



INTRODUCTION TO DOCKER

Agenda



In this session, you will learn about:

- Overview of Compute Service
- Virtualization Vs Containerization
- What is Docker?
- Why Docker Containers?
- Docker Terminologies
- Docker Editions

Overview of Compute Service

- Bare Metal Infrastructure
- Virtualized Infrastructure
- Containerized Infrastructure

Bare-Metal Servers

- Bare-metal servers are '**physical**' servers. Which is a **single-tenant** physical server.

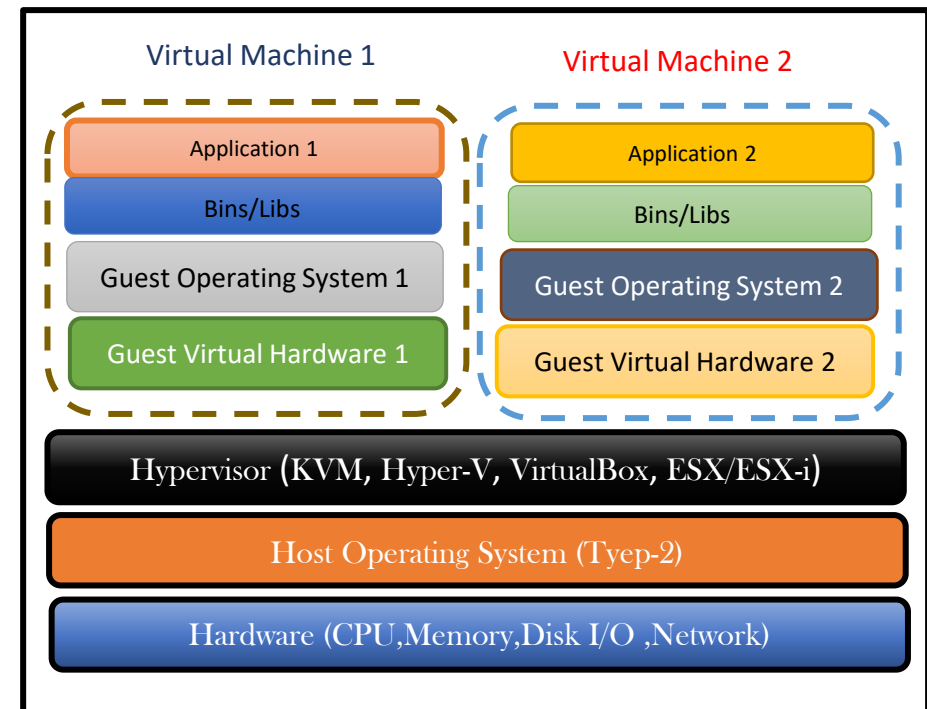


Disadvantages of Bare-Metal Servers

- One-App One-Server
- More expensive (hardware level)
- Mis-match of capacity
- Expensive maintenance

What is Virtualization?

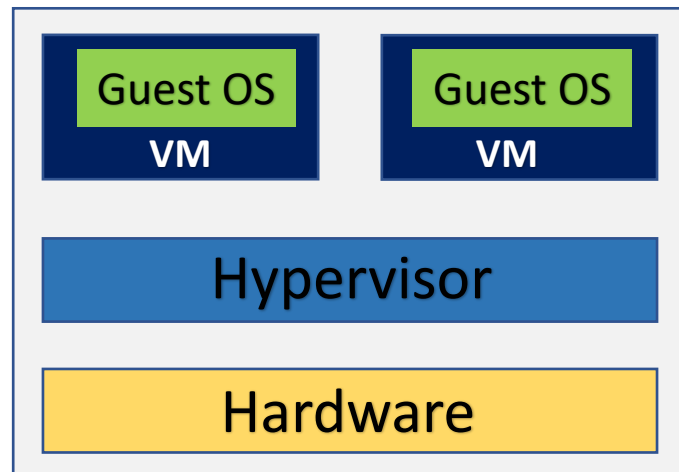
- Virtualization is the technique of **virtualizing** the underlying Infrastructure, such as Memory, CPU, Storage...
- **Guest** operating systems run on top of a **Host** operating system (Hypervisor).
- We can run different flavors of operating systems in different virtual machines all running on the same Infrastructure.
- Virtualization eliminates the need for extra hardware resource.



