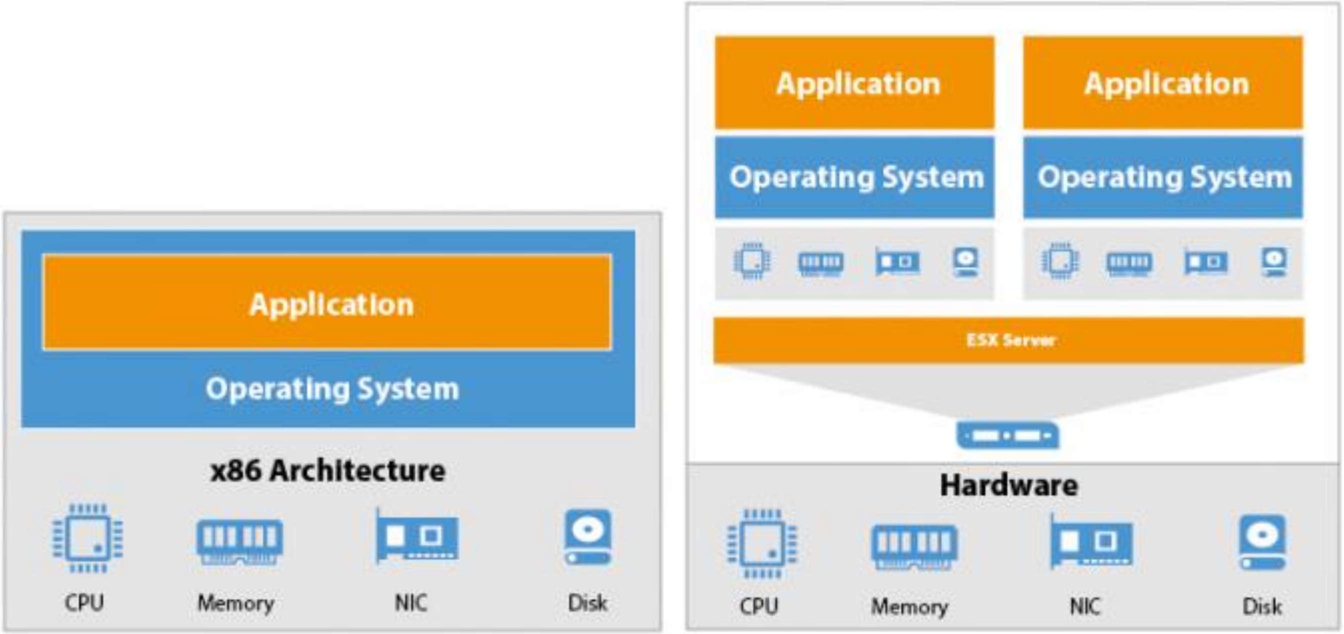
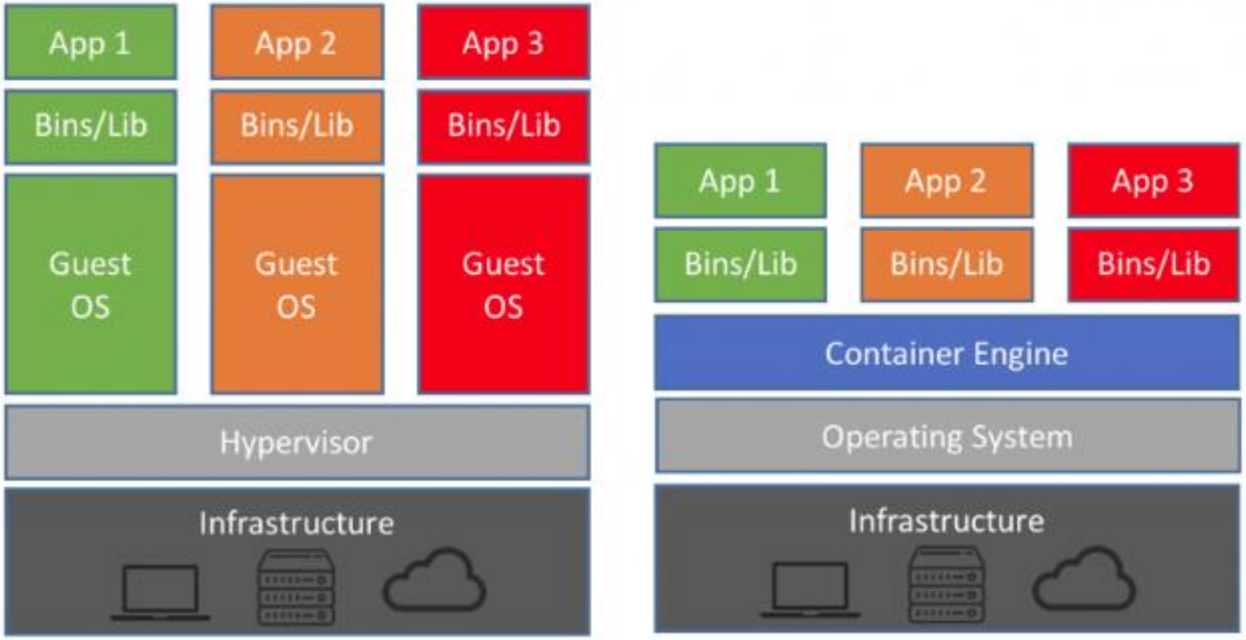
Docker – Getting Started

