Lab Exercise 9- Managing Namespaces in Kubernetes

Step 1: Understand Namespaces

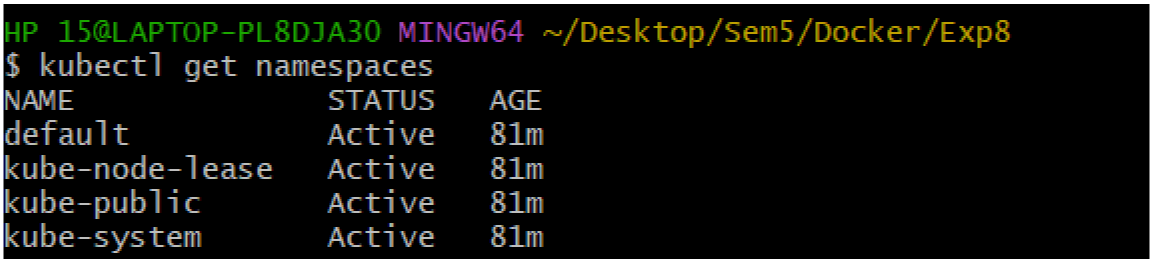
Namespaces provide a mechanism for scoping resources in a cluster. Namespaces can be used to:

- 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
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp8
$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   81m
kube-node-lease   Active   81m
kube-public       Active   81m
kube-system       Active   81m
```

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.

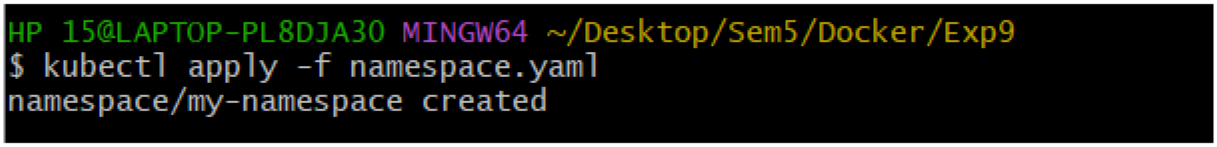
Using YAML File

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:

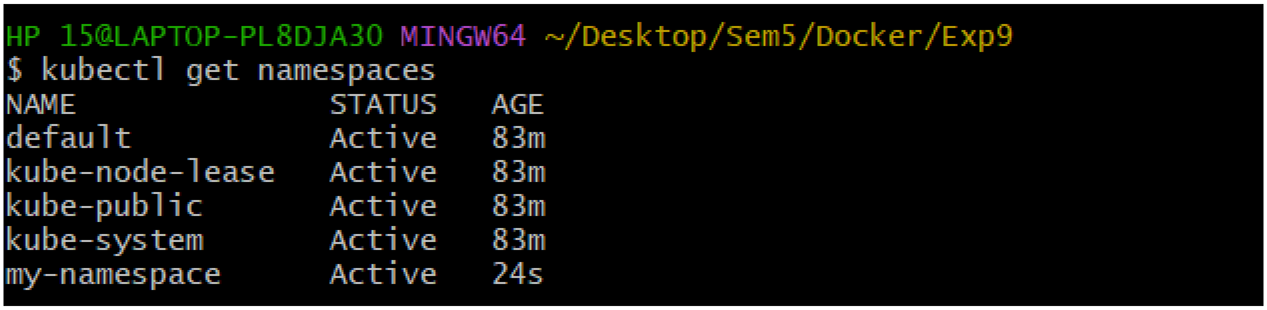
```
kubectl apply -f my-namespace.yaml
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl apply -f namespace.yaml
namespace/my-namespace created
```

Verify that the namespace is created:

```
kubectl get namespaces
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   83m
kube-node-lease   Active   83m
kube-public       Active   83m
kube-system       Active   83m
my-namespace      Active   24s
```

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.

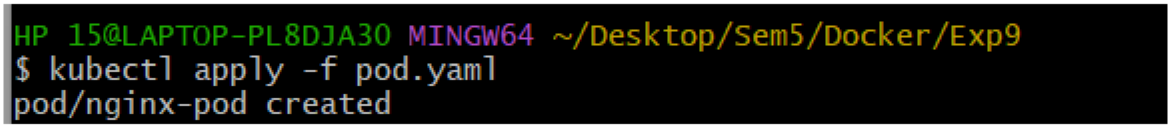
Deploy a Pod in the Namespace

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
```

Apply this YAML to create the Pod:

