Lab Exercise 9- Managing Namespaces in Kubernetes

Step 1: Understand Namespaces

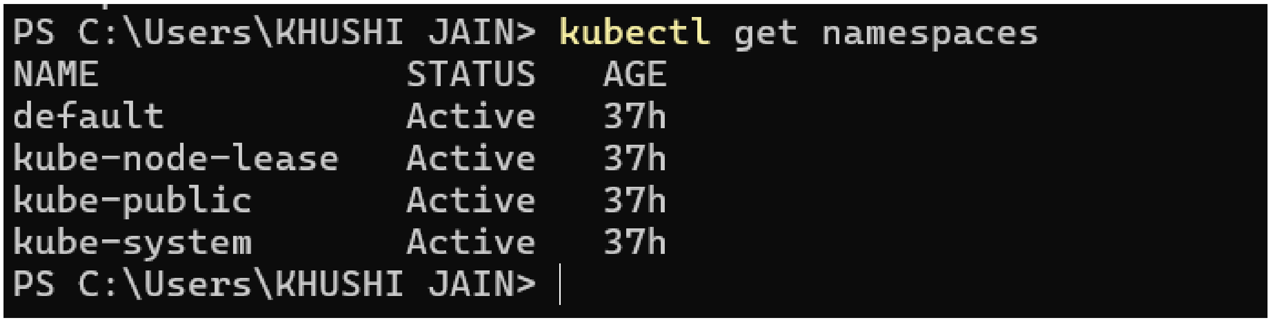
Namespaces provide a mechanism for scoping resources in a cluster. Namespaces can be used to:

- 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
```



```
PS C:\Users\KHUSHI JAIN> kubectl get namespaces
NAME              STATUS   AGE
default           Active   37h
kube-node-lease   Active   37h
kube-public       Active   37h
kube-system       Active   37h
PS C:\Users\KHUSHI JAIN>
```

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.

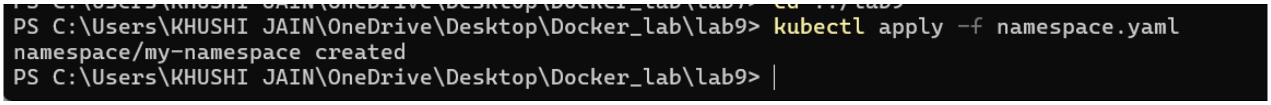
Using YAML File

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
```

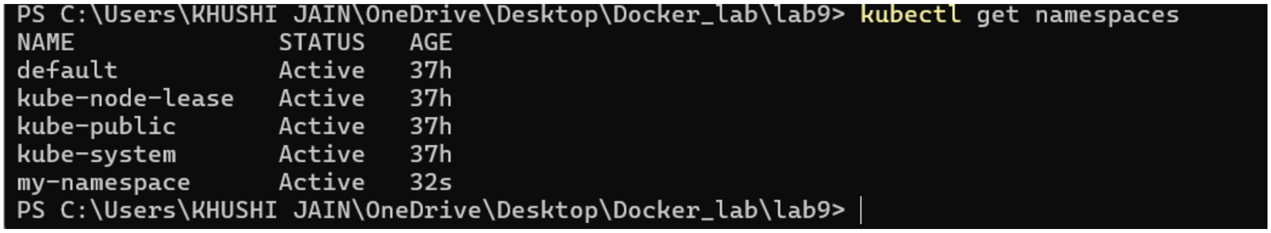


```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl apply -f namespace.yaml
namespace/my-namespace created
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> |
```

Verify that the namespace is created:

```
kubectl get namespaces
```

You should see my-namespace listed in the output.



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl get namespaces
NAME              STATUS   AGE
default           Active   37h
kube-node-lease   Active   37h
kube-public       Active   37h
kube-system       Active   37h
my-namespace      Active   32s
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> |
```

Step 4: Deploy Resources in a Namespace

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

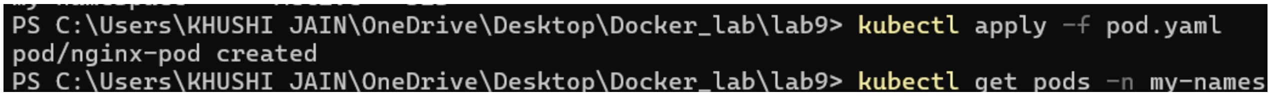
Deploy a Pod in the Namespace

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
```

