**Pratik Agrawal**

**500123601**

**Devops B2**

# Lab Exercise 12 - 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).

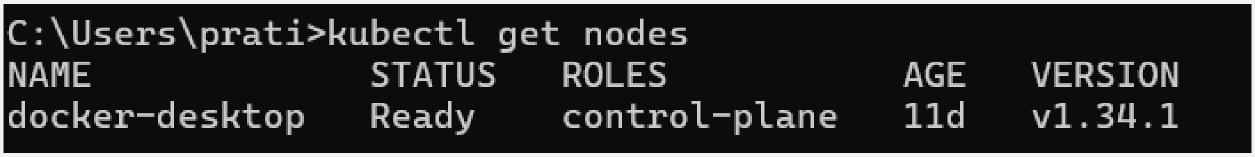
### Step 2: Verify Kubernetes Cluster

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

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

Check nodes:

```
kubectl get nodes
```



```
C:\Users\prati>kubectl get nodes
NAME             STATUS   ROLES           AGE   VERSION
docker-desktop   Ready    control-plane   11d   v1.34.1
```

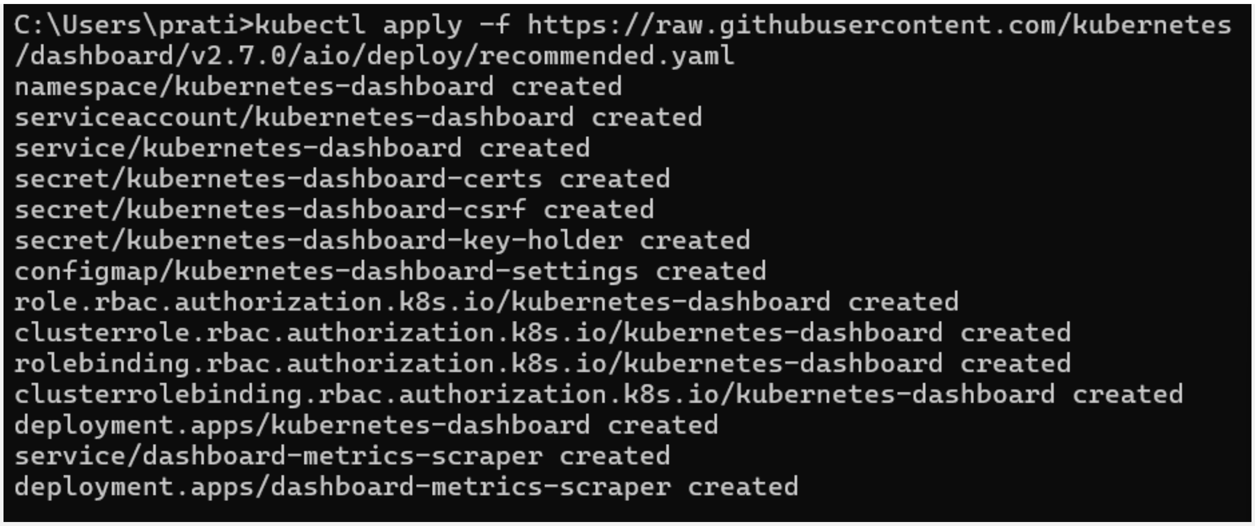
Expected output:

Node status should be **Ready**

### Step 3: Deploy Kubernetes Dashboard

Apply the official Kubernetes Dashboard manifest:

