Introduction

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
- Docker Internals
- Containerization for MicroServices

Overview of Compute Service

- Bare Metal Infrastructure
- Virtualized Infrastructure
- Containerized Infrastructure

Bare-Metal Servers

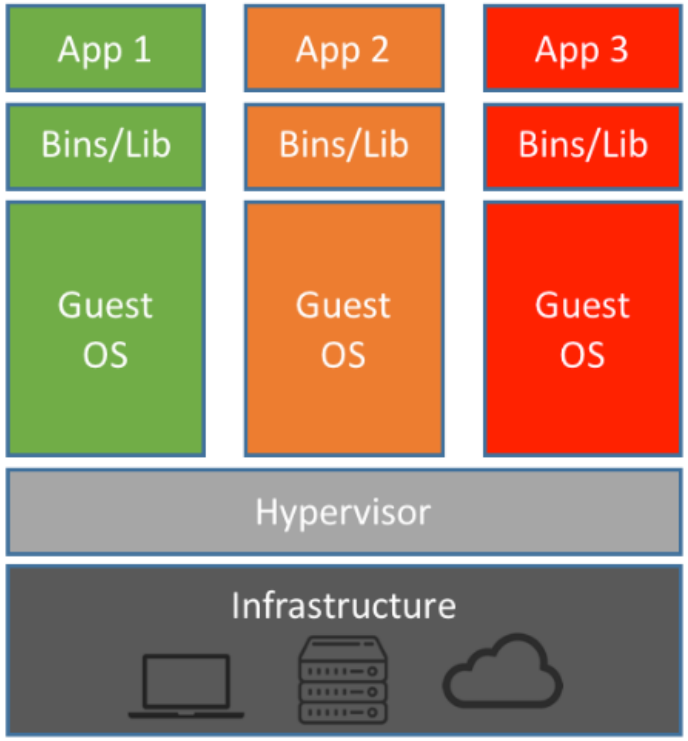
- Bare-metal servers are '**physical**' servers. Which is a **single-tenant** physical server.

Disadvantages of Bare-Metal Servers

- One-App One-Server
- More expensive
- 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.

