



nIE

Future Edition



Docker 101

Docker 101



Damian Flynn

MVP Cloud & Datacenter + Cisco Champion

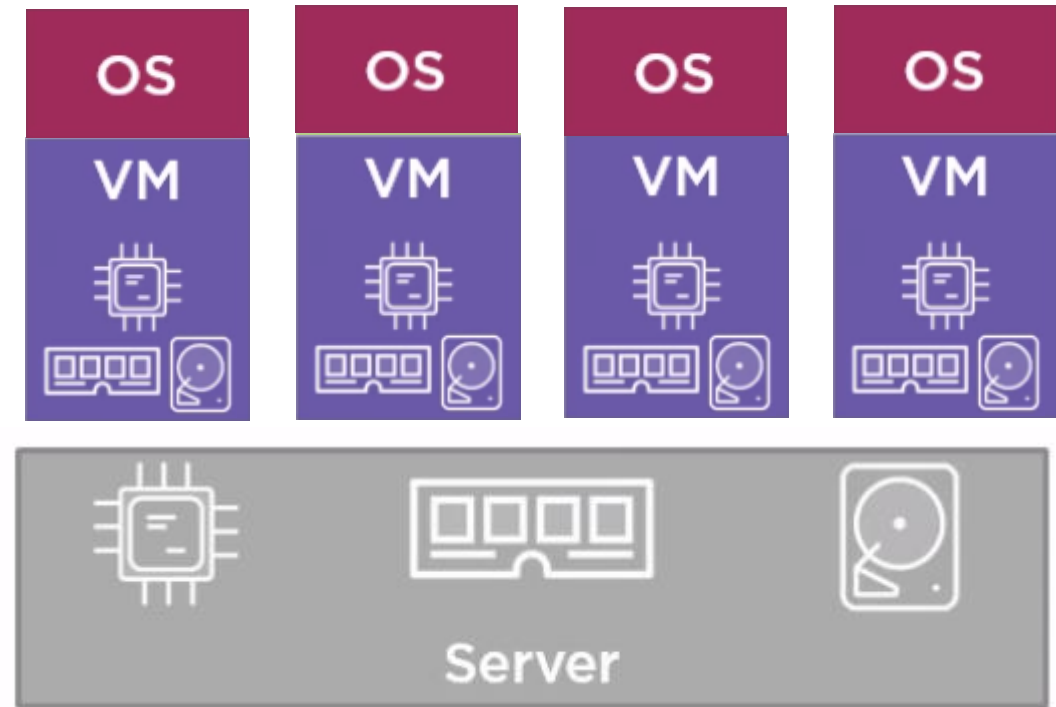


@Damian_Flynn

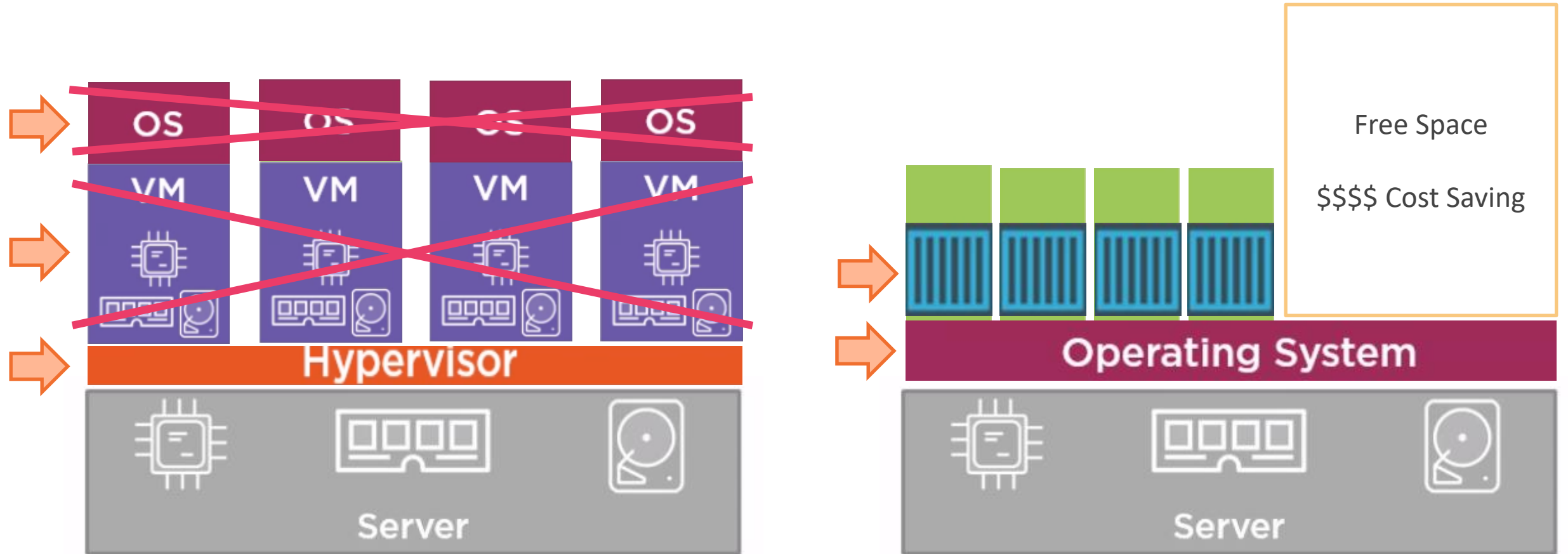
www.DamianFlynn.Com

Hyper-Visor Challenge's

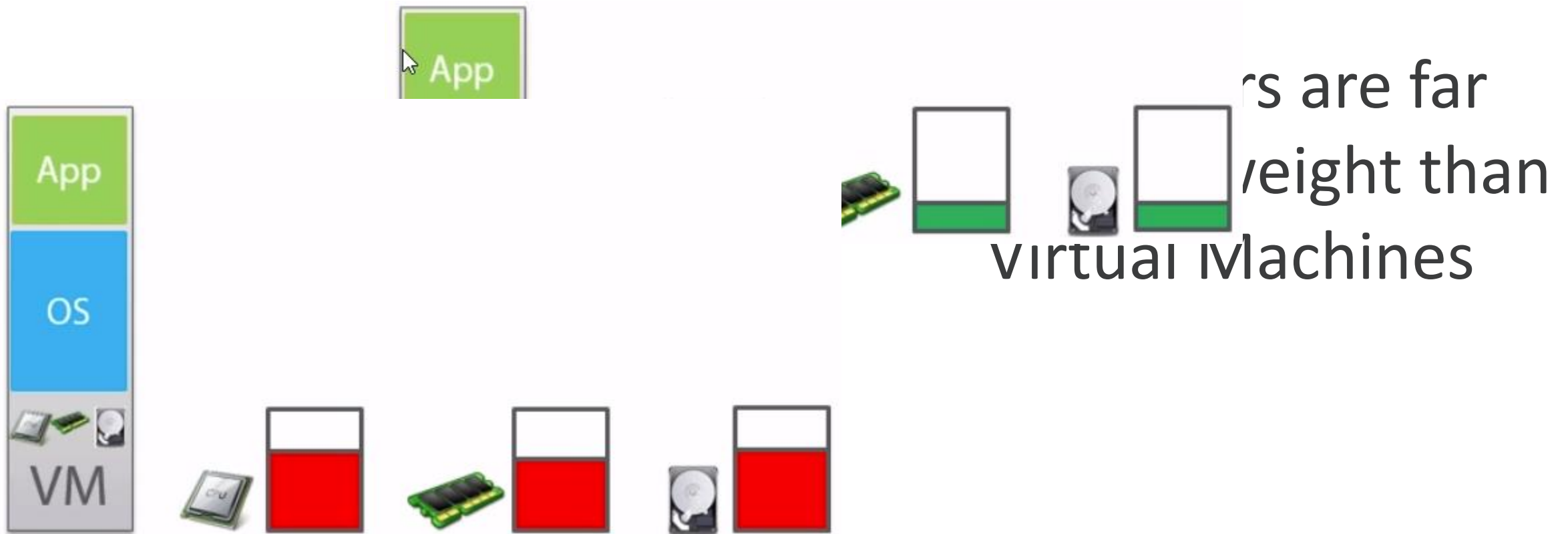
- Multiple Application Instances
- Instance Per Virtual Machine
 - Dedicated Host Resource Slice
 - Dedicated OS Instance
 - CPU
 - RAM
 - Disk
 - OS License
 - Administration



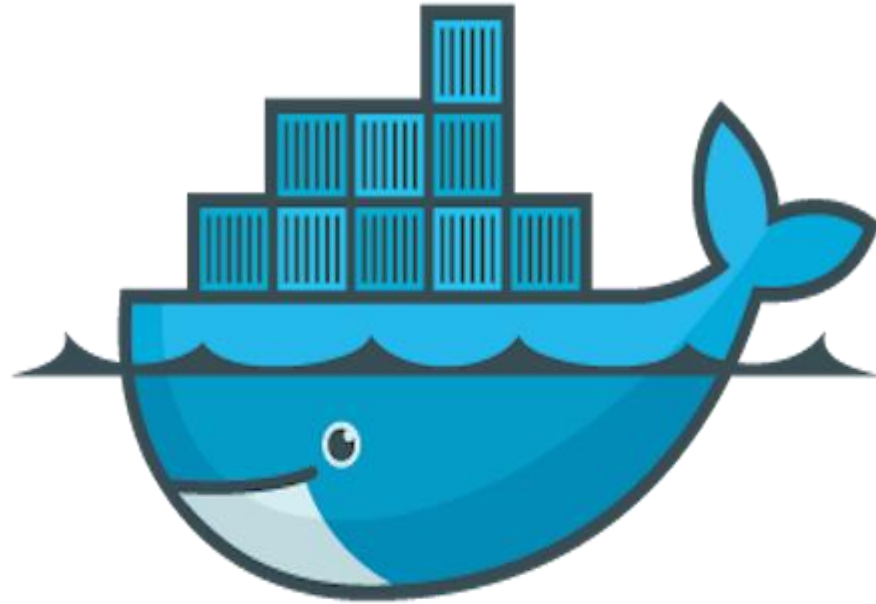
Containers



Containers



Docker



docker

Docker



Docker

Overview Timeline Followers Contributors

 [FOLLOW](#)

STATISTICS

 493  10K

Overview UPDATE

Acquisitions
5 Acquisitions

Funding Received
\$180M in 5 Rounds from 10 Investors

Headquarters: **San Francisco, California**

Description: Docker is an open platform for distributed applications that allows developers and sysadmins to build, ship & run distributed applications.


Founders: **Solomon Hykes**

Categories: **Software, Open Source, Development Platforms**

Website: <https://www.docker.com>



Docker x

← → ↺  GitHub, Inc. [US] <https://github.com/docker>

 Search GitHub Pull requests

 **Docker**
<http://www.docker.com>

 Repositories  People 223  Teams 2

Filters ▾ Find a repository...

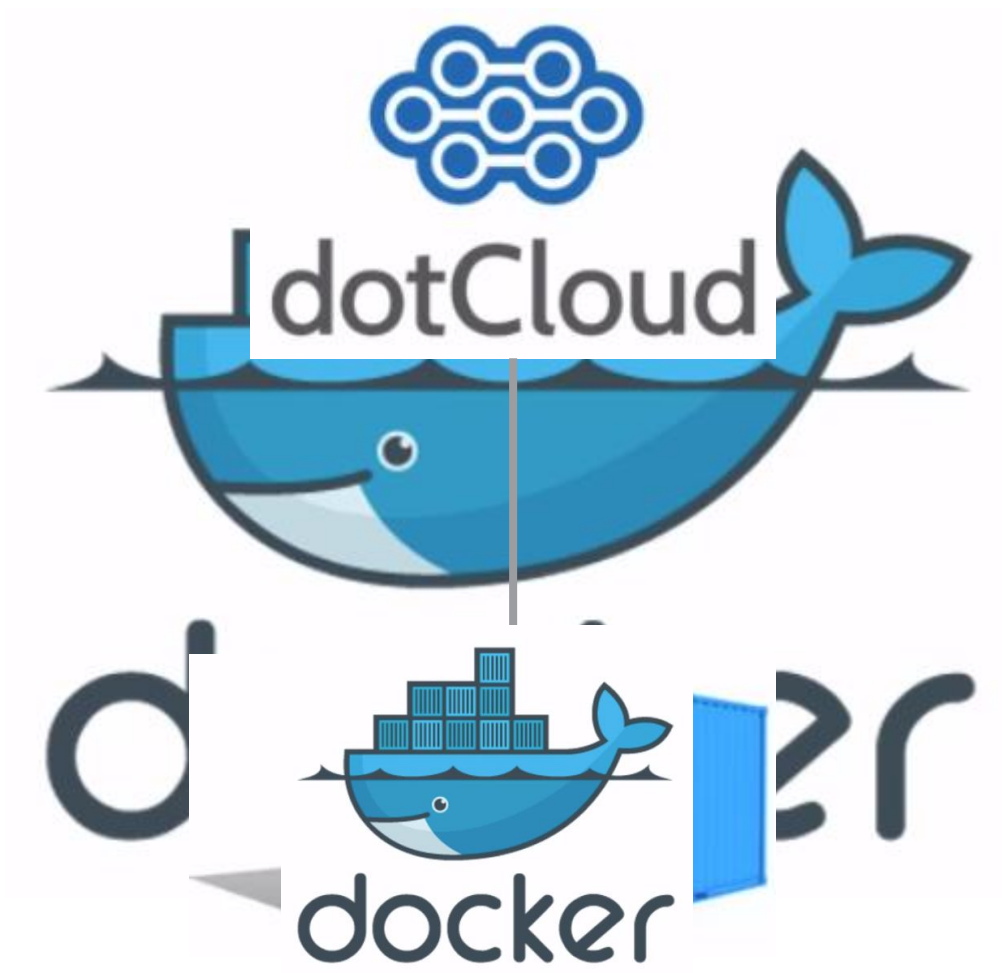
machine
Machine management for a container-centric world
Updated 7 minutes ago