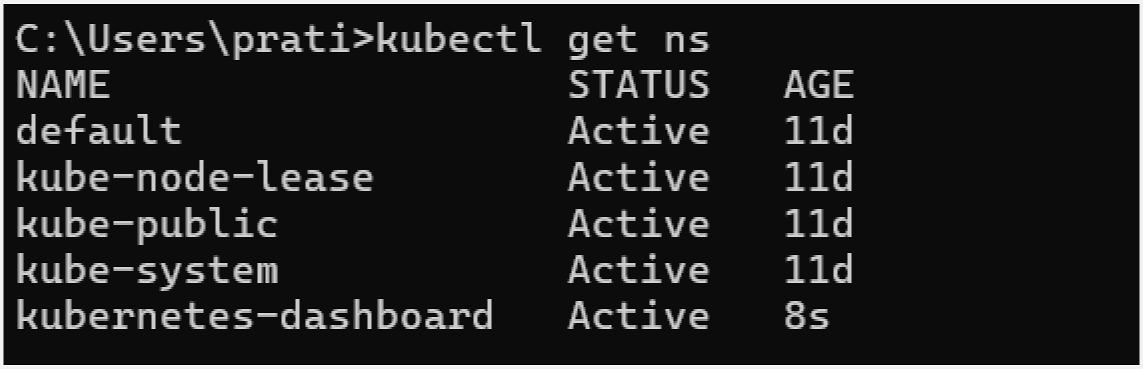
```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```



```
C:\Users\prati>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
```

Verify namespace creation:

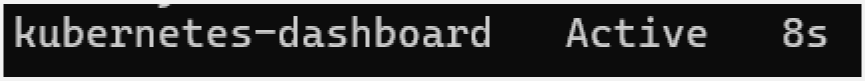
```
kubectl get ns
```



```
C:\Users\prati>kubectl get ns
NAME                   STATUS   AGE
default                Active   11d
kube-node-lease        Active   11d
kube-public            Active   11d
kube-system            Active   11d
kubernetes-dashboard   Active   8s
```

You should see:

```
kubernetes-dashboard
```

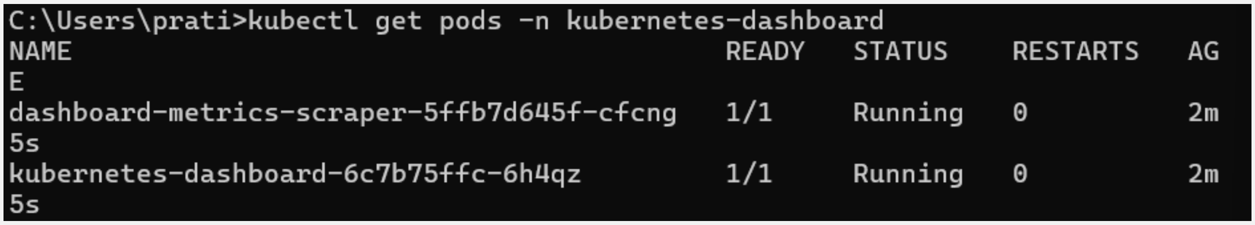


```
kubernetes-dashboard   Active   8s
```

### Step 4: Verify Dashboard Pods

Check dashboard pods:

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



```
C:\Users\prati>kubectl get pods -n kubernetes-dashboard
NAME                                         READY   STATUS    RESTARTS   AG
E
dashboard-metrics-scraper-5ffb7d645f-cfcng   1/1     Running   0          2m
5s
kubernetes-dashboard-6c7b75ffc-6h4qz         1/1     Running   0          2m
5s
```

Expected status:

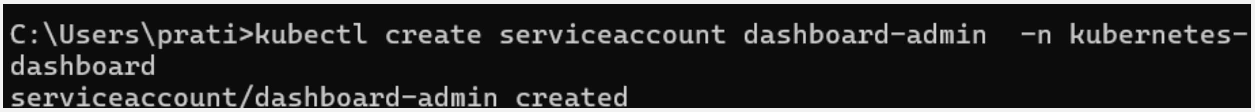
Running

---

### Step 5: Create Admin User for Dashboard Access

Create a service account:

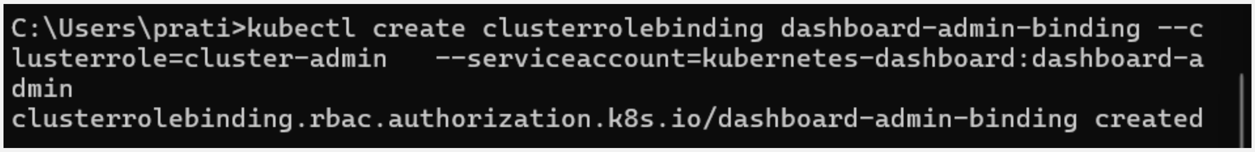
```
kubectl create serviceaccount dashboard-admin  -n kubernetes-dashboard
```



