Amirkabir University of Technology
(Tehran Polytechnic)

Operating System-Level Virtualization

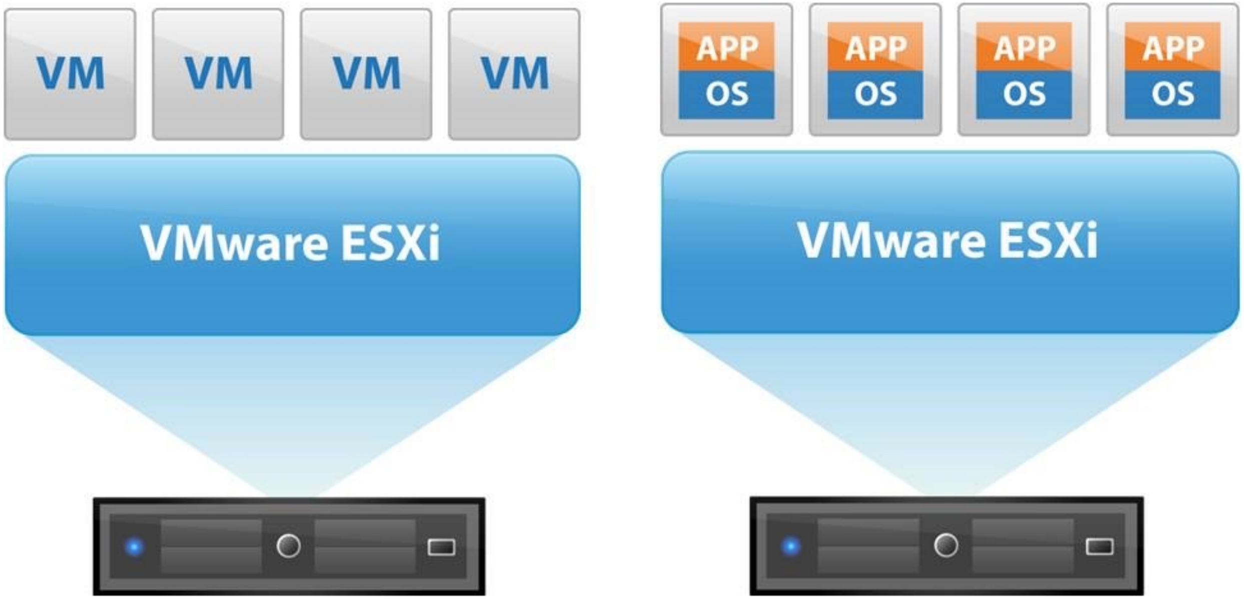
Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Spring 2023

Understanding Full Virtualization, Paravirtualization, and Hardware Assist

- You should read VMWare white paper
  - There will be exam questions from the paper (up to page 10)



A quick review

Virtualization approach	Method	Performance (boot time, runtime, ..)	Isolation
Hardware-level	Binary translation		
	Paravirtualization		
	Hardware-assisted		
OS-level	Containers		

Operating System-level Virtualization

- Creating **different** and **separated** execution environments for applications that are managed concurrently.
- OS kernel allows for multiple isolated user space instances.

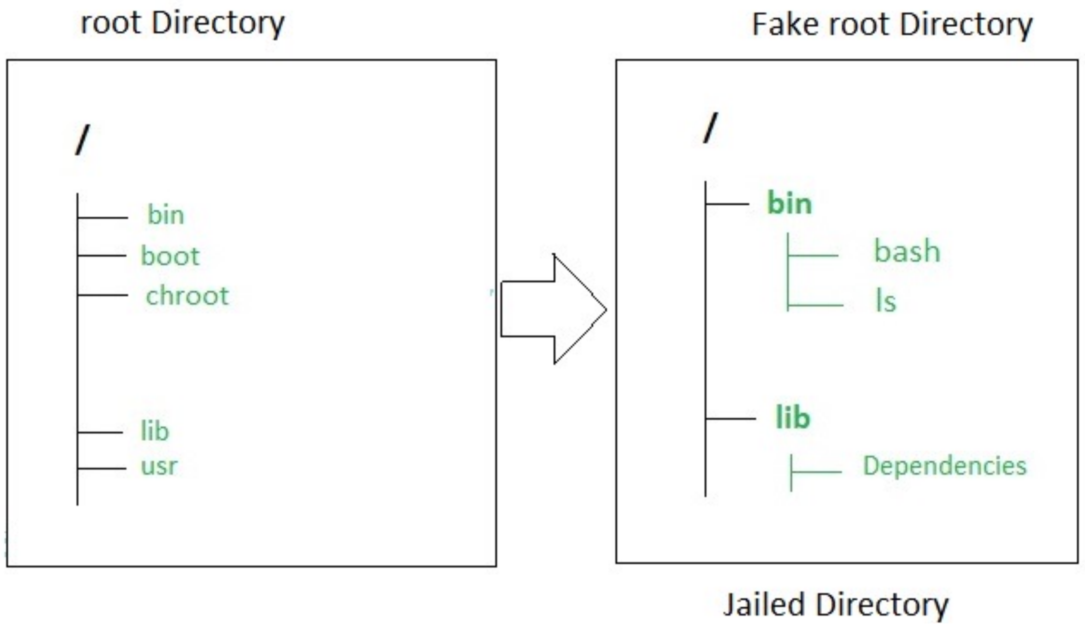
	Is there hypervisor?	How many OSs are involved?
Hardware-level	YES	Multiple OSs
OS-level	NO	Single OS

