Static persistent volumes

Creating a NFS

NFS is one famous implementation you can use to deploy your own persistent volume

I'm running one on my aws server -

```
version: '3.7'

services:
  nfs-server:
    image: itsthenetwork/nfs-server-alpine:latest
    container_name: nfs-server
    privileged: true
    environment:
      SHARED_DIRECTORY: /exports
    volumes:
      - ./data:/exports:rw
    ports:
      - "2049:2049"
    restart: unless-stopped
```

💡 Make sure the 2049 port on your machine is open

Creating a pv and pvc

Create a persistent volume claim and persistent volume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  storageClassName: nfs
  nfs:
    path: /exports
    server: 52.66.197.168
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: nfs
```

Create a pod

```
apiVersion: v1
kind: Pod
metadata:
  name: mongo-pod
spec:
  containers:
  - name: mongo
    image: mongo:4.4
    command: ["mongod", "--bind_ip_all"]
    ports:
    - containerPort: 27017
    volumeMounts:
    - mountPath: "/data/db"
      name: nfs-volume
  volumes:
  - name: nfs-volume
```

Try it out

- Put some data in mongodb

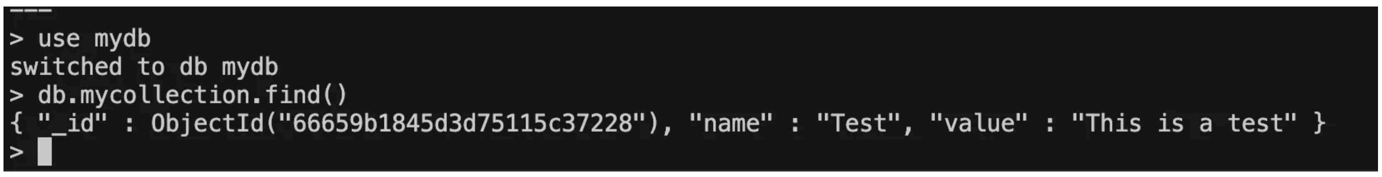
```
kubectl exec -it mongo-pod -- mongo
use mydb
db.mycollection.insert({ name: "Test", value: "This is a test" })
exit
```

- Delete and restart the pod

```
kubectl delete pod mongo-pod
kubectl apply -f mongo.yml
```

- Check if the data persists

```
kubectl exec -it mongo-pod -- mongo
use mydb
db.mycollection.find()
```



```
> use mydb
switched to db mydb
> db.mycollection.find()
{ "_id" : ObjectId("66659b1845d3d75115c37228"), "name" : "Test", "value" : "This is a test" }
>
```