Deploy on Kubernetes

*Estimated reading time: 3 minutes*

Docker Desktop includes a standalone Kubernetes server and client, as well as Docker CLI integration. The Kubernetes server runs locally within your Docker instance, is not configurable, and is a single-node cluster.

The Kubernetes server runs within a Docker container on your local system, and is only for local testing. When Kubernetes support is enabled, you can deploy your workloads, in parallel, on Kubernetes, Swarm, and as standalone containers. Enabling or disabling the Kubernetes server does not affect your other workloads.

See [Docker Desktop for Windows > Getting started](https://docs.docker.com/docker-for-windows/#kubernetes) to enable Kubernetes and begin testing the deployment of your workloads on Kubernetes.

Use Docker commands

You can deploy a stack on Kubernetes with docker stack deploy, the docker-compose.yml file, and the name of the stack.

docker stack deploy --compose-file /path/to/docker-compose.yml mystack

docker stack services mystack

You can see the service deployed with the kubectl get services command.

Specify a namespace

By default, the default namespace is used. You can specify a namespace with the --namespace flag.

docker stack deploy --namespace my-app --compose-file /path/to/docker-compose.yml mystack

Run kubectl get services -n my-app to see only the services deployed in the my-app namespace.

Override the default orchestrator

While testing Kubernetes, you may want to deploy some workloads in swarm mode. Use the DOCKER\_STACK\_ORCHESTRATOR variable to override the default orchestrator for a given terminal session or a single Docker command. This variable can be unset (the default, in which case Kubernetes is the orchestrator) or set to swarm or kubernetes. The following command overrides the orchestrator for a single deployment, by setting the variable before running the command.

set DOCKER\_STACK\_ORCHESTRATOR=swarm

docker stack deploy --compose-file /path/to/docker-compose.yml mystack

Alternatively, the --orchestrator flag may be set to swarm or kubernetes when deploying to override the default orchestrator for that deployment.

docker stack deploy --orchestrator swarm --compose-file /path/to/docker-compose.yml mystack

**Note**

Deploying the same app in Kubernetes and swarm mode may lead to conflicts with ports and service names.

Use the kubectl command

The windows Kubernetes integration provides the Kubernetes CLI command at C:\>Program Files\Docker\Docker\Resources\bin\kubectl.exe. 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. For more information about kubectl, see the [official kubectl documentation](https://kubernetes.io/docs/reference/kubectl/overview/). You can test the command by listing the available nodes:

kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready master 3h v1.8.2

Example app

Docker has created the following demo app that you can deploy to swarm mode or to Kubernetes using the docker stack deploy command.

version: "3.9"

services:

web:

image: dockersamples/k8s-wordsmith-web

ports:

- "80:80"

words:

image: dockersamples/k8s-wordsmith-api

deploy:

replicas: 5

endpoint\_mode: dnsrr

resources:

limits:

memory: 50M

reservations:

memory: 50M

db:

image: dockersamples/k8s-wordsmith-db

If you already have a Kubernetes YAML file, you can deploy it using the kubectl command.