

WHAT IS THIS "DOCKER" ?

Jean-Marc Meessen



I had a dream !

- My own copy of the database
 - ... that I can break at will
- My own iso-prod test environment
 - ... that I can break at will
- Easily share my configuration with colleagues.
- DEVOPS !

...And it became true !

PAS DEVOPS 

T'as pas PHP 5.6 installé ?
C'est TON problème mon
p'tit gars !



CommitStrip.com

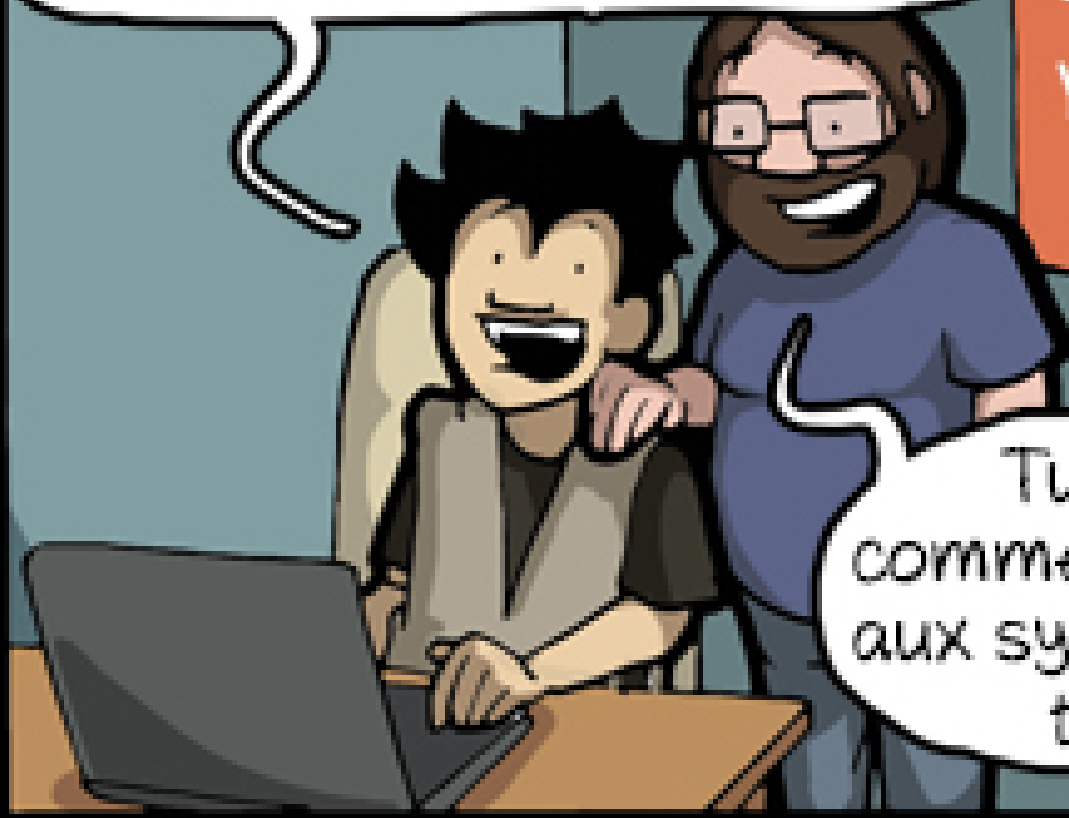
DEVOPS

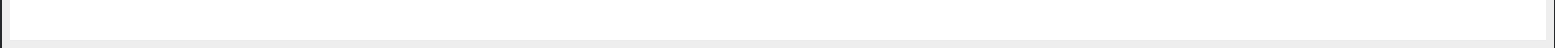


J'ai tout packagé dans une image Docker dispo sur notre repo privé, t'as plus qu'à !

YOU
BUILD IT,
YOU RUN
IT

Tu sais
comment parler
aux sysadmins,
toi !





HELLO !