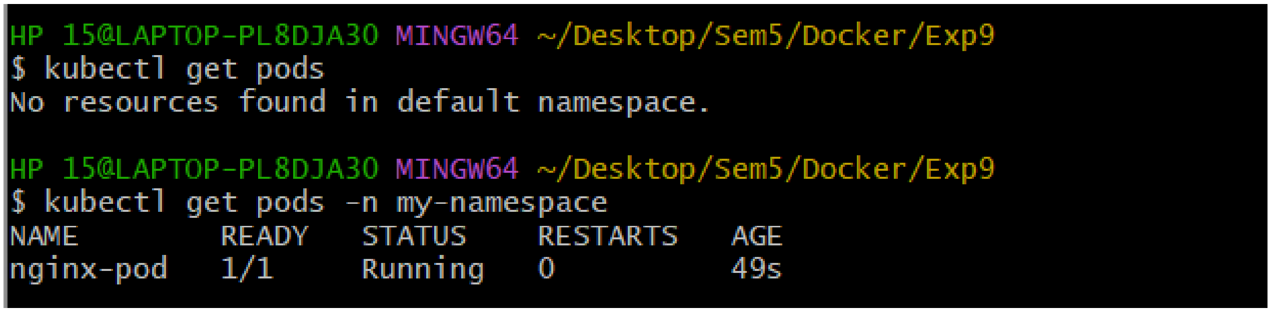
```
kubectl apply -f nginx-pod.yaml
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl apply -f pod.yaml
pod/nginx-pod created
```

Check the status of the Pod within the namespace:

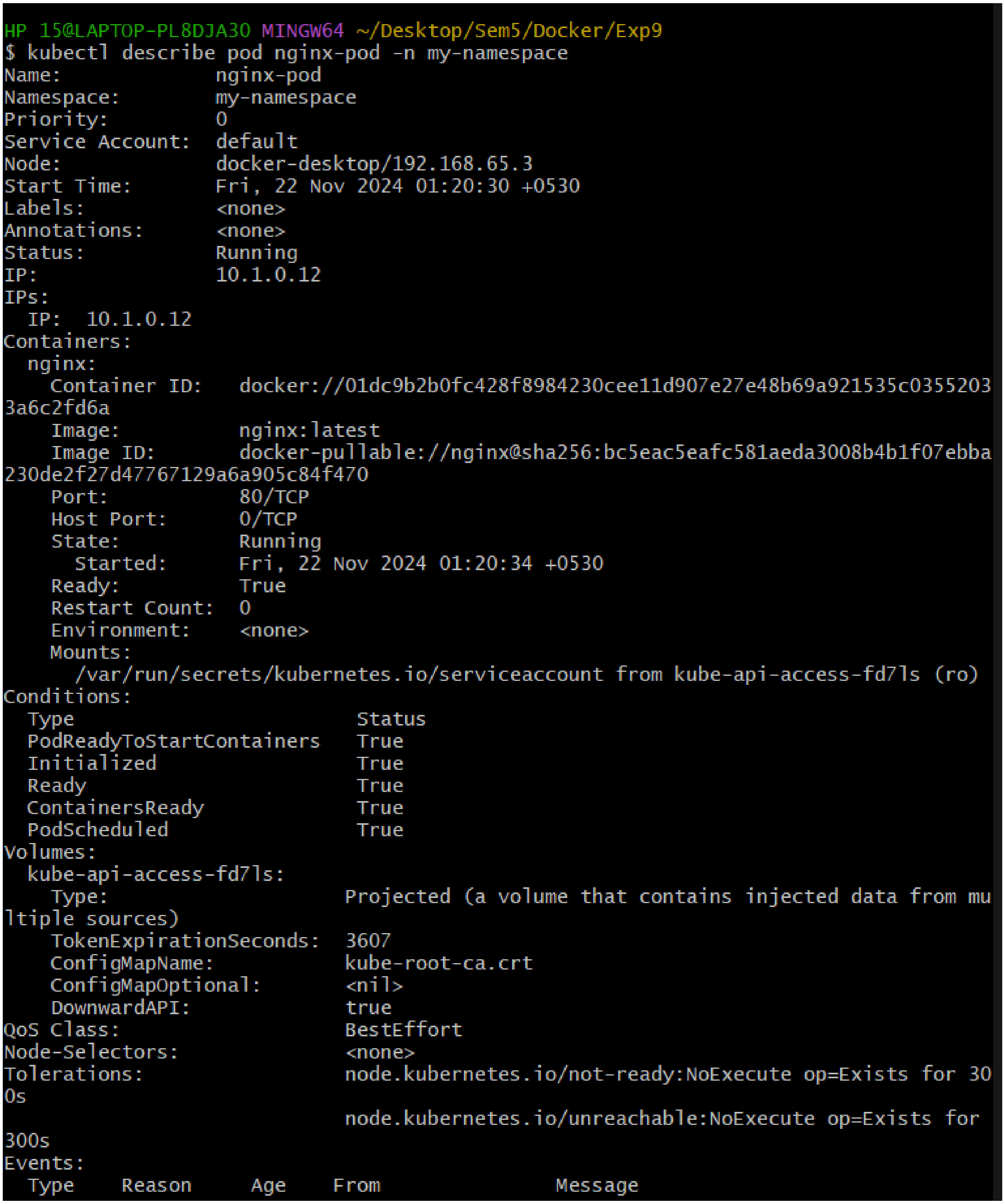
```
kubectl get pods -n my-namespace
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl get pods
No resources found in default namespace.

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          49s
```