Traditional Vs Virtualization



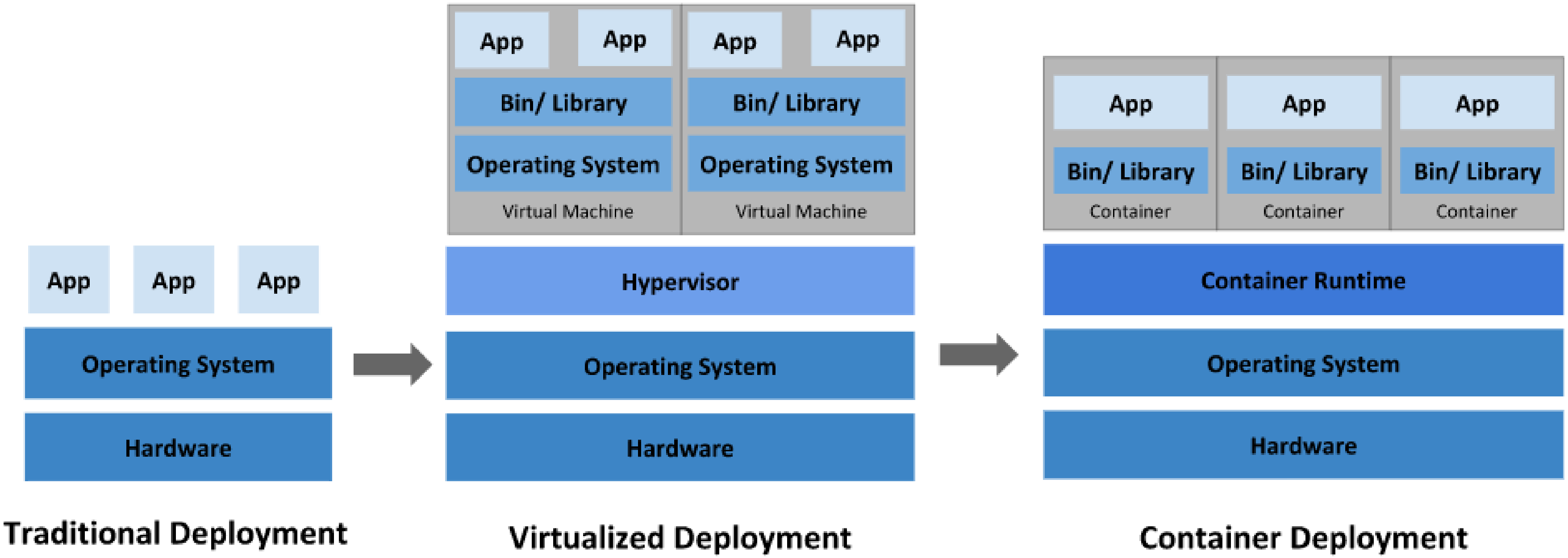
Virtualization Vs Container



Machine Virtualization

Containers

Traditional Vs Virtualization Vs Container



App App App

Operating System

Hardware

Traditional Deployment

App App App App

Bin/ Library Bin/ Library

Operating System Operating System