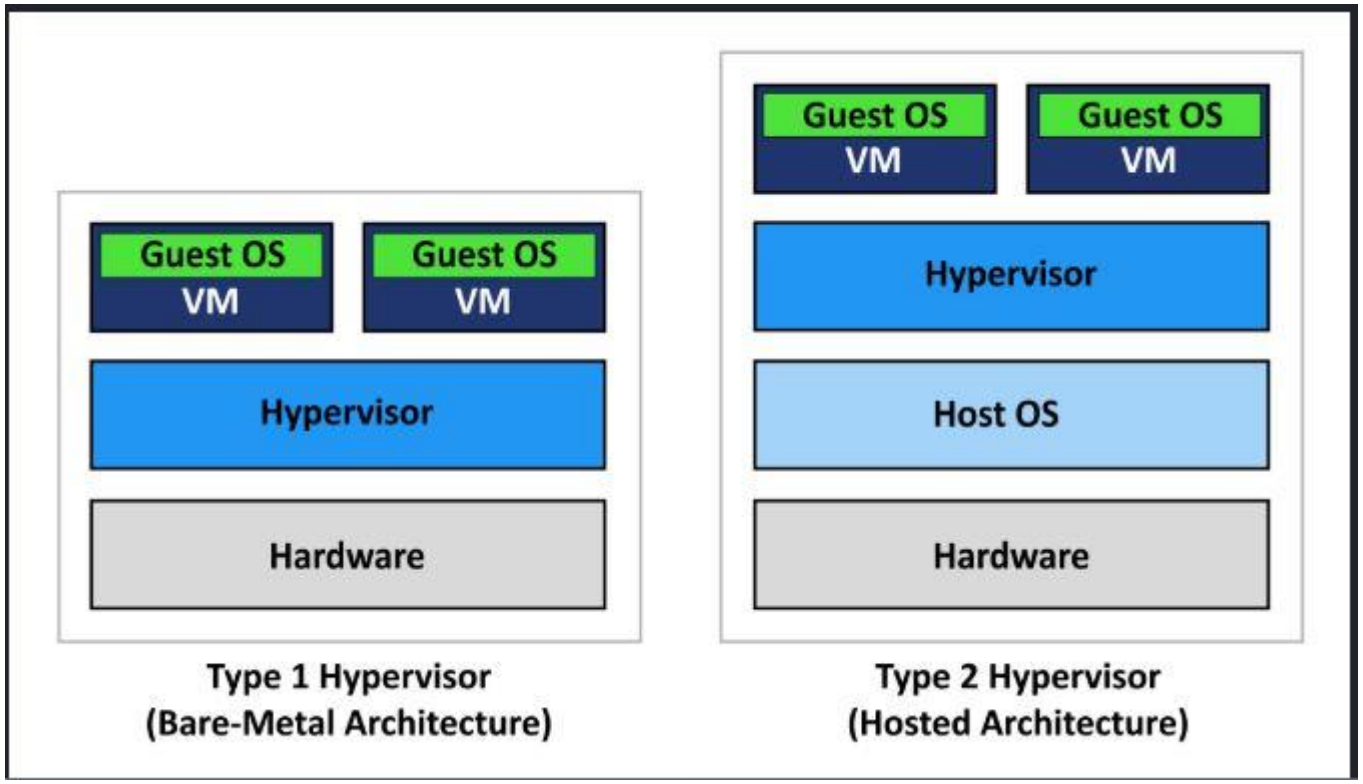


Machine Virtualization

Hypervisors

- **Hypervisor:** Also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs).
- A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.
- **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…

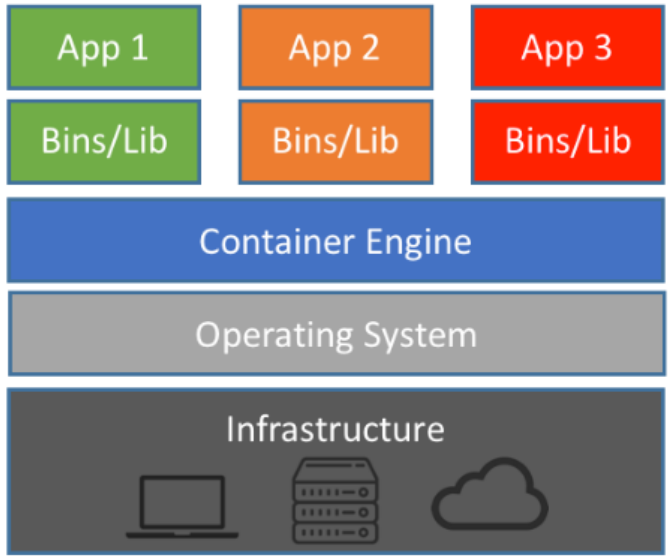


Disadvantages of Virtualization

- 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.

