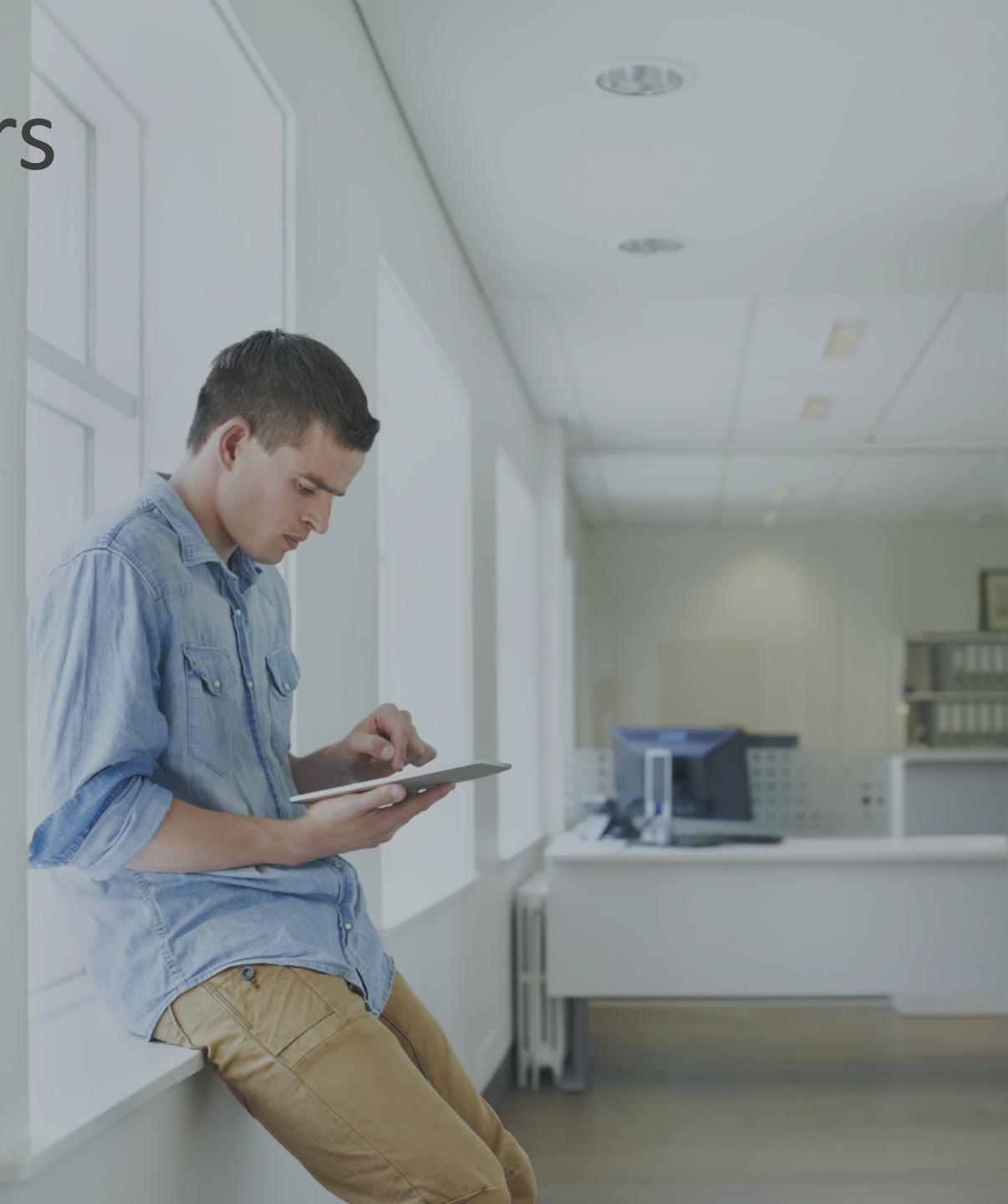
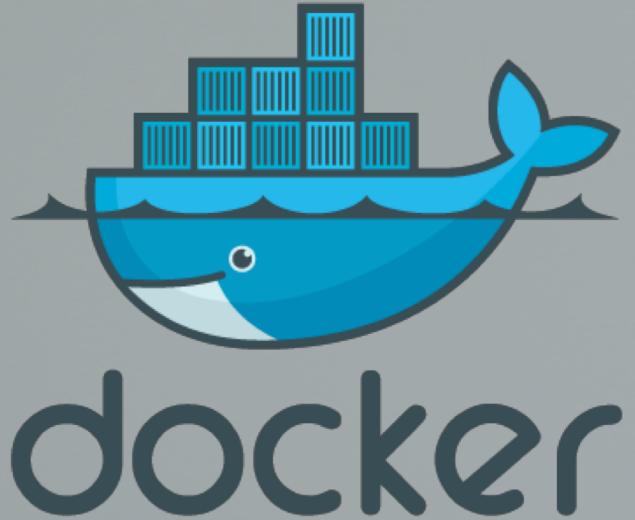


