Lab Exercise 13- Managing Namespaces in Kubernetes

Step 1: Understand Namespaces

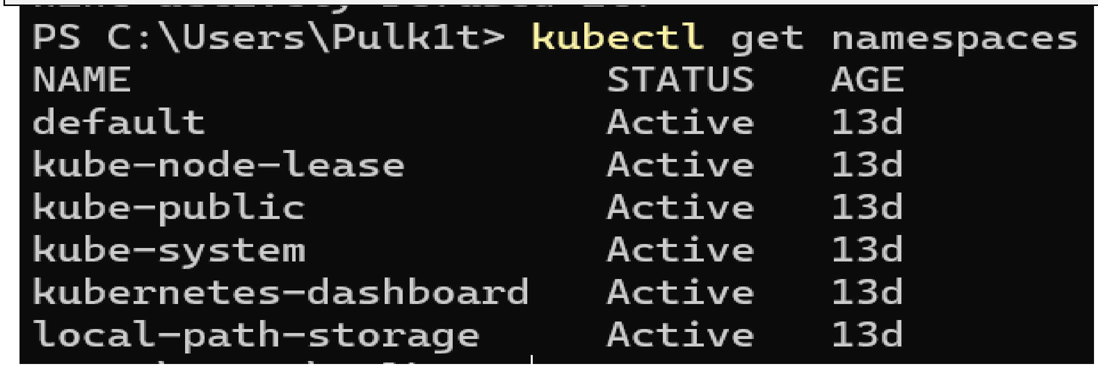
Namespaces provide a mechanism for scoping resources in a cluster. Namespaces can be used to:

- 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
```



```
PS C:\Users\Pulk1t> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   13d
kube-node-lease        Active   13d
kube-public            Active   13d
kube-system            Active   13d
kubernetes-dashboard   Active   13d
local-path-storage     Active   13d
```

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.

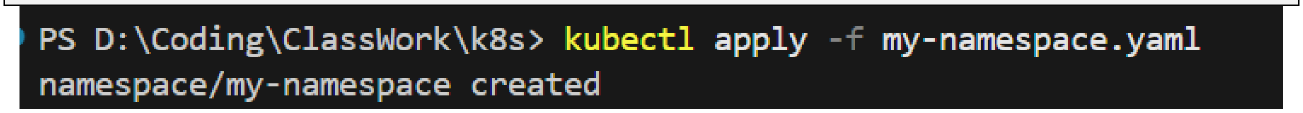
Using YAML File

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
```

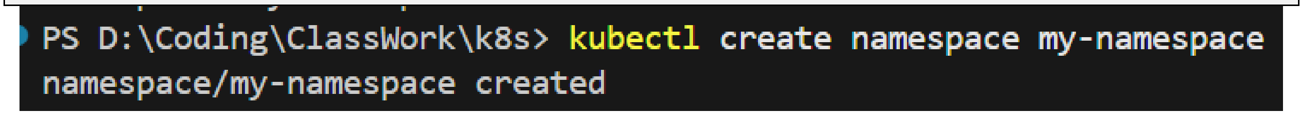


```
PS D:\Coding\ClassWork\k8s> kubectl apply -f my-namespace.yaml
namespace/my-namespace created
```

Using kubectl Command

Alternatively, create a namespace using the kubectl command:

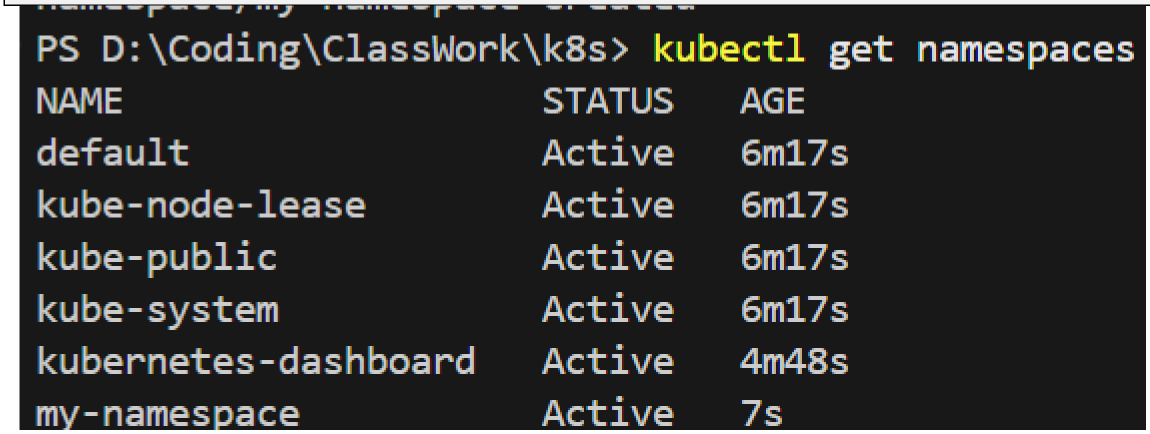
```
kubectl create namespace my-namespace
```



```
PS D:\Coding\ClassWork\k8s> kubectl create namespace my-namespace
namespace/my-namespace created
```

Verify that the namespace is created:

```
kubectl get namespaces
```



```
PS D:\Coding\ClassWork\k8s> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   6m17s
kube-node-lease        Active   6m17s
kube-public            Active   6m17s
kube-system            Active   6m17s
kubernetes-dashboard   Active   4m48s
my-namespace           Active   7s
```

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.

