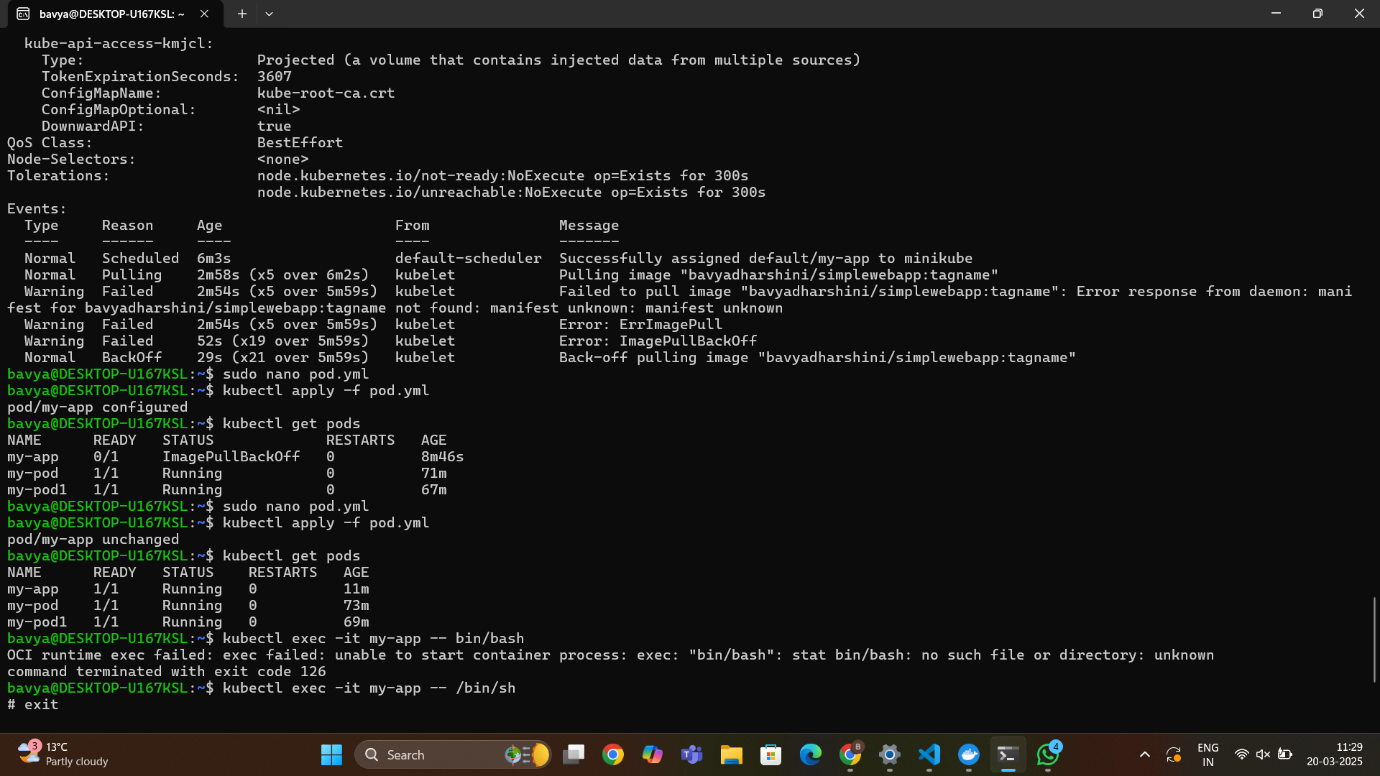
