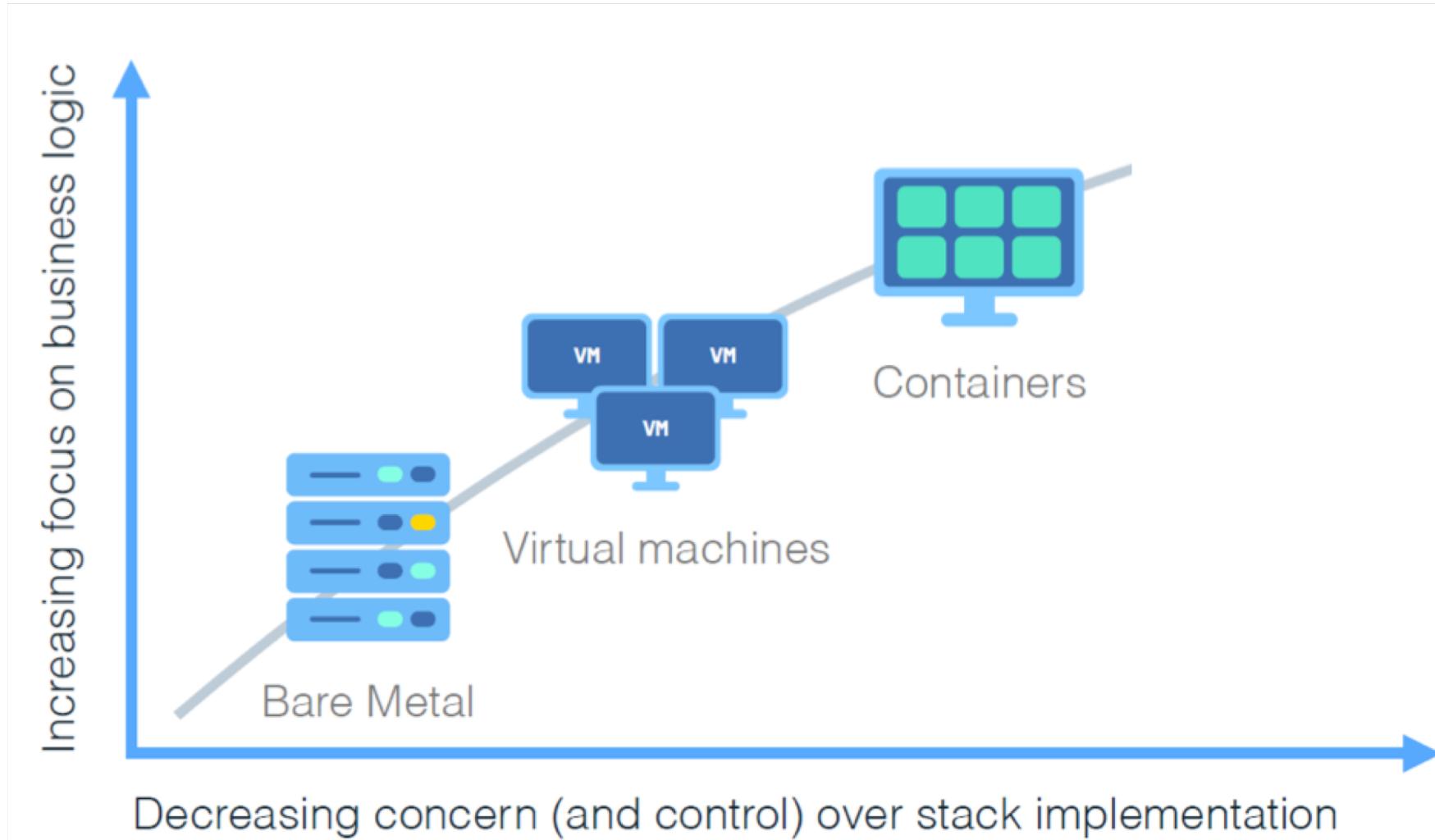




Quick overview of
Container

Evolution Of Containers



The challenge

Multiplicity
of Stacks

Static website:

- Nginx
- OpenSSL
- Bootstrap 2
- ModSecurity

User DB:

- PostgreSQL
- pgv8
- v8

Web front end:

- Ruby
- Rails
- Sass
- Unicorn

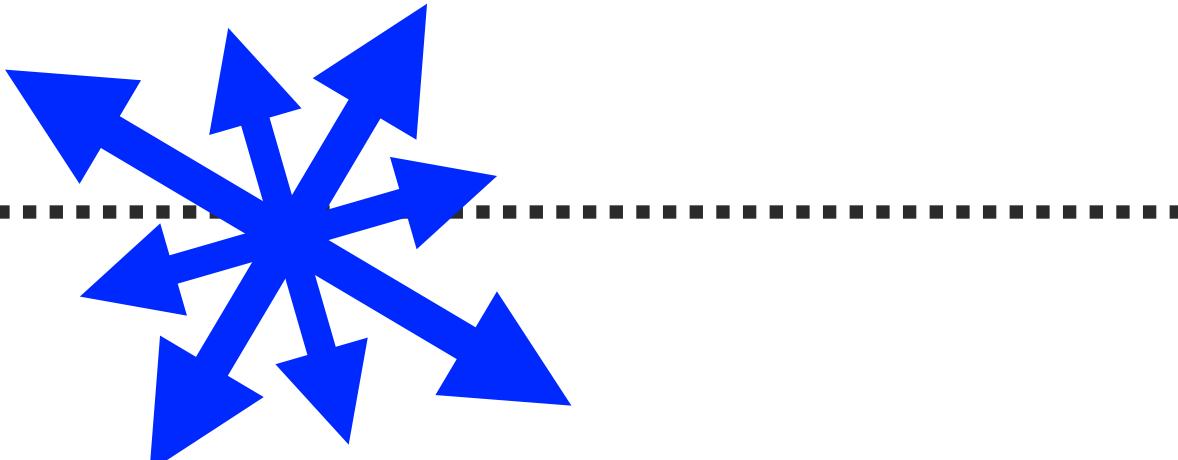
Queue:

- Redis
- Redis-sentinel

Analytics DB:

- Hadoop
- Hive
- Thrift
- OpenJDK

Do services
and apps interact
appropriately?



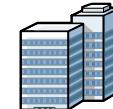
Multiplicity of
hardware
environments



Development
VM



QA server



Customer
Data Center



Public Cloud



Production
Cluster



Contributor's
laptop

Can I migrate
smoothly and
quickly?

Docker: A shipping container for code

Multiplicity
of Stacks

Static website

User DB

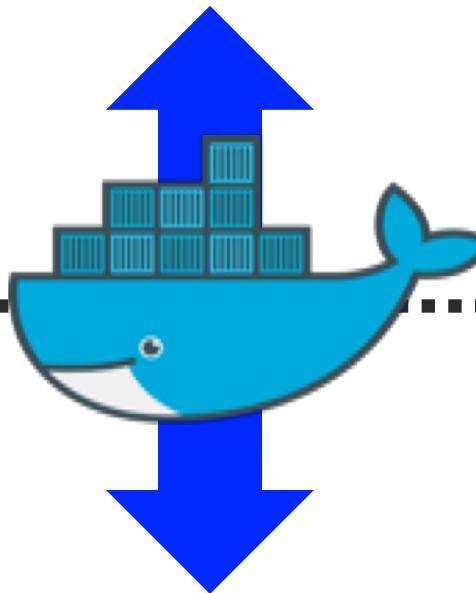
Web front end

Queue

Analytics DB

Do services
and apps interact
appropriately?

An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container...



Multiplicity of
hardware
environments



Development
VM



QA server



Customer
Data Center



Public Cloud



Production
Cluster



Contributor's
laptop

Can I migrate
smoothly and
quickly?

What are Containers?

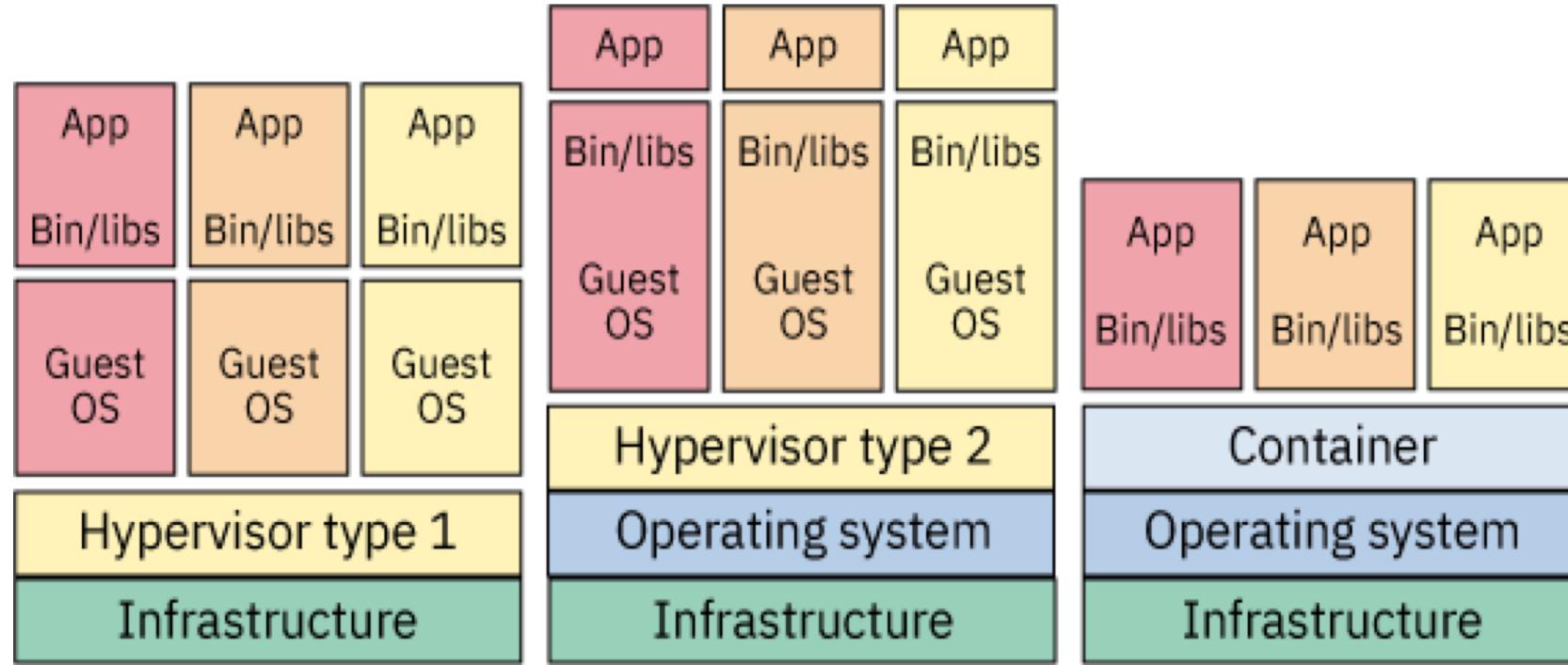
A group of processes run in isolation

- Similar to VMs but managed at the process level
- All processes MUST be able to run on the shared kernel

Each container has its own set of "namespaces" (isolated view)

- **PID** - process IDs
- **USER** - user and group IDs
- **UTS** - hostname and domain name
- **NS** - mount points
- **NET** - Network devices, stacks, ports
- **IPC** - inter-process communications, message queues
- **cgroups** - controls limits and monitoring of resources

VM vs Container



Each VM has its own
OS

Containers share
the same base
Kernel

App, bins/libs/OS must all
be runnable on the shared
kernel

If OS files aren't needed
they can be excluded.

Why Containers?

