Case Study: OS Hot Topics

Instructor: Dr. Liting Hu

Introduction to Docker

What is Docker?

- Docker is an open-source project that automates the deployment of applications inside software containers, by providing an additional layer of **abstraction** and **automation** of **operating system–level virtualization** on Linux.

Docker History

- A dotCloud (PAAS provider) project
- Initial commit January 18, 2013
- Docker 0.1.0 released March 25, 2013
- 18,600+ github stars, 3800+ forks,740 Contributors…. and continues
- dotCloud pivots to docker inc. October 29, 2013

Docker Features

- Light-Weight
  - Minimal overhead (*cpu/io/network*)
  - Based on Linux containers
  - Uses layered filesystem to save space (AUFS/LVM)
  - Uses a copy-on-write filesystem to track changes
- Portable
  - Can run on any Linux system that supports LXC (today).
  - 0.7 release includes support for RedHat/Fedora family.
  - Raspberry pi support.
  - Future plans to support other container tools (lmctfy, etc.)
  - Possible future support for other operating systems (Solaris, OSX, Windows?)
- Self-sufficient
  - A Docker container contains everything it needs to run
  - Minimal Base OS
  - Libraries and frameworks
  - Application code
  - A docker container should be able to run anywhere that Docker can run.

The Challenge

Multiplicity of Stacks

Static website

nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB

postgresql + pgv8 + v8

Queue

Redis + redis-sentinel

Analytics DB

hadoop + hive + thrift + OpenJDK

Background workers

Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend

Ruby + Rails + sass + Unicorn

API endpoint

Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Do services and apps interact appropriately?