- Jean-Marc MEESEN
- Brussels, Belgium
- Customer Success Manager @ Cloudbees
 - (Since January)
- Before (@ Worldline)
 - Senior ESB Java Developer
 - Development Infrastructure Expert
 - Mentor



CONTACT INFO



- jean-marc@meessen-web.org
- Twitter: @jm_meessen



AND YOU ?

- Developers ?
- Ops ?
- Security ?
- Managers ?

YOU AND DOCKER ?




- Never heard about it ?
- Some "Proof of Concept" ?
- Use it every day ?
- In Production ?

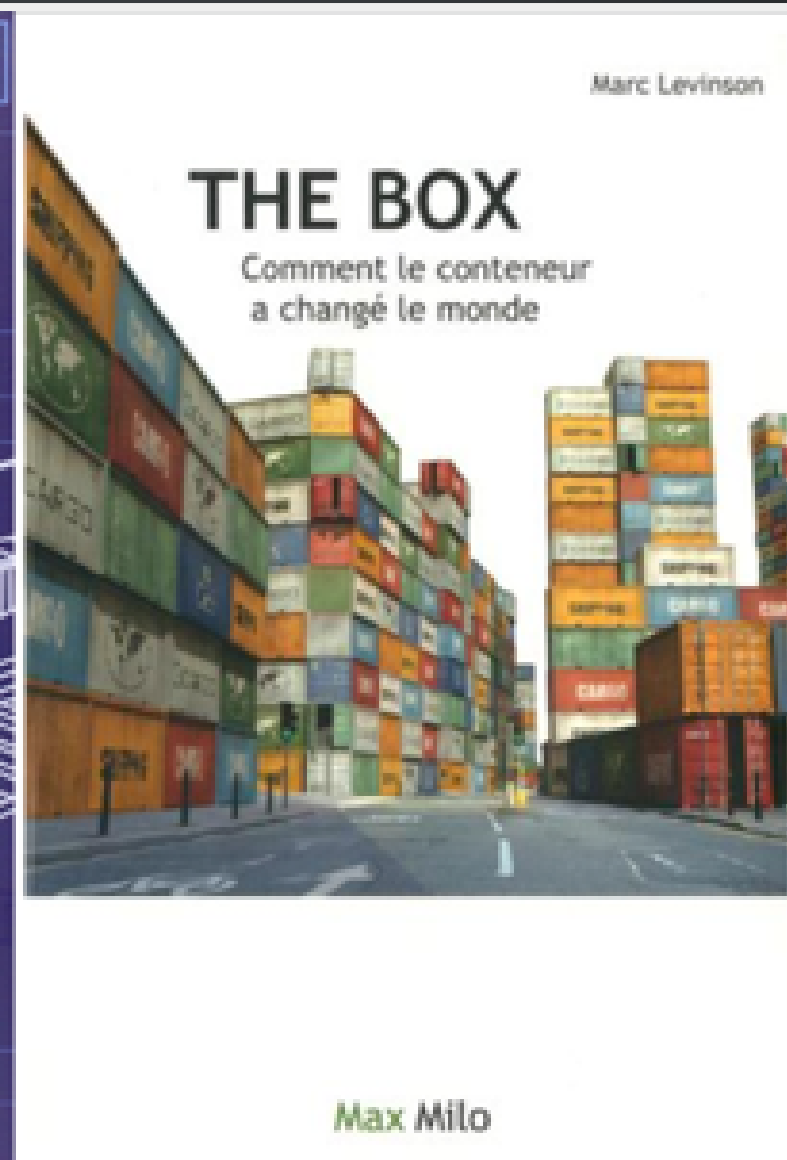
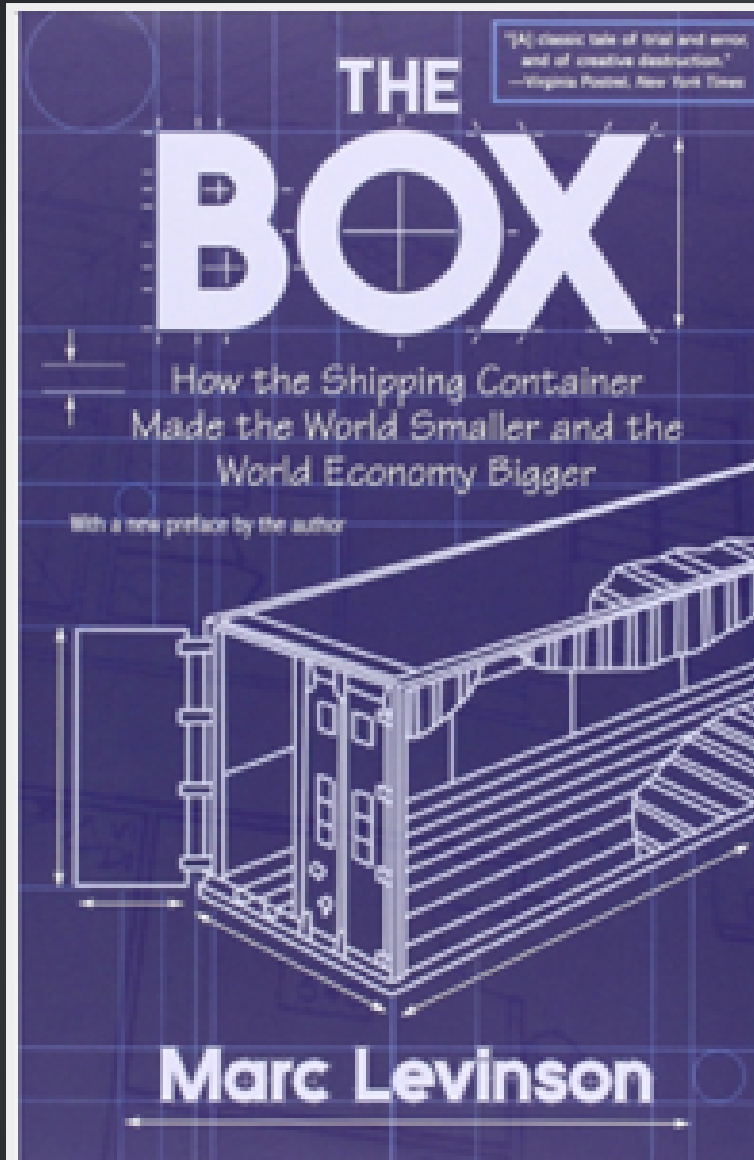
TODAY'S TALK

