Lab Exercise 13- 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
```

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
```

Using kubectl Command

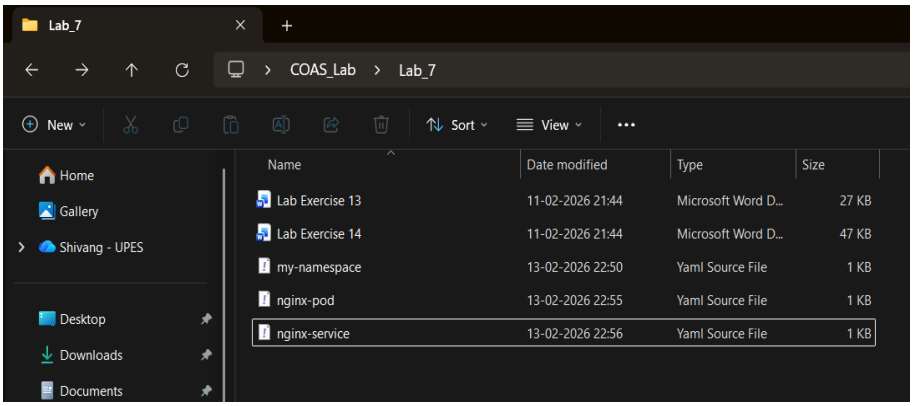
Alternatively, create a namespace using the kubectl command:

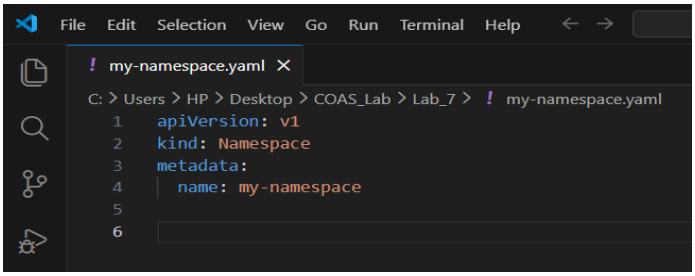
```
kubectl create namespace my-namespace
```

Verify that the namespace is created:

```
kubectl get namespaces
```

You should see my-namespace listed in the output.






```
PS C:\Users\HP> cd C:\Users\HP\Desktop\COAS_Lab\Lab_7
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   3d
kube-node-lease        Active   3d
kube-public            Active   3d
kube-system            Active   3d
kubernetes-dashboard   Active   3d
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> wsl
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.6.87.2-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Feb 13 22:49:42 IST 2026

  System load:  1.41               Processes:             39
  Usage of /:   0.6% of 1006.85GB  Users logged in:       1
  Memory usage: 17%                IPv4 address for eth0: 172.22.242.177
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_7# touch my-namespace.yaml
root@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_7# exit
logout
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl apply -f my-namespace.yaml
namespace/my-namespace created
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl create namespace my-namespace
Error from server (AlreadyExists): namespaces "my-namespace" already exists
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   3d
kube-node-lease        Active   3d
kube-public            Active   3d
kube-system            Active   3d
kubernetes-dashboard   Active   3d
my-namespace           Active   31s
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7>
```

Step 4: Deploy Resources in a 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
spec:
  containers:
  - name: nginx
    image: nginx:latest
    ports:
    - containerPort: 80
```

Apply this YAML to create the Pod:

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

Check the status of the Pod within the namespace:

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

To describe the Pod and see detailed information:

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

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
spec:
  selector:
    app: nginx-pod
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: ClusterIP
```

Apply this YAML to create the Service:

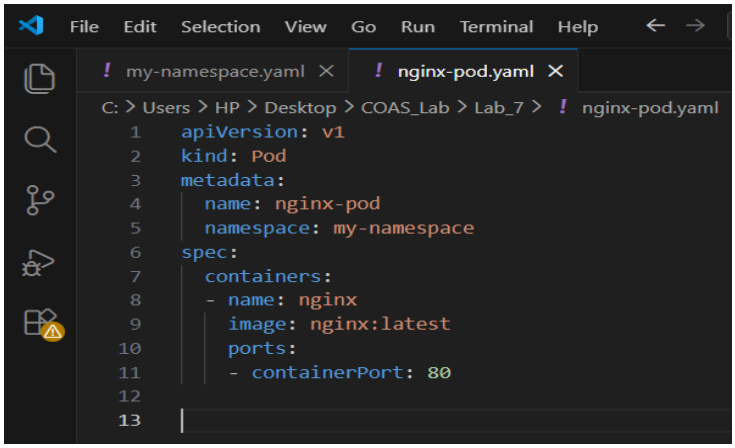
```
kubectl apply -f nginx-service.yaml
```

Check the status of the Service within the namespace:

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

To describe the Service and see detailed information:

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




```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> wsl
root@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_7# touch nginx-pod.yaml
root@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_7# touch nginx-service.yaml
root@Shivang:/mnt/c/Users/HP/Desktop/COAS_Lab/Lab_7# exit
logout
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl get pods -n my-namespace
NAME        READY   STATUS              RESTARTS   AGE
nginx-pod   0/1     ContainerCreating   0          12s
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> 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:       Fri, 13 Feb 2026 22:57:26 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.1.0.38
IPs:
  IP:  10.1.0.38
Containers:
  nginx:
    Container ID:   docker://902c699f3d7fcc3b87c80a6c194a3c21665a84f495f0193559e4d06092dc6ee3
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Fri, 13 Feb 2026 22:57:38 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-szs4q (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
```

```
Volumes:
  kube-api-access-szs4q:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    Optional:                false
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  30s   default-scheduler  Successfully assigned my-namespace/nginx-pod to docker-desktop
  Normal  Pulling    30s   kubelet            Pulling image "nginx:latest"
  Normal  Pulled     19s   kubelet            Successfully pulled image "nginx:latest" in 11.714s (11.714s including waiting). Image size: 62939286 bytes.
  Normal  Created    19s   kubelet            Created container: nginx
  Normal  Started    19s   kubelet            Started container nginx
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7>
```

```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl apply -f nginx-service.yaml
service/nginx-service created
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl get services -n my-namespace
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.105.50.212   <none>        80/TCP    12s
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> 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:                       10.105.50.212
IPs:                      10.105.50.212
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
Endpoints:
Session Affinity:         None
Internal Traffic Policy:  Cluster
Events:                   <none>
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7>
```

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
```

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
```

Verify the current context's namespace:

```
kubectl config view --minify | grep namespace
kubectl config view --minify | Select-String namespace      #For Poweshell
```

```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          4m13s
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl config set-context --current --namespace=my-namespace
Context "docker-desktop" modified.
```

```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl config set-context --current --namespace=default
Context "docker-desktop" modified.
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl config view --minify | Select-String namespace

    namespace: default
```

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
```

Ensure that the namespace and all its resources are deleted:

```
kubectl get namespaces
```

```
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl delete -f nginx-pod.yaml
pod "nginx-pod" deleted from my-namespace namespace
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl delete -f nginx-service.yaml
service "nginx-service" deleted from my-namespace namespace
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl delete namespace my-namespace
namespace "my-namespace" deleted
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   3d
kube-node-lease        Active   3d
kube-public            Active   3d
kube-system            Active   3d
kubernetes-dashboard   Active   3d
PS C:\Users\HP\Desktop\COAS_Lab\Lab_7>
```