Multiplicity of hardware environments

Development VM

QA server

Customer Data Center

Public Cloud

Disaster recovery

Production Servers

Production Cluster

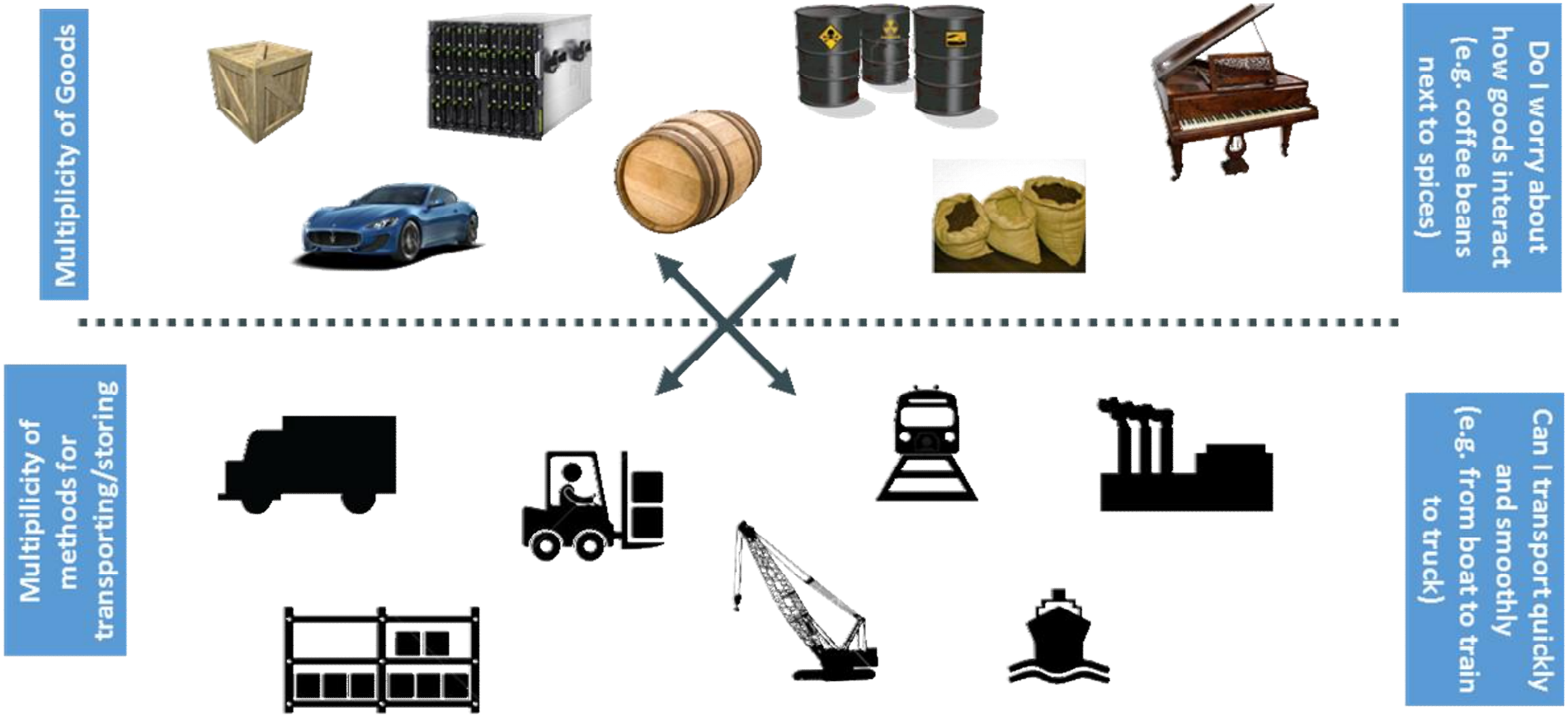
Contributor's laptop

Can I migrate smoothly and quickly?

The Matrix From Hell

	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?