Operating System-level Virtualization (cont.)

- The kernel is also responsible ***for***
  - ***Sharing the system resources among instances***
  - ***Limiting the impact of instances on each other***.
- A ***user space instance*** contains a proper view of
  - the file system, which is completely isolated
  - separate IP addresses
  - software configurations
  - access to devices.

Operating System-level Virtualization (cont.)

- An evolution of the ***chroot mechanism in Unix systems***.
  - Changes the file system root directory for a process and its children.
  - The process and its children ***cannot have access to other portions of the file system than those accessible under the new root directory***.



https://www.geeksforgeeks.org/chroot-command-in-linux-with-examples/

Operating System-level Virtualization (cont.)

- Unix systems ***expose devices as parts of the file system***
  - Using ***chroot*** it is possible to completely ***isolate a set of processes***
- Following the same principle, operating system-level virtualization aims to ***provide separated and multiple execution containers*** for running applications.

Operating System-level Virtualization (cont.)

- An ***efficient solution*** for server consolidation scenarios in which ***multiple application servers share the same technology***:
  - Operating system
  - Application server framework
  - Other components.

Operating System-level Virtualization (cont.)

- When different application servers ***are aggregated*** into ***one physical server***, each server is run in a *different user space*, completely ***isolated*** from the others.
- Examples of operating system-level virtualizations are:
  - FreeBSD Jails
  - IBM Logical Partition (LPAR)
  - SolarisZones
  - Containers and Docker.

Containers

- OS-level virtualization also called containerization.

- A ***container is an isolated virtual env.*** which can run an application.

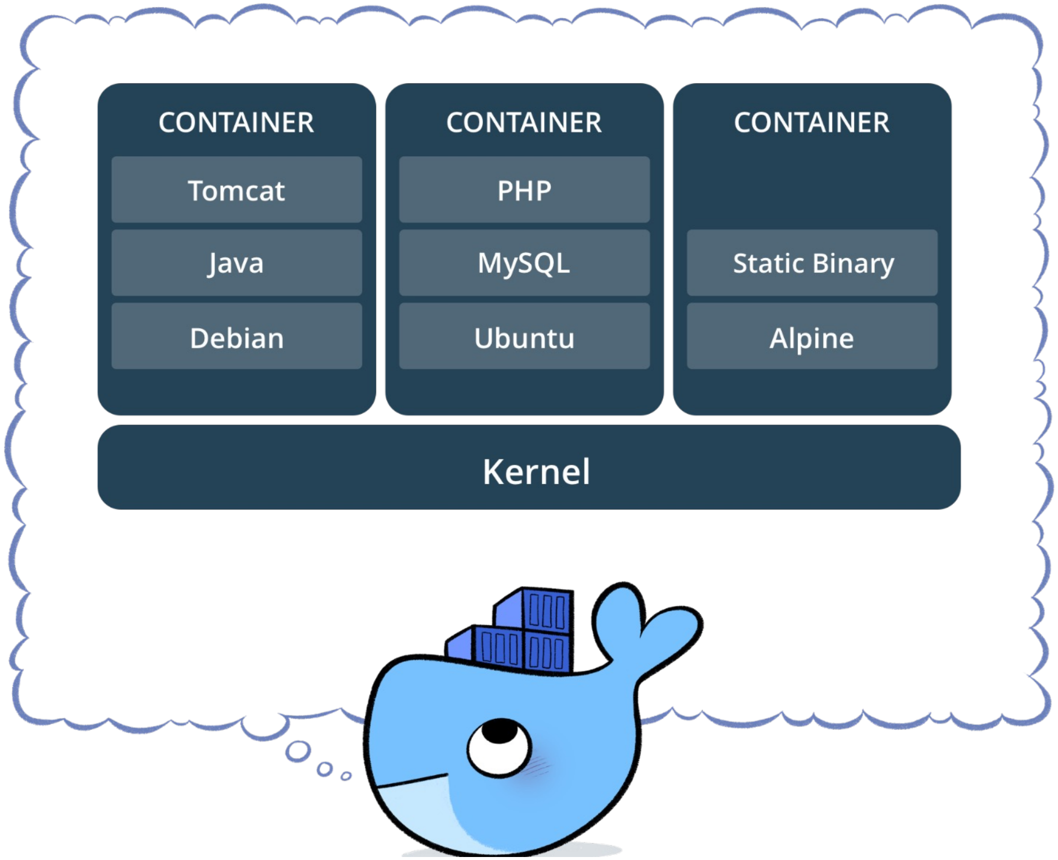
- Several containers can be created on each operating system, to each of which a ***subset of the computer's resources*** is allocated.

