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.

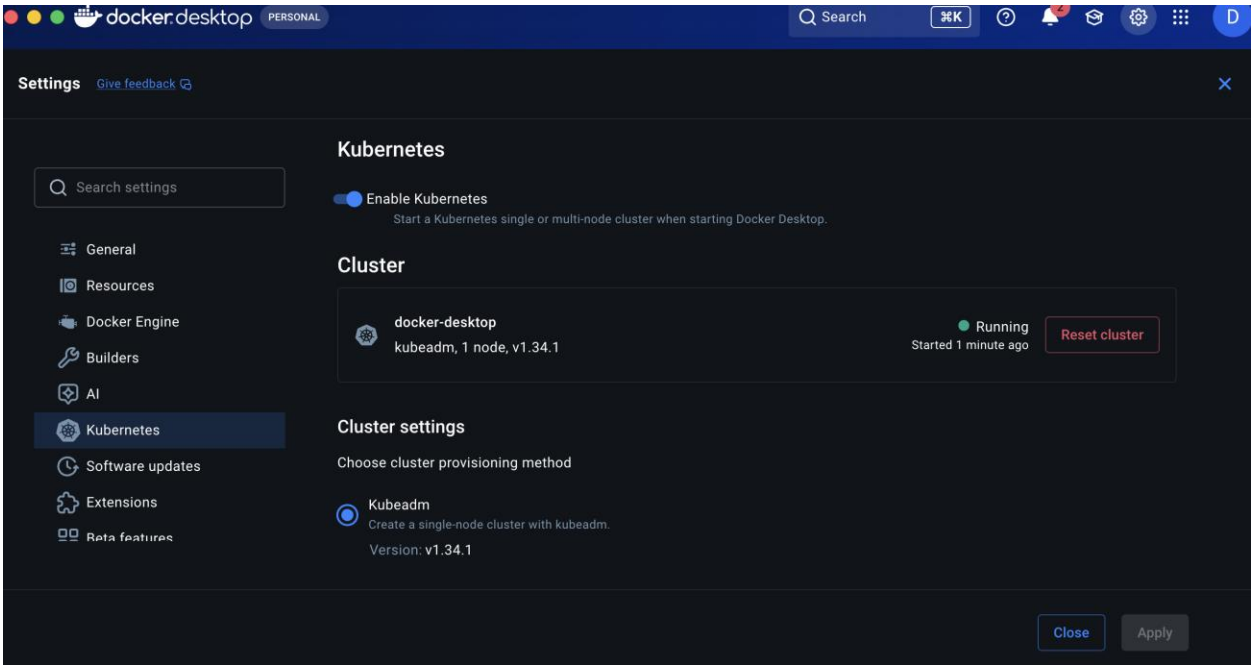
Prerequisites

- 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

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl version --client
Client Version: v1.35.0
Kustomize Version: v5.7.1
```

Check cluster status:

- kubectl cluster-info

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:6443
CoreDNS is running at https://127.0.0.1:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

Check nodes:

```
kubectl get nodes
```

Expected output:

Node status should be **Ready**

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl get nodes
NAME             STATUS   ROLES           AGE     VERSION
docker-desktop   Ready    control-plane   5m29s   v1.34.1
```

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
```

```
devanksilswal@devanks-MacBook-Air ex_s % 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
```

You should see:

```
kubernetes-dashboard
```

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl get ns
NAME                   STATUS   AGE
default                Active   6m7s
kube-node-lease        Active   6m7s
kube-public            Active   6m7s
kube-system            Active   6m7s
kubernetes-dashboard   Active   7s
```

Step 4: Verify Dashboard Pods

Check dashboard pods:

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

Expected status:

Running

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl get pods -n kubernetes-dashboard
NAME                                         READY   STATUS              RESTARTS   AGE
dashboard-metrics-scraper-5ffb7d645f-kg4qh   0/1     ContainerCreating   0          19s
kubernetes-dashboard-6c7b75ffc-vktjt         1/1     Running             0          19s
```

Step 5: Create Admin User for Dashboard Access

Create a service account:

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

```
devanksilswal@devanks-MacBook-Air ex_s % 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
```

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluste
r-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
```