Apply this YAML to create the Pod:

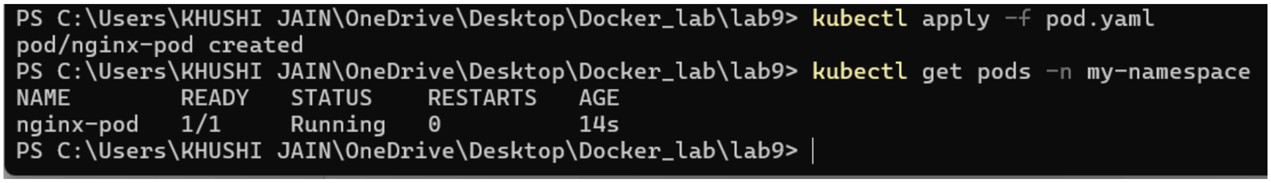
```
kubectl apply -f nginx-pod.yaml
```



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl apply -f pod.yaml
pod/nginx-pod created
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl get pods -n my-names
```

Check the status of the Pod within the namespace:

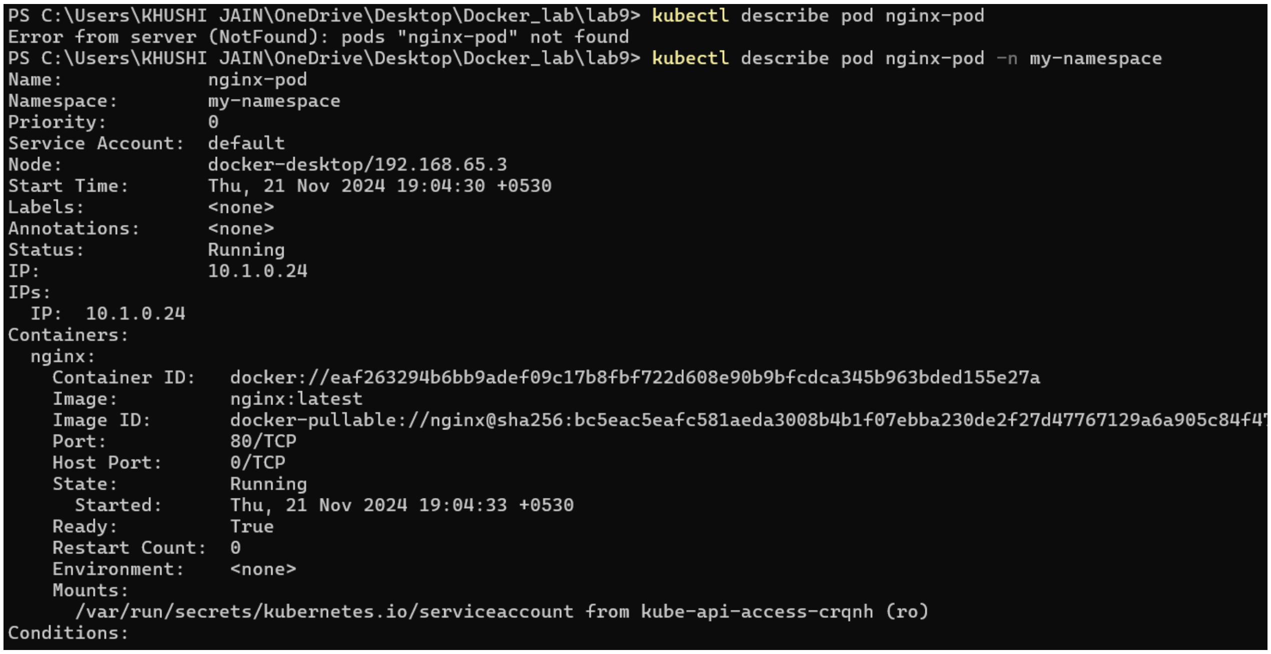
```
kubectl get pods -n my-namespace
```



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl apply -f pod.yaml
pod/nginx-pod created
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          14s
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> 
```

To describe the Pod and see detailed information:

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



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl describe pod nginx-pod
Error from server (NotFound): pods "nginx-pod" not found
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> 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:       Thu, 21 Nov 2024 19:04:30 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.1.0.24
IPs:
  IP:  10.1.0.24
Containers:
  nginx:
    Container ID:   docker://eaf263294b6bb9adef09c17b8fbf722d608e90b9bfcdca345b963bded155e27a
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f4
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 21 Nov 2024 19:04:33 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-crqnh (ro)
Conditions:
```

Create a Service in the Namespace

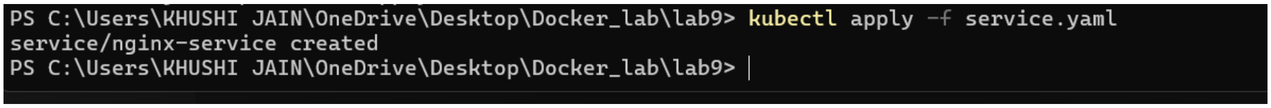
Create a YAML file named nginx-service.yaml with the following content:

```
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
```

Apply this YAML to create the Service:

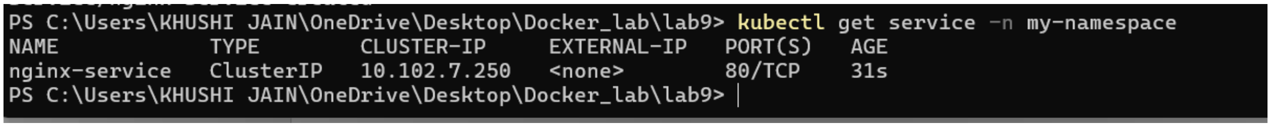
```
kubectl apply -f nginx-service.yaml
```



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl apply -f service.yaml
service/nginx-service created
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> |
```

Check the status of the Service within the namespace:

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



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl get service -n my-namespace
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.102.7.250   <none>        80/TCP    31s
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> |
```

To describe the Service and see detailed information:

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

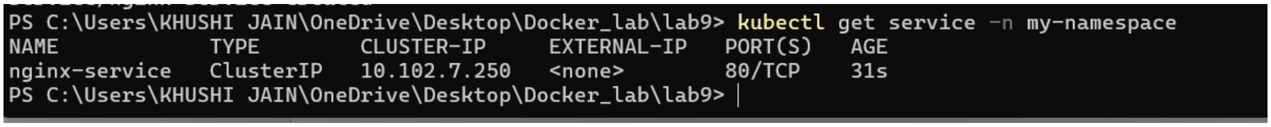
Step 5: Switching Context Between Namespaces

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
```

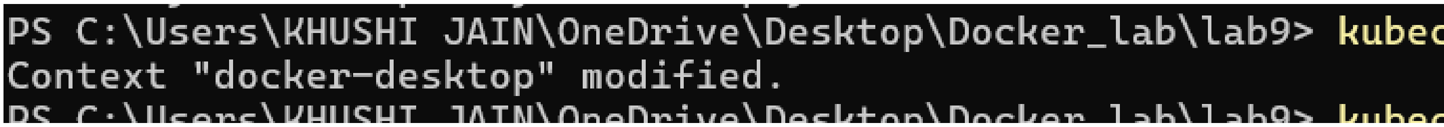


```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl get service -n my-namespace
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.102.7.250   <none>        80/TCP    31s
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> |
```

Set Default Namespace for kubectl Commands

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

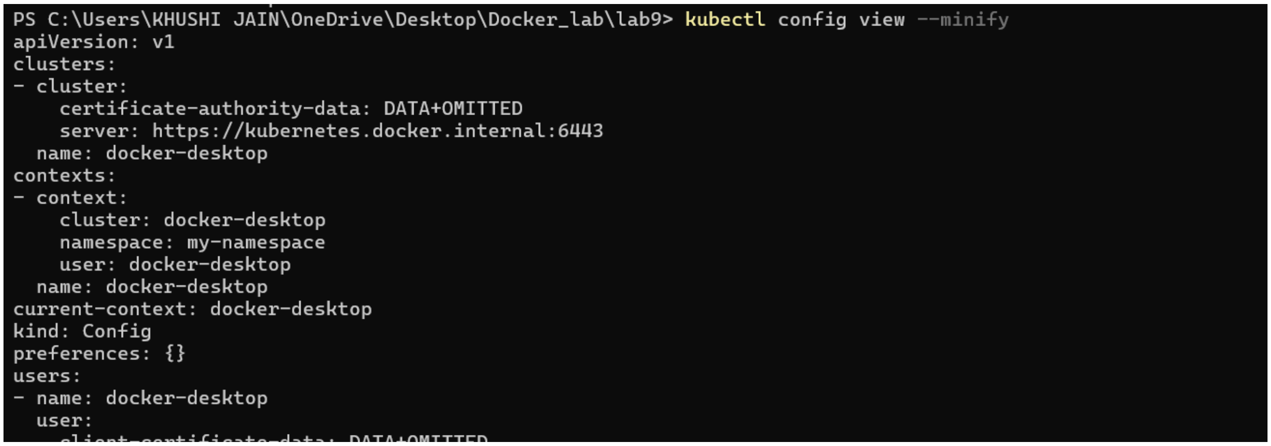
```
kubectl config set-context --current --namespace=my-namespace
```



