Lab Exercise 9- Managing Namespaces in Kubernetes

Step 1: Understand Namespaces

Namespaces provide a mechanism for scoping resources in a cluster. Namespaces can be used to:

- Create environments for different applications or teams.
- Apply policies like resource quotas or network policies on a per-namespace basis.
- Separate operational environments (like development and production).

Step 2: List Existing Namespaces

To list all the namespaces in your Kubernetes cluster:

```
kubectl get namespaces
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl get namespaces
NAME              STATUS   AGE
default           Active   4d21h
kube-node-lease   Active   4d21h
kube-public       Active   4d21h
kube-system       Active   4d21h

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

You will typically see default namespaces like default, kube-system, and kube-public.

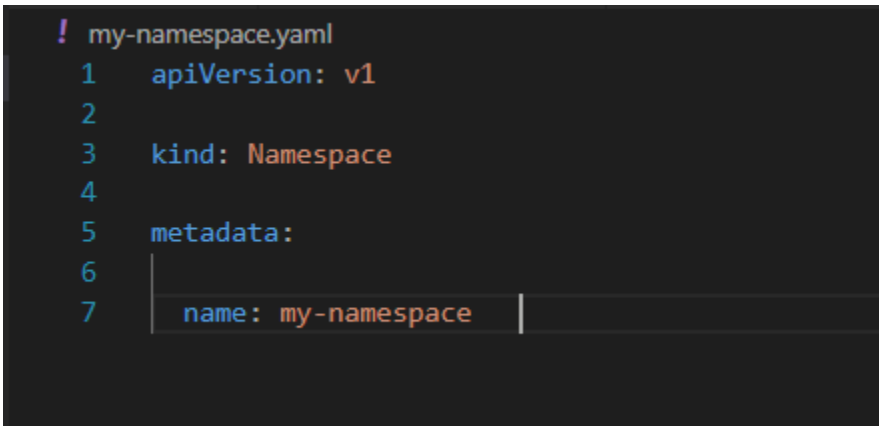
Step 3: Create a Namespace

You can create a namespace using a YAML file or directly with the kubectl command.

Using YAML File

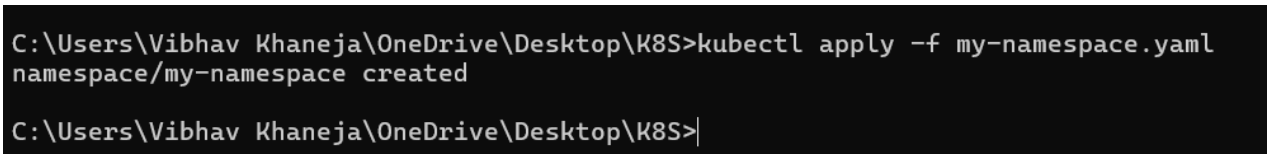
Create a file named ***my-namespace.yaml*** with the following content:

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-namespace
```



Apply this YAML to create the namespace:

```
kubectl apply -f my-namespace.yaml
```



Verify that the namespace is created:

```
kubectl get namespaces
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl get namespaces
NAME              STATUS   AGE
default           Active   4d21h
kube-node-lease   Active   4d21h
kube-public       Active   4d21h
kube-system       Active   4d21h
my-namespace      Active   24s

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

You should see my-namespace listed in the output.

Step 4: Deploy Resources in a Namespace

Create resources such as Pods, Services, or Deployments within the new namespace.

Deploy a Pod in the Namespace

Create a YAML file named ***nginx-pod.yaml*** with the following content:

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: my-namespace   # Specify the namespace for the Pod.
spec:
  containers:
  - name: nginx
    image: nginx:latest
    ports:
    - containerPort: 80
```

nginx-pod.yaml

```yaml
apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod

  namespace: my-namespace   # Specify the namespace for the Pod.

spec:

  containers:

  - name: nginx

    image: nginx:latest

    ports:

    - containerPort: 80
```

