Containers & Docker Security LAB

SUBMITTED TO
Dr. Hitesh Kumar Sharma

SUBMITTED BY
Siddharth Agarwal
500107594
R2142220663
Btech CSE DevOps B1

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
```

```
sidag@sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES
CDS/lab/exp9
$ kubectl get namespaces
NAME                   STATUS   AGE
default                Active   60m
kube-node-lease        Active   60m
kube-public            Active   60m
kube-system            Active   60m
local-path-storage     Active   59m
```

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

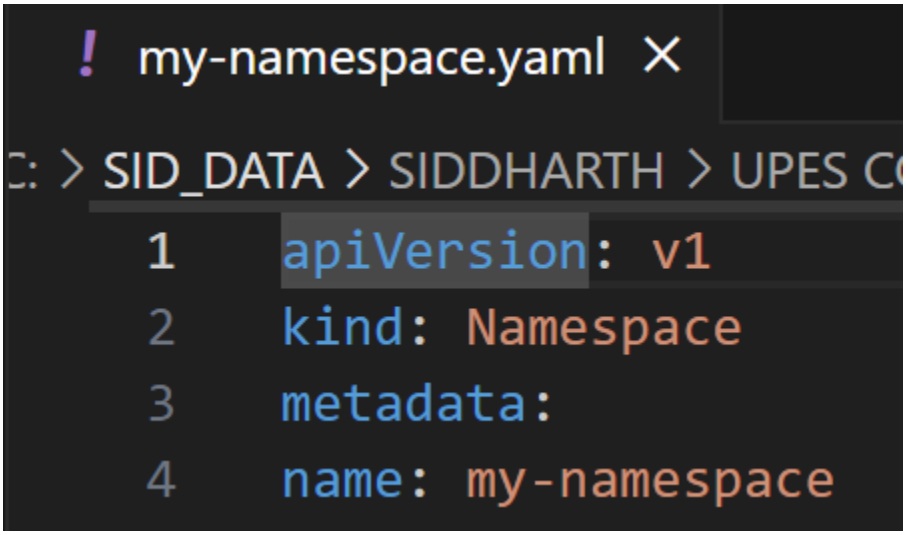
Step 3: Create a Namespace

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

Using YAML File

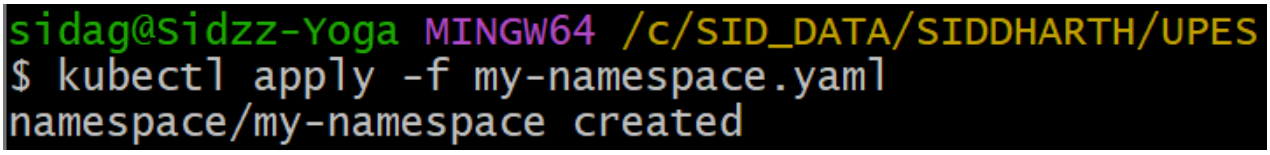
Create a file named ***my-namespace.yaml*** with the following content:

```
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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES
$ kubectl get namespaces
NAME                   STATUS   AGE
default                Active   66m
kube-node-lease        Active   66m
kube-public            Active   66m
kube-system            Active   66m
local-path-storage     Active   66m
my-namespace           Active   32s
```

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
```

nginx-pod.yaml

C: > SID_DATA > SIDDHARTH > UPES COLLEGE STUDY MATERIAL > SEM5 > CDS > lab > exp9 >

```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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES
$ 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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES
$ kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          28s
```

To describe the Pod and see detailed information:

kubectl describe pod nginx-pod -n my-namespace

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES COLLEGE STUDY MATERIAL/SEM5/CDS/lab/exp9
$ kubectl describe pod nginx-pod -n my-namespace
Name:             nginx-pod
Namespace:        my-namespace
Priority:         0
Service Account:  default
Node:             kind-control-plane/172.18.0.2
Start Time:       Thu, 21 Nov 2024 20:11:57 +0530
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.0.11
IPs:
  IP:  10.244.0.11
Containers:
  nginx:
    Container ID:   containerd://de4ca05e75cc0d9f3dfaeee258c760c70a52785f3f138b09412135a355bba041
    Image:          nginx:latest
    Image ID:       docker.io/library/nginx@sha256:bc5eac5eafc581aeda3008b4b1f07ebba230de2f27d477671
29a6a905c84f470
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 21 Nov 2024 20:11:59 +0530
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-wp5nm (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  kube-api-access-wp5nm:
    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  68s   default-scheduler  Successfully assigned my-namespace/nginx-pod to kind-c
ontrol-plane
  Normal  Pulling    68s   kubelet            Pulling image "nginx:latest"
  Normal  Pulled     66s   kubelet            Successfully pulled image "nginx:latest" in 2.059s (2.
059s including waiting). Image size: 72955450 bytes.
  Normal  Created    66s   kubelet            Created container nginx
  Normal  Started    66s   kubelet            Started container nginx
```

Create a Service in the Namespace

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

```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
```

nginx-service.yaml

C: > SID_DATA > SIDDHARTH > UPES COLLEGE STUDY MATERIAL > SE

```yaml
apiVersion: v1
kind: Service
metadata:
    name: nginx-service
    namespace: my-namespace   # Specify the
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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/
$ 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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES COLLEGE STUDY MATERIAL/
$ kubectl get services -n my-namespace
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
nginx-service   ClusterIP   10.96.250.209   <none>        80/TCP    42s
```

To describe the Service and see detailed information:

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

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES COLLEGE S
$ 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:                       10.96.250.209
IPs:                      10.96.250.209
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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES
$ kubectl get pods -n my-namespace
NAME        READY   STATUS    RESTARTS   AGE
nginx-pod   1/1     Running   0          8m12s
```

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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES COLLEGE STUDY
$ kubectl config set-context --current --namespace=my-namespace
Context "kind-kind" modified.
```

Verify the current context's namespace:

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

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES COLL
$ 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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SIDDHARTH/UPES
$ 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
```

```
sidag@Sidzz-Yoga MINGW64 /c/SID_DATA/SID
$ kubectl get namespaces
NAME                   STATUS   AGE
default                Active   82m
kube-node-lease        Active   82m
kube-public            Active   82m
kube-system            Active   82m
local-path-storage     Active   82m
```