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

k8s cluster ip service

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

For all our pods a static ip address will be created

When you want to expose the pods by using cluster ip service and can specify service type cluster IP

what all the pods running for our application, all the pods will be mapped to one static ip address that is the cluster ip service

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

Manifest yml fie for creating cluster ip service

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

---

apiVersion: v1

kind: Pod

metadata:

name: javawebapppod

labels:

app: javawebapp

spec:

containers:

- name: javawebappcontainer

image: vinodses/my-web-app

ports:

- containerPort: 8080

---

apiVersion: v1

kind: Service

metadata:

name: javawebappsvc

spec:

type: ClusterIP

selector:

app: javawebapp

ports:

- port: 80

targetport: 8080

...

When we use service type as Cluster IP then one static ip will be created to access our pods with in the cluster

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

Name spaces

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

Name spaces are used to group the k8s resources

generally, we do like this

frontend-app-pods === under one namespace - vinodses-fe-ns

backend-app-pods === under one name space - vinodses-be-ns

database-app-pods === under one name space - vinodses-db-ns

logically we are grouping the resources in k8s cluster by using namespaces

We can create multiple namespace in a k8s cluster

e.g.: vinodses-fe-ns, vinodses-be-ns, vinodses-db-ns

How to check the name spaces in the k8s cluster

#kubectl get ns

It will show all the name spaces in our k8s cluster

get the pods available in kube-system namespace

$ kubectl get pods -n kube-system

Note: If we don't give namespace then k8s will check under default namespace

We can create a name space in two different ways

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

1 Using kubectl create ns command

2 By using manifest yml file

Approach -1

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

create a name space directly

# kubectl create ns vinodses

# kubectl get ns

Approach -2

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

We need to specify the manifest yml file

---

apiVersion: v1

kind: Namespace

metadata:

name: vinodses-ns

...

Create a pod and service resource using my own namespace

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

---

apiVersion: v1

kind: Namespace

metadata:

name: vinodses-ns

---

apiVersion: v1

kind: Pod

metadata:

name: javawebapppod

namespace: vinodses-ns

labels:

app: javawebapp

spec:

containers:

- name: javawebappcontainer

image: vinodses/my-web-app

ports:

- containerPort: 8080

---

apiVersion: v1

kind: Service

metadata:

name: javawebappsvc

namespace: vinodses-ns

spec:

type: LoadBalancer

selector:

app: javawebapp

ports:

- port: 80

targetPort: 8080

...

# kubectl apply -f k8s-pod-service-namespace.yml

It will execute the manifest yml file and creating pods, service and namespace

# kubectl get ns

You can view the available namespaces in our k8s cluster default namespace

# kubectl get pods -n vinodses-ns

It will display the available pods inside the vinodses-ns namespace

# kubectl get svc -n vinodses-ns

Display the service resources under vinodses-ns namespace

# kubectl get all -n vinodses-ns

What all the resources created inside the vinodses-ns namespace

# kubectl delete ns vinodses-ns

will delete all the pods and services under the vinodses-ns namespace