To describe the Pod and see detailed information:

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



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ 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, 22 Nov 2024 01:20:30 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.1.0.12
IPs:
  IP:  10.1.0.12
Containers:
  nginx:
    Container ID:   docker://01dc9b2b0fc428f8984230cee11d907e27e48b69a921535c0355203
3a6c2fd6a
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba
230de2f27d47767129a6a905c84f470
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Fri, 22 Nov 2024 01:20:34 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fd7ls (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-fd7ls:
    Type:                    Projected (a volume that contains injected data from mu
ltiple 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 30
0s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for
300s
Events:
  Type    Reason     Age    From               Message
```

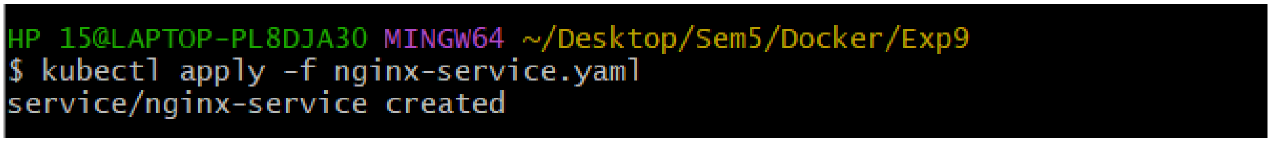
Create a Service in the Namespace

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:

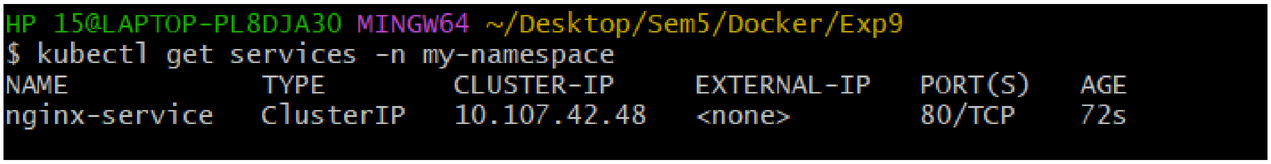
```
kubectl apply -f nginx-service.yaml
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl apply -f nginx-service.yaml
service/nginx-service created
```

Check the status of the Service within the namespace:

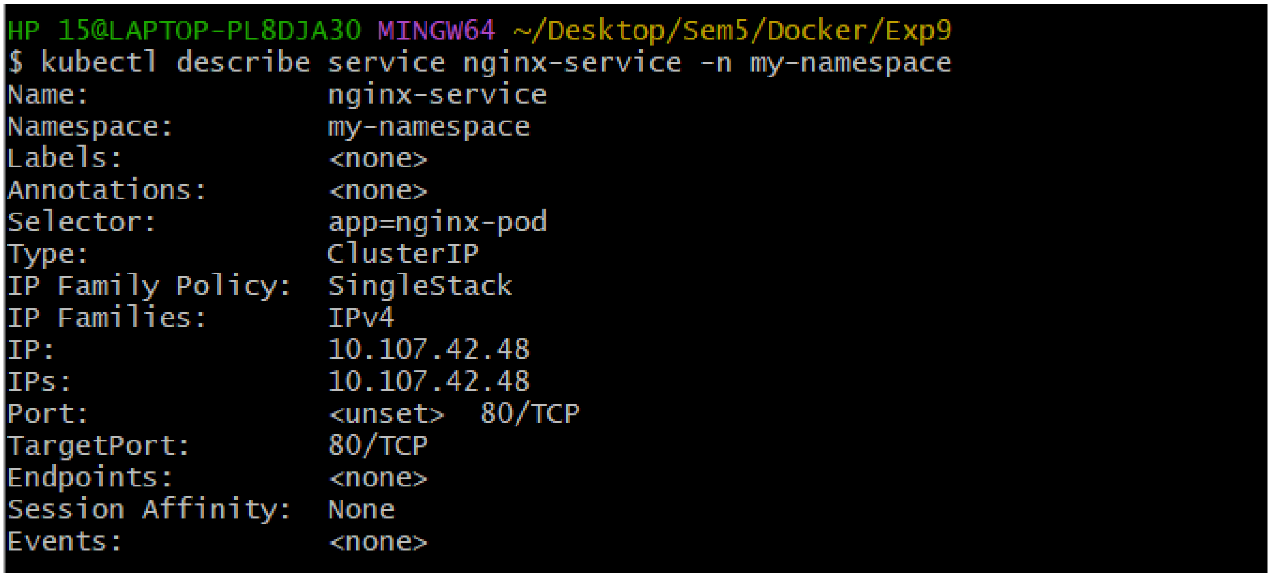
```
kubectl get services -n my-namespace
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl get services -n my-namespace
NAME            TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.107.42.48   <none>        80/TCP    72s
```

To describe the Service and see detailed information:

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



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ 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:                10.107.42.48
IPs:               10.107.42.48
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         <none>
Session Affinity:  None
Events:            <none>
```