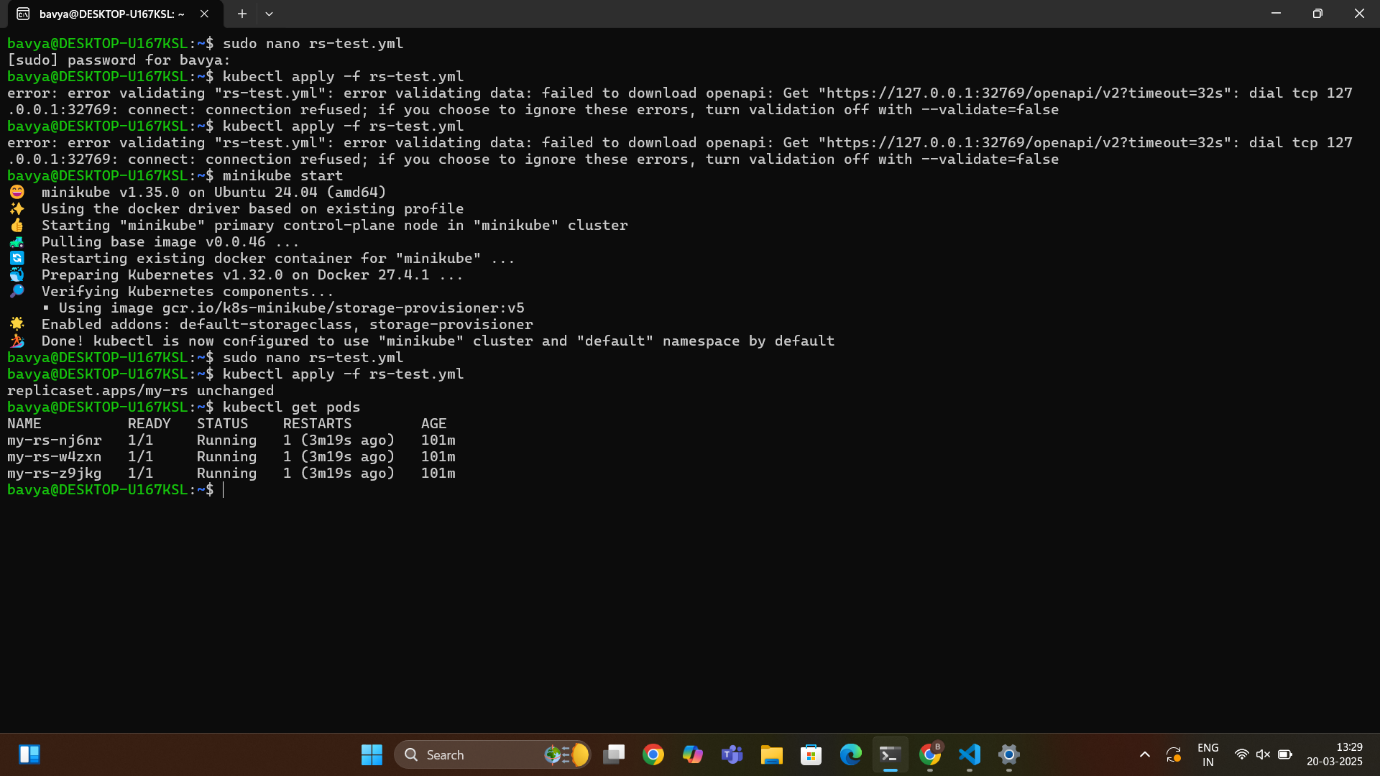