- Programs running inside a container **can only see the container's contents** and devices assigned to the container.

Docker

- Docker is the company driving the container movement .
- A container image is
  - A lightweight, stand-alone, executable package of a piece of software
  - It includes everything needed to run it: code, runtime, system tools, system libraries, settings.
- Available for both Linux and Windows based apps, containerized software ***will always run the same, regardless of the environment***.

Docker



CONTAINER

Tomcat

Java

Debian

CONTAINER

PHP

MySQL

Ubuntu

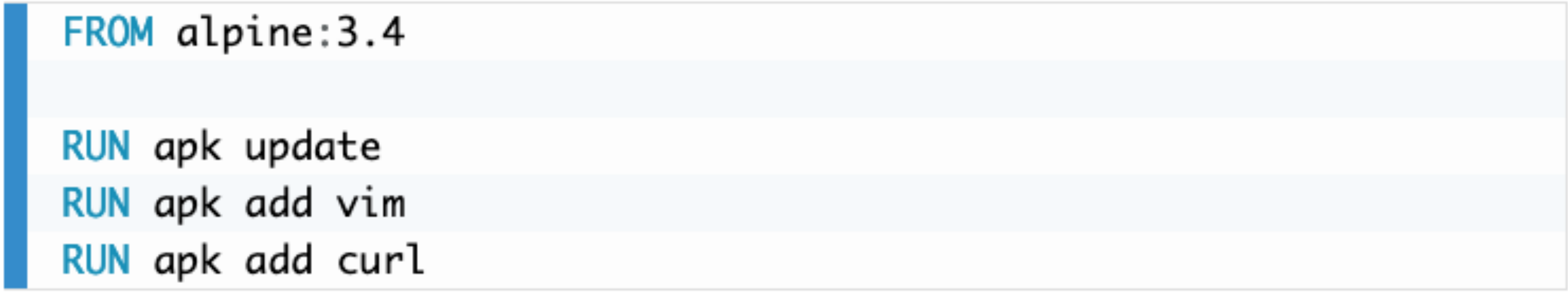
CONTAINER

Static Binary

Alpine

Kernel

Dockerfile example

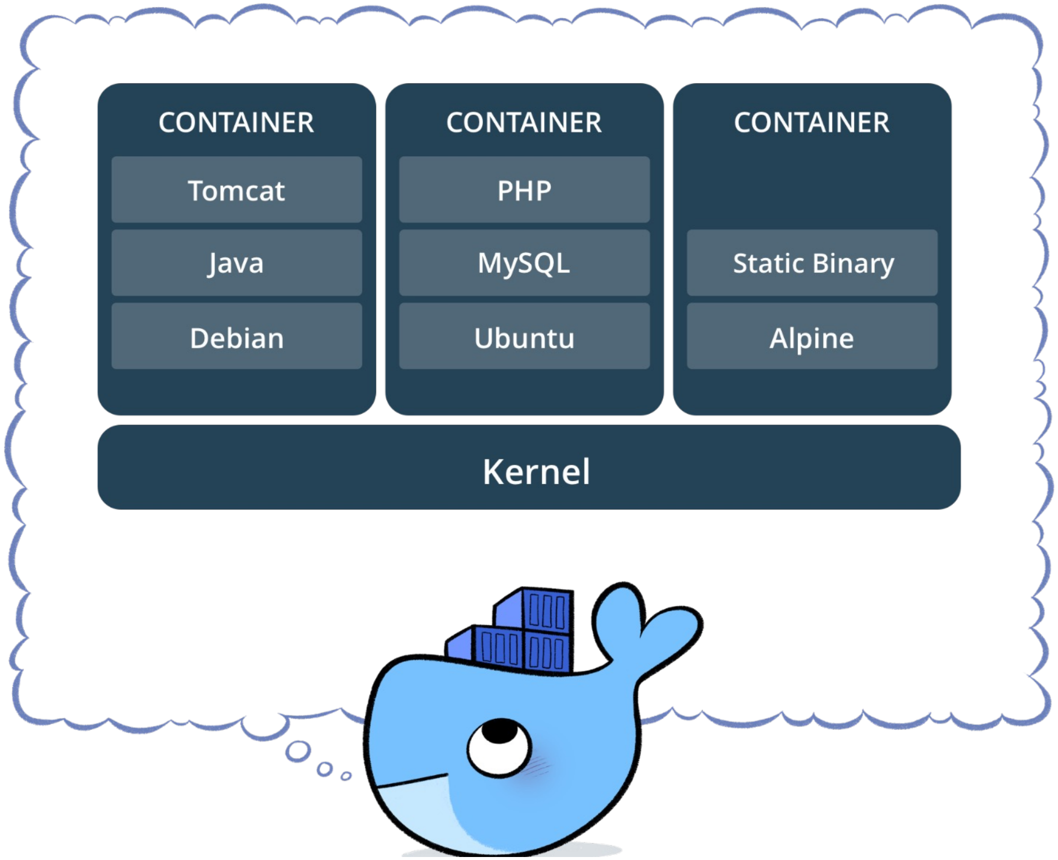


```
FROM alpine:3.4

RUN apk update
RUN apk add vim
RUN apk add curl
```

https://takacsmark.com/dockerfile-tutorial-by-example-dockerfile-best-practices-2018/

Docker



CONTAINER

Tomcat

Java

