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

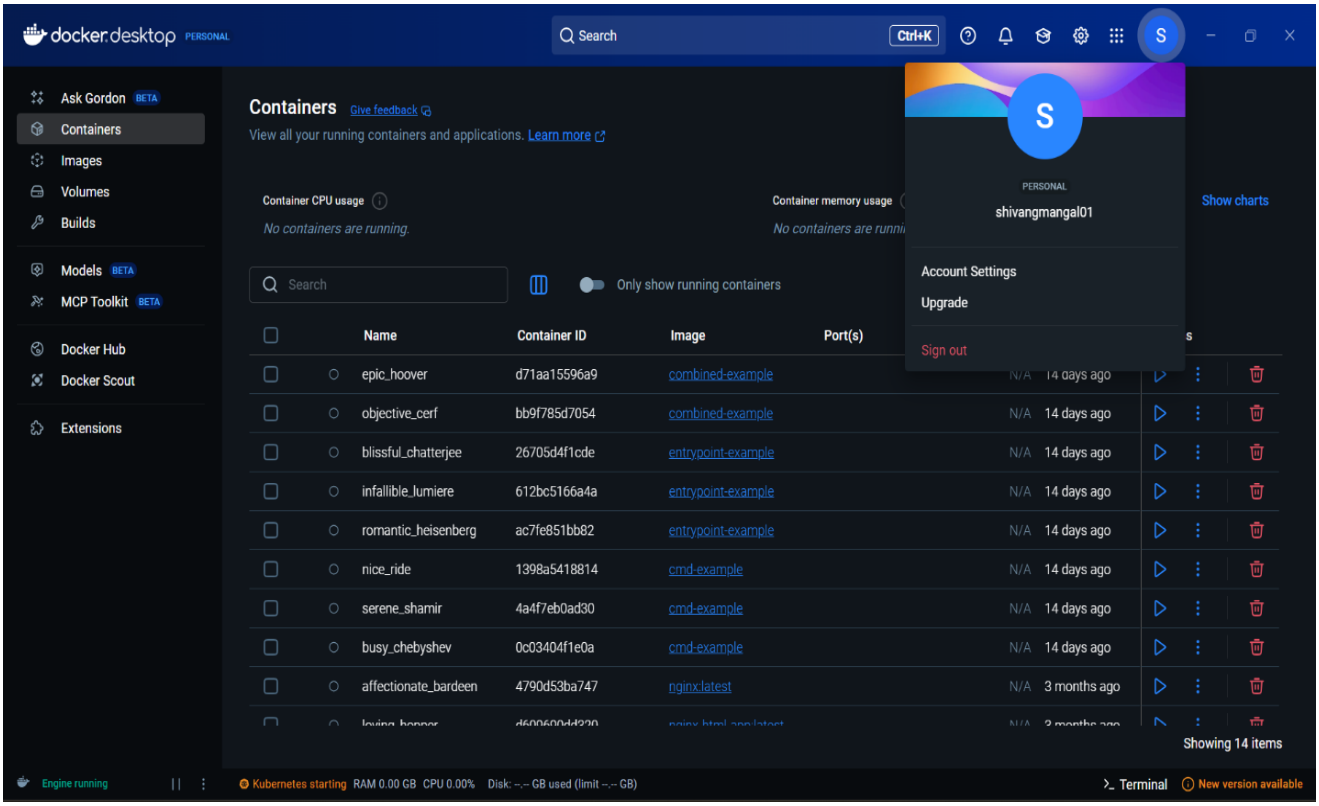
- Windows 10 / 11
- Docker Desktop installed
- Docker Desktop Kubernetes enabled
- Internet connection
- kubectl (comes bundled with Docker Desktop)