Virtual Machine Virtual Machine

Hypervisor

Operating System

Hardware

Virtualized Deployment

App App App

Bin/ Library Bin/ Library Bin/ Library

Container Container Container

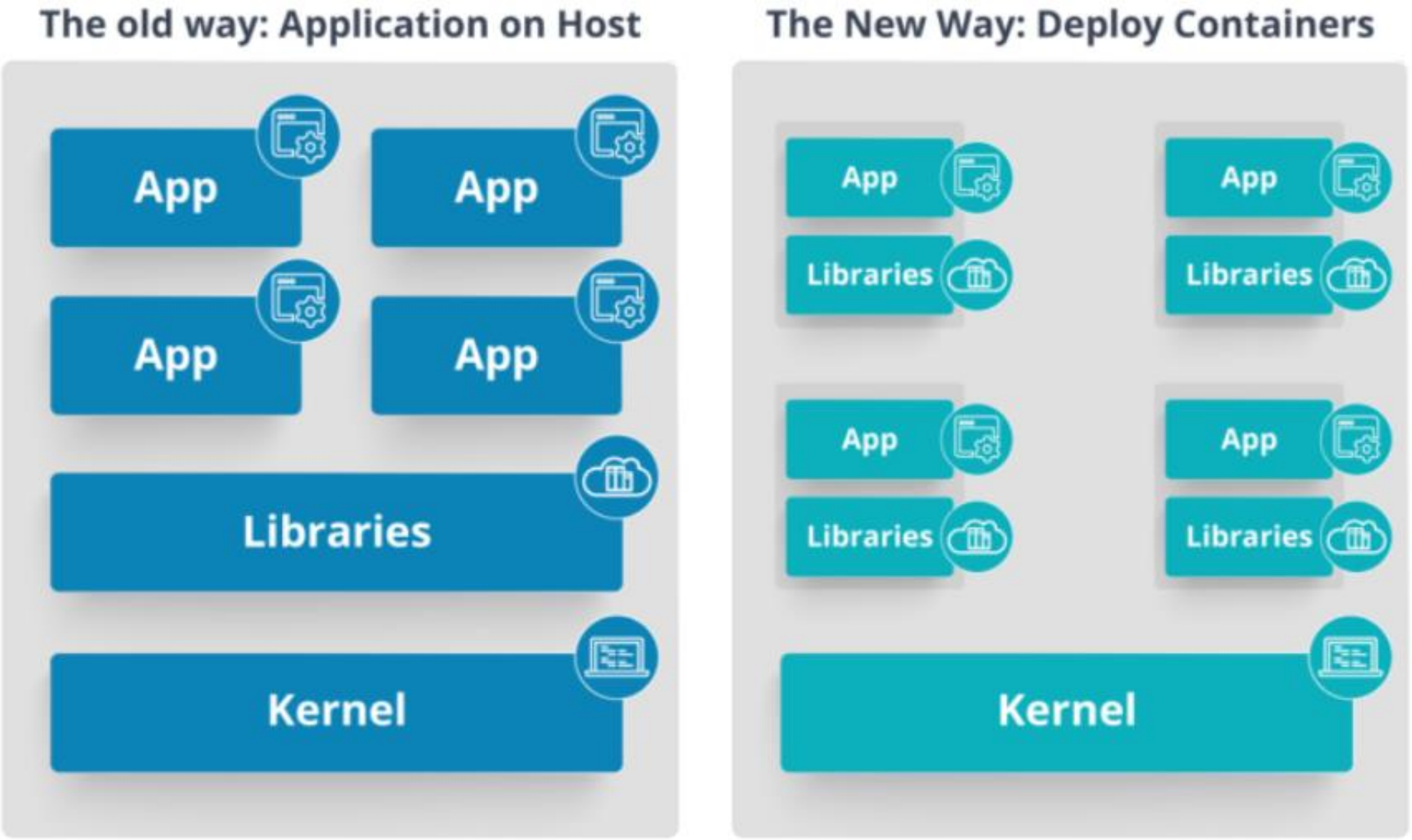
Container Runtime

Operating System

Hardware

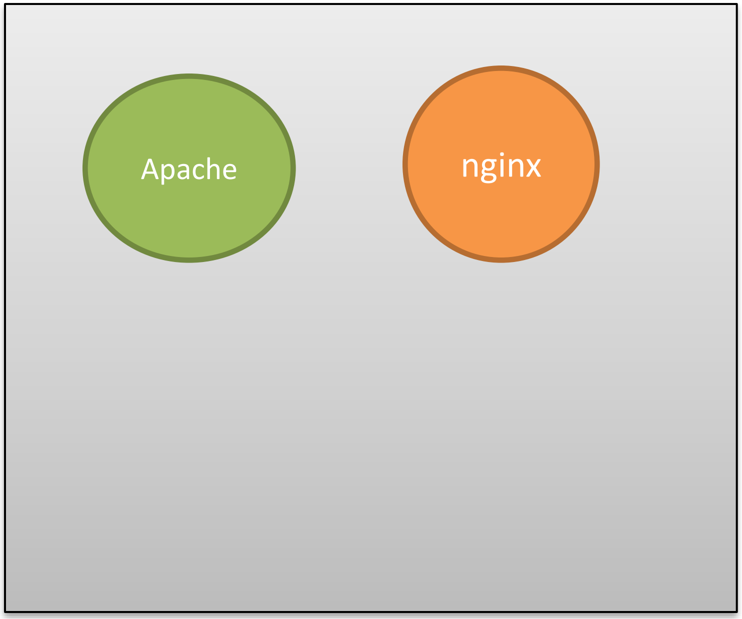
Container Deployment

