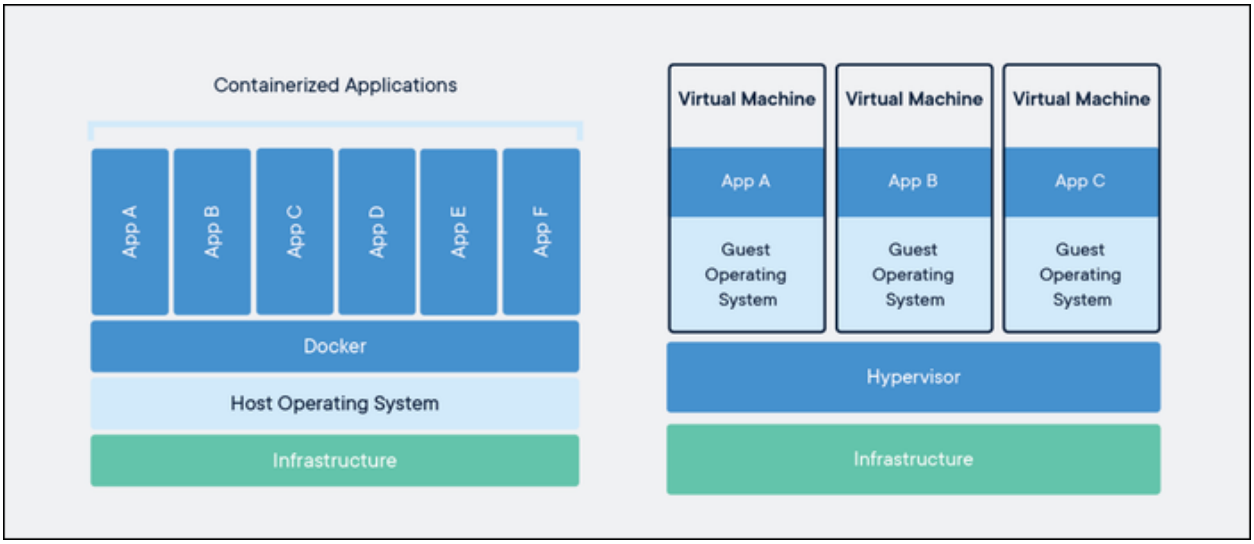
Disclaimer: if you intend to use our native binaries with our VS Code extension, then this Docker setup is unnecessary.

Getting Started with Docker

A Guide to What Docker is & How to Install it

Introduction

So what is Docker? You can think of Docker as a middleman between full on virtualization and running something normally on your hardware. While virtualization would emulate an entire operating system on your computer, Docker uses something called containers. Containers, unlike VMs, use the host operating system and virtualizes the software running.



Containers only have exactly what they need inside of them, with nothing else. This means they are generally super lightweight so you can run a whole bunch of them on one machine. Instead of needing an entire desktop/virtual machine to run a simple application, like a web server for example, you can just throw it in a container and call it a day.

In a VM you call what manages the VM's the hypervisor (here's more information on hypervisors if you want to learn more), in Docker that's the Docker Engine.

Installing Docker

Docker is available for Linux, MacOS, and Windows, making it something very useful for developers who run into the “but it works on my machine” issue. It can also allow you to create a container, and anyone can run it on nearly any device without caring about the operating system. We will be using a package manager for all the installations. If you want to learn more about them/need to install one on your system, refer to the document Package Managers.

Windows Installation

There are many ways to install Docker on Windows, but we will be using Chocolatey, a package manager for Windows.

In an administrator PowerShell console, run the command. There may be a confirmation request in the console, confirm it or the installation will halt.

choco install docker-desktop

MacOS Installation

We are going to be using Homebrew for this, so install it if you haven't already. All you need to do is run the following command in your terminal if you have Homebrew installed.

brew install --cask docker

Running the Docker Desktop

For Windows and MacOS you need to run the Docker Desktop application that was installed on your local computer during the above steps. You need to do this to start the Docker Engine. Go through the recommended setup options (there is no need to create an account right now, skip where you can).

Verify your Install

To verify your install, run the command “docker run hello-world” and if you get an error message, you have not properly installed Docker.

Note: You will most likely need to restart your CLI to use the command after installation.

Docker Hub Account

Now that you have verified your installation, you must create a Docker Hub account or be logged in to Docker Hub. Think of Docker Hub as GitHub for Docker. To create a Docker Hub account, run “docker login” and follow the steps there.

Note: It is best to use a PAT (Personal Access Token)

Images

A Docker image describes the parameters of the container and how to create it. We don't always have to create our own images because there are a lot of them already available out there for us to use whenever we want! You can find a very large collection of them at the Docker Hub. Our VS Code extension handles the work of installing the Docker image and creating the container, so don’t worry about this. All you need to do is log in to Docker Hub and have Docker Desktop running in the background while the VS Code extension is running!