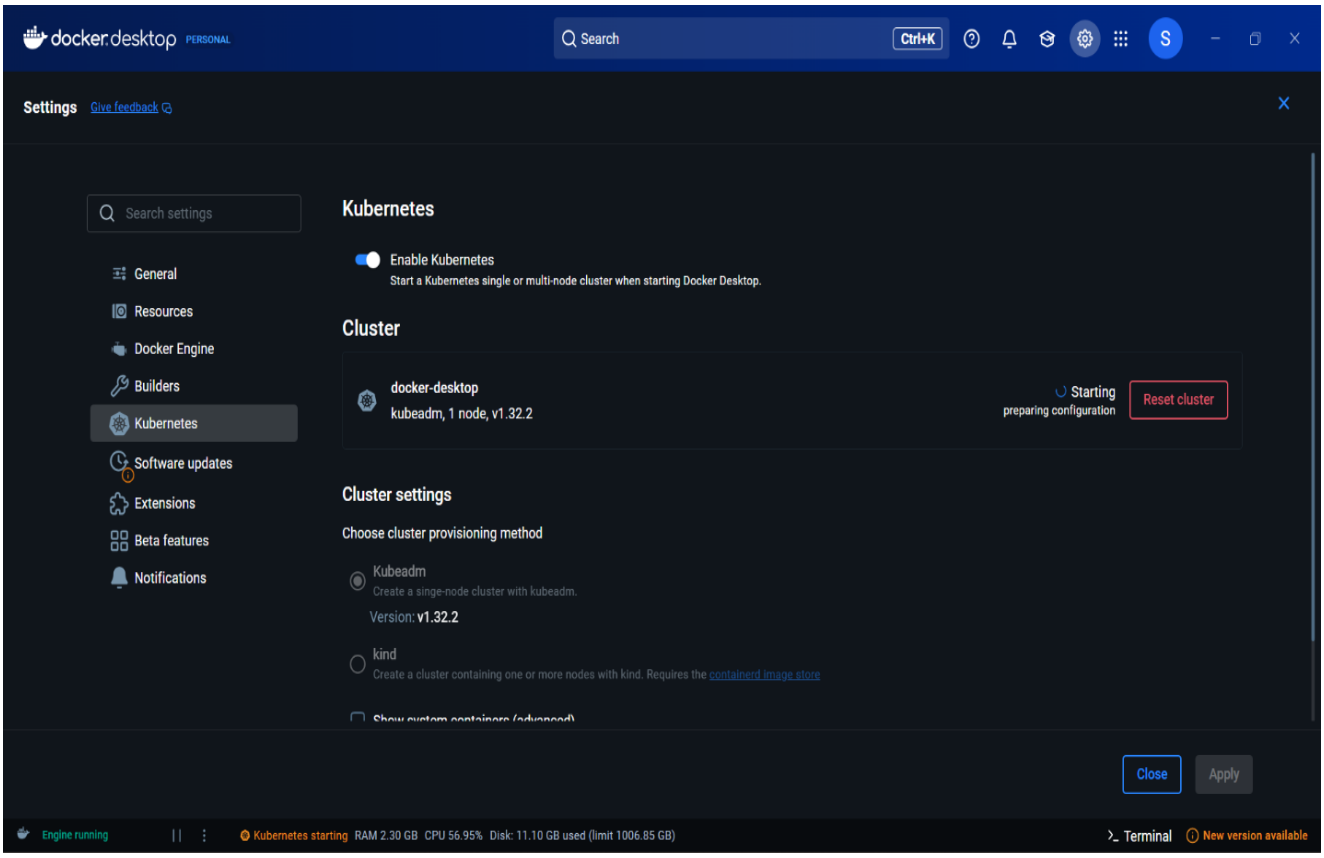
Step 1: Enable Kubernetes in Docker Desktop

