ROBIN
SYSTEMS

SUN5617: Docker 101 for Oracle DBAs

Adeesh Fulay
Director of Products
@AdeeshF
adeesh@robinsystems.com
robinsystems.com

ABOUT ME

Adeesh Fulay
Director of Products
@AdeeshF
adeesh@robinsystems.com
linkedin.com/in/adeeshfulay

- Over 14 years of experience across virtualization, storage, middleware, databases, & big data applications
- Director of Products, Robin Systems
  - Application Virtualization Platform for big data apps and databases
- Previously @
  - VMware - vSphere, vCloud Director, vCloud Air (public cloud)
  - Oracle America – OEM, DBLM, DBaaS, DB Cloning, Job System, Reporting, Plugins, etc …
  - mValent Inc (acquired by Oracle) - Config mgmt of Middleware & Database products

AGENDA

- Why Containers?
- Linux Container Overview
- Look Under the Hood
- Getting Hands Dirty!!!
- Containerized Databases
- Summary

ROBIN

WHY CONTAINERS?

THE ANALOGY: CARGO TRANSPORT

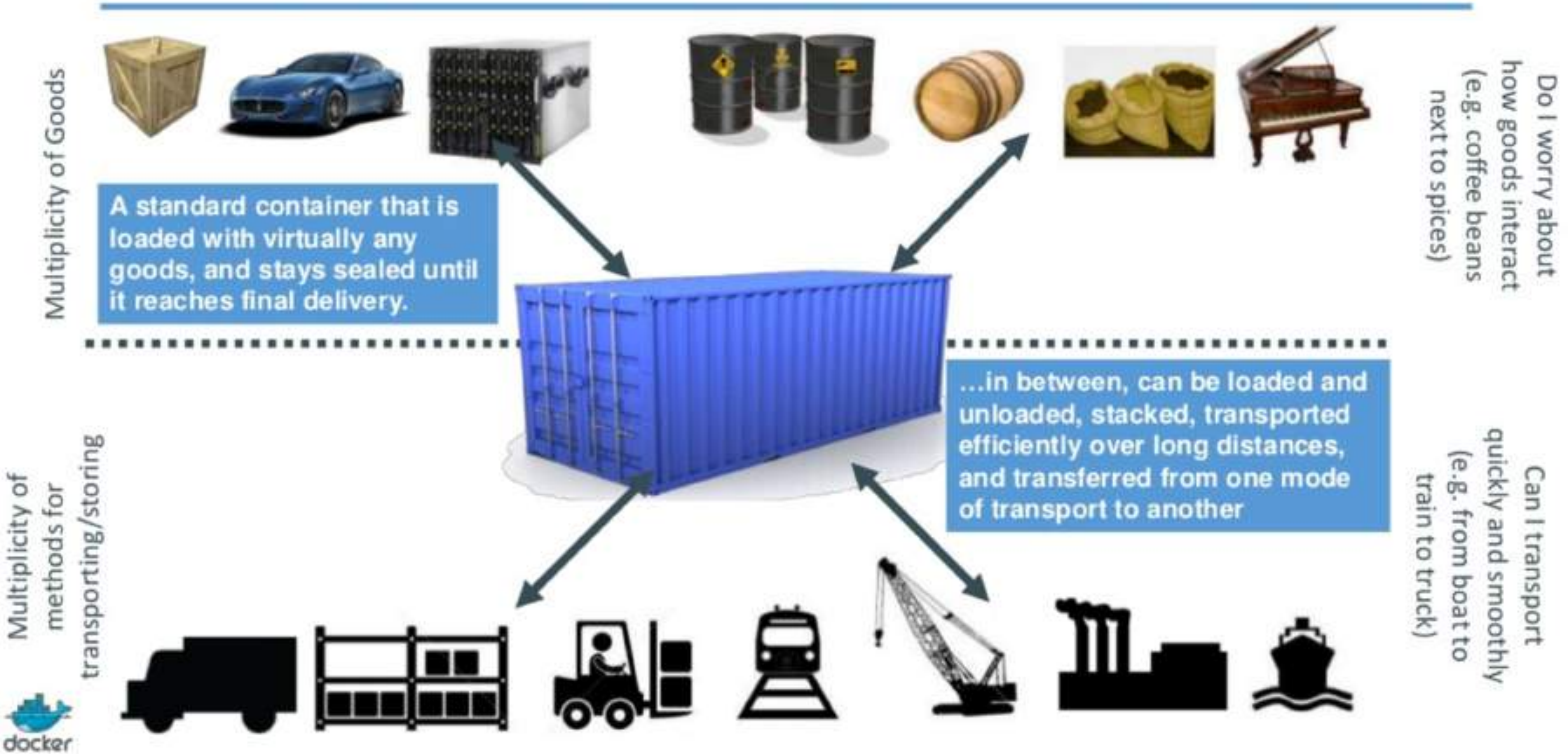
Multiplicity of Goods

Do I worry about how goods interact (e.g. coffee beans next to spices)

Multipilicity of methods for transporting/storing

Can I transport quickly and smoothly (e.g. from boat to train to truck)

CARGO SOLUTION: INTERMODAL SHIPPING CONTAINER



Multiplicity of Goods

A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.

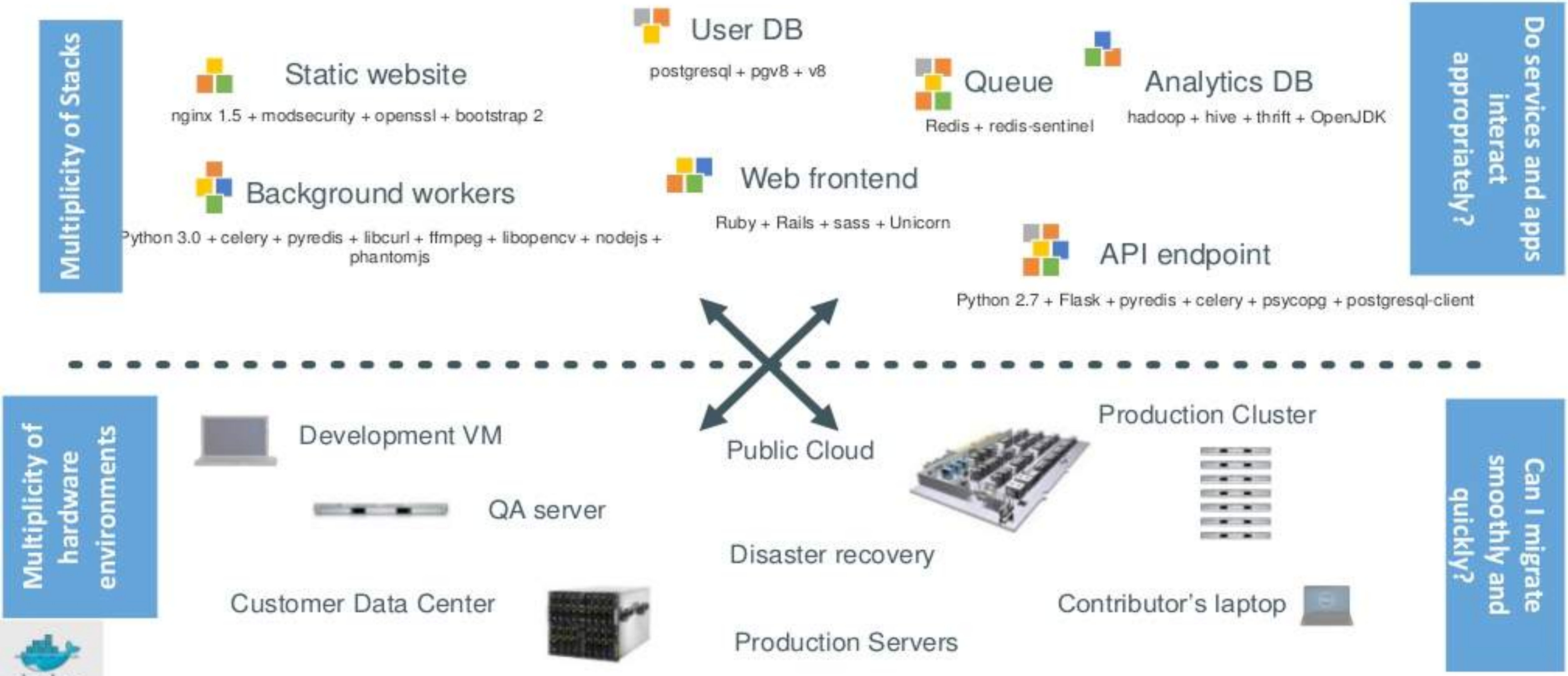
Do I worry about how goods interact (e.g. coffee beans next to spices)

…in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another

Multiplicity of methods for transporting/storing

Can I transport quickly and smoothly (e.g. from boat to train to truck)

