Deploying a 12-Factor App on Kubernetes with Minikube

For this task, I set up everything locally on my Mac under this folder:
/Users/gesinelinn/Documents/Masterstudium/Essex/Cloud Operation/

Inside, I created these files:
Dockerfile
requirements.txt
app.py
k8s/
  deployment.yaml
  config.yaml

Step 1: Start Minikube

I already had Docker and kubectl installed. Minikube initially failed to start because of low resources, so I restarted it with more memory:

```
minikube start --memory 4096 --cpus 2
```

Then I switched my shell to use Minikube's Docker daemon (I forgot this at first and had to rebuild):

```
eval $(minikube docker-env)
```

Step 2: Build a Container

I wrote a small Flask app in app.py that reads environment variables for APP_NAME and shows them on / and /healthz. My Dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 8080
CMD ["gunicorn", "--bind", "0.0.0.0:8080", "app:app"]
```

Built the image:

```
docker build -t my-twelvefactor-app:v1 .
```

Step 3: Kubernetes Config and Deployment

I kept configs simple. In k8s/config.yaml:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  APP_NAME: "kube-test-app"
  DEBUG: "true"
```

And my deployment in k8s/deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: kube-test-web
spec:
  replicas: 2  # TODO: adjust later if needed
  selector:
    matchLabels:
      app: kube-test-web
  template:
    metadata:
      labels:
        app: kube-test-web
    spec:
      containers:
      - name: web
        image: my-twelvefactor-app:v1
        ports:
        - containerPort: 8080
        envFrom:
        - configMapRef:
            name: app-config
---
apiVersion: v1
kind: Service
metadata:
  name: kube-test-service
spec:
  type: NodePort
  selector:
    app: kube-test-web
  ports:
  - port: 80
    targetPort: 8080
```

Applied everything:

kubectl apply -f k8s/

First time I had a typo in the config, so I had to fix and re-apply. Once fixed:

```
kubectl get pods
# NAME                          READY   STATUS    RESTARTS   AGE
# kube-test-web-7d5f987df9-bkql   1/1     Running   0          30s
```

Step 4: Access and Logs

Opened the app:

```
minikube service kube-test-service --url
# http://127.0.0.1:50038
```

Checked logs (12-Factor principle: logs go to stdout):

```
kubectl logs -l app=kube-test-web -f
```

Reflection

My main challenges were forgetting to switch Docker to Minikube and a YAML typo. Following the 12-Factor approach helped me keep the app stateless, externalize config, and make it easy to scale by changing replicas. This setup gave me a real sense of how Kubernetes manages deployments and scaling.