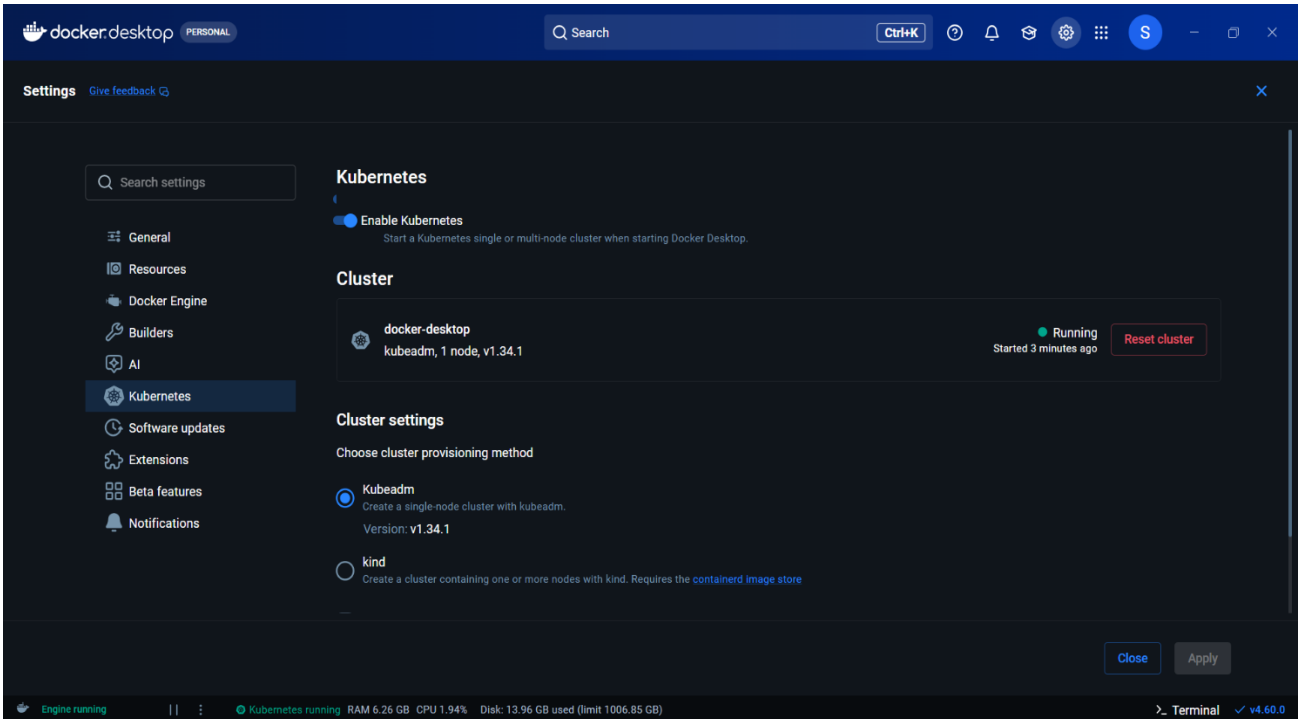
1. Open **Docker Desktop**
2. Go to **Settings**
3. Select **Kubernetes**
4. Check **Enable Kubernetes**
5. Click **Apply & Restart**

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

Shivang Mangal 500124202 Batch – 2 DevOps







Step 2: Verify Kubernetes Cluster

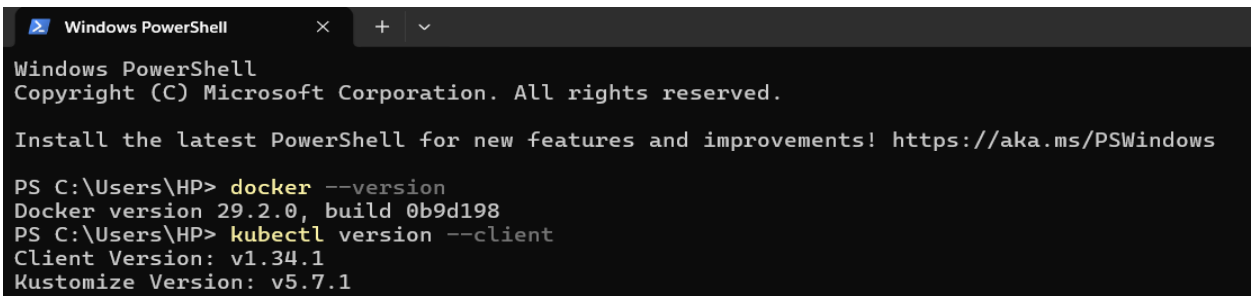
Open **PowerShell** or **Command Prompt** and run:

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

Check nodes:

```
kubectl get nodes
```

Expected output: Node status should be **Ready**



```
PS C:\Users\HP> kubectl cluster-info
Kubernetes control plane is running at https://kubernetes.docker.internal:6443
CoreDNS is running at https://kubernetes.docker.internal:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
PS C:\Users\HP> kubectl get nodes
NAME             STATUS   ROLES           AGE   VERSION
docker-desktop   Ready    control-plane   14m   v1.32.2
PS C:\Users\HP>
```

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
```

Verify namespace creation:

```
kubectl get ns
```

You should see:

```
kubernetes-dashboard
```

```
PS C:\Users\HP> 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
PS C:\Users\HP> kubectl get ns
NAME                   STATUS   AGE
default                Active   15m
kube-node-lease        Active   15m
kube-public            Active   15m
kube-system            Active   15m
kubernetes-dashboard   Active   9s
```

Step 4: Verify Dashboard Pods

Check dashboard pods:

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

Expected status: Running

```
PS C:\Users\HP> kubectl get pods -n kubernetes-dashboard
NAME                                         READY   STATUS    RESTARTS   AGE
dashboard-metrics-scraper-5bd45c9dd6-gn6cg   1/1     Running   0          92s
kubernetes-dashboard-79cbcf9fb6-pv62z        1/1     Running   0          92s
PS C:\Users\HP>
```

Step 5: Create Admin User for Dashboard Access

Create a service account:

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

Create cluster role binding:

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

```
PS C:\Users\HP> kubectl create serviceaccount dashboard-admin  -n kubernetes-dashboard
serviceaccount/dashboard-admin created
PS C:\Users\HP> kubectl create clusterrolebinding dashboard-admin-binding --clusterrole=cluster-admin   --serviceaccount=kubernetes-dashboard:dashboard-admi
n
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-binding created
PS C:\Users\HP>
```

Step 6: Generate Dashboard Login Token

Run the following command to get the token:

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

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

```
PS C:\Users\HP> kubectl -n kubernetes-dashboard create token dashboard-admin
eyJhbGciOiJSUzI1NiIsImtpZCI6IlJwYzJMSjBxT3lWV0lxa0t4b1BNZ0ZFeGlLYjNZNldNdFptemU2SG9oMUUifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bHQuc3ZjLmNsdXN0ZXIub
G9jYWwiXSwiZXhwIjoxNzcwNzQ3MjAyLCJpYXQiOjE3NzA3NDM2MDIsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiNDU2YTMwZDAtMTFmNy00Z
DgyLTlhOWYtNjRiOTA3YjBlYmE2Iiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hcmQtYWRtaW4iL
CJ1aWQiOiI4MDllYjNiMi05Njg2LTQ3Y2ItOTE3MC02NjI1NjIxOGNjZDYifX0sIm5iZiI6MTc3MDc0MzYwMiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOmRhc
2hib2FyZC1hZG1pbiJ9.Gw04YPnu0yUZHaMbR-d_yNv7CbslycugZSL0Oih8wQ3BA2oFBqc-4BvrGeqBcQsMMkCjIHpQuFv-Gs6nVbXOwfLOVt51KZX_OFjx2Xn1_Dg-TRpE4jQyHx3_iTGUZM14iqs50p97
2p3DXPy8AsFhf_O5RnWt1VJ4uX0uGD2CY2tw_kApe8n5lcc978W7165I1bXByram-SMkBFVJtqe7GJ0q1eAnqrqoyAx4L2E1USJ2Y0XAmFh3KZr3uaUpxSMskprar6qkJh1drm_BC_cQG3EFLb0kxngdgXvM
f6lw9XYGa19Wop5rZVjRrkitqChekjUqwUeYH8n_sj3OUpl8BQ
PS C:\Users\HP>
```

Step 7: Start Kubernetes Dashboard

Run the proxy command:

```
kubectl proxy
```

Keep this terminal **running**.