Intro to Docker Containers



Docker Introduction

What is Docker?!?

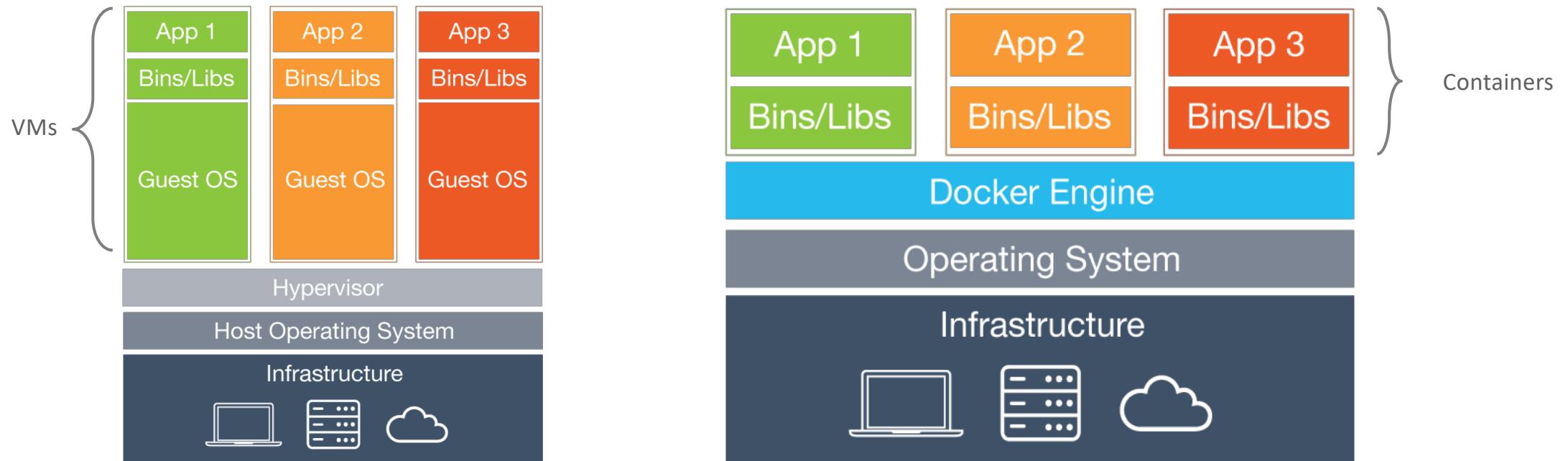
Docker is an Open platform for developers and sysadmins to build, ship and run distributed applications.

It can run on most Linux distributions, Windows and Mac OS running Docker Engine (Toolbox).

It is supported by most of cloud providers and provide a popular Dev/Test, CI & DevOps platform for many use cases.