Debian

CONTAINER

PHP

MySQL

Ubuntu

CONTAINER

Static Binary

Alpine

Kernel

What about running Linux docker image in Windows?

Windows Subsystem for Linux (WSL)

- “A full Linux kernel built by Microsoft, allowing Linux distributions to run without having to manage Virtual Machines.”

- “With Docker Desktop running on WSL 2, users can leverage Linux workspaces and avoid having to maintain both Linux and Windows build scripts.”

- “In addition, WSL 2 provides improvements to file system sharing, boot time, and allows access to some cool new features for Docker Desktop users.”

https://docs.docker.com/desktop/windows/wsl/

Containers vs. Virtual Machines

- Virtual machines (VMs) are an abstraction of physical hardware turning one server into many servers.

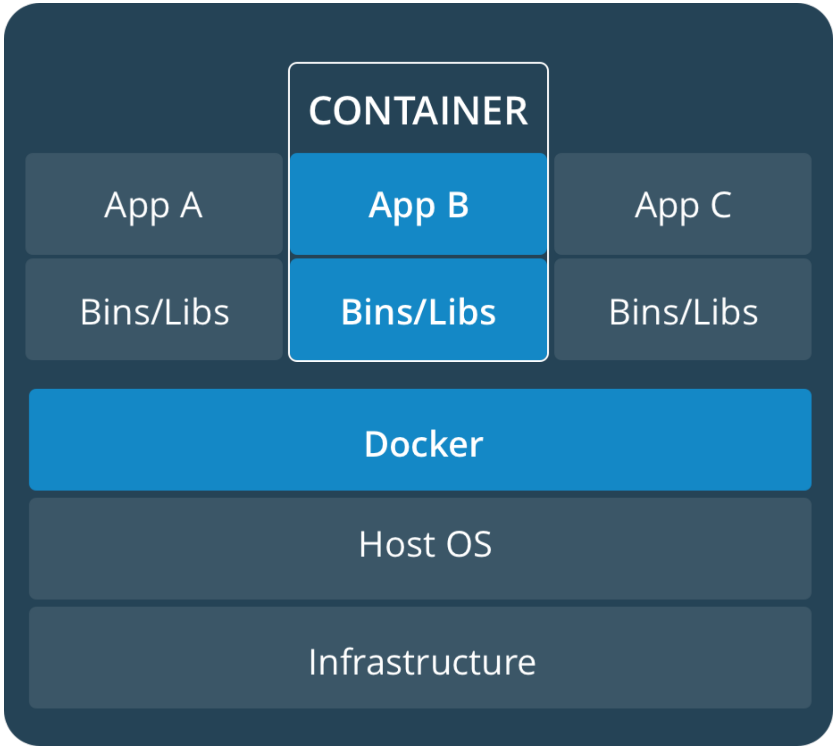
- A VM includes a full copy of an operating system, one or more apps, necessary binaries & libraries-taking up tens of GBs.