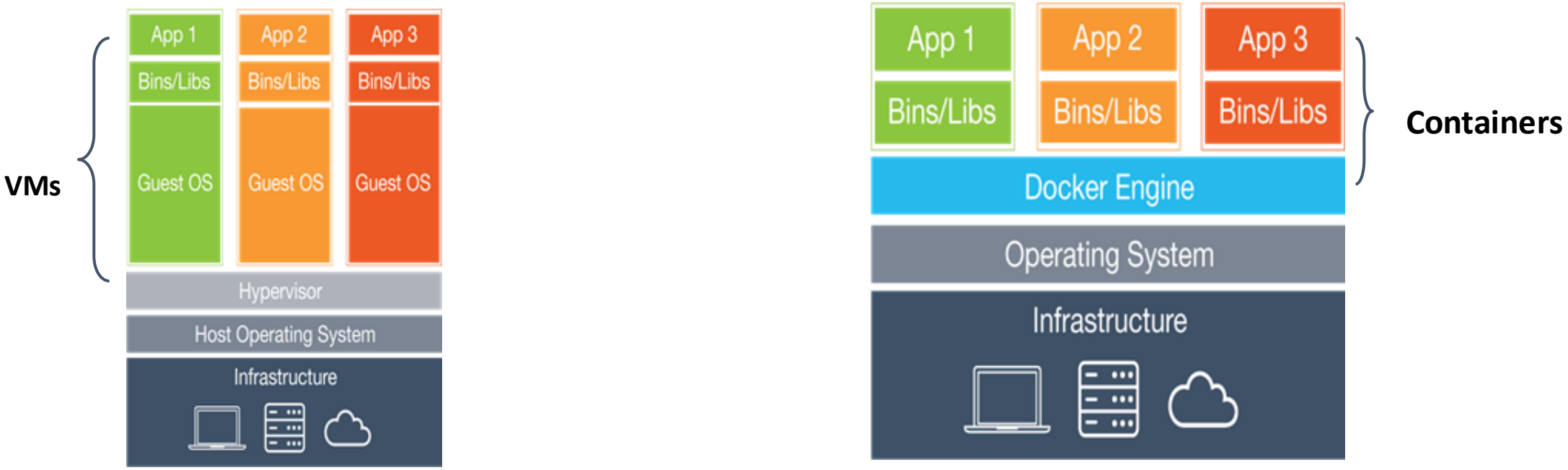
Containerization

- Containers are a method of **operating system 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 **libraries** & **resources** as and when needed unlike virtual machines.
- Containers are Lightweight and Faster than Virtual Machines.
- Containers can also run on top of VMs.



Containers

Virtualization vs Containerization



Virtualization	Containerization
• Method of Hardware level Virtualization	• Method of OS level Virtualization
• Each VM needs dedicated OS	• Containers share container image
• Larger in size	• Smaller in size
• Dedicated Kernel	• Share the Host kernel
• Each VM will have its own Libraries and Binaries	• Share relevant Libraries and Binaries
• Longer boot process	• Shorter boot process
• Takes more time for creating	• Takes seconds
• Consumes more resources	• Consume less resources
• Migrating virtualized application is challenging due to hardware incompatibility	• Migrating Containerized application is much easier
• Takes more time of Developer to setup Environment	• Increases the Developer productivity

What is Docker?

- 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

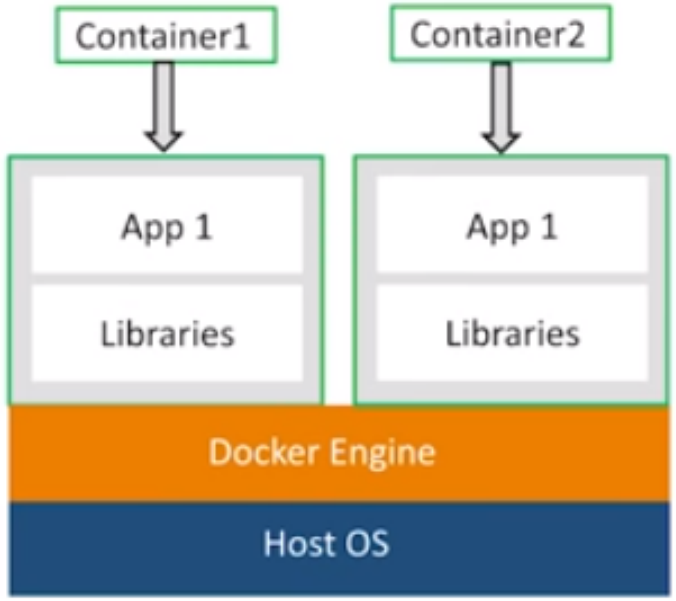
- Developed using Linux core components, in **2013.**
- It was developed as an internal project at a **platform-as-a-service** company called **dotCloud** and later renamed as **Docker.**