Basic Architecture and comparison to VM's

Virtual Machines vs. Containers



Virtual Machines

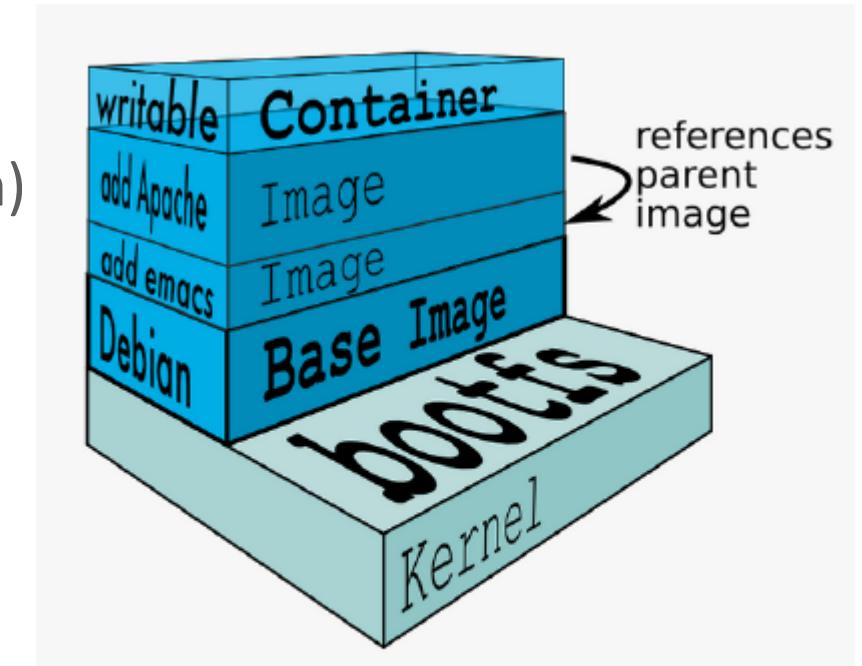
- Each virtual machine (VM) includes the app, the necessary binaries and libraries and an entire guest operating system

Containers

- Containers include the app & all of its dependencies, but share the kernel with other containers.
- Run as an isolated process in userspace on the host OS
- Not tied to any specific infrastructure – containers run on any computer, infrastructure and cloud.