Fast startup time - only takes milliseconds to:

- Create a new directory
- Lay-down the container's filesystem
- Setup the networks, mounts, ...
- Start the process

Better resource utilization

- Can fit far more containers

	CONTAINER BENEFITS	VIRTUAL MACHINE BENEFITS
Consistent Runtime Environment	✓	✓
Application Sandboxing	✓	✓
Small Size on Disk	✓	
Low Overhead	✓	

Docker :

Why we need Docker

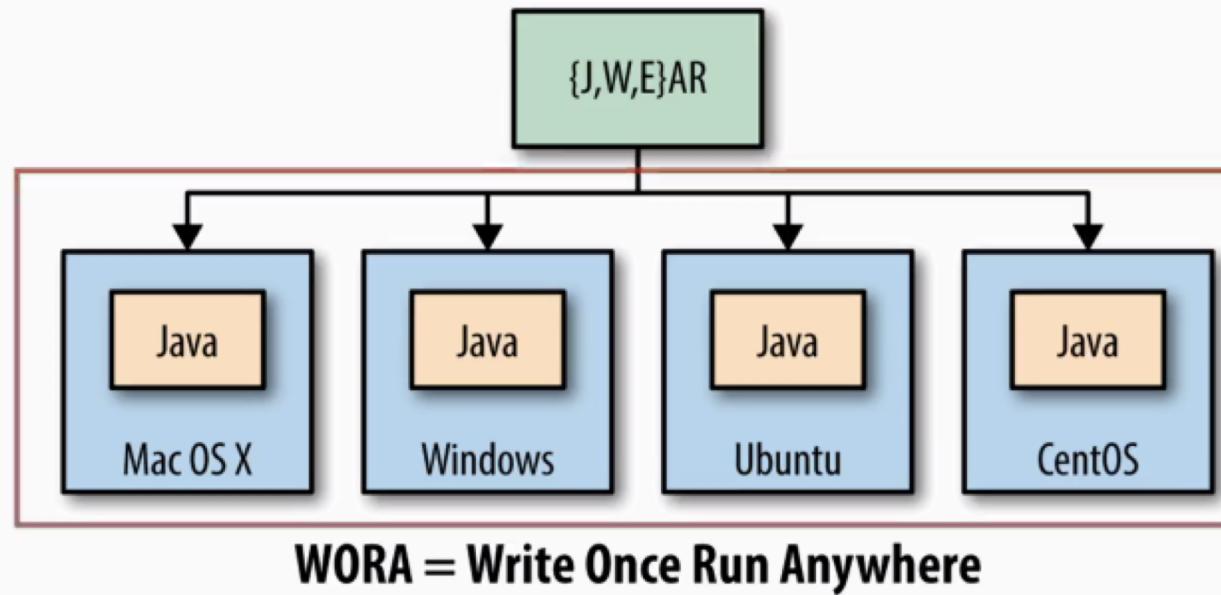
Package Once and Deploy Anywhere

Docker Concepts

Containerization

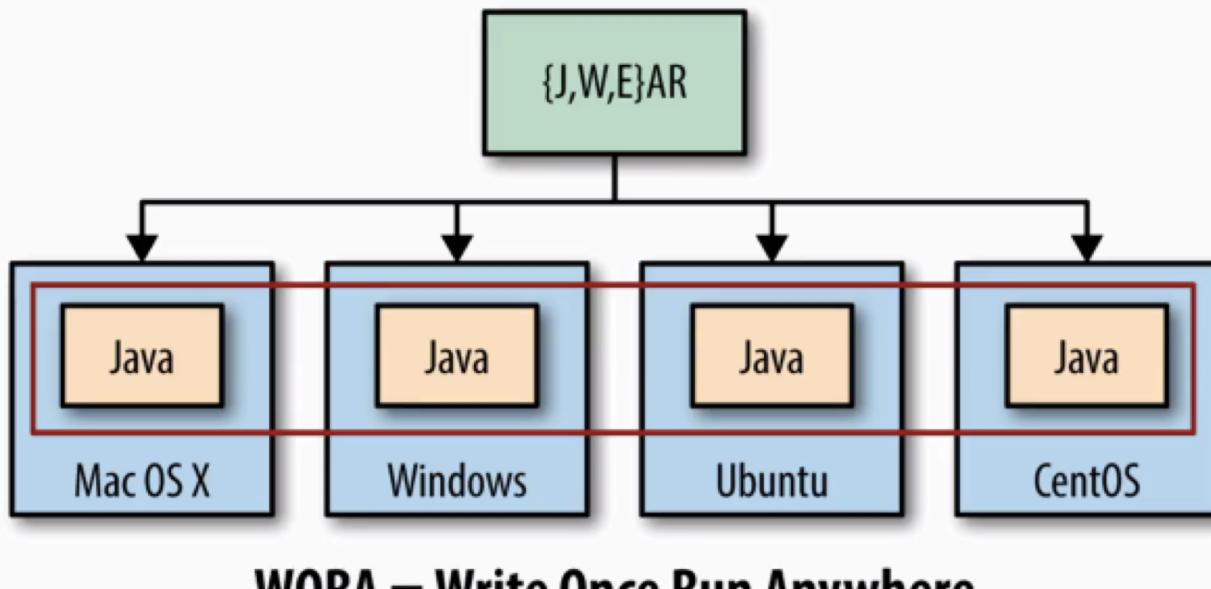
Isolation

Java Application



- Each OS has its Java Software
- With same version of Software

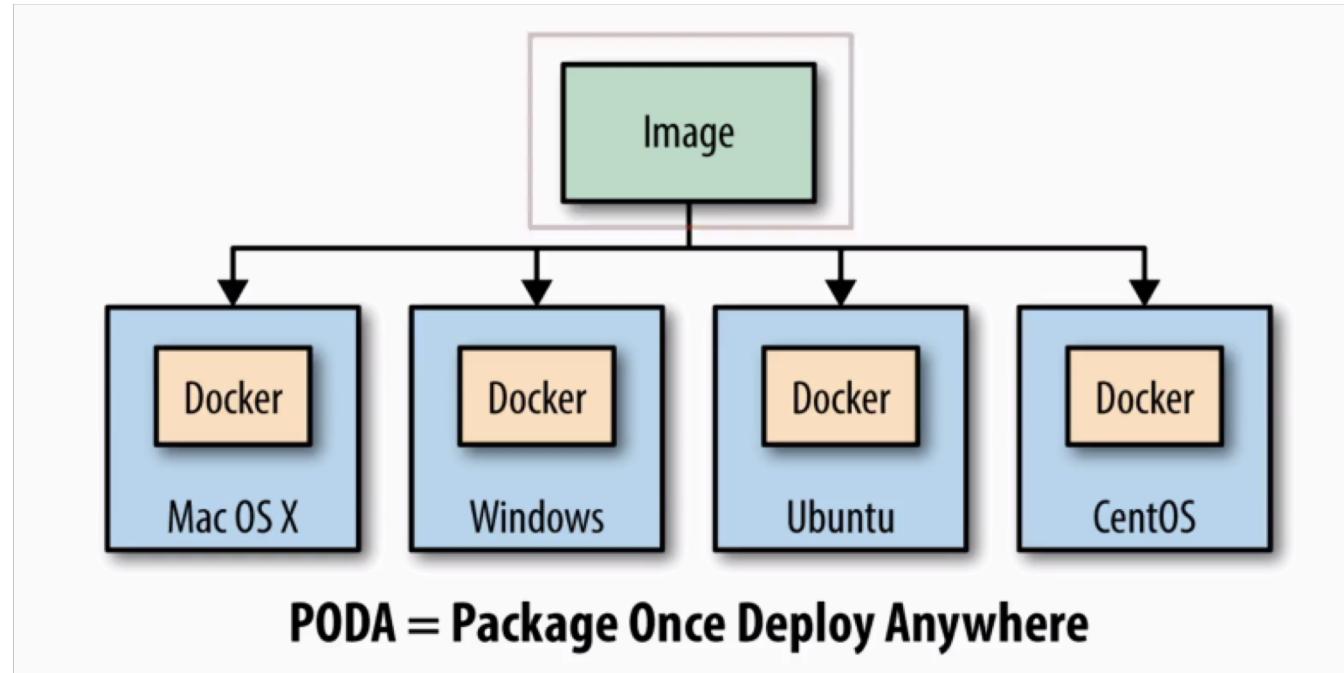
Dependencies



Java installed
(Proper version)

—

One Step Ahead : Docker



- No installation of software
- Docker Image using Docker Platform
- (so one step ahead)