Cargo Transport Pre-1960

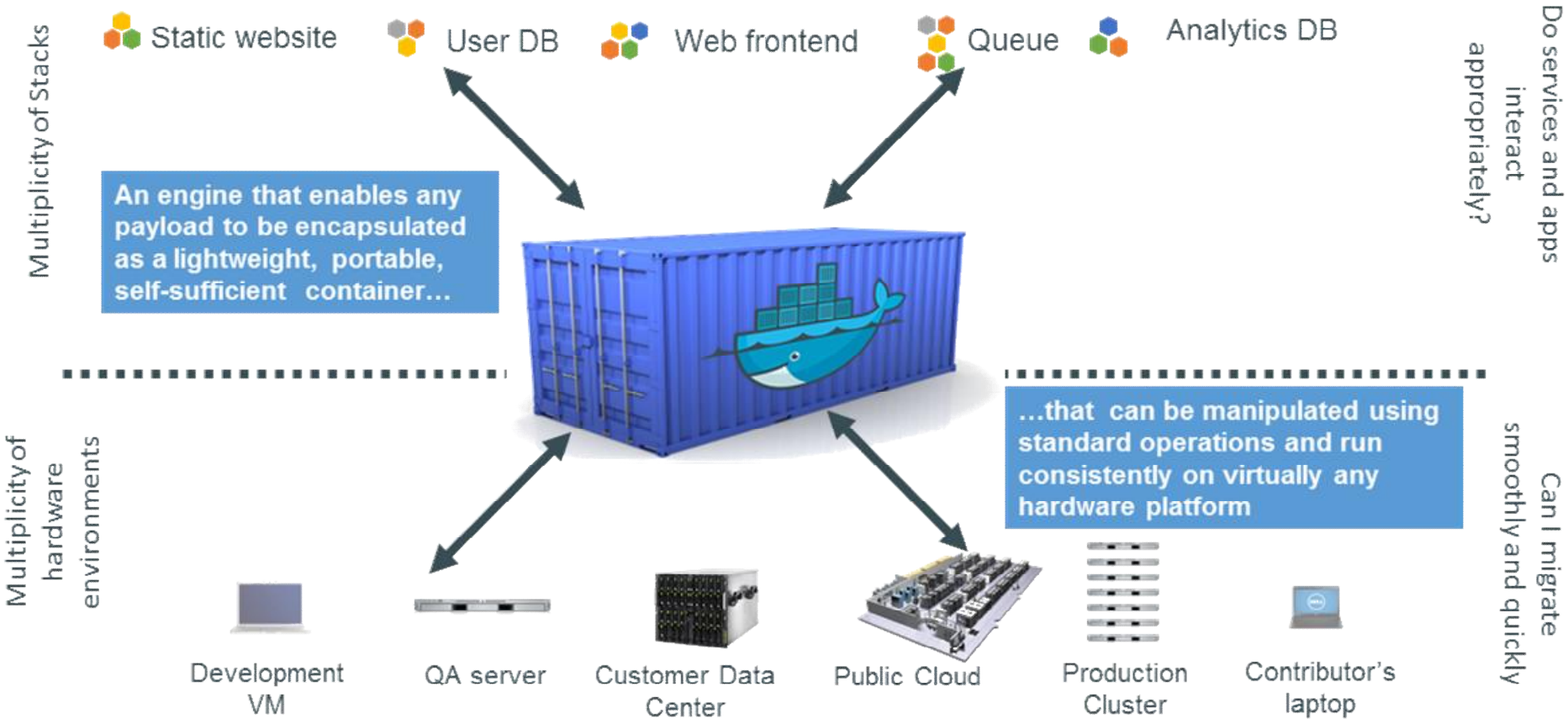


Also a Matrix from Hell

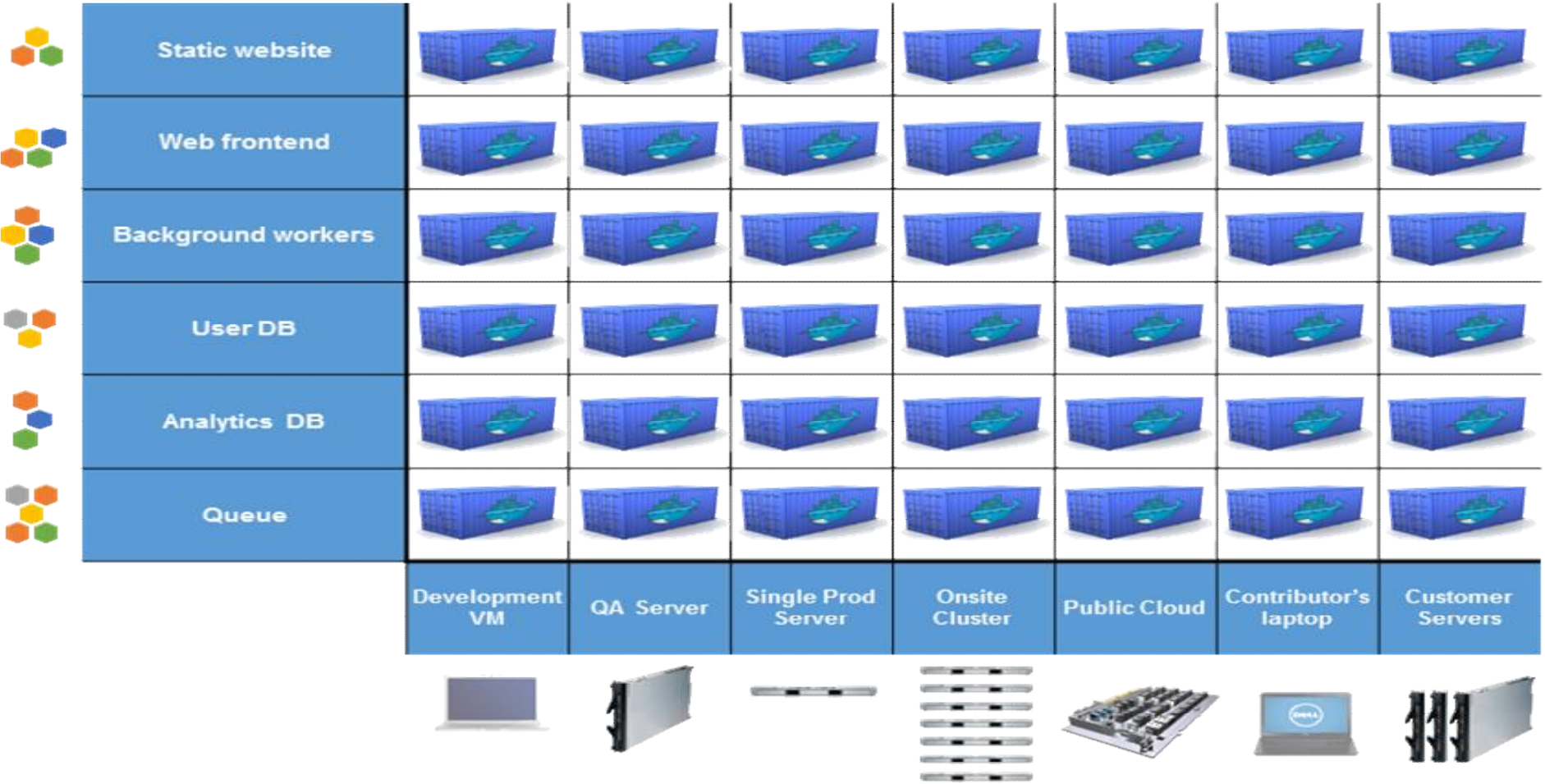
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

