




DOCKER - AN OVERVIEW

- Salman Shah
- Mahim Agrawal



Things covered today

- What is Docker?
- Terms and Definitions
- How is Docker different from VMs?
- Docker - Complete Overview
- Dockerfile and Docker Compose
- Running your first Docker Container



Prerequisites

- IP Addresses and Ports
- Virtual Machines
- Editing configuration files
- Basic familiarity with the ideas of code dependencies and building
- Machine resource usage terms, like CPU percentages, RAM use in bytes, etc.



What is Docker?

It is a tool designed to make it easier to run applications on any machine regardless of the OS or any other configurations that might differ from the machine it was developed and tested on.



Terms and Definitions

- Containers + Images
- Services
- Swarms
- Stacks



Image and Container

An **image** is a lightweight, stand-alone, executable package that includes everything needed to run a piece of software, including the code, a runtime, libraries, environment variables, and config files.

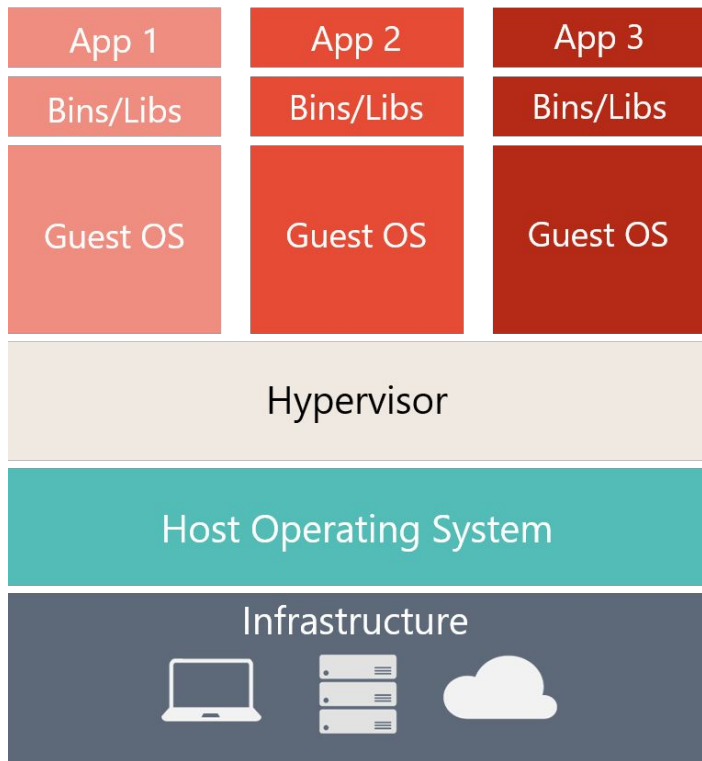
A **container** is a runtime instance of an image—what the image becomes in memory when actually executed. It runs completely isolated from the host environment by default, only accessing host files and ports if configured to do so.



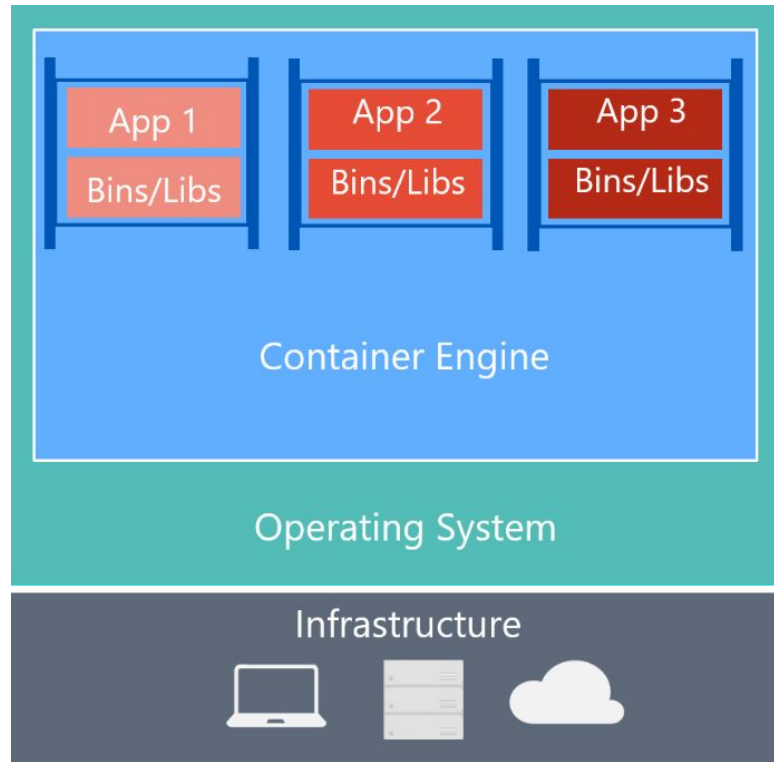
How is it different from VMs?

