

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\Devanshi\Desktop\COS_Lab\lab_13> kubectl get namespaces
NAME                STATUS    AGE
default             Active    28s
kube-node-lease     Active    28s
kube-public         Active    28s
kube-system         Active    28s
```

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

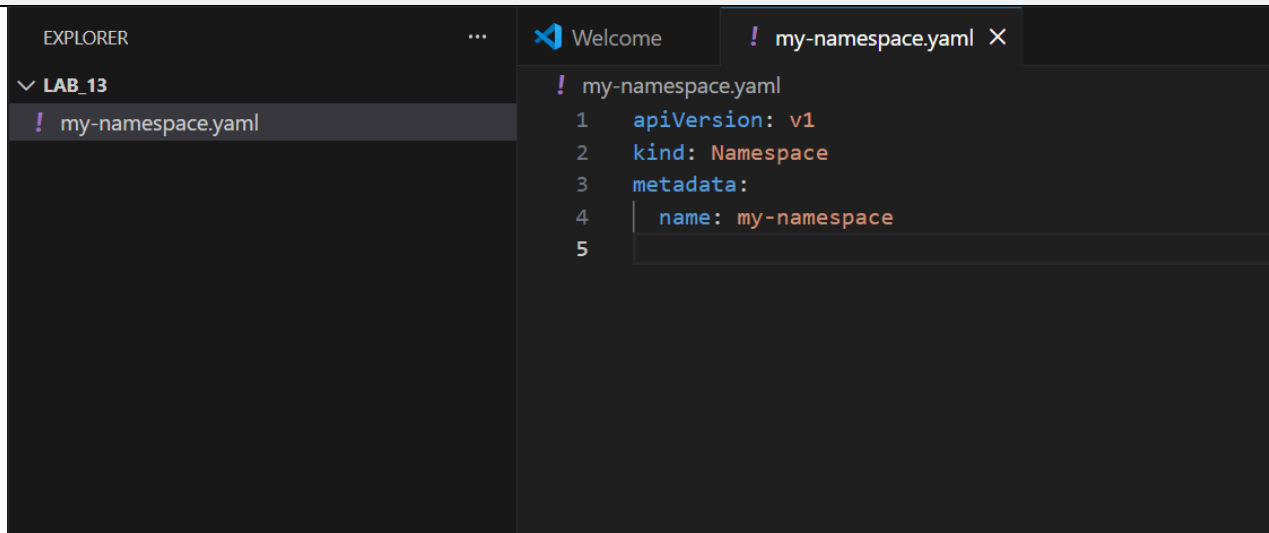
Step 3: Create a Namespace

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 C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl apply -f my-namespace.yaml
namespace/my-namespace created
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get namespaces
NAME                STATUS   AGE
default             Active   2m38s
kube-node-lease     Active   2m38s
kube-public         Active   2m38s
kube-system         Active   2m38s
my-namespace        Active   21s
```