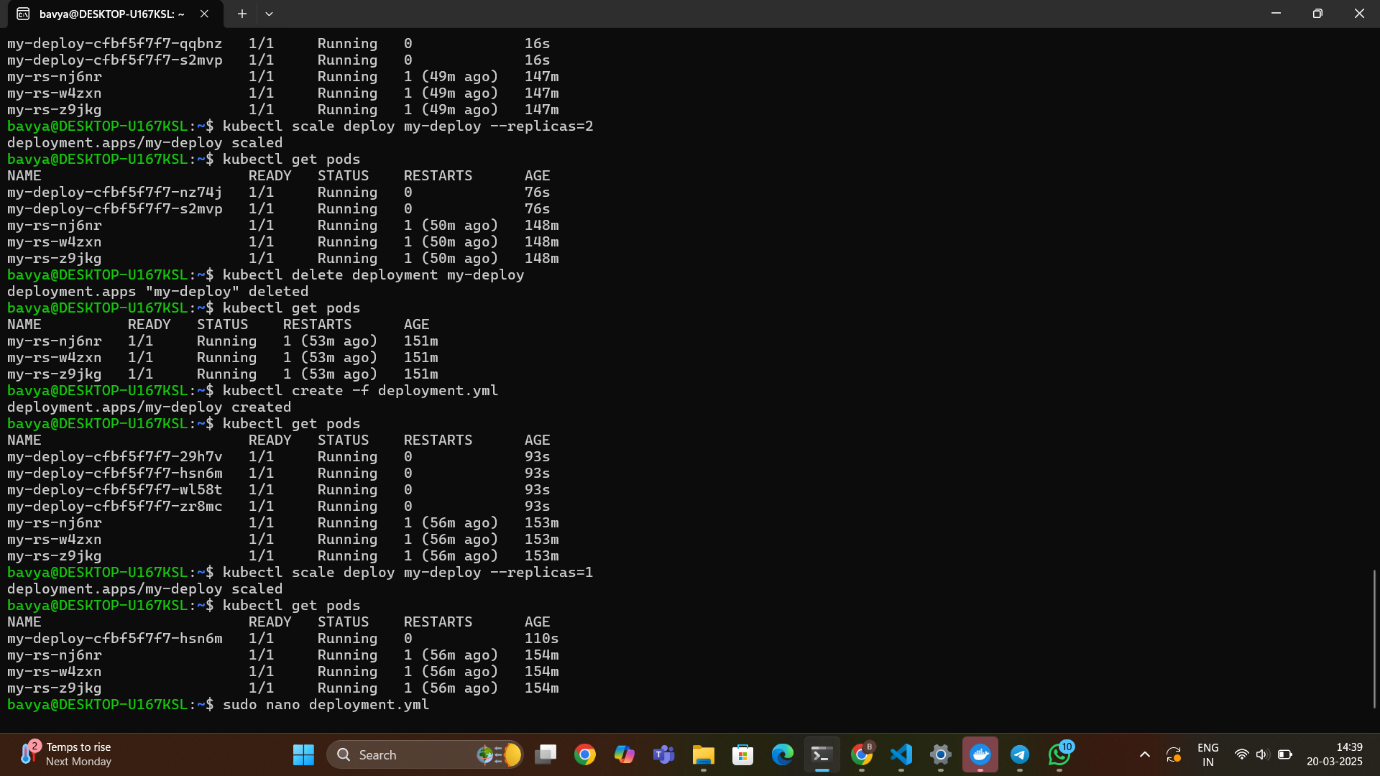
**DEVOPS**