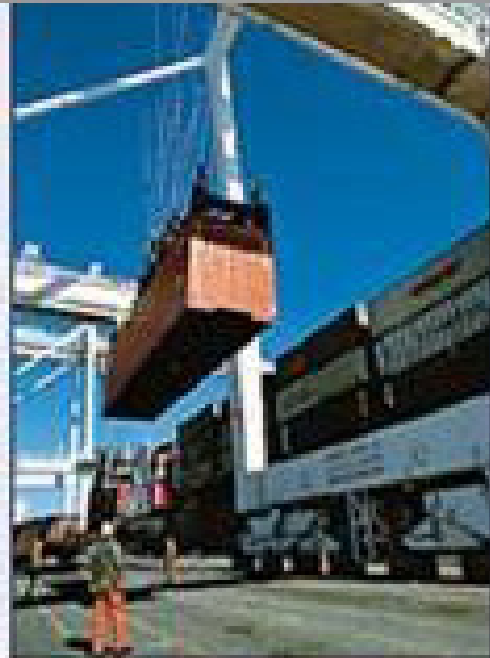
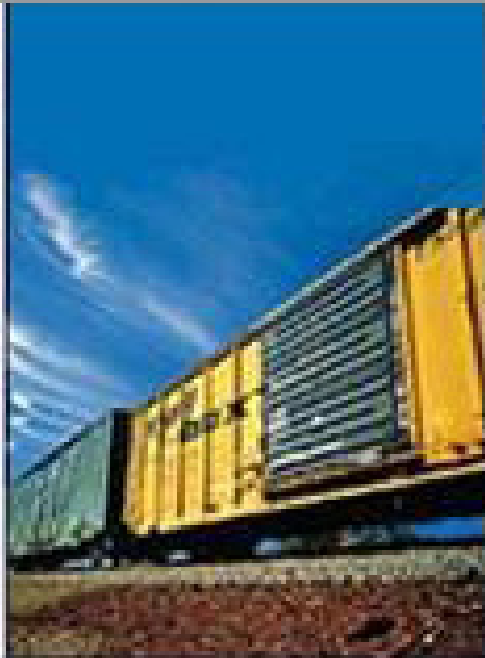
- What are "containers" ?
- How to start ?
- The principles (single container)
- Multiple containers (Docker Compose)
- Multi-hosts (Docker Swarm)