THE BUSINESS 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

Public Cloud

Production Cluster

Disaster recovery

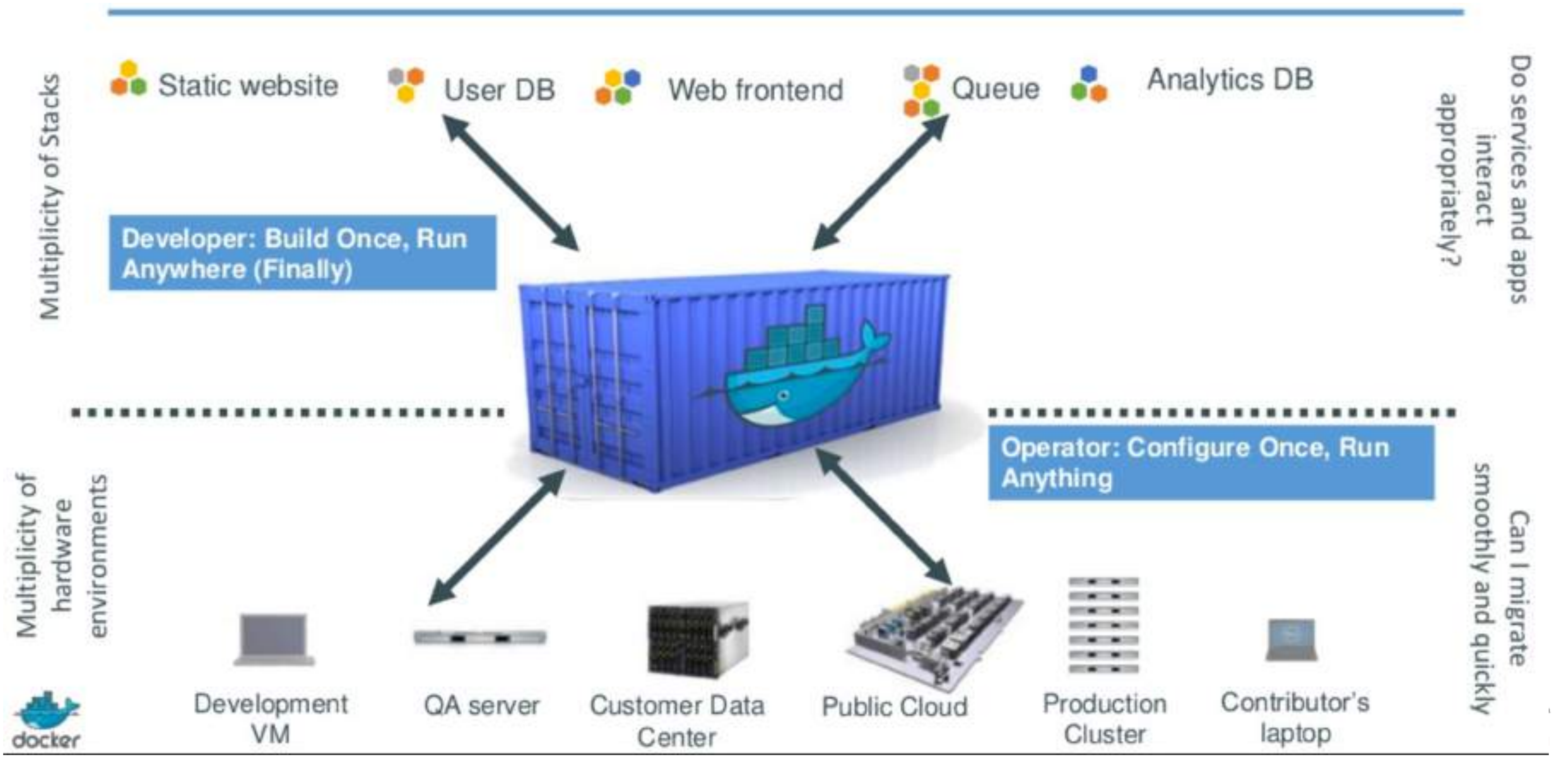
Customer Data Center

Contributor's laptop

Production Servers

Can I migrate smoothly and quickly?

BUSINESS SOLUTION: LINUX CONTAINERS



Multiplicity of Stacks

Static website

User DB

Web frontend

Queue

Analytics DB

Do services and apps interact appropriately?

Developer: Build Once, Run Anywhere (Finally)

Operator: Configure Once, Run Anything

Multiplicity of hardware environments

Can I migrate smoothly and quickly

Development VM

QA server

Customer Data Center

Public Cloud

Production Cluster

Contributor's laptop

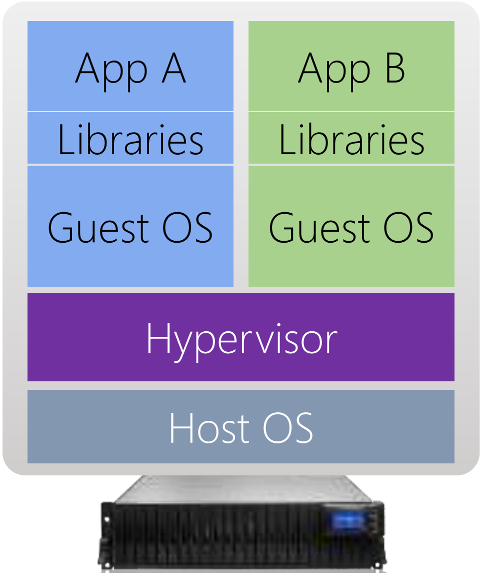
docker

ROBIN

LINUX CONTAINERS OVERVIEW

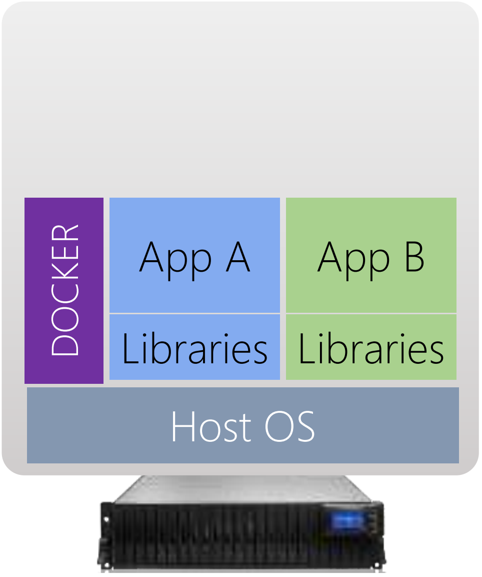
VIRTUALIZATION

Each virtual machine includes not only the app binary, but the entire operating system (Guest OS) and necessary libraries (which may weight 10s of GB)



CONTAINERS

Each container contains only the necessary application binary and its dependent libraries (which may weight 10s of MB). The operating system is shared by all containers.



UNDERSTANDING CONTAINERS

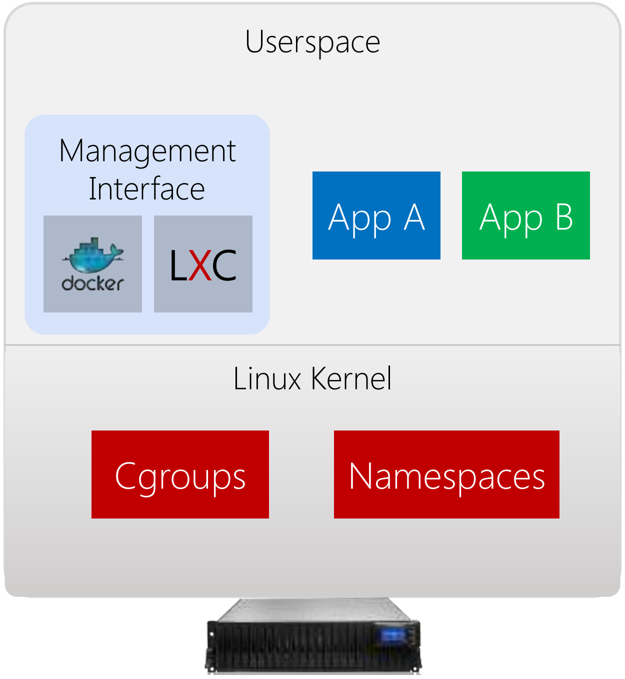
Each container has:

1. Its own network interface (and IP address)

2. Its own file system

3. Isolation (security)
   - Container A & B can't harm (or even see) each other
   - Uses Linux kernel's "**namespaces**" for this

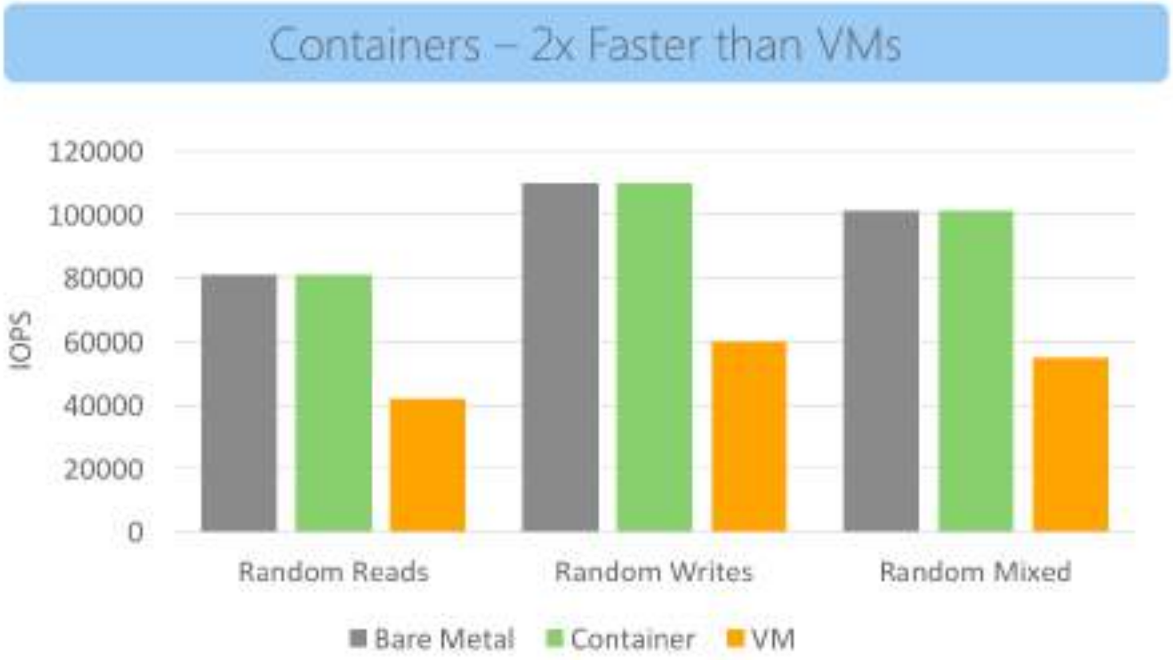
4. Isolation (resource usage)
   - Soft & hard quotas for CPU, RAM and IO
   - Uses Linux kernel's "**cgroups**" for this



Wait, this looks like a virtual machine!
So, what's the difference?