docker
Docker - the open-source application container engine



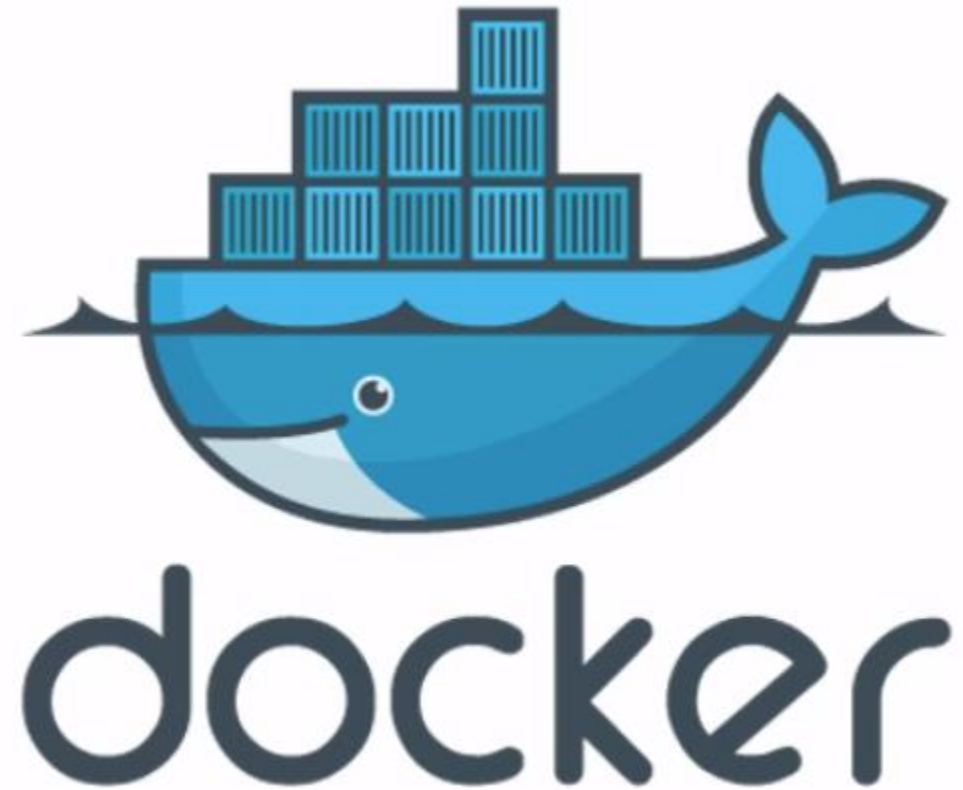
OPEN CONTAINER INITIATIVE

Docker Inc.



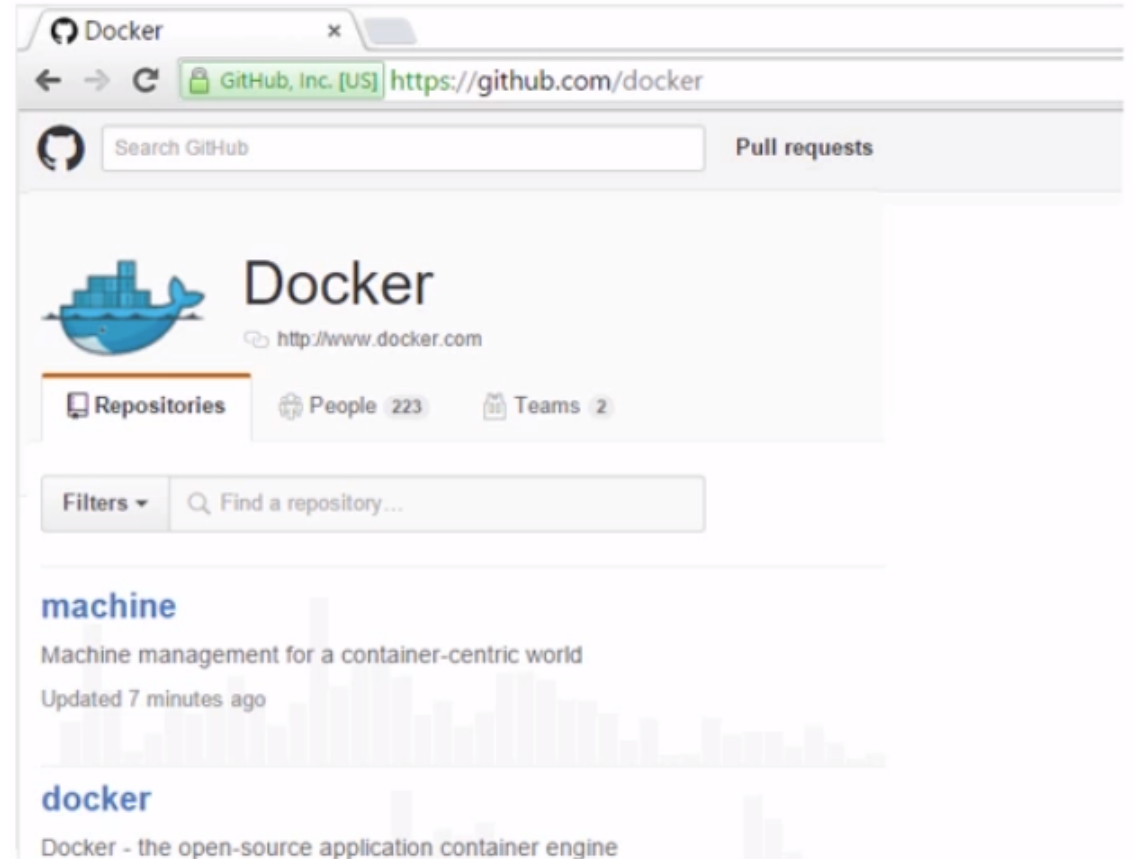
Docker Inc.

- Leading Technology company
- Billion Dollar Valuation
- Raised over \$150M in venture Capital
- Approx. 200 Employees
- Acquired companies

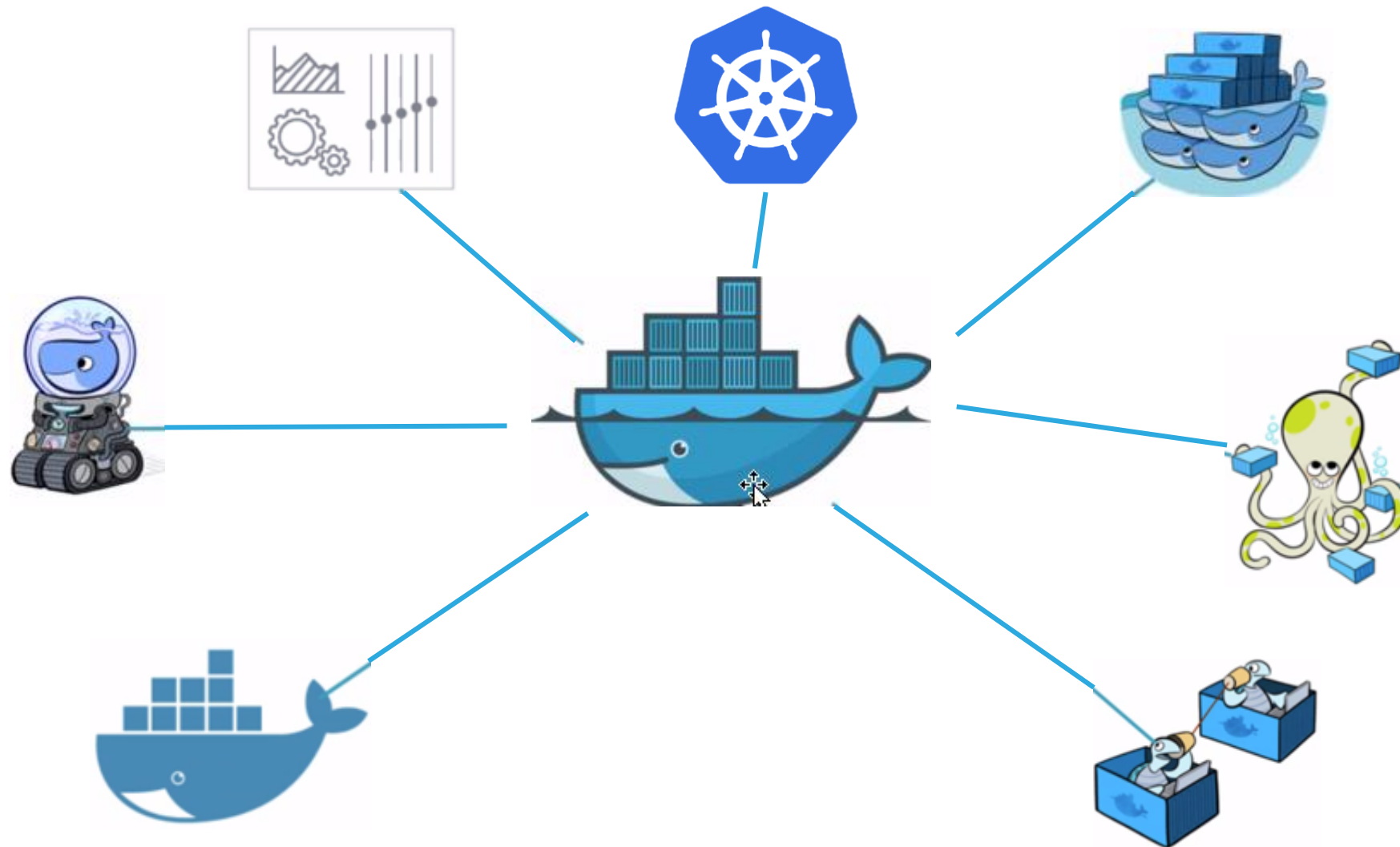


Docker Project

- Open Source
 - Apache License 2.0
- Focused on Tools & Process Efficiency Improvement
- Multiple tools

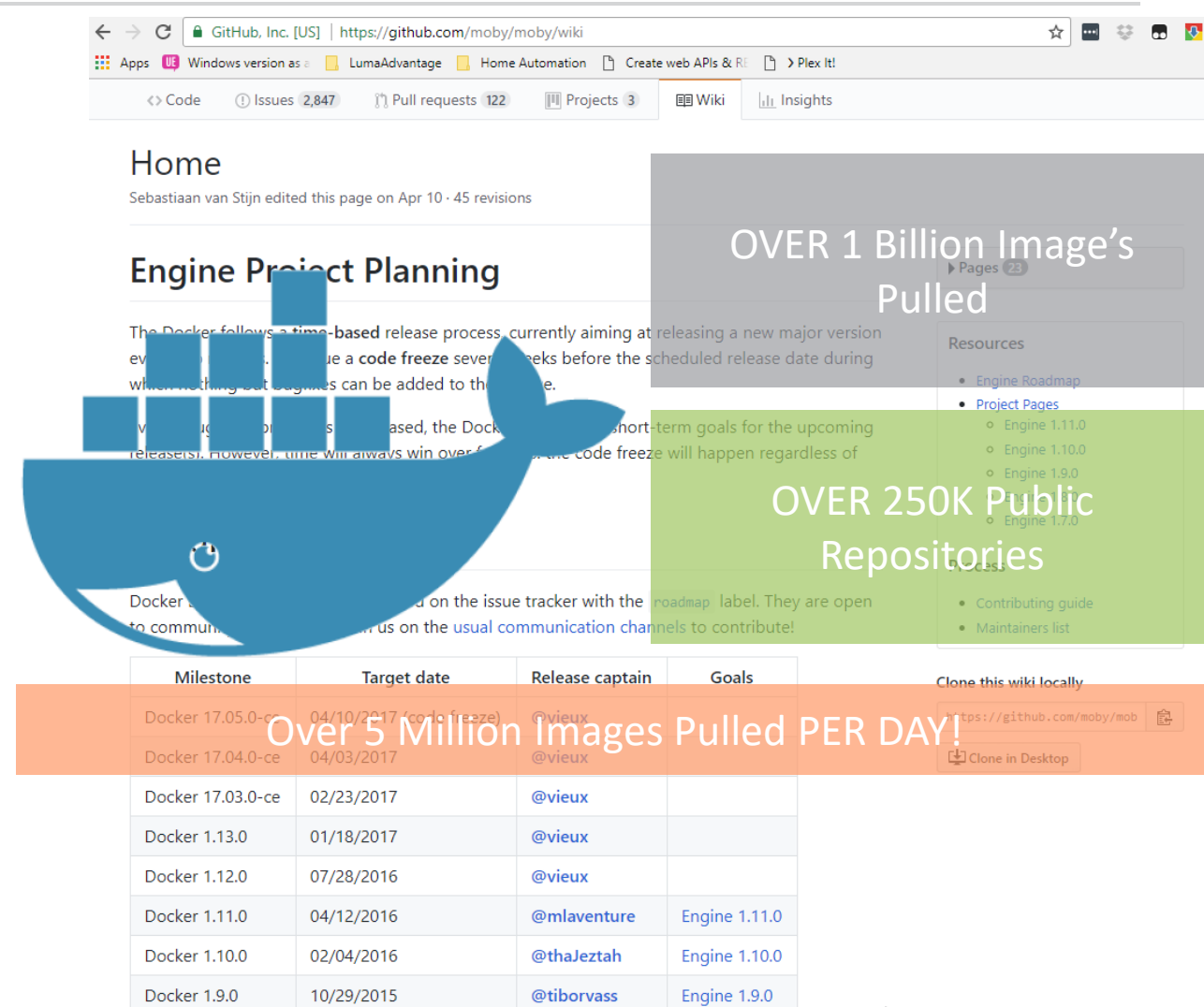


Docker Project



Docker Project

- Open Source
 - Apache License 2.0
- Focused on Tools & Process Efficiency Improvement
- Multiple tools
- Biggest Industry Players Involved