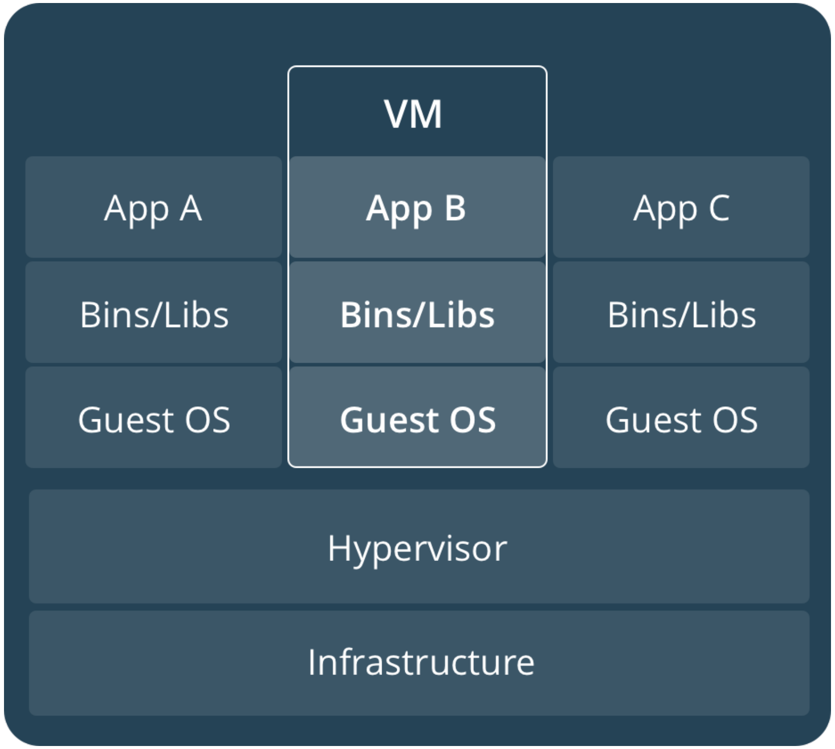
- VMs can also be ***slow to boot***.

Containers vs. Virtual Machines (Cont.)