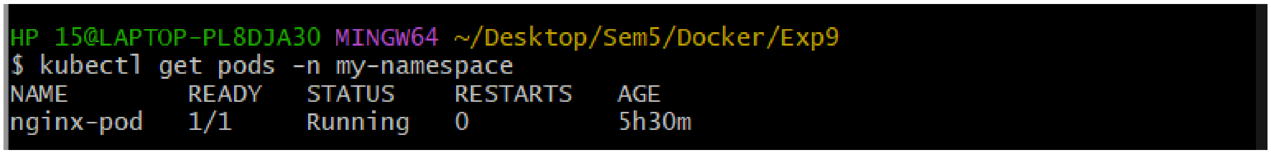
Step 5: Switching Context Between Namespaces

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
```

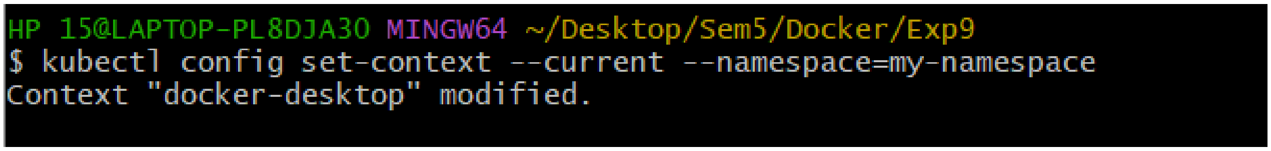