The screenshot shows the Docker project's GitHub repository page. A large blue whale logo is overlaid on the page. The page content includes the 'Home' section, 'Engine Project Planning' section, and a table of milestones. A large orange banner at the bottom of the screenshot reads 'Over 5 Million Images Pulled PER DAY!'. To the right of the banner, there are two green boxes: one stating 'OVER 1 Billion Image's Pulled' and another stating 'OVER 250K Public Repositories'.

Home
Sebastiaan van Stijn edited this page on Apr 10 · 45 revisions

Engine Project Planning

The Docker follows a time-based release process, currently aiming at releasing a new major version every 6 weeks. We have a code freeze several weeks before the scheduled release date during which nothing but bugfixes can be added to the code. When a new version is released, the Docker team will release a short-term goals for the upcoming releases. However, time will always win over feature requests and the code freeze will happen regardless of the roadmap label. They are open to community members to contribute!

Milestone	Target date	Release captain	Goals
Docker 17.05.0-ce	04/10/2017 (code freeze)	@vieux	
Docker 17.04.0-ce	04/03/2017	@vieux	
Docker 17.03.0-ce	02/23/2017	@vieux	
Docker 1.13.0	01/18/2017	@vieux	
Docker 1.12.0	07/28/2016	@vieux	
Docker 1.11.0	04/12/2016	@mlaventure	Engine 1.11.0
Docker 1.10.0	02/04/2016	@thajeztah	Engine 1.10.0
Docker 1.9.0	10/29/2015	@tiborvass	Engine 1.9.0

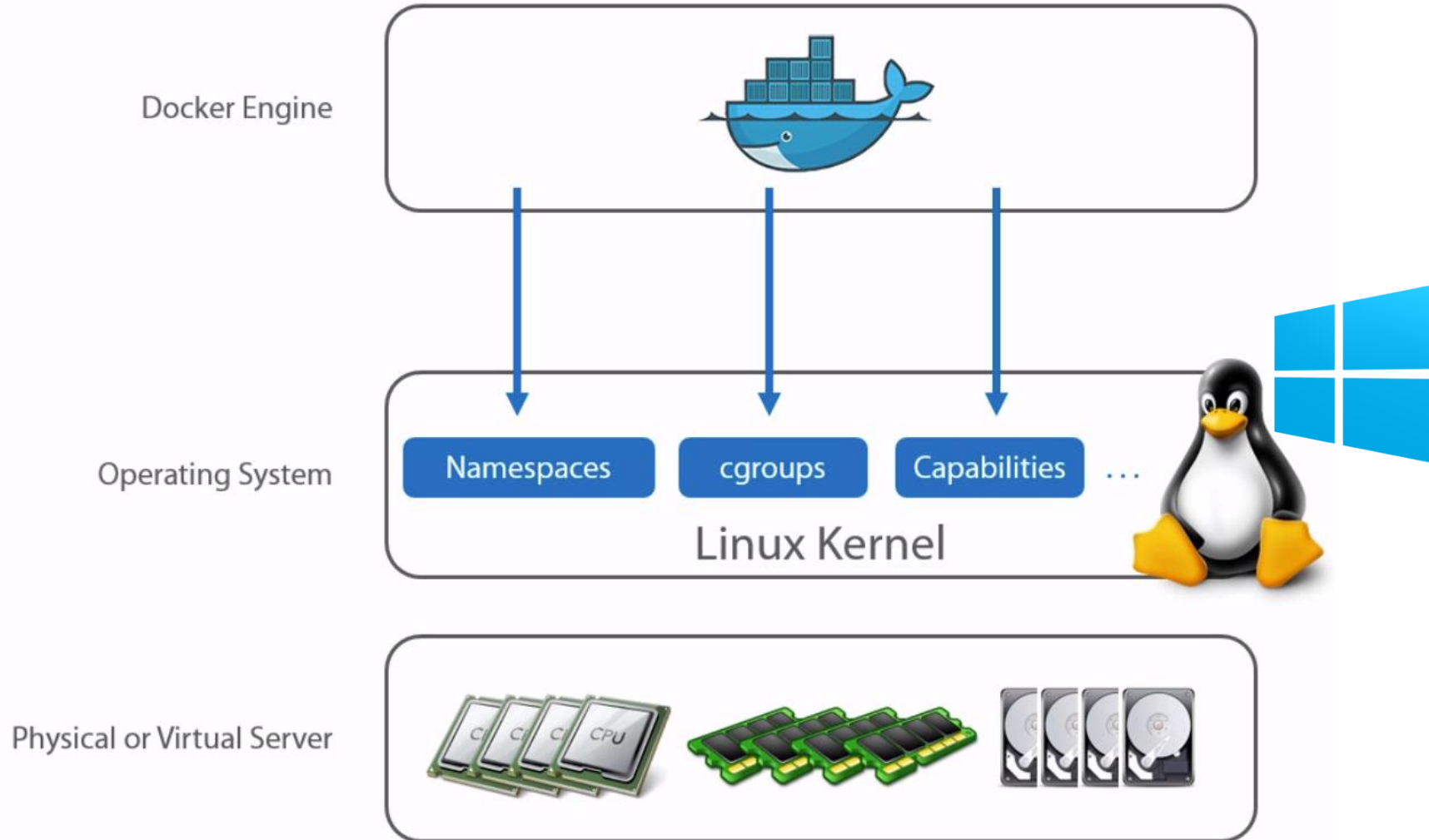
OVER 1 Billion Image's Pulled

OVER 250K Public Repositories

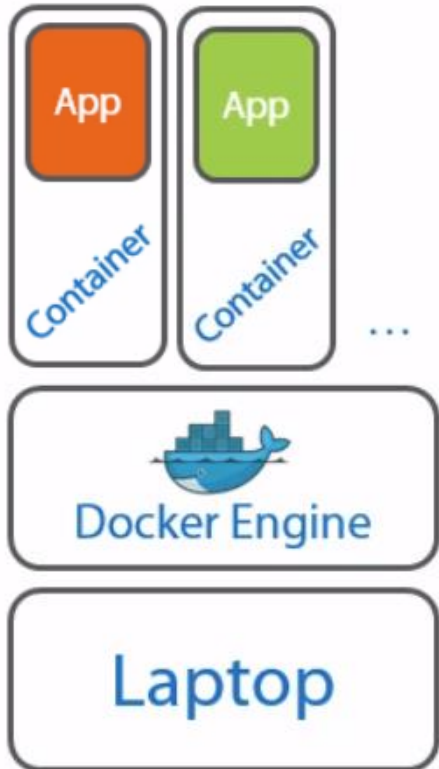
Over 5 Million Images Pulled PER DAY!

Clone this wiki locally
<https://github.com/moby/moby>
Clone in Desktop

Docker Relationship



Docker Container Standardization



Open Container Initiative



rkt
specification

Open Container Initiative

- Lightweight Governance Council
 - Standardize
 - Container Format
 - Container Runtime
- Vendor Neutral
- Platform Neutral
- Formed June 2015
- Linux Foundation



Summary

Docker Inc.

Silicon Valley Startup

Venture Capital Backed

Growing through Acquisitions

Docker Project

More Efficient
than Physical

Wasted Resources
&
Duplicate OPEX

Open Container Initiative

Greater Efficiencies

Less mature than Hyper-Visors

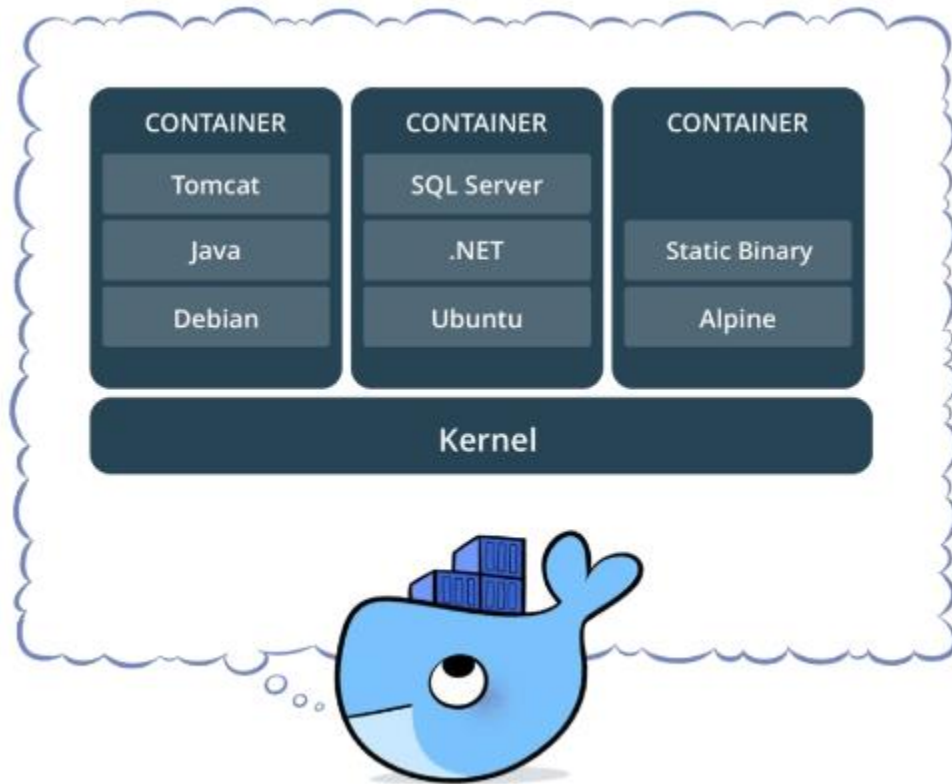
Maturing Eco-System

Docker Containers are NOT Virtual Machines

- Easy connection to ma
- Fundamentally
 - Different Architecture
 - Different Benefit's