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
```

```
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl create namespace my-namespace1
namespace/my-namespace1 created
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get namespaces
NAME                STATUS    AGE
default             Active    3m1s
kube-node-lease     Active    3m1s
kube-public         Active    3m1s
kube-system         Active    3m1s
my-namespace        Active    44s
my-namespace1       Active    3s
```

You should see my-namespace listed in the output.

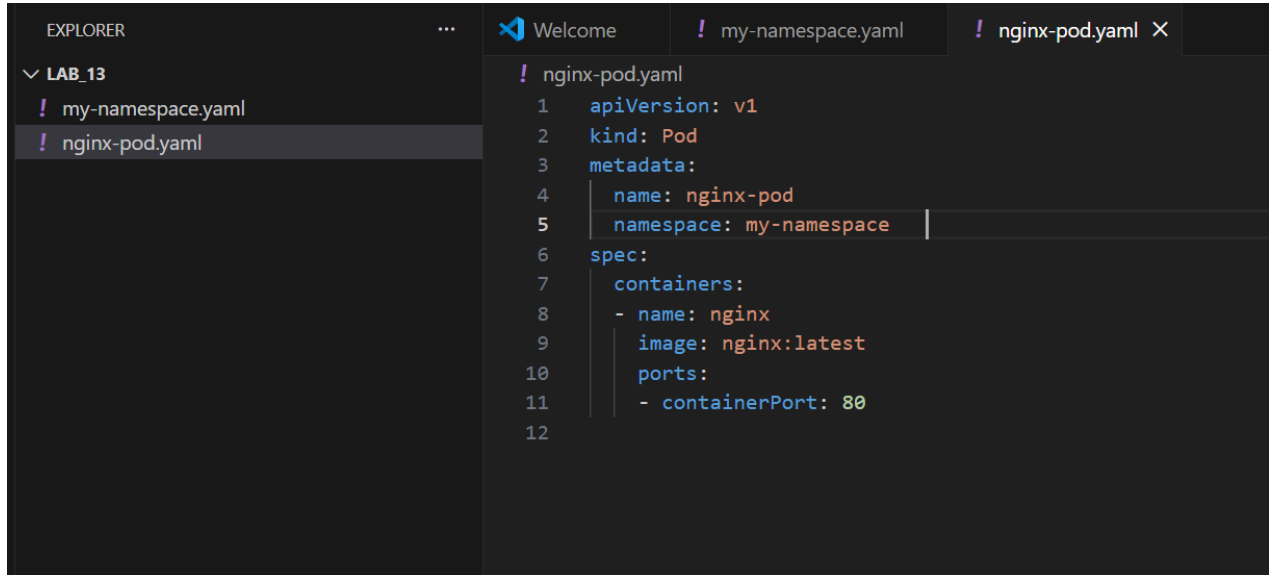
Step 4: Deploy Resources in a Namespace

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
```



```
EXPLORER
  LAB_13
    my-namespace.yaml
    nginx-pod.yaml

! nginx-pod.yaml
1  apiVersion: v1
2  kind: Pod
3  metadata:
4    name: nginx-pod
5    namespace: my-namespace
6  spec:
7    containers:
8    - name: nginx
9      image: nginx:latest
10     ports:
11     - containerPort: 80
12
```