Docker Terminologies

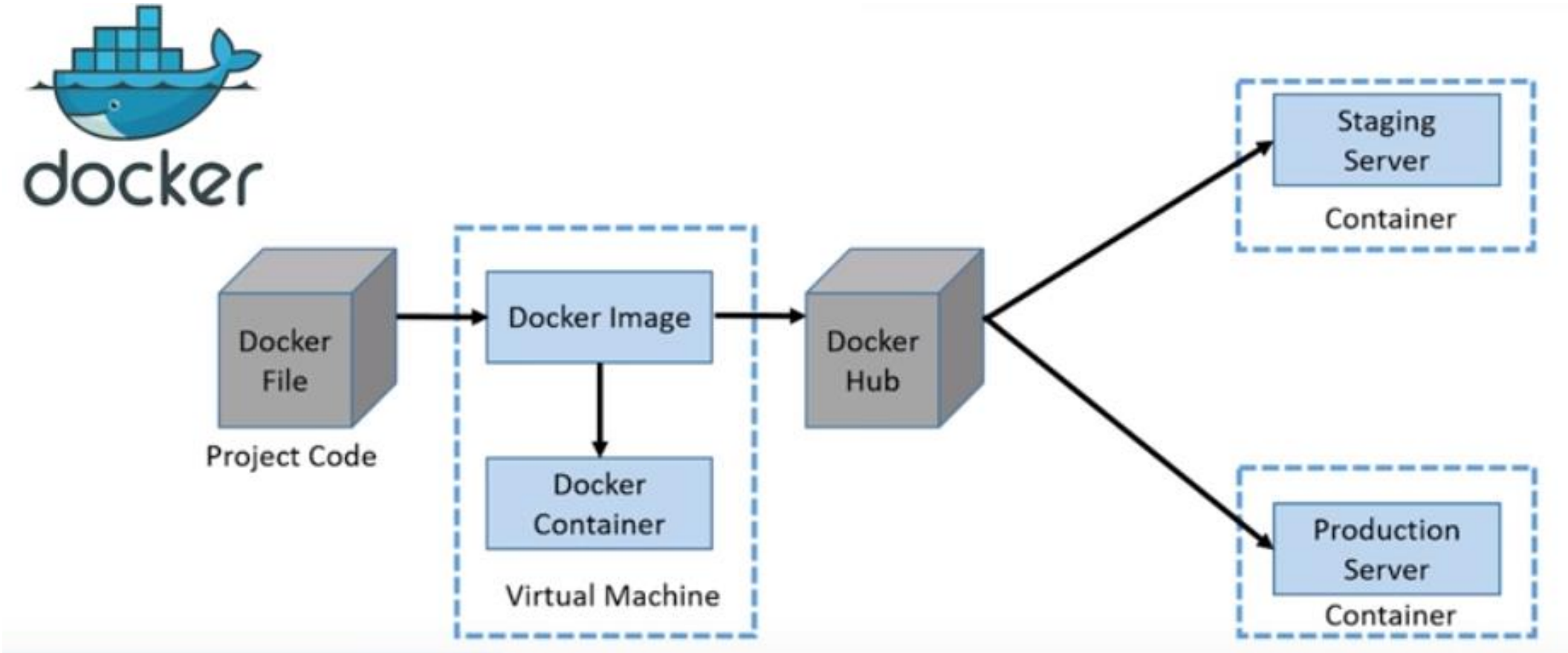
- Runs applications within Docker containers
- Alternative to VMs & use host's OS

3 terminologies to remember

- Docker Image is built using a Dockerfile
- Dockerfile contains all the application dependencies
- Docker container is an instance of a docker image



How Docker works?