What is a container



- Standardized Packaging for software and dependencies

Isolated apps from each other

Works for all major Linux distributions

Containers native to Windows Server 2016

Standardized Packaging

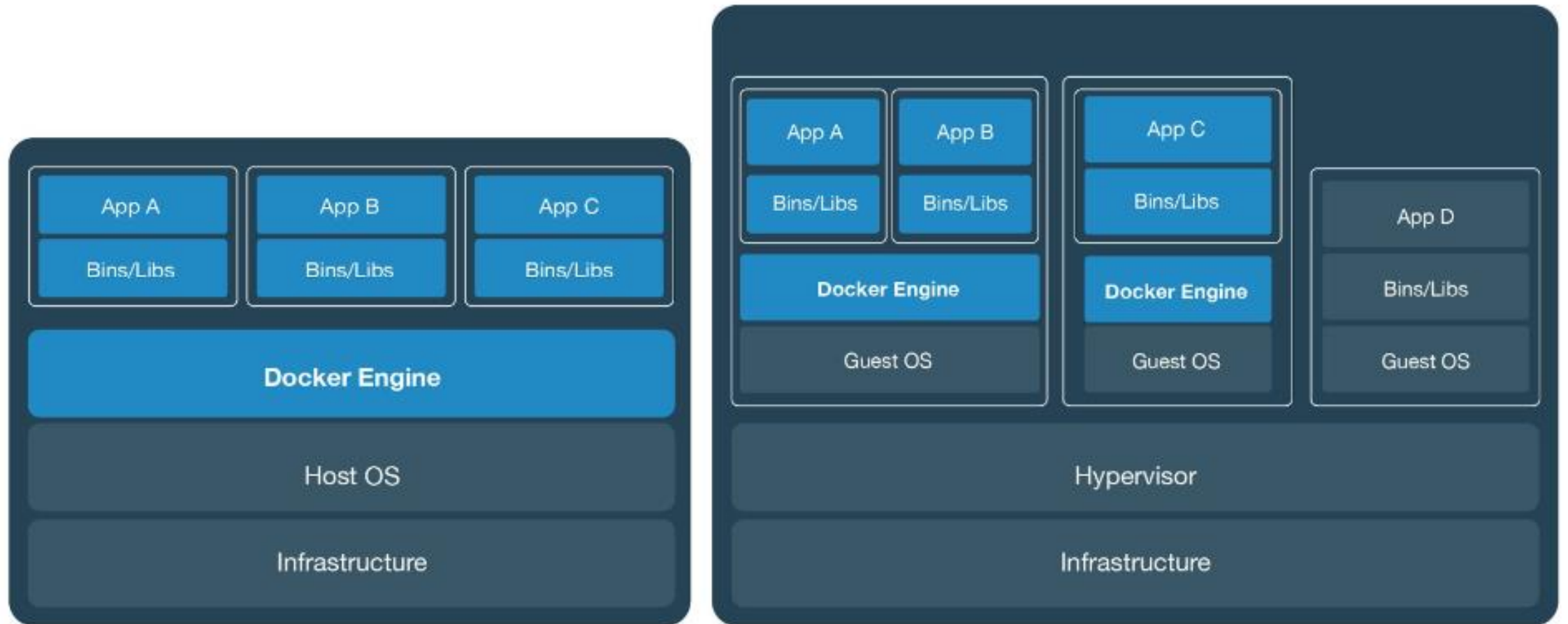


Beware of Smugglers

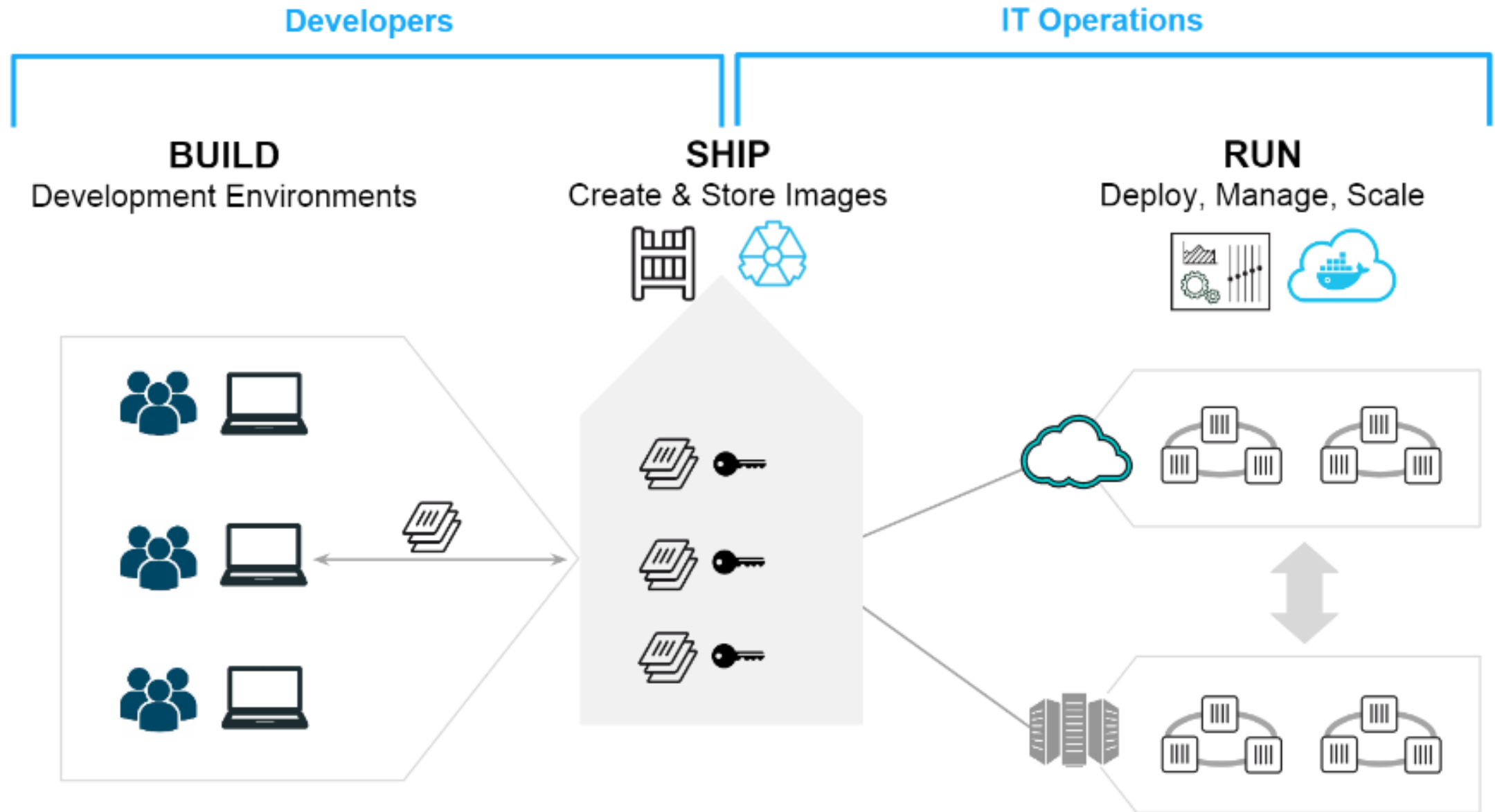


- Containers CAN contain malicious code
- Ensure your TRUST the content of the container

They are NOT Mutually Exclusive!



Using Docker

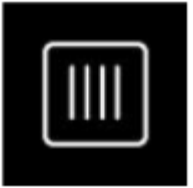


Vocabulary



Docker Image

- The basis of a docker container, represents a full application



Docker Container

- The standard unit in which the application service resides and executes



Docker Engine

- Creates, ships and runs Docker containers deployable on a physical or virtual, hosted locally, in a datacenter or cloud service provider



Registry Service

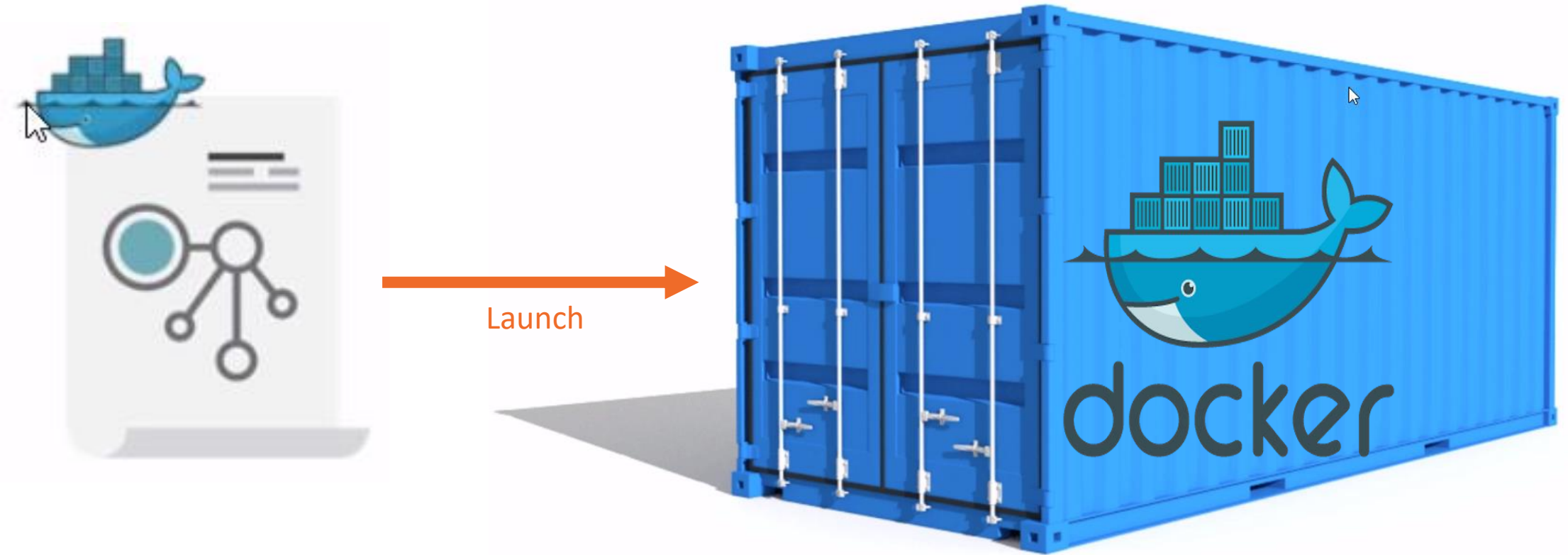
- Cloud or server based storage and distribution service for your