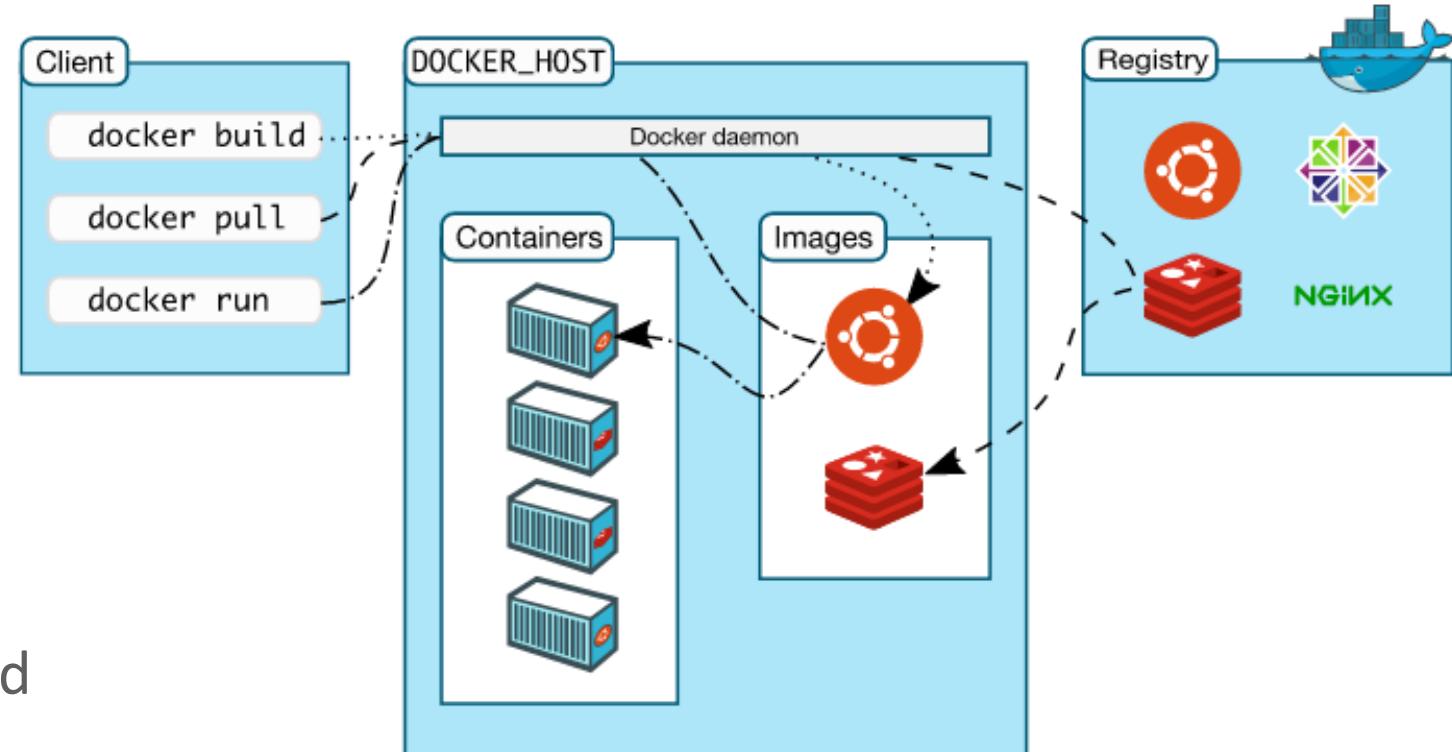
Docker Images

- An image is a collection of files and some meta data
- Images are comprised of multiple layers, multiple layers referencing/based on another image (Union File System)
- Each image contains software you want to run
- Every image contains a base layer
- Layers are read only



Docker Architecture

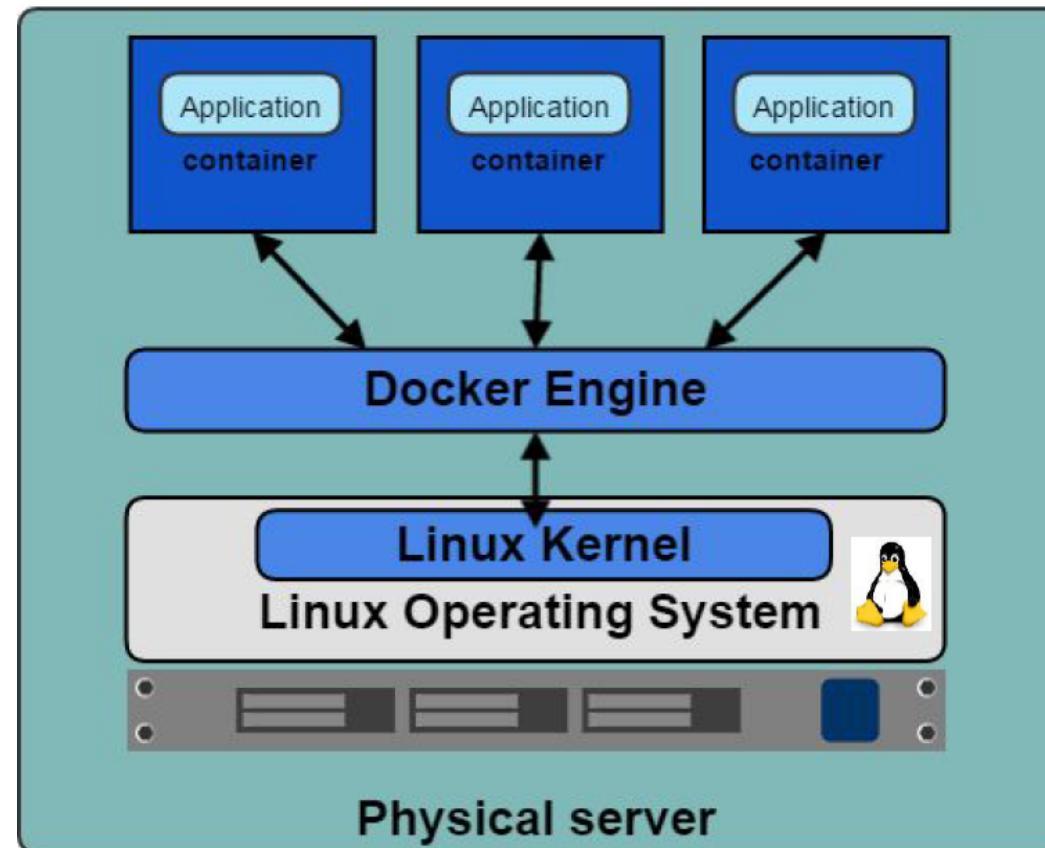
- Docker client – Command Line Interface (CLI) for interfacing with the Docker
- Dockerfile – Text file of Docker instructions used to assemble a Docker Image
- Image – Hierarchies of files built from a Dockerfile, the file used as input to the docker build command
- Container – Running instance of an Image using the docker run command
- Registry – Image repository