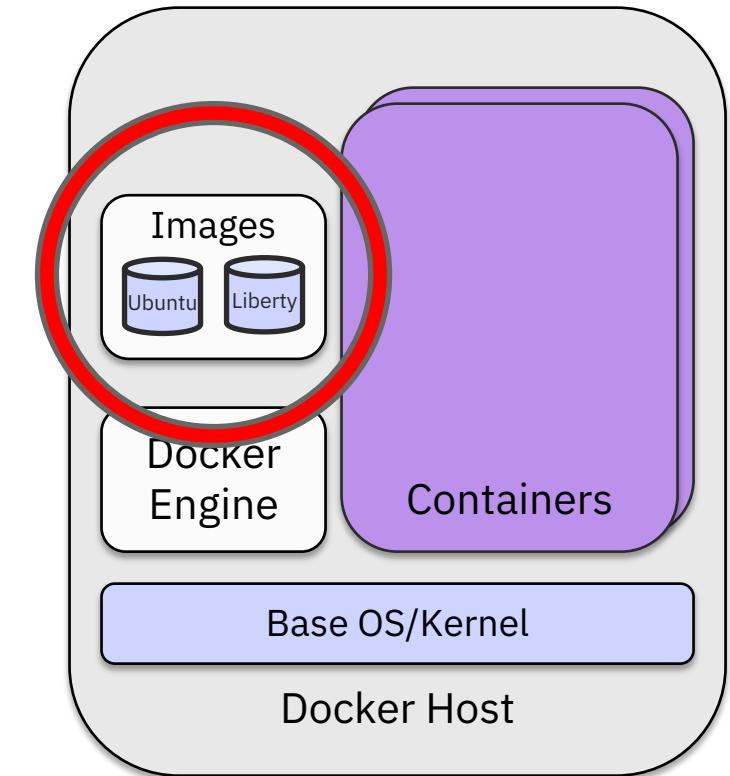
Docker Images

Tar file containing a container's filesystem + metadata

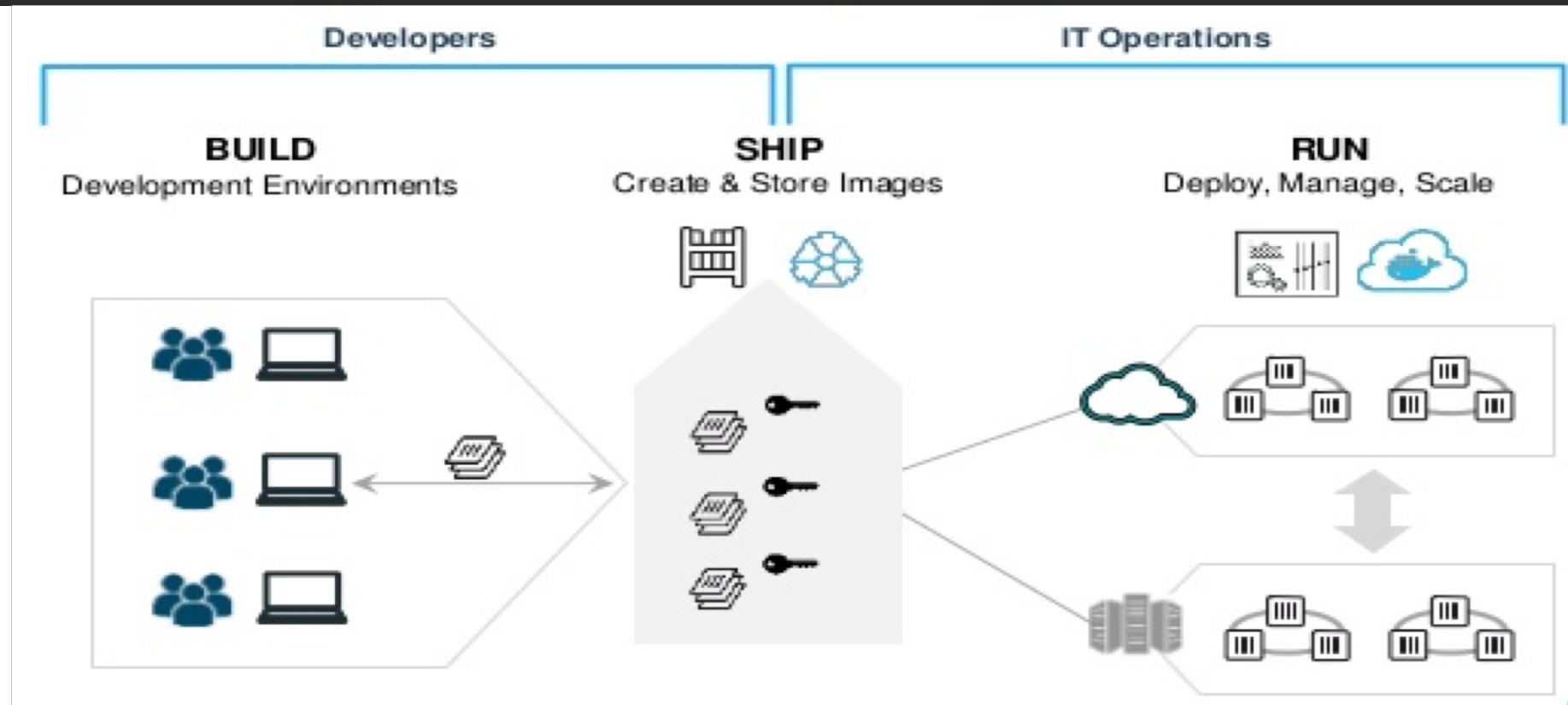
For sharing and redistribution

- Global/public registry for sharing: DockerHub

Similar, in concept, to a VM image

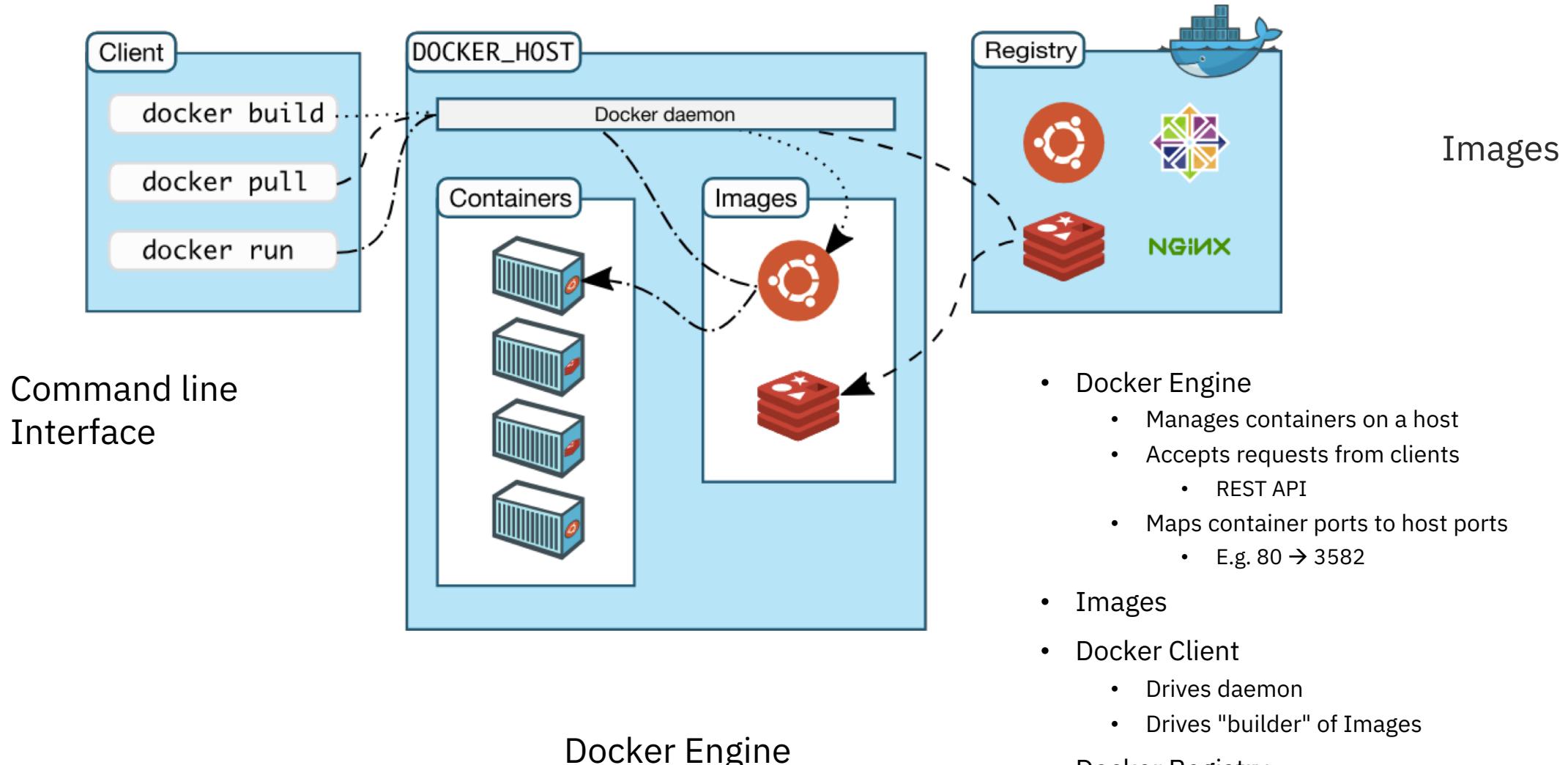


Phases : Build , Ship, Run



- Tools to create containerized application
- Package App – dependencies, infra as template – This is called Image
- Share Apps in Secure and collaboration manner
- Docker images stored, shared & managed in **Docker Registry**
- **Docker Hub(Public)** default registry for all images
- Deploy, manage and scale apps
- Container is runtime representation of image
- Lifecycle of container – run, start, scale, stop, move, delete