Docker Images



Docker Images

Docker Images



Docker Container's



Image

Build Time



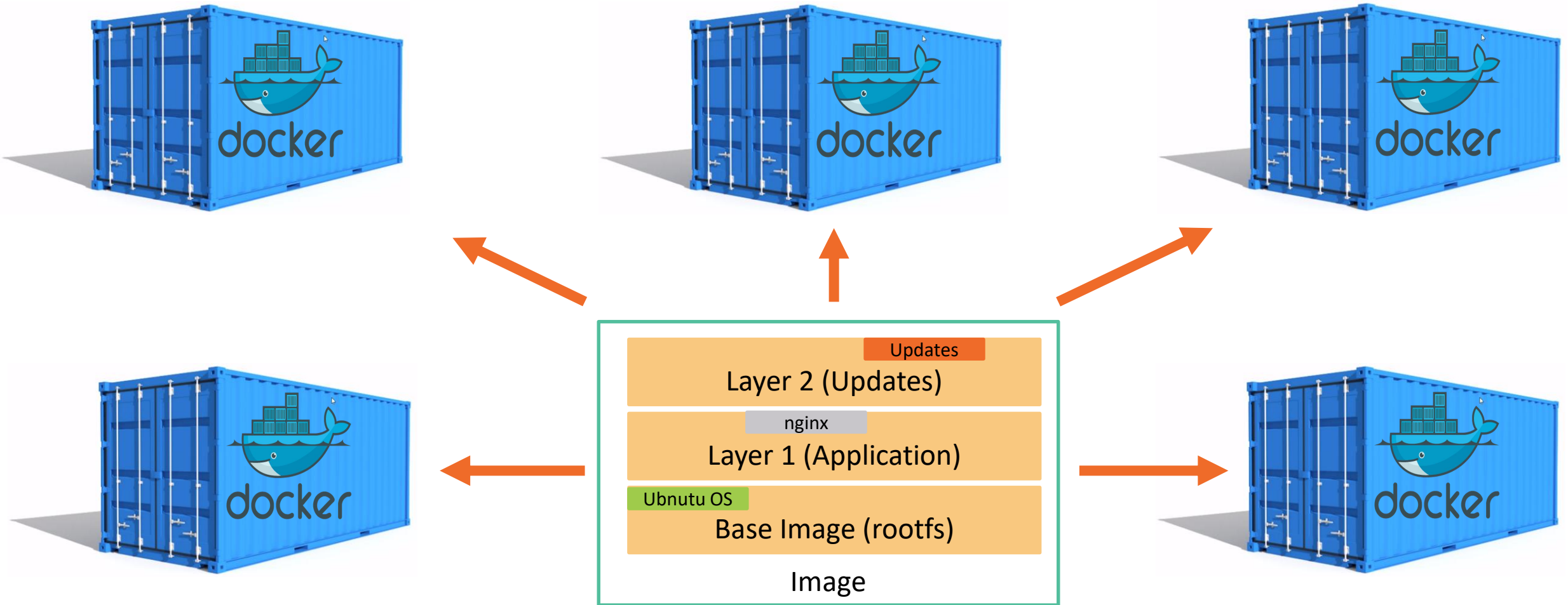
Launch



Container

Run Time

Images



Single Image deployed as Multiple Containers

Docker 101 – Hand On

- Check your images

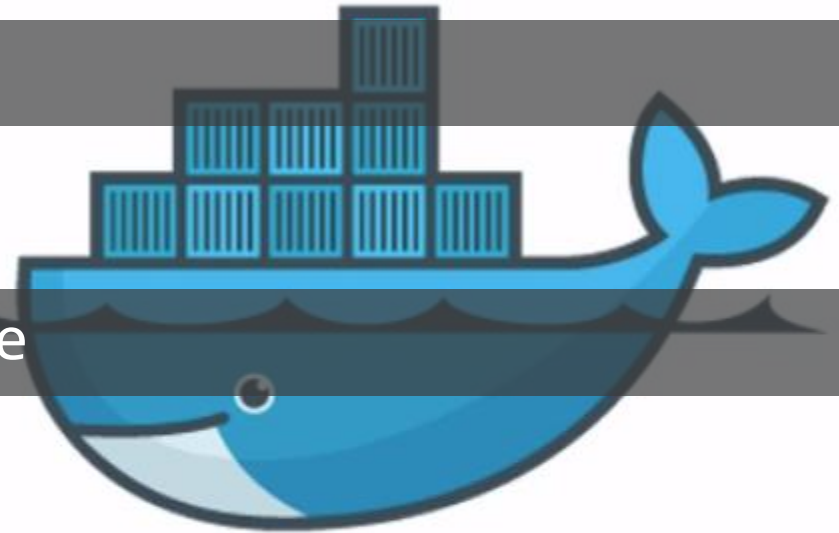
```
docker images
```

- Pull Sample Image

```
docker pull microsoft/WindowsServerCore
```

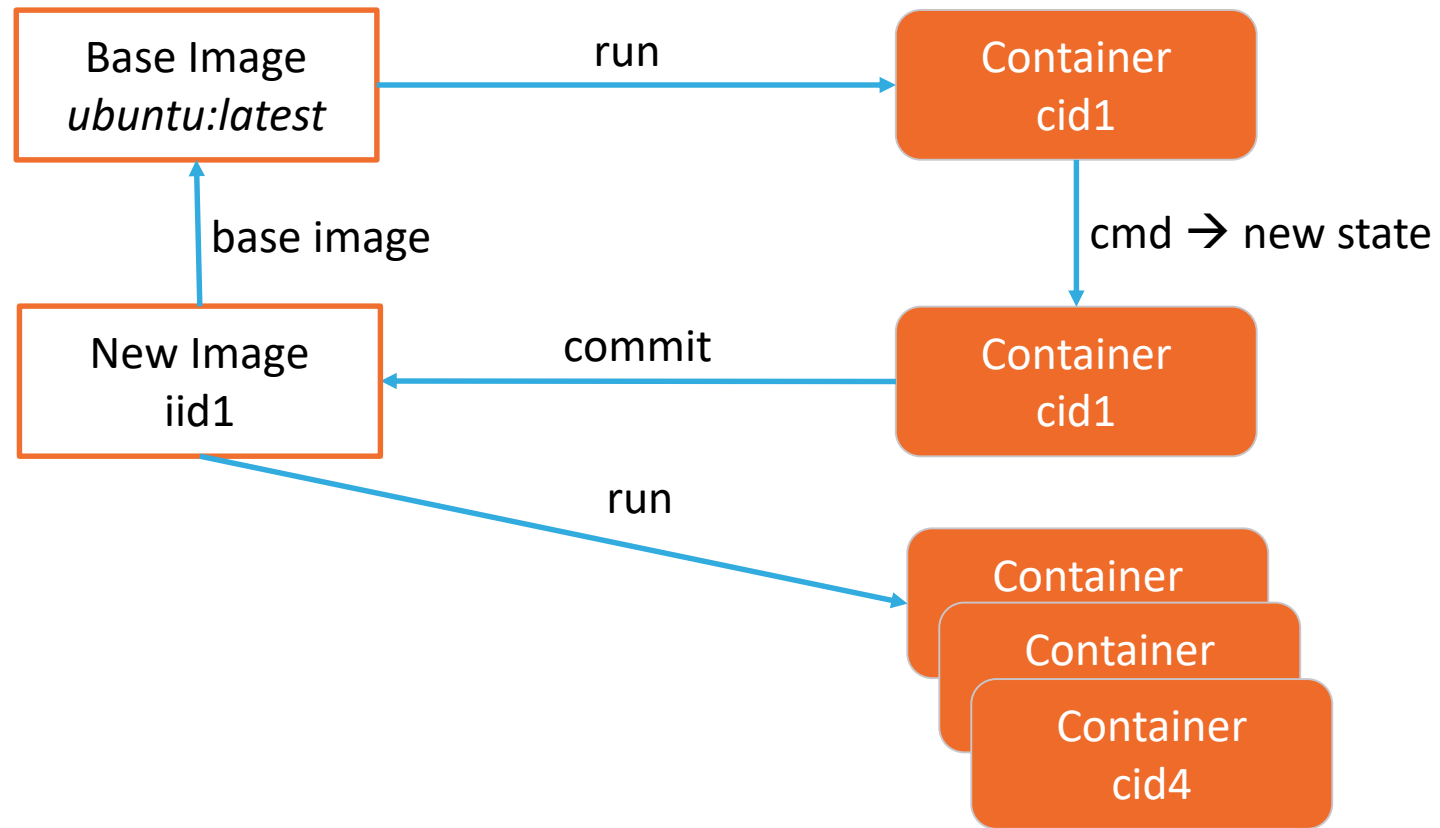
- Start your Container

```
docker run -it microsoft/WindowsServerCore cmd
```

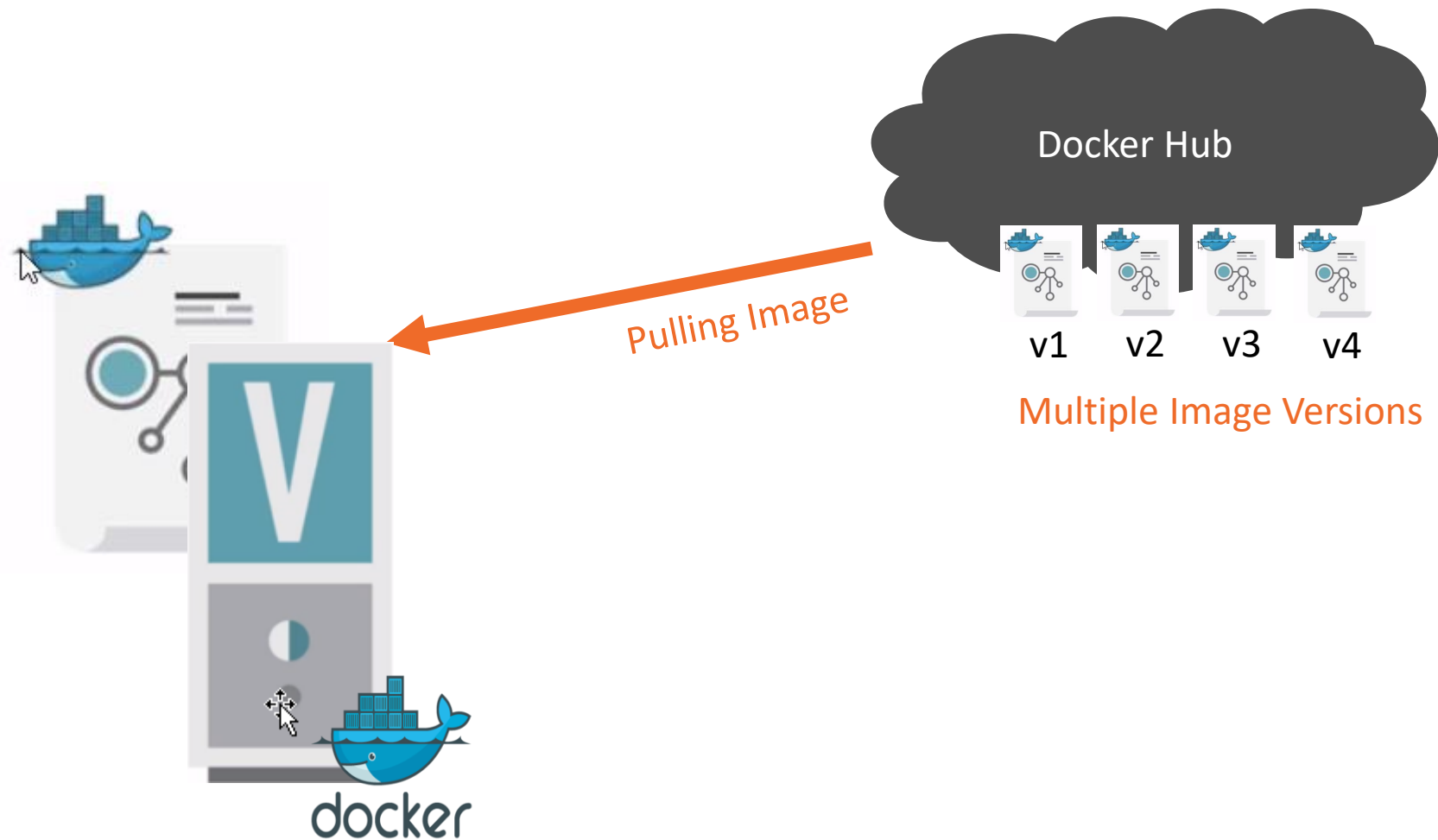


docker

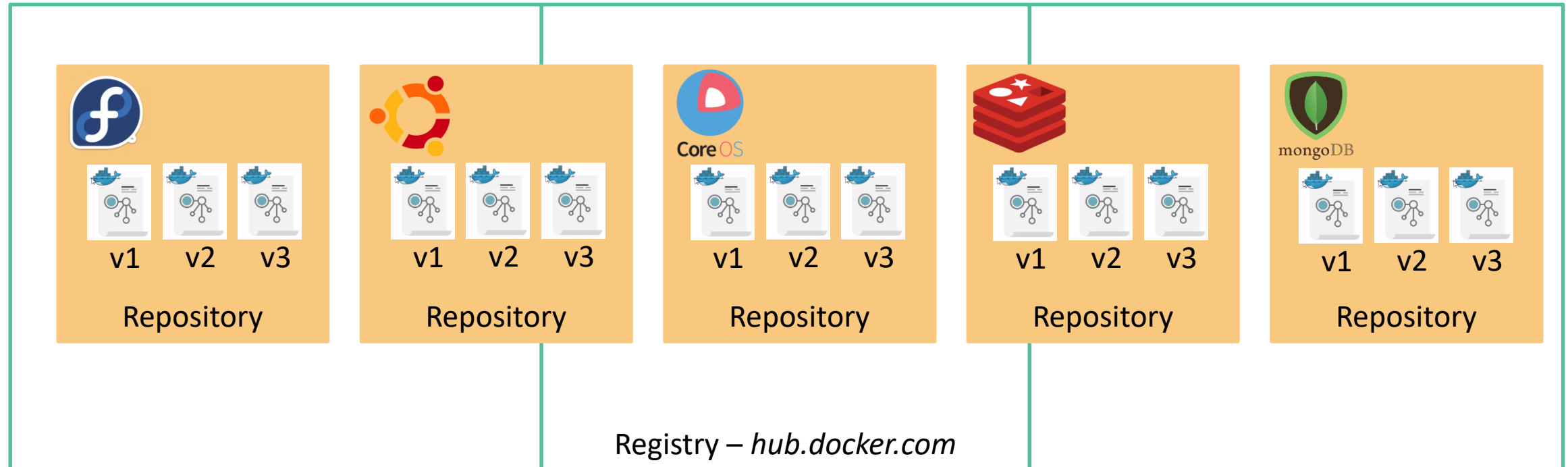
Images & Containers



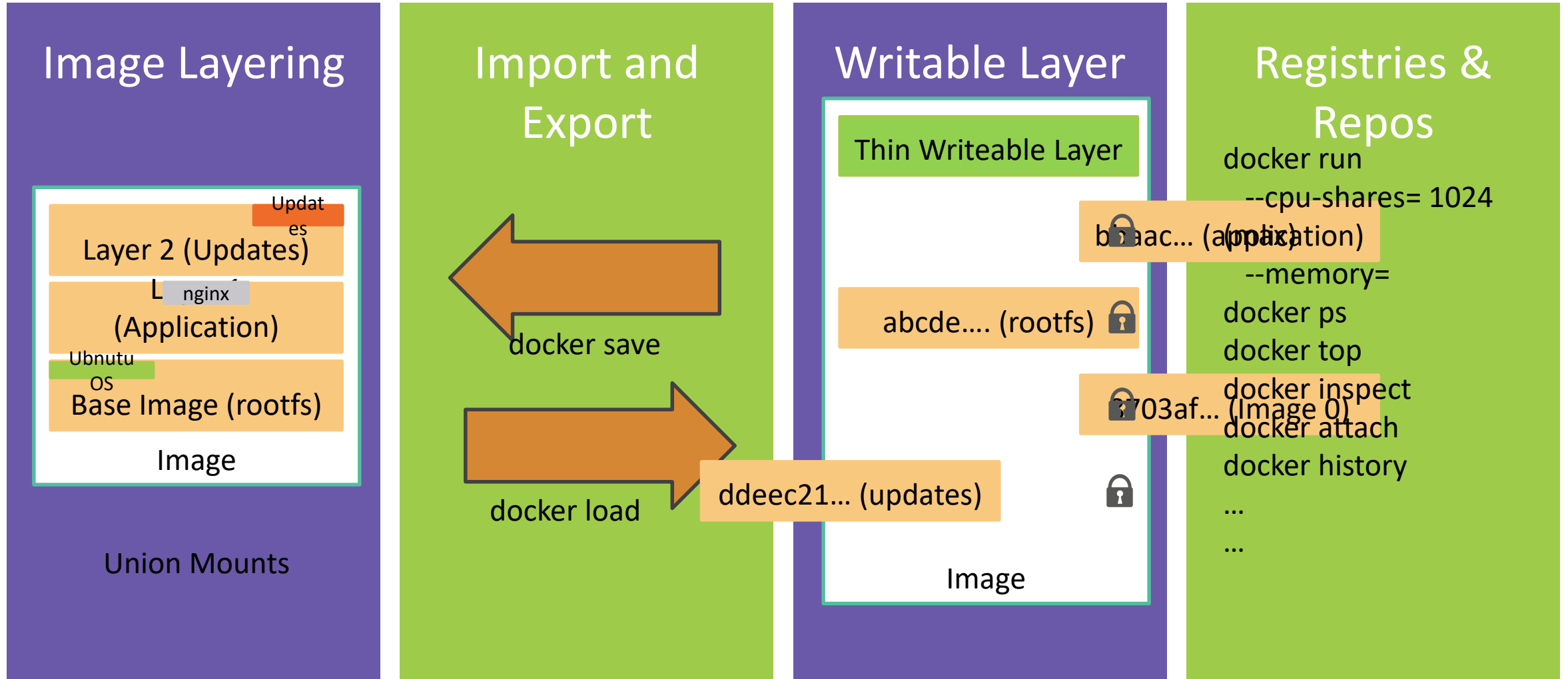
Docker Images



Docker Registries



Summary



Dockerfile

- 'Dockerfile'
- Plain Text
- Simple Format
- Instructions to Build image
- Start in an empty directory

