Contain Yourself

Intro to Docker and Containers

Nicola Kabar

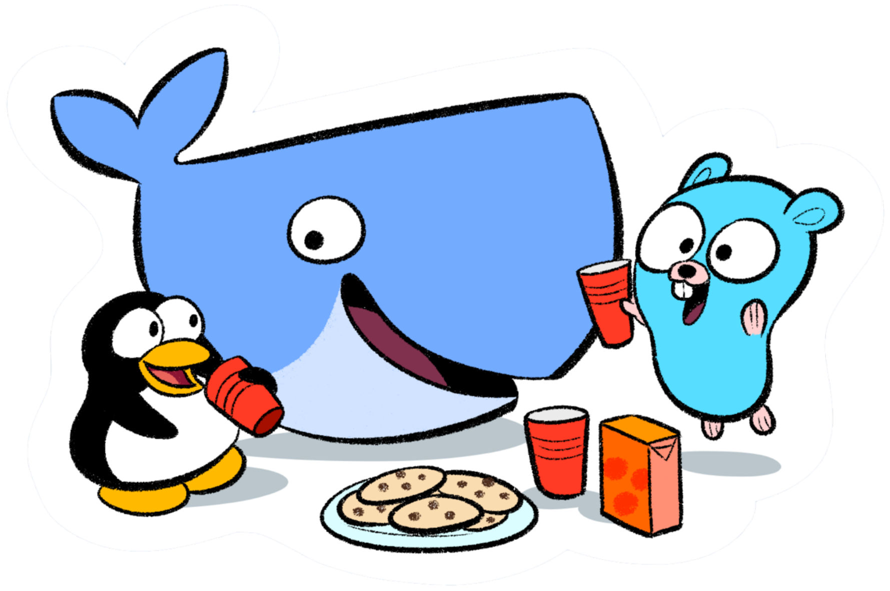
- @nicolakabar || nicola@docker.com
- Solutions Architect at Docker
- Help Customers Design Solutions based on Docker
- <3 Python, RESTful APIs, Containers

Mano Marks

- @ManoMarks || mano@docker.com
- Developer Relations Director at Docker
- Help Developers use Docker
- <3 Geographic Information Systems, Mobile, and Containers

Agenda

- Background Info
- What is Docker ?
- How Does Docker Work?
- Docker In Action (Demo!)
- Why Docker?
- Getting Started
- Q/A



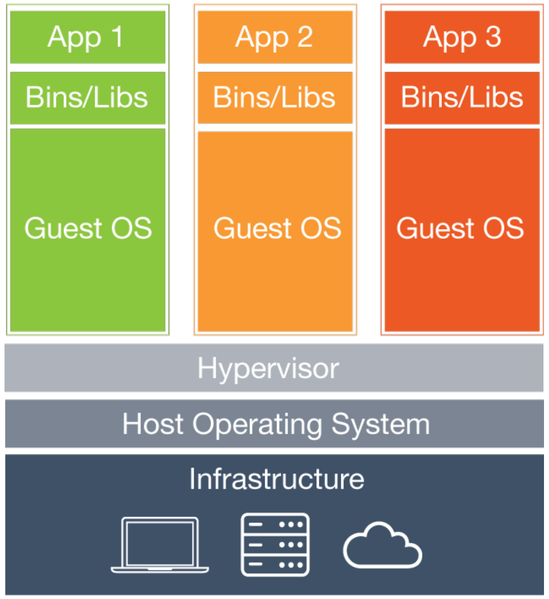
Let's start with some history….

Traditional Architecture (pre-2000)

- One Server = One Application
- Single Stack = Single Language
- More compute = More servers
- Expensive, Slow, Inefficient

Virtualization (2000s)

- One Server = Multiple VMs = Multiple Stacks = Multiple Applications
- More compute = More VMs
- 10s of VMs per Server
- Enabled Cloud Computing



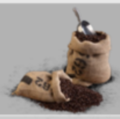
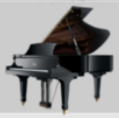
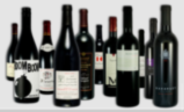
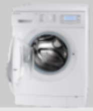
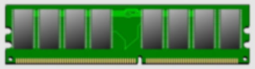
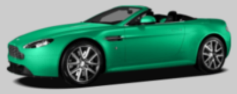
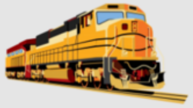
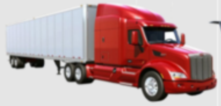
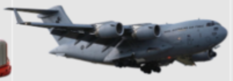
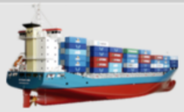
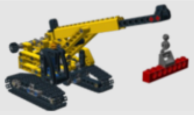
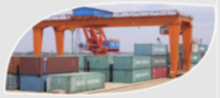
GREAT ☺

But it's Complex,Heavy,and Expensive!

The Matrix From Hell

	?	?	?	?	?	?
NODE JS	?	?	?	?	?	?
MySQL RAILS	?	?	?	?	?	?
php	?	?	?	?	?	?
	?	?	?	?	?	?
cassandra	?	?	?	?	?	?