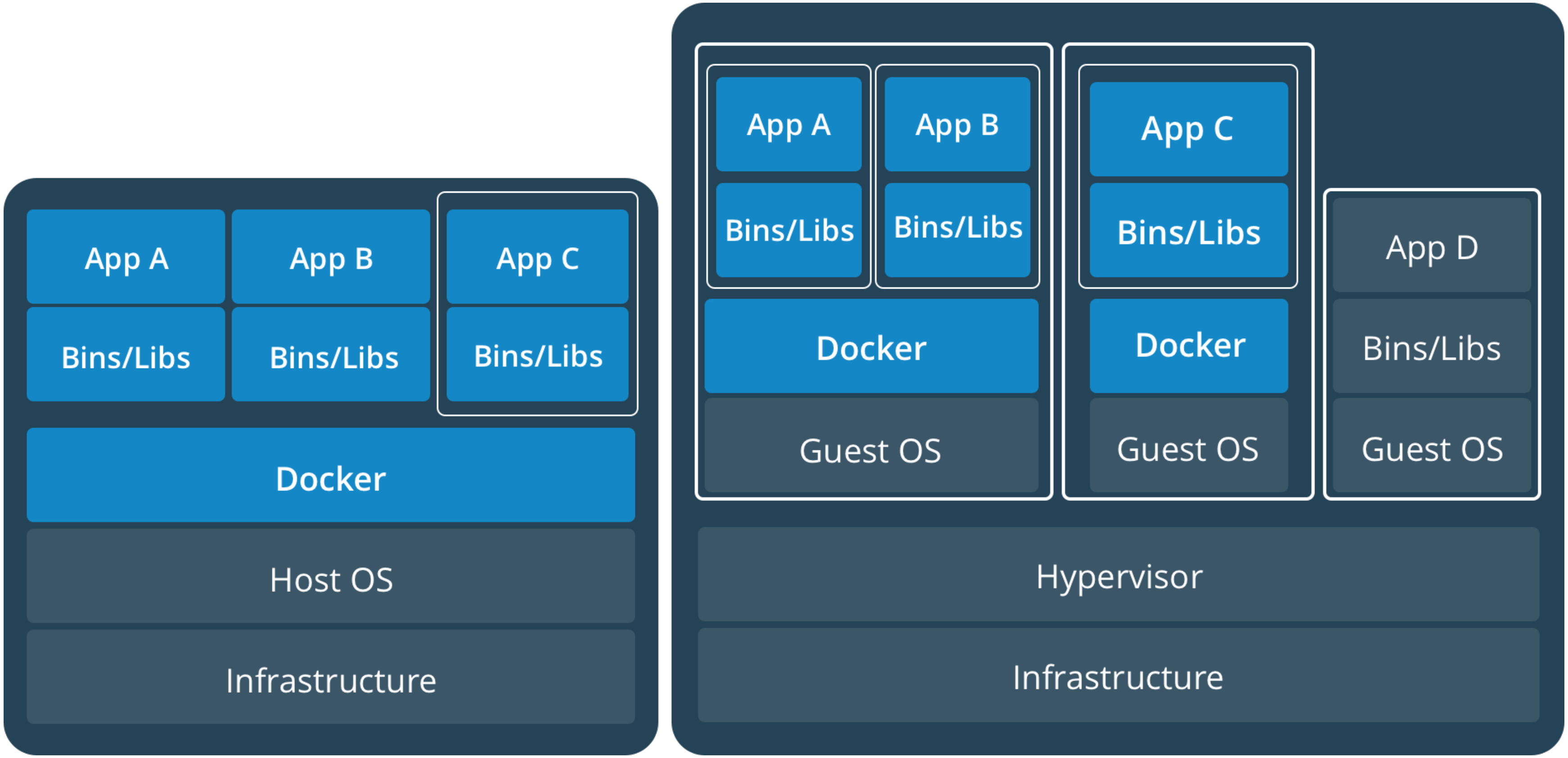
- Containers are ***an abstraction at the app layer*** that packages code and dependencies together.
- Multiple containers can run on the same machine **and share the OS kernel with other containers.**
  - each running as isolated processes in user space.
- Containers take up less space than VMs
  - Container images are typically tens of MBs in size
  - ***Start almost instantly***.

Containers vs. Virtual Machines (Cont.)