Docker File

```
— . . — . . — . . — . .  
— . . — . . — . . — . .  
— . . — . . — . . — . .  
— . . — . . — . . — . .  
— . . — . . — . . — . .
```

Dockerfile

Our Base Image

FROM node:4.6

Set our working folder

WORKDIR /app

Add any files in our directory recursively to the image

ADD . /app

Start the Node Package manager and install requested packages

RUN npm install

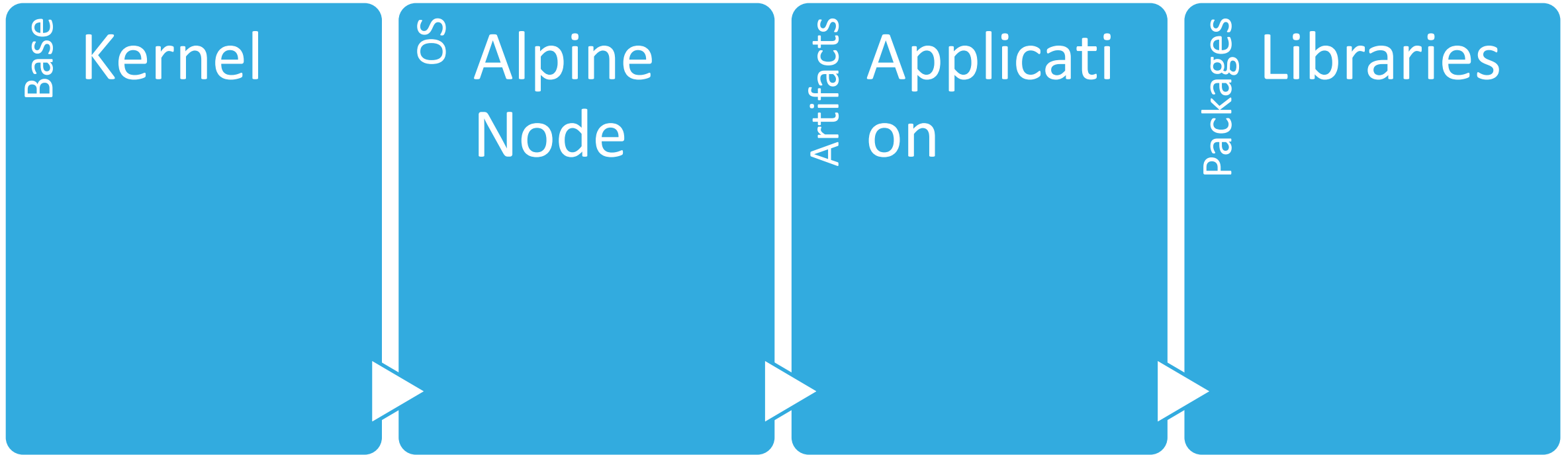
What port number the container should expose

EXPOSE 3000

run the application

CMD npm start

Each command creates a layer



Dockerfile

```
docker build -t helloworld:0.1 .
```

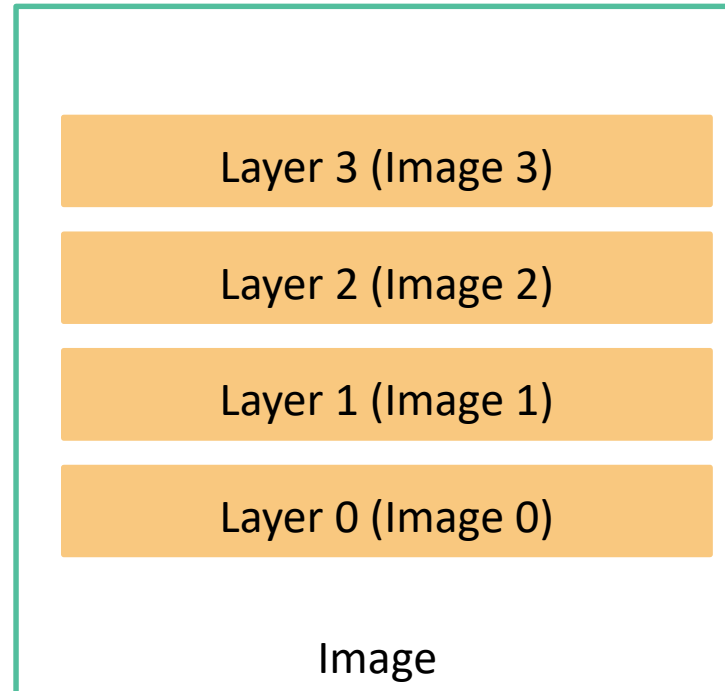
Layers on a Physical Disk

- Logical file system by grouping different file system primitives into branches
 - Directories, File Systems, Sub volumes, Snapshots
- Each Branch represents a layer in a docker image
- Containers will share common layers on the host
- Enabled images to be constructed as required, instead of monolithic images
- At startup the container adds a writeable layer on top of this file system

Copy on Write

- Super efficient
 - Sub second instantiation times from containers
 - New container can consume < 1Mb of Space
- Containers appears to be a copy of the original image
 - Really its just a link to the original shared image
- A change to the file system, will presented as “Copied Up”

Images



Images

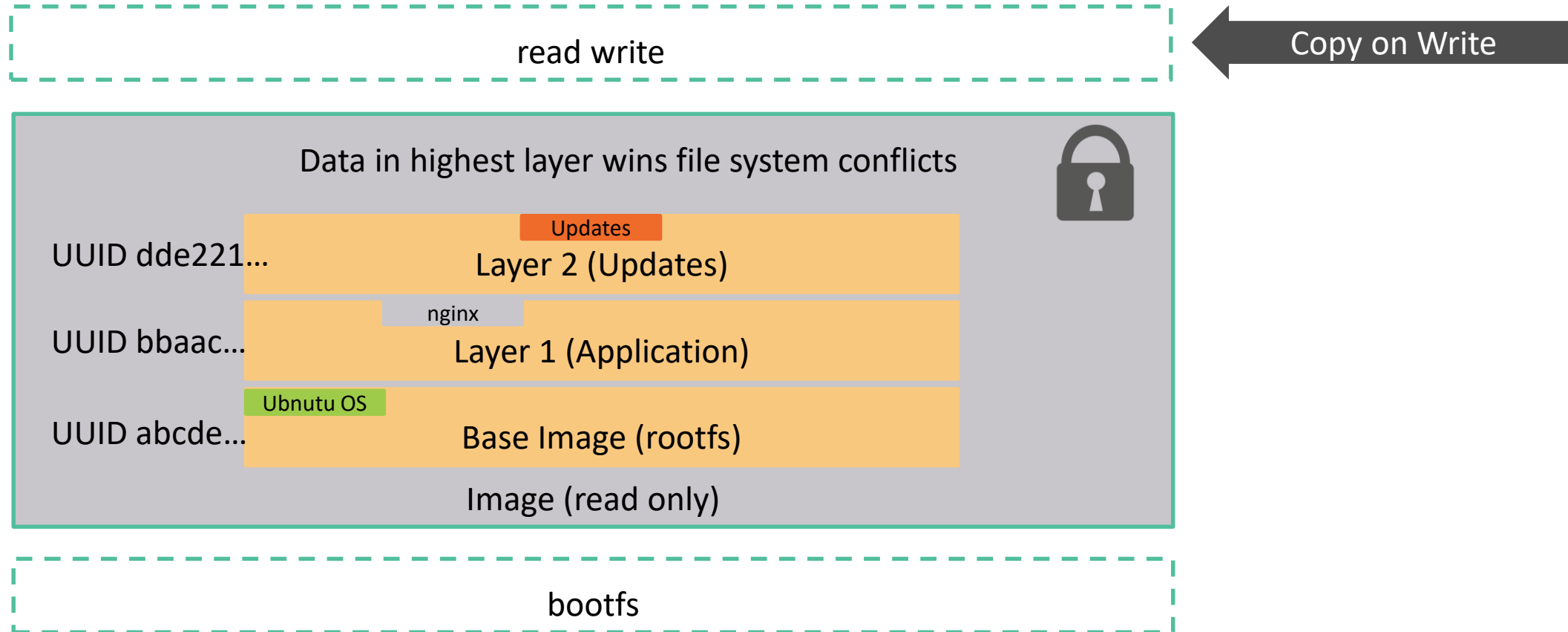
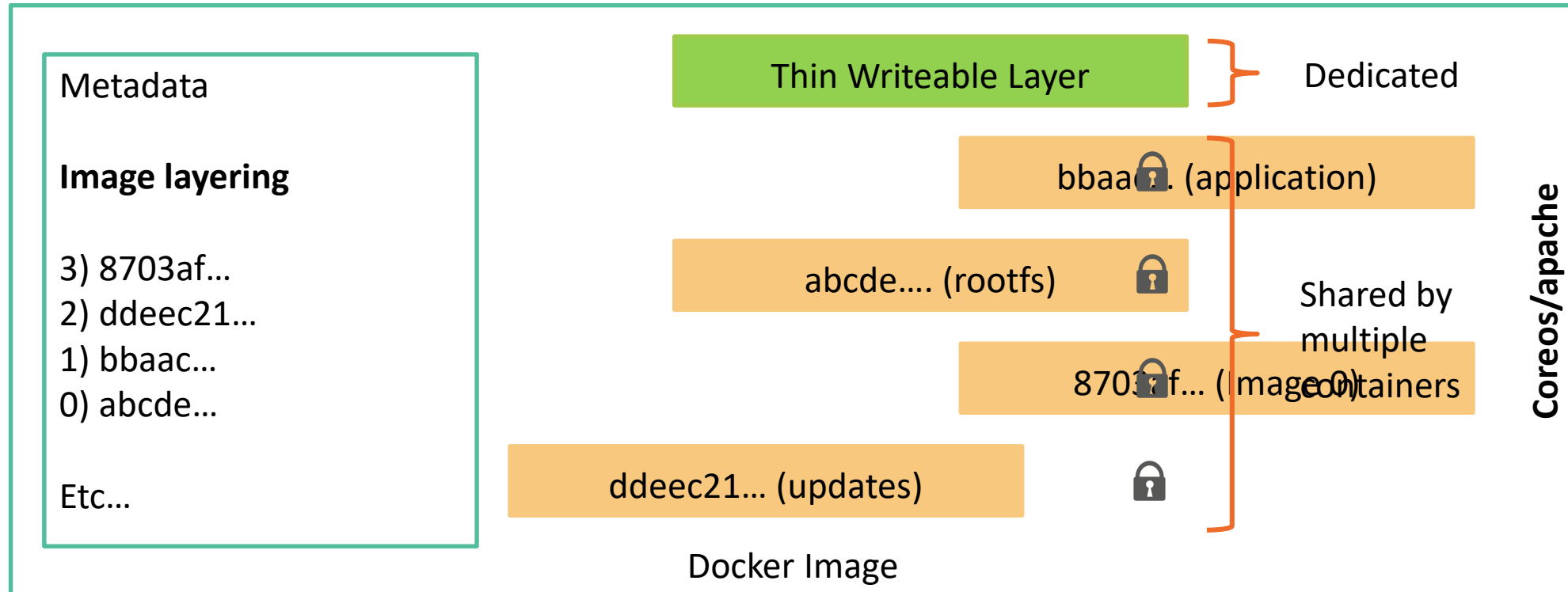