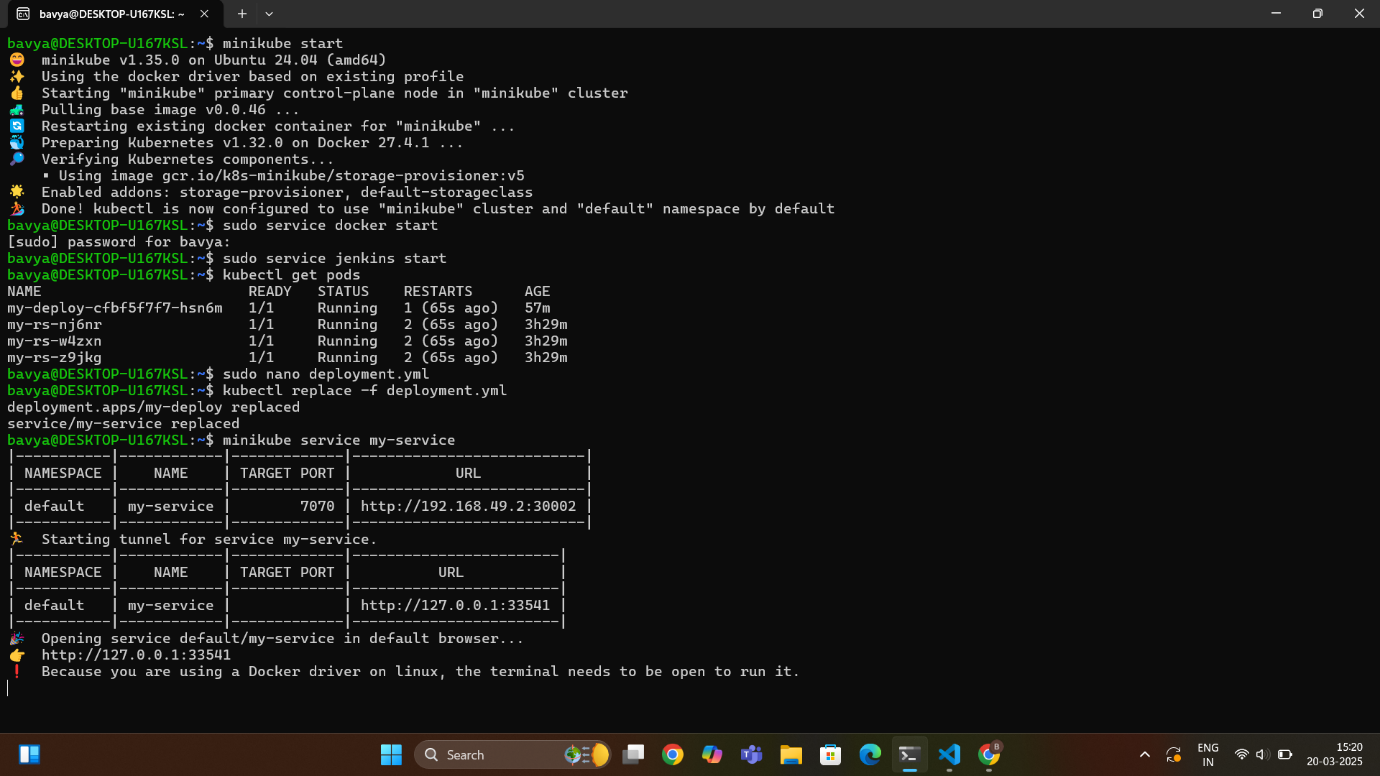
DAY – 4



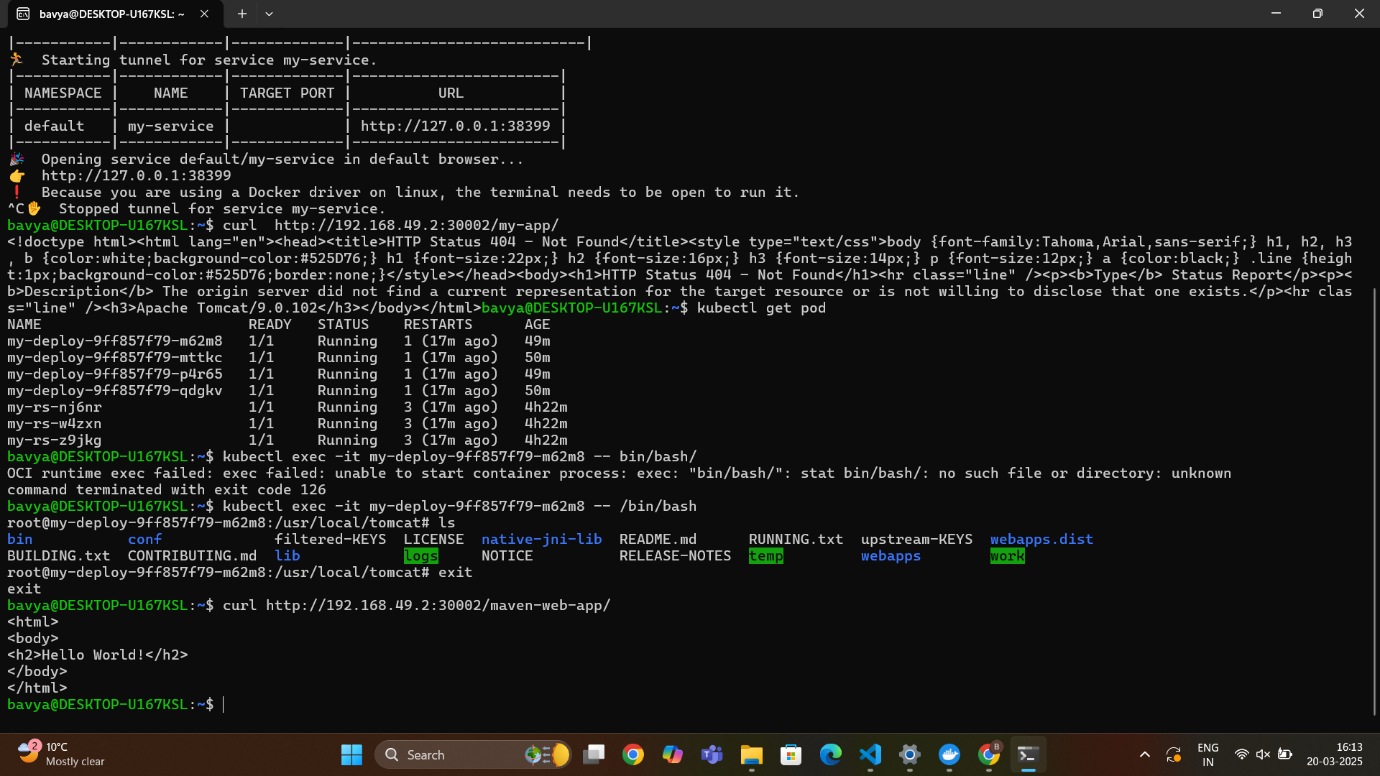
 Pod Creation Output

Replicas Output

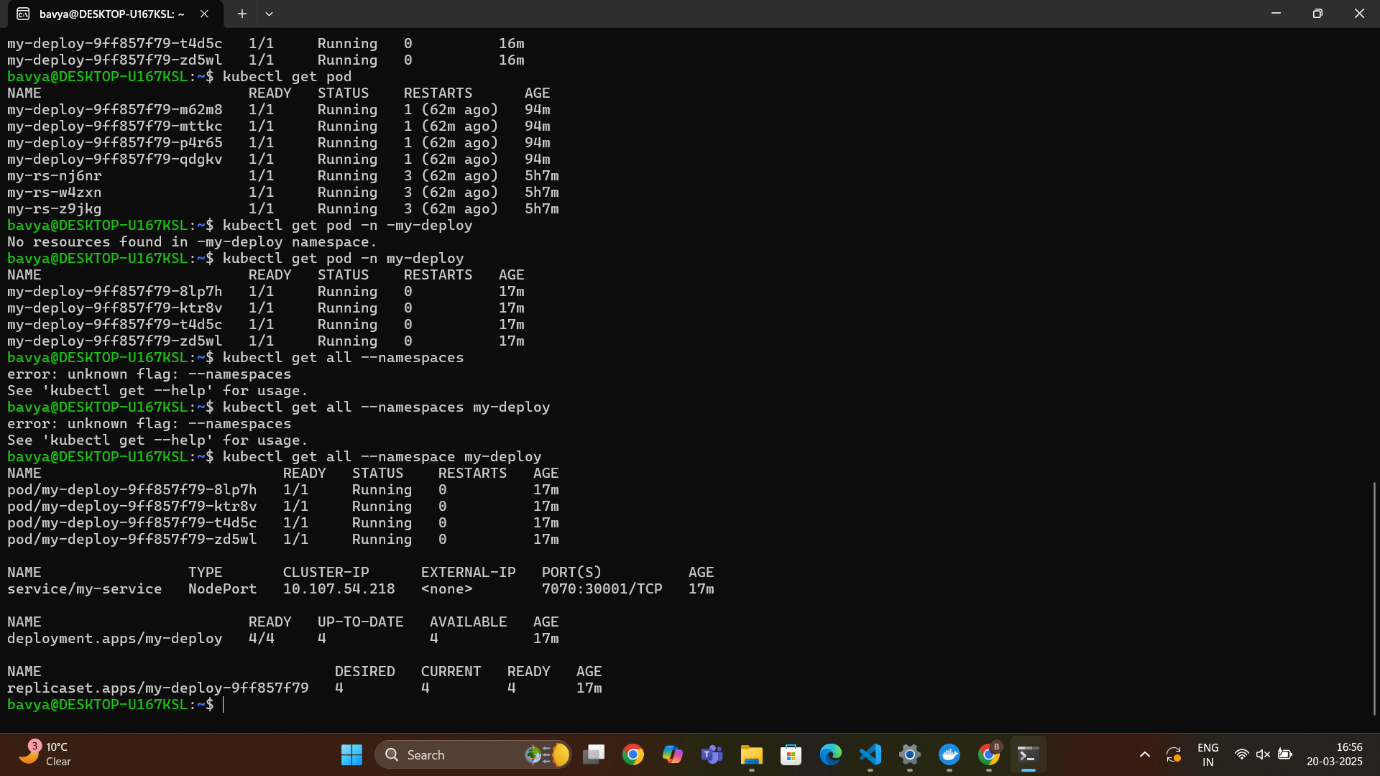


 Deployment Creation Output

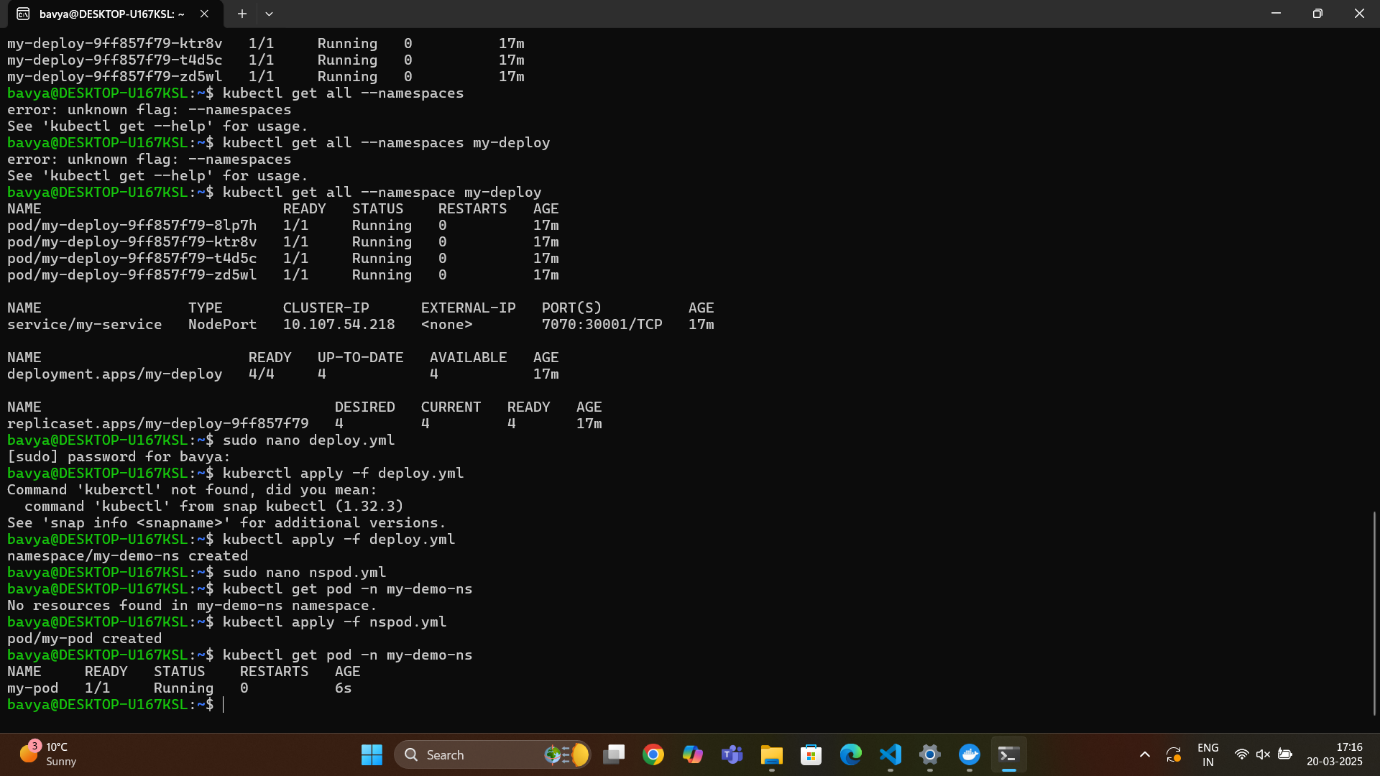
My-Service



My-Service Output



Namespace



Nspod

**Pod Creation:**

**a.**1. minikube start

2. kubectl get pod

3. kubectl delete all --all

4. kubectl run my-pod --image=nginx --port=80

5. kubectl get pod

6. kubectl get pod -o wide

7. kubectl describe my-pod1

8. kubectl exec -it my-pod1 -- bin/bash

pod.yml:

{

apiVersion: v1

kind: Pod

metadata:

name: my-app

spec:

containers:

- name: my-app-container

image: bavyadharshini/simplewebapp:tagname

ports:

- containerPort: 9090

