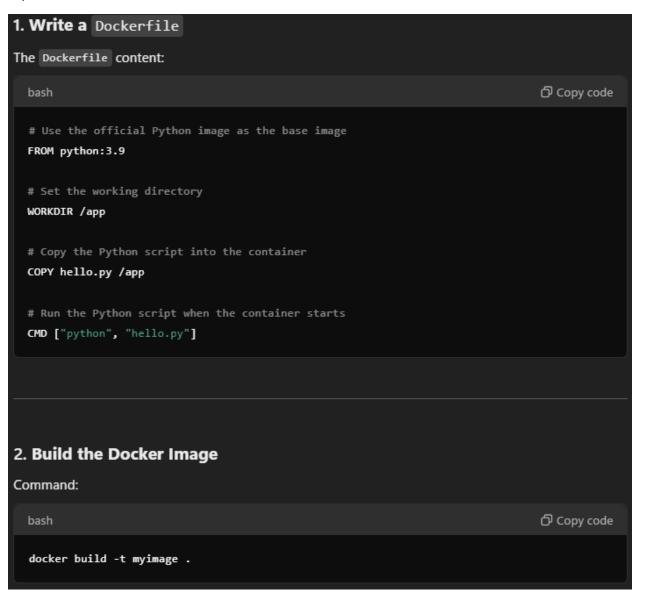
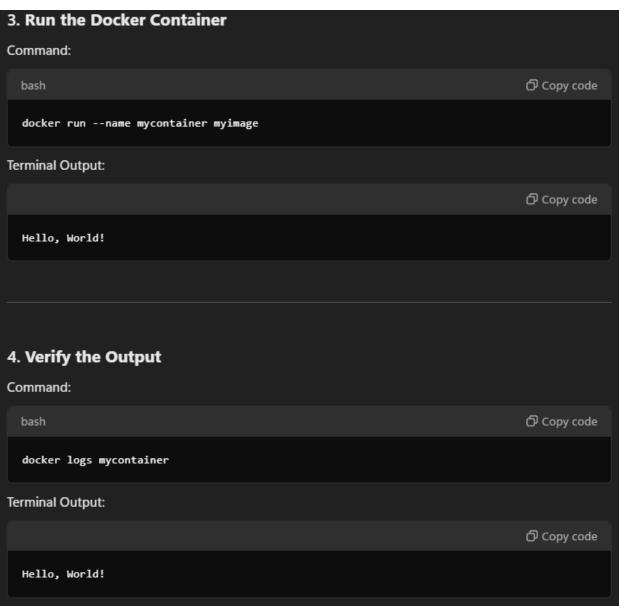


```
7. docker pull ubuntu:16.04
     bash
                                                                                 (7) Copy code
     16.04: Pulling from library/ubuntu
     abc123: Already exists
     def456: Pull complete
     Digest: sha256:abcdef1234567890
     Status: Downloaded newer image for ubuntu:16.04
   This pulls the Ubuntu 16.04 image.
8. docker push myimage
     bash
                                                                                 Copy code
     The push refers to repository [docker.io/library/myimage]
     abc123: Layer already exists
     def456: Layer already exists
     myimage: Digest: sha256:abcdef1234567890 size: 1234
```

Exp 7

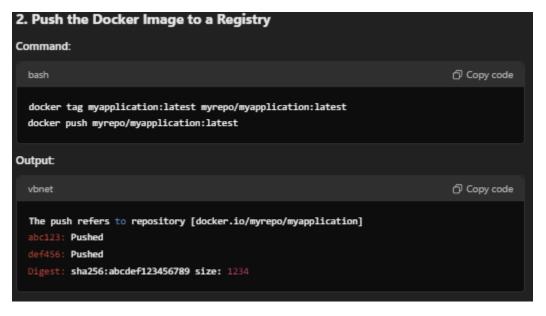


```
Sending build context to Docker daemon 3.072kB
Step 1/4: FROM python:3.9
---> abc123456789
Step 2/4 : WORKDIR /app
 ---> Running in 123abc456def
Removing intermediate container 123abc456def
 ---> abc123456789
Step 3/4 : COPY hello.py /app
 ---> Using cache
 ---> abc123456789
Step 4/4 : CMD ["python", "hello.py"]
 ---> Running in 123abc456def
Removing intermediate container 123abc456def
---> abc123456789
Successfully built abc123456789
Successfully tagged myimage:latest
```