App Deployments- Old Vs New Way



What is Container?

- A container is simply another process on your machine that has been isolated from all other processes on the host machine
- You can create, start, stop, move, or delete a container



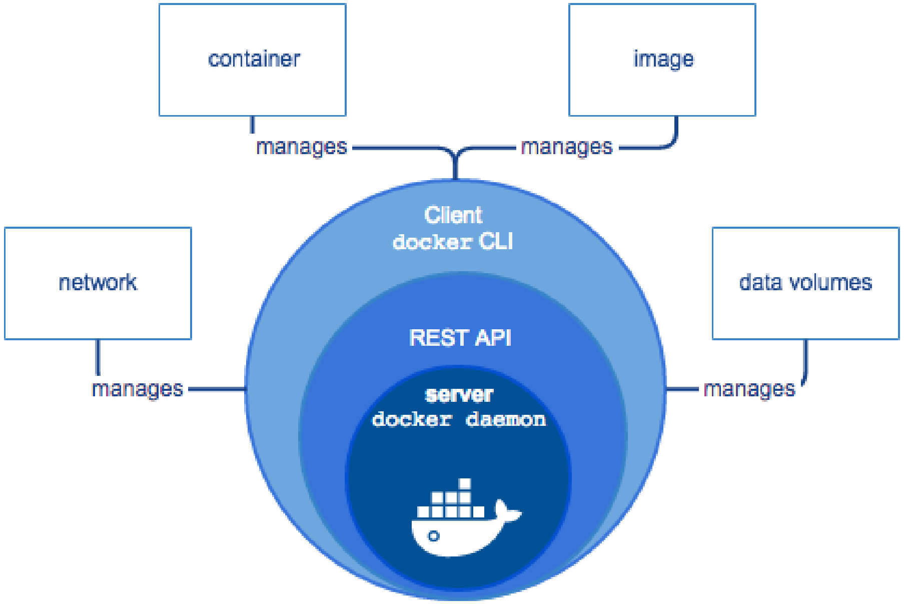
Host - Linux

Docker Introduction

- What is Docker?

