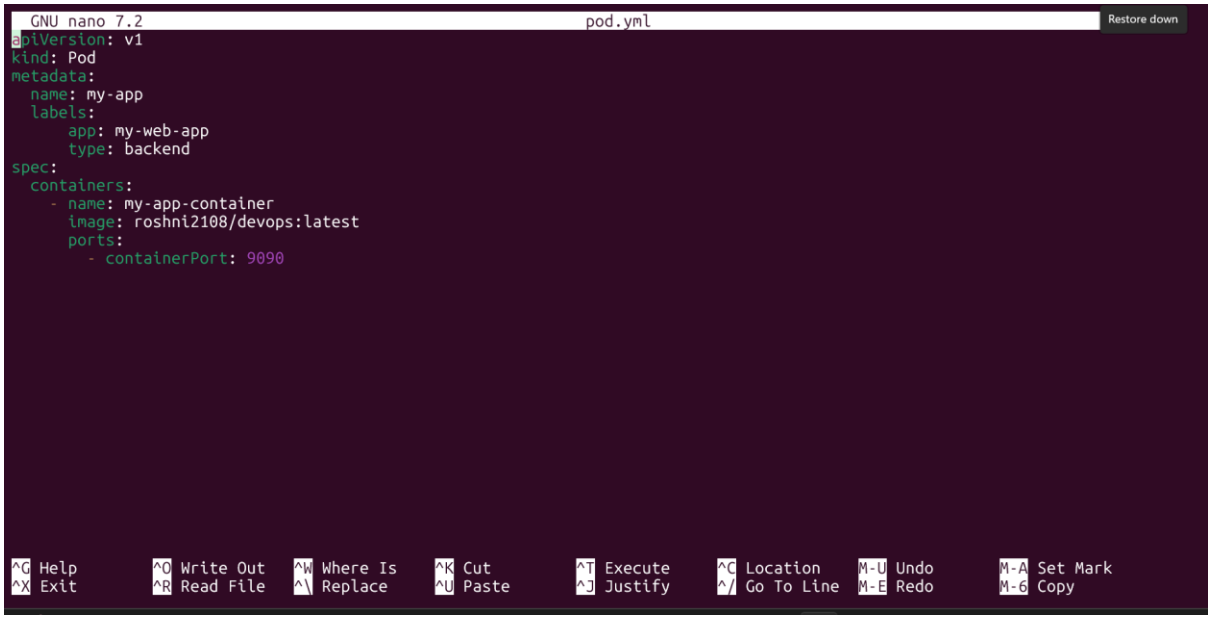
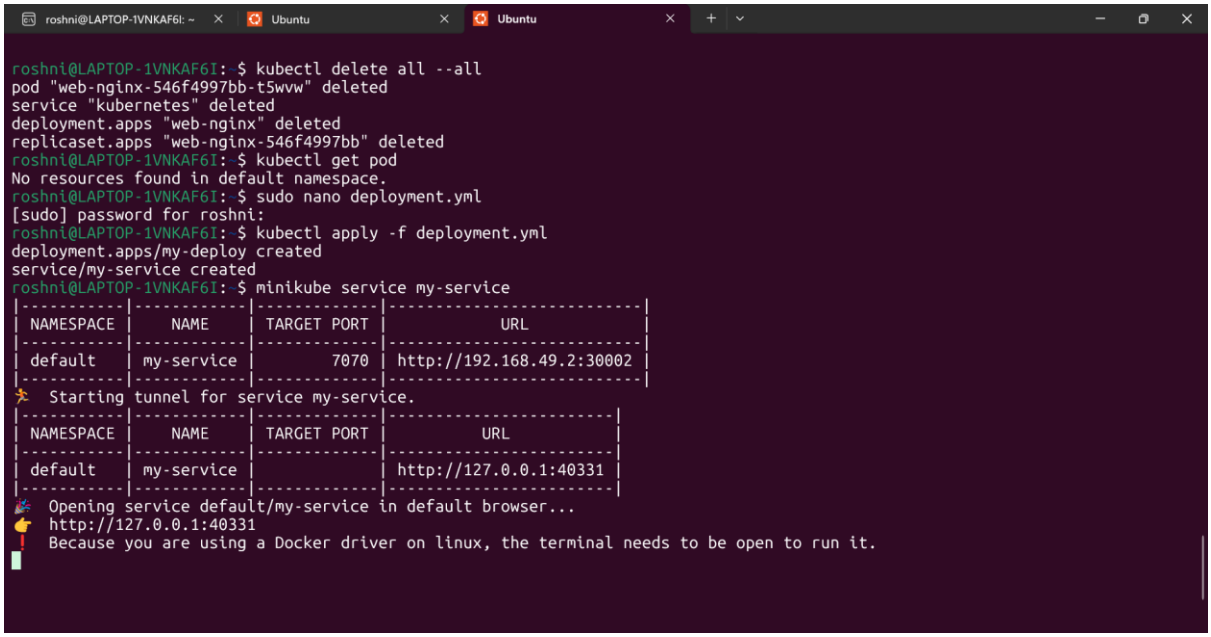
DAY 4





1. Create a pod using run command

$ kubectl run <pod-name> --image=<image-name> --port=<container-port>

2. View all the pods

(In default namespace)

$ kubectl get pods

(In All namespace)

$ kubectl get pods -A

(For a specific namespace)

$ kubectl get pods -n kube-system

(For a specific type)

$ kubectl get pods <pod-name>

$ kubectl get pods <pod-name> -o wide

$ kubectl get pods <pod-name> -o yaml

$ kubectl get pods <pod-name> -o json

3. Describe a pod (View Pod details)

$ kubectl describe pod <pod-name>

4. View Logs of a pod

$ kubectl logs <pod-name>

5. Execute any command inside Pod (Inside Pod OS)

$ kubectl exec <pod-name> -- <command>

6. Delete Deployment

$ kubectl delete deploy <deployment-name>

POD.YML

```
apiVersion: v1
kind: Pod
metadata:
  name: my-app
spec:
```

```
  containers:
    - name: my-app-container
      image: <images>      ports:
        - containerPort: 9090
```

DEPLOYMENT.YML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 4
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
        ports:
```

- containerPort: 7070

kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>

1. Create ReplicaSet by executing above YAML file

$ kubectl create -f rs-test.yml

Do necessary modifications if exist, else create new

$ kubectl apply -f rs-test.yml

Completely Modify Pod Template

$ kubectl replace –f rs-test.yml

2. View ReplicaSets

$ kubectl get replicasets

$ kubectl get rs

$ kubectl get rs –o wide

$ kubectl get rs <replica-set-name> –o json

$ kubectl get rs <replica-set-name> –o yaml

3. View ReplicaSet Description

$ kubectl describe rs <replica-set-name>

4. We can modify generated/updated YAML file

$ kubectl edit rs <replica-set-name>

DEPLOYMENT.YML (service)

apiVersion: apps/v1

kind: Deployment

metadata:

```
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - name: my-app
        image:
        ports:
        - containerPort: 9000

---

apiVersion: v1
kind: Service
metadata:
  name: my-service
  labels:
```

```
    app: my-service
spec:
  type: NodePort
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30002
  selector:
    apptype: web-backend
```