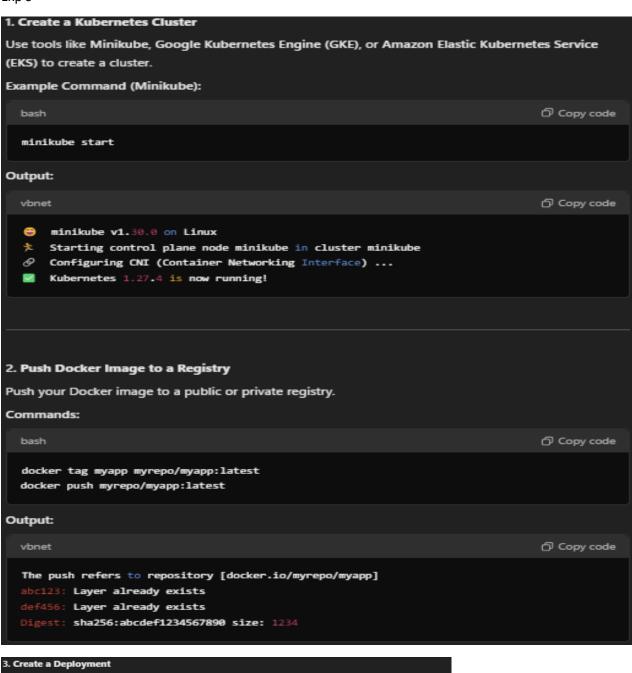


D

```
1. Build a Docker Image
Command:
  bash
                                                                                Copy code
  docker build -t myapplication .
Output:
                                                                                Copy code
  vbnet
  Sending build context to Docker daemon 3.072kB
  Step 1/3 : FROM python:3.9
   ---> abc123456789
  Step 2/3 : COPY app/ /app
   ---> Using cache
  Step 3/3 : CMD ["python", "/app/app.py"]
   ---> Running in 123abc456def
  Successfully built abc123456789
  Successfully tagged myapplication:latest
```



3. Deploy the Image to Kubernetes Create a Deployment YAML file (deployment.yaml): yaml apiVersion: apps/v1 name: myapplication-deployment app: myapplication app: myapplication - name: myapplication image: myrepo/myapplication:latest Apply the Deployment: (7) Copy code bash kubectl apply -f deployment.yaml Output: (7) Copy code bash deployment.apps/myapplication-deployment created 4. Monitor the Deployment Check the Deployment Status: Copy code bash kubectl get deployments Output: mathematica (7) Copy code Application started successfully

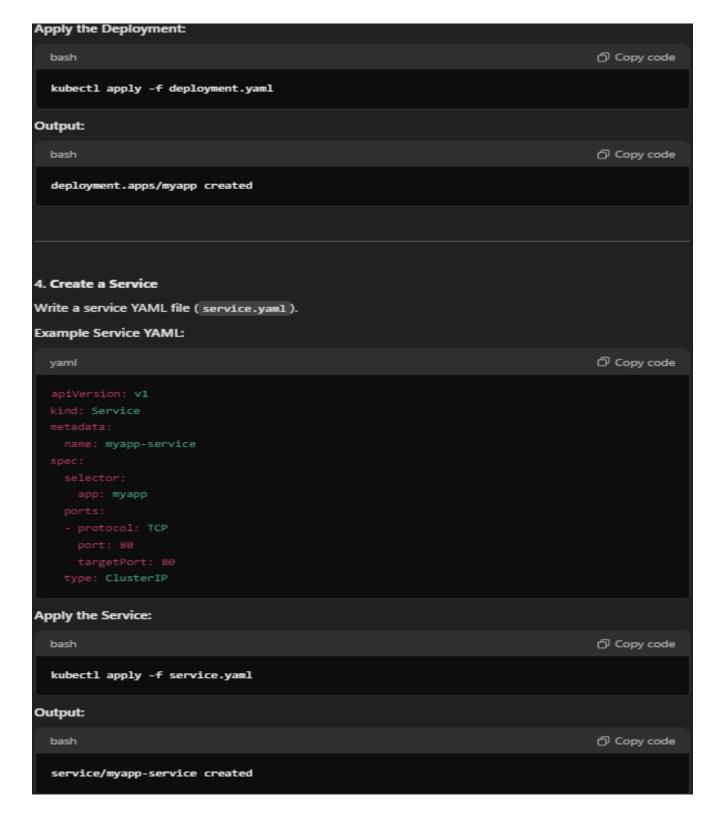


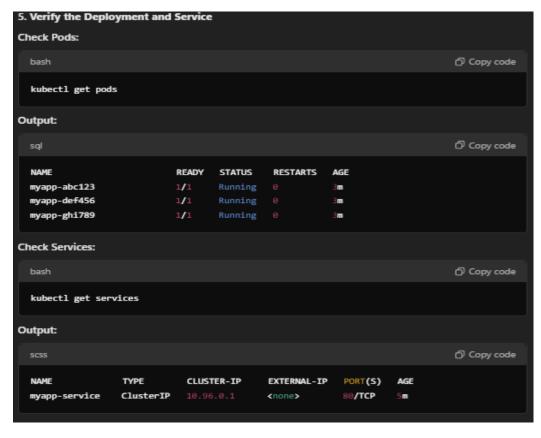
```
3. Create a Deployment
Write a deployment YAML file (deployment.yaml).

Example Deployment YAML:

yaml

apiVersion: apps/v1
kind: Deployment
metadata:
name: myapp
spec:
replicas: 3
selector:
matchLabels:
app: myapp
template:
metadata:
labels:
app: myapp
spec:
containers:
- name: myapp
image: myrepo/myapp:latest
ports:
- containerPort: 80
```





Exp 10