Solution: Intermodal Shipping Container

Docker is a Container System for Code

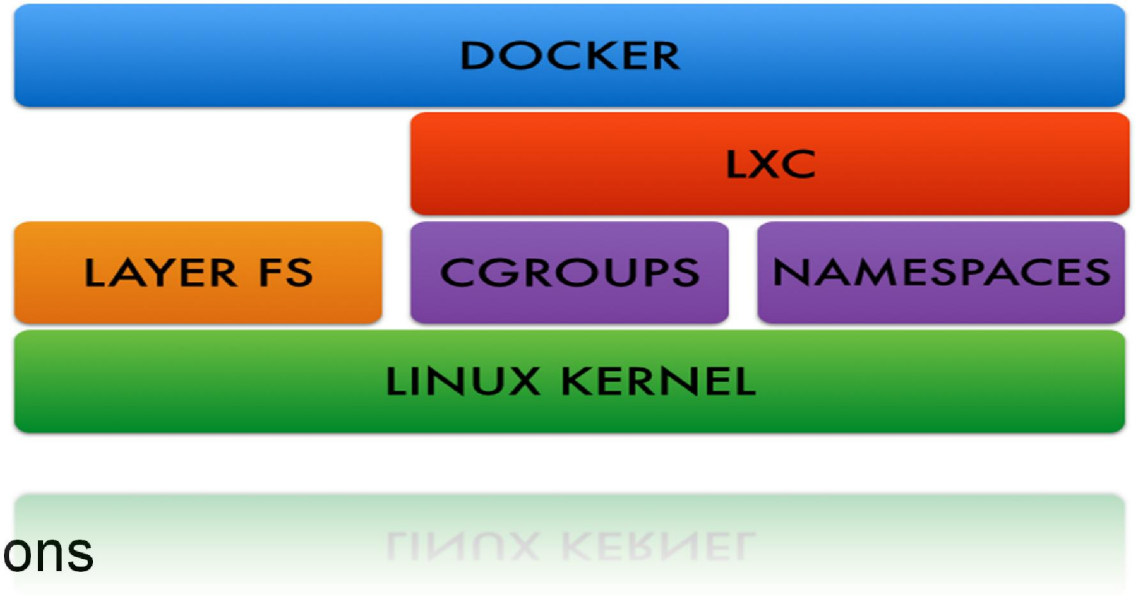


Docker Eliminates the Matrix from Hell



Docker Architecture

- Docker Engine
  - CLI
  - Docker Daemon
  - Docker Registry
- Docker Hub
  - Cloud service
    - Share Applications
    - Automate workflows
    - Assemble apps from components
- Docker images
- Docker containers



