

# Lab Exercise 7 :- Start and Access

## Kubernetes Dashboard

**Name:- Vansh Bhatt**

**Sap ID:- 500125395**

**Batch:- DevOps B1**

**To:- Hitesh Sharma Sir**

---

### Objective

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

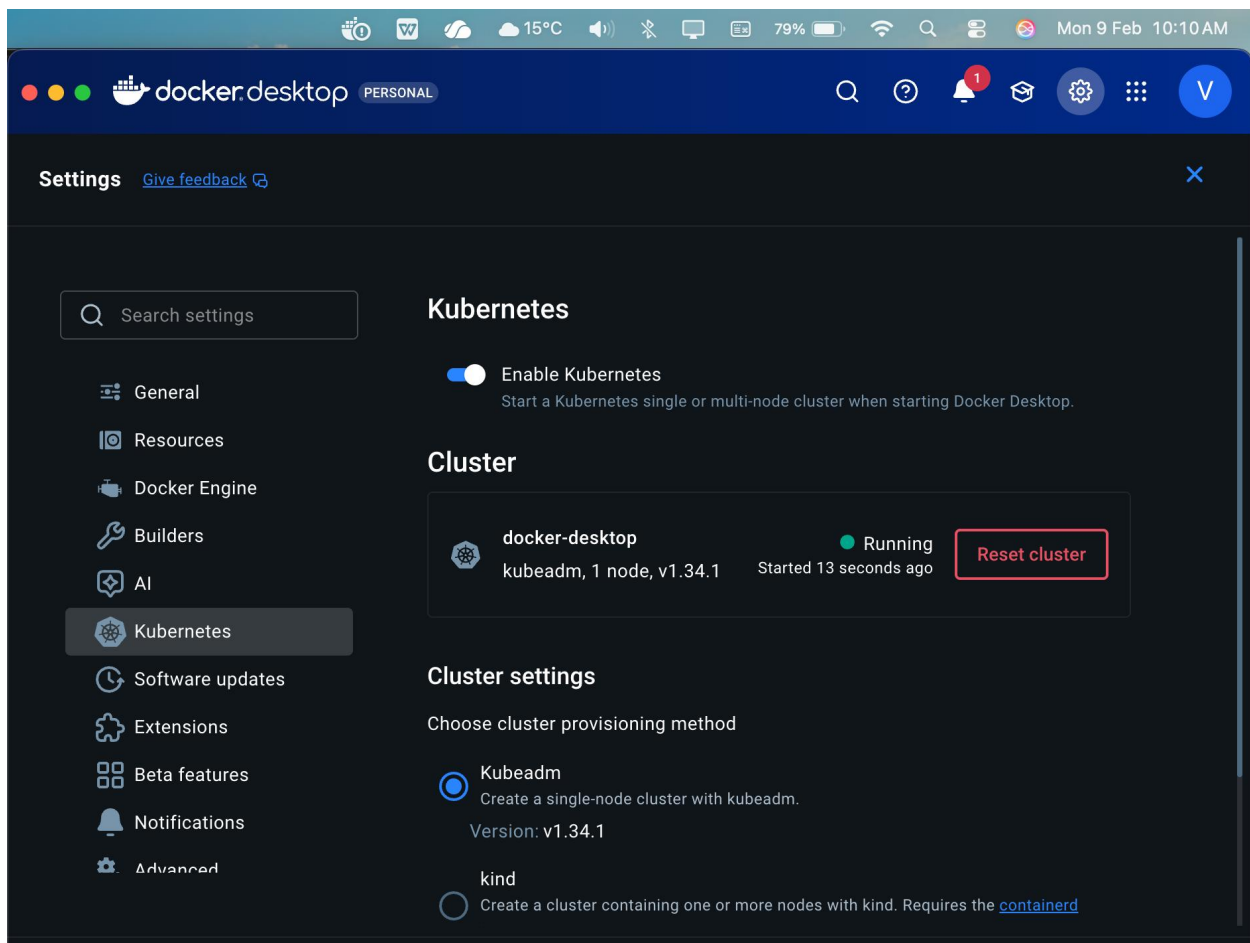
---

### 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**



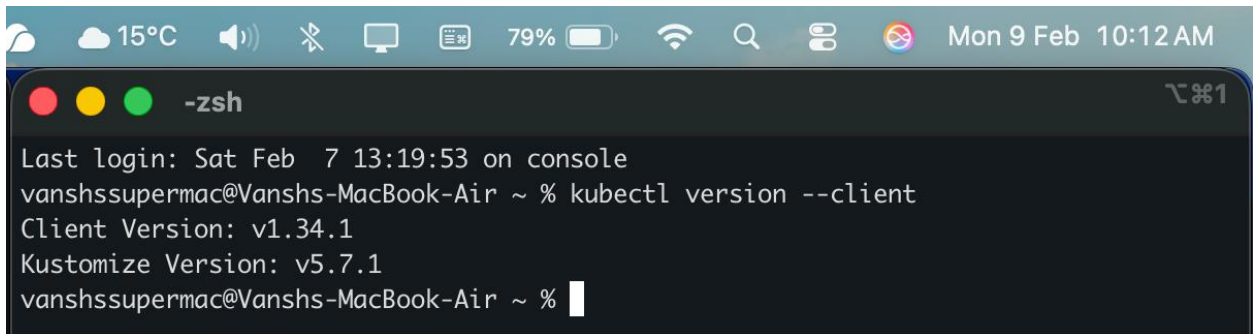
Wait until Kubernetes status shows **Running** (green).

---

## Step 2: Verify Kubernetes Cluster

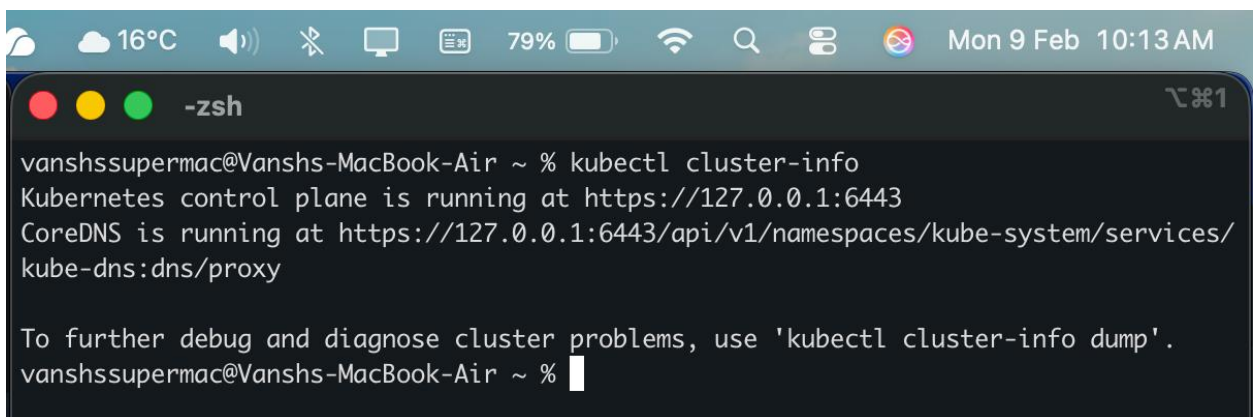
Open **PowerShell** or **Command Prompt** and run:

- `kubectl version --client`