VMs emulate the hardware and create a whole virtual OS.

Docker uses the host OS kernel and needs only the executable and its dependencies.



Virtual Machine



Docker Container



Services

- In a distributed application, different pieces of the app are called “services.” For example, if you imagine a video sharing site, it probably includes a service for storing application data in a database, a service for video transcoding in the background after a user uploads something, a service for the front-end, and so on.
- Simple example could be have a basic Rails app which has a service for running Rails, another service for running Postgresql, a third service for running Nginx/Apache and so on.



Swarms

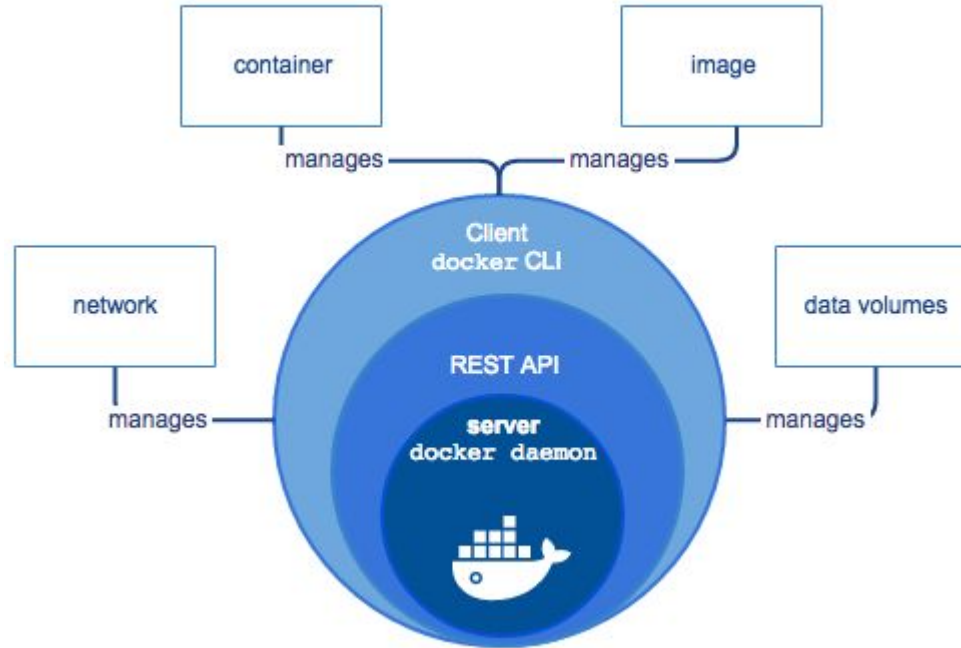
A swarm is a group of machines that are running Docker and joined into a cluster. After that has happened, you continue to run the Docker commands you're used to, but now they are executed on a cluster by a **swarm manager**. The machines in a swarm can be physical or virtual. After joining a swarm, they are referred to as **nodes**.



Stacks

A stack is a group of interrelated services that share dependencies, and can be orchestrated and scaled together. A single stack is capable of defining and coordinating the functionality of an entire application (though very complex applications may want to use multiple stacks).

Docker Overview



The CLI uses the Docker REST API to control or interact with the Docker daemon through scripting or direct CLI commands. Many other Docker applications use the underlying API and CLI.



Why Docker

Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. With Docker, you can manage your infrastructure in the same ways you manage your applications.

By taking advantage of Docker's methodologies for shipping, testing, and deploying code quickly, you can significantly reduce the delay between writing code and running it in production.



Writing your first Dockerfile

Check out the demo in case you are interested!



Docker Compose

Note the difference between docker-compose and docker stacks