

## Task

1. Create two namespaces and name them ns1 and ns2

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl apply -f D:/Github/CKA2024/Day-10/ns_test.yaml
namespace/ns1 created
```

```
Day-10 > ! ns_test.yaml > {} metadata
      io.k8s.api.core.v1.Namespace (v1@namespace.json)
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: ns1
5
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl create ns ns2
namespace/ns2 created
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl get ns
NAME                STATUS    AGE
default             Active   6d16h
demo                Active   5d23h
kube-node-lease     Active   6d16h
kube-public         Active   6d16h
kube-system         Active   6d16h
local-path-storage  Active   6d16h
ns1                 Active   4m59s
ns2                 Active   13s
```

2. Create a deployment with a single replica in each of these namespaces with the image as nginx and name as deploy-ns1 and deploy-ns2, respectively

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl create deploy deploy-ns1 --image=nginx -n ns1
deployment.apps/deploy-ns1 created
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl get deploy -n ns1
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
deploy-ns1  1/1     1             1           17s
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl create deploy deploy-ns2 --image=nginx -n ns2
deployment.apps/deploy-ns2 created
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl get deploy -n ns2
NAME        READY   UP-TO-DATE   AVAILABLE   AGE
deploy-ns2  1/1     1             1           19s
```

- Get the IP address of each of the pods (Remember the kubectl command for that?)

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl get pods -n ns1 -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
deploy-ns1-5d965bfsfd-6vk8v        1/1     Running   0           10m   10.244.2.6
ka-cluster3-worker                 <none>   <none>     <none>     <none>
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
• $ kubectl get pod -n ns2 -o wide
NAME                                READY   STATUS    RESTARTS   AGE   IP
deploy-ns2-587b785459-gc9rz        1/1     Running   0           10m   10.244.2.7
cka-cluster3-worker                 <none>   <none>     <none>     <none>
```

### 3. Exec into the pod of deploy-ns1 and try to curl the IP address of the pod running on deploy-ns2

- Your pod-to-pod connection should work, and you should be able to get a successful response back.

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl exec -it deploy-ns1-5d965bf5fd-6vk8v -n ns1
-- sh
# curl 10.244.2.7
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
# exit
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl exec -it deploy-ns2-587b785459-gc9rz -n
ns2 -- sh
# curl 10.244.2.6
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
# exit
```

### 4. Now scale both of your deployments from 1 to 3 replicas.

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl scale --replicas=3 deploy/deploy-ns1 -n ns1
deployment.apps/deploy-ns1 scaled
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl scale --replicas=3 deploy/deploy-ns2 -n
ns2
deployment.apps/deploy-ns2 scaled
```

### 5. Create two services to expose both of your deployments and name them svc-ns1 and svc-ns2

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl get pods -n ns1
NAME                                READY    STATUS    RESTARTS    AGE
deploy-ns1-5d965bf5fd-5l8mb         1/1      Running   0            60s
deploy-ns1-5d965bf5fd-6vk8v         1/1      Running   0            15m
deploy-ns1-5d965bf5fd-lkk8b         1/1      Running   0            60s

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl expose deploy/deploy-ns1 --name=svc-ns1 --po
rt=80 -n ns1
service/svc-ns1 exposed
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl get pods -n ns2
NAME                                READY    STATUS    RESTARTS    AGE
deploy-ns2-587b785459-9gqqb         1/1      Running   0            38s
deploy-ns2-587b785459-gc9rz         1/1      Running   0            14m
deploy-ns2-587b785459-xcnmg         1/1      Running   0            38s

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl expose deploy/deploy-ns2 --name=svc-ns2
--port=80 -n ns2
service/svc-ns2 exposed
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl get pod -n ns1 -o wide
NAME                                READY    STATUS    RESTARTS    AGE    IP
deploy-ns1-5d965bf5fd-5l8mb         1/1      Running   0            25m    10.244.1.7
cka-cluster3-worker2                <none>    <none>     0            40m    10.244.2.6
deploy-ns1-5d965bf5fd-6vk8v         1/1      Running   0            25m    10.244.1.8
cka-cluster3-worker2                <none>    <none>     0            40m    10.244.2.6

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl get svc -n ns1
NAME    TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
svc-ns1 ClusterIP    10.96.112.126 <none>         80/TCP     11m
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl get pod -n ns2 -o wide
NAME                                READY    STATUS    RESTARTS    AGE    IP    NODE
deploy-ns2-587b785459-9gqqb         1/1      Running   0            25m    10.244.2.8    cka
-cluster3-worker                    <none>    <none>     0            39m    10.244.2.7    cka
deploy-ns2-587b785459-gc9rz         1/1      Running   0            25m    10.244.1.9    cka
-cluster3-worker                    <none>    <none>     0            40m    10.244.2.6    cka
deploy-ns2-587b785459-xcnmg         1/1      Running   0            25m    10.244.1.9    cka
-cluster3-worker2                  <none>    <none>     0            40m    10.244.2.6    cka

Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
● $ kubectl get svc -n ns2
NAME    TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)    AGE
svc-ns2 ClusterIP    10.96.72.248  <none>         80/TCP     10m
```

6. exec into each pod and try to curl the IP address of the service running on the other namespace.

- This curl should work.

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl exec -it deploy-ns1-5d965bf5fd-5l8mb -n ns1
-- sh
# curl 10.96.72.248
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

```
Bhakti@LAPTOP-DNC3NQI0 MINGW64 /d/Github/CKA2024 (main)
$ kubectl exec -it deploy-ns2-587b785459-9gqqb -n ns2 -- sh
# curl 10.96.112.126
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

</body>
</body>
</body>
</html>
```

7. Now try curling the service name instead of IP. (You will notice that you are getting an error and cannot resolve the host.)

- Now use the FQDN of the service and try to curl again, this should work.

```
# cat /etc/resolv.conf
search ns1.svc.cluster.local svc.cluster.local cluster.local
nameserver 10.96.0.10
options ndots:5
```

```
# curl svc-ns2.ns2.svc.cluster.local
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
#
```

```
# curl svc-ns1.ns1.svc.cluster.local
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
#
```

8. In the end, delete both the namespaces, which should delete the services and deployments underneath them.