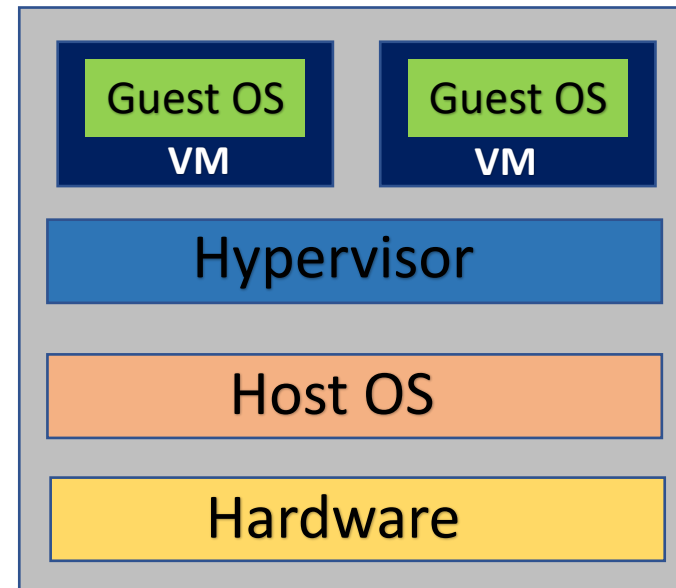
Hypervisors

- **Hypervisor:** A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as Memory, CPU and Storage.
- **Types of Hypervisor:**
 - Type-1 Hypervisors – Runs directly on top of Hardware.
 - KVM, Xen, Hyper-V, ESX/ESXi...
 - Type-2 Hypervisors – Runs on top of Host OS.
 - Oracle VB, VMware Workstation...

Hypervisors...



Type 1 Hypervisor
(Bare-Metal Architecture)



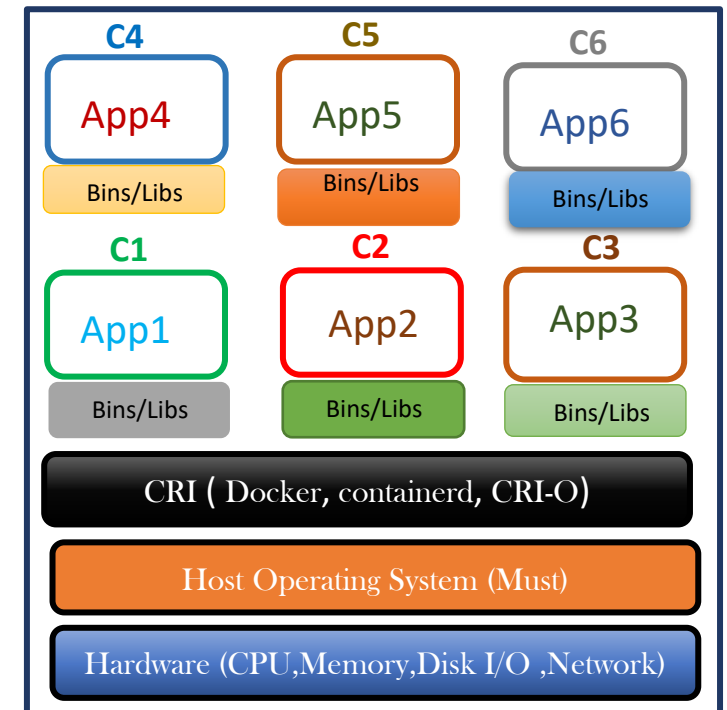
Type 2 Hypervisor
(Hosted Architecture)

VMs are great, but they're far from perfect!

- Each guest OS will have its own kernel and set of libraries and dependencies.
- Since each VM includes an OS and a virtual copy of all the hardware the OS requires, VMs require significant RAM and CPU resources.
- VMs incur a lot of overhead beyond what is being consumed by your application logic.
- Since each VM has its own dedicated OS, License cost is involved.
- Patching, Upgrades, Security, Hardening requires larger team and time.
- Boot up process is longer and takes more time.