WHY CONTAINERS?

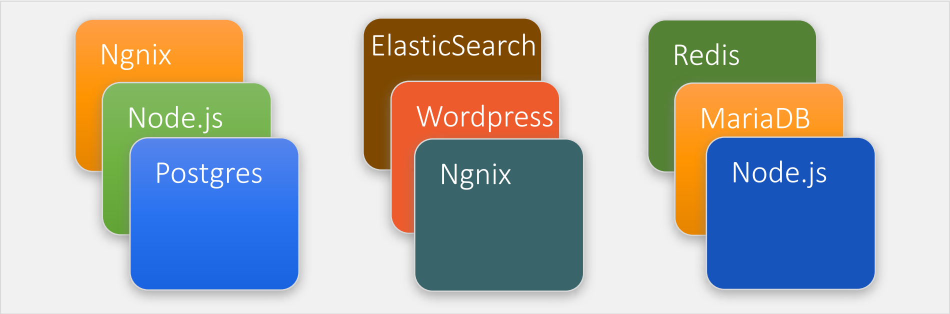
- OS-Based Lightweight Virtualization Technology
- Content & Resource Isolation
- Better Performance
- Better Efficiency & Smaller footprint
- Build, ship, deploy anywhere
- Separation of Duties


Containers – 2x Faster than VMs

Source: IBM Research
http://domino.research.ibm.com/library/cyberdig.nsf/papers/0929052195DD819C85257D2300681E7B/$File/rc25482.pdf

TYPES OF CONTAINERS

APPLICATION CONTAINERS



Ngnix
Node.js
Postgres

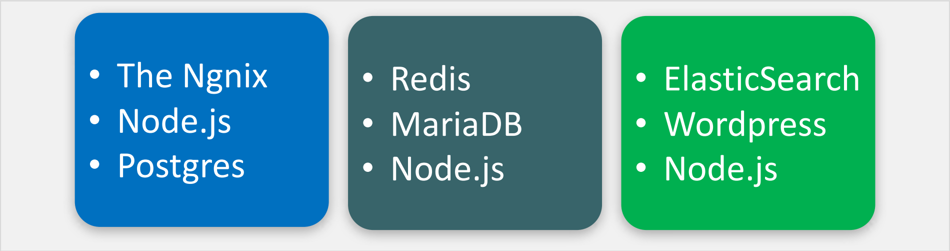
ElasticSearch
Wordpress
Ngnix

Redis
MariaDB
Node.js

- Each container runs a single application (single concern per container philosophy)
- Most popular container format. Example – Docker
- Requires applications to be repackaged and reconfigured to work with Docker image format
- Patch/Upgrade entails replacing container image

Great for Modern Applications

SYSTEM CONTAINERS



- The Ngnix
- Node.js
- Postgres

- Redis
- MariaDB
- Node.js

- ElasticSearch
- Wordpress
- Node.js

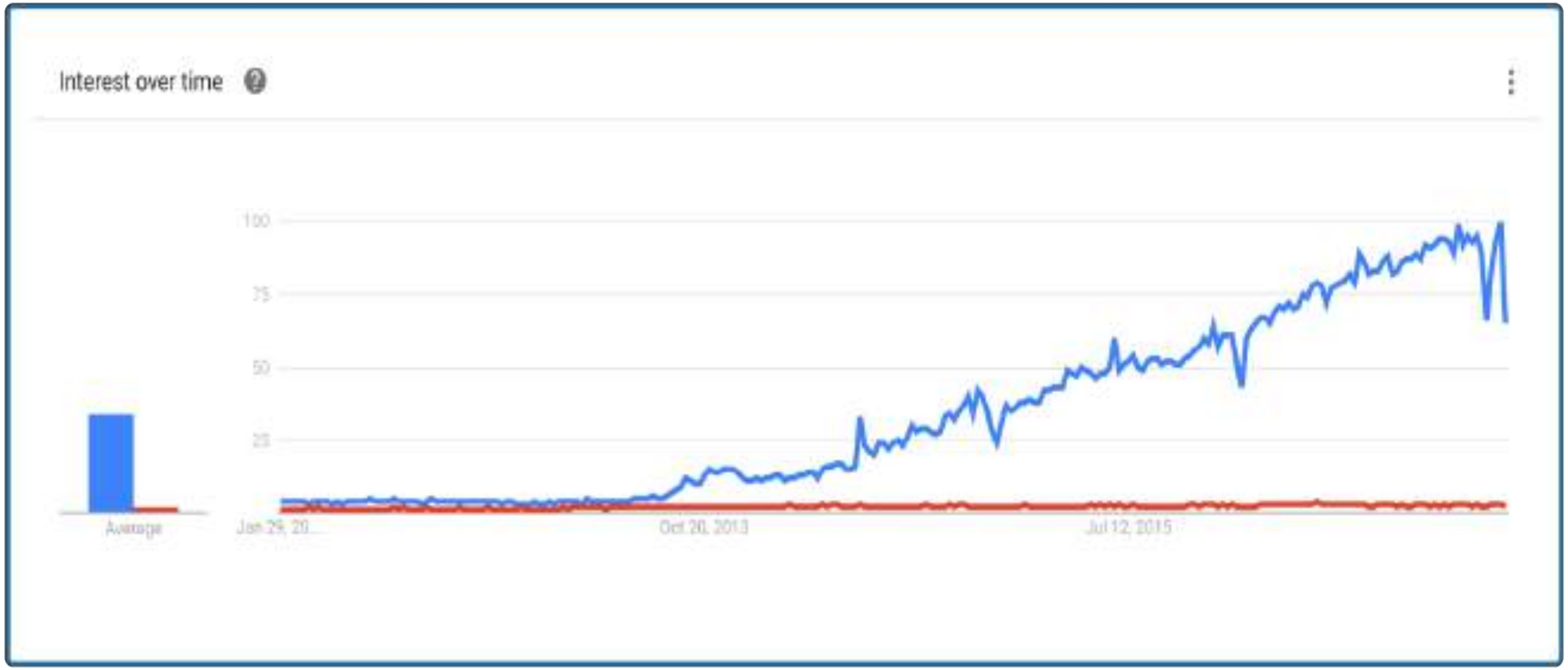
- Each container runs an entire service stack (multiple applications per container)
- Meant to be used as lightweight VM. Examples – LXC, OpenVZ, Solaris Zones
- No need to repackage applications in any special way
- Supports in-place patch/upgrade & SSH access

Great for Traditional Applications

Blog on LXC vs Docker: https://robinsystems.com/blog/containers-deep-dive-lxc-vs-docker-comparison/

WHY DOCKER?

Compare: ○ Docker vs ● LXC



A value of 100 is the peak popularity for the term.

~450K

images on DockerHub

~8 Billion

pulls from DockerHub

ROBIN

ORACLE SUPPORTS BOTH DOCKER AND LXC

Official Oracle on Docker repository:
https://github.com/oracle/docker-images

- General
  - OracleJava
  - OpenJDK
- Database
  - RDBMS
  - MySQL
  - NoSQL
- Middleware
  - Glassfish
  - WebLogic
  - Coherence
  - Tuxedo
  - HTTP Server

