Please read this: <https://www.docker.com/blog/docker-hearts-wsl-2/>

We will replace the Hyper-V VM we currently use by a WSL 2 integration package. This package will provide the same features as the current Docker Desktop VM: Kubernetes 1-click setup, automatic updates, transparent HTTP proxy configuration, access to the daemon from Windows, transparent bind mounts of Windows files, and more.

A technical preview of Docker Desktop for WSL 2 will be available for download in July. It will run side by side with the current version of Docker Desktop, so you can continue to work safely on your existing projects. If you are running the latest Windows Insider build, you will be able to experience this first hand. In the coming months, we will add more features until the WSL 2 architecture is used in Docker Desktop for everyone running a compatible version of Windows.

Additionally I recommend to watch this video: <https://www.youtube.com/watch?v=lwhMThePdIo>

Good comparison you will find here: <https://blog.logrocket.com/working-with-node-js-on-hyper-v-and-wsl2/>

cite:

Comparing Hyper-V to WSL2

The biggest difference between running Ubuntu Linux in a Hyper-V virtual machine versus running the operating system in WSL2 lies in the ability to access the Ubuntu user interface in Hyper-V.

The user interface allows you to install and use more than just command-line tools.

Depending on your system’s hardware performance, you likely found that WSL2 is the faster option. To expedite the process of running Ubuntu Linux on Hyper-V, you could set up SSH access to the virtual machine.

This speeds up command-line access. However, WSL2 may still have the upper hand since it does not require SSH to enable access.

WSL INTEGRATION

In WSL 2 mode, you can configure which WSL 2 distributions will have the Docker WSL integration.

By default, the integration will be enabled on your default WSL distribution. To change your default WSL distro, run wsl --set-default <distro name>. (For example, to set Ubuntu as your default WSL distro, run wsl --set-default ubuntu).

You can also select any additional distributions you would like to enable the WSL 2 integration on.

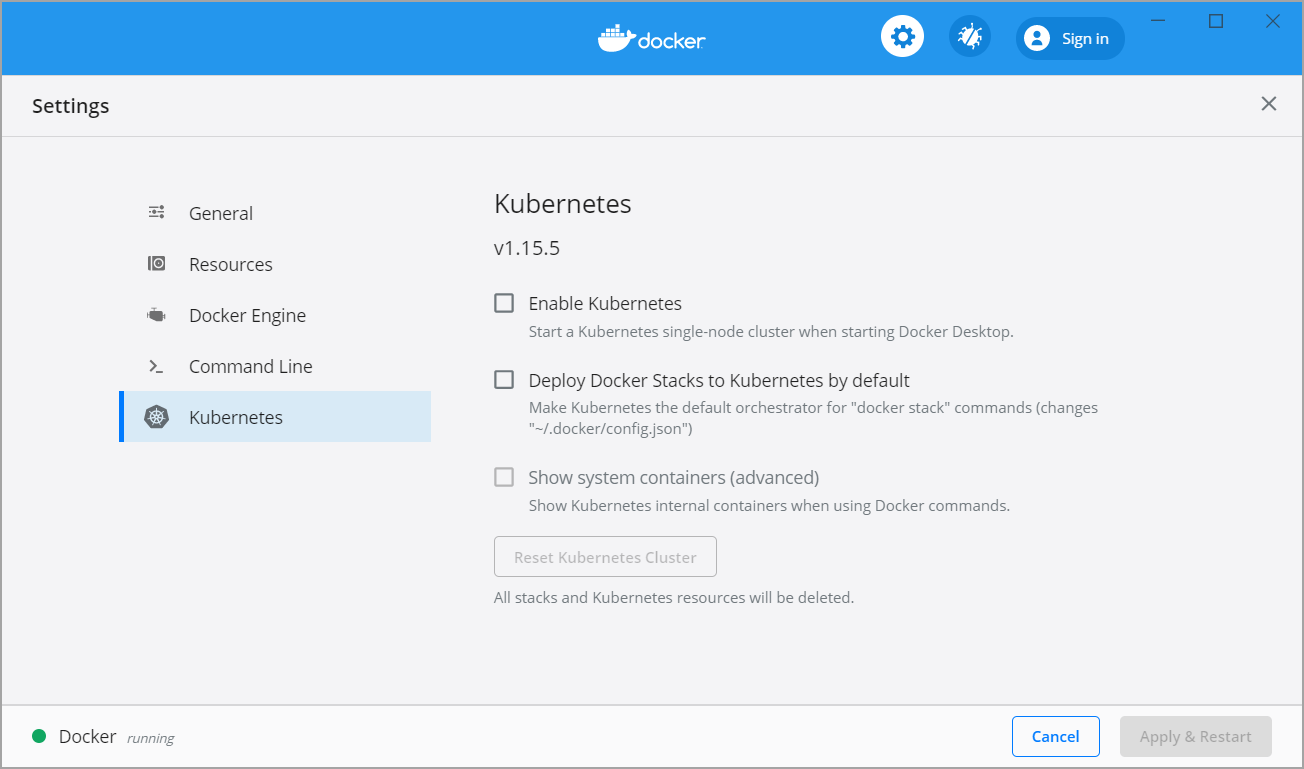
For more details on configuring Docker Desktop to use WSL 2, see [Docker Desktop WSL 2 backend](https://docs.docker.com/docker-for-windows/wsl/).

### **Kubernetes**

**Note**

The Kubernetes tab is not available in Windows container mode.

Docker Desktop includes a standalone Kubernetes server that runs on your Windows host, so that you can test deploying your Docker workloads on Kubernetes.



The Kubernetes client command, kubectl, is included and configured to connect to the local Kubernetes server. If you have kubectl already installed and pointing to some other environment, such as minikube or a GKE cluster, be sure to change context so that kubectl is pointing to docker-desktop:

> kubectl config get-contexts

> kubectl config use-context docker-desktop

To enable Kubernetes support and install a standalone instance of Kubernetes running as a Docker container, select **Enable Kubernetes**.

To set Kubernetes as the [default orchestrator](https://docs.docker.com/docker-for-windows/kubernetes/#override-the-default-orchestrator), select **Deploy Docker Stacks to Kubernetes by default**.

By default, Kubernetes containers are hidden from commands like docker service ls, because managing them manually is not supported. To make them visible, select **Show system containers (advanced)**. Most users do not need this option.

Click **Apply & Restart** to save the settings. This instantiates images required to run the Kubernetes server as containers, and installs the kubectl.exe command in the path.

* When Kubernetes is enabled and running, an additional status bar item displays at the bottom right of the Docker Desktop Settings dialog. The status of Kubernetes shows in the Docker menu and the context points to docker-desktop.
* To disable Kubernetes support at any time, clear the **Enable Kubernetes** check box. The Kubernetes containers are stopped and removed, and the /usr/local/bin/kubectl command is removed.
* To delete all stacks and Kubernetes resources, select **Reset Kubernetes Cluster**.
* If you installed kubectl by another method, and experience conflicts, remove it.

For more information on using the Kubernetes integration with Docker Desktop, see [Deploy on Kubernetes](https://docs.docker.com/docker-for-windows/kubernetes/).