Docker Architecture



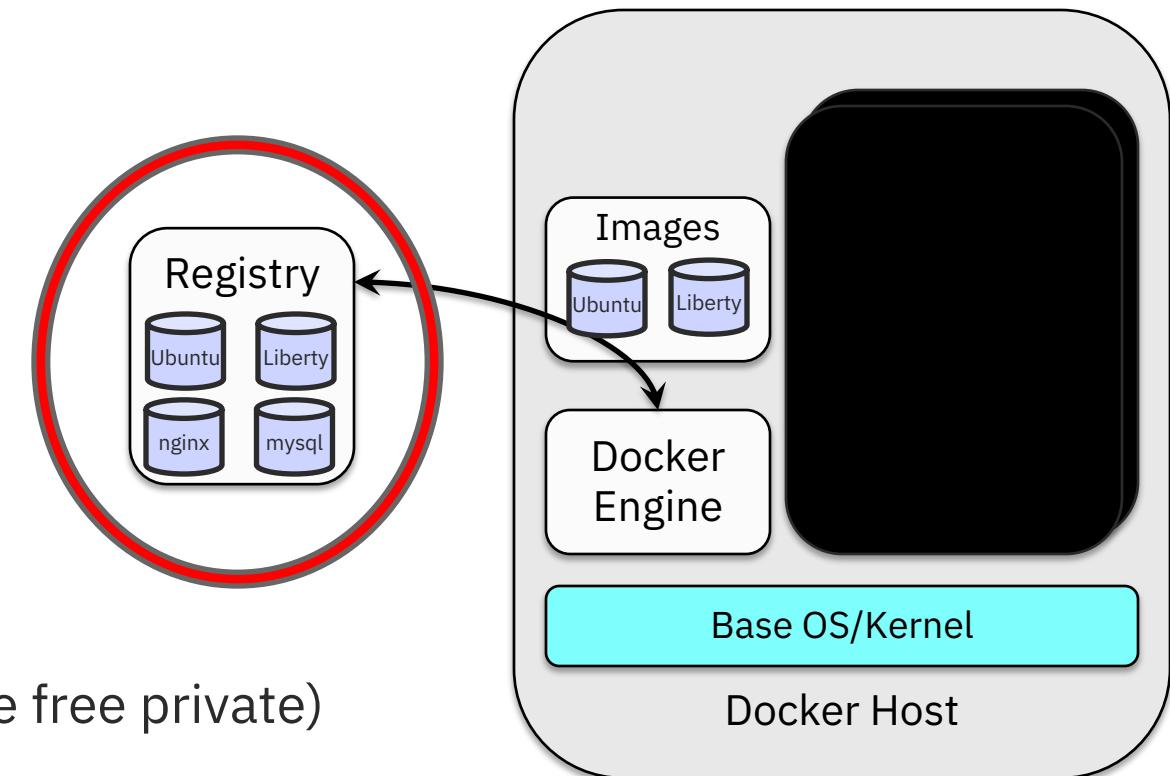
Docker Registry

Creating and using images is only part of the story

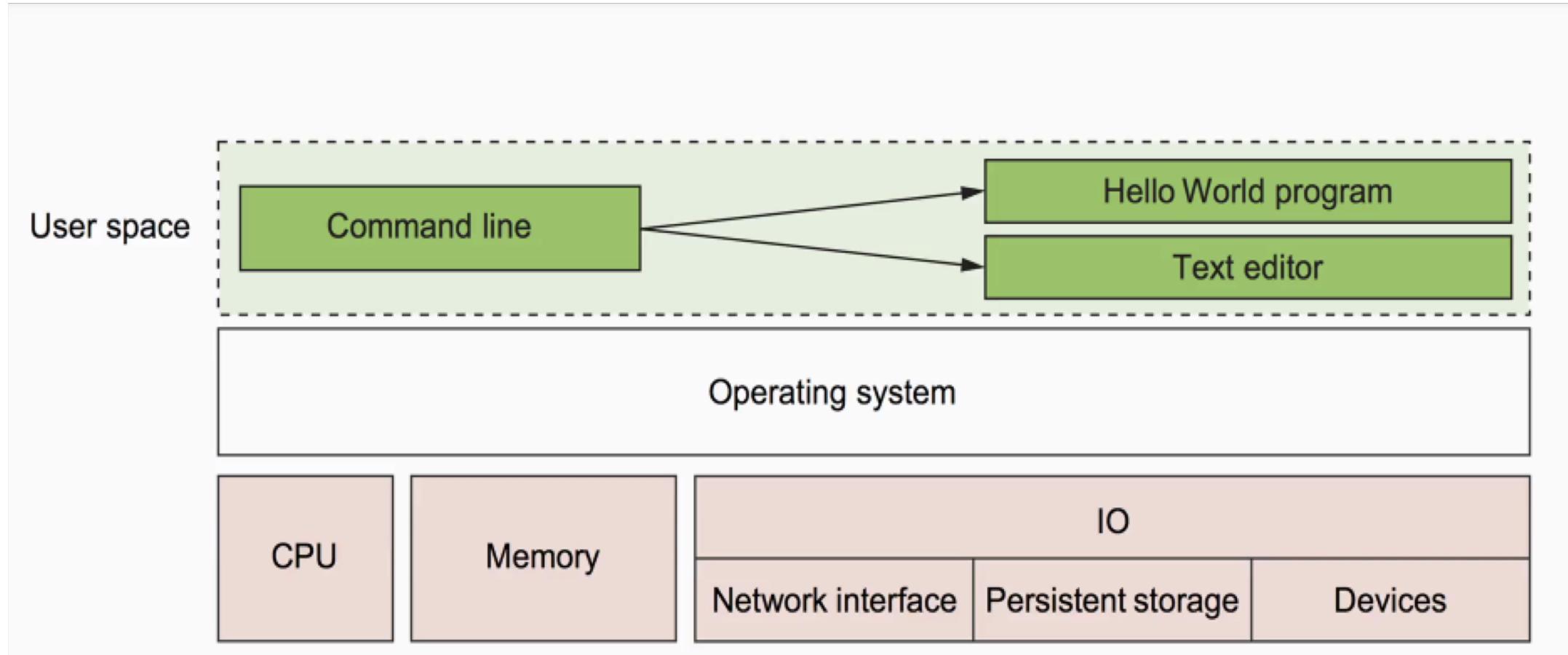
Sharing them is the other

DockerHub - <http://hub.docker.com>

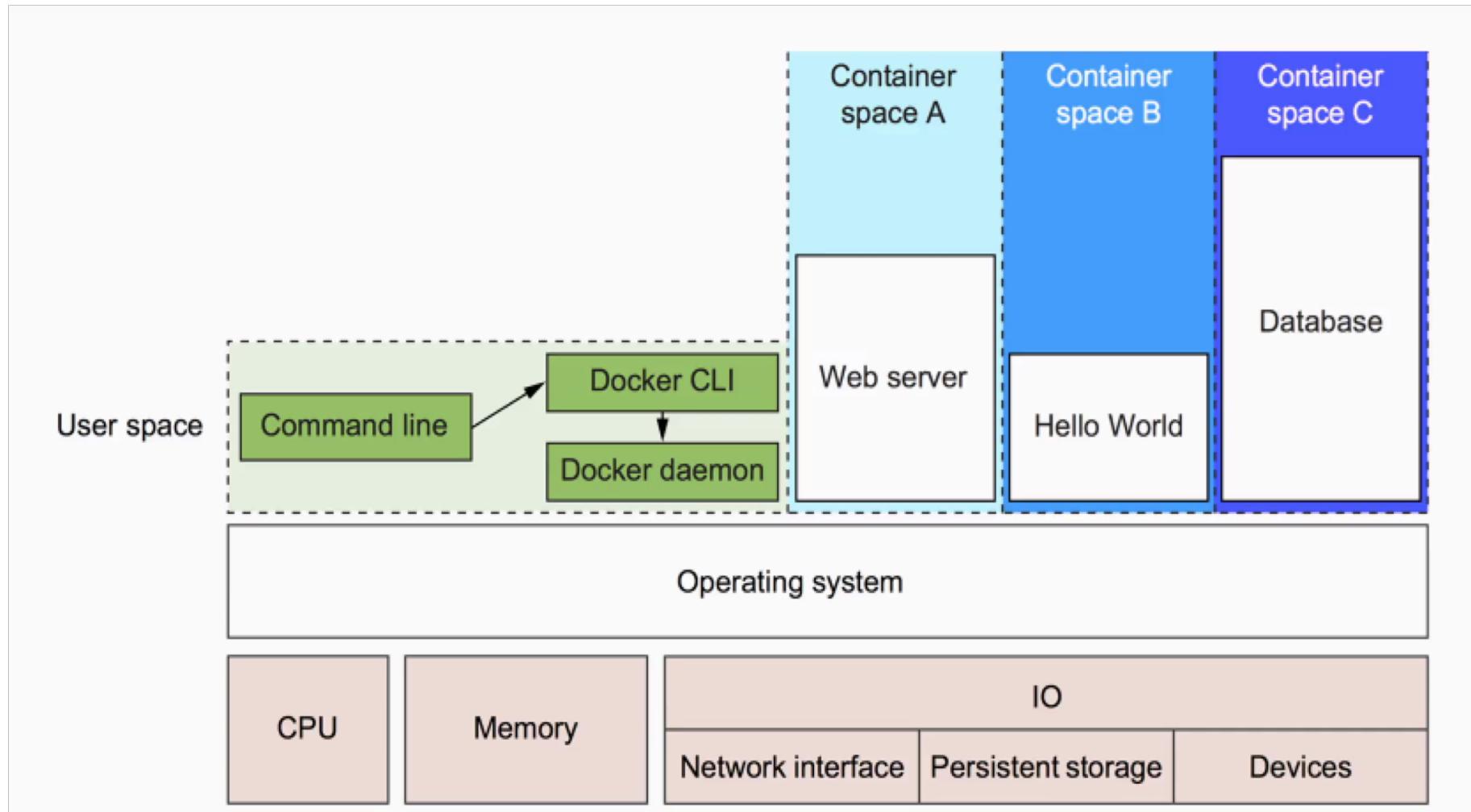
- Public registry of Docker Images
- Hosted by Docker Inc.
- Free for public images, pay for private ones (one free private)
- By default docker engines will look in DockerHub for images
- Web interface for searching, descriptions of images



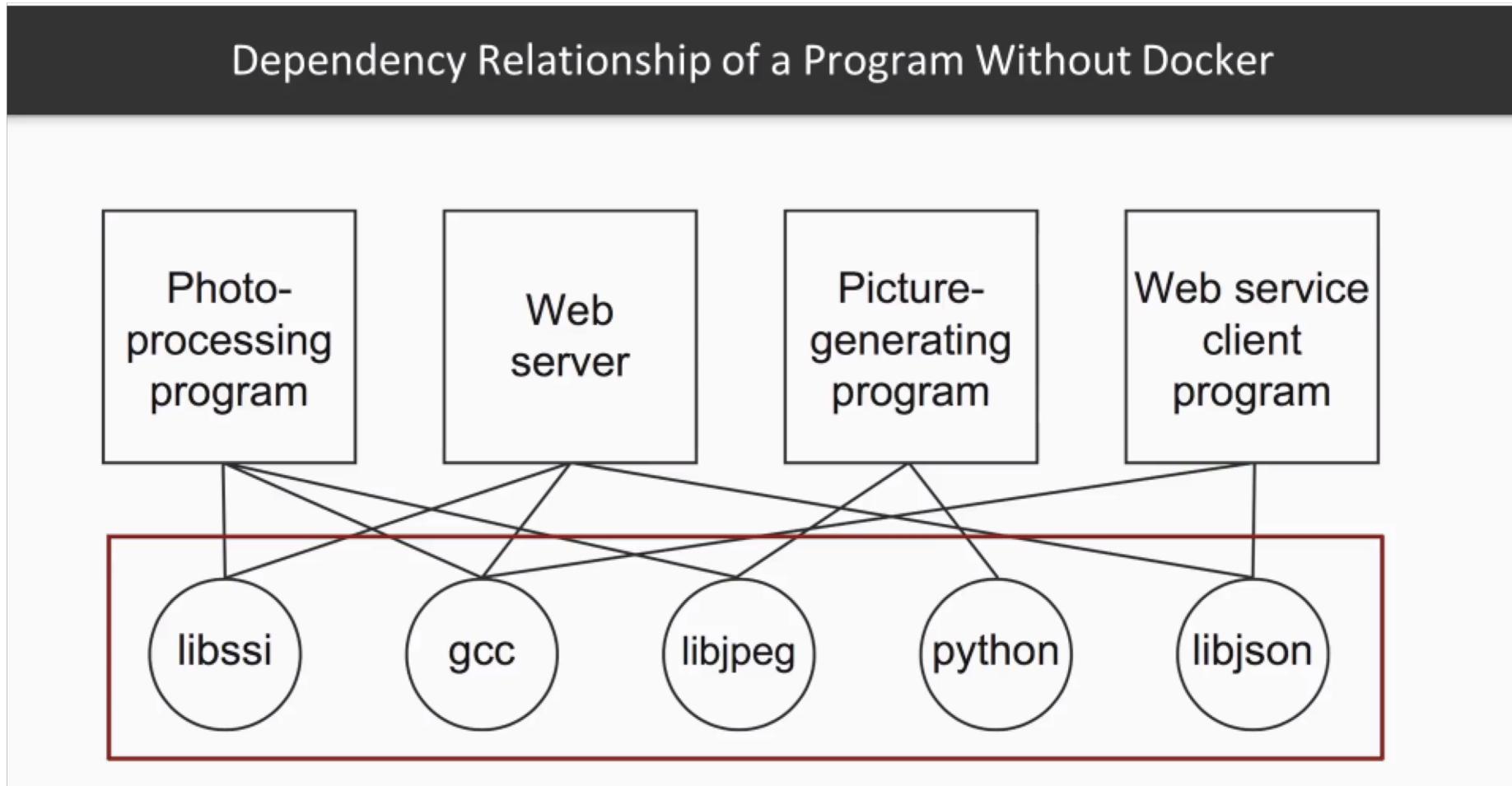
Isolation : Linux



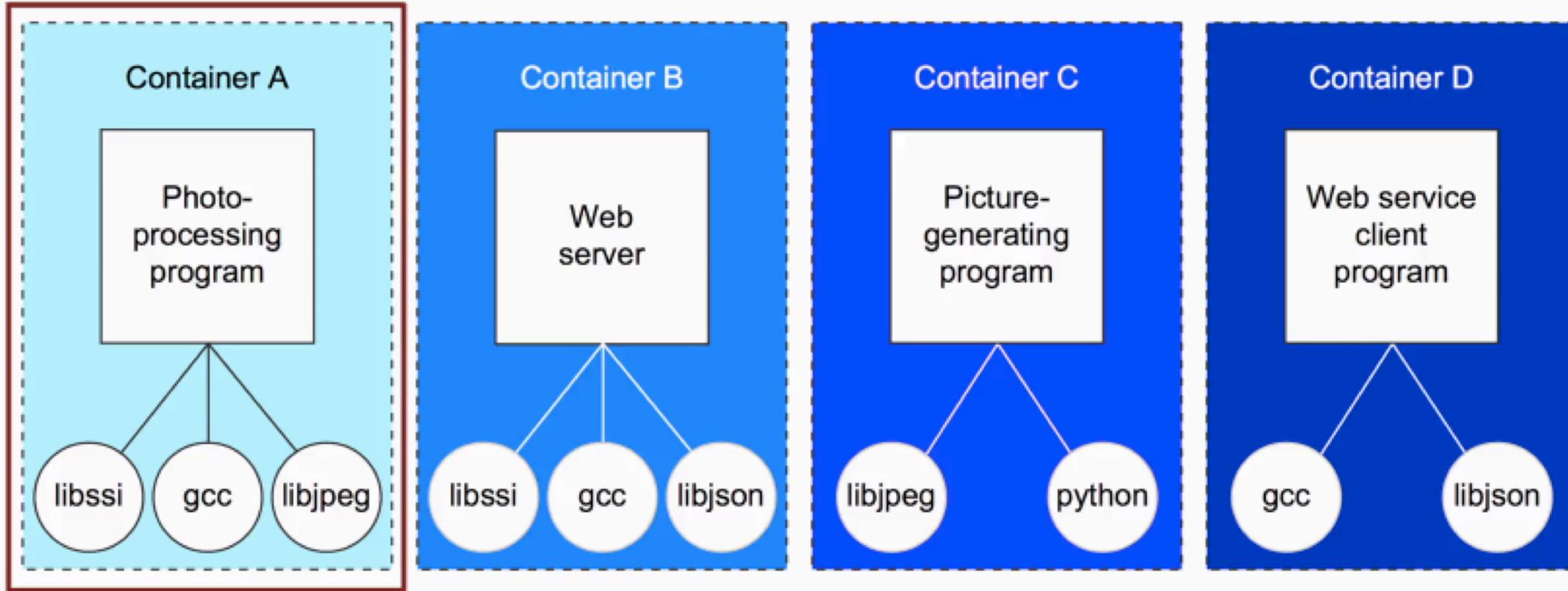
Docker Isolation : Process Level



Running programs without docker



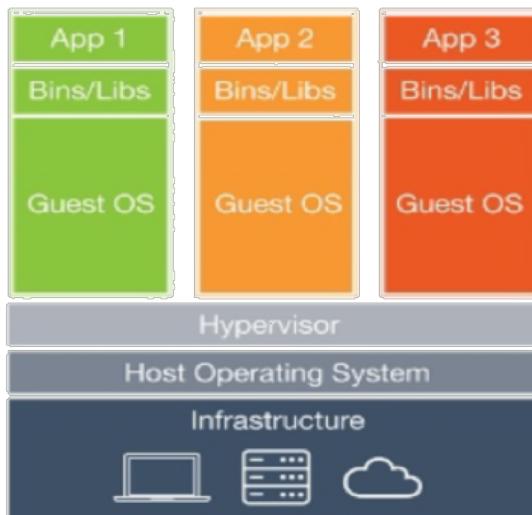
Each App : With needed Dependencies



So : What is Docker?

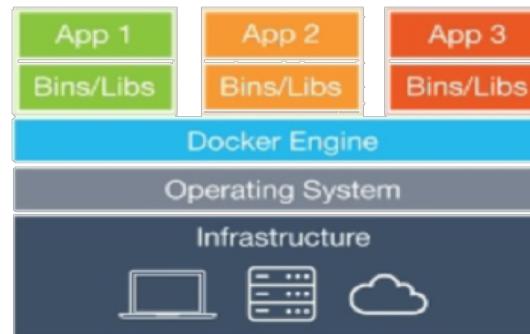
1. At its core, Docker is tooling to manage containers
 - Docker is not a technology, it's a tool or platform
 - Simplified existing technology to enable it for the masses
2. Tooling to manage containers
 - Containers are not new
 - Docker just made them easy to use
3. Docker creates and manages the lifecycle of containers
 - Setup filesystem
 - CRUD container
 - Setup networks
 - Setup volumes / mounts
 - Create: start new process telling OS to run it in isolation

