Name : Gourav Das
Sap : 500122586
Batch : B2 Devops

Lab Exercise 12

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
```

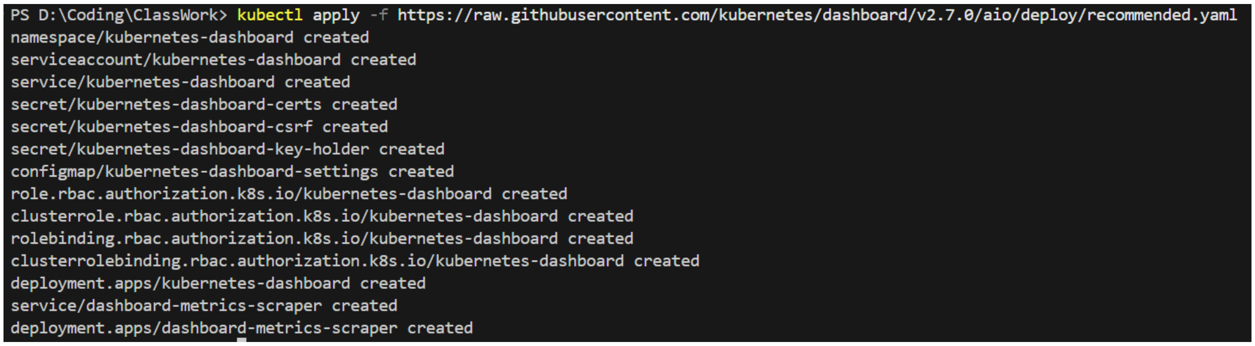
Expected output:

Node status should be **Ready**

Step 3: Deploy Kubernetes Dashboard

Apply the official Kubernetes Dashboard manifest:

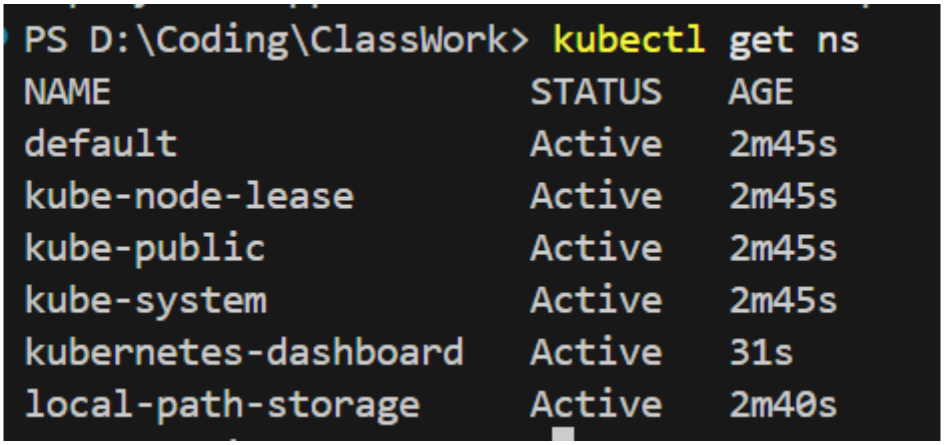
```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```



```
PS D:\Coding\ClassWork> 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
```



```
PS D:\Coding\ClassWork> kubectl get ns
NAME                   STATUS   AGE
default                Active   2m45s
kube-node-lease        Active   2m45s
kube-public            Active   2m45s
kube-system            Active   2m45s
kubernetes-dashboard   Active   31s
local-path-storage     Active   2m40s
```

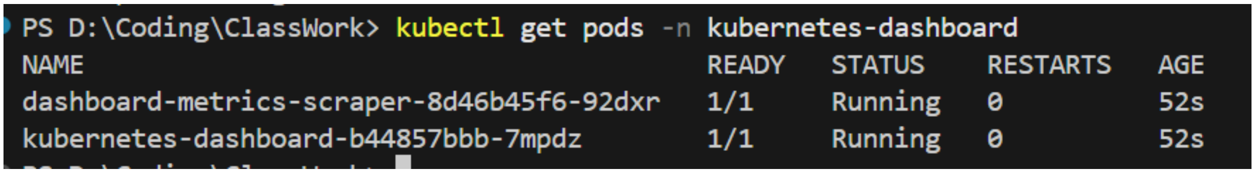
You should see:

```
kubernetes-dashboard
```

Step 4: Verify Dashboard Pods

Check dashboard pods:

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



```
PS D:\Coding\ClassWork> kubectl get pods -n kubernetes-dashboard
NAME                                         READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-8d46b45f6-92dxr   1/1     Running   0          52s
kubernetes-dashboard-b44857bbb-7mpdz        1/1     Running   0          52s
```

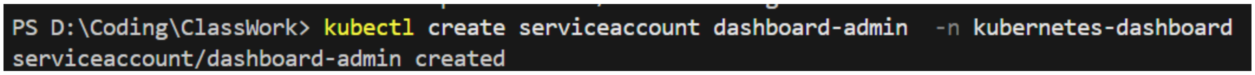
Expected status:

Running

Step 5: Create Admin User for Dashboard Access

Create a service account:

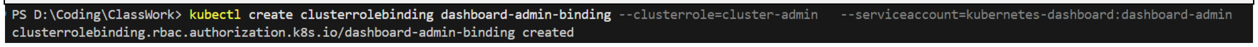
```
kubectl create serviceaccount dashboard-admin  -n kubernetes-dashboard
```



```
PS D:\Coding\ClassWork> 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
```

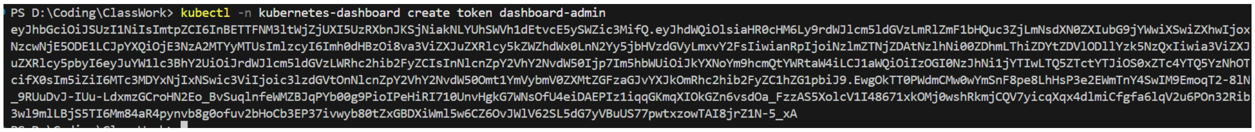


```
PS D:\Coding\ClassWork> kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin   --serviceaccount=kubernetes-dashboard:dashboard-admin
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-binding created
```

Step 6: Generate Dashboard Login Token

Run the following command to get the token:

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



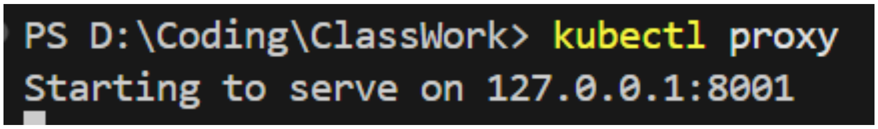
```
PS D:\Coding\ClassWork> kubectl -n kubernetes-dashboard create token dashboard-admin
eyJhbGciOiJSUzI1NiIsImtpZCI6InBETTFNM3ltWjZjUXI5UzRXbnJKSjNiakNLYUhSWVh1dEtvcE5ySWZic3MifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjox
NzcwNjE5ODE1LCJpYXQiOjE3NzA2MTYyMTUsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiNzlmZTNjZDAtNzlhNi00ZDhmLThiZDYtZDVlODllYzk5NzQxIiwia3ViZXJ
uZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hcmQtYWRtaW4iLCJ1aWQiOiIzOGI0NzJhNi1jYTIwLTQ5ZTctYTJiOS0xZTc4YTQ5YzNhOT
cifX0sIm5iZiI6MTc3MDYxNjIxNSwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOmRhc2hib2FyZC1hZG1pbiJ9.EwgOkTT0PWdmCMw0wYmSnF8pe8LhHsP3e2EWmTnY4SwIM9EmoqT2-8lN
_9RUuDvJ-IUu-LdxmzGCroHN2Eo_BvSuqlnfeWMZBJqPYb00g9PioIPeHiRI710UnvHgkG7WNsOfU4eiDAEPIz1iqqGKmqXIOkGZn6vsdOa_FzzAS5XolcV1I48671xkOMj0wshRkmjCQV7yicqXqx4dlmiCfgfa6lqV2u6POn32Rib
3wl9mlLBjS5TI6Mm84aR4pynvb8g0ofuv2bHoCb3EP37ivwyb80tZxGBDXiWml5w6CZ6OvJWlV62SL5dG7yVBuUS77pwtxzowTAI8jrZ1N-5_xA
PS D:\Coding\ClassWork>
```

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

Step 7: Start Kubernetes Dashboard

Run the proxy command:

```
kubectl proxy
```



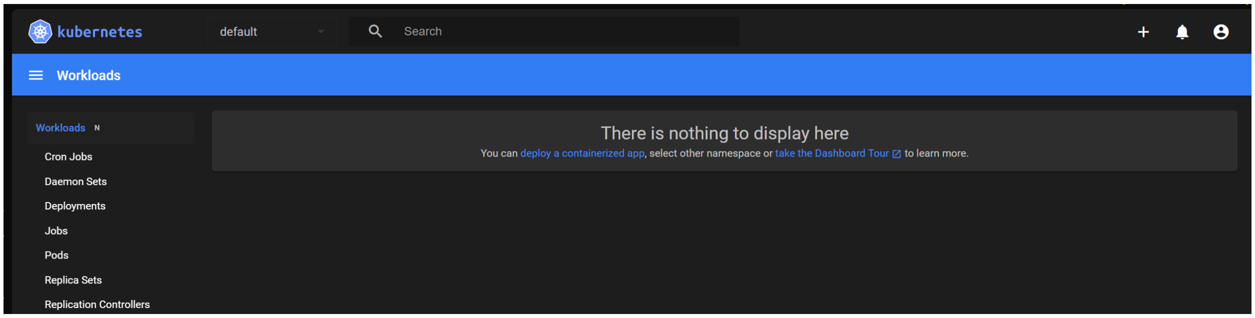
```
PS D:\Coding\ClassWork> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

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/
```



There is nothing to display here

You can deploy a containerized app, select other namespace or take the Dashboard Tour to learn more.

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