WELCOME TO THE

Oracle Container Registry

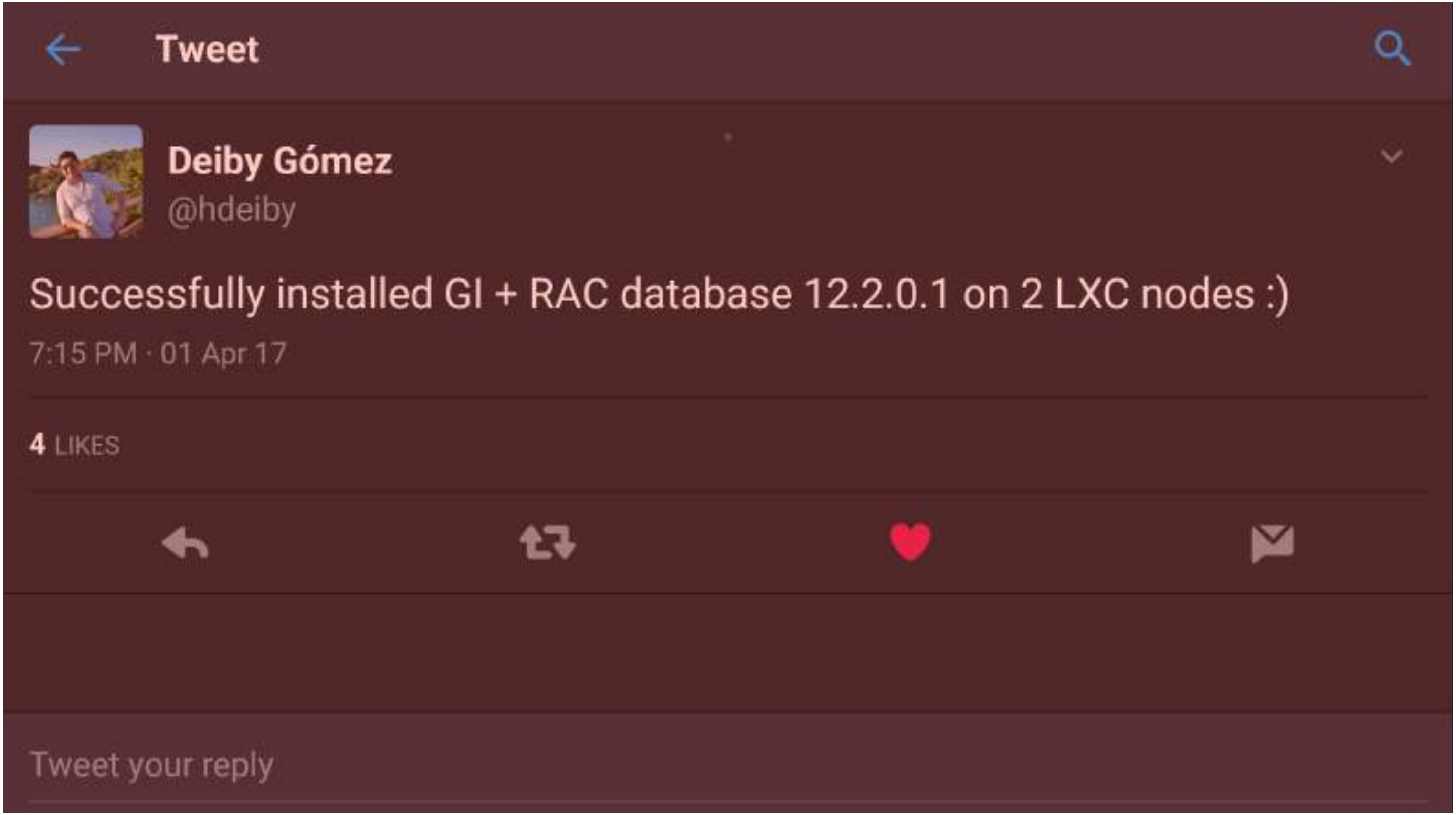
The Oracle Container Registry provides Oracle tested and approved Docker and other container images for Oracle products.

You must accept licensing terms before using Docker or other tools to pull images into your local registry or environment.

Please sign in to continue.

	Oracle Database	Oracle RAC (CRS + ASM)
LXC (Test & Production)	✓	✓
Docker (Dev Only)	✓	✗

LXC Support: http://www.oracle.com/technetwork/database/virtualizationmatrix-172995.html



Tweet

Deiby Gómez
@hdeiby

Successfully installed GI + RAC database 12.2.0.1 on 2 LXC nodes :)