}

9. kubectl apply -f pod.yml

10. kubectl get pods

11. kubectl exec -it my-app -- /bin/sh

**b.**

1. minikube ip

2. kubectl get rs

3. kubectl create deployment web-nginx --image=nginx --replicas=1

4. kubectl get deploy

5. kubectl create pod my-pod --image=nginx

6. kubectl delete deployment web-nginx

7. kubectl delete pod my-pod

**Replicas Creation:**

1. sudo nano rs-test.yml

Yml File:

{

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: my-rs

labels:

name: my-rs

spec:

replicas: 3

selector:

matchLabels:

apptype: web-backend

template:

metadata:

labels:

apptype: web-backend

spec:

containers:

- name: my-app

image: bavyadharshini/simplewebapp:latest

ports:

- containerPort: 9099

}

2. kubectl apply -f rs-test.yml

3. kubectl get pods

4. kubectl get rs

5. kubectl exec -it my-rs-nj6nr -- /bin/bash

6. kubectl describe pod my-rs-nj6nr

**Deployment Creation:**

**a.**

1. sudo nano deployment

Yml File:

{

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: bavyadharshini/simplewebapp:latest

ports:

- containerPort: 7076

}

2. kubectl apply -f deployment.yml

3. kubectl get pods

4. kubectl scale deploy my-deploy --replicas=2

**b.**

1. kubectl create -f deployment.yml

2. kubectl get pods

3. kubectl scale deploy my-deploy --replicas=1

4. kubectl replace -f deployment.yml

**My-Service:**

1. sudo nano deployment.yml

Yml File:

{

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: 7070

---

apiVersion: v1

kind: Service

metadata:

name: my-service

labels:

app: my-service

type: backend-app

spec:

type: NodePort

ports:

- targetPort: 7070

port: 7070

nodePort: 30002

selector:

apptype: web-backend

}

2. kubectl replace -f deployment.yml

3. minikube service my-service

**Namespace:**

1. kubectl get pod -n -my-deploy

2. kubectl get all --namespace my-deploy

**Nspod:**

1. sudo nano deploy.yml

deploy.yml

{

apiVersion: v1

kind: Namespace

metadata:

name: my-demo-ns

}

2. kubectl apply -f deploy.yml

3. sudo nano nspod.yml

nspod.yml

{

apiVersion: v1

kind: Pod

metadata:

name: my-pod

namespace: my-demo-ns

spec:

containers:

- name: my-container

image: nginx:latest

}

4. kubectl apply -f nspod.yml

5. kubectl get pod -n my-demo-ns