Docker Container

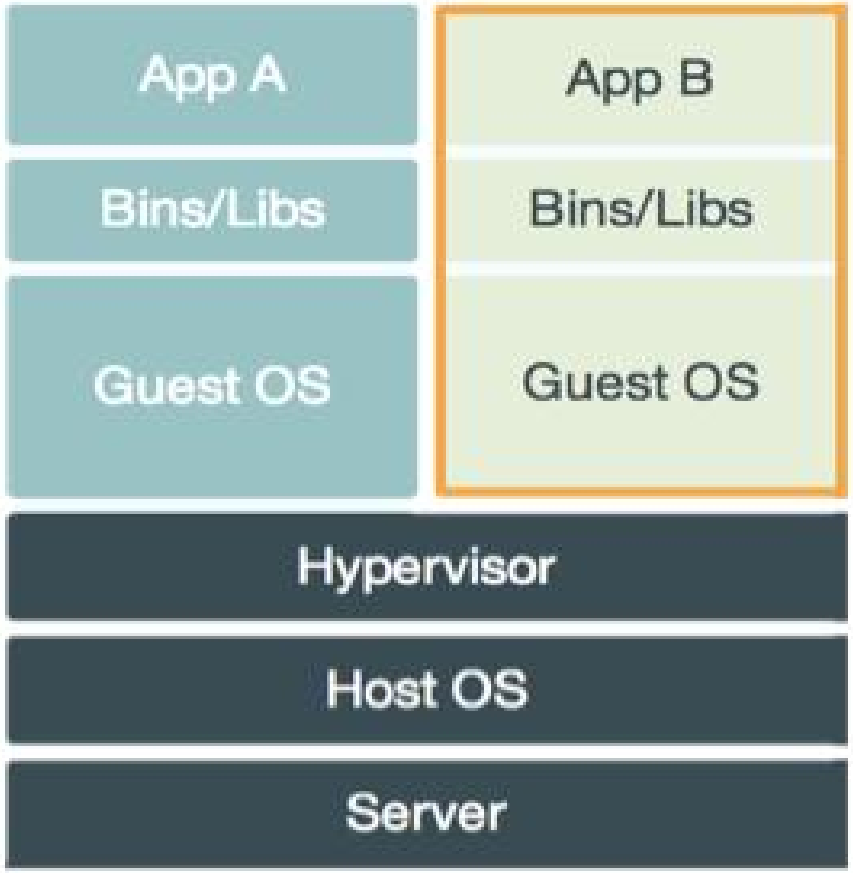
Units of software delivery (ship it!)

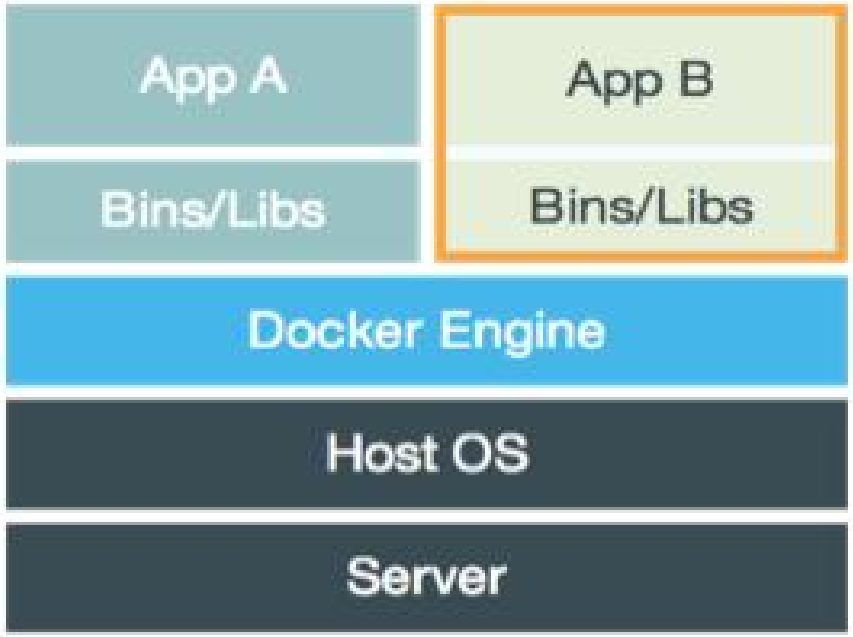
- Run everywhere
  - regardless of kernel version
  - regardless of host distro
  - (but container and host architecture must match*)
- Run anything
  - if it can run on the host, it can run in the container
  - i.e., if it can run on a Linux kernel, it can run