What are "containers" ?



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





	Static website	?	?	?	?	?	?	?
	Web frontend	?	?	?	?	?	?	?
	Background workers	?	?	?	?	?	?	?
	User DB	?	?	?	?	?	?	?
	Analytics DB	?	?	?	?	?	?	?
	Queue	?	?	?	?	?	?	?
		Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers
								

DOCKER CONTAINERS

- is not a virtualization technique,
- rather an **isolation** technology.

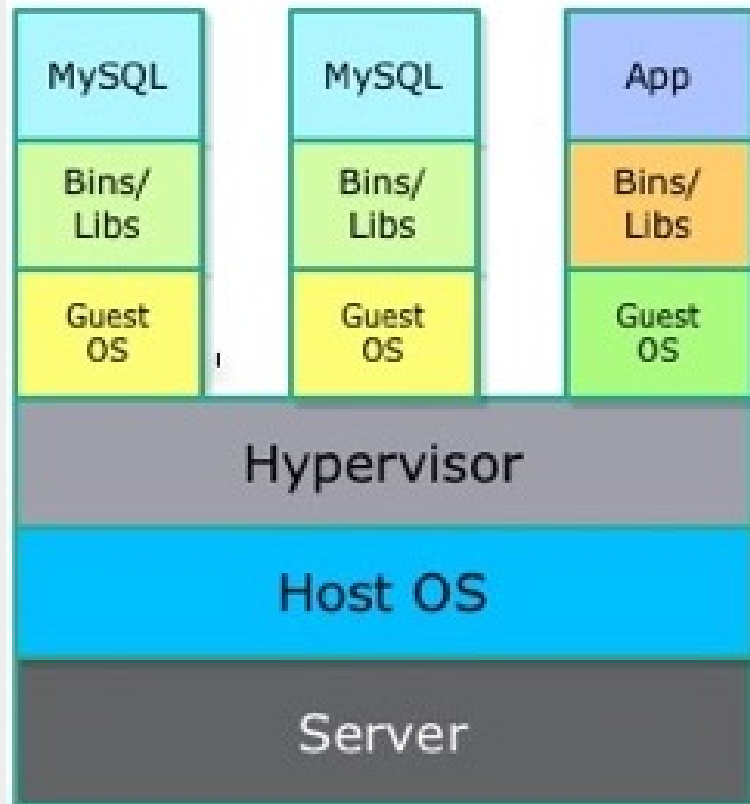
VM



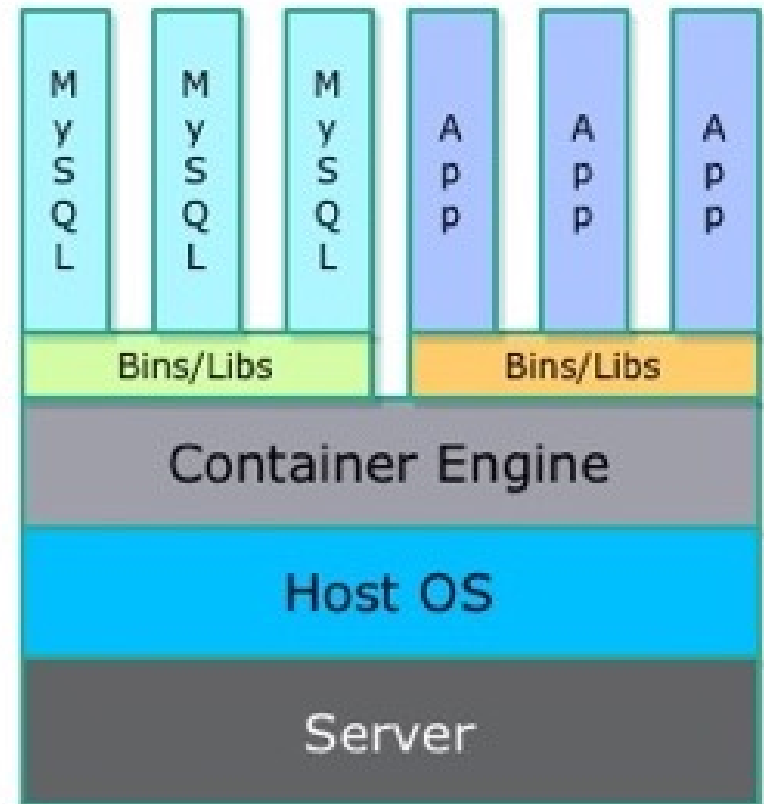
Containers



Virtual Machines



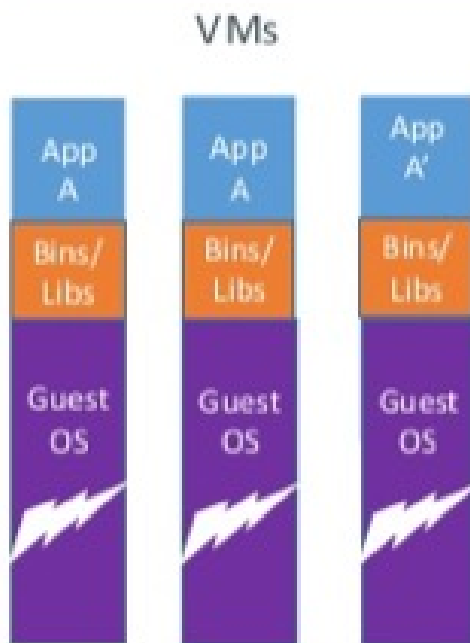
Containers



DOCKER CONTAINERS ARE :

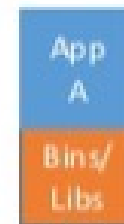
- NameSpaces
 - isolates and virtualizes system resources
 - (process, mounts, networking)
- cgroups (control groups)
 - limits CPU, memory, IOs, etc

Why are Docker containers lightweight?



VMs

Every app, every copy of an app, and every slight modification of the app requires a new virtual server



Original App
(No OS to take up space, resources, or require restart)



Copy of App
No OS. Can share bins/libs



Modified App

Union file system allows us to only save the diffs between container A and container A'

APPLICATIONS PACKAGED WITH SYSTEM DEPENDENCIES

- new packaging paradigm
- one application works on Ubuntu with Python 2
- second application works on Centos 7.2 with Python 3

WHAT DOCKER SOLVES

- Escape the dependencies hell
- Fast iterative Infrastructure improvement
- Container "loader" & Container "shipper"
 - (no more "it worked in Dev, now it's OPS problem")
- easy onboarding of Devs.
- "Own test environment"

The background of the slide is a dense, repeating pattern of colorful shipping containers in various colors including blue, green, yellow, red, purple, and brown. Each container has some faint, illegible text and markings on its side.

How to start ?

SANDBOX

- <http://play-with-docker.com/>
 - Validate captcha
 - 4 hours to play with "Docker instances"
 - Just click "Add a new instance"
 - Machine starts and you have access to the command line

NEED A CONTAINER-ENABLED "KERNEL"

- 64-bit Linux installation
- version 3.10 or higher of the Linux kernel
- recommend using the Docker repositories
 - See the Docker online documentation
<https://www.docker.com/products/overview>

AND ON WINDOWS OR MAC OS X ?

