

Lab Exercise 7 - Start and Access Kubernetes Dashboard

Objective

To enable Kubernetes in Docker Desktop, deploy the Kubernetes Dashboard, and access it securely using a web browser on Windows.

Prerequisites

- Windows 10 / 11
 - Docker Desktop installed
 - Docker Desktop Kubernetes enabled
 - Internet connection
 - kubectl (comes bundled with Docker Desktop)
-

Step 1: Enable Kubernetes in Docker Desktop

1. Open **Docker Desktop**
2. Go to **Settings**
3. Select **Kubernetes**
4. Check **Enable Kubernetes**
5. Click **Apply & Restart**

Wait until Kubernetes status shows **Running** (green).

Step 2: Verify Kubernetes Cluster

Open **PowerShell** or **Command Prompt** and run:

- `kubectl version --client`
- Check cluster status:
- `kubectl cluster-info`

Check nodes:

```
kubectl get nodes
```

Expected output:

Node status should be **Ready**

```
PS C:\Users\Devanshi> kubectl version --client
Client Version: v1.32.2
Kustomize Version: v5.5.0
PS C:\Users\Devanshi> kubectl cluster-info
Kubernetes control plane is running at https://kubernetes.docker.internal:6443
CoreDNS is running at https://kubernetes.docker.internal:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS C:\Users\Devanshi> kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
docker-desktop      Ready     control-plane  3m45s  v1.32.2
```

Step 3: Deploy Kubernetes Dashboard

Apply the official Kubernetes Dashboard manifest:

```
kubectl apply -f
```

```
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```

Verify namespace creation:

```
kubectl get ns
```

You should see:

```
kubernetes-dashboard
```

```
PS C:\Users\Devanshi> kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
PS C:\Users\Devanshi> kubectl get ns
NAME                STATUS   AGE
default              Active   5m11s
kube-node-lease      Active   5m11s
kube-public           Active   5m11s
kube-system           Active   5m11s
kubernetes-dashboard Active   10s
```

Step 4: Verify Dashboard Pods

Check dashboard pods:

```
kubectl get pods -n kubernetes-dashboard
```

```
PS C:\Users\Devanshi> kubectl get pods -n kubernetes-dashboard
NAME                                READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-5bd45c9dd6-bqbxp   1/1     Running   0           106s
kubernetes-dashboard-79cbcf9fb6-46r7d        1/1     Running   0           106s
```

Expected status:

Running

Step 5: Create Admin User for Dashboard Access

Create a service account:

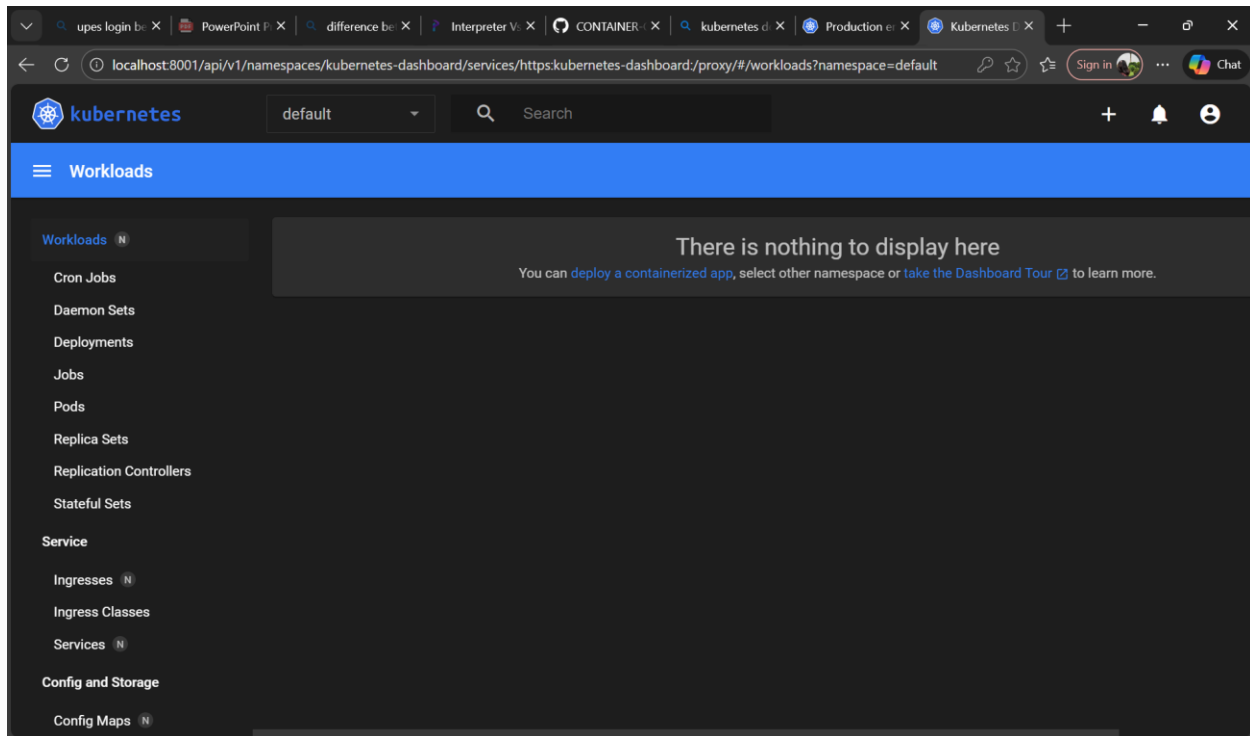
```
kubectl create serviceaccount dashboard-admin -n kubernetes-dashboard
```

```
PS C:\Users\Devanshi> kubectl create serviceaccount dashboard-admin -n kubernetes-dashboard
serviceaccount/dashboard-admin created
```

Create cluster role binding:

```
kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin
--serviceaccount=kubernetes-dashboard:dashboard-admin
```

```
PS C:\Users\Devanshi> kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin --serviceaccount=kubernetes-dashboard:dashboard-admin
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-binding created
```

Step 10: Explore Dashboard

You can now view:

- Nodes
- Pods
- Deployments
- Services
- Namespaces
- ConfigMaps and Secrets