Lab Exercise 12 - Start and Access Kubernetes Dashboard

Name-Misha

SAP ID- 500119679

Batch-2

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**

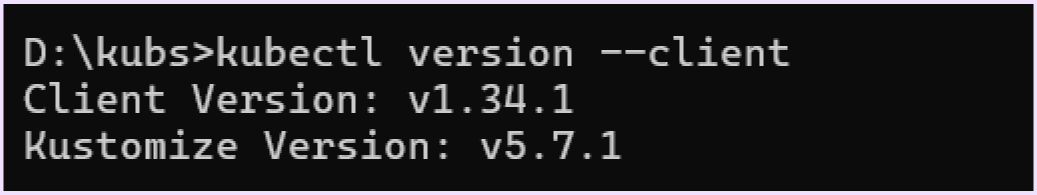
3. Select **Kubernetes**
4. Check **Enable Kubernetes**
5. Click **Apply & Restart**

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

Step 2: Verify Kubernetes Cluster

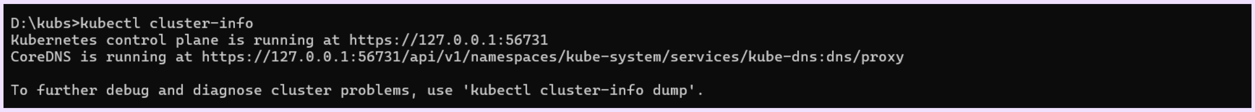
Open **PowerShell** or **Command Prompt** and run:

- kubectl version –client



```
D:\kubs>kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1
```

- Check cluster status:
- kubectl cluster-info

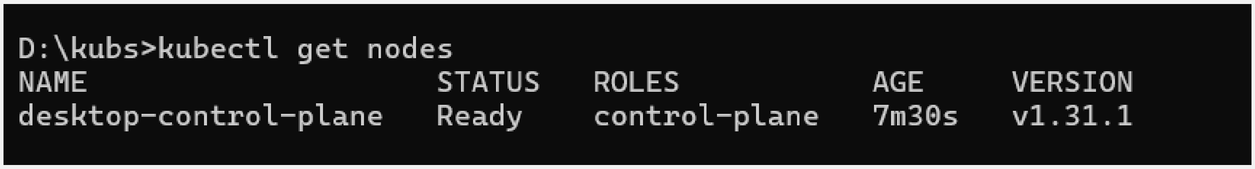


```
D:\kubs>kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:56731
CoreDNS is running at https://127.0.0.1:56731/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Check nodes:

kubectl get nodes



```
D:\kubs>kubectl get nodes
NAME                     STATUS   ROLES           AGE     VERSION
desktop-control-plane    Ready    control-plane   7m30s   v1.31.1
```

Expected output:

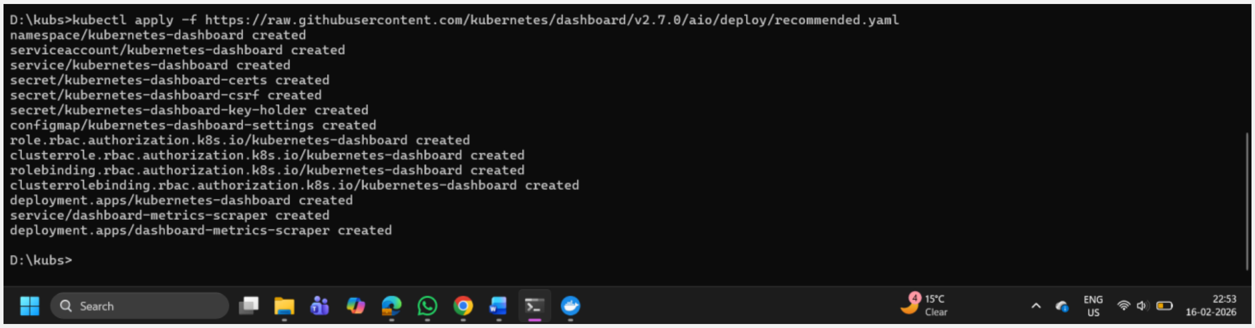
Node status should be **Ready**

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
```