```
Mon 9 Feb 10:12 AM
-vsh
Last login: Sat Feb  7 13:19:53 on console
vanshssupermac@Vanshs-MacBook-Air ~ % kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1
vanshssupermac@Vanshs-MacBook-Air ~ %
```

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

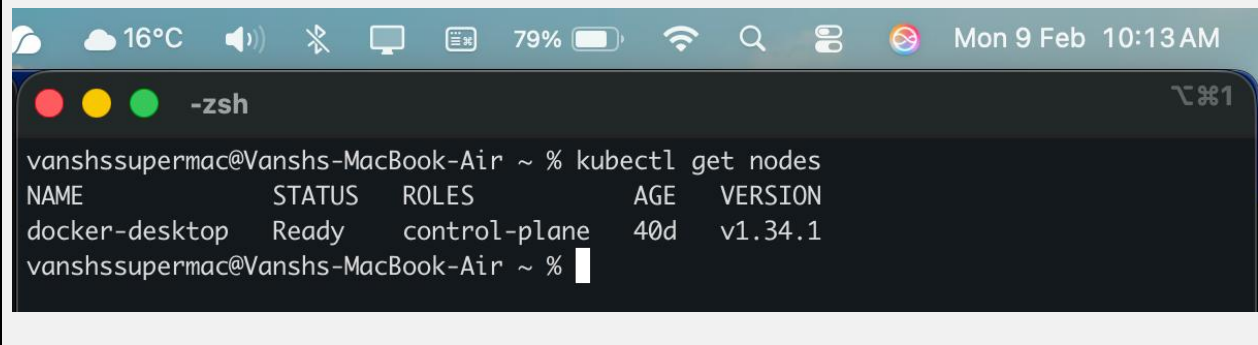


```
Mon 9 Feb 10:13 AM
-vsh
vanshssupermac@Vanshs-MacBook-Air ~ % kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:6443
CoreDNS is running at https://127.0.0.1:6443/api/v1/namespaces/kube-system/services/
kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
vanshssupermac@Vanshs-MacBook-Air ~ %
```