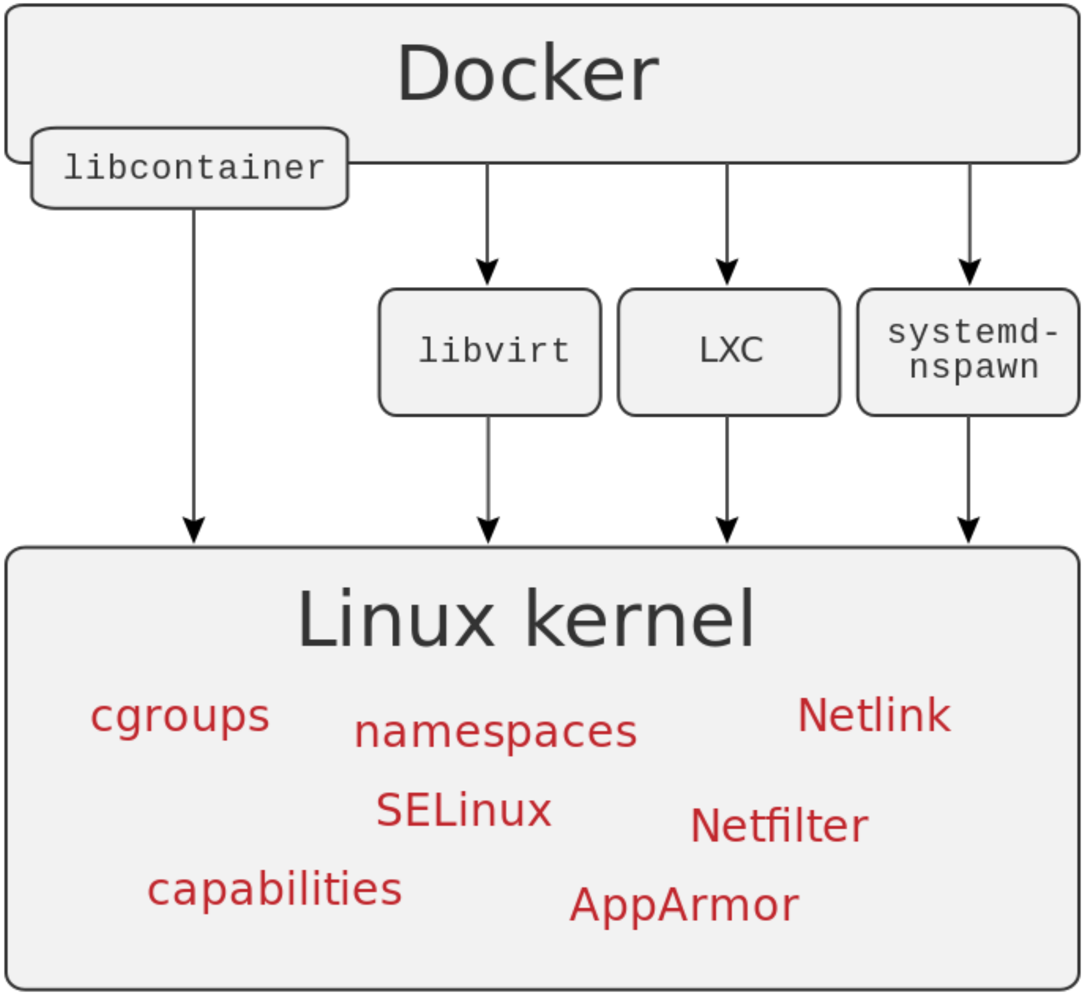




Containers vs. Virtual Machines (Cont.)