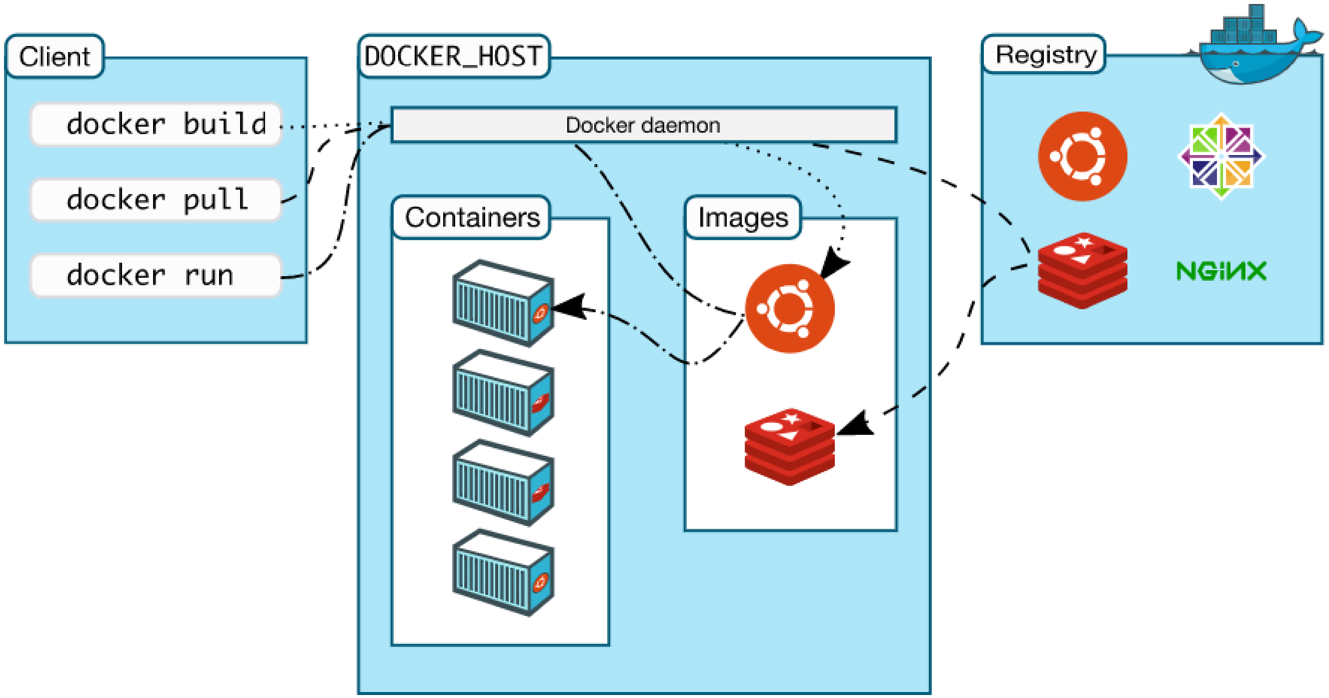
  - Open platform for developing, shipping, and running applications
  - Enables you to separate your applications from your infrastructure so you can deliver software quickly
  - Build once run anywhere
  - Provides the ability to package and run an application in a loosely isolated environment called a container
  - Portable
  - Docker didn’t invent container, It made easier to work with container

Docker – Core Components



Architecture

- **Client** – Way of interacting with Docker Daemon by using commands. Client talks to daemon.
- **Docker Daemon** – Its is called **dockerd** who manages images, containers, networks, volumes etc. Its core component.
- **Registry** – Stores Docker images
- Docker client, daemon, registry can be on different hosts