Docker Container Vs VMWare



IBM
CODE

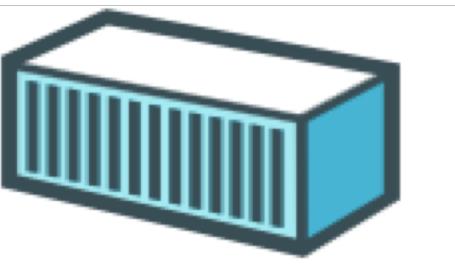
vmware®



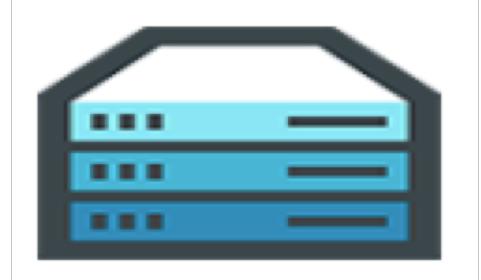
docker



Build

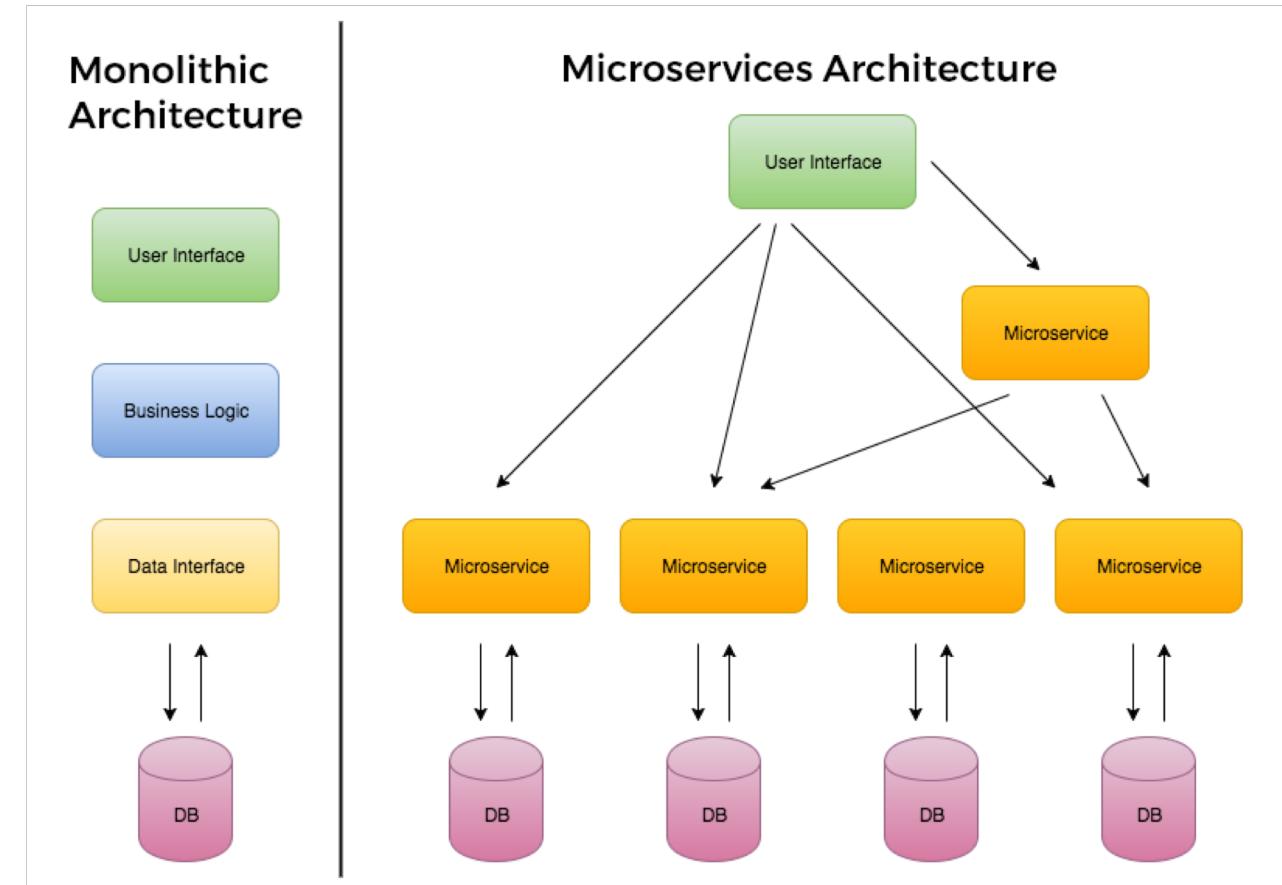
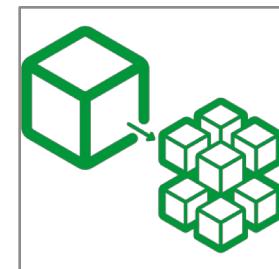


Ship



Run

Microservices & Cloud Native Apps

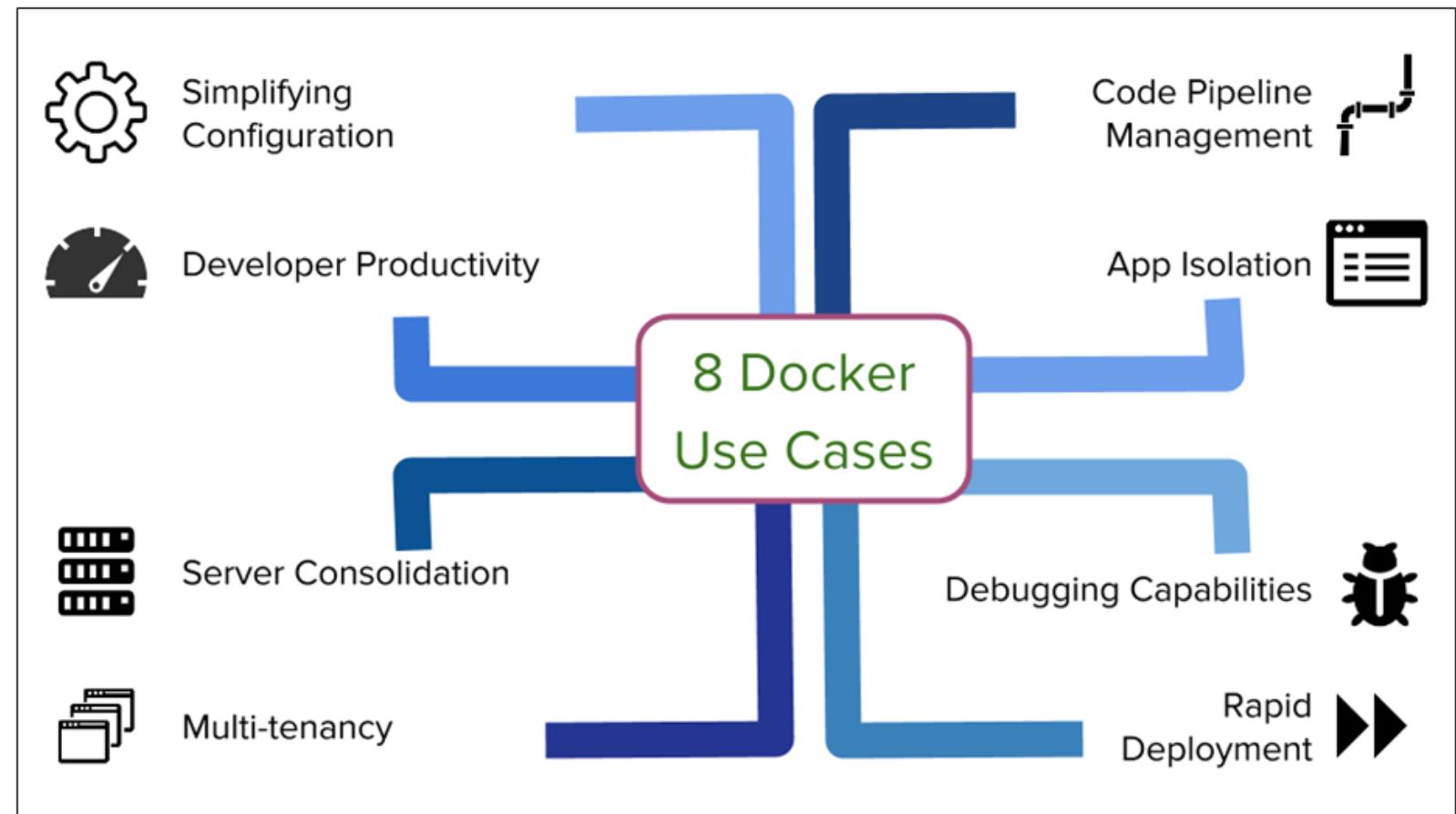


Docker – Use cases

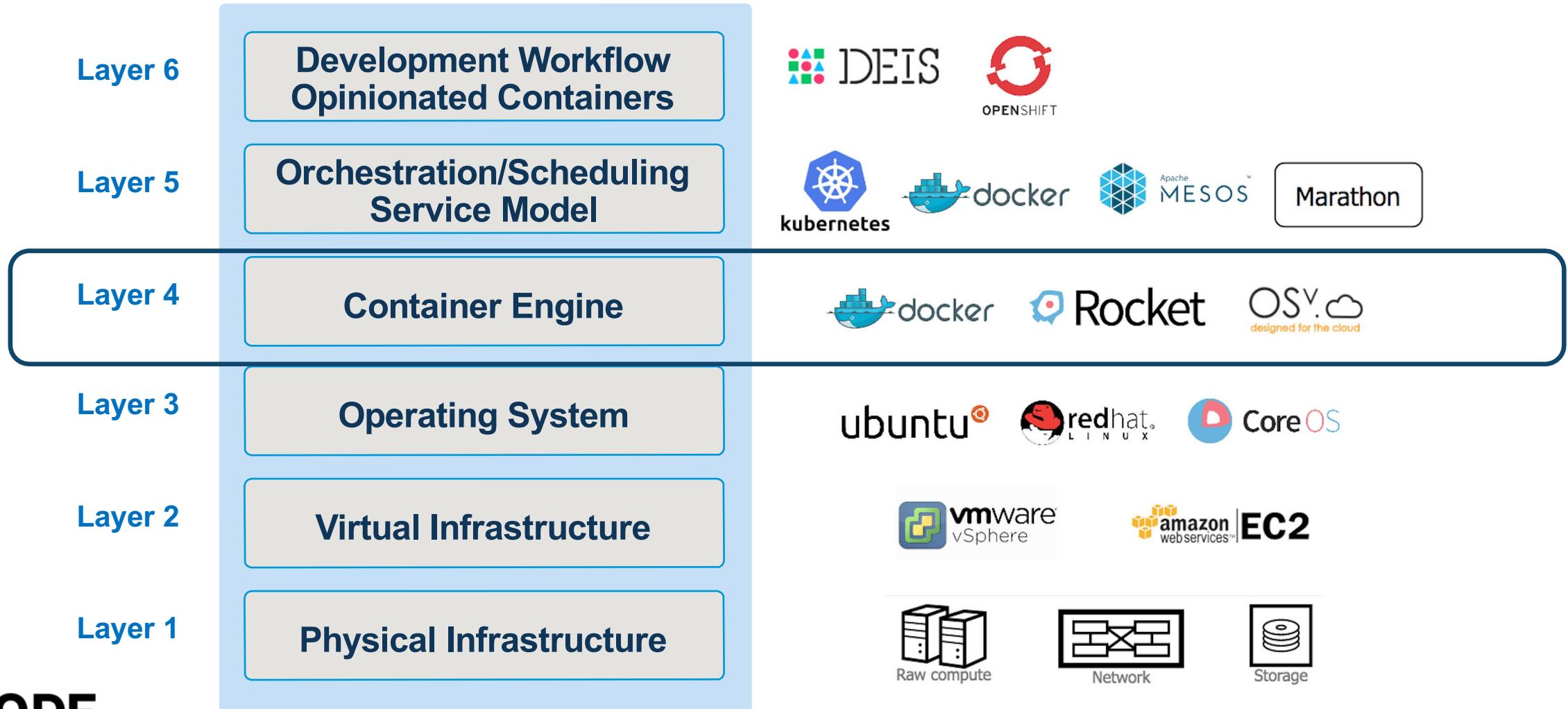
Server Apps like web servers, mail servers, databases, and proxies

Desktop software's, tools , email clients etc

Running these in a container and as a user with reduced privileges will help protect your system from attack



