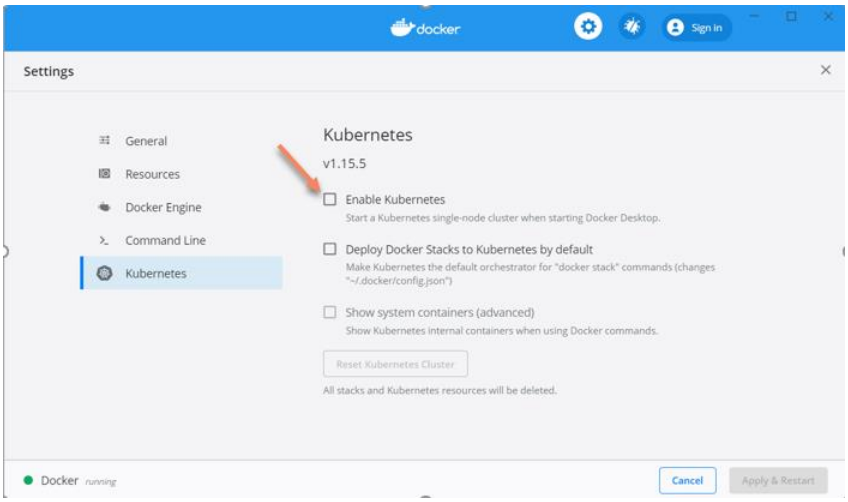
Deploying Applications to Kubernetes

1. Docker Desktop
2. Minikube

Kubernetes Installation

A local machine Kubernetes solution can help developers to configure and run a Kubernetes cluster in their local development environments and test their application during all development phases, without investing significant effort to configure and manage a Kubernetes cluster.

Docker Desktop for Windows and Mac includes a standalone Kubernetes server that runs on our Windows host, so that we can test deploying our Docker workloads on Kubernetes.



- To enable Kubernetes support and install a standalone instance of Kubernetes running as a Docker container, select **Enable Kubernetes**.
- This instantiates images required to run the Kubernetes server as containers, and installs the **kubectl.exe** command in the path. If we have kubectl already installed and pointing to some other environment, such as minikube, be sure to change context so that kubectl is pointing to docker-desktop:
- 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** (Kubernetes cluster)
- To delete all stacks and Kubernetes resources, select **Reset Kubernetes Cluster**.
- 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.

Note: By default, Kubernetes containers are hidden from commands like **docker service ls**, because managing them manually is not supported. To make them visible, check "Show system containers (advanced)" checkbox under Kubernetes menu.

Testing the installation

We can run a quick and easy test, to make sure that Kubernetes is actually running on the machine. Open command prompt / terminal window and run the command:

```
kubectl version
```

To get detailed information about the cluster:

```
kubectl cluster-info
```

Kubernetes should report that both Kubernetes master and KubeDNS are running on localhost:6443

Kubectl performs all its operations against the current context:

```
kubectl config get-contexts
```

Note: Following files contains all Clusters and Contexts information

Windows: C:\Users\<user-name>\.kube**config**

Linux/Mac: /home/<username>/.kube/**config**

To set the current context:

```
kubectl config use-context  minikube
```

View the cluster and context configuration

```
kubectl config view
```

Note that the above command shows the content of the file C:\Users\<user-name>\.kube**config OR** /home/training/.kube/config

Setting up the Kubernetes tooling on Windows 10 WSL
https://itnext.io/setting-up-the-kubernetes-tooling-on-windows-10-wsl-d852ddc6699c

Online Emulator: https://labs.play-with-k8s.com/

Installing Minikube on Ubuntu

Update System and install packages

sudo apt-get update -y

sudo apt-get upgrade -y

sudo apt-get install curl

sudo apt-get install apt-transport-https

Install VirtualBox Hypervisor

sudo apt install virtualbox virtualbox-ext-pack

Install Minikube

wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64

sudo cp minikube-linux-amd64 /usr/local/bin/minikube

sudo chmod 755 /usr/local/bin/minikube

minikube version

Install Kubectl

curl -LO https://storage.googleapis.com/kubernetes-release/release/`curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl

chmod +x ./kubectl

sudo mv ./kubectl /usr/local/bin/kubectl

kubectl version -o json

Start Minikube

minikube start

kubectl config view

kubectl cluster-info

kubectl get nodes

kubectl get pod

Other Minikube comands

minikube status

minikube stop

minikube delete

minikube addons list

minikube dashboard

Reference: https://phoenixnap.com/kb/install-minikube-on-ubuntu

Installing Minikube on Windows

Docker Desktop for Windows/Mac uses Type-1 hypervisor such as Hyper-V, which are better compared to Type-2 hypervisors, such as VirtualBox. Minikube supports both hypervisors. Unfortunately, there are limitations in which technology we are using, since we cannot have Type-1 or Type-2 hypervisors running at the same time on our machine:

Hyper-V can run on three versions of Windows 10: Windows 10 Enterprise, Windows 10 Professional, and Windows 10 Education.

Step1) Install a Hypervisor

If we do not already have a hypervisor installed, install one of these:

- Hyper-V
- VirtualBox

Step2: Install Chocolatey package manager for Windows.

For installation of Chocolatey, use the following command from PowerShell in administrative mode:

```
PS:> Set-ExecutionPolicy Bypass -Scope Process -Force; iex ((New-Object
System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))
```

Step3: We can **install kubectl** according to the instructions available at https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl-on-windows

Option 1) Install Kubectl.exe using Chocolatey command.

```
choco install kubernetes-cli
```

Step4: Minikube Installation

Install Minikube using Chocolatey: The easiest way to install Minikube on Windows is using Chocolatey (run as an administrator):

C:\> choco install minikube

After Minikube has finished installing, close the current CLI session and restart. Minikube should have been added to our path automatically.

Step4: Start Minikube and create a cluster:

C:\> minikube start

C:\> minikube status

C:\> minikube stop

Command to redirect docker cli to minikube host (On Windows / Mac with Docker Desktop installed)

echo $(minikube docker-env)

eval $(minikube docker-env)

docker ps