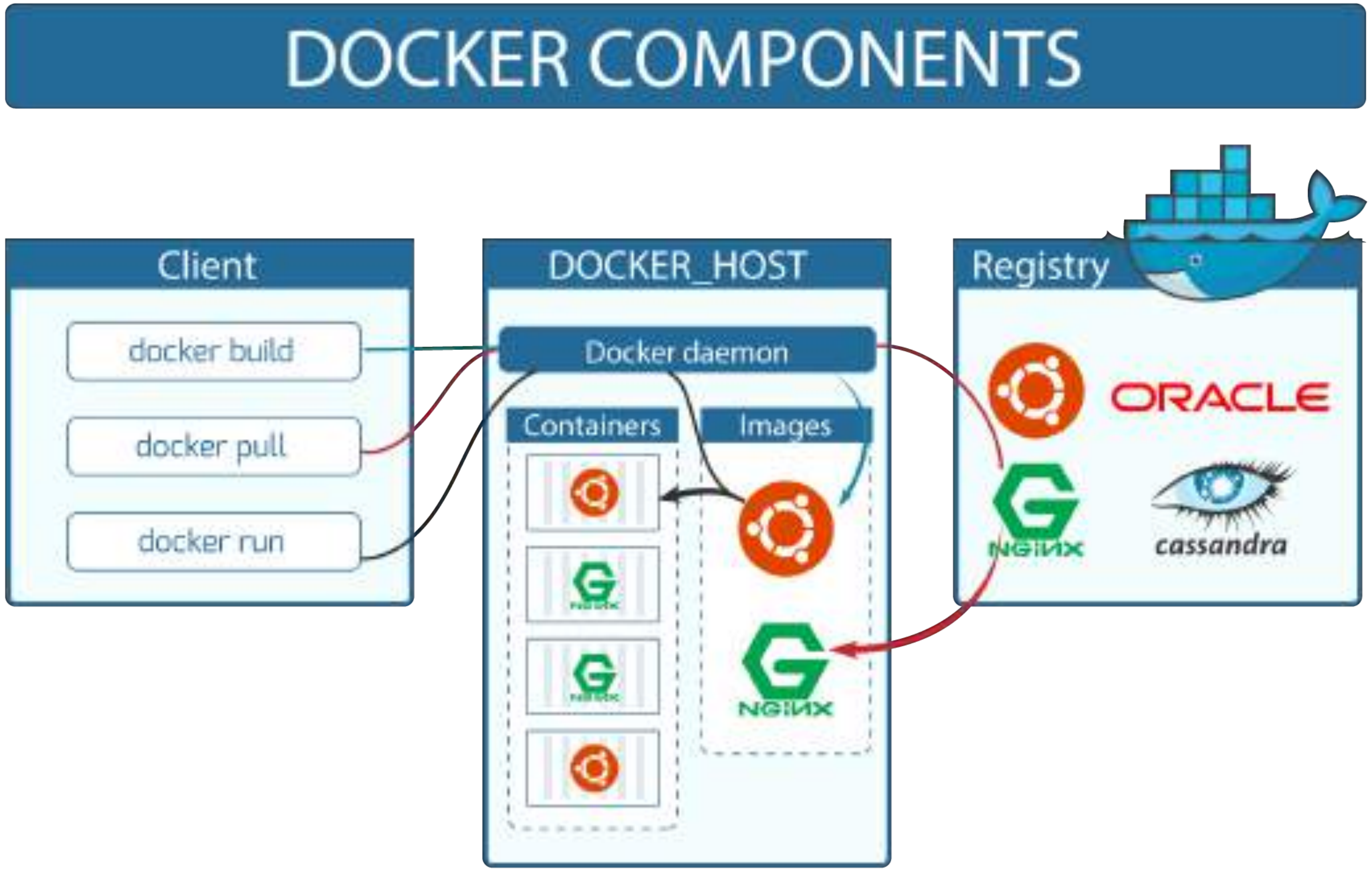
7:15 PM · 01 Apr 17

4 LIKES

ROBIN

UNDER THE HOOD

DOCKER ARCHITECTURE



<u>Key Components</u>

Docker Engine / Daemon

Docker Containers

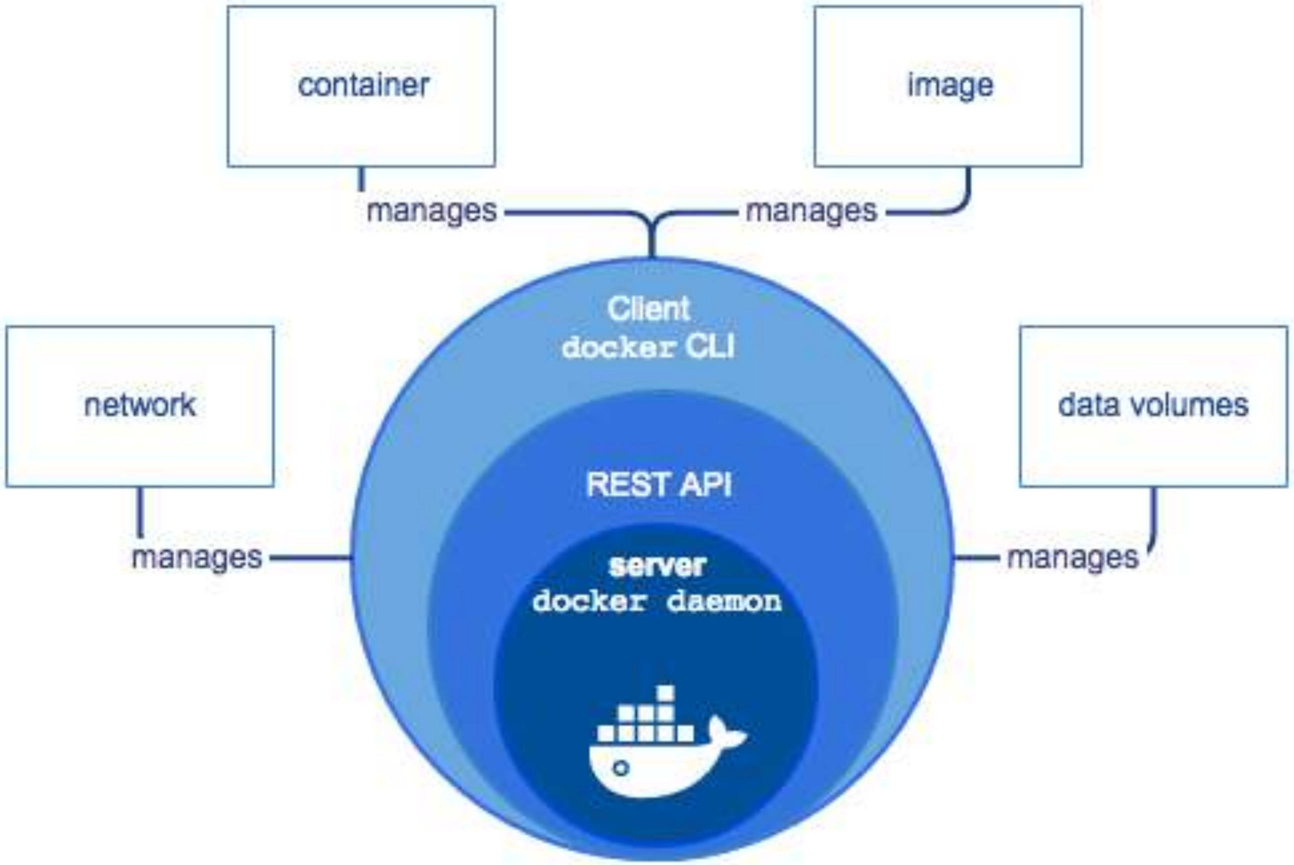
Docker Images

Docker Registry

Docker Client

Client can be on the same or different host as the daemon. They communicate over sockets or REST APIs.

DOCKER ENGINE

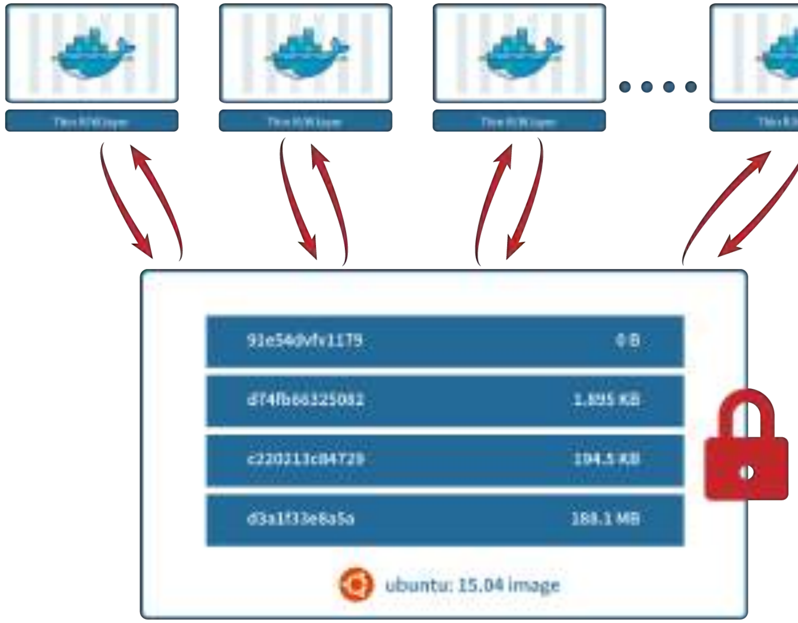


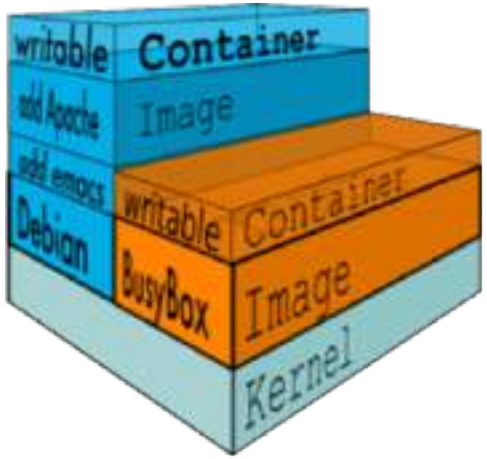
Docker Engine: Responsible for managing networking, images, containers, volumes, plugins, orchestration, etc

Install Docker Engine on your laptops and servers

IMAGES

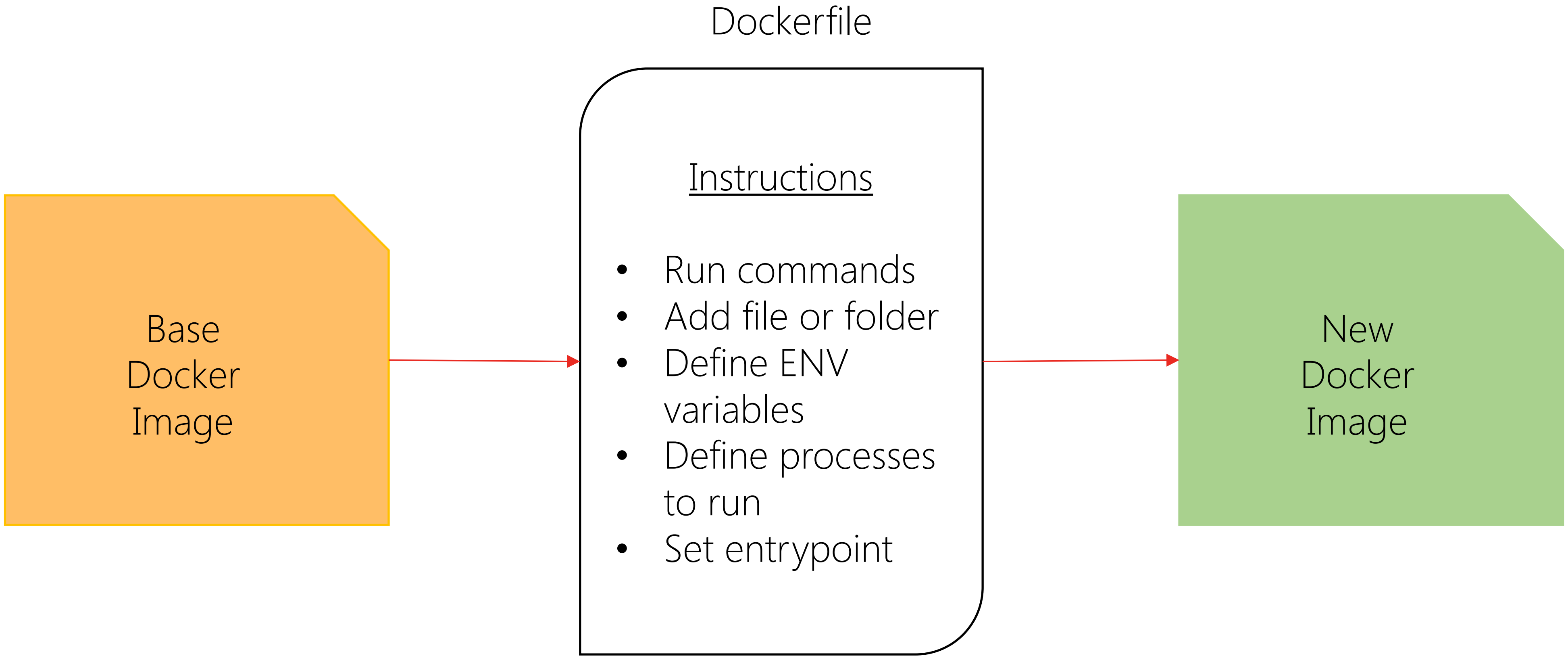
- Images are made up of multiple r/o layers
- Containers are a thin r/w layer on top
- Layers are shared by multiple images
- Storage drivers
  - AUFS (Ubuntu, OSX)
  - Device mapper (RHEL, CentOS)
  - BTRFS (Oracle Linux)
  - Overlay, Overlay2
  - ZFS
- Layer default location: /var/lib/docker





Additional Reading: https://adeeshfulay.wordpress.com/2017/09/08/understanding-docker-images-and-layers/