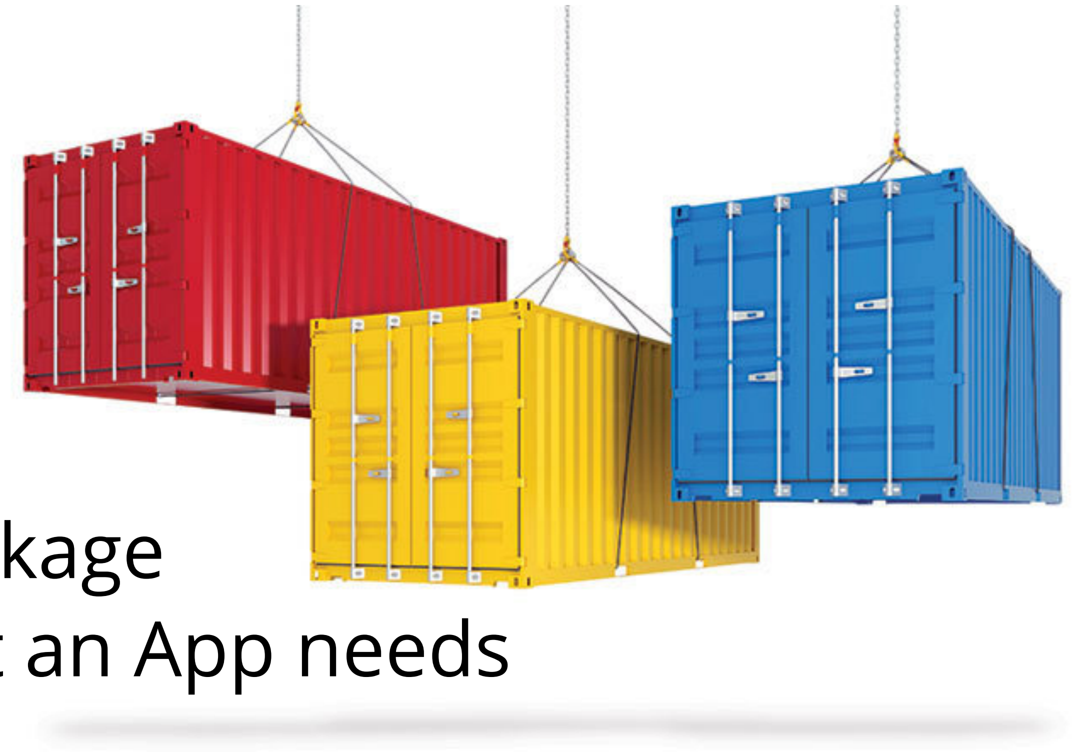
Another Matrix From Hell

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

Solution:
the *intermodal shipping container*

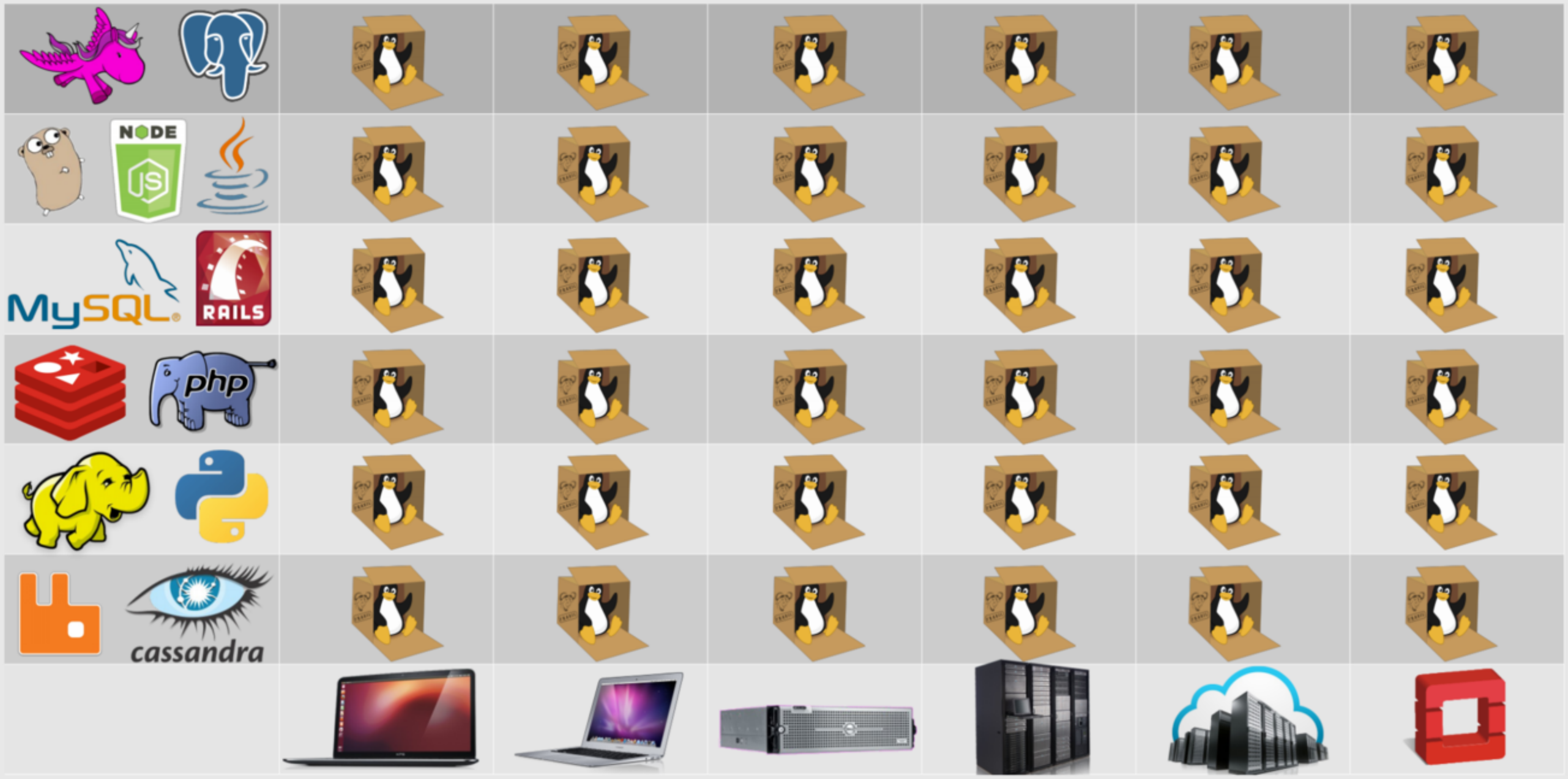
Containerization=

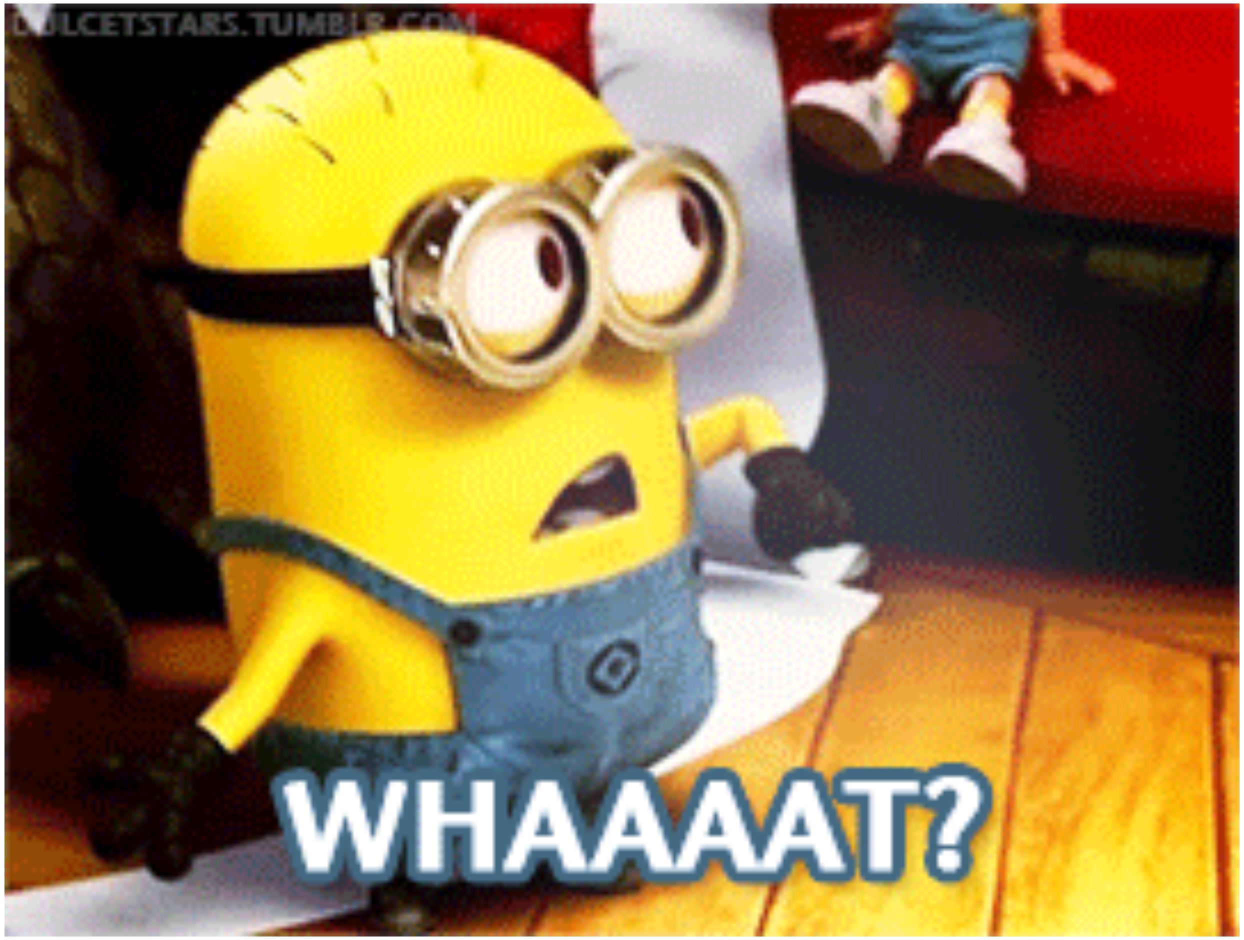
Operating System Virtualization



- Lightweight
- Isolated
- Runnable
- Portable
- New Way to Package **Everything** that an App needs to run

Solved!




DULCETSTARS.TUMBLR.COM
WHAAAAT?

So What Exactly is Docker ?

"open platform to easily ***build,ship,run*** lightweight, portable, self-sufficient app containers anywhere."

