Lab 9: Persistent Storage

Training Goals Covered:

- Attach storage to a pod and explore data mapping.
- Use declarative configuration (YAML files).

Steps:

1. Create a Persistent Volume (2GB) mapped to the host path “/mnt/data”.

Shell

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: my-pv
spec:
  storageClassName: manual
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/data/"
```

Shell

```
k apply -f pv.yaml
```

2. Create a Persistent Volume Claim (1GB).

None

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

Shell

```
k apply -f pvc.yaml
```

Shell

```
k get pv
NAME    CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM
STORAGECLASS   VOLUMEATTRIBUTESCLASS   REASON   AGE
my-pv   2Gi        RWO            Retain           Bound    default/my-pvc
manual         <unset>                          3m17s
```

Shell

```
k get pvc
NAME     STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS
VOLUMEATTRIBUTESCLASS   AGE
my-pvc   Bound    my-pv    2Gi        RWO            manual         <unset>
2m53s
```

3. Deploy a Pod using the Persistent Storage claimed. Use nginx image, container port 80 and "/usr/share/nginx/html" as mount path.

Shell

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
```

```
spec:
  volumes:
    - name: my-pv-volume
      persistentVolumeClaim:
        claimName: my-pvc
  containers:
    - name: nginx
      image: nginx
      ports:
        - containerPort: 80
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: my-pv-volume
```

Shell

```
k apply -f pod.yaml
```

Shell

```
k describe po my-pod | grep -A1 "Mounts:"
    Mounts:
      /usr/share/nginx/html from my-pv-volume (rw)
```

4. Create a sample file with a random text in the mount path refereed.

Shell

```
kubectl exec -it my-pod -- /bin/bash -c "echo 'Hello from Persistent Storage' >
/usr/share/nginx/html/file.txt"
```

Shell

```
kubectl exec -it my-pod -- cat /usr/share/nginx/html/file.txt
```

5. Check the content of the file in the host path.

Shell

```
k get po -o wide
NAME     READY   STATUS    RESTARTS   AGE   IP            NODE
NOMINATED NODE   READINESS GATES
my-pod   1/1     Running   0          33s   10.244.1.29   minikube-m02   <none>
<none>
```

Access to the node where the pod was scheduled:

Shell

```
minikube ssh --node minikube-m02
docker@minikube-m02:~$ cat /data/file.txt
Hello from Persistent Storage
docker@minikube-m02:~$
```

6. Delete the pod created on step 3.

Shell

```
k delete po my-pod
```

7. Recreate the pod. Check the file is accessible in the pod.

Shell

```
k apply -f pod.yaml
```

Shell

```
k exec -it my-pod -- /bin/bash
root@my-pod:/# cat /usr/share/nginx/html/file.txt
Hello from Persistent Storage
root@my-pod:/#
```