Source: Docker docs and <https://docs.docker.com/glossary/>

Docker Engine

- Container execution and admin
- Uses Linux Kernel namespaces and control groups
- Namespaces provide for isolated workspace

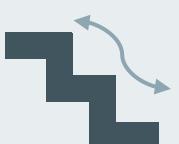
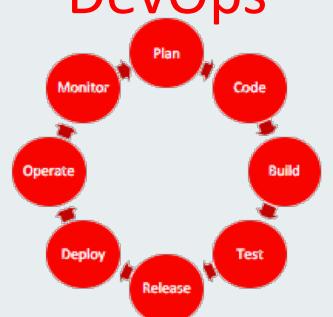
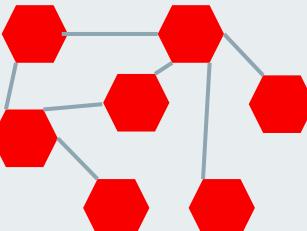
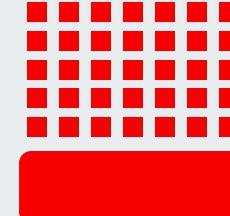
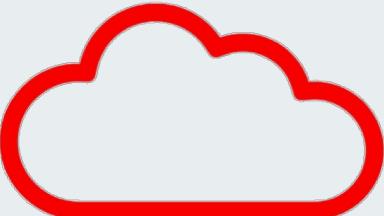


Source: Docker docs and <https://docs.docker.com/glossary/>

Docker Features....

- Light-Weight
 - Minimal overhead (cpu/io/network)
 - Based on Linux containers
 - Decrease storage consumption
 - Uses layered filesystem to save space (AUFS/LVM)
- Portable
 - Run it Everywhere! - Linux, Mac OS or Windows operating system that has Docker installed.
 - Raspberry pi support.
 - Move from one environment to another by using the same Docker technology.
- 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.

History and Multi-Dimensional Evolution of Computing

Development Process	Application Architecture	Deployment and Packaging	Application Infrastructure
Waterfall 	Monolithic 	Physical Server 	Datacenter 
Agile 	N-Tier 	Virtual Servers 	Hosted 
DevOps 	Microservices 	Containers 	Cloud 