DOCKERFILES



Dockerfile

Base Docker Image

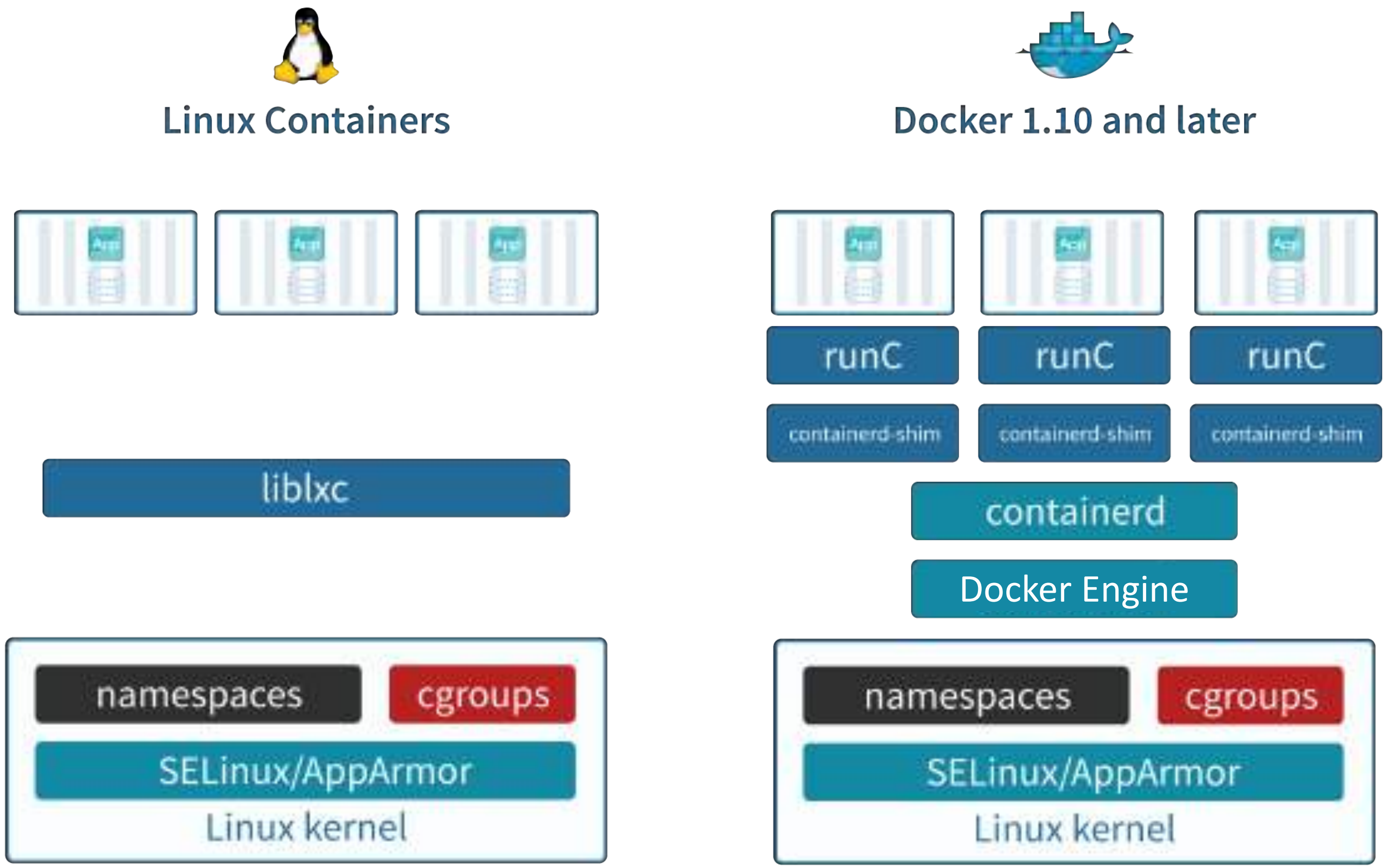
Instructions

- Run commands
- Add file or folder
- Define ENV variables
- Define processes to run
- Set entrypoint

New Docker Image

$ Docker build –t <tag> <dir with dockerfile>

CONTAINER MANAGEMENT



Linux Containers

liblxc

namespaces cgroups

SELinux/AppArmor

Linux kernel

Docker 1.10 and later

runC runC runC

containerd-shim containerd-shim containerd-shim

containerd

Docker Engine

namespaces cgroups

SELinux/AppArmor

Linux kernel

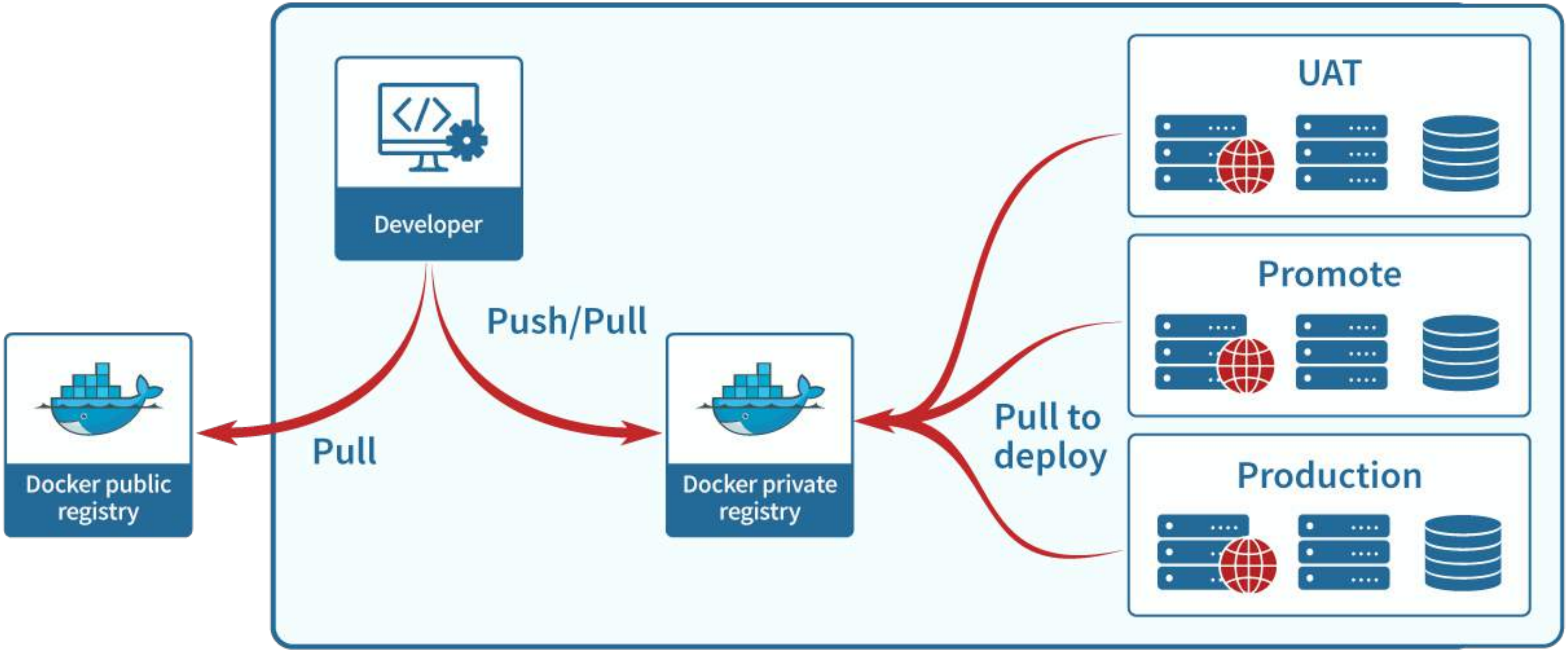
Container is like a directory. It is created from an image, and contains everything required to run an app. Containers like VMs have state and are portable.

Containerd: Simple daemon that uses runtimes to manage containers

runC: Docker's default container runtime. Can be replaced with other OCI compliant runtimes.

IMAGE REGISTRY

Registry is a distribution component of Docker. It can be private or public (Docker Hub: https://hub.docker.com)



