



Task 2: Provision and Mount Storage

Objective: Provision a PersistentVolumeClaim (PVC), mount it inside the container, and verify persistent storage.



Steps Taken

1. Deploy new deployment with volume mounted
 2. Defined a PVC in `pvc.yaml` requesting 1Gi of storage.
 3. Applied the PVC manifest.
 4. Verified the PVC reached **Bound** status.
 5. Updated the Deployment to mount the volume inside the container at `/app/data`.
 6. Verified volume mounting via `kubectl describe pod`.
 7. Wrote a file inside the mounted volume using `kubectl exec`, then read it to verify persistence.
-



Screenshots

- **Screenshot 1:** `kubectl apply -f deployment.yaml`
(Deploy new deployment with volume mounted)

```
❯ ❯ ~/Desktop/k8s/nodeJS--as-k8s/manifests > ❯ main !3 ?2 .
❯ > kubectl apply -f deployment.yaml
deployment.apps/nodejs-app created
```

- **Screenshot 2:** `kubectl apply -f pvc.yaml`
(PVC creation)

```
❯ ❯ ~/Desktop/k8s/nodeJS--as-k8s/manifests > ❯ main !3 ?2 .
❯ > kubectl apply -f pvc.yaml
persistentvolumeclaim/nodejs-app-pvc unchanged
```

- **Screenshot 3:** `kubectl get pvc`
(Shows that PVC is in Bound state)

```

~/Desktop/k8s/nodeJS--as-k8s/manifests > main !3 ?2
> kubectl get pvc

```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	VOLUMEATTRIBUTESCLASS	AGE
nodejs-app-pvc	Bound	pvc-c113a8da-2f4e-4889-a7a9-aa94f1038dab	1Gi	RWO	standard	<unset>	5m16s

- **Screenshot 4:** `kubectl describe pod <pod-name>`
(Verifies volume is mounted at /app/data)

```

Environment:      <none>
Mounts:
  /app/data from nodejs-storage (rw)
  /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-p98nc (ro)
Conditions:

```

- **Screenshot 5:** `kubectl exec` and file operations
Commands:

```

kubectl exec -it <pod-name> -- sh
echo "hello" > /app/data/test.txt
cat /app/data/test.txt

```

```

~/Desktop/k8s/nodeJS--as-k8s/manifests > main !3 ?2
> kubectl exec -it pods/nodejs-app-77dcb56dbc-8xfhq -- sh

/app # echo "hello" > /app/data/test.txt
/app # cat /app/data/test.txt
hello
/app # exit

~/Desktop/k8s/nodeJS--as-k8s/manifests > main !3 ?2
> kubectl exec -it pods/nodejs-app-77dcb56dbc-99kb5 -- sh
/app # cat /app/data/test.txt
hello
/app #

```

- **Screenshot 6 (optional):** Recreate Pod and check file again
(Proves data persisted across Pod restarts)

```
❏ > ❏ ~/Desktop/k8s/nodeJS--as-k8s/manifests > ❏ main !3 ?2 .....
❏ > kubectl delete pod nodejs-app-77dcb56dbc-8xfhq nodejs-app-77dcb56dbc-99kb5
pod "nodejs-app-77dcb56dbc-8xfhq" deleted
pod "nodejs-app-77dcb56dbc-99kb5" deleted

❏ > ❏ ~/Desktop/k8s/nodeJS--as-k8s/manifests > ❏ main !3 ?2 .....
❏ > kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nodejs-app-77dcb56dbc-4s4m2         1/1     Running   0           52s
nodejs-app-77dcb56dbc-9vfj4         1/1     Running   0           52s
```

✅ Outcome

The application successfully mounted the persistent volume. Files written to the volume remained intact even after restarting the Pod, proving that persistent storage works correctly.