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
```

```
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get pods -n my-namespace
NAME          READY   STATUS    RESTARTS   AGE
nginx-pod     1/1     Running   0           7s
```

To describe the Pod and see detailed information:

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

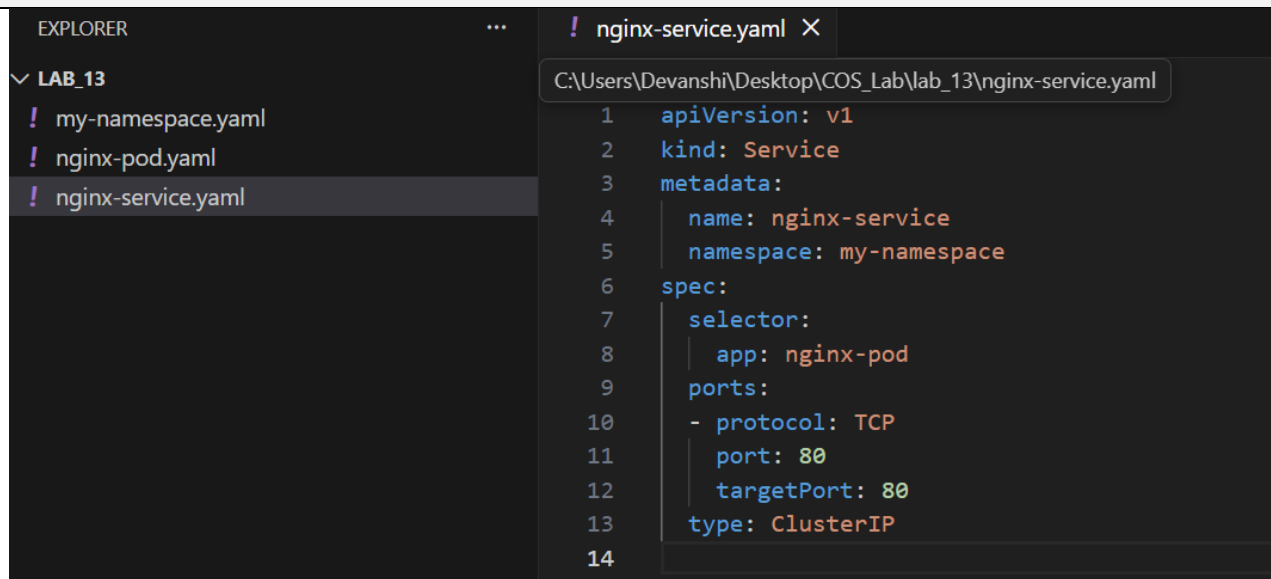
```
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl describe pod nginx-pod -n my-namespace
Name:          nginx-pod
Namespace:     my-namespace
Priority:       0
Service Account: default
Node:          docker-desktop/192.168.65.3
Start Time:    Tue, 17 Feb 2026 15:31:49 +0530
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:            10.1.0.32
IPs:
  IP: 10.1.0.32
Containers:
  nginx:
    Container ID:  docker://7cb285c22e20a5fe9a554fbc2e9f2b8772cd371408aab18c244803bad4eb5ae9
    Image:         nginx:latest
    Image ID:      docker-pullable://nginx@sha256:341bf0f3ce6c5277d6002cf6e1fb0319fa4252add24ab6a0e262e0056d313208
    Port:         80/TCP
    Host Port:    0/TCP
    State:        Running
      Started:    Tue, 17 Feb 2026 15:31:52 +0530
    Ready:        True
    Restart Count: 0
    Environment:  <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-sjlbj (ro)
Conditions:
  Type                               Status
  PodReadyToStartContainers         True
  Initialized                       True
  Ready                             True
  ContainersReady                   True
  PodScheduled                      True
Volumes:
  kube-api-access-sjlbj:
    Type:              Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:      kube-root-ca.crt
    ConfigMapOptional:  <nil>
    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   19s   default-scheduler  Successfully assigned my-namespace/nginx-pod to docker-desktop
  Normal  Pulling     18s   kubelet         Pulling image "nginx:latest"
```

- 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
```



```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: nginx-service
5    namespace: my-namespace
6  spec:
7    selector:
8      app: nginx-pod
9    ports:
10     - protocol: TCP
11       port: 80
12       targetPort: 80
13     type: ClusterIP
14
```

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\Devanshi\Desktop\COS_Lab\lab_13> kubectl apply -f nginx-service.yaml
service/nginx-service created
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get services -n my-namespace
NAME          TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
nginx-service ClusterIP     10.104.63.4   <none>         80/TCP     9s
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> 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:            10.104.63.4
IPs:           10.104.63.4
Port:          <unset> 80/TCP
TargetPort:    80/TCP
Endpoints:
Session Affinity: None
Internal Traffic Policy: Cluster
Events:        <none>
```

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 C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get pods -n my-namespace
NAME          READY   STATUS    RESTARTS   AGE
nginx-pod     1/1     Running   0           26m
```

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 C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl config set-context --current --namespace=my-namespace
Context "docker-desktop" modified.
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-pod     1/1     Running   0           27m
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get pods -n default
No resources found in default namespace.
```

Verify the current context's namespace:

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

```
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> 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
```

```
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl delete -f nginx-pod.yaml
pod "nginx-pod" deleted
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl delete -f nginx-service.yaml
service "nginx-service" deleted
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl delete namespace my-namespace
namespace "my-namespace" deleted
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl delete namespace my-namespace1
namespace "my-namespace1" deleted
```

Ensure that the namespace and all its resources are deleted:

```
kubectl get namespaces
```

```
PS C:\Users\Devanshi\Desktop\COS_Lab\lab_13> kubectl get namespaces
NAME              STATUS    AGE
default           Active    35m
kube-node-lease   Active    35m
kube-public        Active    35m
kube-system        Active    35m
```