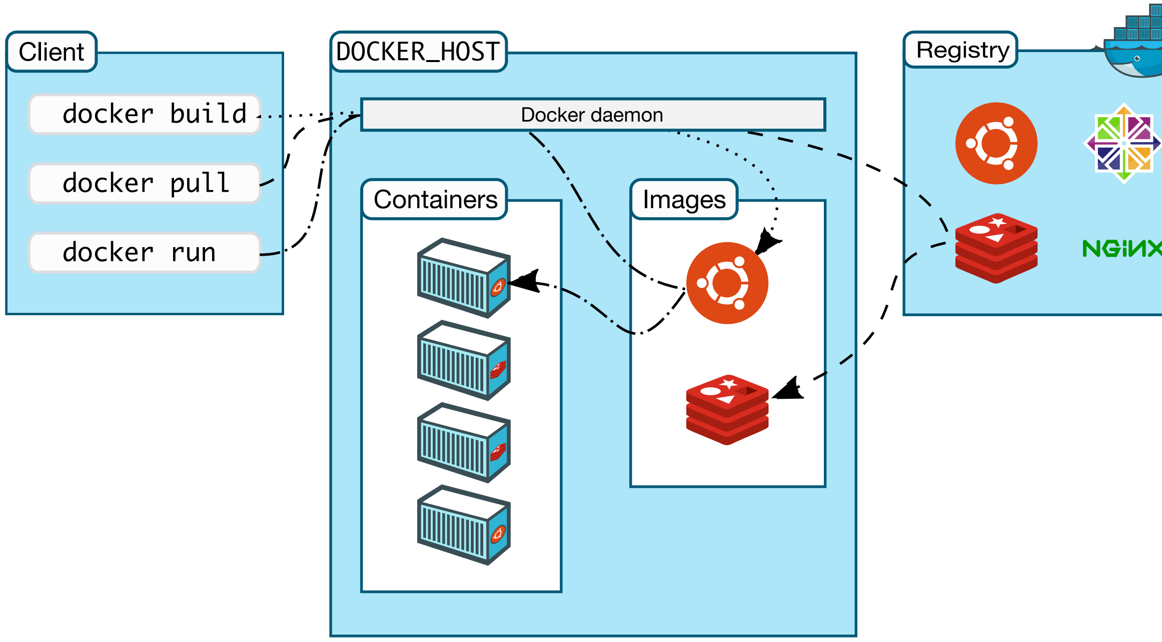
Engine

Client

Images

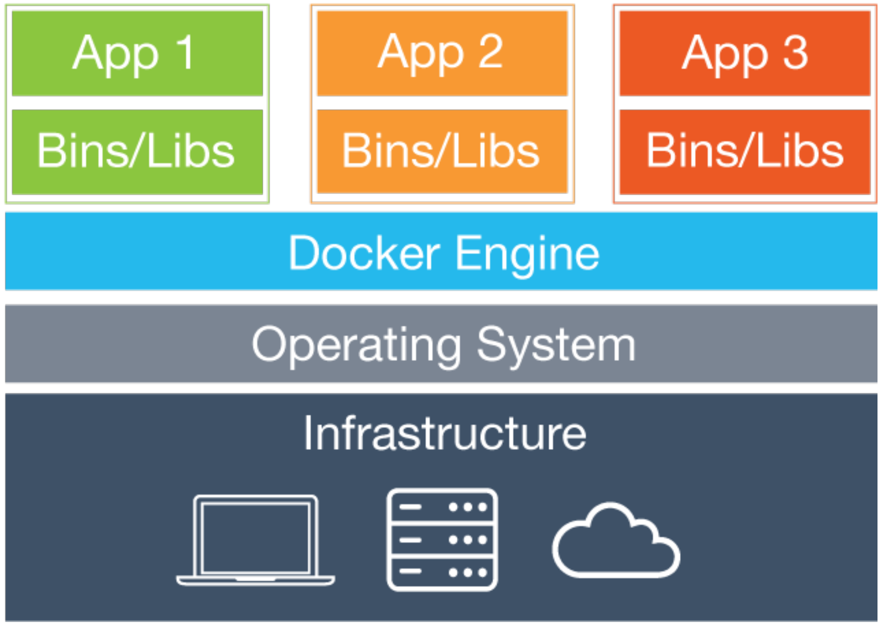
Containers

Registry



Engine

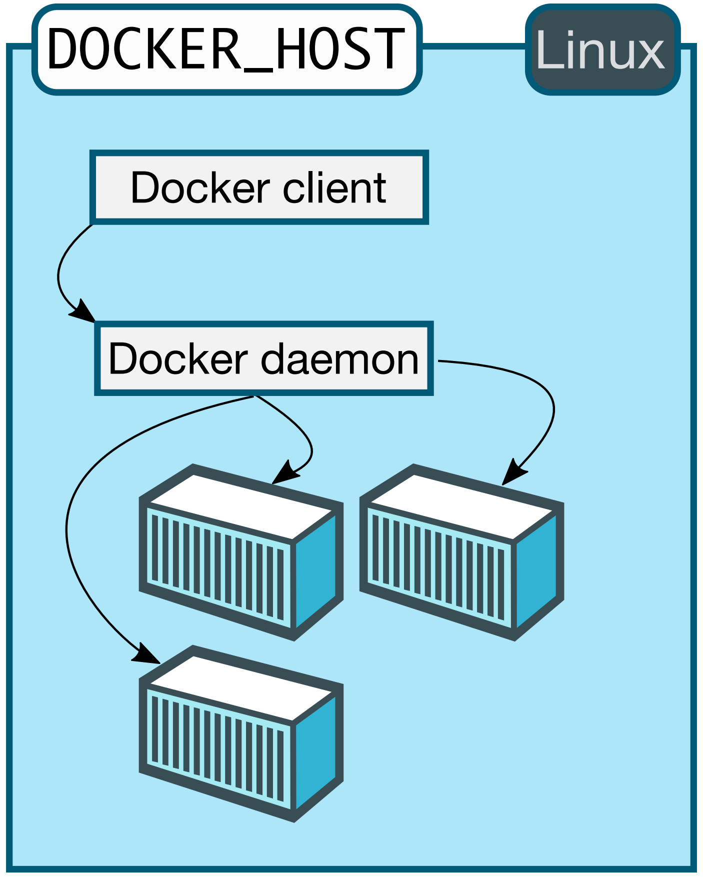
- Daemon on Host
- Linux or Windows*
- Build Images
- Pull Images
- Push Images
- Run Containers
- Manage Containers
- HTTP REST API



*To Be Supported in Windows Server 2016

Client

- Use HTTP
- Installed with Engine
- Local or Remote Calls
- GUI vs CLI



Basic Docker Commands

```
$docker version
Client:
 Version:      1.8.2
 API version:  1.20
 Go version:   go1.4.2
 Git commit:   0a8c2e3
 Built:        Thu Sep 10 19:19:00 UTC 2015
 OS/Arch:      linux/amd64

Server:
 Version:      1.8.2
 API version:  1.20
 Go version:   go1.4.2
 Git commit:   0a8c2e3
 Built:        Thu Sep 10 19:19:00 UTC 2015
 OS/Arch:      linux/amd64
```

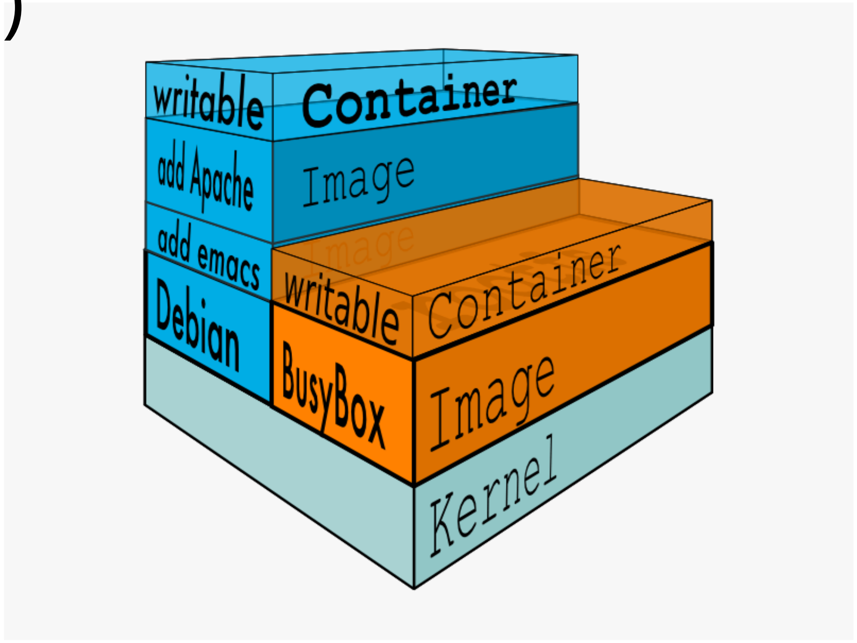
```
$docker info
Containers: 15
Images: 220
Storage Driver: aufs
 Root Dir: /var/lib/docker/aufs
 Backing Filesystem: extfs
 Dirs: 250
 Dirperm1 Supported: true
Execution Driver: native-0.2
Logging Driver: json-file
Kernel Version: 3.19.0-26-generic
Operating System: Ubuntu 14.04.3 LTS
CPUs: 1
Total Memory: 993.2 MiB
Name: dev1
ID: OXRP:6VZL:PLXK:Y7SU:EFEI:2KF5:PILP:UKXH
Debug mode (server): true
```

```
$docker run busybox /bin/sh -c "while true; do echo Hello World; sleep 1; done"
Hello World
Hello World
Hello World
Hello World
```

```
$docker ps
CONTAINER ID        IMAGE      COMMAND                   CREATED            STATUS            P
04a335b35403        busybox   "/bin/sh -c 'while tr"   1 seconds ago     Up 1 seconds
```