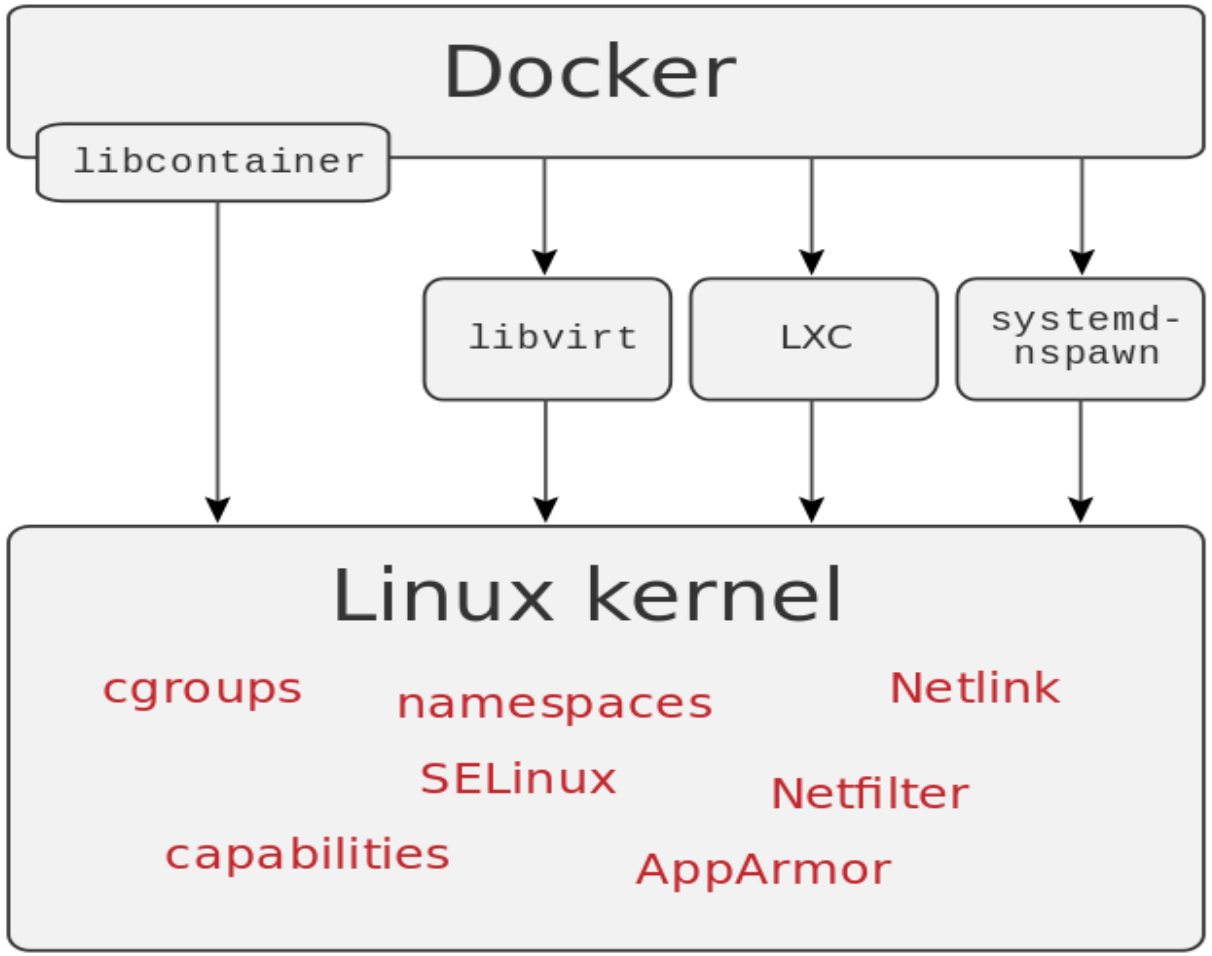
Docker Editions

- **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.

Time-based Release Schedule

- Starting with Docker 18.03, Docker uses a time-based release schedule.
  - **Docker CE Edge** - Monthly.
  - **Docker CE Stable** - Quarterly, with patch releases as needed.
  - **Docker EE** - Twice a year, with patch releases as needed.

Understanding the Docker Internals



Drawbacks of Docker

- Managing a large number of containers is challenging – especially when it comes to clustering containers.
- Solutions:
  - Docker SWARM
  - Kubernetes
  - RANCHER
  - OpenShift