```
PS C:\Users\HP> 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

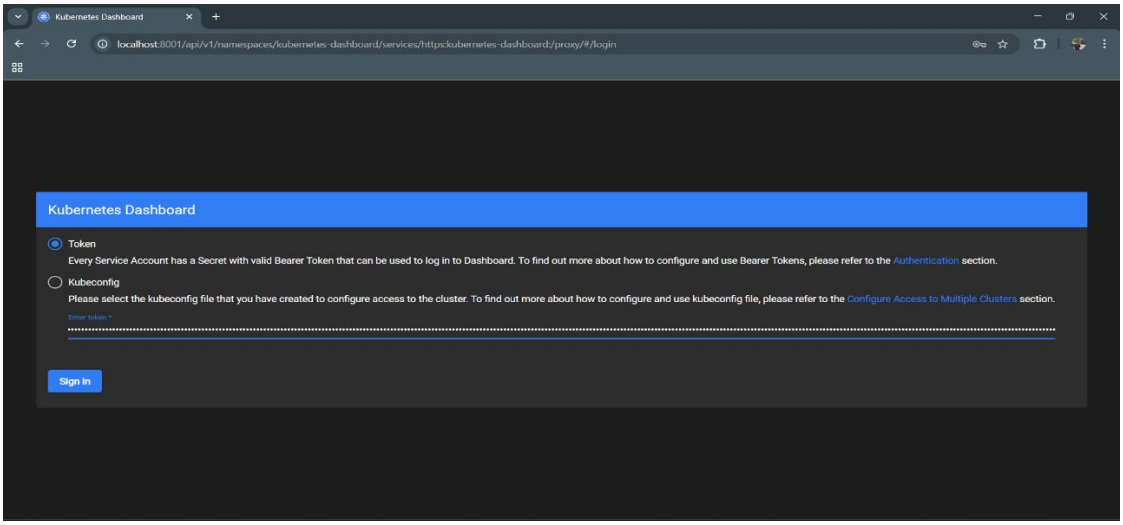
1. Select **Token** authentication
2. Paste the token generated earlier
3. Click **Sign In**

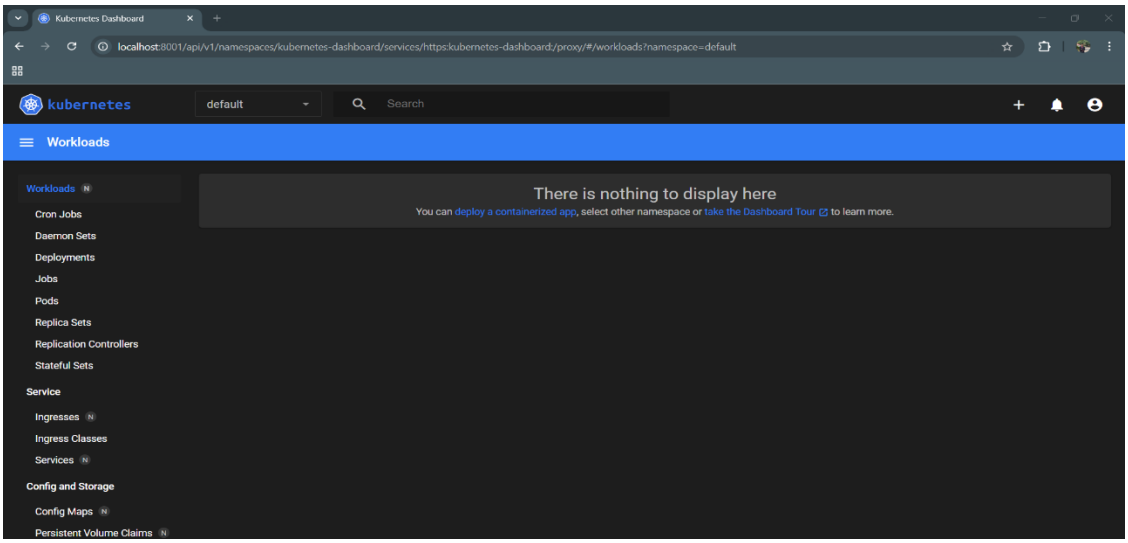
You should now see the **Kubernetes Dashboard UI**.