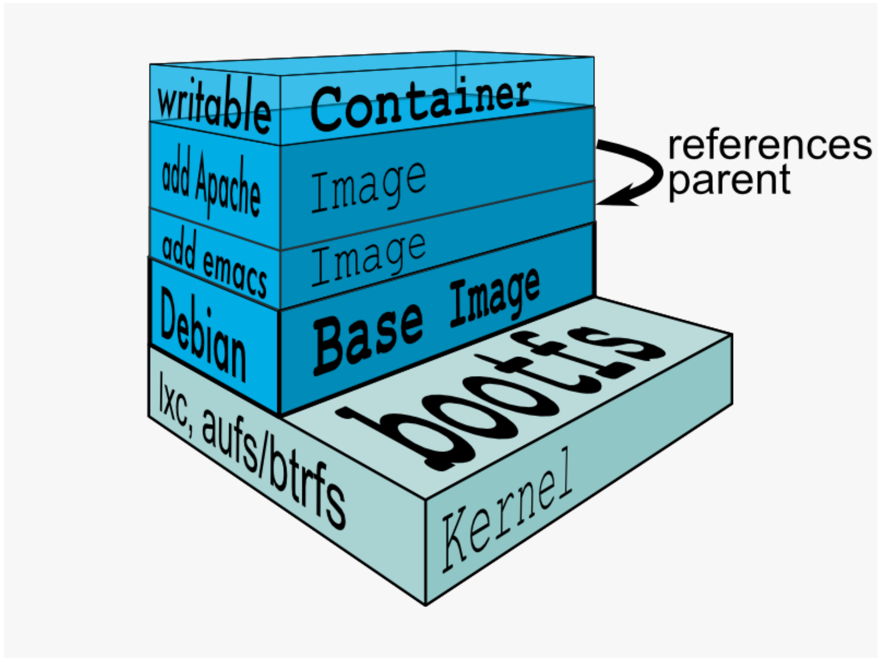
Images

- Read-Only Collection of Files
- Parent Image
- Base Image (OS-like)
- Immutable + Reusable
- Union File System (UFS)



Containers

- "VM-like"
- Run Isolated Processes in Read-Write Layer
- Created from an Image
- Copy-On-Write (COW)



How is it Possible?

- Linux Kernel Features --> **Isolation**
  - namespaces:
    - pid,user,network,ipc --> What Can You **See**
  - cgroups:
    - cpu,memory, disk I/O -->What Can you **Use**
- UFS + COW --> **Speed** + **Disk Utilization**

