Two way we can scaleup or scale down the pod count

1 By using manifest yml

2 Using kubectl command

# kubectl scale rs javawebrs --replicas 3

Note: When we execute above command replicaset will check how many pods are currently running based on that it will decide scale up or scale down.

Note: If we want to delete the pods then we have to delete the resource which created those pods.

# kubectl delete rs javawebrs

To delete the replica set from our cluster

Note: In ReplicaSet, scale up & scale down is manual process.

K8S supports Auto Scaling when we use 'Deployment' to create pods.

===============================================================

create a manifest yml file for replicaset and service resource

===============================================================

---

apiVersion: apps/v1

kind: ReplicaSet

metadata:

name: javawebrs

spec:

replicas: 2

selector:

matchLabels:

app: javawebapp

template:

metadata:

name: javawebpod

labels:

app: javawebapp

spec:

containers:

- name: javawebcontainer

image: vinodses/my-web-app

ports:

- containerPort: 8080

---

apiVersion: v1

kind: Service

metadata:

name: javawebappsvc

spec:

type: LoadBalancer

selector:

app: javawebapp

ports:

- port: 80

targetPort: 8080

...

===========

Deployment

===========

Deployment is one of the k8s recourse to create and manage the pods

It is the most recommended approach to deploy our application in k8s

Deployment will support or manage our pod life cycle

Also, we have several advantages

1 will get the zero downtime

2 autoscaling

3 Rolling update and roll back

Deployment strategies

1 Recreate (all at once)

2 Rolling update (one by one)

Recreate means it will delete all existing pods and will create new pods.

Rolling Update means it will delete and create new pod one by one.