Why Docker is Hot

Why Docker is Hot – Its simple, Devs love it



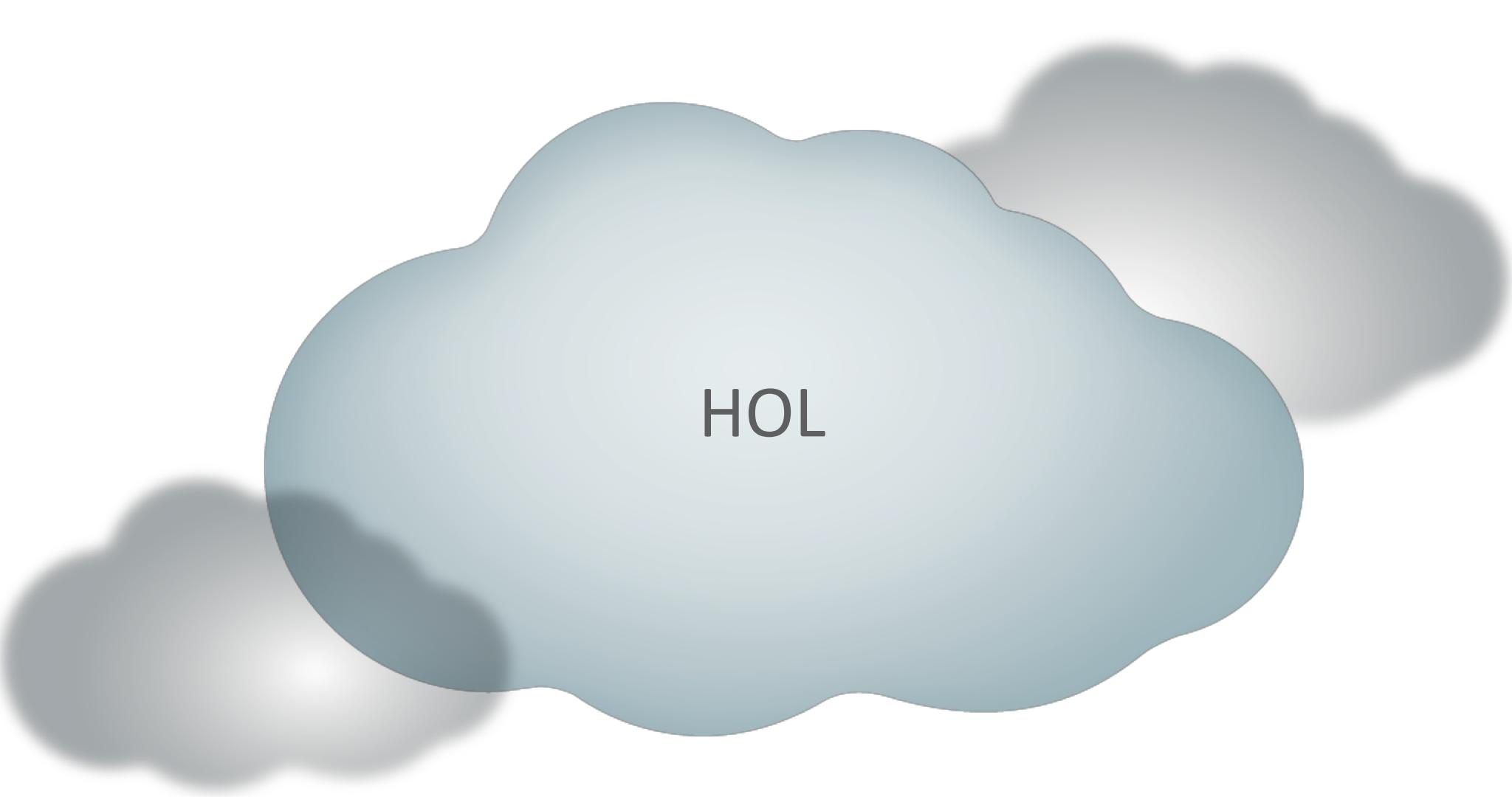
Dev/Test of Legacy Apps

New App Dev (including parts of legacy apps)

Code Agility, CI/CD Pipeline, DevOps

Adoption of Open Source

Microservices & Cloud Native Apps



HOL

Oracle Cloud and Docker Containers

Compute CS

Docker logo: A blue whale carrying a stack of shipping containers.
Kubernetes logo: A blue hexagon with a white steering wheel inside.
MESOS logo: A blue hexagonal grid pattern.
Docker Registry logo: A white cloud containing a stack of colorful shipping containers.

DIY Container Management

Container CS

The dashboard shows the following metrics:

- Deployments: 4 (All healthy)
- Hosts: 2 (All active and reachable)
- Resource Pools: 3 (All healthy)

Services, Stacks, Deployments, and Resource Pools sections show counts of 27, 4, 4, and 3 respectively.

Oracle Managed Container Service

Application Container CS

Java EE logo: A steaming coffee cup.
Java logo: A steaming coffee cup.
Node.js logo: The word "node" in lowercase with a green hexagon above it, followed by ".js".
Redis logo: A red bird.
PHP logo: The letters "php" in a blue oval.
Python logo: The Python logo icon followed by the word "python".