Step 10: Explore Dashboard

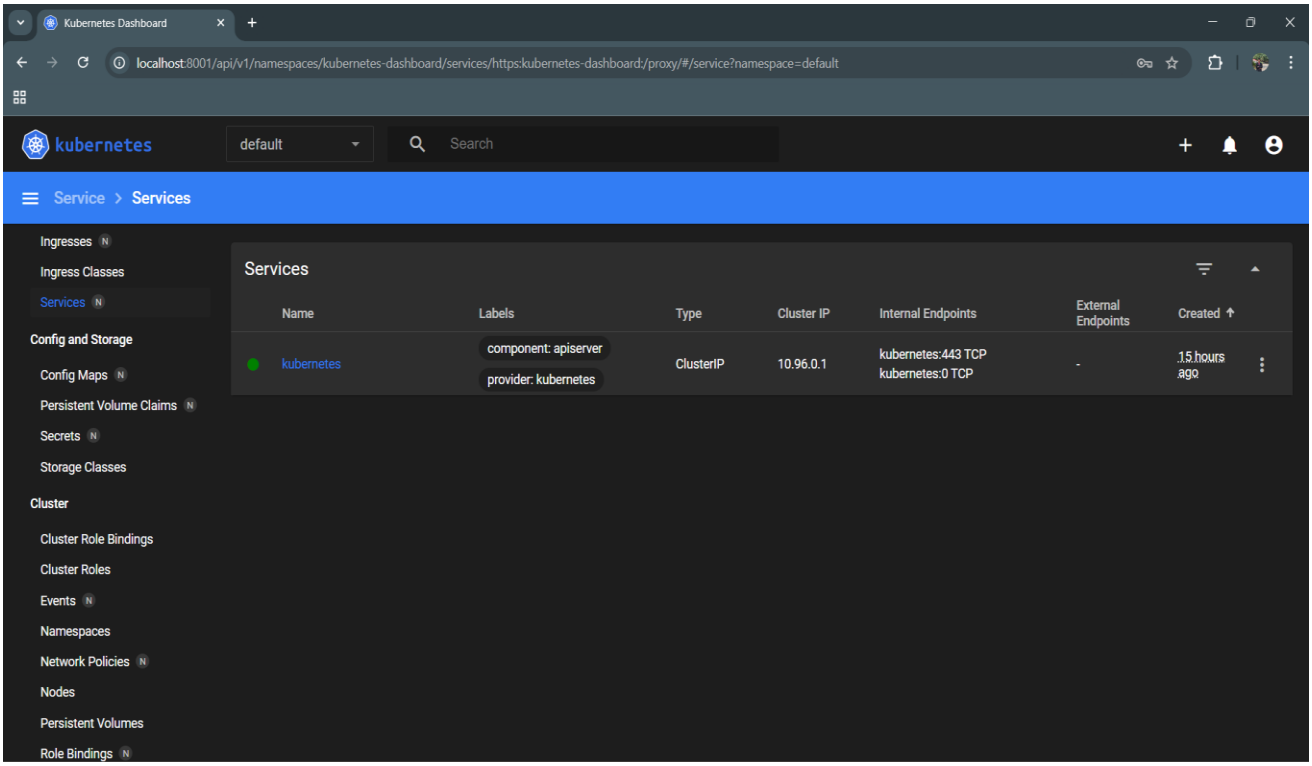
You can now view:

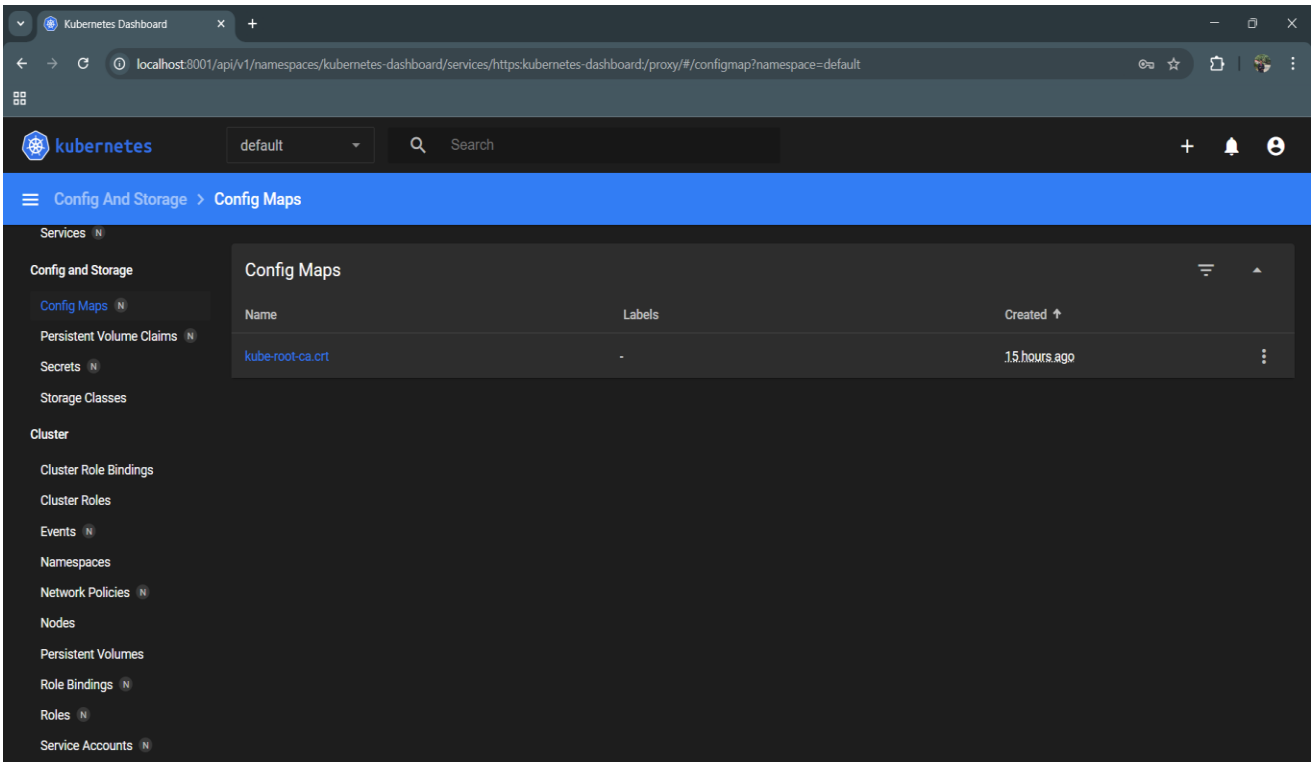
- Nodes
- Pods
- Deployments
- Services
- Namespaces
- ConfigMaps and Secrets