Layers : Containers



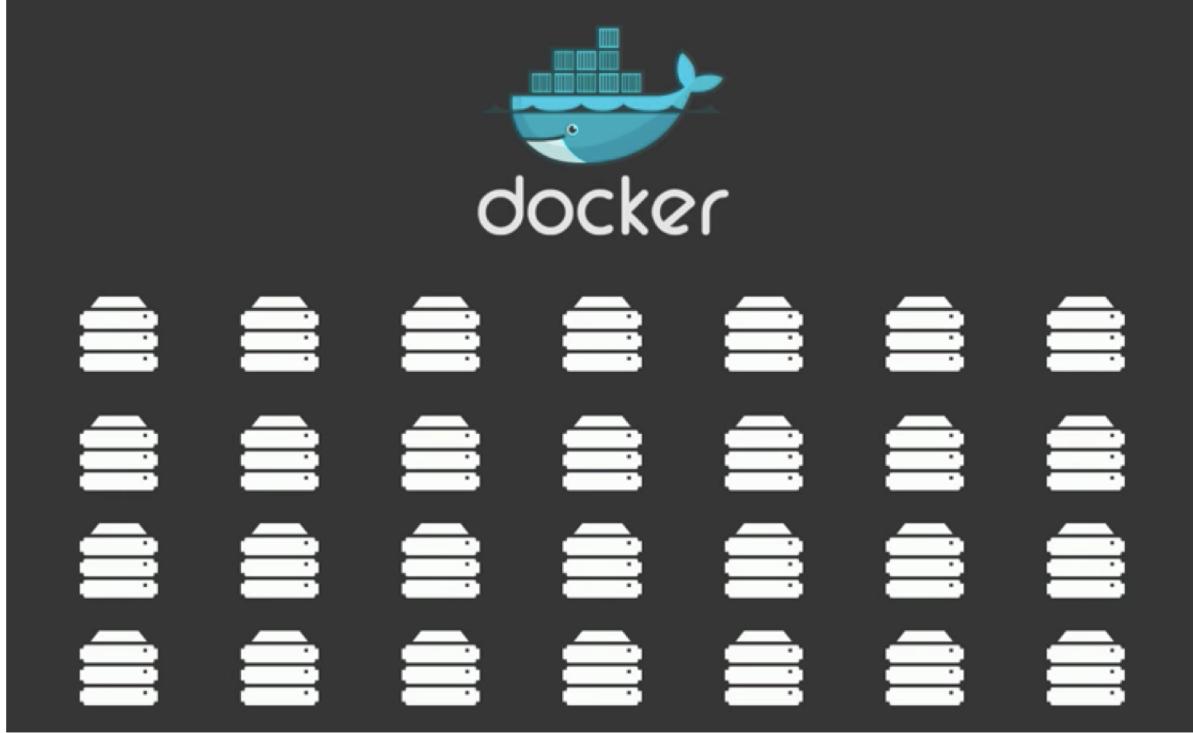
Get Started with IBM Cloud

Create IBM Cloud Account

Deployed on 3 servers – Simple



Jump from 3 – 40,50 servers



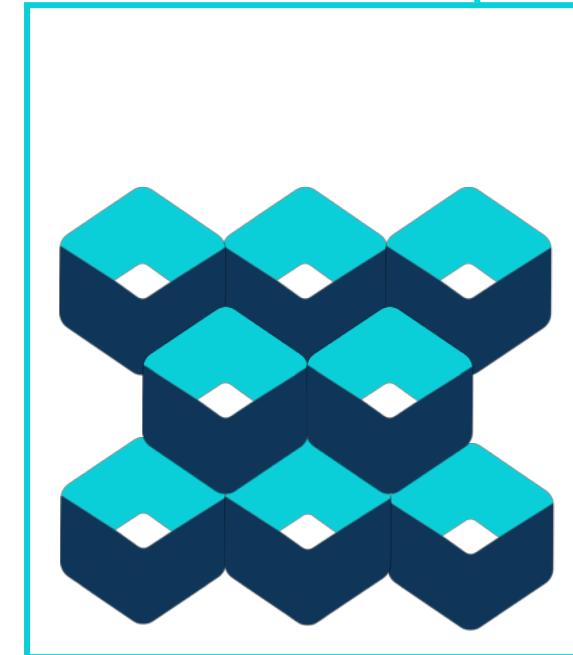
- Scale out
- Keep track
- Where to put your containers
- What container – Where?
- May be containers other than docker as well

Containers are great but ... can lead into lack of control & chaos

Kubernetes – (Κυβερνήτης - Captain in Greek)

Regain control with Containers and Kubernetes

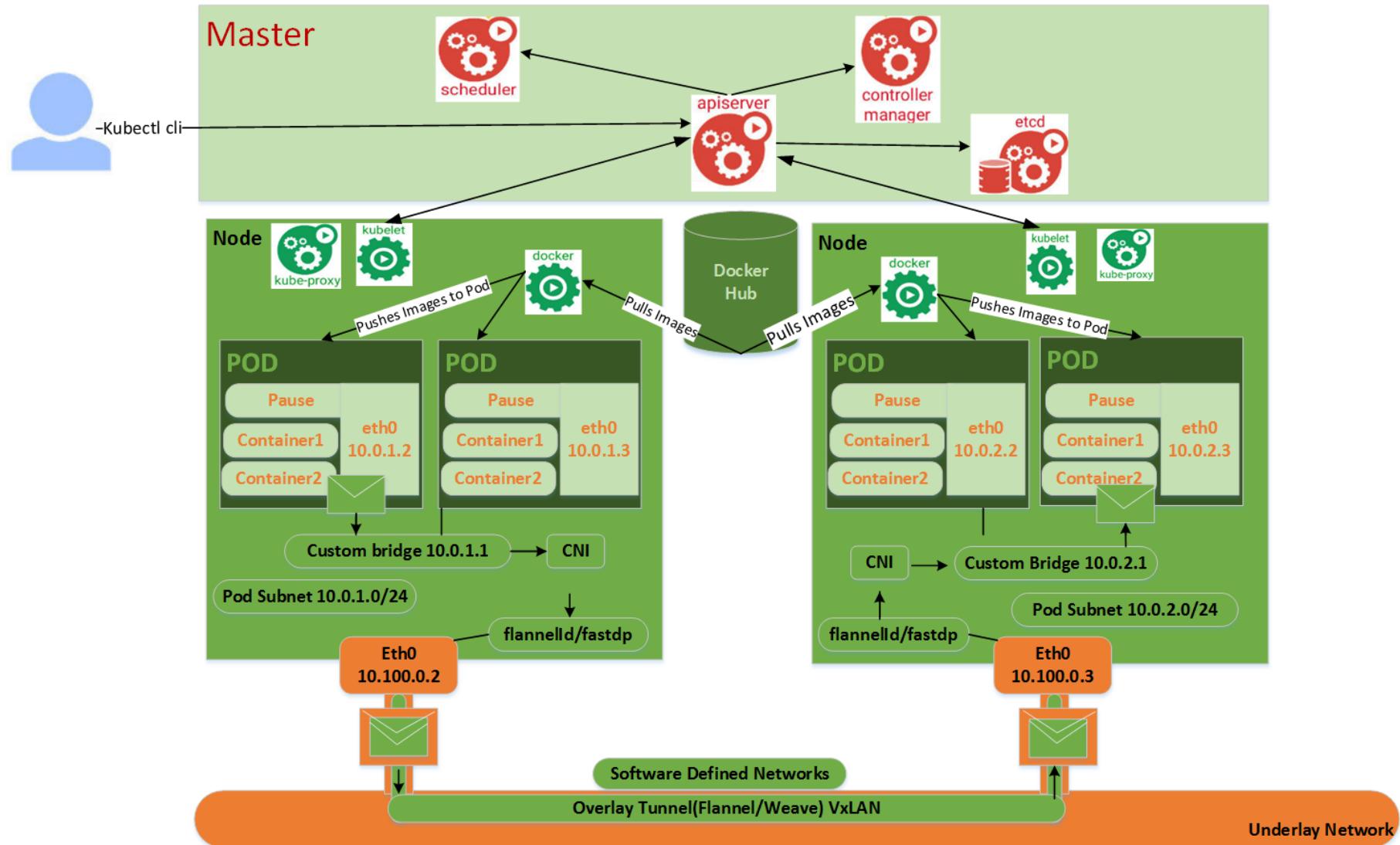
- Organize and Govern the Container Chaos



What is K8s?

- Based on Google's Borg & Omega - Open Source - Container Orchestrator
- Open Governance
 - Cloud Native Compute Foundation
- Adoption by Enterprise
 - RedHat, Microsoft, IBM and Amazon
- Help to automate DevOps – Deployment, scaling and management of containerized apps
- Helps organize container in logical units(pods, nodes)
- Helps in resource monitoring and logging

Kubernetes Architecture





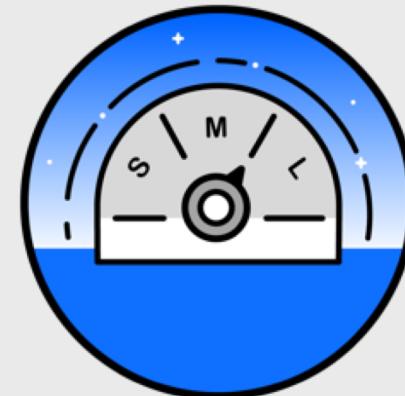
Kubernetes Capabilities



Intelligent Scheduling



Self-healing



Horizontal scaling



Service discovery & load balancing



Automated rollouts and rollbacks



Secret and configuration management



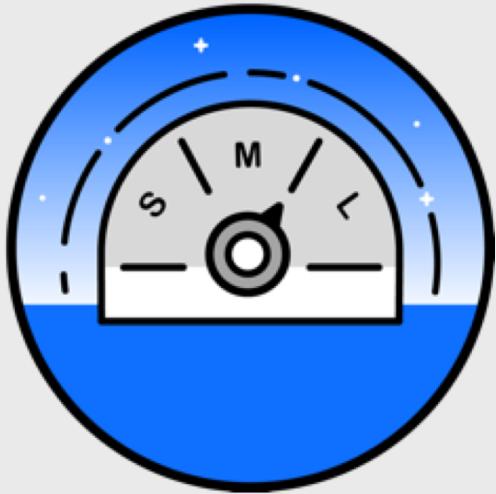
Intelligent Scheduling

- Automatically places containers based on required resources
- Supports mixed workloads to drive increased utilization



Self-healing

- Restarts containers that fail
- Replaces and reschedules containers when nodes die
- Kills containers that don't respond to your user-defined health check



Horizontal scaling

- Scale your application with a simple command
- Automatic scaling based on real-time usage



Service discovery and load balancing



- Simple discovery of services through a single DNS name
- Manage access to container applications through IP address or HTTP route.
- Automatically load balance traffic and route around failure



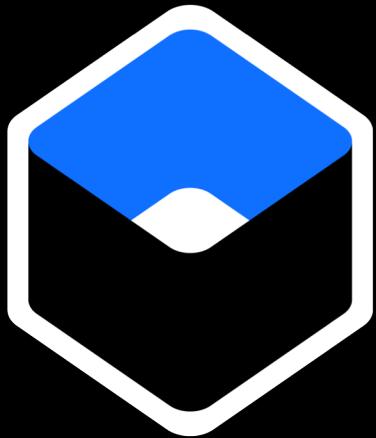
Automated rollouts and rollbacks

- Roll out changes to your application or its configuration, while monitoring application health to ensure things stay up
- If something goes wrong, Kubernetes will rollback the change for you

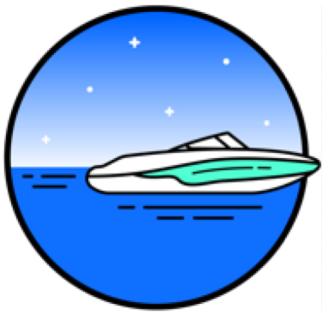


Secret and configuration management

- Safely store application credentials and secrets
- Deploy and update secrets and application configuration without rebuilding your image and without exposing secrets in your stack configuration.