Developer

Push/Pull

Pull

Docker public registry

Docker private registry

Pull to deploy

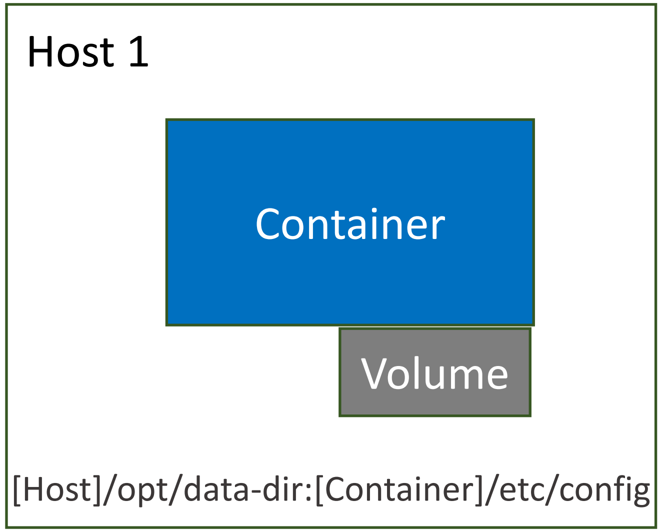
UAT

Promote

Production

DOCKER VOLUMES

Local Storage



Host 1

Container

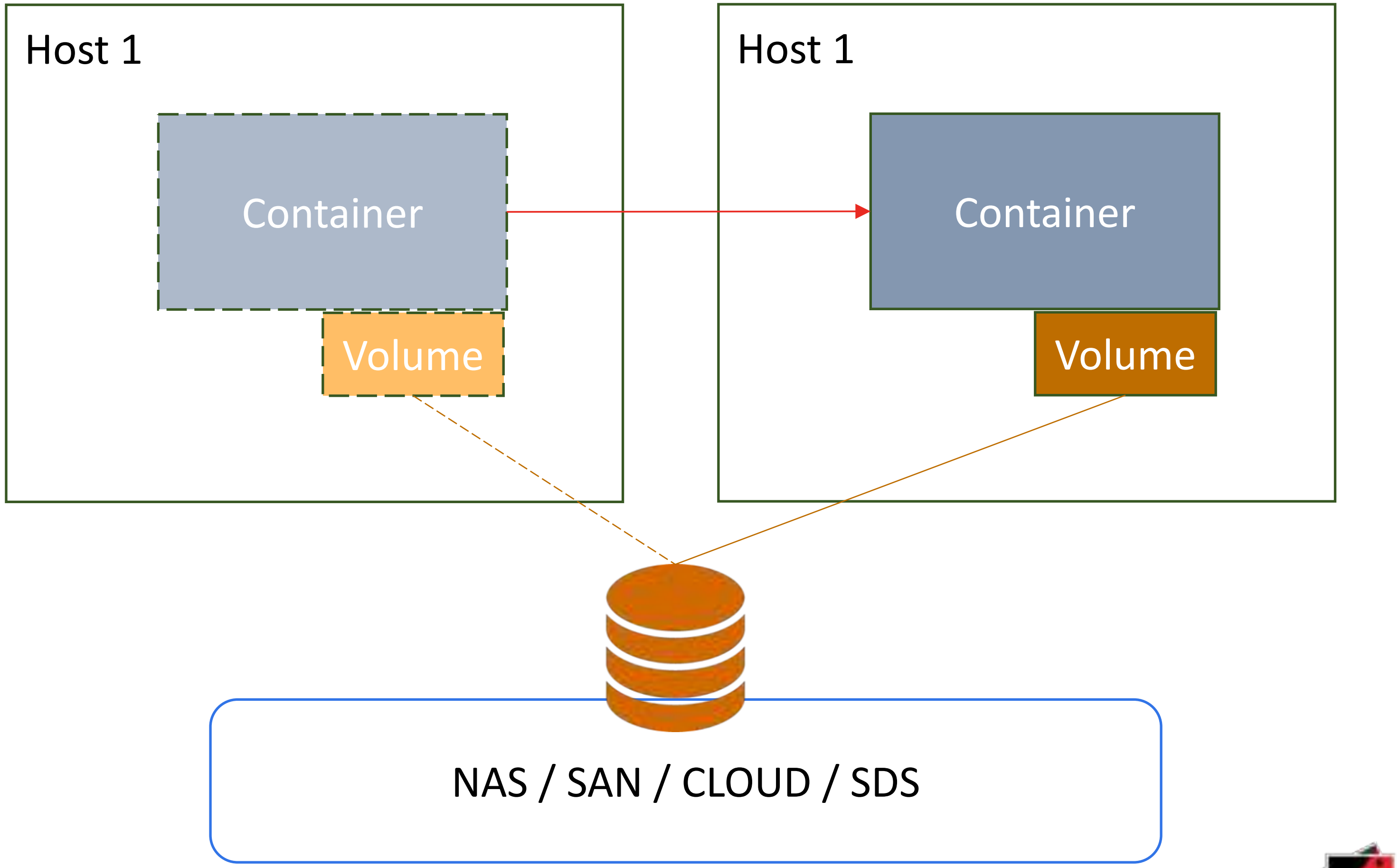
Volume

[Host]/opt/data-dir:[Container]/etc/config

Local Storage

Choose between mobility & performance

Distributed / Shared Storage



Host 1

Container

Volume

Host 1

Container

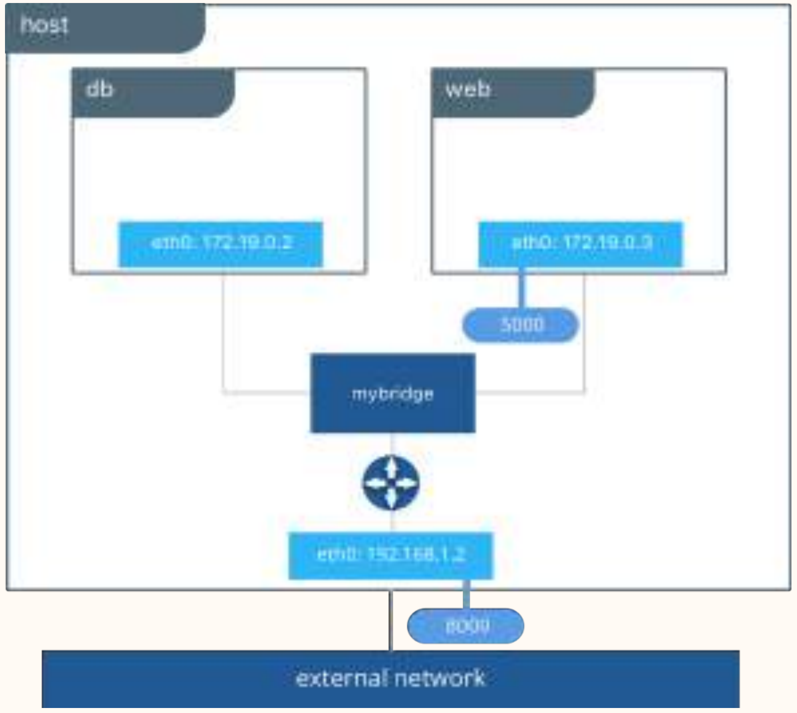
Volume

NAS / SAN / CLOUD / SDS

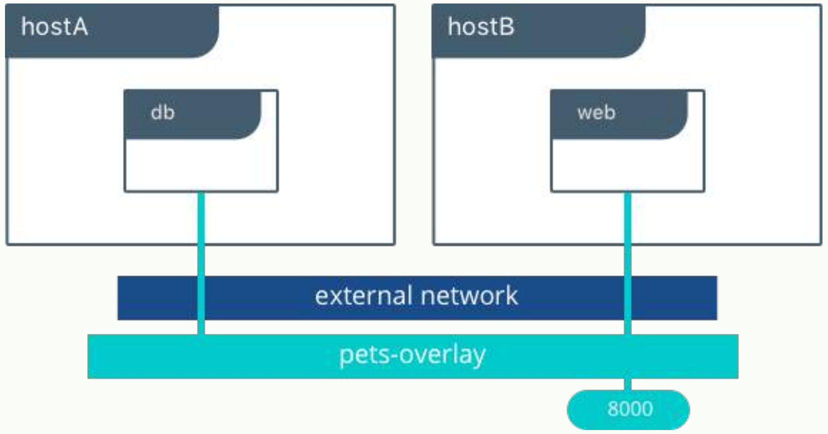
DOCKER NETWORKING

Network types	Details
None	No external n/w
Host	Share host n/w stack
Bridge	NATing, expose ports, poor performance
macvlan	Uses host subnet, VLAN, swarm mode
Overlay	Multi-host, VXLAN, swarm mode
3rd Party Plugins	OpenvSwitch, Weave, …..

Reference: https://blog.docker.com/2016/12/understanding-docker-networking-drivers-use-cases/



Bridge Network



Overlay Network

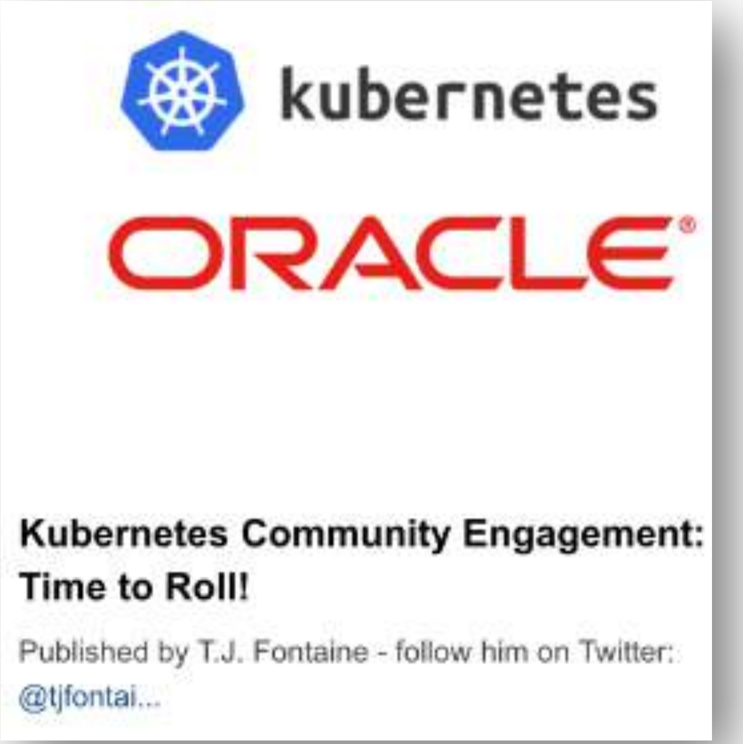
CONTAINER ORCHESTRATION

- Multi-container, Multi-Host scheduling
- Supports multiple container formats- Docker & LXC
- Placement, Affinity and Anti-affinity rules
- Multi-Tenancy, RBAC, AuthN, & AuthZ
- Scaling Options – Up/Down, Out/Back
- Management of Storage & Networking
- Data Management - Snapshot, restore, time travel, clone
- High Availability & Auto Failover
- Service discovery and load balancing
- Extensible – Onboard custom apps

kubernetes MESOS docker SWARM RANCHER HashiCorp Nomad ROBIN

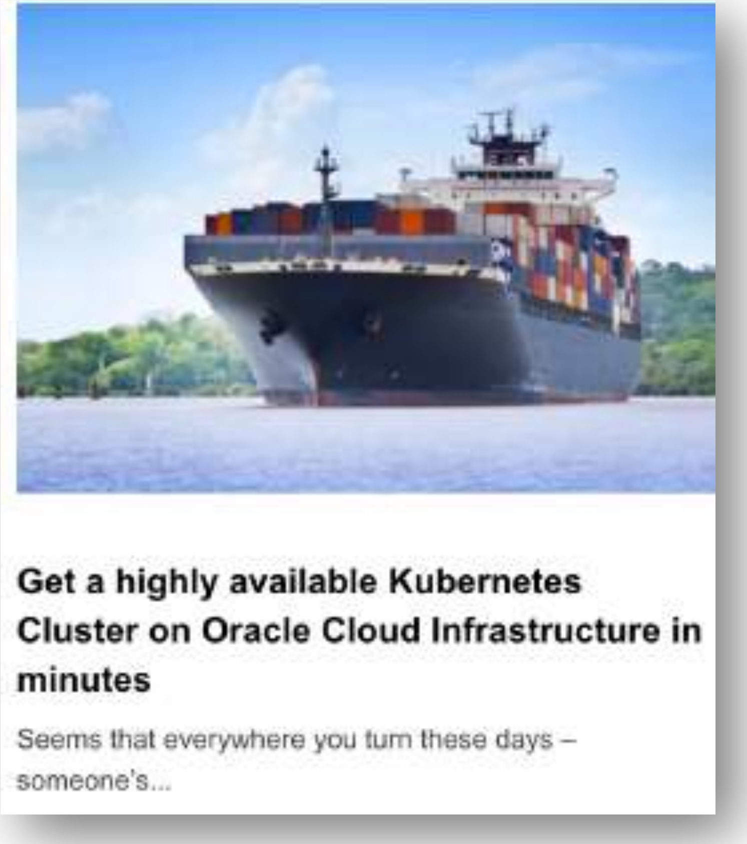
ORACLE, DOCKER, & KUBERNETES

Oracle is late to the party, but making big investments now to catch up!!