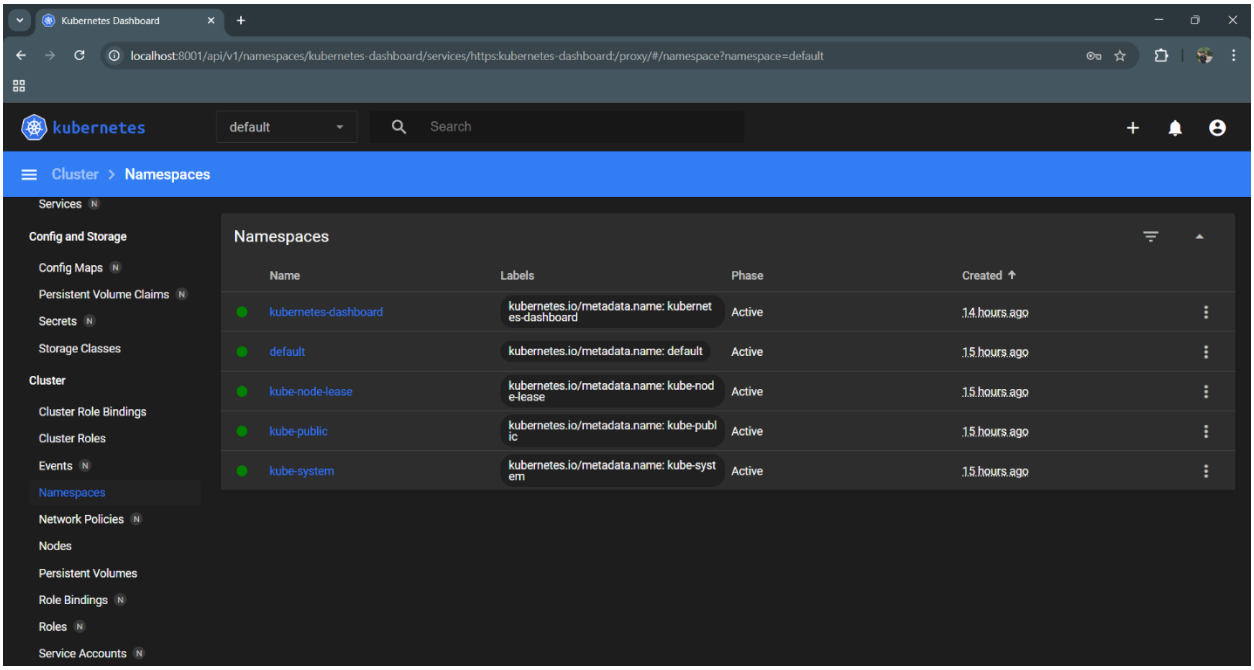


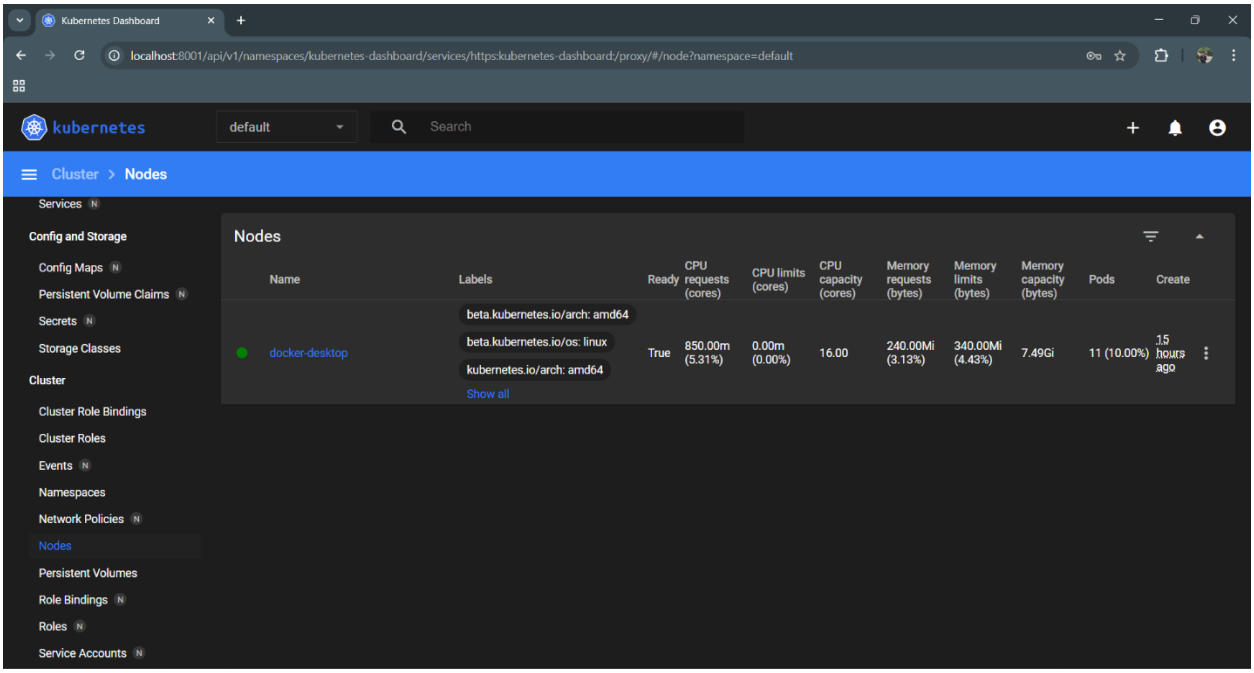


Shivang Mangal 500124202 Batch – 2 DevOps






Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/namespace?namespace=default
kubernetes
default
Search
Cluster > Namespaces
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Network Policies
Nodes
Persistent Volumes
Role Bindings
Roles
Service Accounts
Namespaces
Name
Labels
Phase
Created
kubernetes-dashboard
kubernetes.io/metadata.name: kubernetes-dashboard
Active
14 hours ago
default
kubernetes.io/metadata.name: default
Active
15 hours ago
kube-node-lease
kubernetes.io/metadata.name: kube-node-lease
Active
15 hours ago
kube-public
kubernetes.io/metadata.name: kube-public
Active
15 hours ago
kube-system
kubernetes.io/metadata.name: kube-system
Active
15 hours ago


Kubernetes Dashboard
localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/node?namespace=default
kubernetes
default
Search
Cluster > Nodes
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Events
Namespaces
Network Policies
Nodes
Persistent Volumes
Role Bindings
Roles
Service Accounts
Nodes
Name
Labels
Ready
CPU requests (cores)
CPU limits (cores)
CPU capacity (cores)
Memory requests (bytes)
Memory limits (bytes)
Memory capacity (bytes)
Pods
Create
docker-desktop
beta.kubernetes.io/arch: amd64
beta.kubernetes.io/os: linux
kubernetes.io/arch: amd64
Show all
True
850.00m (5.31%)
0.00m (0.00%)
16.00
240.00Mi (3.13%)
340.00Mi (4.43%)
7.49Gi
11 (10.00%)
15 hours ago