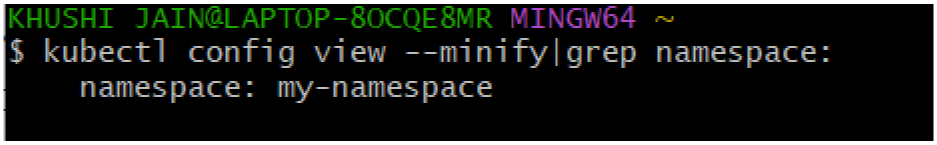
```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubec
Context "docker-desktop" modified.
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubec
```

Verify the current context’s namespace:

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



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl config view --minify
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://kubernetes.docker.internal:6443
  name: docker-desktop
contexts:
- context:
    cluster: docker-desktop
    namespace: my-namespace
    user: docker-desktop
  name: docker-desktop
current-context: docker-desktop
kind: Config
preferences: {}
users:
- name: docker-desktop
  user:
    client-certificate-data: DATA+OMITTED
```

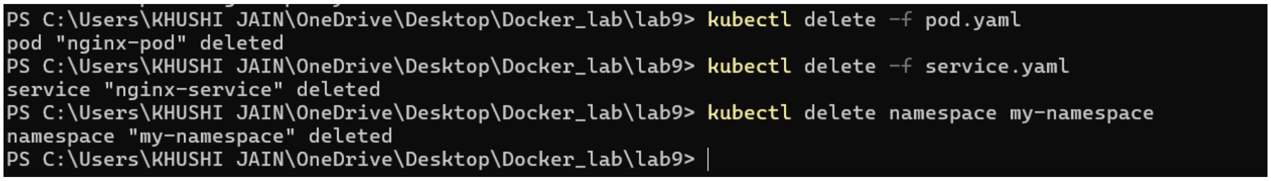


```
KHUSHI JAIN@LAPTOP-8OCQE8MR MINGW64 ~
$ kubectl config view --minify|grep namespace:
    namespace: my-namespace
```

Step 6: Clean Up Resources

To delete the resources and the namespace you created:

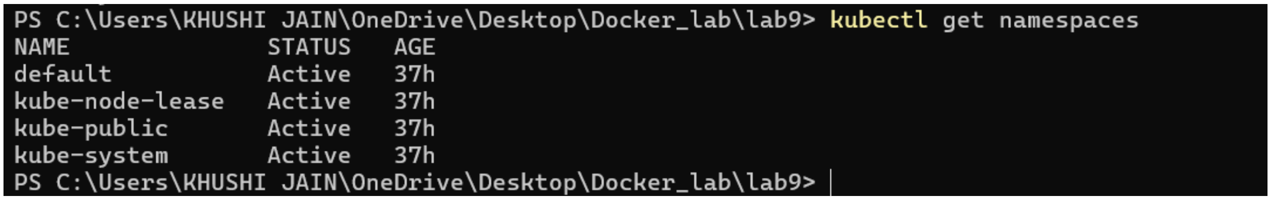
```
kubectl delete -f nginx-pod.yaml
kubectl delete -f nginx-service.yaml

kubectl delete namespace my-namespace
```



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl delete -f pod.yaml
pod "nginx-pod" deleted
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl delete -f service.yaml
service "nginx-service" deleted
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl delete namespace my-namespace
namespace "my-namespace" deleted
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9>
```

Ensure that the namespace and all its resources are deleted:

```
kubectl get namespaces
```



```
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9> kubectl get namespaces
NAME              STATUS   AGE
default           Active   37h
kube-node-lease   Active   37h
kube-public       Active   37h
kube-system       Active   37h
PS C:\Users\KHUSHI JAIN\OneDrive\Desktop\Docker_lab\lab9>
```