Image layering is accomplished through **union mounts**

Containers



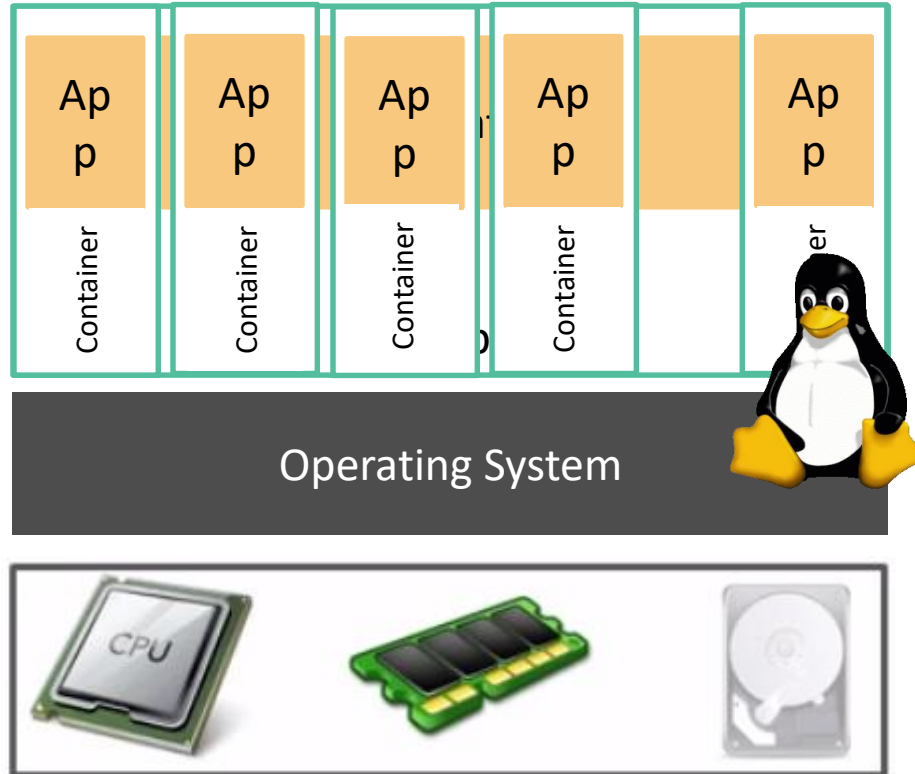
Every **container** gets its own writable top layer

nic

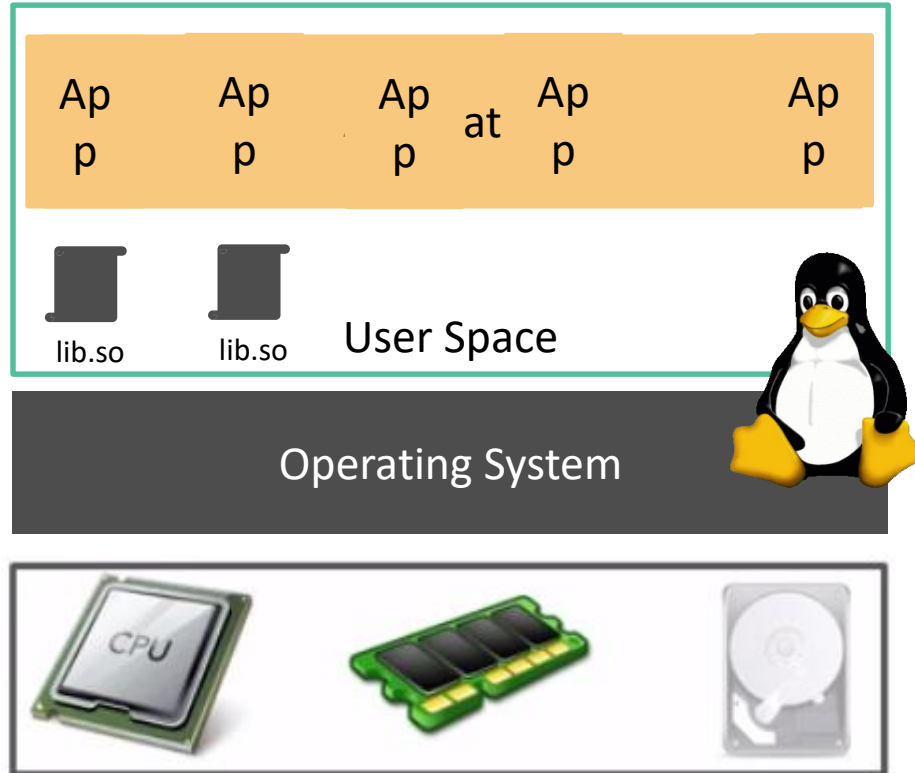
Resources

Slides and demos from the conference will be available at github.com/nordicinfrastructureconference/2018 (bit.ly/2y7JhA3)