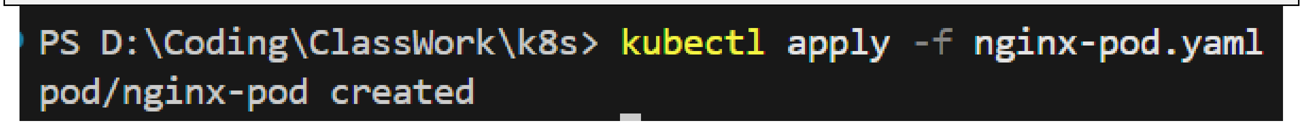
Deploy a Pod in the Namespace

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

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

Apply this YAML to create the Pod:

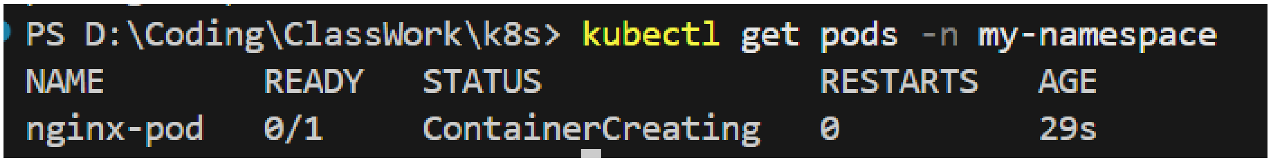
```
kubectl apply -f nginx-pod.yaml
```



```
PS D:\Coding\ClassWork\k8s> kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
```

Check the status of the Pod within the namespace:

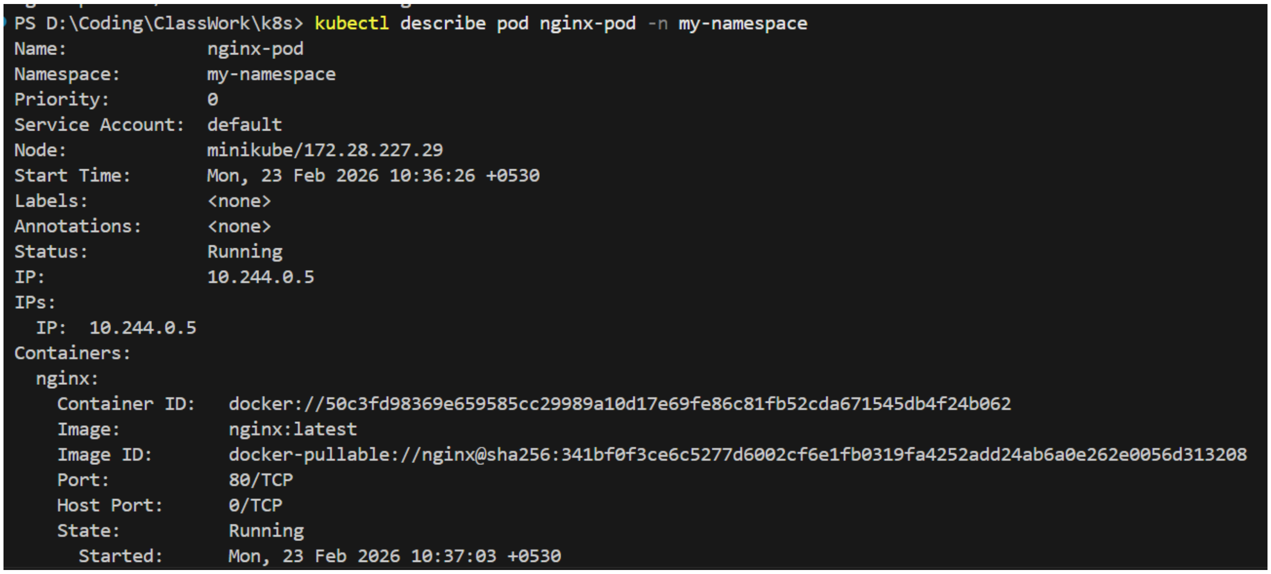
```
kubectl get pods -n my-namespace
```



```
PS D:\Coding\ClassWork\k8s> kubectl get pods -n my-namespace
NAME        READY   STATUS              RESTARTS   AGE
nginx-pod   0/1     ContainerCreating   0          29s
```

To describe the Pod and see detailed information:

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



```
PS D:\Coding\ClassWork\k8s> kubectl describe pod nginx-pod -n my-namespace
Name:             nginx-pod
Namespace:        my-namespace
Priority:         0
Service Account:  default
Node:             minikube/172.28.227.29
Start Time:       Mon, 23 Feb 2026 10:36:26 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.5
IPs:
  IP:  10.244.0.5
Containers:
  nginx:
    Container ID:   docker://50c3fd98369e659585cc29989a10d17e69fe86c81fb52cda671545db4f24b062
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Mon, 23 Feb 2026 10:37:03 +0530
```

