DEVEPOS

DAY 4 Task

Pod

1. Create a pod using run command
$ kubectl run <pod-name> --image=<image-name> --port=<container-port>
$ kubectl run my-pod --image=nginx --port=80

2. View all the pods
(In default namespace)
$ kubectl get pods
(In All namespace)

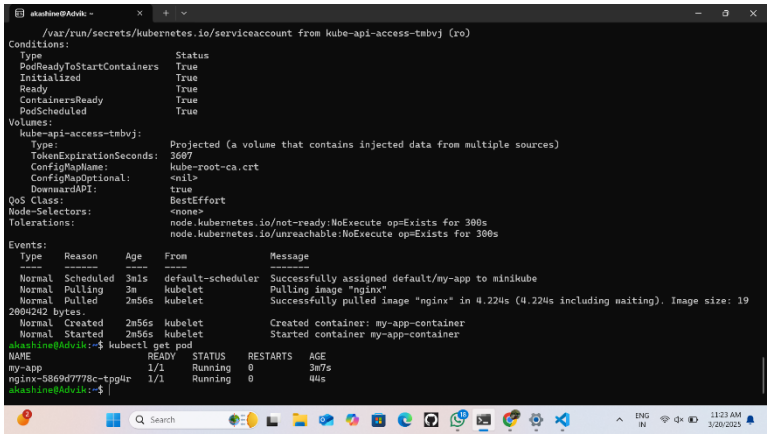
$ kubectl get pods -A
# For a specific namespace
$ kubectl get pods -n kube-system

# 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>
$ kubectl describe pod my-pod
4. View Logs of a pod
$ kubectl logs <pod-name>
$ kubectl logs my-pod

5. Execute any command inside Pod (Inside Pod OS)
$ kubectl exec <pod-name> -- <command>

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  labels:
     app: my-web-app
          type: backend
spec:
  containers:
    - name: nginx-container
      image: nginx
      ports:
        - containerPort: 80
```



Replica

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>
change replicas: count to any other value then (ESC):wq

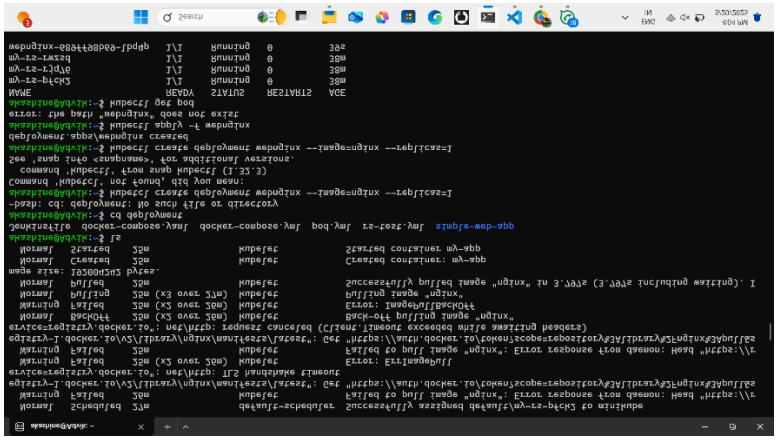
# We can modify our YAML file and then execute apply command
$ kubectl apply -f rs-test.yml

We can Even scale using command also
$ kubectl scale replicaset <replicaset-name> --replicas=<desired-replica-count>

5. Delete ReplicaSet
$ kubectl delete rs <replica-set-name>
$ kubectl delete -f rs-test.yml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 3
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
```

```
  type: RollingUpdate
template:
  metadata:
    labels:
      apptype: web-backend
  spec:
    containers:
    - name: my-app
      image:  nginx
      ports:
          - containerPort: 7070
```



Deploy

1. Create Deployment by executing above YAML file
$ kubectl create -f web-deploy.yml
# Do necessary modifications if exist, else create new
$ kubectl create -f web-deploy.yml
# Completely Modify Pod Template
$ kubectl replace –f web-deploy.yml
#Create deploy
kubectl create deployment webnginx2 --image=nginx:latest --replicas=1

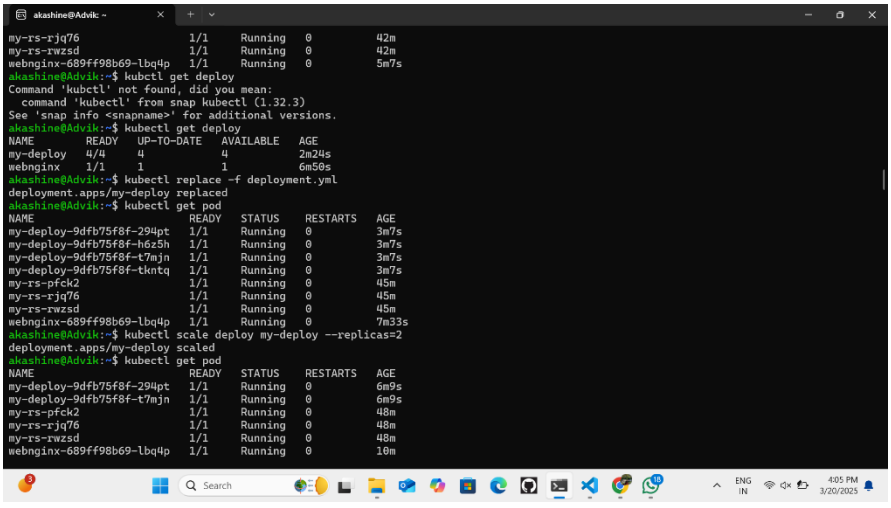
2. View Deployments
$ kubectl get deployments
$ kubectl get deploy
$ kubectl get deploy -o wide
$ kubectl get deploy <deployment-name> -o json
$ kubectl get deploy <deployment-name> -o yaml
3. View Deployment Description
$ kubectl describe deploy <deployment-name>
4. We can modify generated/updated YAML file
$ kubectl edit deploy <deployment-name>
change replicas: count to any other value then (ESC):wq

