**✅ Kubernetes Files**

**📂 kubernetes/namespace.yml**

apiVersion: v1

kind: Namespace

metadata:

name: my-application

**📂 kubernetes/mysql/pv-mysql.yml**

apiVersion: v1

kind: PersistentVolume

metadata:

name: mysql-pv

spec:

capacity:

storage: 1Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/mnt/data"

**📂 kubernetes/mysql/pvc-mysql.yml**

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: mysql-pvc

spec:

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 1Gi

**📂 kubernetes/mysql/mysql-deployment.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: mysql

namespace: my-application

spec:

selector:

matchLabels:

app: mysql

replicas: 1

template:

metadata:

labels:

app: mysql

spec:

containers:

- name: mysql

image: mysql:8.0

env:

- name: MYSQL\_ROOT\_PASSWORD

value: root

- name: MYSQL\_DATABASE

value: quantumsoft

- name: MYSQL\_USER

value: root

- name: MYSQL\_PASSWORD

value: root

ports:

- containerPort: 3306

volumeMounts:

- name: mysql-storage

mountPath: /var/lib/mysql

volumes:

- name: mysql-storage

persistentVolumeClaim:

claimName: mysql-pvc

**📂 kubernetes/mysql/mysql-service.yml**

apiVersion: v1

kind: Service

metadata:

name: mysql-service

namespace: my-application

spec:

selector:

app: mysql

ports:

- protocol: TCP

port: 3306

targetPort: 3306

clusterIP: None

**✅ Backend Files**

**📂 backend/Dockerfile**

FROM openjdk:21

WORKDIR /app

COPY target/backend.jar /app/backend.jar

EXPOSE 8080

ENTRYPOINT ["java", "-jar", "backend.jar"]

**📂 kubernetes/backend/backend-deployment.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: backend

namespace: my-application

spec:

replicas: 1

selector:

matchLabels:

app: backend

template:

metadata:

labels:

app: backend

spec:

containers:

- name: backend

image: backend:latest

ports:

- containerPort: 8080

env:

- name: SPRING\_DATASOURCE\_URL

value: jdbc:mysql://mysql-service:3306/mydb

- name: SPRING\_DATASOURCE\_USERNAME

value: user

- name: SPRING\_DATASOURCE\_PASSWORD

value: password

**📂 kubernetes/backend/backend-service.yml**

apiVersion: v1

kind: Service

metadata:

name: backend-service

namespace: my-application

spec:

selector:

app: backend

ports:

- protocol: TCP

port: 8080

targetPort: 8080

type: ClusterIP

**✅ Frontend Files**

**📂 frontend/Dockerfile**

FROM nginx:alpine

COPY dist/ /usr/share/nginx/html

EXPOSE 80

**📂 kubernetes/frontend/frontend-deployment.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

name: frontend

namespace: my-application

spec:

replicas: 1

selector:

matchLabels:

app: frontend

template:

metadata:

labels:

app: frontend

spec:

containers:

- name: frontend

image: frontend:latest

ports:

- containerPort: 80

**📂 kubernetes/frontend/frontend-service.yml**

apiVersion: v1

kind: Service

metadata:

name: frontend-service

namespace: my-application

spec:

selector:

app: frontend

ports:

- protocol: TCP

port: 80

targetPort: 80

type: NodePort