Large Scale Computing

Lab 6

Adrian Madej

1. Short description of running the application

As part of the project, a web application was deployed in a local Kubernetes cluster using kind. The application includes the following components:

- An NFS server with a dynamic provisioner, installed via Helm,
- A PersistentVolumeClaim with ReadWriteMany access mode,
- A Pod running an nginx HTTP server that mounts the NFS volume,
- A Kubernetes Job that writes a sample index.html file to the shared volume,
- A NodePort Service that exposes the HTTP server for external access.

The entire configuration and commands used to deploy the application are included in the GitHub repository - [Large-Scale-Computing/lab_6 at main · ltmollo/Large-Scale-Computing](#) .

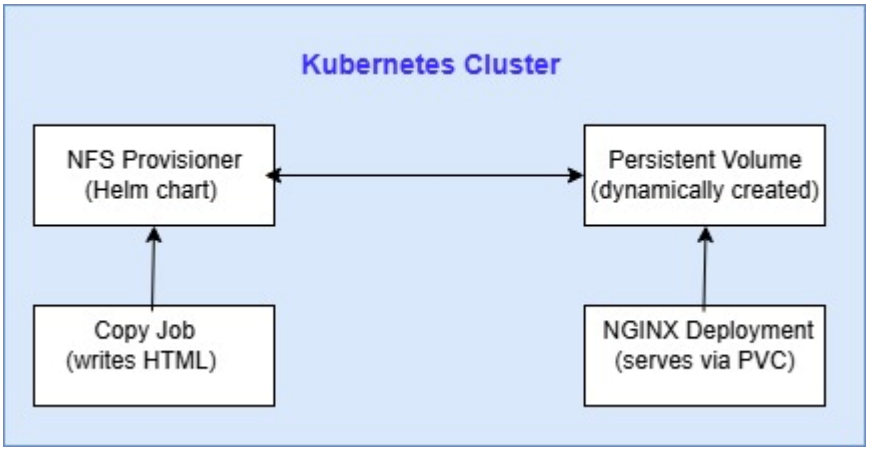
A screenshot of a web page:

Example outputs:

```
PS C:\Users\ltmol\Desktop\studia\lsc\lab_6> kubectl get pods
NAME                                     READY   STATUS    RESTARTS   AGE
nfs-server-nfs-server-provisioner-0      1/1     Running   0          73s
PS C:\Users\ltmol\Desktop\studia\lsc\lab_6> kubectl get storageclass
NAME                 PROVISIONER                                     RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
nfs-sc (default)     cluster.local/nfs-server-nfs-server-provisioner   Delete          Immediate              true                   75s
standard (default)   rancher.io/local-path                           Delete          WaitForFirstConsumer   false                  10m
```

Stats

2. Architecture diagram of the created application



3. Component roles and connections

- **NFS Provisioner**: Deployed via a Helm chart, it dynamically provisions NFS-backed Persistent Volumes in the cluster.
- **Persistent Volume (PV)**: A volume automatically created by the NFS provisioner and shared across multiple pods using the `ReadWriteMany` access mode.
- **Persistent Volume Claim (PVC)**: A claim to the dynamic PV, used by both the HTTP server pod and the Job pod.
- **Job**: A one-time pod that mounts the PVC and writes a sample `index.html` file into the shared volume.
- **Nginx Deployment**: A continuously running pod that mounts the same PVC and serves its contents via an `nginx` HTTP server.
- **Service (NodePort)**: Exposes the `nginx` pod outside the cluster so the application can be accessed through `http://localhost:8080`.

This architecture demonstrates the usage of shared storage in Kubernetes, enabling one pod to write data that is later served by another.