Docker Images

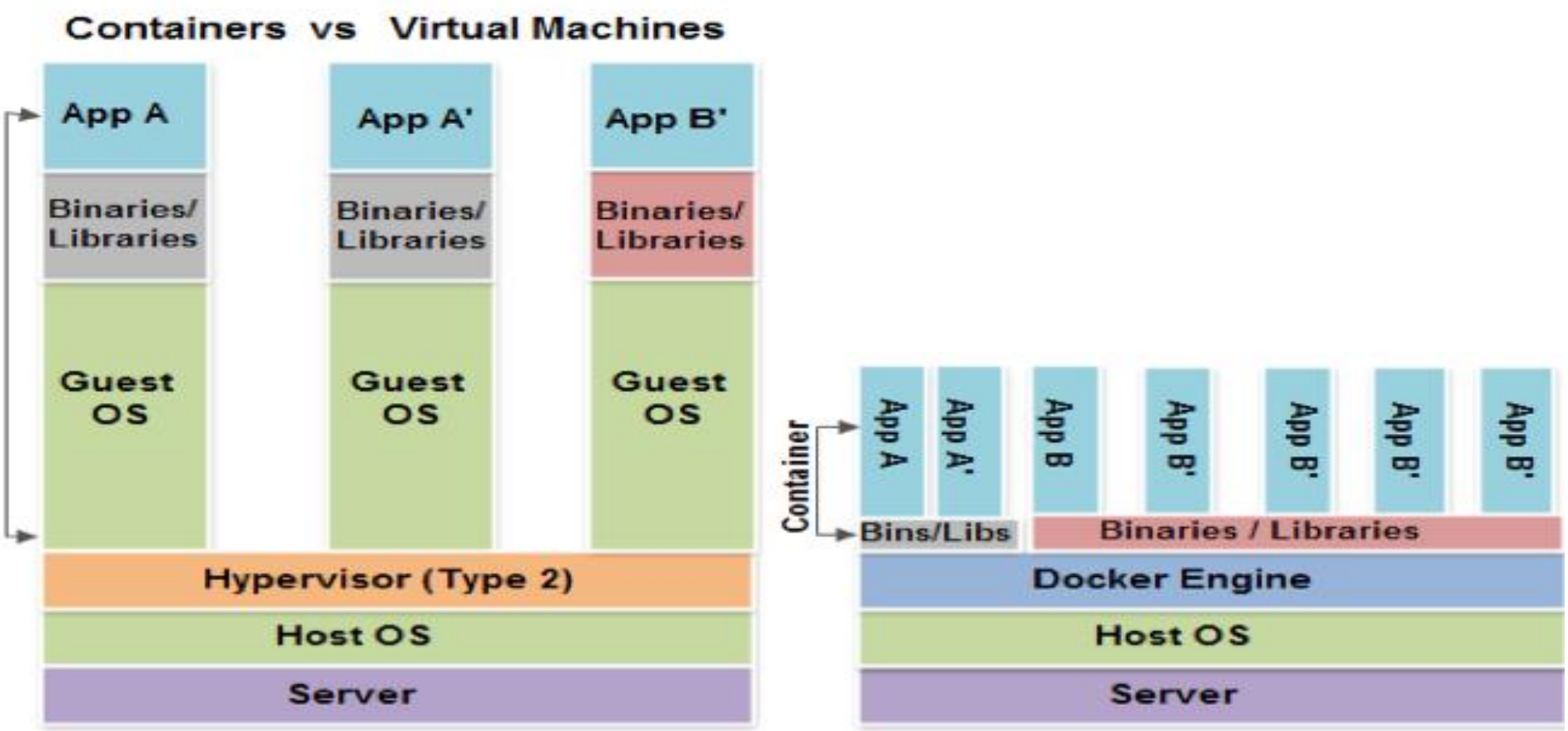
- NOT A VHD
- NOT A FILESYSTEM
- a read-only Layer
- do not have state
- Basically a tar file
- Has a hierarchy
  - Arbitrary depth
- Fits into the Docker Registry

Virtual Machine Versus Container





Virtual Machine Versus Container



Docker Container Lifecycle

- The Life of a Container
  - Conception
    - **BUILD** an Image from a Dockerfile
  - Birth
    - **RUN** (create+start) a container
  - Reproduction
    - **COMMIT** (persist) a container to a new image
    - **RUN** a new container from an image
  - Sleep
    - **KILL** a running container
  - Wake
    - **START** a stopped container
  - Death
    - **RM** (delete) a stopped container
- Extinction
  - **RMI** a container image (delete image)