Building Images

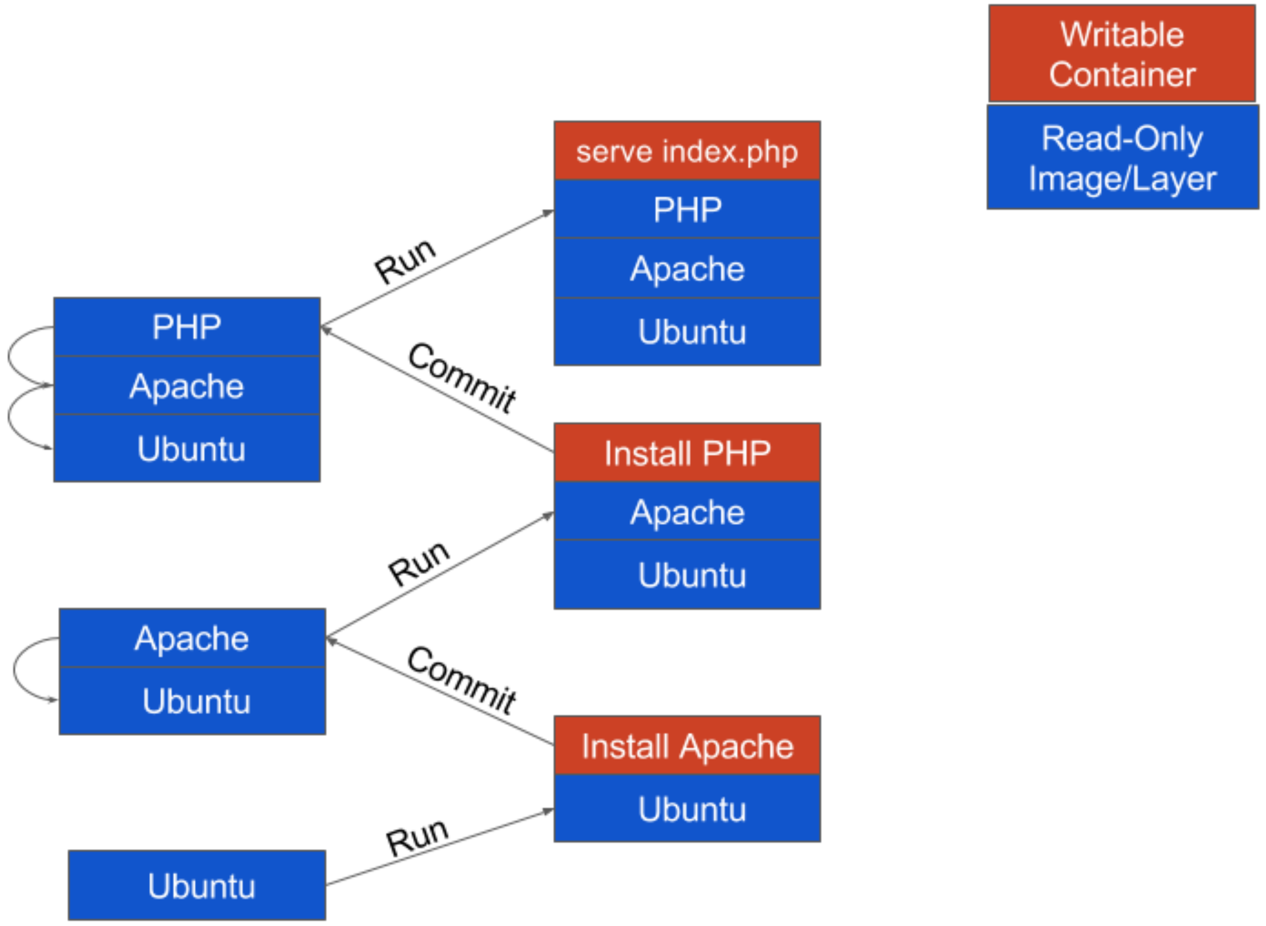
- Process of creating, altering, and committing containers
- Three Options:
  - Manual Run + Commit
  - Import tarball
  - Dockerfile

```
docker run -it ubuntu:12.04
root@05bfafc8e5a8:/# apt-get -y install apache2
...
<snippit>
....
root@05bfafc8e5a8:/# exit
docker commit 05bfafc8e5a8 myimage
1ae55d7aacc0ca202
```

```
# A basic Apache+PHP Image
FROM ubuntu:12.04

MAINTAINER Nicola Kabar version: 0.1

RUN apt-get update && apt-get install -y apache2

RUN apt-get install -y php5

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

RUN rm -rf /var/www/*
ADD index.php /var/www/

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

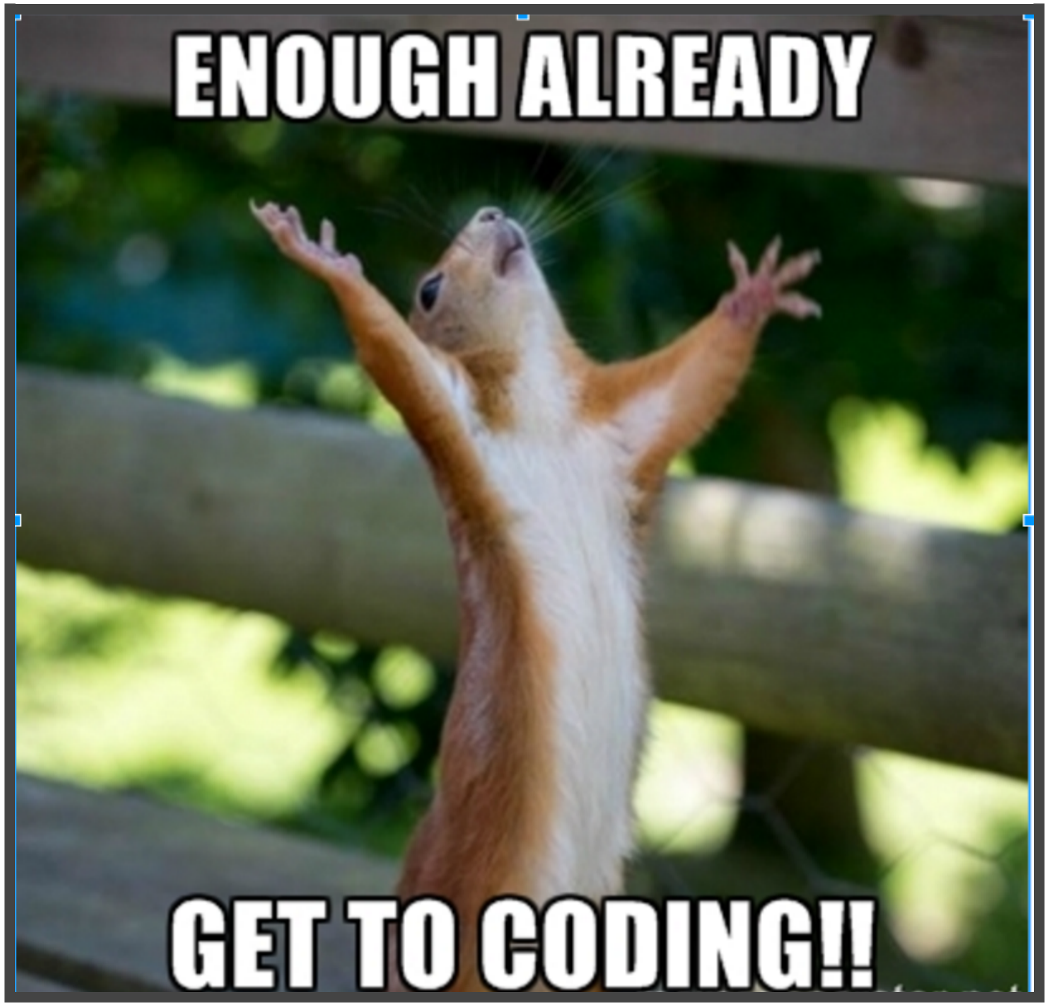
```
$docker build -t myimage .
```

Image Building Process



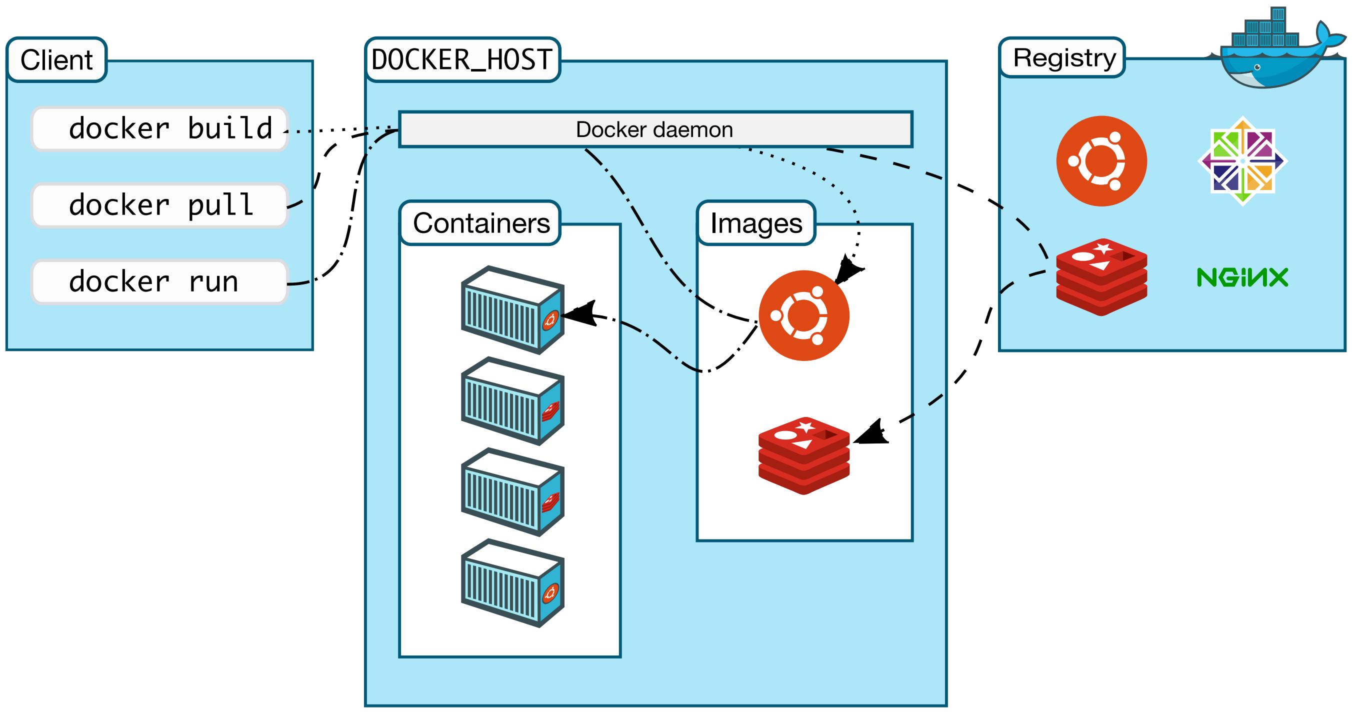
Registry

- Image Distribution
- Cloud Version: Docker Hub (hub.docker.com)
- Official Images
- Team Collaboration
- Workflow Automation


