Program 4

Prerequisites:

Docker (you can check docker installation using command docker version)

minikube (you can check minikube installation using command minikube version)

kubectl (you can check kubectl installation using command kubectl version - - client)

steps

1. open cmd and run following command
   minikube start --driver=docker
   minikube status
   kubeectl get nodes

2. Make one folder with name 'prgm4' on Desktop and open in vscode

3. **Create** `app.py`

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "Hello from App 1!! Kubernetes is managing this."

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

4. **Create** `requirements.txt`

```
flask==3.0.0
```

5. **Create** `Dockerfile`

```
FROM python:3.12-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY app.py .
EXPOSE 5000
CMD ["python", "app.py"]
```

6. Build Docker image
   docker build -t prgm4 .

7. Tag Docker tag
   docker tag prgm4 rahulkrchaudhary12/prgm4

8. Push Docker Image
   docker push rahulkrchaudhary12/prgm4

9. create deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: hw-deployment # Name of the deployment
spec:
  replicas: 2 # Number of replicas for the deployment
  selector:
    matchLabels:
      app: prgm4 # This label is used to match the pods
  template:
    metadata:
      labels:
        app: prgm4 # This label is used to identify the pods
    spec:
      containers:
      - name: prgm4 # Name of the container
        image: rahulkrchaudhary12/prgm4:latest # Image to be used for the con
```

```
        ports:
        - containerPort: 5000
```

10. Create service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: hw-service # Name of the service
spec:
  type: NodePort # Type of the service
  selector:
    app: prgm4 # This label is used to match the pods
  ports:
    - port: 5000 # Port on which the service will be exposed
      targetPort: 5000 # Port on which the container is listening
```

11. Apply Deployment & Service(run following command one by one in terminal)

    kubectl apply -f deployment.yaml
    kubectl apply -f service.yaml
    kubectl get pods
    kubectl get svc

12. Access the App in Browser

    minikube service hw-service

13. **Scale to 3 Replicas**:
    kubectl scale deployment hw-deployment --replicas=3

14. **Verify**:

    kubectl get deployment
    kubectl get pods

15. Common command
    minikube status

minikube stop
kubectl delete all --all

minikube service list