Let’s Install

- Available on multiple platforms like Linux, Windows and MAC
- https://docs.docker.com/get-docker/
- Preferred - **Linux**

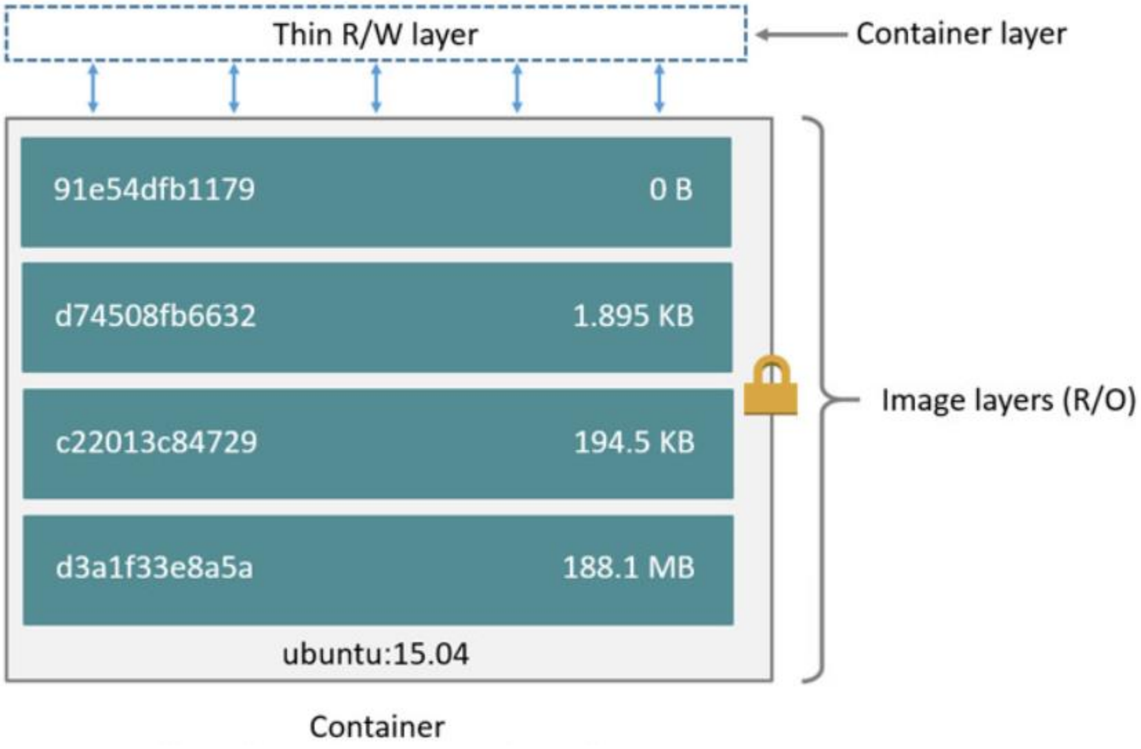
Docker Desktop for Windows

Docker for Linux

Docker Desktop for Mac

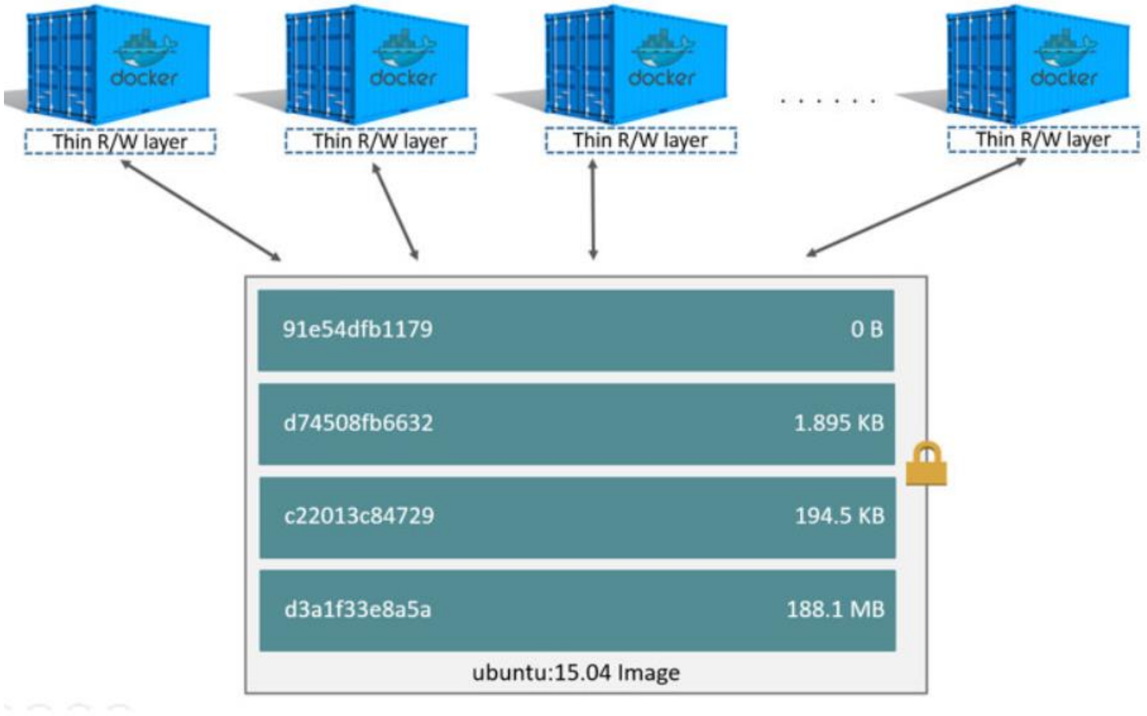
Container Image

- An image is a read-only template with instructions for creating a Docker container
- It must contain everything needed to run an application - all dependencies, configuration, scripts, binaries etc
- An image is based on another image, with some additional customization



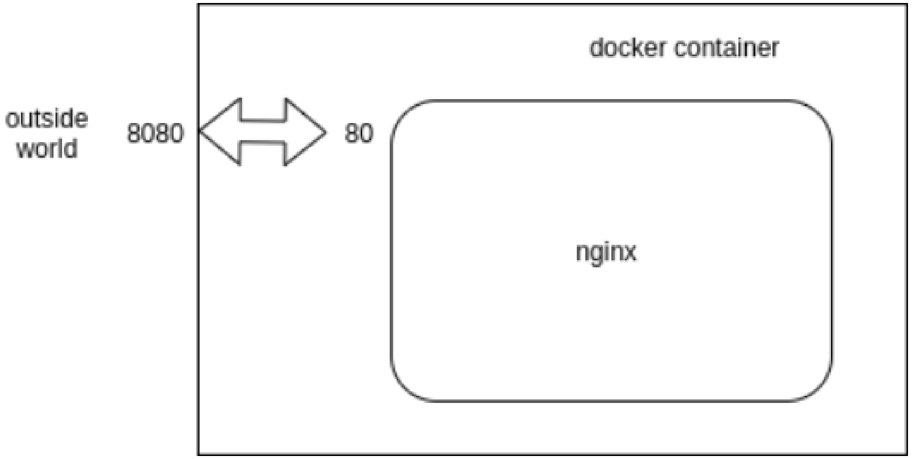
Container

- A runnable instance of an image
- A container is simply another process on your machine that has been isolated from all other processes on the host machine
- You can create, start, stop, move, or delete a container



Port Mapping/Binding

- To access the application running inside container
- Host port mapped to Container Application Port
- E.g. docker container run –dt <hostPort:containerPort> nginx
- In below diagram, docker container run –d –p 8080:80 nginx



Example of port binding: Bind port 80 of the Docker container to port 8080 of the host machine.

Container Login / Exec

- Exec – Used to run the command in running container
- Exec – Used to login to container
- E.g. docker container exec –it <container Name> ls
- docker container exec –it <container Name> bash