```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          5h30m
```

Set Default Namespace for kubectl Commands

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

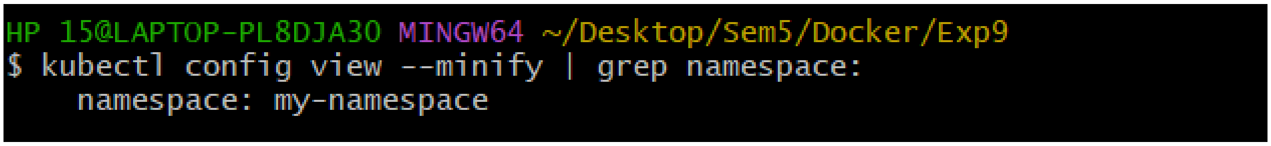
```
kubectl config set-context --current --namespace=my-namespace
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl config set-context --current --namespace=my-namespace
Context "docker-desktop" modified.
```

Verify the current context's namespace:

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

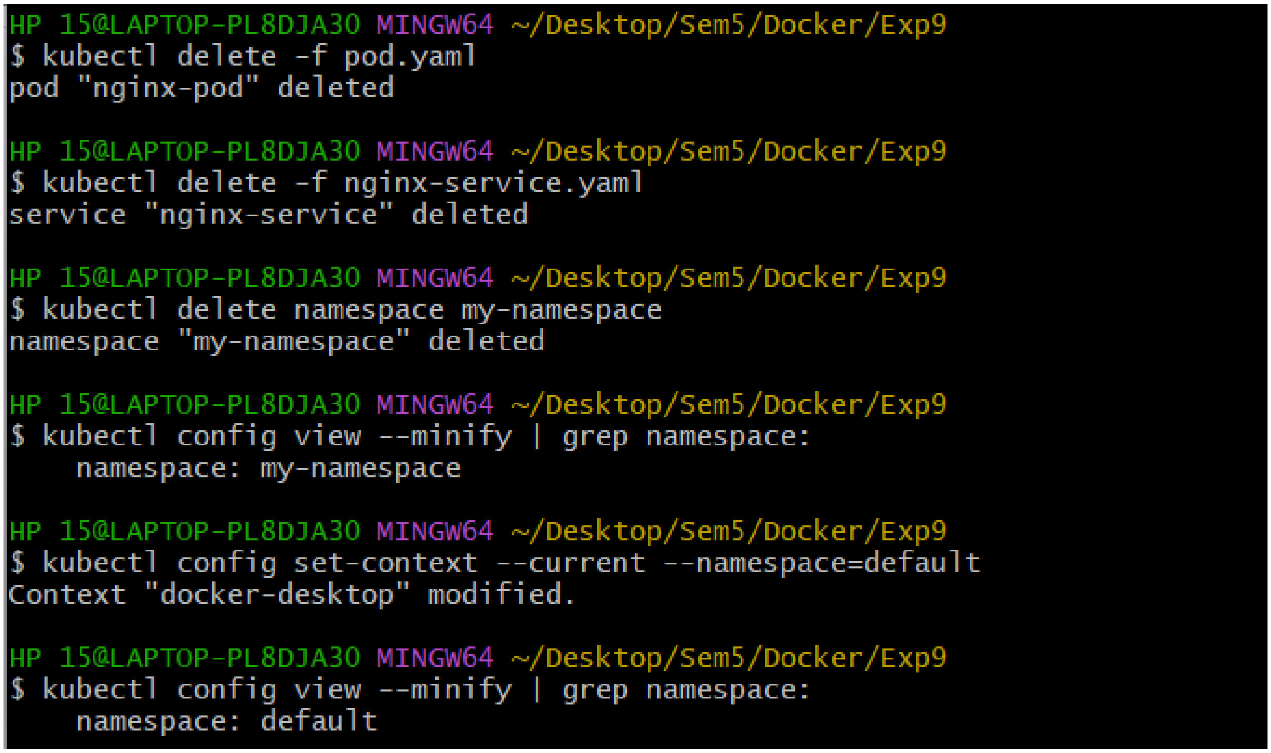


```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl config view --minify | grep namespace:
    namespace: my-namespace
```

Step 6: Clean Up Resources

To delete the resources and the namespace you created:

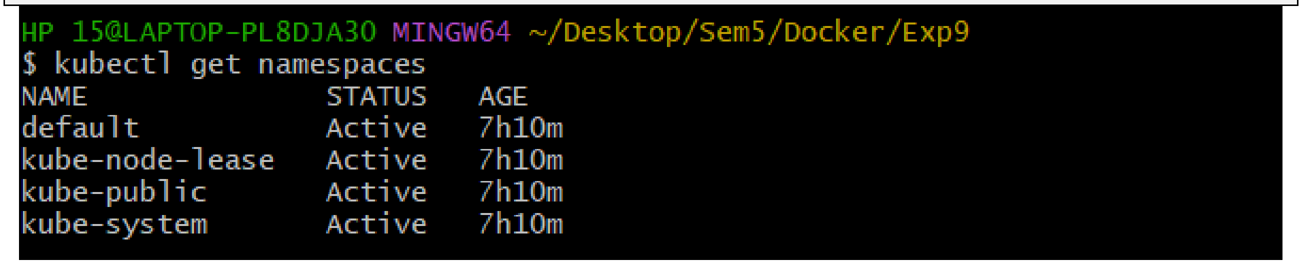
```
kubectl delete -f nginx-pod.yaml
kubectl delete -f nginx-service.yaml
kubectl delete namespace my-namespace
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl delete -f pod.yaml
pod "nginx-pod" deleted

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl delete -f nginx-service.yaml
service "nginx-service" deleted

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl delete namespace my-namespace
namespace "my-namespace" deleted

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl config view --minify | grep namespace:
    namespace: my-namespace

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl config set-context --current --namespace=default
Context "docker-desktop" modified.

HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl config view --minify | grep namespace:
    namespace: default
```

Ensure that the namespace and all its resources are deleted:

```
kubectl get namespaces
```



```
HP 15@LAPTOP-PL8DJA30 MINGW64 ~/Desktop/Sem5/Docker/Exp9
$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   7h10m
kube-node-lease   Active   7h10m
kube-public       Active   7h10m
kube-system       Active   7h10m
```