```
C:\Users\prati>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
```



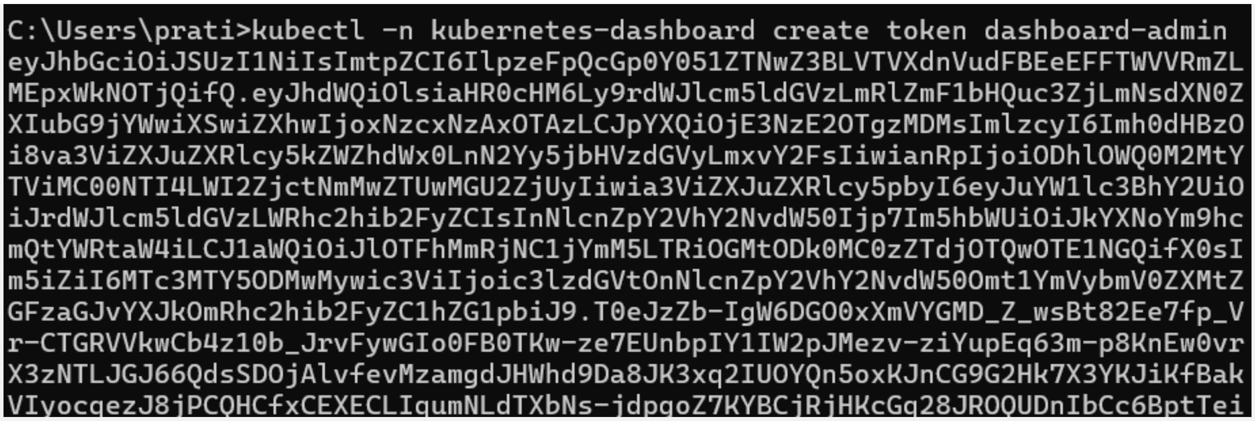
```
C:\Users\prati>kubectl create clusterrolebinding dashboard-admin-binding --c
lusterrole=cluster-admin   --serviceaccount=kubernetes-dashboard:dashboard-a
dmin
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-binding created
```

---

### Step 6: Generate Dashboard Login Token

Run the following command to get the token:

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



```
C:\Users\prati>kubectl -n kubernetes-dashboard create token dashboard-admin
eyJhbGciOiJSUzI1NiIsImtpZCI6IlpzeFpQcGp0Y051ZTNwZ3BLVTVXdnVudFBEeEFFTWVVRmZL
MEpxWkNOTjQifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0Z
XIubG9jYWwiXSwiZXhwIjoxNzcxNzAxOTAzLCJpYXQiOjE3NzE2OTgzMDMsImlzcyI6Imh0dHBzO
i8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiODhlOWQ0M2MtY
TViMC00NTI4LWI2ZjctNmMwZTUwMGU2ZjUyIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiO
iJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hc
mQtYWRtaW4iLCJ1aWQiOiJlOTFhMmRjNC1jYmM5LTRiOGMtODk0MC0zZTdjOTQwOTE1NGQifX0sI
m5iZiI6MTc3MTY5ODMwMywic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZ
GFzaGJvYXJkOmRhc2hib2FyZC1hZG1pbiJ9.T0eJzZb-IgW6DGO0xXmVYGMD_Z_wsBt82Ee7fp_V
r-CTGRVVkwCb4z10b_JrvFywGIo0FB0TKw-ze7EUnbpIY1IW2pJMezv-ziYupEq63m-p8KnEw0vr
X3zNTLJGJ66QdsSDOjAlvfevMzamgdJHWhd9Da8JK3xq2IUOYQn5oxKJnCG9G2Hk7X3YKJiKfBak
VIyocqezJ8jPCQHCfxCEXECLIqumNLdTXbNs-jdpgoZ7KYBCjRjHKcGq28JROQUDnIbCc6BptTei
```

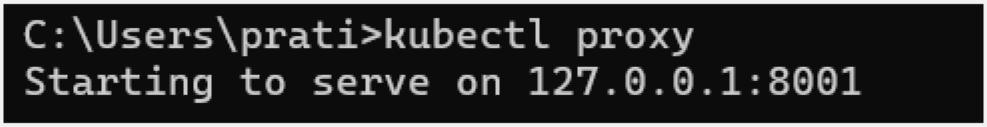
Copy the generated token (you will paste it in the browser later).

---

### Step 7: Start Kubernetes Dashboard

Run the proxy command:

```
kubectl proxy
```



```
C:\Users\prati>kubectl proxy
Starting to serve on 127.0.0.1:8001
```

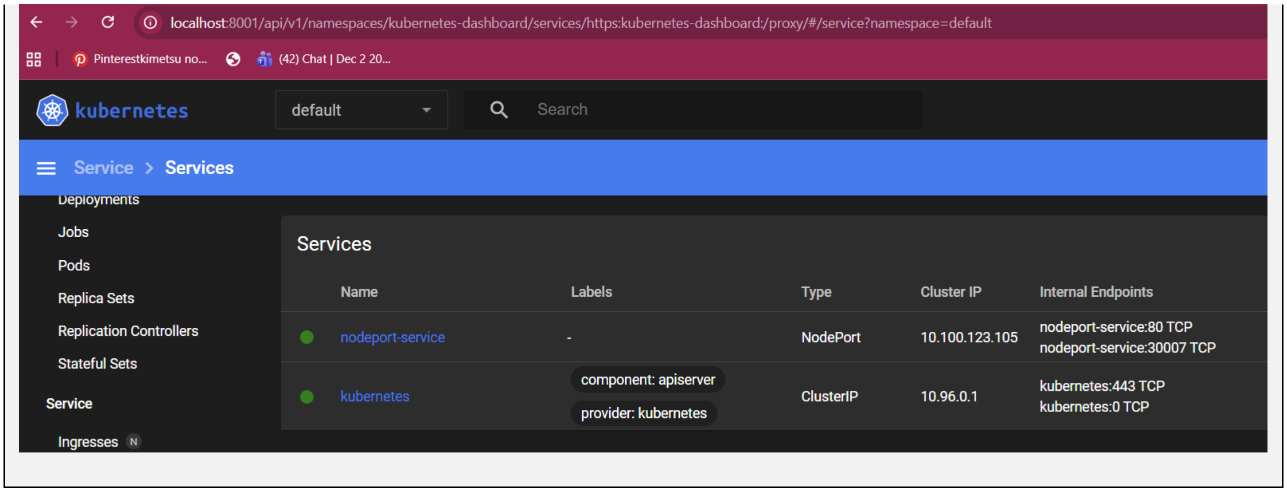
Keep this terminal **running**.

---

### Step 8: Access Kubernetes Dashboard in Browser

Open a web browser and paste the following URL:

```
http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/
```



Services

| | Name | Labels | Type | Cluster IP | Internal Endpoints |
|---|---|---|---|---|---|
| ● | nodeport-service | - | NodePort | 10.100.123.105 | nodeport-service:80 TCP<br>nodeport-service:30007 TCP |
| ● | kubernetes | component: apiserver<br>provider: kubernetes | ClusterIP | 10.96.0.1 | kubernetes:443 TCP<br>kubernetes:0 TCP |

## Step 9: Login to Dashboard

1. Select **Token** authentication
2. Paste the token generated earlier
3. Click **Sign In**

You should now see the **Kubernetes Dashboard UI**.

## Step 10: Explore Dashboard

You can now view:

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