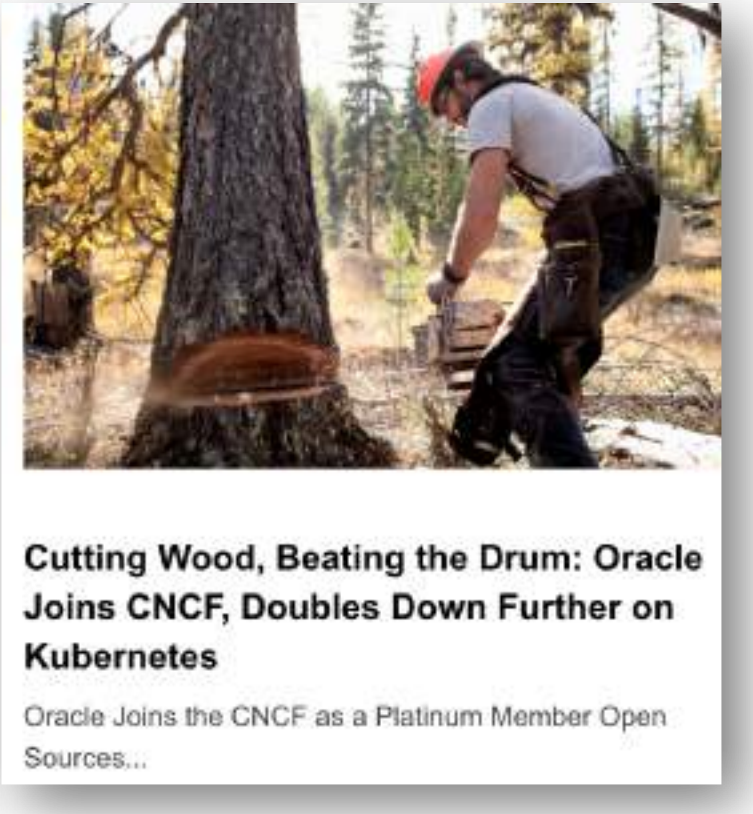
Kubernetes Community Engagement: Time to Roll!

Published by T.J. Fontaine - follow him on Twitter: @tjfontai...



Get a highly available Kubernetes Cluster on Oracle Cloud Infrastructure in minutes

Seems that everywhere you turn these days – someone's...



Cutting Wood, Beating the Drum: Oracle Joins CNCF, Doubles Down Further on Kubernetes

Oracle Joins the CNCF as a Platinum Member Open Sources...

ROBIN

GETTING HANDS DIRTY!!!

- DEMO

Mike Rowe of Dirty Jobs

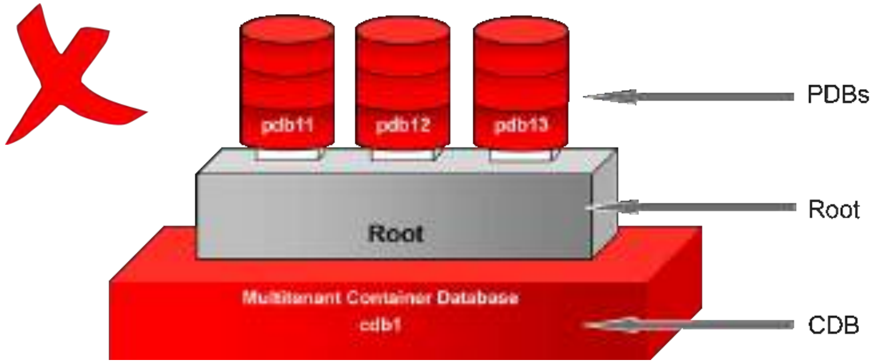
DEMO: DOCKER EXAMPLES

- Docker info
- Dockerhub & Oracle container registry
- Docker images & Docker pull
- Dockerfile
- Docker run –d <oradb> -v …
- Docker ps -a
- Docker exec –ti <cname> bash
- Docker logs …
- Docker inspect <oradb>
- Docker network ls
- Docker volume ls

ROBIN

Containerized Databases

- Good Idea or Disaster?



OPERATIONAL CHALLENGES

How do I apply patches and upgrades without bringing down the application?

How do I handle spikes and growth?

How can I rapidly clone my production for test/dev runs?

How do I avoid overprovisioning from the start?

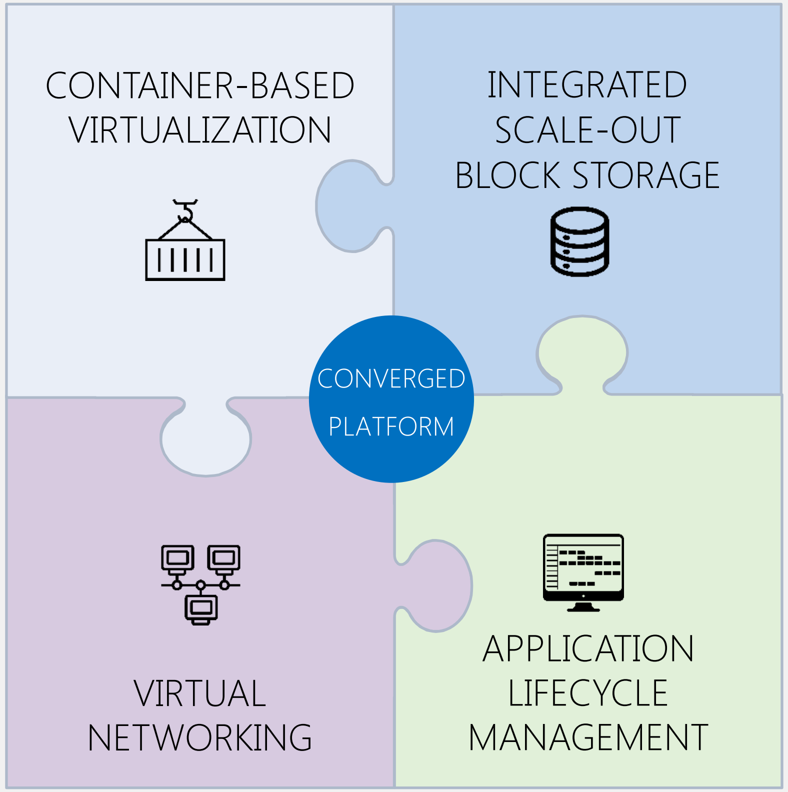
How do I get the absolute best performance?

How do I quickly deploy my applications?

Can I run multiple applications on the same setup without worrying about noisy neighbors?

Robin AVP: Virtualize And Manage BigData & Database Apps with Bare Metal Performance

Hortonworks cloudera Spark kafka elasticsearch. cassandra mongoDB ORACLE MySQL PostgreSQL weblogic Custom Apps



CONTAINER-BASED VIRTUALIZATION

INTEGRATED SCALE-OUT BLOCK STORAGE

CONVERGED PLATFORM

VIRTUAL NETWORKING

APPLICATION LIFECYCLE MANAGEMENT

No Virtualization Overhead
40% - 60% performance advantage

Guaranteed Min and Max IOPS
Consolidate without losing predictability

Space-efficient snapshots and clones in minutes
Application time travel (storage, app, network and config)

Application-aware Infrastructure plumbing
Deploy applications guaranteeing compute+storage anti/affinity rules

Push-button Application Provisioning, Elastic Scaling
10x time saving by managing application not infrastructure

CONSOLIDATION OPTIONS FOR ORACLE DATABASES

Options → Criteria ↓	Virtual Machine	Oracle 12c Multitenant	Containers on Robin
Performance overhead	Significant Hypervisor layer, Guest OS	Significant Shared Redo logs, Noisy neighbors problem	Negligible Completely independent, no hypervisor
Availability	High One VM doesn't impact another one	Medium CDB shutdown takes down all PDBs with it	High Just like VMs
Isolation	Excellent	Good Shared buffer cache	Excellent
Performance predictability	Poor Cannot cap IOPS at the hypervisor layer	Good IOPS control only available on Exadata	Excellent. Built into the platform
Agility	Good	Excellent	Excellent
Manageability	High OS sprawl	High Challenges in getting patching window	Low No OS sprawl, no additional licenses

https://robinsystems.com/solutions/relational-databases/

THANK YOU

🌐 www.robinsystems.com ✉ info@robinsystems.com

Adeesh Fulay
Director of Products
@AdeeshF
✉ adeesh@robinsystems.com