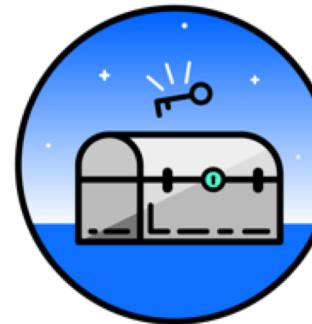
Cluster Management Capabilities



Simplified cluster
management



Design your
own cluster



Container security
& isolation



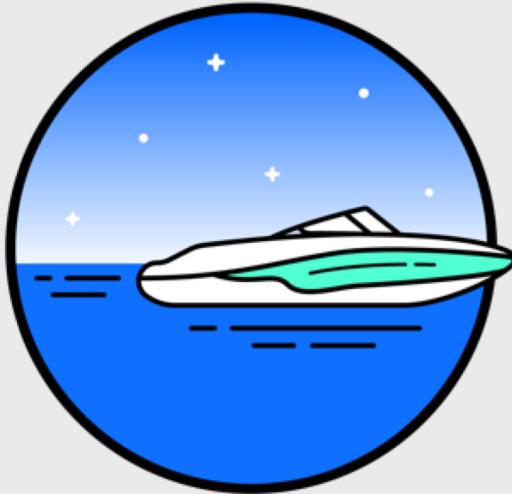
Extend with
IBM Cloud & Watson



Native open-source
experience



Integrated
operational tools



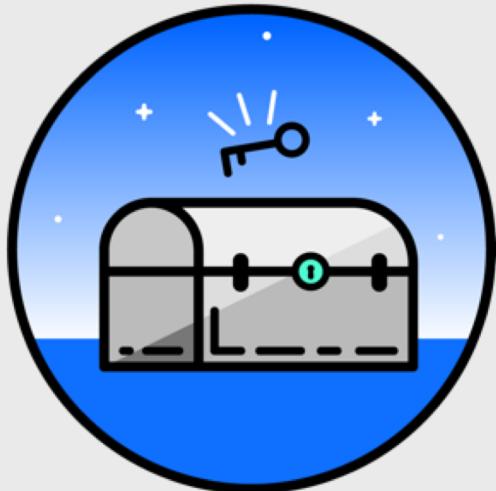
Simplified Cluster Management

- Intuitive graphical user experience
- CLI and API alternatives
- Fully managed master nodes
- Highly available (HA) masters
- User controlled worker node management
- Worker node auto-recovery



Design Your Own Cluster

- Tunable capacity
- Select between shared and dedicated compute using virtual server instances
- Bare metal worker nodes enabling Trusted Compute
- Multizone clusters in IBM Cloud multizone regions and single zone clusters in 25+ datacenters
- Edge nodes
- Configurable networking and storage
- Integrated VPN in-cluster providing IPSec tunnels



Container Security & Isolation

- Isolated compute, networking, and storage
- Automatic encryption of secrets and volumes
- Customer managed keys using HSM backed IBM Key Protect
- Default LUKS encryption of /var/lib/docker
 - Every worker node in each cluster has a unique encryption key
- Store your images securely in your hosted private registry
- Vulnerability Advisor provides Docker image and running container scanning to detect vulnerabilities and configuration weaknesses
- Image signing by integrating with Docker Notary
- Image security deployment enforcement controls



Extend IBM Cloud Services

- Enhance your application with Watson, IoT, Analytics and Data Services
- Persistent volumes using IBM Cloud storage (file, block, object)
- IP and application Load Balancing
- Integrated with IBM Cloud identity and access management
- Control access and billing using Resource Groups



Native Kubernetes Experience

- Seamless experience moving from local development to IBM Cloud
- 100% Kubernetes API and tools
- Certified Kubernetes provider
- Conformance tested for Kubernetes 1.9, 1.10, 1.11
- Supports Kubernetes dashboard
- Leverage Docker images



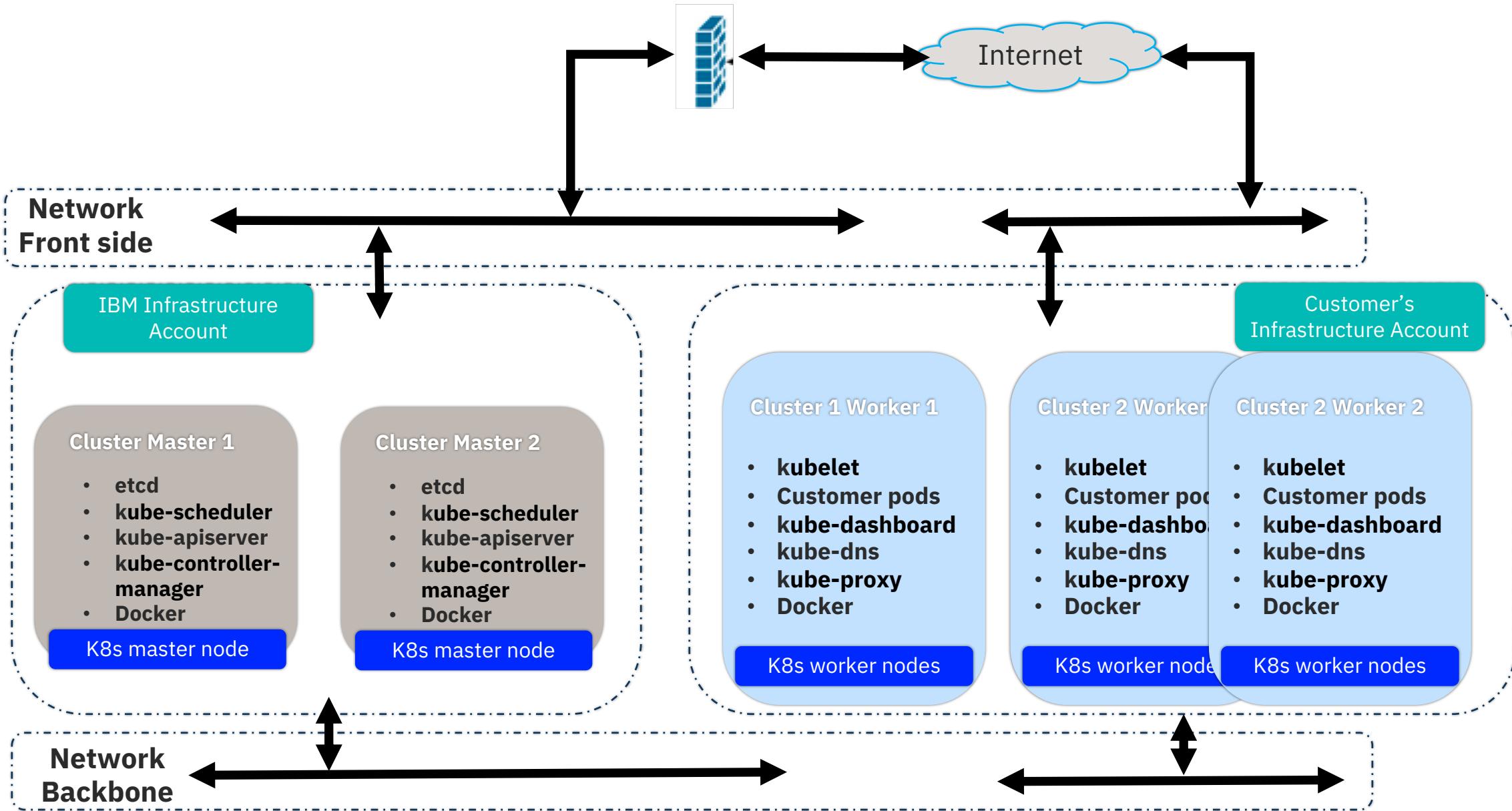
IBM Cloud
Kubernetes Service



Integrated Operational Tools

- Built-in log and metrics collection with IBM Cloud log and monitoring services
- Use with IBM DevOps tools such as Delivery Pipeline
- Supports popular add-ons including Prometheus, Weave, Sysdig, fluentd and others

IBM Cloud Kubernetes Service



In-cluster Capabilities

Create and Manage Kubernetes Clusters

The screenshot displays the IBM Cloud Kubernetes Service interface across four panels:

- Clusters Overview:** Shows a list of existing clusters (iccs_cluster1, iccs_cluster2, neuvector_ibm) with details like state, creation date, location, and Kube Version. A "Create Cluster" button is visible.
- Terminal Session:** A command-line interface (CLI) session showing cluster listing and detailed cluster information (cj).
- Cluster Overview:** A detailed view of the iccs_cluster1 cluster, including its summary, worker nodes (3 Ready), and logs.
- Workers Overview:** A table showing the status of individual workers (kube-dal10-w1, w2, w3) with their public and private IP addresses, machine type, and status.

Terminal Session Details:

```
bx cs clusters
OK
Name          ID
cj            337c42379cd044fb82070462bdb09215
istio2        33ec4642167541ae9b902cb20dc6f8a7
istiotest     2545b88130f7415cb8f2b0b8e90f97d7
mycluster     79f5a344558a456e988dc8f1406b20f4
trusted       395079b801364d7e83918ea94c4fd7ca
State         Created           Workers
normal        2017-05-03T12:28:46+0000   3
normal        2017-05-10T18:20:45+0000   3
normal        2017-05-05T04:15:07+0000   2
normal        2017-05-11T02:48:19+0000   1
normal        2017-04-11T00:37:55+0000   3
```

```
bx cs cluster-get cj
Retrieving cluster cj...
OK
Name:          cj
ID:            337c42379cd044fb82070462bdb09215
Created:       2017-05-03T12:28:46+0000
State:         normal
Master URL:   https://169.47.234.18:10317
Ingress host: cj.us-south.containers.mybluemix.net
Ingress secret: cj
Workers:       3
```

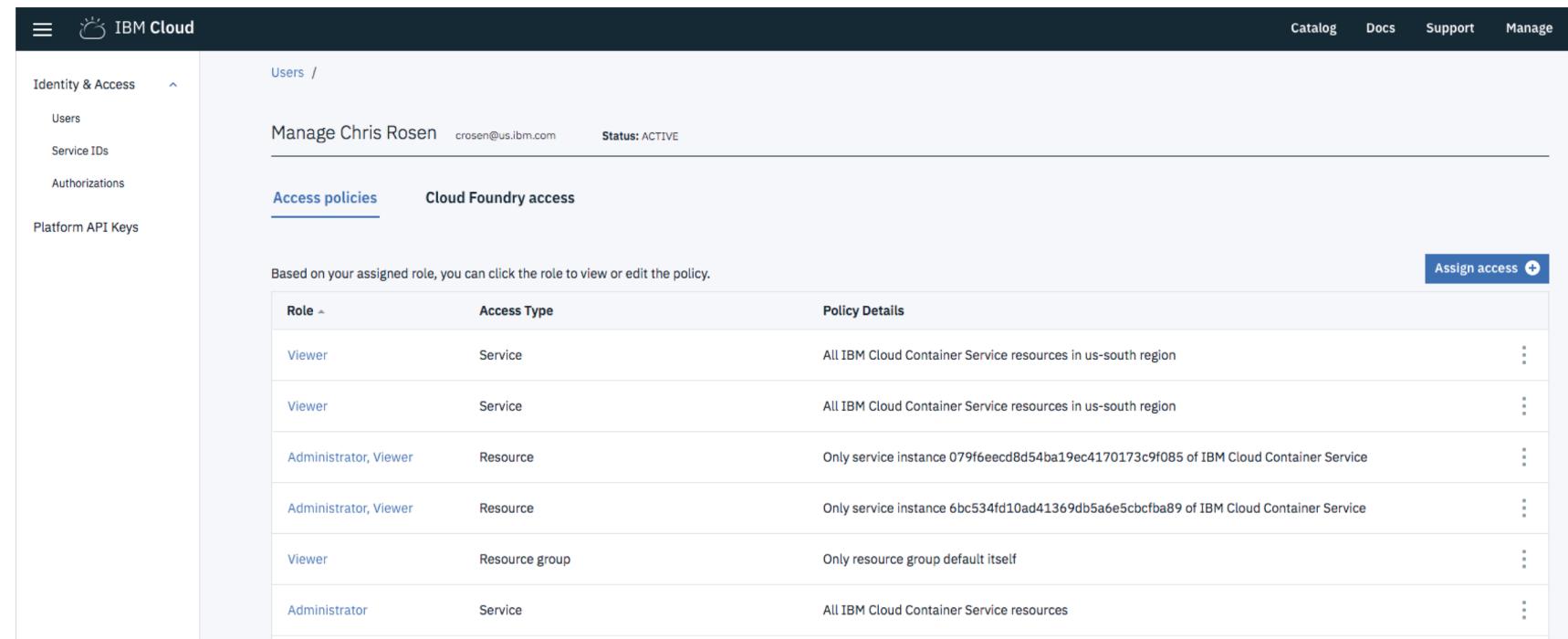
```
bx cs workers cj
Listing cluster workers...
OK
ID          Public IP    Private IP    Machine Type  State  Status
kube-dal10-cr337c42379cd044fb82070462bdb09215-w1 169.47.220.231 10.171.61.21 u1c.2x4  normal  Ready
kube-dal10-cr337c42379cd044fb82070462bdb09215-w2 169.47.220.235 10.171.61.58 u1c.2x4  normal  Ready
kube-dal10-cr337c42379cd044fb82070462bdb09215-w3 169.47.220.201 10.171.61.51 u1c.2x4  normal  Ready
```

Kubernetes Cluster Management Experience

Kubernetes In-Cluster Experience

RBAC with IBM Cloud IAM

Use IBM Cloud authorization service (IAM) to set Kubernetes RBAC policies
Consistent way to set authorization across cloud resources



The screenshot shows the IBM Cloud Identity & Access Management (IAM) interface. The left sidebar has a collapsed menu icon, the IBM Cloud logo, and the 'Identity & Access' section with sub-options: 'Users', 'Service IDs', 'Authorizations', and 'Platform API Keys'. The main area shows a user profile for 'Chris Rosen' (crosen@us.ibm.com) with 'Status: ACTIVE'. Below the profile, there are two tabs: 'Access policies' (which is selected) and 'Cloud Foundry access'. A note says: 'Based on your assigned role, you can click the role to view or edit the policy.' A table lists the assigned roles and their details:

Role	Access Type	Policy Details
Viewer	Service	All IBM Cloud Container Service resources in us-south region
Viewer	Service	All IBM Cloud Container Service resources in us-south region
Administrator, Viewer	Resource	Only service instance 079f6eeecd8d54ba19ec4170173c9f085 of IBM Cloud Container Service
Administrator, Viewer	Resource	Only service instance 6bc534fd10ad41369db5a6e5cbcfa89 of IBM Cloud Container Service
Viewer	Resource group	Only resource group default itself
Administrator	Service	All IBM Cloud Container Service resources

A blue button in the top right corner says 'Assign access +'.

