Lab Exercise 12 - Start and Access Kubernetes Dashboard

NAME : JIYA TYAGI

SAP ID: 500119743

BATCH 2 DEVOPS

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:

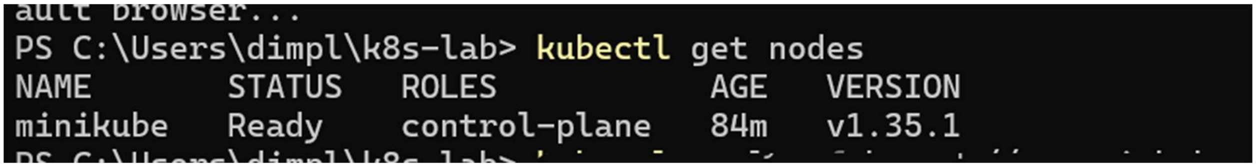
- kubectl version --client
- Check cluster status:
- kubectl cluster-info

Check nodes:

```
kubectl get nodes
```

Expected output:

Node status should be **Ready**

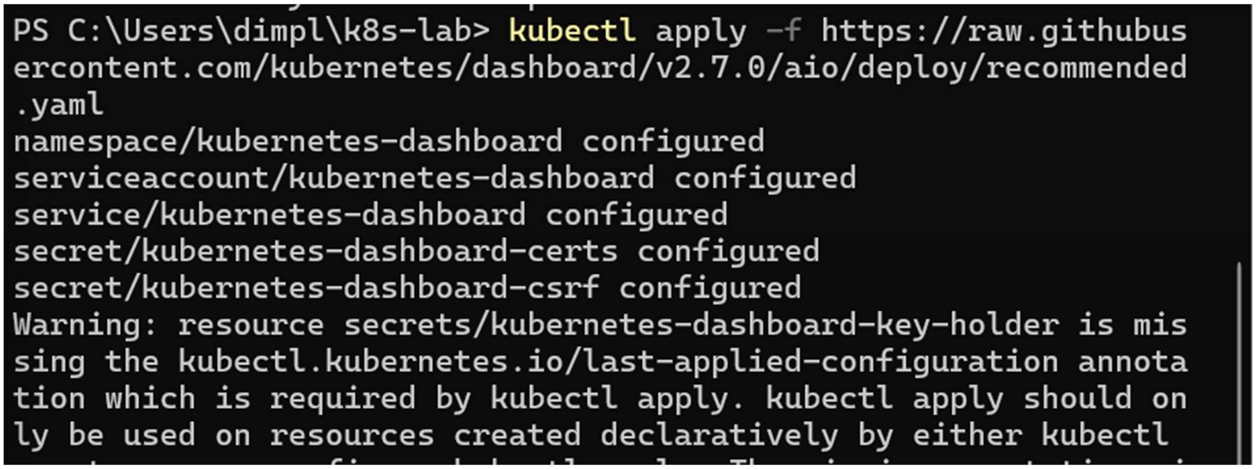


```
ault browser...
PS C:\Users\dimpl\k8s-lab> kubectl get nodes
NAME       STATUS   ROLES           AGE   VERSION
minikube   Ready    control-plane   84m   v1.35.1
```

Step 3: Deploy Kubernetes Dashboard

Apply the official Kubernetes Dashboard manifest:

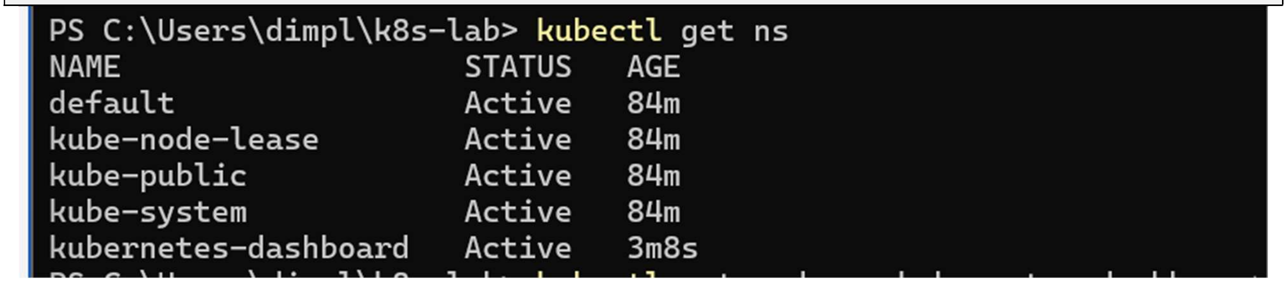
```
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml
```



```
PS C:\Users\dimpl\k8s-lab> kubectl apply -f https://raw.githubus
ercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended
.yaml
namespace/kubernetes-dashboard configured
serviceaccount/kubernetes-dashboard configured
service/kubernetes-dashboard configured
secret/kubernetes-dashboard-certs configured
secret/kubernetes-dashboard-csrf configured
Warning: resource secrets/kubernetes-dashboard-key-holder is mis
sing the kubectl.kubernetes.io/last-applied-configuration annota
tion which is required by kubectl apply. kubectl apply should on
ly be used on resources created declaratively by either kubectl
```

Verify namespace creation:

```
kubectl get ns
```



```
PS C:\Users\dimpl\k8s-lab> kubectl get ns
NAME                   STATUS   AGE
default                Active   84m
kube-node-lease        Active   84m
kube-public            Active   84m
kube-system            Active   84m
kubernetes-dashboard   Active   3m8s
```

You should see:

```
kubernetes-dashboard
```

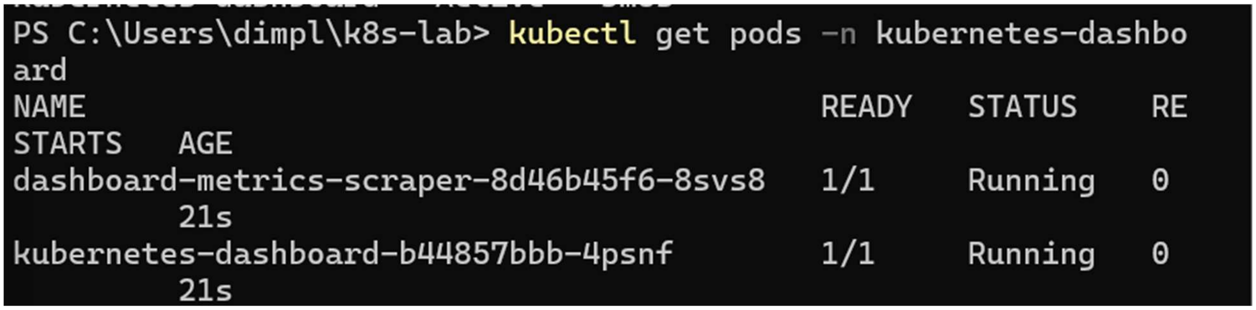
Step 4: Verify Dashboard Pods

Check dashboard pods:

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

Expected status:

Running

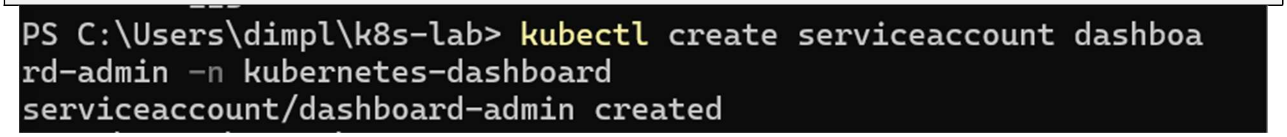


```
PS C:\Users\dimpl\k8s-lab> kubectl get pods -n kubernetes-dashbo
ard
NAME                                         READY   STATUS    RE
STARTS   AGE
dashboard-metrics-scraper-8d46b45f6-8svs8    1/1     Running   0
         21s
kubernetes-dashboard-b44857bbb-4psnf         1/1     Running   0
         21s
```

Step 5: Create Admin User for Dashboard Access

Create a service account:

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

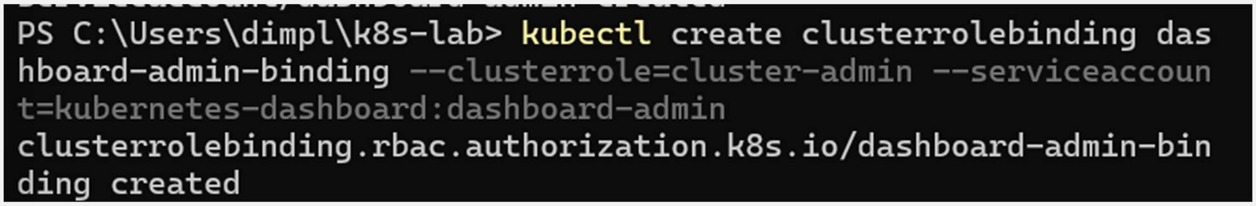


```
PS C:\Users\dimpl\k8s-lab> kubectl create serviceaccount dashboa
rd-admin -n kubernetes-dashboard
serviceaccount/dashboard-admin created
```

Create cluster role binding:

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

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

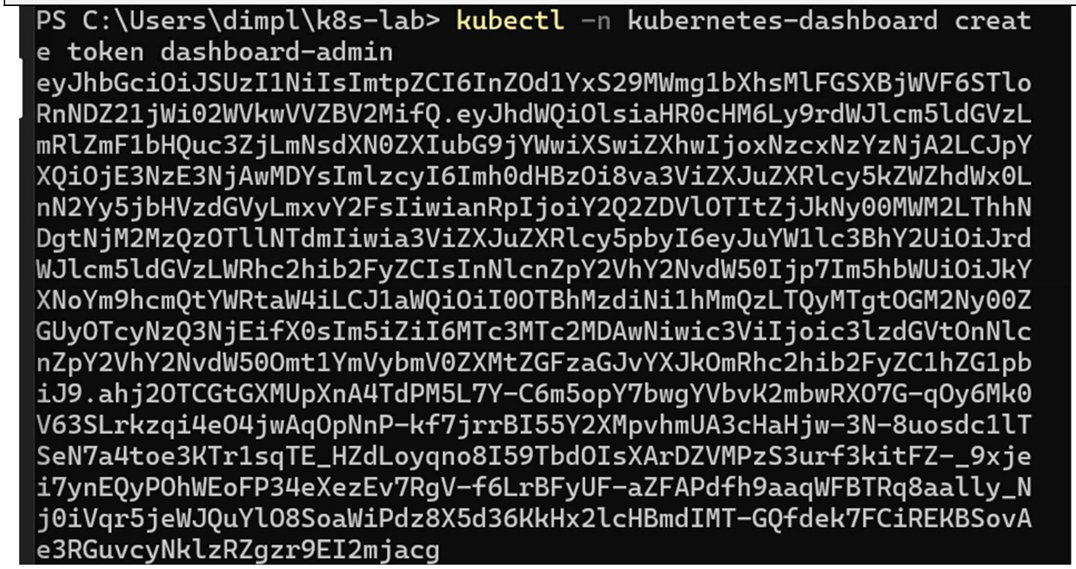


```
PS C:\Users\dimpl\k8s-lab> kubectl create clusterrolebinding das
hboard-admin-binding --clusterrole=cluster-admin --serviceaccoun
t=kubernetes-dashboard:dashboard-admin
clusterrolebinding.rbac.authorization.k8s.io/dashboard-admin-bin
ding created
```

Step 6: Generate Dashboard Login Token

Run the following command to get the token:

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



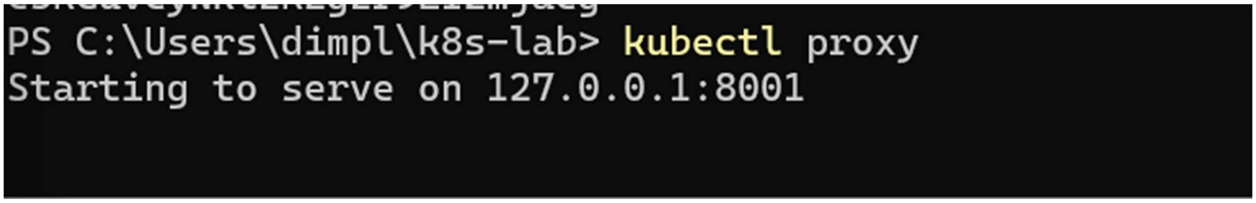
```
PS C:\Users\dimpl\k8s-lab> kubectl -n kubernetes-dashboard creat
e token dashboard-admin
eyJhbGciOiJSUzI1NiIsImtpZCI6InZOd1YxS29MWmg1bXhsMlFGSXBjWVF6STlo
RnNDZ21jWi02WVkwVVZBV2MifQ.eyJhdWQiOlsiaHR0cHM6Ly9rdWJlcm5ldGVzL
mRlZmF1bHQuc3ZjLmNsdXN0ZXIubG9jYWwiXSwiZXhwIjoxNzcxNzYzNjA2LCJpY
XQiOjE3NzE3NjAwMDYsImlzcyI6Imh0dHBzOi8va3ViZXJuZXRlcy5kZWZhdWx0L
nN2Yy5jbHVzdGVyLmxvY2FsIiwianRpIjoiY2Q2ZDVlOTItZjJkNy00MWM2LThhN
DgtNjM2MzQzOTllNTdmIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrd
WJlcm5ldGVzLWRhc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkY
XNoYm9hcmQtYWRtaW4iLCJ1aWQiOiI0OTBhMzdiNi1hMmQzLTQyMTgtOGM2Ny00Z
GUyOTcyNzQ3NjEifX0sIm5iZiI6MTc3MTc2MDAwNiwic3ViIjoic3lzdGVtOnNlc
nZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaGJvYXJkOmRhc2hib2FyZC1hZG1pb
iJ9.ahj2OTCGtGXMUpXnA4TdPM5L7Y-C6m5opY7bwgYVbvK2mbwRXO7G-qOy6Mk0
V63SLrkzqi4eO4jwAqOpNnP-kf7jrrBI55Y2XMpvhmUA3cHaHjw-3N-8uosdc1lT
SeN7a4toe3KTr1sqTE_HZdLoyqno8I59TbdOIsXArDZVMPzS3urf3kitFZ-_9xje
i7ynEQyPOhWEoFP34eXezEv7RgV-f6LrBFyUF-aZFAPdfh9aaqWFBTRq8aally_N
j0iVqr5jeWJQuYlO8SoaWiPdz8X5d36KkHx2lcHBmdIMT-GQfdek7FCiREKBSovA
e3RGuvcyNklzRZgzr9EI2mjacg
```

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

Step 7: Start Kubernetes Dashboard

Run the proxy command:

```
kubectl proxy
```



```
PS C:\Users\dimpl\k8s-lab> kubectl proxy
Starting to serve on 127.0.0.1:8001
```

Keep this terminal **running**.

Step 8: Access Kubernetes Dashboard in Browser

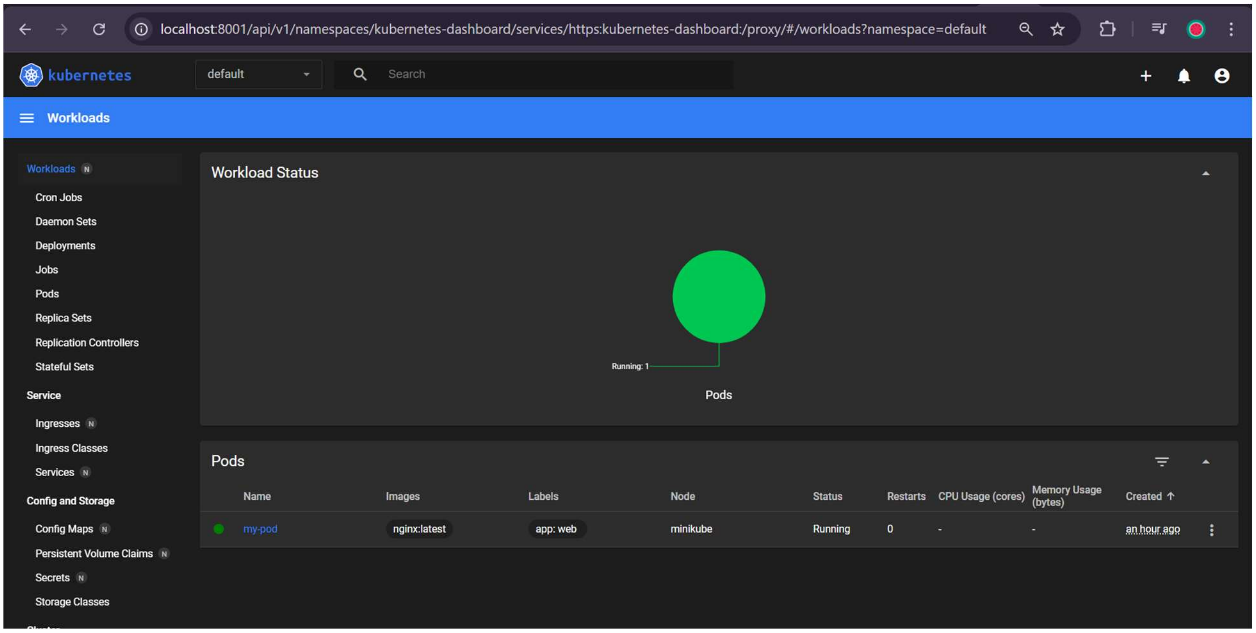
Open a web browser and paste the following URL:

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

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**.



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

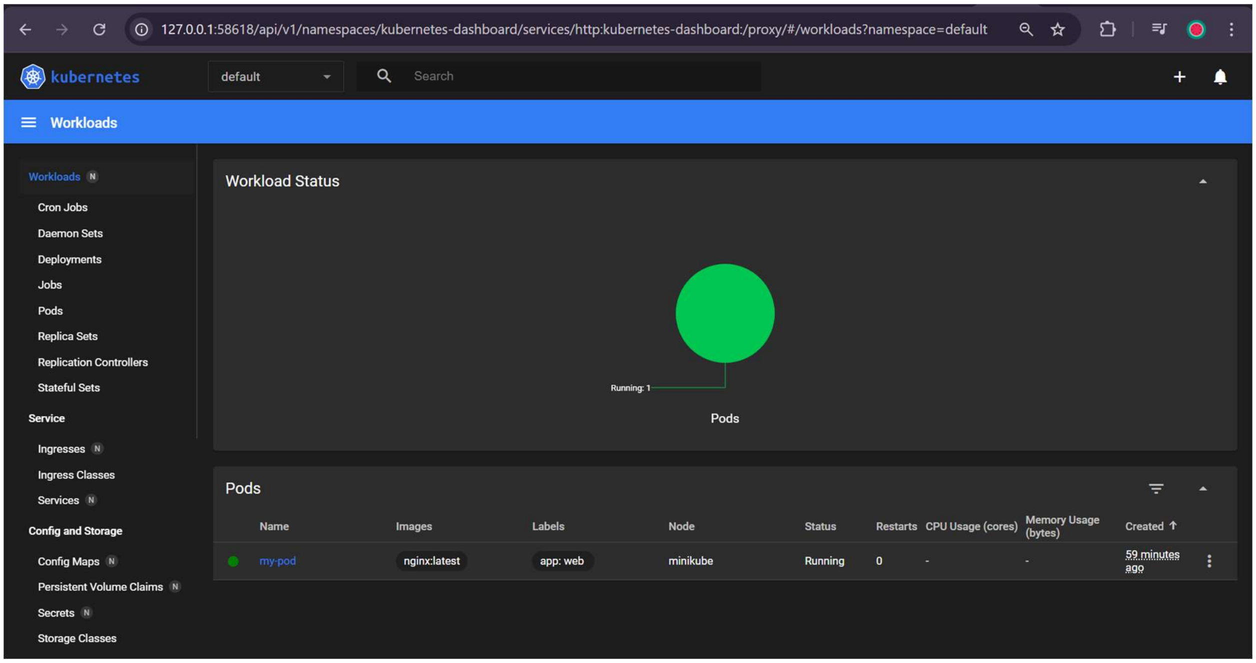
Pods

	Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
●	my-pod	nginx:latest	app: web	minikube	Running	0	-	-	an hour ago

Step 10: Explore Dashboard

You can now view:

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



127.0.0.1:58618/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/workloads?namespace=default

	Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created ↑
●	my-pod	nginx:latest	app: web	minikube	Running	0	-	-	59 minutes ago