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl -n kubernetes-dashboard create token dashboard-admin
eyJhbGciOiJSUzI1NiIsImtpZCI6InBWeVRrUmZHOWh5UzM2TE05UkpjYVJVUG5kaE55ZF9FeDd6enV5SjcwSVkifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJl
cm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNzcxOTQ1NjcwLCJpYXQiOjE3NzE5NDIwNzAsImlzcyI6Imh0dHBzOi8va3ViZXJ
uZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiNzUzYzQyYmItMjVmOS00ODZmLTk2NmItZmM5MGYxYjA0ZDEyIiwia3ViZXJuZXRlcy
5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hcmQtYWRtaW4iLCJ1aWQiO
iIxMmExNmIyMS01ZGVjLTQ4Y2EtODJlNC01ZDEzMGZjNTY2Y2MifX0sIm5iZiI6MTc3MTk0MjA3MCwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1
YmVybmV0ZXMtZGFzaGJvYXJkOmRhc2hib2FyZC1hZG1pbiJ9.VVcefmMfRna7sNZnZzK0cbhCpFLLTgngebhPV0AbiQGpUxb3Tdyxk8cUftbVFbibSH_IkB
s951NvJ356GclVSHf4Gnetf-U_sOWTILWf8kIZWyxJm5vEmNxjcIYzZjKVOFWCPLGIDeLHpXkgH-2ygu09xQwLyGgn0gylcIOlz7V7Qfle0sJ4ucs-pz99W
o14wrStZihfa3-NYHLvmXMx1c8_h9Ddz2qJCRytoYgEucpPIhRz_XYsNQK9-2juyHMq9oqUtt4dr4mnAvhIK3COSc1C6AHkIMebnOInDPIkHtMbIXFOT0XS
bn-FoxwfiOWegbZmPzpRuMAsz_DqWZ2AnQ
```

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**.

```
devanksilswal@devanks-MacBook-Air ex_s % kubectl proxy
Starting to serve on 127.0.0.1:8001
```

Step 8: Access Kubernetes Dashboard in Browser

Open a web browser and paste the following URL:

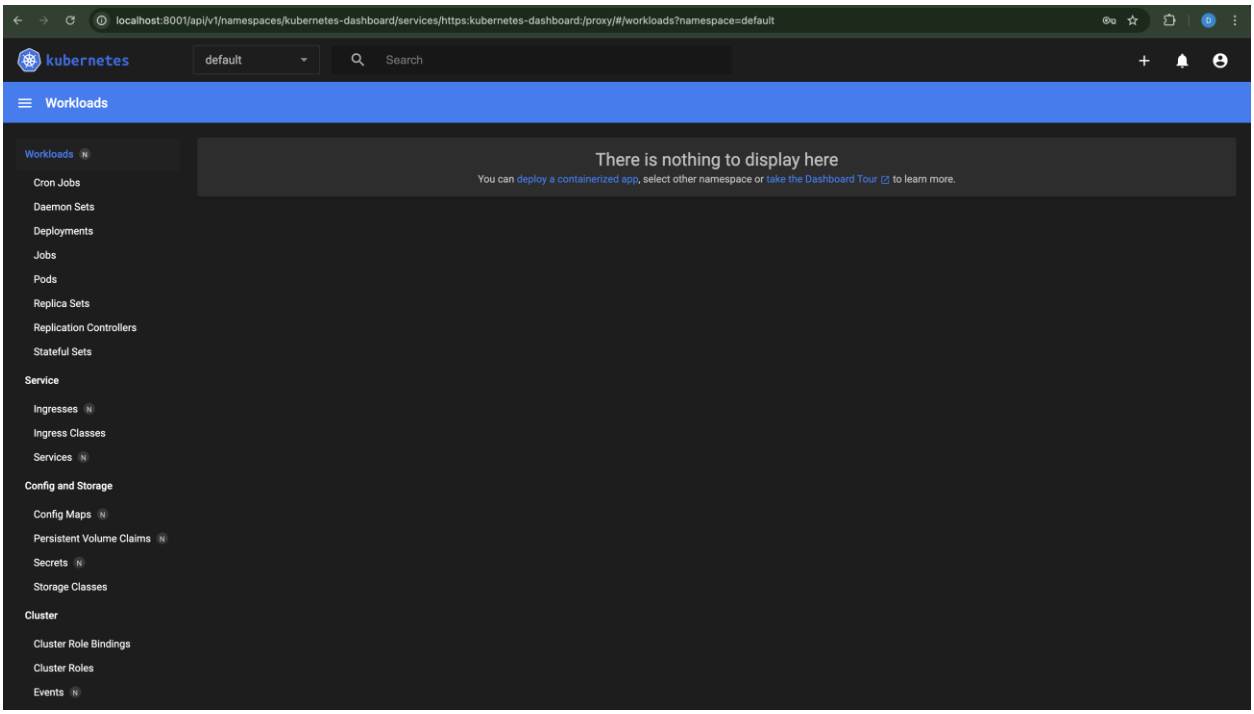
```
http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/
```

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