ENOUGH ALREADY
GET TO CODING!!

Docker In Action

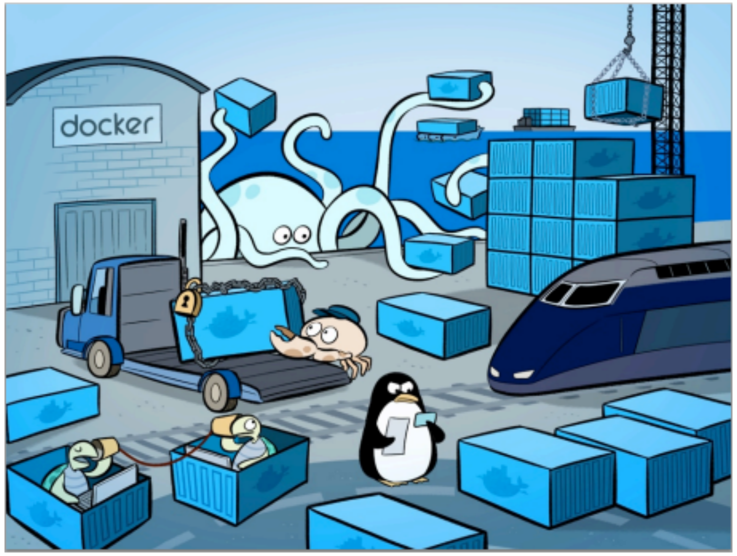
1. Run a Container
2. Build An Image from Dockerfile
3. Push the Image to Docker Hub
4. Pull the Image from Docker Hub
5. Run a Container from the Image



so why Docker ?

Docker Stats

- Written in Go(lang)
- Open-sourced in March,2013 (github.com/docker/docker)
- 1300+ Contributors
- Docker Jobs Openings 43,000+ (*)
- 150,000+ Dockerfiles on Github
- 90,000+ Repositories on Docker Hub
- 100s of Millions of Images Downloaded
- Millions of Developers Use It
- 90+ Official Images



*source: indeed.com

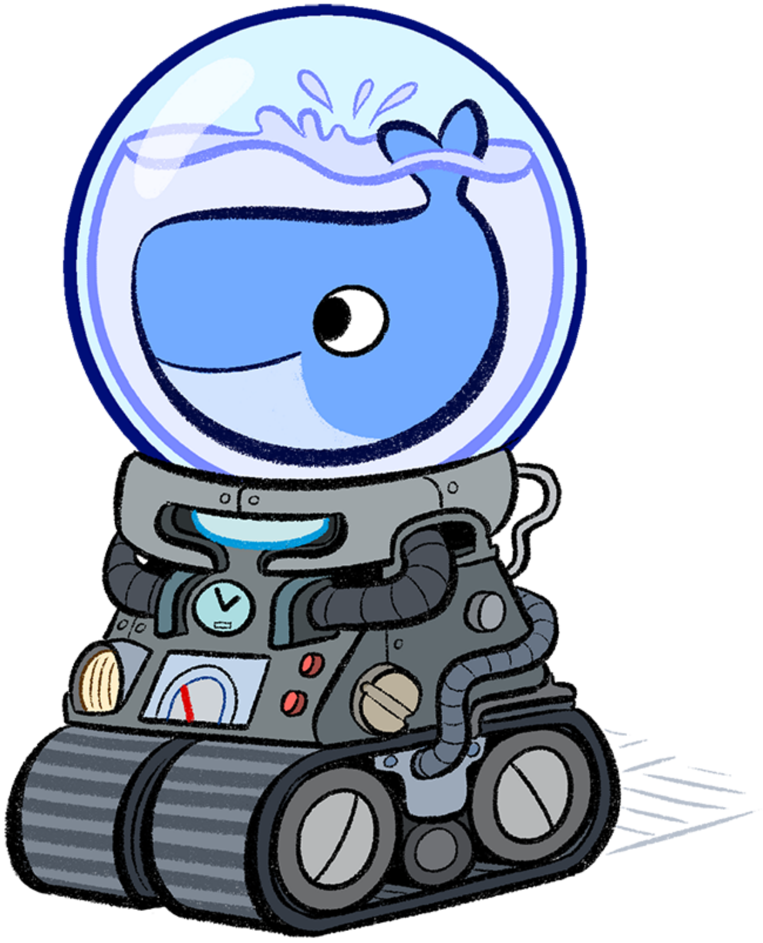
Killer Features of Docker

- Speed
- Lightweight
- Reliable Deployment
- Portability
- …

Changing Software Architecture

Use Cases

- Development
- Testing
- Production

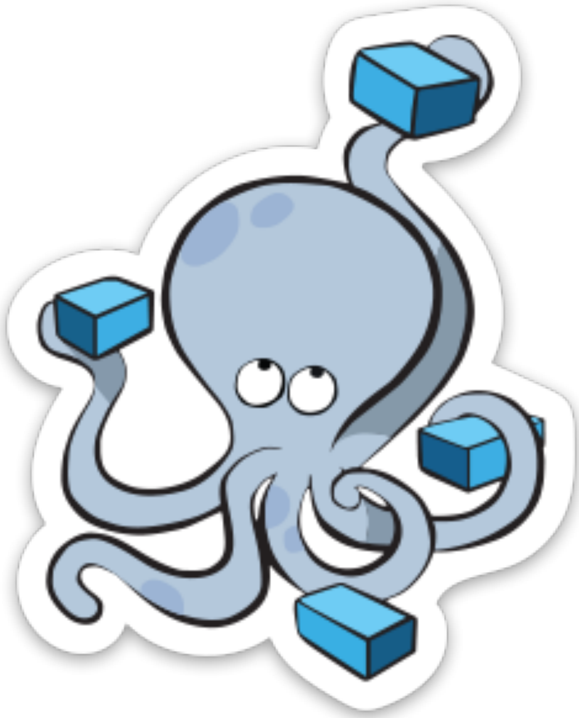


bioboxes

a standard for creating interchangable
bioinformatics software containers

- Docker registry of research algorithms
- Research reproducibility
