Name: Shoaib Akhtar

Roll No: 20P-0147

Section: BS-CS (7A)

Step 1: Persistent Volumes and Persistent Volume Claims

Persistent volumes are used to provide storage that persists beyond the life cycle of an individual pod. Persistent volume claims are requests for storage by pods.

//1.1 Persistent Volume (persistent-volume.yaml)

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: frontend-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /data/frontend

---

apiVersion: v1
kind: PersistentVolume
metadata:
  name: backend-pv
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: /data/backend
```

//.2 Persistent Volume Claim (persistent-volume-claim.yaml)

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: frontend-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi

---

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: backend-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
```

Step 2: Deployments

Deployments manage the creation and scaling of pods. Here, we define deployments for the front-end and back-end services.

// 2.1 Frontend Deployment (frontend-deployment.yaml)

```
apiVersion: apps/v1

kind: Deployment

metadata:

  name: frontend-deployment

spec:

  replicas: 2

  selector:

    matchLabels:

      app: frontend

  template:

    metadata:
```

```
    labels:
      app: frontend
  spec:
    containers:
    - name: frontend
      image: your-frontend-image:tag
      ports:
      - containerPort: 80
      volumeMounts:
      - name: frontend-storage
        mountPath: /app/data
```

2.2 Backend Deployment (backend-deployment.yaml)

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
      - name: backend
        image: your-backend-image:tag
```

```
ports:
- containerPort: 3306
volumeMounts:
- name: backend-storage
  mountPath: /var/lib/mysql
```

Step 3: Apply Configurations

Apply the configurations to create persistent volumes, persistent volume claims, and deployments.

```
kubectl apply -f persistent-volume.yaml
kubectl apply -f persistent-volume-claim.yaml
kubectl apply -f frontend-deployment.yaml
kubectl apply -f backend-deployment.yaml
```

Step 4: Verify Deployments

Check the status of the deployments, services, and pods.

```
kubectl get pv
kubectl get pvc
kubectl get deployments
kubectl get pods
```