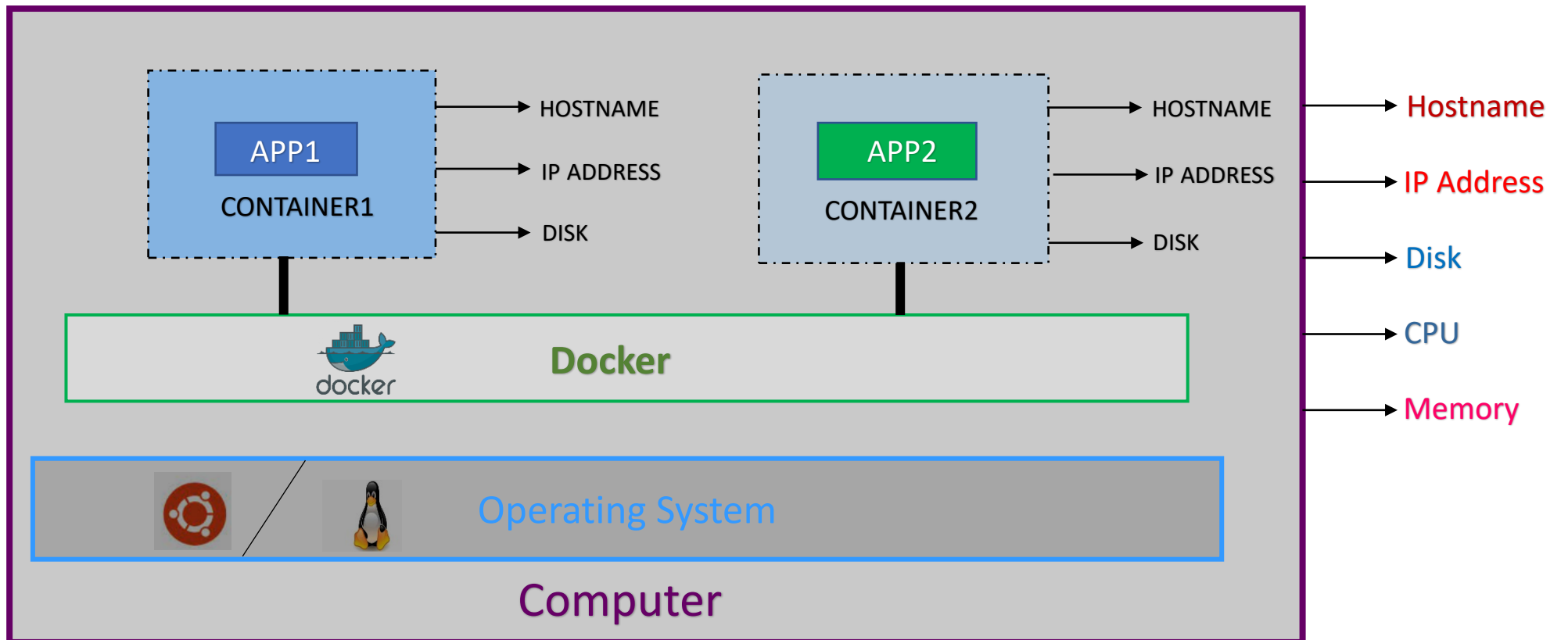
Containerization

- Containers are a method of **operating system level virtualization**.
- **Containers** allow you to run an application and its dependencies in resource-isolated processes.
- **No guest OS** overhead and utilizes a host's operating system.
- Containers share relevant **OS libraries & resources** as and when needed unlike virtual machines.
- Containers are **Lightweight** and **Faster** than Virtual Machines.