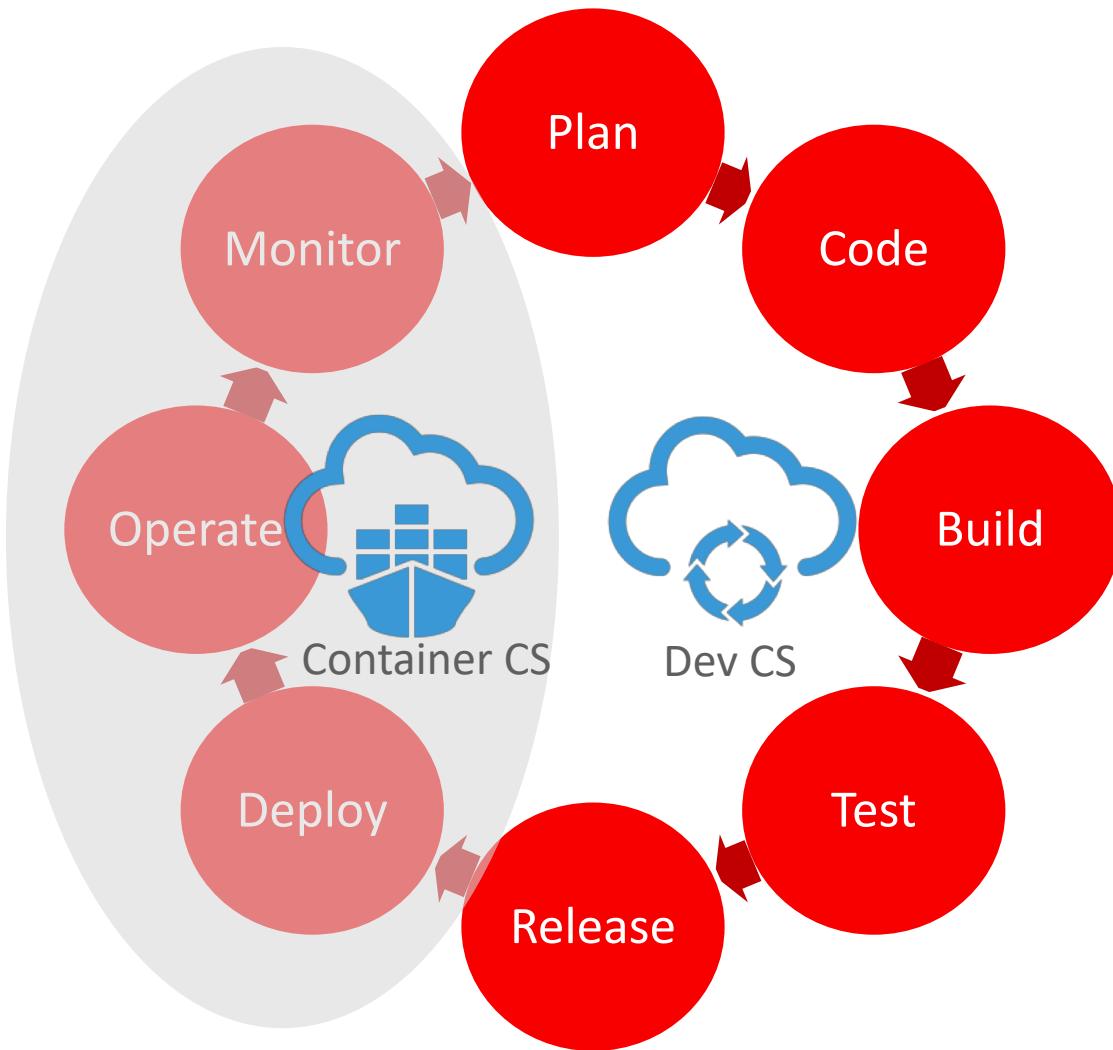
Docker-based Cloud Polyglot Platform

IaaS

CaaS

PaaS

The Docker DevOps Cycle

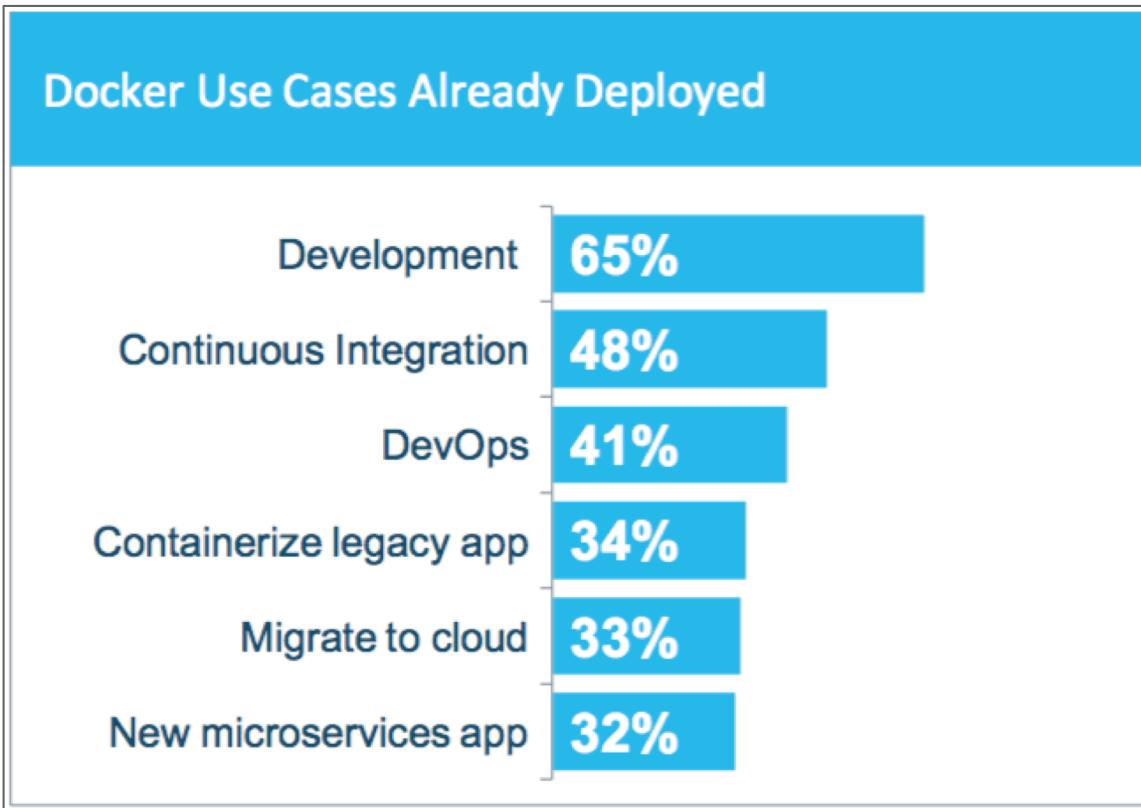


OPC services featuring:

- Oracle Developer Cloud Service
 - Now Includes Docker Image Build Function*
- Oracle Container Cloud Service
 - Deploy, Operate, Monitor

Use Cases

How Containers are Being Used – Survey Says:



- Developer productivity a top use case today
- Building out CI/CD pipelines
 - Consistent container image moves through pipeline
 - Preventing “it worked in dev” syndrome
- Application modernization and portability are also key adoption drivers (Prem <-> cloud)

SOURCE: THE EVOLUTION OF THE MODERN SOFTWARE SUPPLY CHAIN, DOCKER SURVEY 2016

Additional Resources

Resource	Location
Entry Level Hands-on Lab	<u>https://github.com/oracle/cloud-native-devops-workshop/tree/master/containers/docker001</u>
Oracle Container Cloud Service	<u>https://cloud.oracle.com/en_US/container</u>
Official Image Registries	<u>Oracle Images on the Docker Store</u> <u>Oracle Container Registry</u>



Questions?