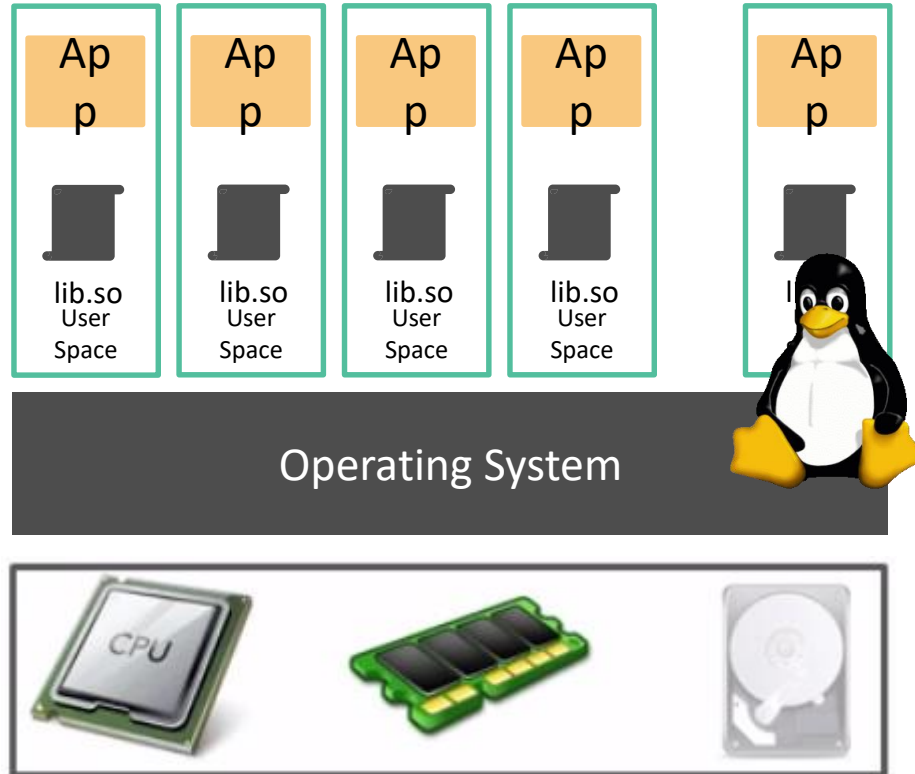
Containers



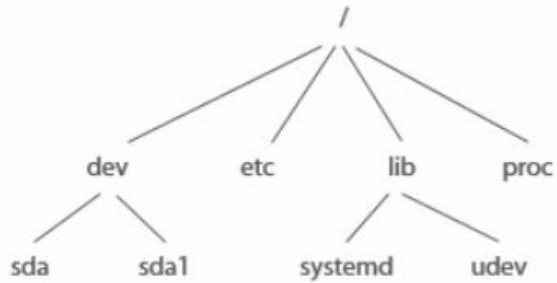
How Containers Work



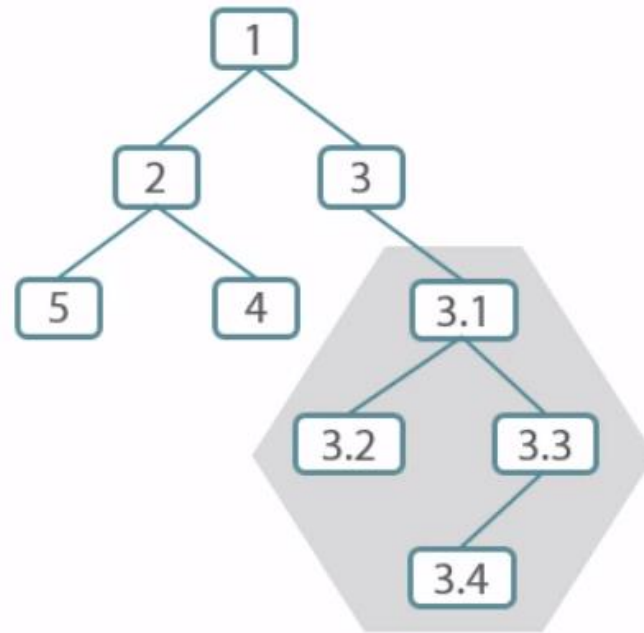
How Containers Work



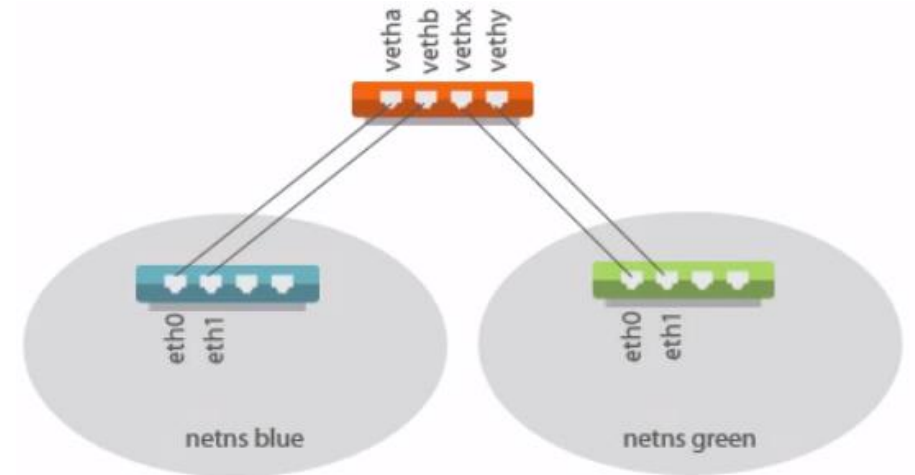
How Containers Work



File System

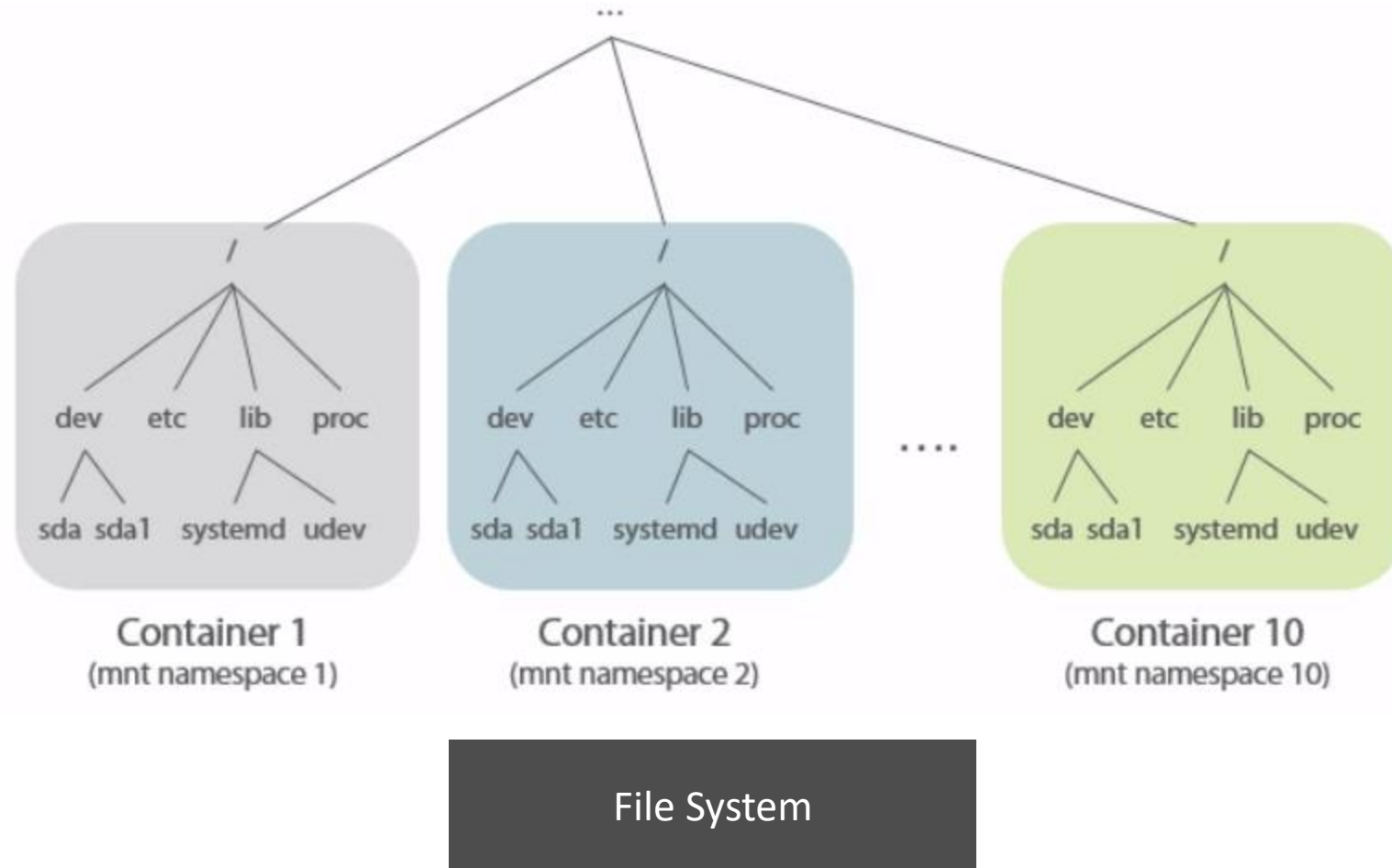


Process Tree

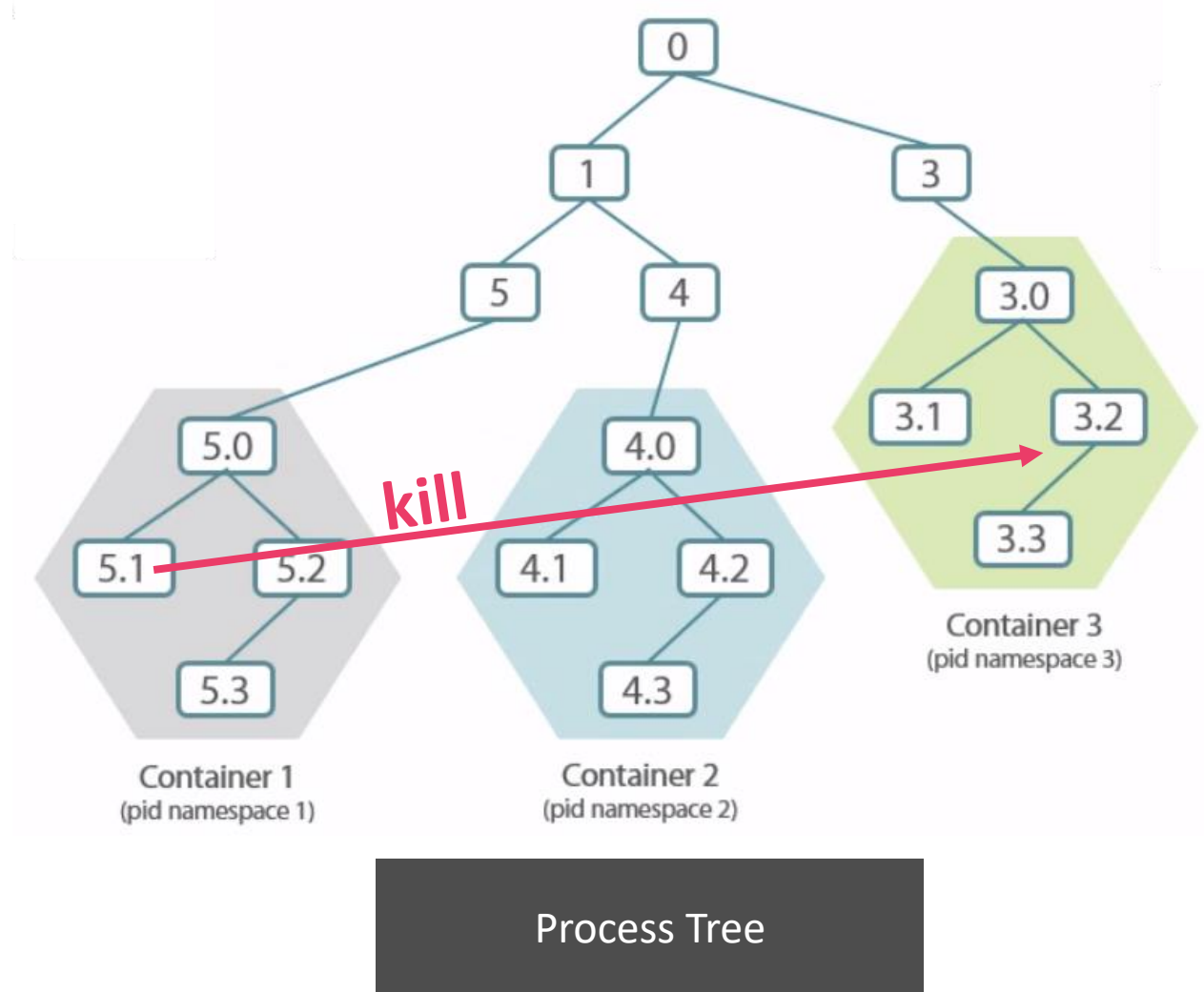


Networking

How Containers Work

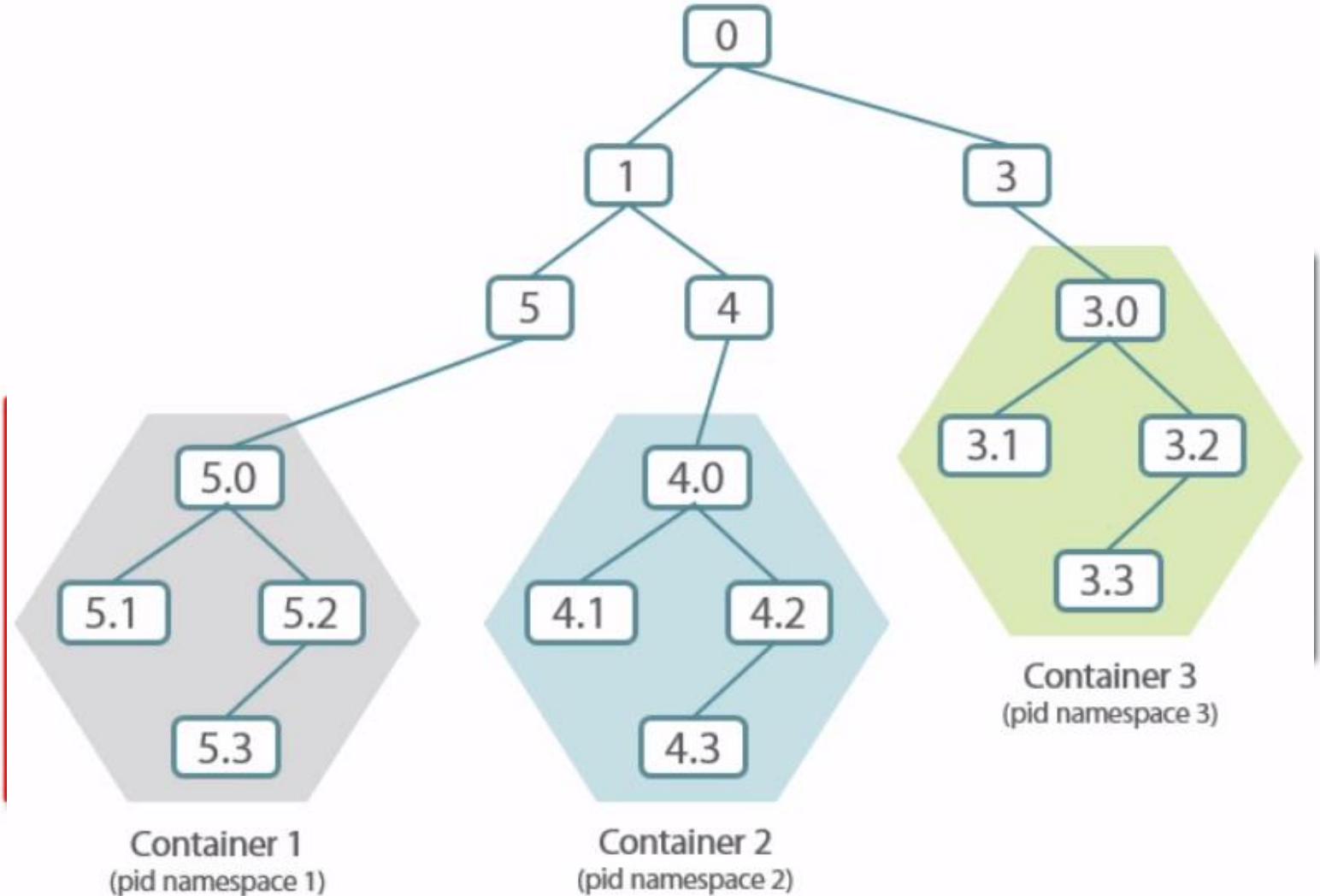


How Containers Work

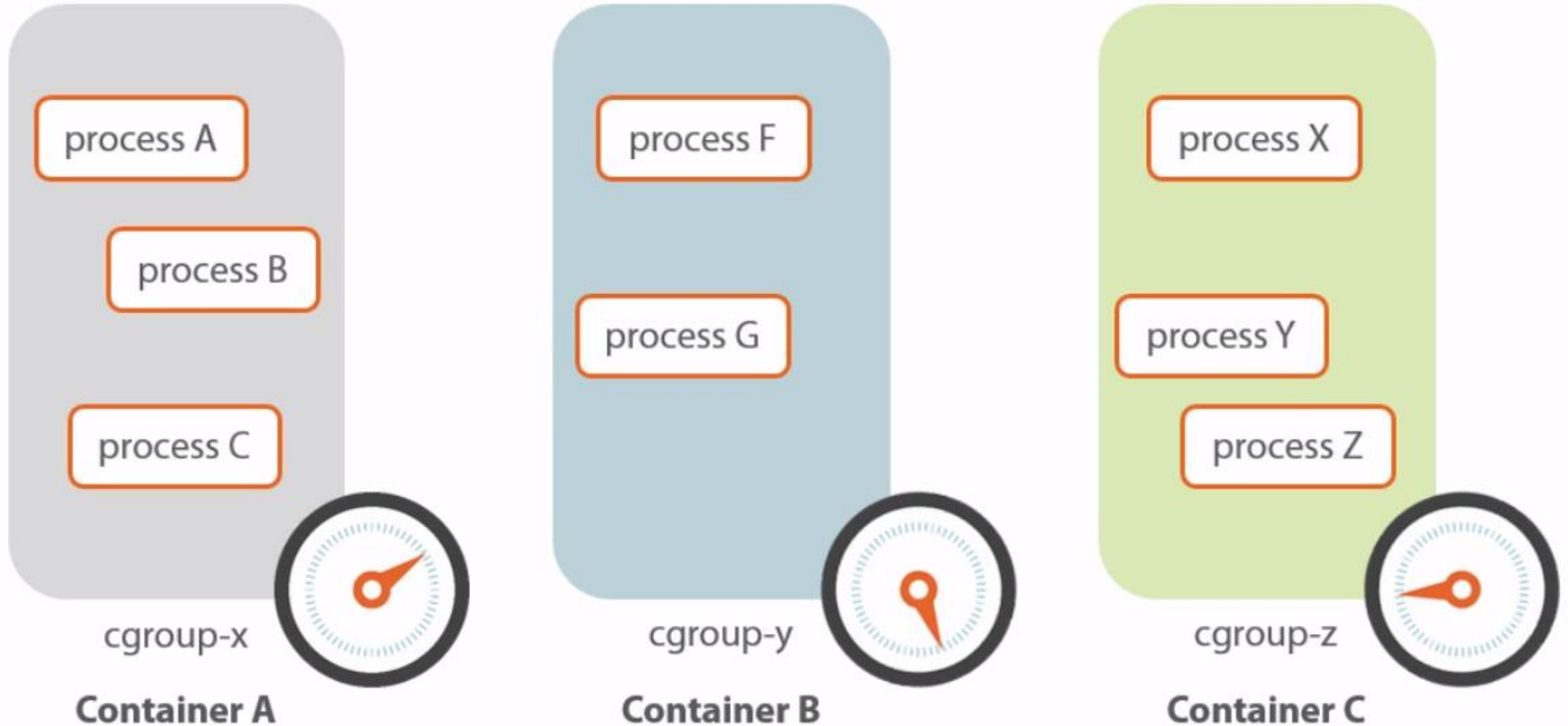


How Containers Work – Kernel Namespaces

The **pid** Namespace



How Containers Work – Control Groups



How Containers Work – Capabilities

root

CAP_AUDIT_CONTROL



CAP_CHOWN



CAP_DAC_OVERRIDE



CAP_KILL



CAP_NET_BIND_SERVICE



CAP_SETUID



.



Personal Preparation

- Knowledge
- Experience

