

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).

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage ¹
No containers are running.

Container memory usage
No containers are running.

Search

Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	Created	Actions
<input type="checkbox"/>	epic_hoover	d71aa15596a9	combined-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	objective_cerf	bb9f785d7054	combined-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	blissful_chatterjee	26705d4f1cde	entrypoint-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	infallible_lumiere	612bc5166a4a	entrypoint-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	romantic_heisenberg	ac7fe851bb82	entrypoint-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	nice_ride	1398a5418814	cmd-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	serene_shamir	4a4f7eb0ad30	cmd-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	busy_chebyshev	0c03404f1e0a	cmd-example		N/A 14 days ago	Play Stop Delete
<input type="checkbox"/>	affectionate_bardeen	4790d53ba747	nginx:latest		N/A 3 months ago	Play Stop Delete
<input type="checkbox"/>	loving_hopper	4600600d4d20	nginx:latest		N/A 2 months ago	Play Stop Delete

Showing 14 items

Engine running | Kubernetes starting RAM 0.00 GB CPU 0.00% Disk -- GB used (limit -- GB) [Terminal](#) [New version available](#)

Settings [Give feedback](#)

Search settings

General
Resources
Docker Engine
Builders
Kubernetes
Software updates
Extensions
Beta features
Notifications

Kubernetes

☒ Enable Kubernetes
Start a Kubernetes single or multi-node cluster when starting Docker Desktop.

Cluster

docker-desktop
kubeadm, 1 node, v1.32.2

Starting
preparing configuration [Reset cluster](#)

Cluster settings

Choose cluster provisioning method

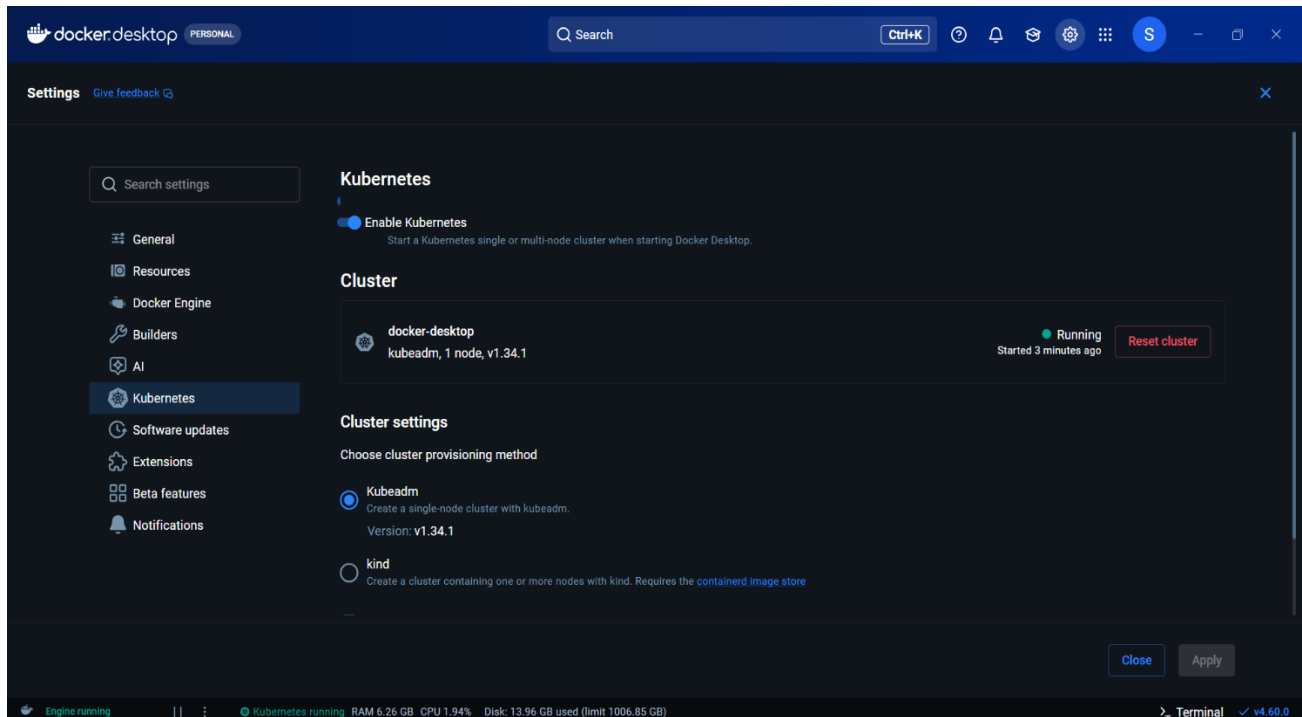
☒ Kubeadm
Create a single-node cluster with kubeadm.
Version: v1.32.2

☐ kind
Create a cluster containing one or more nodes with kind. Requires the [containerd image store](#)

☐ Show custom containers (advanced)

[Close](#) [Apply](#)

Engine running | Kubernetes starting RAM 2.30 GB CPU 56.95% Disk 11.10 GB used (limit 1006.85 GB) [Terminal](#) [New version available](#)



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**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\HP> docker --version
Docker version 29.2.0, build 0b9d198
PS C:\Users\HP> kubectl version --client
Client Version: v1.34.1
Kustomize Version: v5.7.1
```

```
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-admin
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
eyJhbGciOiJSUzI1NiIsImtpZCI6IiJmYzJMSjBxT3lwV0lxa0t4b1BNZ0ZFeGllYjNZNDNdFtemU2SG9oMUUifQ.eyJhdWQiOi0lsiaHR0cHM6Ly9rdWJlcm5ldGVzLmRlZmF1bH0uc3ZjLmNsdXN0ZXIubG9jYVwwiXSwiZmxwIjoxNzcwNzQ3MjAyLCJpYXQ0jE3NzA3NDM2MDIsImZcyI6Imh0dHBz0i8va3ViZXJlcy5kZWZhdWx0LnN2Yy5jbHVzdGVyLmV2Y2F5IiwianRpIjoibDU2YTMwZDA0MTFmNy00ZDgyLTlhOWYtNjRiOTA3YjBLYmE2Iiwia3ViZXJlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLWRRc2hib2FyZCIsInNlcnZpY2VhY2NvdW50Ijp7Im5hbWUiOiJkYXNoYm9hcmQtYWRtaW4iLmRlZmF1bH0uc3ZjLmNsdXN0ZXIuMj03Y2I0TE3MC02NjI1NjIxOGNjZDViX0sIm5iZiI6MTc3MDc0MzYwMiwiIiwic3ViIjoic3ZldGVtOnNlcnZpY2VhY2NvdW50Omt1YmVybmV0ZXMtZGFzaG9vYXJkOmRhc2hib2FyZC1hZG1pbjI9. Gw04Ypnu0yUZHambR-d_yNv7CbslycugZSL00ih8wQ3BA2oFBqc-4BvrGegBcQsMMKcJIHpQuFv-Gs6nVbX0wFLOvt51KZX_OFjx2Xn1_Dg-TRpE4jQyHx3_iTGUZM14iqs50p972p3DXPy8AsFhf_05RnWt1VJ4uX0uGD2CY2tw_kApe8n5Lcc978W7165I1bXByram-SMkBFVJtqe7GJ0q1eAnqrqoyAx4L2E1USJ2Y0XAmFh3KZr3uaUpXSMskprar6qkJh1drn_BC_cQG3EFLb0kxngdgXvMf6Lw9XYGa19Wop5rZVjRrkittqChekjUqWUeYH8n_sj30Upl8BQ
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

