

EXPERIMENT-08

AIM: Integrate Kubernetes and Docker

Objective: To integrate Kubernetes with Docker for orchestrating containerized applications using Pods, Deployments, and Services.

Software Requirements

- Docker installed (<https://www.docker.com/products/docker-desktop/>)
 - Kubernetes installed (Minikube or Docker Desktop with Kubernetes enabled)
 - kubectl command-line tool
 - Code editor (VS Code or any)
 - Sample containerized application
-

The Kubernetes server runs as a single or multi-node cluster, within Docker container(s). This lightweight setup helps you explore Kubernetes features, test workloads, and work with container orchestration in parallel with other Docker functionalities.

Docker Desktop includes a standalone Kubernetes server and client, as well as Docker CLI integration that runs on your machine.

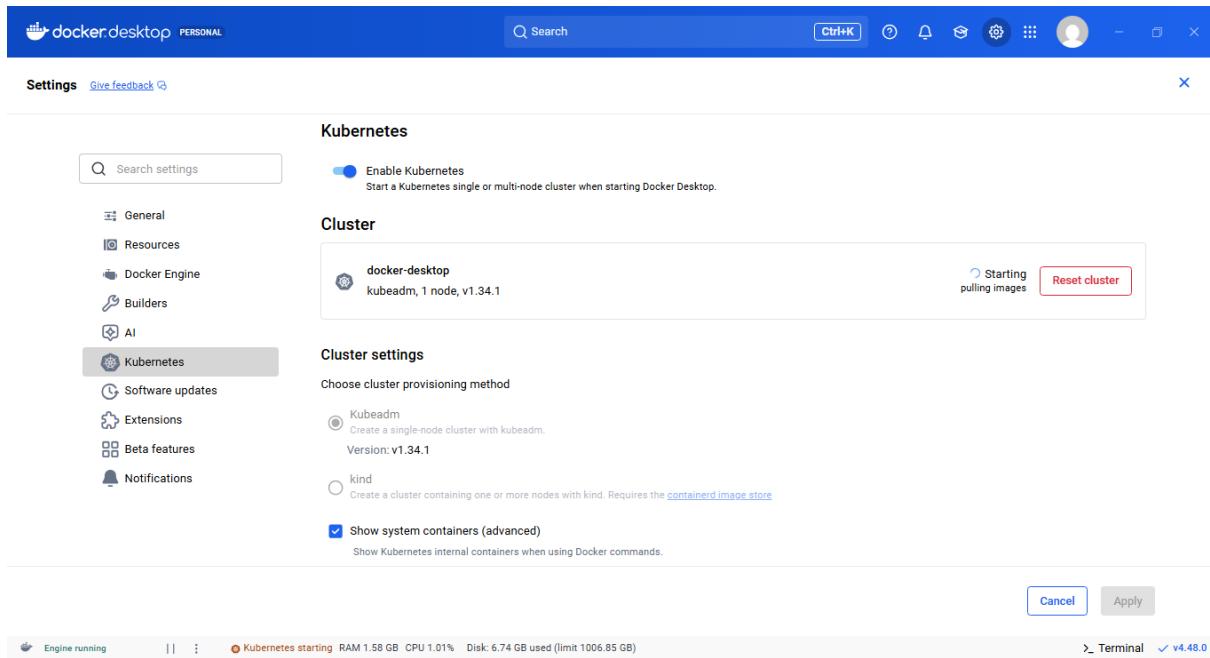
The Kubernetes server runs locally within your Docker instance, is not configurable, and is a single-node cluster. It runs within a Docker container on your local system, and is only for local testing.

Turning on Kubernetes allows you to deploy your workloads in parallel, on Kubernetes, Swarm, and as standalone containers. Turning on or off the Kubernetes server does not affect your other workloads.

Lab Procedure

Install and turn on Kubernetes

1. Open the Docker Desktop Dashboard and navigate to Settings.
2. Select the Kubernetes tab.
3. Toggle on Enable Kubernetes.
4. Choose your [cluster provisioning method](#).
5. Select Apply to save the settings.



This sets up the images required to run the Kubernetes server as containers, and installs the `kubectl` command-line tool on your system at:

`/usr/local/bin/kubectl` (Mac)

or

`C:\Program Files\ Docker\ Docker\resources\bin\kubectl.exe` (Windows).

Docker Desktop for Linux does not include `kubectl` by default. You can install it separately by following the [Kubernetes installation guide](#). Ensure the `kubectl` binary is installed at `/usr/local/bin/kubectl`.

When Kubernetes is enabled, its status is displayed in the Docker Desktop Dashboard footer and the Docker menu.

Docker Desktop does not upgrade your Kubernetes cluster automatically after a new update. To upgrade your Kubernetes cluster to the latest version, select

Reset Kubernetes Cluster

You can check which version of Kubernetes you're on with:

`kubectl version`

```
C:\Users\User>kubectl version
Client Version: v1.34.1
Kustomize Version: v5.7.1
Server Version: v1.34.1
```

Using the `kubectl` command

Kubernetes integration automatically installs the Kubernetes CLI command at:

/usr/local/bin/kubectl on Mac and
C:\Program Files\ Docker\ Docker\ Resources\ bin\ kubectl.exe on Windows.

This location may not be in your shell's **PATH** variable, so you may need to type the full path of the command or add it to the **PATH**.

If you have already installed **kubectl** and it is pointing to some other environment, such as **minikube** or a Google Kubernetes Engine cluster, ensure you change the context so that **kubectl** is pointing to **docker-desktop**:

kubectl config get-contexts

kubectl config use-context docker-desktop

```
C:\Users\User>kubectl config get-contexts
CURRENT      NAME          CLUSTER          AUTHINFO          NAMESPACE
*           docker-desktop  docker-desktop  docker-desktop

C:\Users\User>kubectl config use-context docker-desktop
Switched to context "docker-desktop".
```

Tip

If the **kubectl config get-contexts** command returns an empty result, try:

- Running the command in the Command Prompt or PowerShell.
- Setting the **KUBECONFIG** environment variable to point to your **.kube/config** file.

Verify installation

To confirm that Kubernetes is running, list the available nodes:

kubectl get nodes

```
C:\Users\User>kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
docker-desktop  Ready    control-plane  6d1h   v1.34.1
```

If you installed **kubectl** using Homebrew, or by some other method, and experience conflicts, remove **/usr/local/bin/kubectl**.