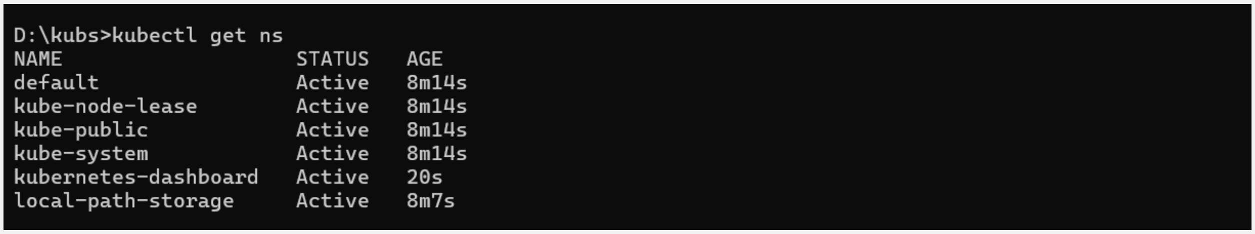


```
D:\kubs>kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf 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

D:\kubs>
```

Verify namespace creation:

```
kubectl get ns
```



```
D:\kubs>kubectl get ns
NAME                   STATUS   AGE
default                Active   8m14s
kube-node-lease        Active   8m14s
kube-public            Active   8m14s
kube-system            Active   8m14s
kubernetes-dashboard   Active   20s
local-path-storage     Active   8m7s
```

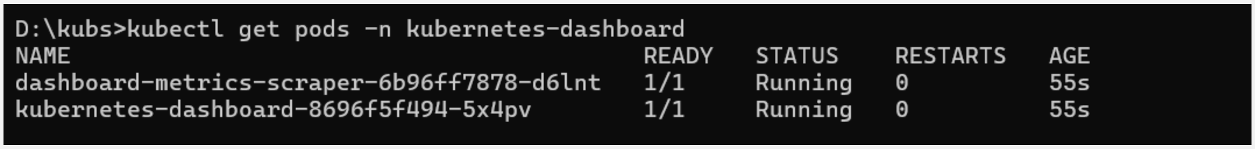
You should see:

```
kubernetes-dashboard
```

Step 4: Verify Dashboard Pods

Check dashboard pods:

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



```
D:\kubs>kubectl get pods -n kubernetes-dashboard
NAME                                        READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-6b96ff7878-d6lnt   1/1     Running   0          55s
kubernetes-dashboard-8696f5f494-5x4pv        1/1     Running   0          55s
```

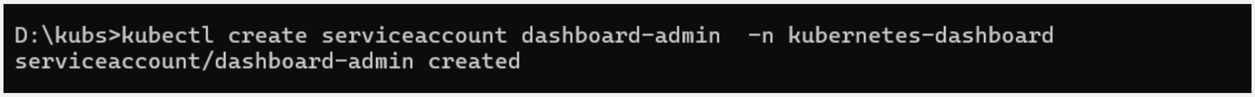
Expected status:

Running

Step 5: Create Admin User for Dashboard Access

Create a service account:

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

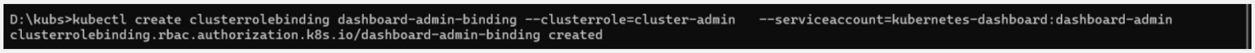


```
D:\kubs>kubectl create serviceaccount dashboard-admin  -n kubernetes-dashboard
serviceaccount/dashboard-admin created
```

Create cluster role binding:

```
kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin  -

-serviceaccount=kubernetes-dashboard:dashboard-admin
```

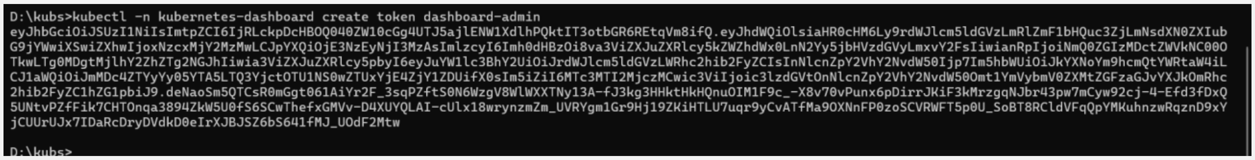


```
D:\kubs>kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin   --serviceaccount=kubernetes-dashboard:dashboard-admin
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-binding created
```

Step 6: Generate Dashboard Login Token

Run the following command to get the token:

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



```
D:\kubs>kubectl -n kubernetes-dashboard create token dashboard-admin
eyJhbGciOiJSUzI1NiIsImtpZCI6IjRLckpDcHBOQ040ZW10cGg4UTJ5ajlENW1XdlhPQktIT3otbGR6REtqVm8ifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIub
G9jYWwiXSwiZXhwIjoxNzcxMjY2MzMwLCJpYXQiOjE3NzEyNjI3MzAsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiNmQ0ZGIzMDctZWVkNC00O
TkwLTg0MDgtMjlhY2ZhZTg2NGJhIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hcmQtYWRtaW4iL
CJ1aWQiOiJmMDc4ZTYyYy05YTA5LTQ3YjctOTU1NS0wZTUxYjE4ZjY1ZDUifX0sIm5iZiI6MTc3MTI2MjczMCwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOmRhc
2hib2FyZC1hZG1pbiJ9.deNaoSm5QTCsR0mGgt061AiYr2F_3sqPZftS0N6WzgV8WlWXXTNy13A-fJ3kg3HHktHkHQnuOIM1F9c_-X8v70vPunx6pDirrJKiF3kMrzgqNJbr43pw7mCyw92cj-4-Efd3fDxQ
5UNtvPZfFik7CHTOnqa3894ZkW5U0fS6SCwThefxGMVv-D4XUYQLAI-cUlx18wrynzmZm_UVRYgm1Gr9Hj19ZKiHTLU7uqr9yCvATfMa9OXNnFP0zoSCVRWFT5p0U_SoBT8RCldVFqQpYMKuhnzwRqznD9xY
jCUUrUJx7IDaRcDryDVdkD0eIrXJBJSZ6bS641fMJ_UOdF2Mtw

D:\kubs>
```

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

Step 7: Start Kubernetes Dashboard

Run the proxy command:

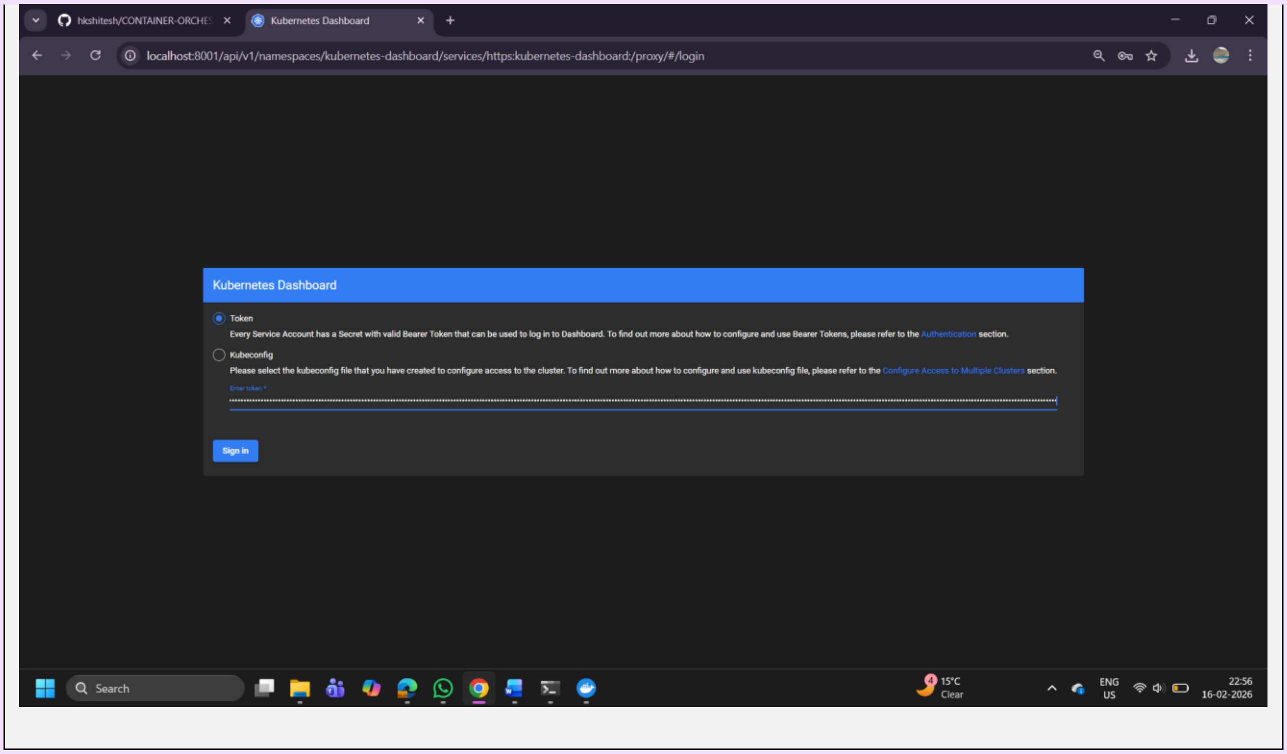
```
kubectl proxy
```

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/



localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login

Kubernetes Dashboard

● Token

Every Service Account has a Secret with valid Bearer Token that can be used to log in to Dashboard. To find out more about how to configure and use Bearer Tokens, please refer to the Authentication section.

○ Kubeconfig

Please select the kubeconfig file that you have created to configure access to the cluster. To find out more about how to configure and use kubeconfig file, please refer to the Configure Access to Multiple Clusters section.

Enter token *

Sign in

Step 9: Login to Dashboard

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

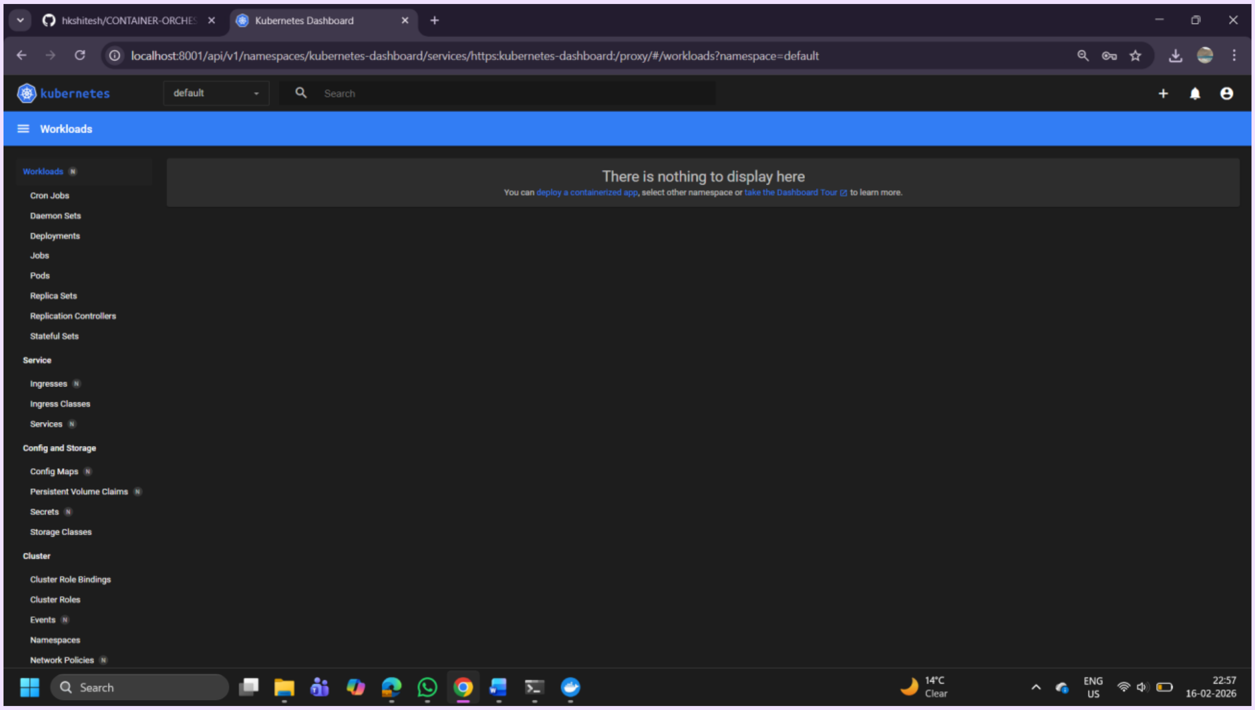
You should now see the **Kubernetes Dashboard UI**.

Step 10: Explore Dashboard

You can now view:

- Nodes
- Pods
- Deployments
- Services

- Namespaces
- ConfigMaps and Secrets



localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/workloads?namespace=default

There is nothing to display here

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