# We can modify our YAML file and then execute apply command
$ kubectl apply -f web-deploy.yml

We can Even scale using command also
$ kubectl scale deploy <deployment-name> --replicas=<desired-replica-count>

5. Delete Deployment
$ kubectl delete deploy <deployment-name>
$ kubectl delete -f web-deploy.yml

```
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
```



Minikube Service

Minikube service

#need to create a yml file
sudo nano deployment.yml

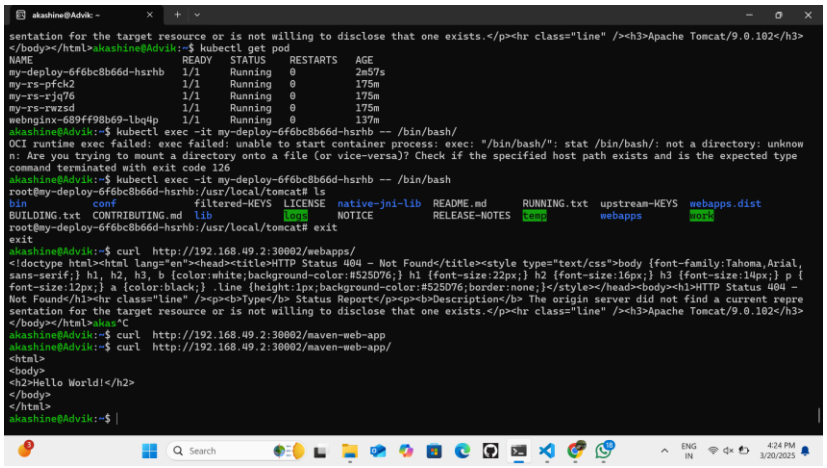
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
      - 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
```

#Apply the deployment
kubectl apply -f deployment.yml

#replace the deployment
kubectl replace -f deployment.yml

#Run the service
minikube service my-service

#curl the url
curl <url>/<file_name>/



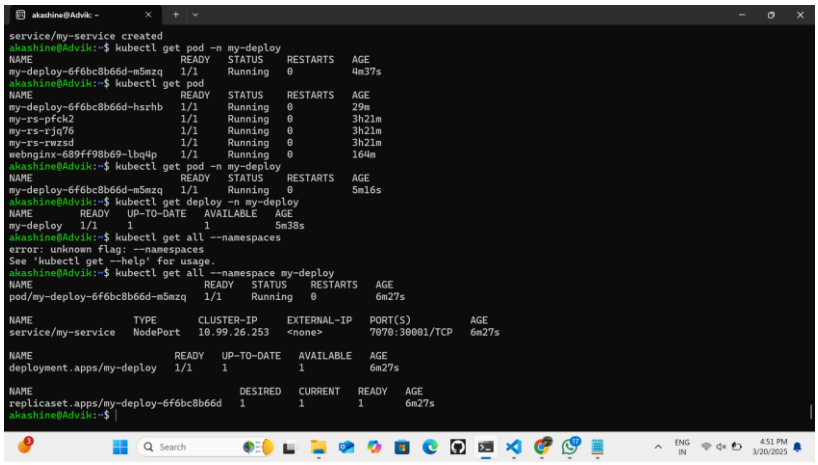
Namespace

# To create a namespace:
$ kubectl create namespace <namespace-name>
$ kubectl create ns my-bank
# To switch to a specific namespace: (make this as default type)
$ kubectl config set-context --current --namespace=<namespace-name>
# To list all namespaces:
$ kubectl get namespaces
# To get resources within a specific namespace:
$ kubectl get <resource-type> -n <namespace-name>
$ kubectl get deploy -n my-bank
$ kubectl get deploy --namespace my-bank
$ kubectl get all --namespace my-bank
# To delete a namespace and all associated resources:
$ kubectl delete namespace <namespace-name>
$ kubectl delete ns my-bank

kubectl create ns my-deploy
kubectl apply -f deploy.yml -n mydeploy

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-demo-ns

apiVersion: v1
kind: Pod
metadata:
  name: my-pod
  namespace: my-demo-ns
spec:
  containers:
  - name: my-container
    image: nginx:latest
```

Namespace



Namespace yml