- Standard interface for DNA Analysis

Getting Started with Docker

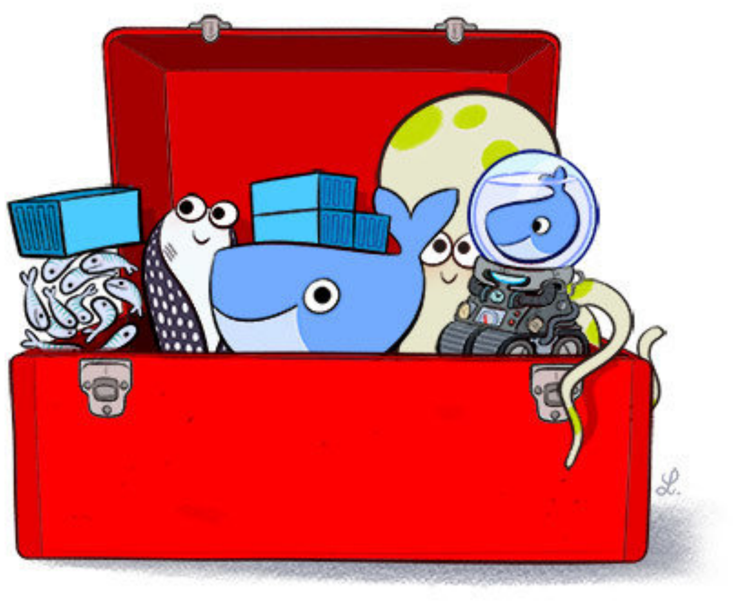


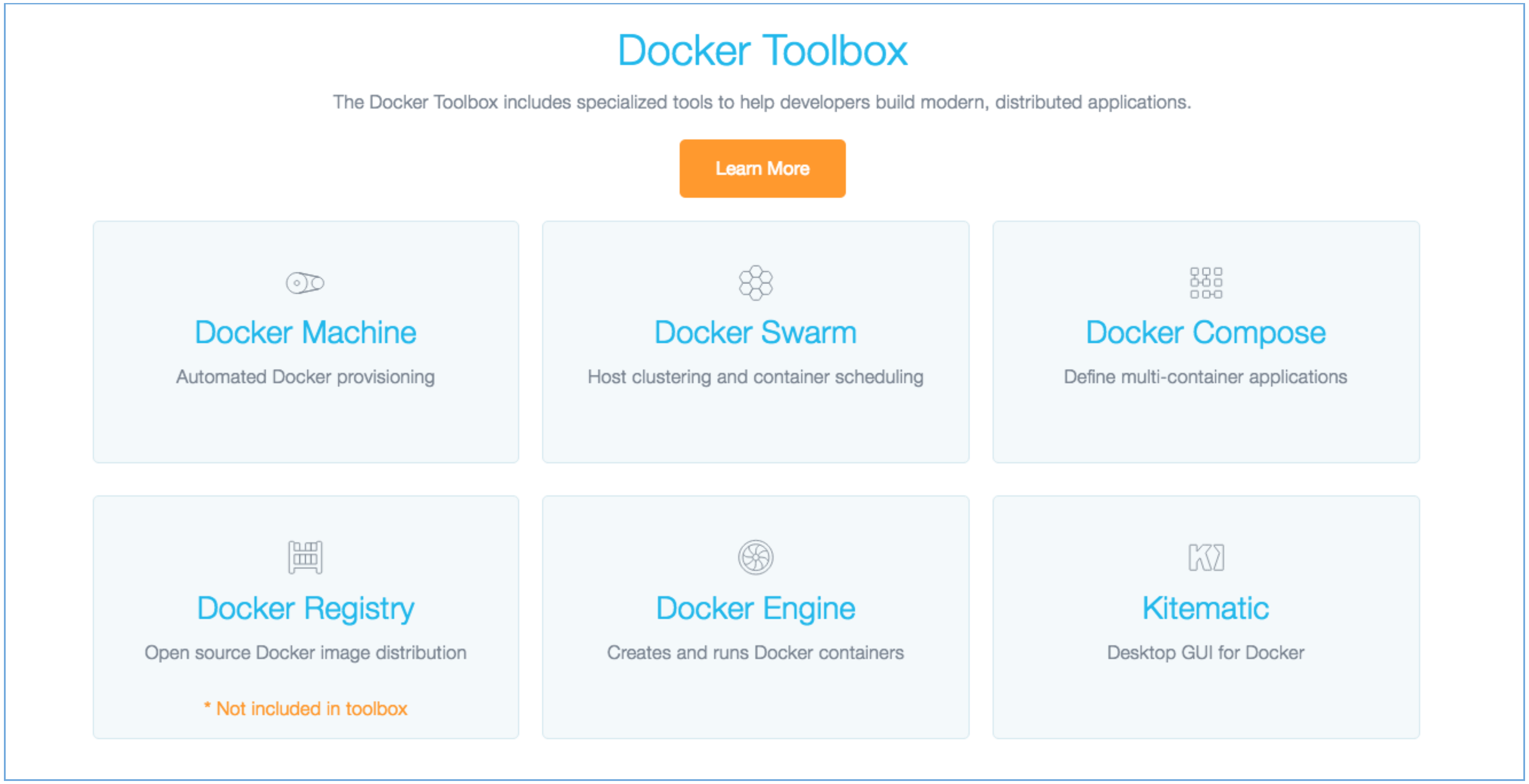
Installing Docker Engine and Docker Toolbox

Linux

Mac/PC

```
#ubuntu/debian --> apt-get
apt-get install docker-engine

#rhel/centos --> yum
yum install docker-engine
```





Docker Toolbox

The Docker Toolbox includes specialized tools to help developers build modern, distributed applications.

Docker Machine

Automated Docker provisioning

Docker Swarm

Host clustering and container scheduling

Docker Compose

Define multi-container applications

Docker Registry

Open source Docker image distribution

* Not included in toolbox

Docker Engine

Creates and runs Docker containers

Kitematic

Desktop GUI for Docker

Plot Twist!

Presenting from a Docker Container!

```
$docker run -d -P nicolaka/cs50
```

Thank you!

@nicolakabar

@manomarks

docker

www.docker.com

Slides:
```
$docker run -d -P nicolaka/cs50
```