiB  100.00% 9.04 Mi
    > gcr.io/k8s-minikube/kicbase...:  500.06 MiB / 500.31 MiB  99.95% 9.39 MiB
🔥  Creating docker container (CPUs=2, Memory=2200MB) ...
🐳  Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
    ■ 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: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
master@master-vm:~/Desktop$ kubectl 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

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
master@master-vm:~/Desktop$ git clone https://github.com/Bhargavkulla/flask-ci-cd.git
Cloning into 'flask-ci-cd'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 24 (delta 7), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (24/24), 8.05 KiB | 30.00 KiB/s, done.
Resolving deltas: 100% (7/7), done.
master@master-vm:~/Desktop$ cd flask-ci-cd
```



```
[+] Building 36.7s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 181B
 => [internal] load metadata for docker.io/library/python:3.9
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/5] FROM docker.io/library/python:3.9@sha256:5ea663a1c6ba266fdcac5949d1d2ea364ce3da2da92a3df95bb3c01437633ad9
 => [internal] load build context
 => => transferring context: 39.89kB
 => CACHED [2/5] WORKDIR /app
 => [3/5] COPY requirements.txt requirements.txt
 => [4/5] RUN pip install -r requirements.txt
 => [5/5] COPY . .
 => exporting to image
 => => exporting layers
 => => writing image sha256:c2f321e0bf8503af270f3e696f1cc153bfa588df5f5db79288c07c3d2894b030
 => => naming to docker.io/bhargavakulla/flask-ci-cd:latest
master@master-vm:~/Desktop/flask-ci-cd$ docker login
```

OUTPUT



```
master@master-vm:~/Desktop/flask-ci-cd$ kubectl get pods
NAME                         READY   STATUS    RESTARTS   AGE
flask-app-dc55b55f7-6wxzb    1/1     Running   0          7m2s
flask-app-dc55b55f7-q5zwk    1/1     Running   0          7m2s
master@master-vm:~/Desktop/flask-ci-cd$
```