Apply this YAML to create the Pod:

```
kubectl apply -f nginx-pod.yaml
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl apply -f nginx-pod.yaml
pod/nginx-pod created

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

Check the status of the Pod within the namespace:

```
kubectl get pods -n my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          30s

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

To describe the Pod and see detailed information:

```
kubectl describe pod nginx-pod -n my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl describe pod nginx-pod -n my-namespace
Name:             nginx-pod
Namespace:        my-namespace
Priority:         0
Service Account:  default
Node:             docker-desktop/192.168.65.3
Start Time:       Tue, 12 Nov 2024 12:59:07 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.1.0.20
IPs:
  IP:  10.1.0.20
Containers:
  nginx:
    Container ID:   docker://648cfde89edf4c28c5167e04b2a1c55091b2ebd26bae5856fa1cce88f17fc13b
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f470
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Tue, 12 Nov 2024 12:59:16 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-msdl2 (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
```

Create a Service in the Namespace

Create a YAML file named ***nginx-service.yaml*** with the following content:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: my-namespace   # Specify the namespace for the Service.
spec:
  selector:
    app: nginx-pod
  ports:
  - protocol: TCP
```

```
  port: 80
  targetPort: 80
type: ClusterIP
```

nginx-service.yaml

```yaml
apiVersion: v1

kind: Service

metadata:

  name: nginx-service

  namespace: my-namespace   # Specify the namespace for the Service.

spec:

  selector:

    app: nginx-pod

  ports:

  - protocol: TCP

    port: 80

    targetPort: 80

  type: ClusterIP
```

Apply this YAML to create the Service:

```
kubectl apply -f nginx-service.yaml
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl apply -f nginx-service.yaml
service/nginx-service created

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

Check the status of the Service within the namespace:

```
kubectl get services -n my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl get services -n my-namespace
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.100.203.214   <none>        80/TCP    30s

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

To describe the Service and see detailed information:

```
kubectl describe service nginx-service -n my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl describe service nginx-service -n my-namespace
Name:              nginx-service
Namespace:         my-namespace
Labels:            <none>
Annotations:       <none>
Selector:          app=nginx-pod
Type:              ClusterIP
IP Family Policy:  SingleStack
IP Families:       IPv4
IP:                10.100.203.214
IPs:               10.100.203.214
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         <none>
Session Affinity:  None
Events:            <none>

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

Step 5: Switching Context Between Namespaces

When working with multiple namespaces, you can specify the namespace in kubectl commands or switch the default context.

Specify Namespace in Commands

You can specify the namespace directly in kubectl commands using the -n or --namespace flag:

```
kubectl get pods -n my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          3m58s

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

Set Default Namespace for kubectl Commands

To avoid specifying the namespace every time, you can set the default namespace for the current context:

```
kubectl config set-context --current --namespace=my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl config set-context --current --namespace=my-namespace
Context "docker-desktop" modified.

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

Verify the current context's namespace:

```
kubectl config view --minify | grep namespace:
```

```
Vibhav Khaneja@LAPTOP-QKI1ILNV MINGW32 ~
$ kubectl config view --minify | grep namespace:
    namespace: my-namespace

Vibhav Khaneja@LAPTOP-QKI1ILNV MINGW32 ~
$
```

Step 6: Clean Up Resources

To delete the resources and the namespace you created:

```
kubectl delete -f nginx-pod.yaml
kubectl delete -f nginx-service.yaml
kubectl delete namespace my-namespace
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl delete -f nginx-pod.yaml
pod "nginx-pod" deleted

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl delete -f nginx-service.yaml
service "nginx-service" deleted

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl delete namespace my-namespace
namespace "my-namespace" deleted
```

Ensure that the namespace and all its resources are deleted:

```
kubectl get namespaces
```

```
C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>kubectl get namespaces
NAME              STATUS   AGE
default           Active   4d21h
kube-node-lease   Active   4d21h
kube-public       Active   4d21h
kube-system       Active   4d21h

C:\Users\Vibhav Khaneja\OneDrive\Desktop\K8S>
```