Check nodes:

`kubectl get nodes`



```
Mon 9 Feb 10:13 AM
-vsh
vanshssupermac@Vanshs-MacBook-Air ~ % kubectl get nodes
NAME             STATUS    ROLES    AGE   VERSION
docker-desktop   Ready    control-plane   40d   v1.34.1
vanshssupermac@Vanshs-MacBook-Air ~ %
```

Expected output:

Node status should be **Ready**

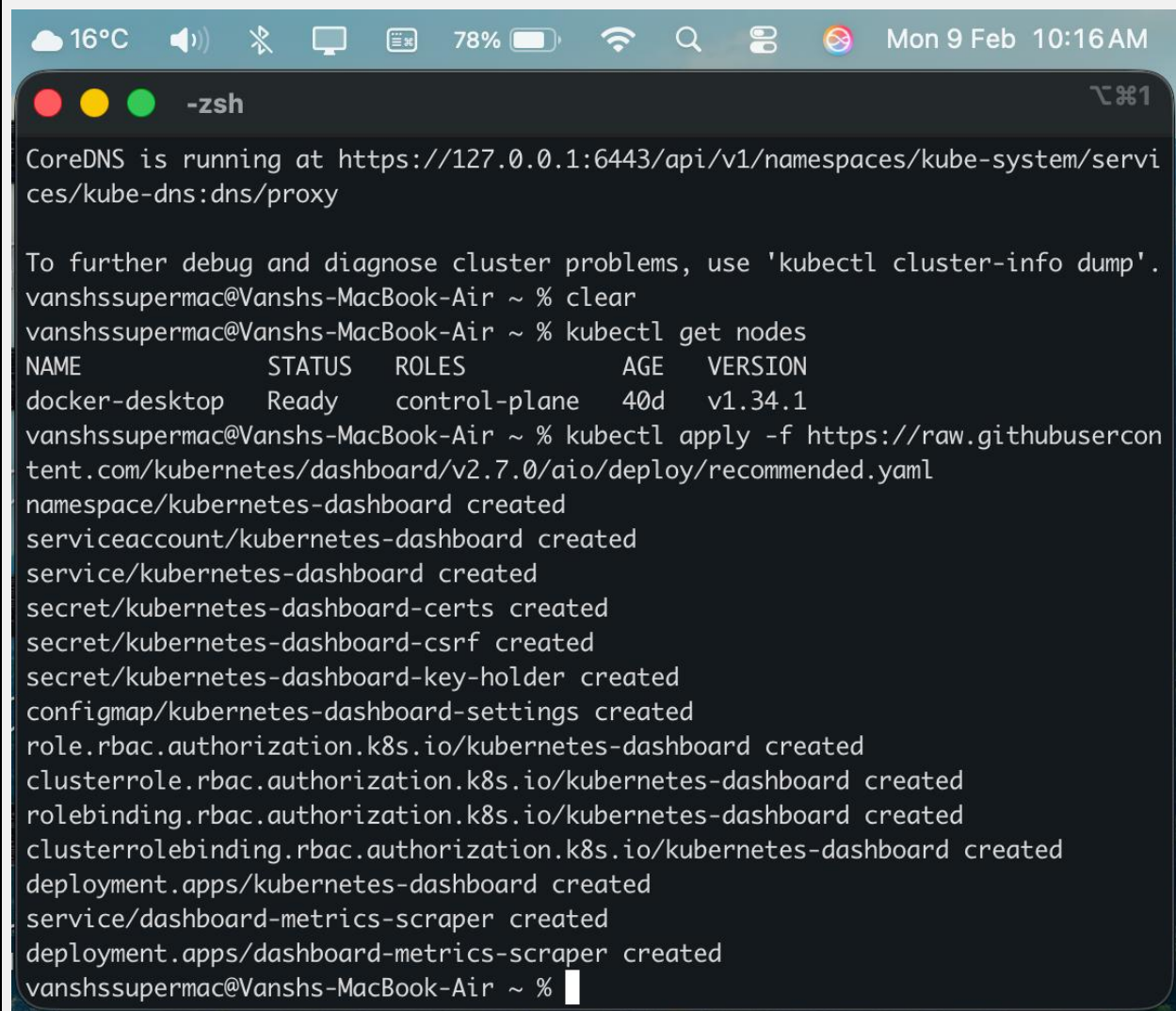
---

### 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>



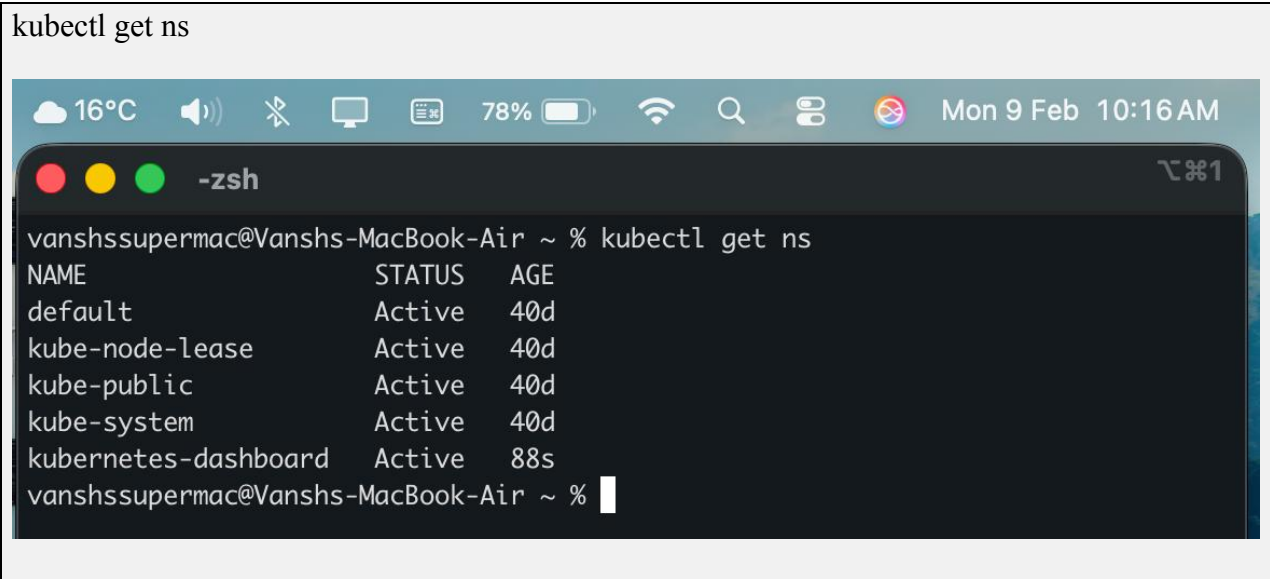
The screenshot shows a macOS terminal window with a dark background. The title bar at the top indicates the temperature is 16°C, battery is at 78%, and the time is Mon 9 Feb 10:16 AM. The terminal window is titled '-zsh'. The output of the command shows CoreDNS is running at a specific URL. It then lists several debugging instructions. The user enters 'clear' and 'kubectl get nodes', which returns a table with one node: 'docker-desktop' in 'Ready' status, with 'control-plane' role, '40d' age, and 'v1.34.1' version. Finally, the user runs 'kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml', which successfully creates a namespace, serviceaccount, service, secrets, configmap, roles, rolebindings, deployment, and scraper for the Kubernetes dashboard.

```
CoreDNS is running at https://127.0.0.1:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
vanshssupermac@Vanshs-MacBook-Air ~ % clear
vanshssupermac@Vanshs-MacBook-Air ~ % kubectl get nodes
NAME                STATUS    ROLES    AGE   VERSION
docker-desktop      Ready     control-plane  40d   v1.34.1
vanshssupermac@Vanshs-MacBook-Air ~ % 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
vanshssupermac@Vanshs-MacBook-Air ~ %
```

Verify namespace creation:

```
kubectrl get ns
```

A terminal window on a Mac showing the command 'kubectrl get ns' and its output. The output lists several namespaces: default, kube-node-lease, kube-public, kube-system, and kubernetes-dashboard, all with a status of 'Active'. The terminal window has a title bar with standard Mac window controls and a status bar at the top showing system information like temperature, battery, and time.