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
```

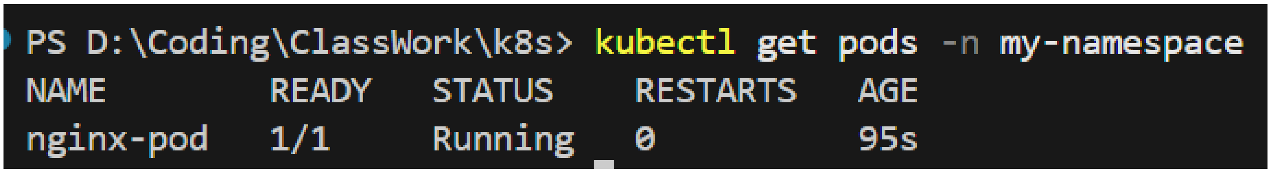
Step 5: Switching Context Between Namespaces

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
```

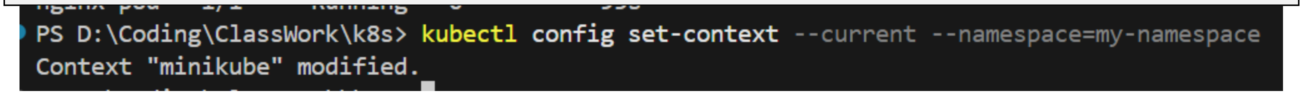


```
PS D:\Coding\ClassWork\k8s> kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          95s
```

Set Default Namespace for kubectl Commands

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

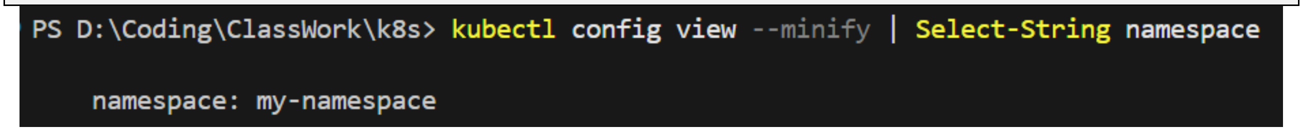
```
kubectl config set-context --current --namespace=my-namespace
```



```
PS D:\Coding\ClassWork\k8s> kubectl config set-context --current --namespace=my-namespace
Context "minikube" modified.
```

Verify the current context’s namespace:

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



```
PS D:\Coding\ClassWork\k8s> kubectl config view --minify | Select-String namespace

    namespace: my-namespace
```

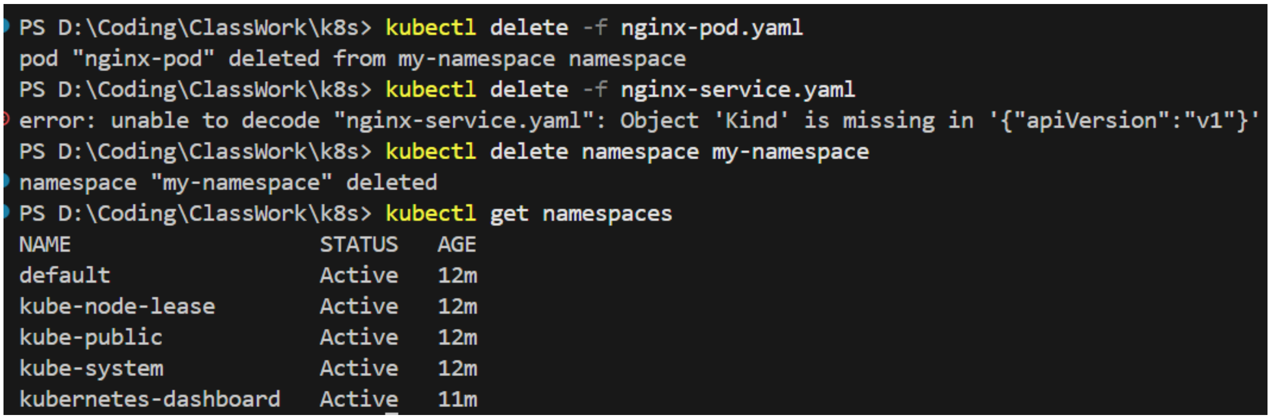
Step 6: Clean Up Resources

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 D:\Coding\ClassWork\k8s> kubectl delete -f nginx-pod.yaml
pod "nginx-pod" deleted from my-namespace namespace
PS D:\Coding\ClassWork\k8s> kubectl delete -f nginx-service.yaml
error: unable to decode "nginx-service.yaml": Object 'Kind' is missing in '{"apiVersion":"v1"}'
PS D:\Coding\ClassWork\k8s> kubectl delete namespace my-namespace
namespace "my-namespace" deleted
PS D:\Coding\ClassWork\k8s> kubectl get namespaces
NAME                   STATUS   AGE
default                Active   12m
kube-node-lease        Active   12m
kube-public            Active   12m
kube-system            Active   12m
kubernetes-dashboard   Active   11m
```