Isolation and scheduling control

Managing isolation within a Cluster

Kubernetes Cluster

rbac (ClusterRole, ClusterRoleBinding)

quotas

- *requests (cpu/mem)*
- *limits (cpu/mem)*
- *storage*
- *object counts*

*network isolation policies
rbac (Roles, RoleBindings)*

Namespace A

*toleration pod
node affinity/anti-affinity
pod affinity/anti-affinity
resource request
resource limit*

Worker node

*labels
taint*

Worker node

Worker node

Worker node

Create Cluster

Single Zone Cluster

Region
US East

Cluster type
Standard

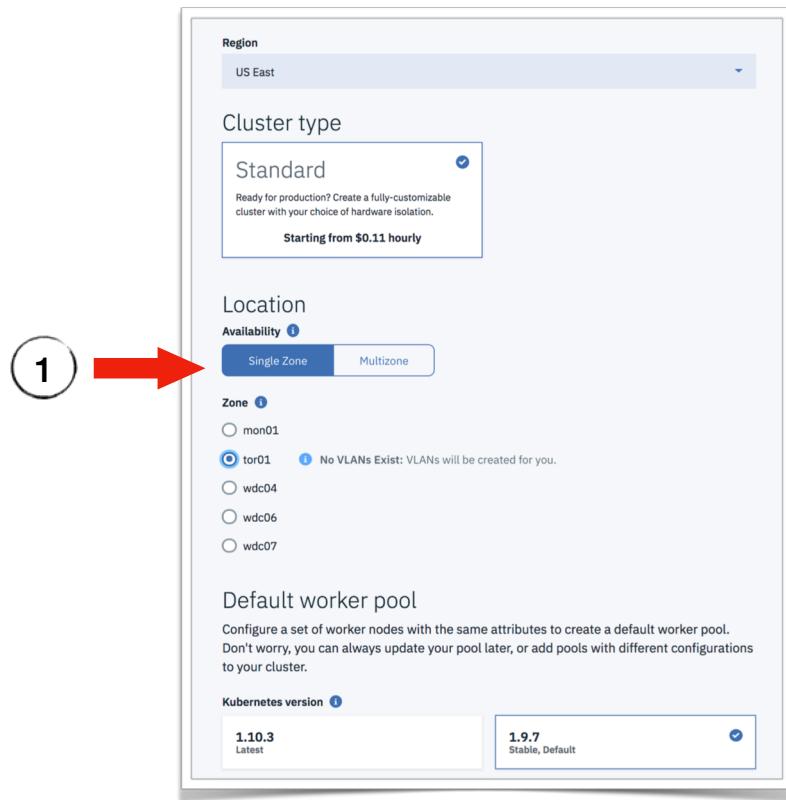
Ready for production? Create a fully-customizable cluster with your choice of hardware isolation.
Starting from \$0.11 hourly

Location
Availability Single Zone Multizone

Zone 1
 tor01 No VLANs Exist: VLANs will be created for you.
 wdc04
 wdc06
 wdc07

Default worker pool
Configure a set of worker nodes with the same attributes to create a default worker pool.
Don't worry, you can always update your pool later, or add pools with different configurations to your cluster.

Kubernetes version
1.10.3
Latest 1.9.7
Stable, Default



Multizone Cluster

Region
US East

Cluster type
Standard

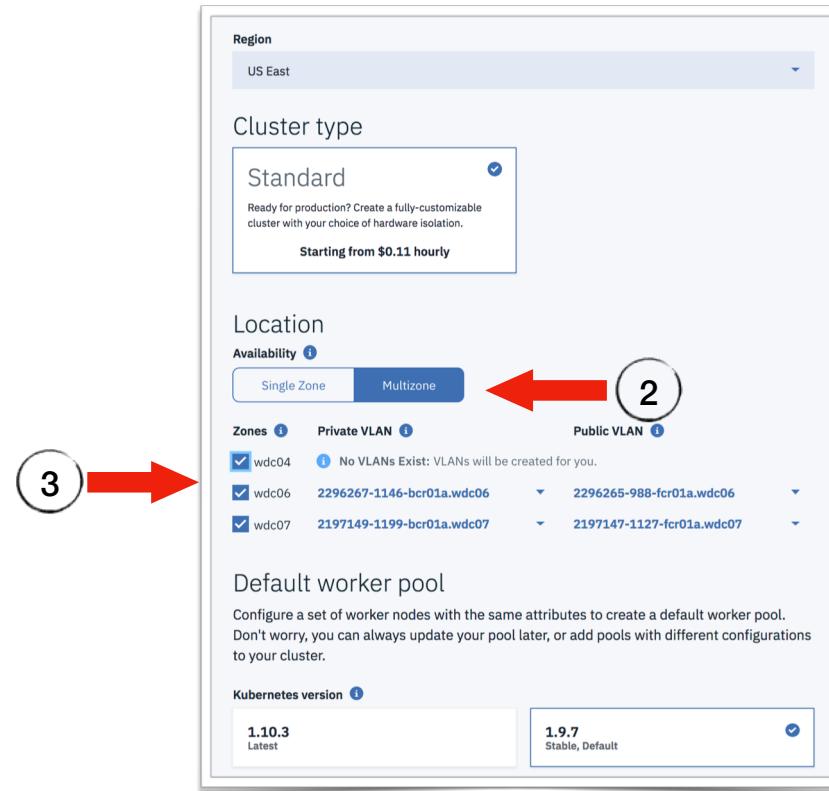
Ready for production? Create a fully-customizable cluster with your choice of hardware isolation.
Starting from \$0.11 hourly

Location
Availability Single Zone Multizone

Zones 1 Private VLAN 1 Public VLAN 1
 wdc04 No VLANs Exist: VLANs will be created for you.
 wdc06 2296267-1146-bcr01a.wdc06 2296265-988-fcr01a.wdc06
 wdc07 2197149-1199-bcr01a.wdc07 2197147-1127-fcr01a.wdc07

Default worker pool
Configure a set of worker nodes with the same attributes to create a default worker pool.
Don't worry, you can always update your pool later, or add pools with different configurations to your cluster.

Kubernetes version
1.10.3
Latest 1.9.7
Stable, Default



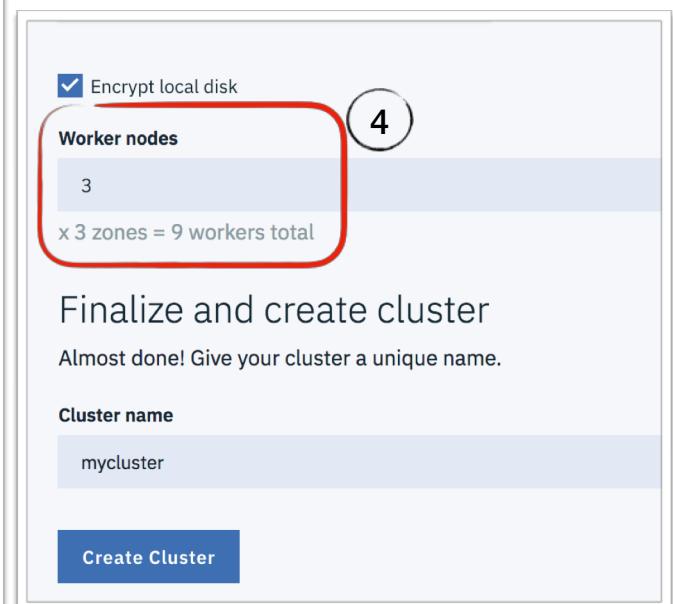
Encrypt local disk

Worker nodes
3
x 3 zones = 9 workers total

Finalize and create cluster
Almost done! Give your cluster a unique name.

Cluster name
mycluster

Create Cluster



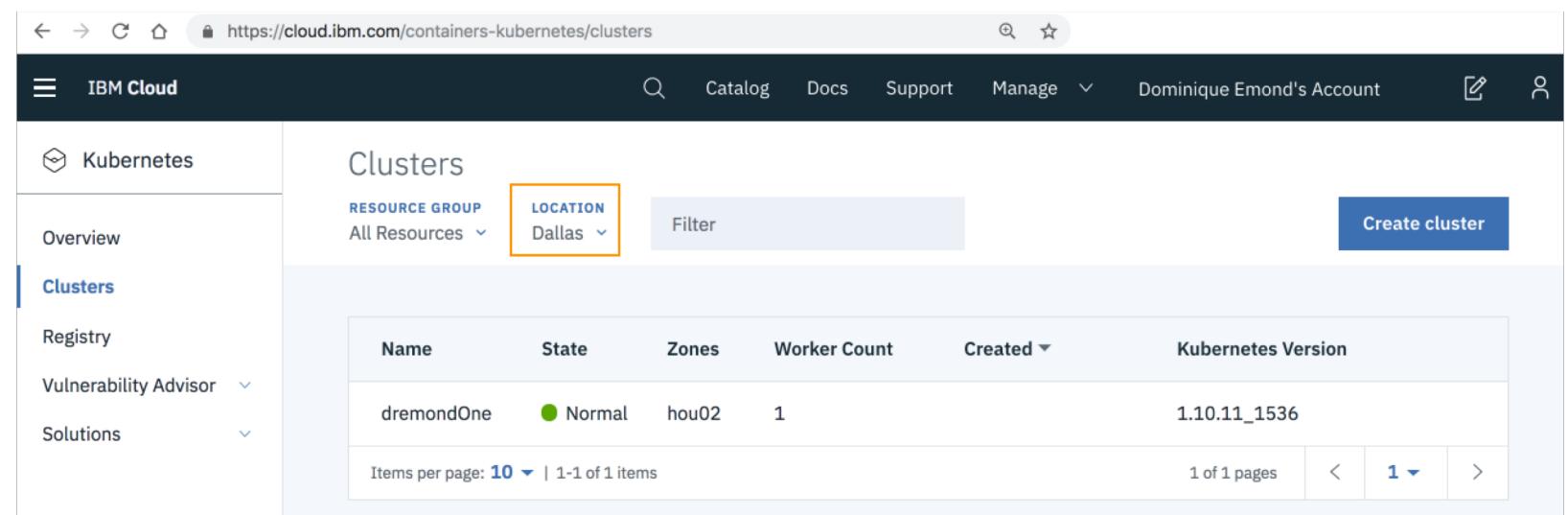
Hands On: Create a Kubernetes Cluster

Flow:

1. Create IBM Account
2. Apply : Promo Code
3. Create Cluster

Technology stack:

- IBM Public Cloud
- Kubernetes



The screenshot shows the IBM Cloud interface for managing Kubernetes clusters. The URL in the browser is https://cloud.ibm.com/containers-kubernetes/clusters. The left sidebar has a 'Clusters' section selected under the 'Kubernetes' heading. The main area displays a table of clusters. One cluster, named 'dremondOne', is listed with the following details: State: Normal, Zones: hou02, Worker Count: 1, and Kubernetes Version: 1.10.11_1536. The 'LOCATION' dropdown menu is highlighted with an orange box, showing 'Dallas' selected. The table has columns for Name, State, Zones, Worker Count, Created, and Kubernetes Version. At the bottom, there are pagination controls: 'Items per page: 10' (with a dropdown), '1-1 of 1 items', '1 of 1 pages', and navigation arrows.

Name	State	Zones	Worker Count	Created	Kubernetes Version
dremondOne	Normal	hou02	1		1.10.11_1536

Resources

Code Patterns

- <https://developer.ibm.com/patterns/category/containers/>

IBM Cloud Kubernetes Service

- <https://cloud.ibm.com/docs/containers?topic=containers-getting-started>