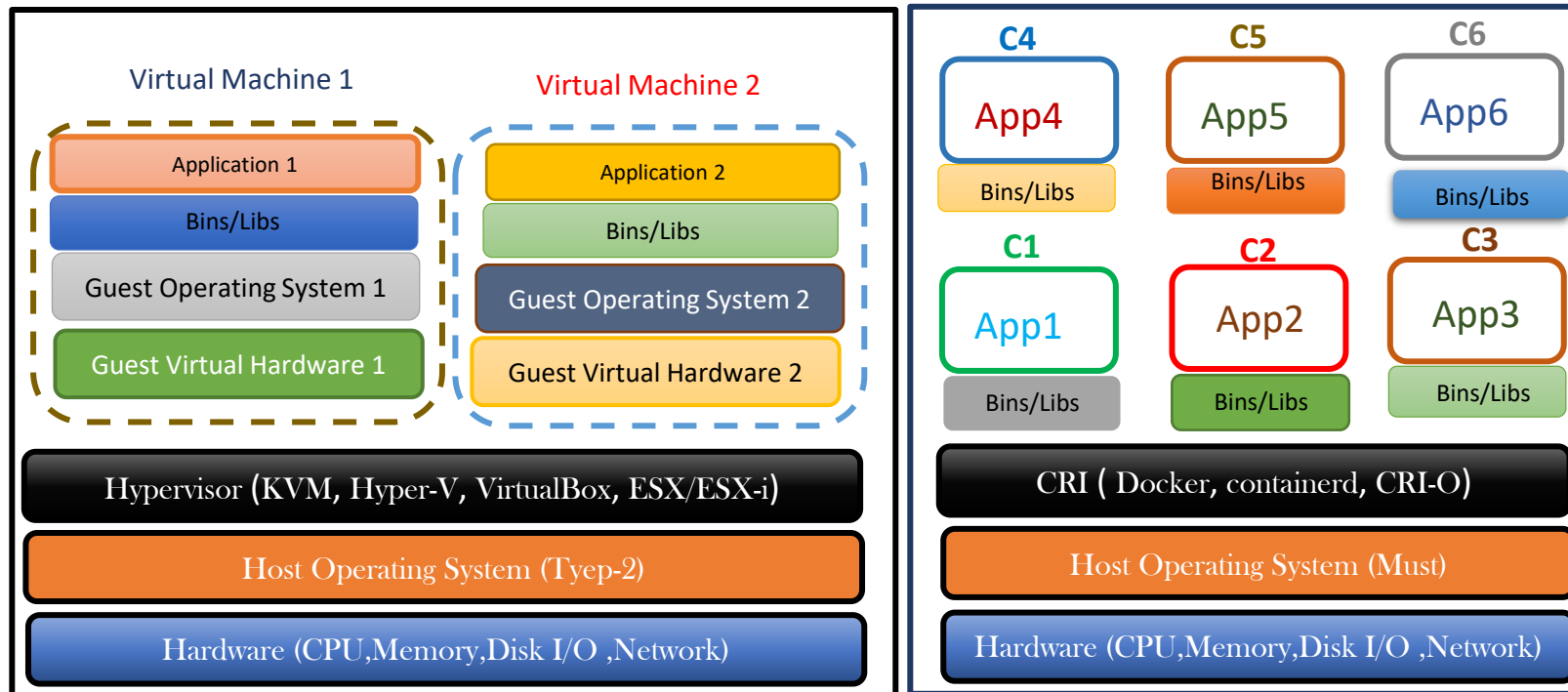
What is Container?

A container is the same idea as a physical container—think of it like a box with an application in it.



Each Container has its own Computer Name, IP address and Disk ...

Virtualization vs Containerization



Virtualization

- Method of Hardware level Virtualization
- Each VM needs dedicated Guest OS
- Larger in size
- Dedicated Kernel
- Each VM will have its own Libraries and Binaries
- Longer boot process
- Takes more time for creating
- Consumes more resources
- Migrating virtualized application is challenging due to hardware incompatibility

Containerization

- Method of OS level Virtualization
- Containers share host machine OS
- Smaller in size
- Share the Host kernel
- Share relevant Libraries and Binaries
- Shorter boot process
- Takes Few seconds
- Consume less resources
- Migrating Containerized application is much easier

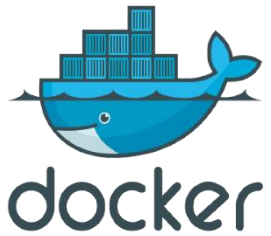
What is Docker?

- Docker, Inc. – The company
- Docker the technology
- Docker is an open source platform for **developing, shipping, and running** applications.
- Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.
- Docker manages the lifecycle of the container.
- The use of containers to deploy applications is called containerization.

History of Docker Inc.

- Developed using Linux core components, in **2013**.
- It was developed as an internal project at a **platform-as-a-service** company called **dotCloud** and later renamed as **Docker**.

