Lab Exercise 9- Managing Namespaces in Kubernetes

Name – Sujal Bhandari
SAP_ID – 500106865
Roll_No – R2142220181

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\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl get namespaces
NAME              STATUS   AGE
default           Active   27d
kube-node-lease   Active   27d
kube-public       Active   27d
kube-system       Active   27d
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

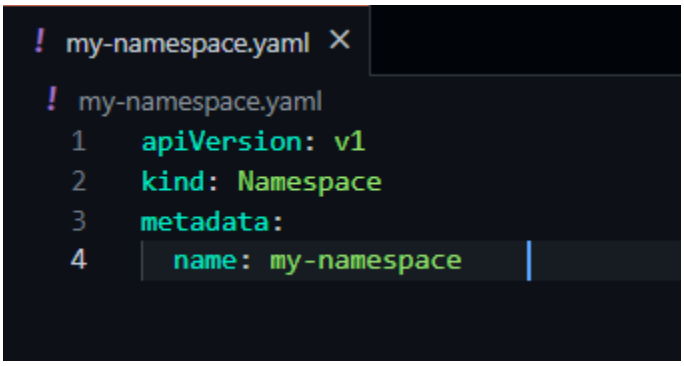
You will typically see default namespaces like default, kube-system, and kube-public.

Step 3: Create a Namespace

You can create a namespace using a YAML file or directly with the kubectl command.

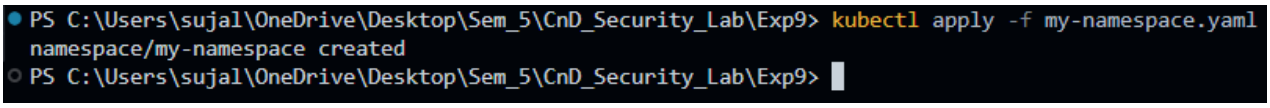
Using YAML File
Create a file named ***my-namespace.yaml*** with the following content:

```
apiVersion: v1
kind: Namespace
metadata:
  name: my-namespace
```



Apply this YAML to create the namespace:

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



Verify that the namespace is created:

```
kubectl get namespaces
```

```
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl get namespaces
NAME              STATUS   AGE
default           Active   27d
kube-node-lease   Active   27d
kube-public       Active   27d
kube-system       Active   27d
my-namespace      Active   19s
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

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
```

```yaml
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\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl apply -f nginx-pod.yaml
pod/nginx-pod created
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

Check the status of the Pod within the namespace:

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

```
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          16s
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

To describe the Pod and see detailed information:

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

```
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\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:       Thu, 21 Nov 2024 17:01:15 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.1.0.16
IPs:
  IP:  10.1.0.16
Containers:
  nginx:
    Container ID:   docker://c2331a06b3a903316569c7d9d5c78ca225be6b19fe3c0af498b6d2d906e5da47
    Image:          nginx:latest
    Image ID:       docker-pullable://nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d47767129a6a905c84f470
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 21 Nov 2024 17:01:19 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-4xbxn (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-4xbxn:
    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  37s   default-scheduler  Successfully assigned my-namespace/nginx-pod to docker-desktop
  Normal  Pulling    37s   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   # Specify the namespace for the Service.
spec:
  selector:
```

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

```yaml
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\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl apply -f nginx-service.yaml
service/nginx-service created
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

Check the status of the Service within the namespace:

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

```
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl get services -n my-namespace
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.111.164.19   <none>        80/TCP    24s
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

To describe the Service and see detailed information:

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

```
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\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:                10.111.164.19
IPs:               10.111.164.19
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         <none>
Session Affinity:  None
Events:            <none>
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

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\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          3m34s
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

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\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9> kubectl config set-context --current --namespace=my-namespace
Context "docker-desktop" modified.
PS C:\Users\sujal\OneDrive\Desktop\Sem_5\CnD_Security_Lab\Exp9>
```

Verify the current context's namespace:

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

```
sujal@HP-Victus MINGW64 ~/OneDrive/Desktop/Sem_5/CnD_Security_Lab/Exp9
$ kubectl config view --minify | grep namespace:
    namespace: my-namespace

sujal@HP-Victus MINGW64 ~/OneDrive/Desktop/Sem_5/CnD_Security_Lab/Exp9
$
```

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
```

```
sujal@HP-Victus MINGW64 ~/OneDrive/Desktop/Sem_5/CnD_Security_Lab/Exp9
$ kubectl delete -f nginx-pod.yaml
kubectl delete -f nginx-service.yaml
kubectl delete namespace my-namespace
pod "nginx-pod" deleted
service "nginx-service" deleted
namespace "my-namespace" deleted
```

Ensure that the namespace and all its resources are deleted:

```
kubectl get namespaces
```

```
sujal@HP-Victus MINGW64 ~/OneDrive/Desktop/Sem_5/CnD_Security_Lab/Exp9
$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   27d
kube-node-lease   Active   27d
kube-public       Active   27d
kube-system       Active   27d

sujal@HP-Victus MINGW64 ~/OneDrive/Desktop/Sem_5/CnD_Security_Lab/Exp9
$
```