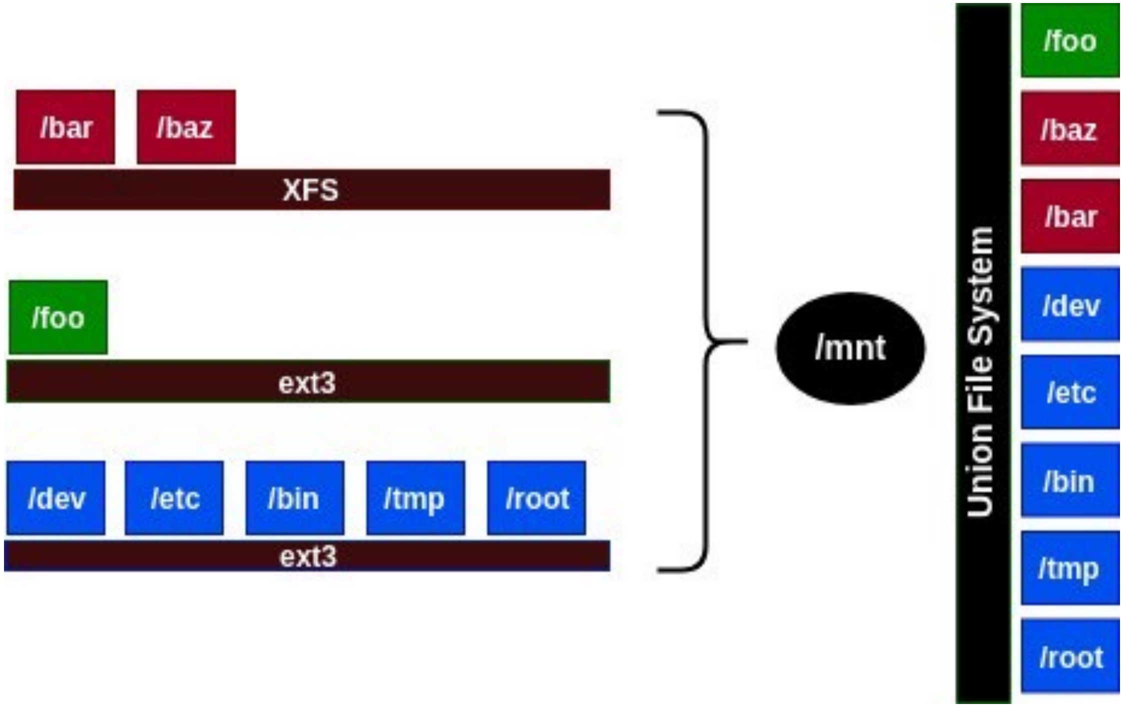


Docker Technology



Docker Technology

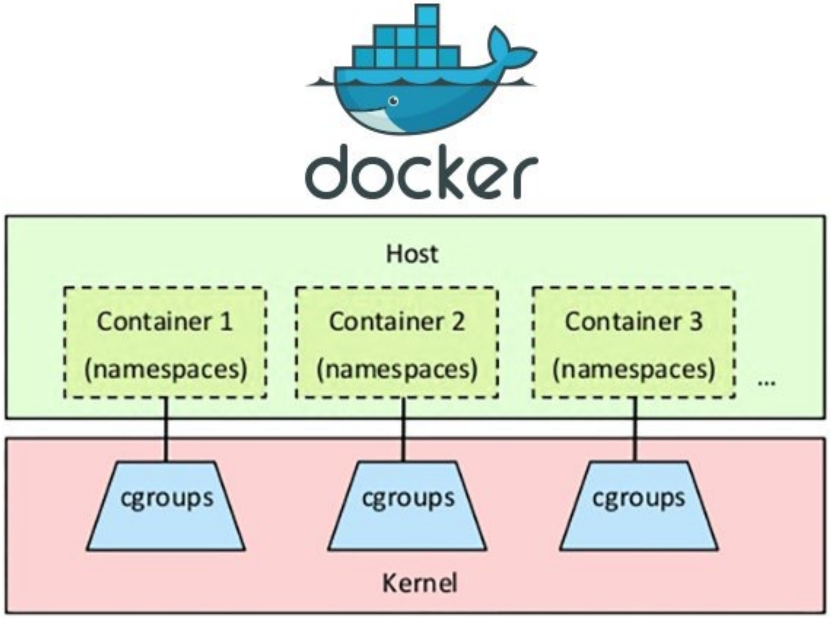
- Docker is developed ***primarily for Linux***, where it uses the resource isolation features of the Linux kernel such as:
  - **cgroups and kernel namespaces**,
  - and a **union-capable file system such as OverlayFS** and others.



https://medium.com/@knoldus/unionfs-a-file-system-of-a-container-2136cd11a779

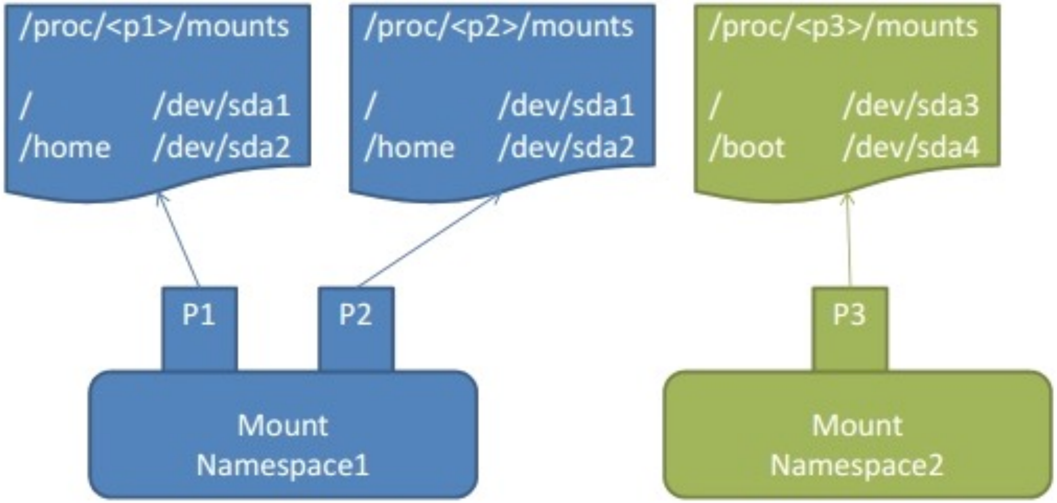
Docker Technology- cgroups

- **cgroups** (abbreviated from **control groups**) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.



Docker Technology- Namespaces

- Feature of the Linux kernel that partitions kernel resources such that one set of processes sees one set of resources while another set of processes sees a different set of resources.



https://wvi.cz/diyC/namespaces/

https://www.nginx.com/blog/what-are-namespaces-cgroups-how-do-they-work/