NAME	STATUS	AGE
default	Active	40d
kube-node-lease	Active	40d
kube-public	Active	40d
kube-system	Active	40d
kubernetes-dashboard	Active	88s

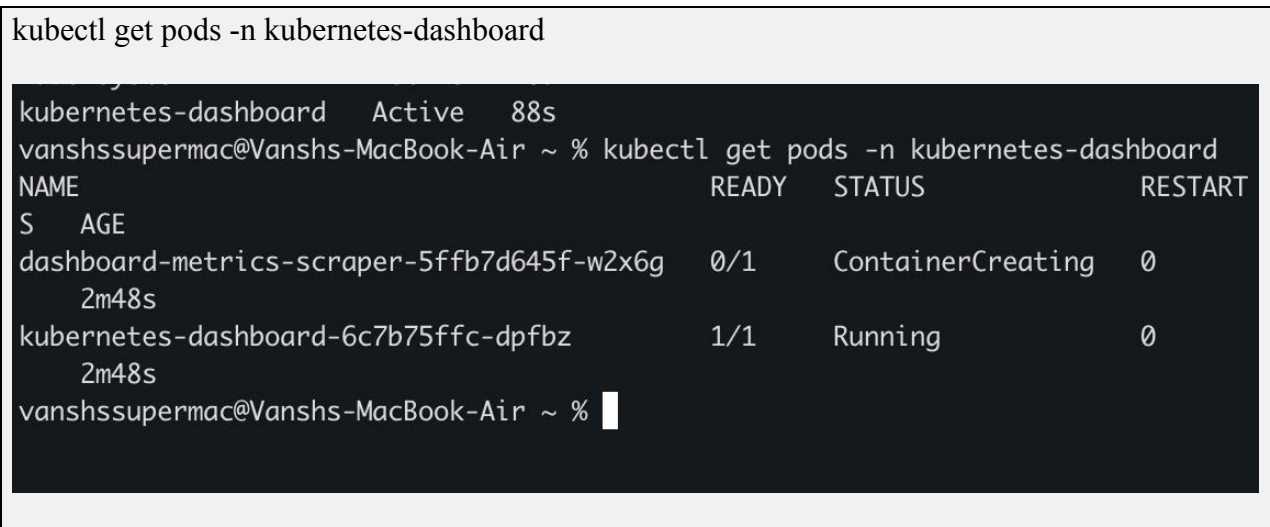
You should see:

```
kubernetes-dashboard
```

#### Step 4: Verify Dashboard Pods

Check dashboard pods:

```
kubectrl get pods -n kubernetes-dashboard
```

A terminal window on a Mac showing the command 'kubectrl get pods -n kubernetes-dashboard' and its output. The output shows two pods: 'dashboard-metrics-scraper' in 'ContainerCreating' state and 'kubernetes-dashboard' in 'Running' state. The terminal window has a title bar with standard Mac window controls and a status bar at the top showing system information like temperature, battery, and time.

NAME	READY	STATUS	RESTART
dashboard-metrics-scraper-5ffb7d645f-w2x6g	0/1	ContainerCreating	0
kubernetes-dashboard-6c7b75ffc-dpfbz	1/1	Running	0

Expected status:

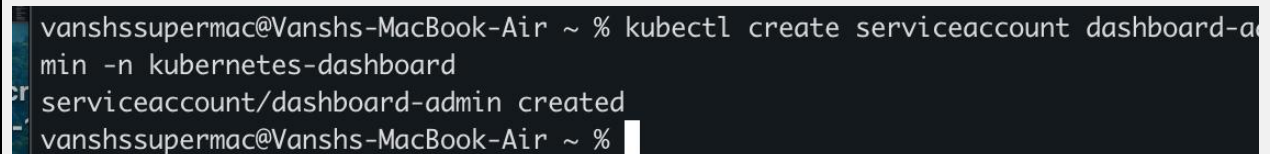
Running

---

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

Create a service account:

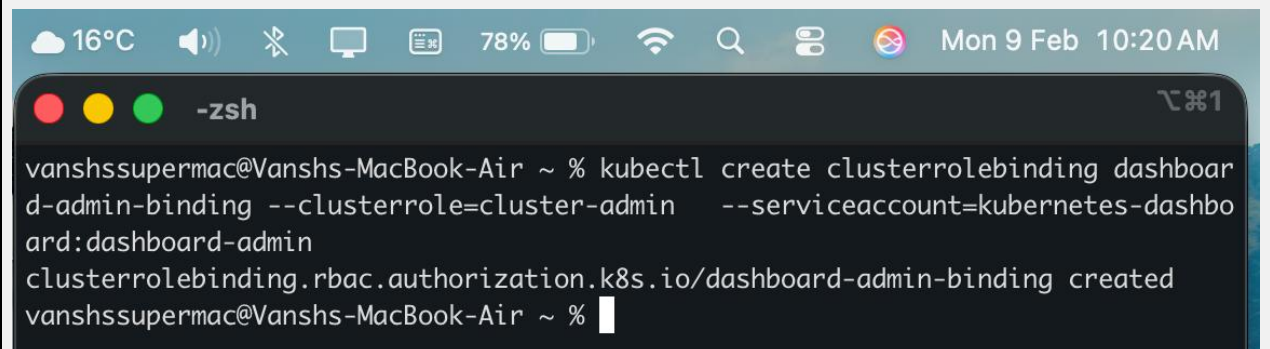
```
kubectl create serviceaccount dashboard-admin -n kubernetes-dashboard
```



A terminal window screenshot showing the command `kubectl create serviceaccount dashboard-admin -n kubernetes-dashboard` being executed. The output is `serviceaccount/dashboard-admin created`. The prompt is `vanshssupermac@Vanshs-MacBook-Air ~ %`.

Create cluster role binding:

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



A terminal window screenshot showing the command `kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin --serviceaccount=kubernetes-dashboard:dashboard-admin` being executed. The output is `clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-binding created`. The prompt is `vanshssupermac@Vanshs-MacBook-Air ~ %`. The terminal window has a title bar with standard macOS window controls and a status bar at the top showing system icons and the date/time (Mon 9 Feb 10:20 AM).

---

### Step 6: Generate Dashboard Login Token

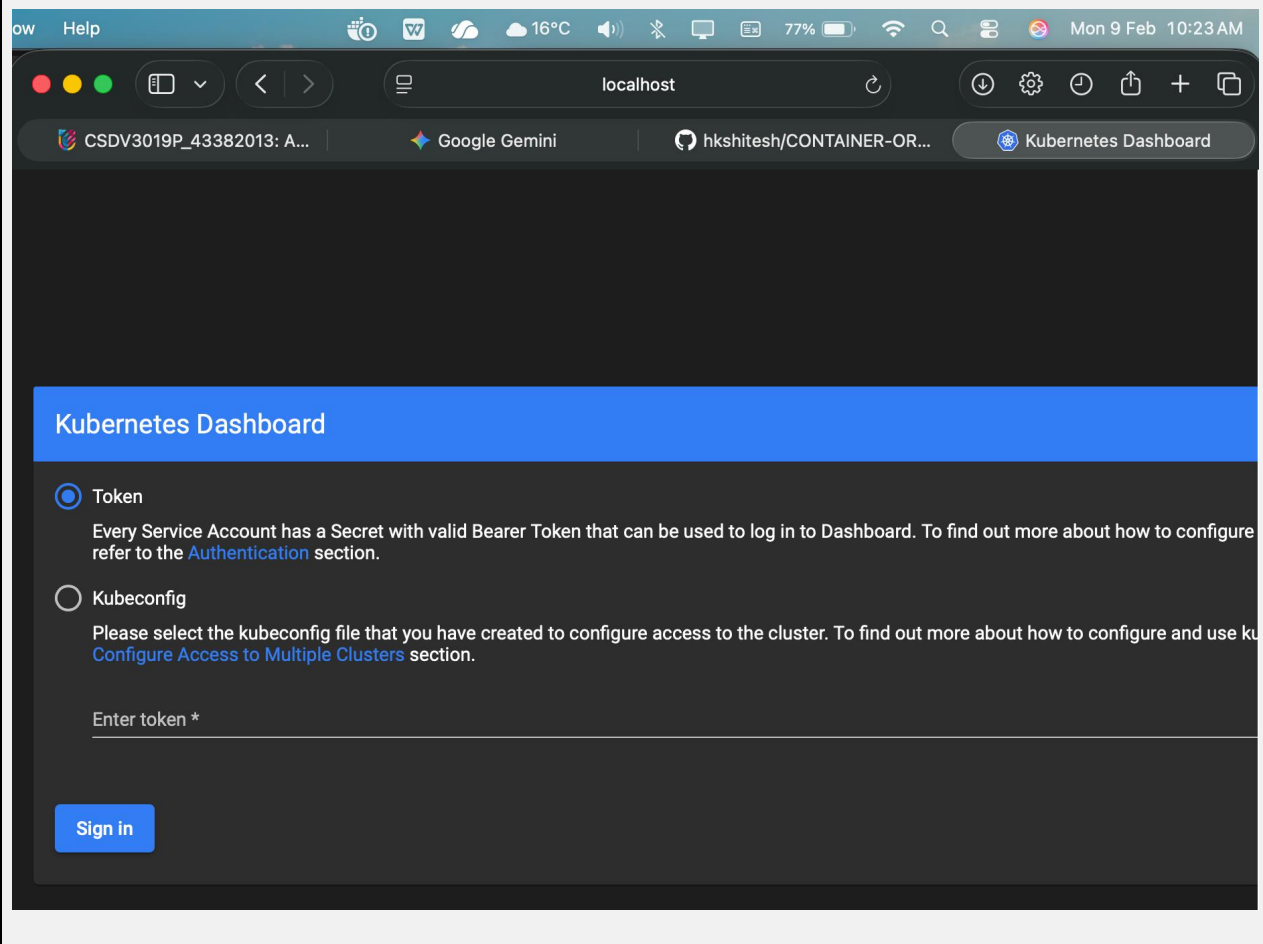
Run the following command to get the token:

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





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

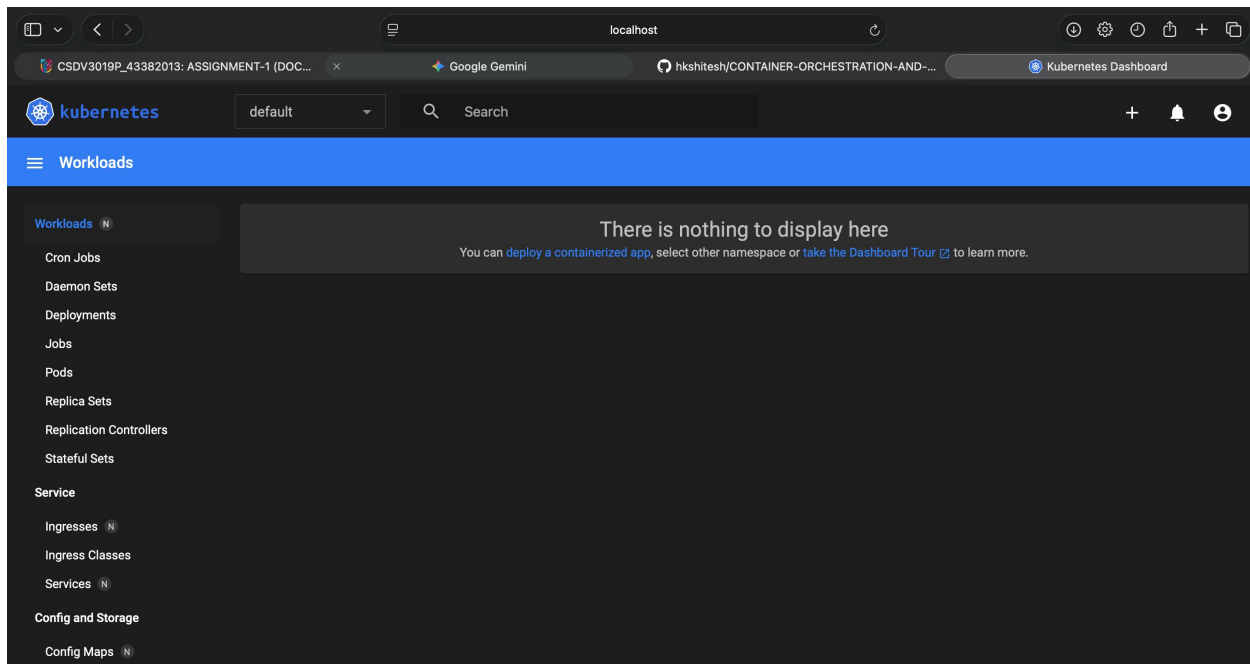


## Step 9: Login to Dashboard

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

# Thank You