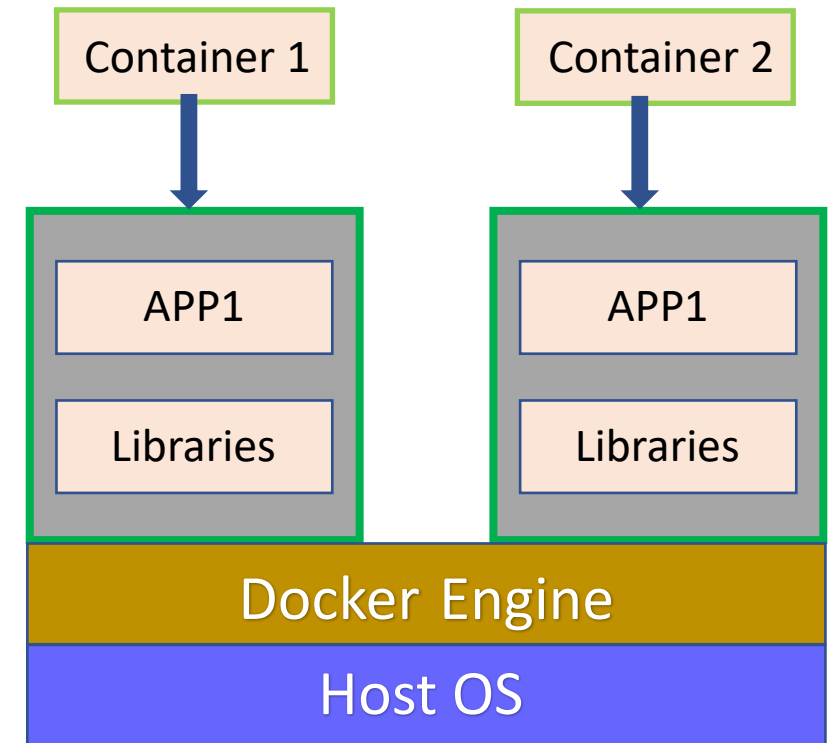
Docker Terminologies



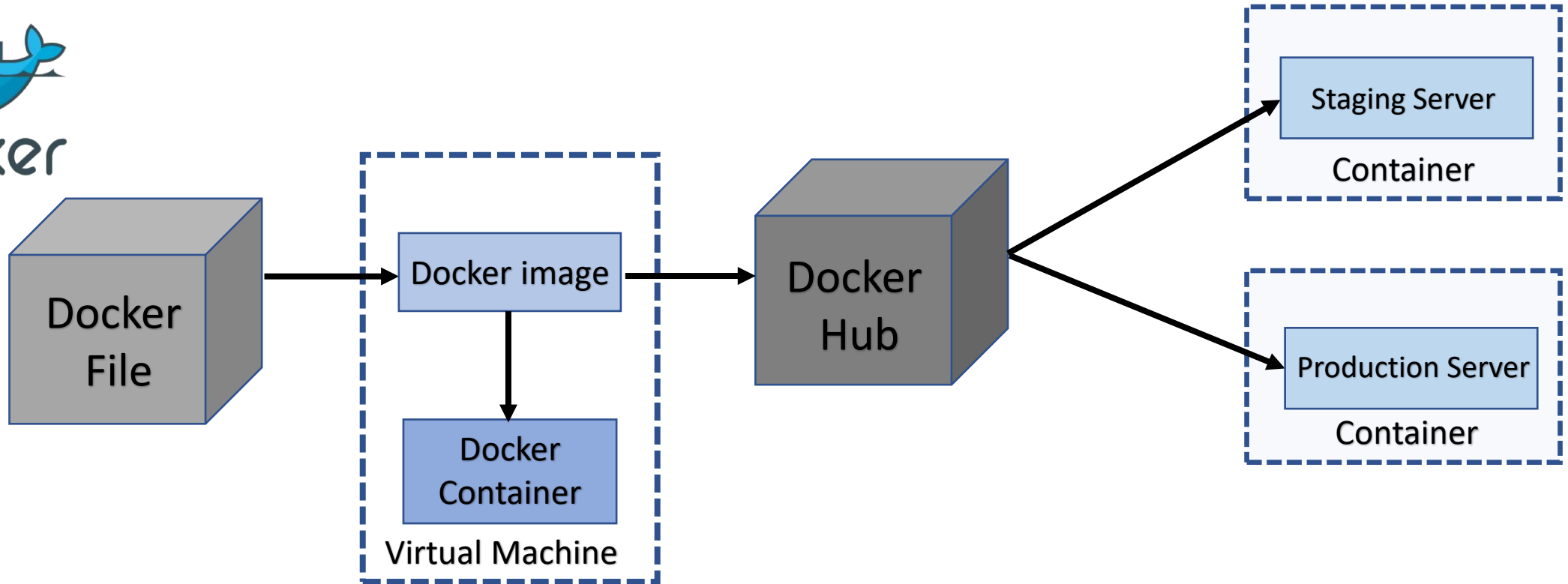
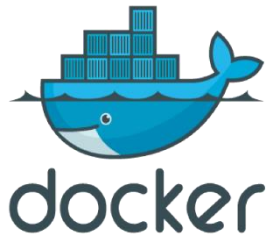
- Runs Application within docker Container
- Alternative to VMs & use host's OS

3 Terminologies to remember

- Dockerfile contain all the application dependencies
- Docker image is build using dockerfile
- Docker container is an instance of a docker image



How Docker Works?



Docker Editions

- **Docker Community Edition** (CE) is ideal for individual developers and small teams looking to get started with Docker and experimenting with container-based apps.
- **Docker Enterprise Edition** (EE) is designed for enterprise development and IT teams who build, ship, and run business critical applications in production at scale. Renamed as MKE – Nov 2019