- Installed in a virtual machine (ex VirtualBox)
 - Running Linux
- Ready made bundles:
 - Docker Toolbox (Mac / Windows)
 - Docker for Mac (native, uses **xhyve**)
- Using (corporate) proxies: advanced topic

DOCKER FOR WINDOWS

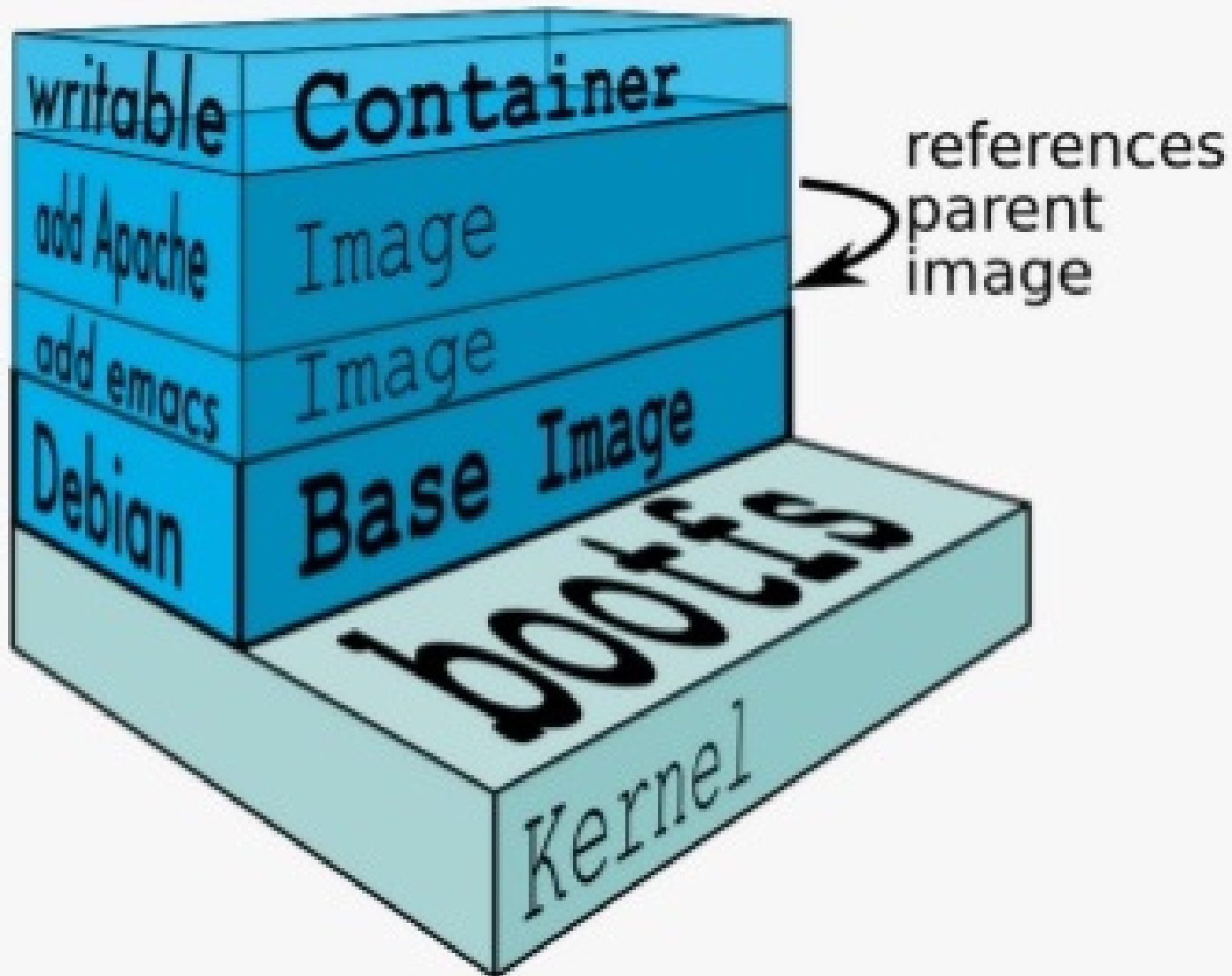
- "on" != "for"
- Runs native Windows applications
- with Hyper-V
- 64bit Windows 10 Pro, Enterprise and Education

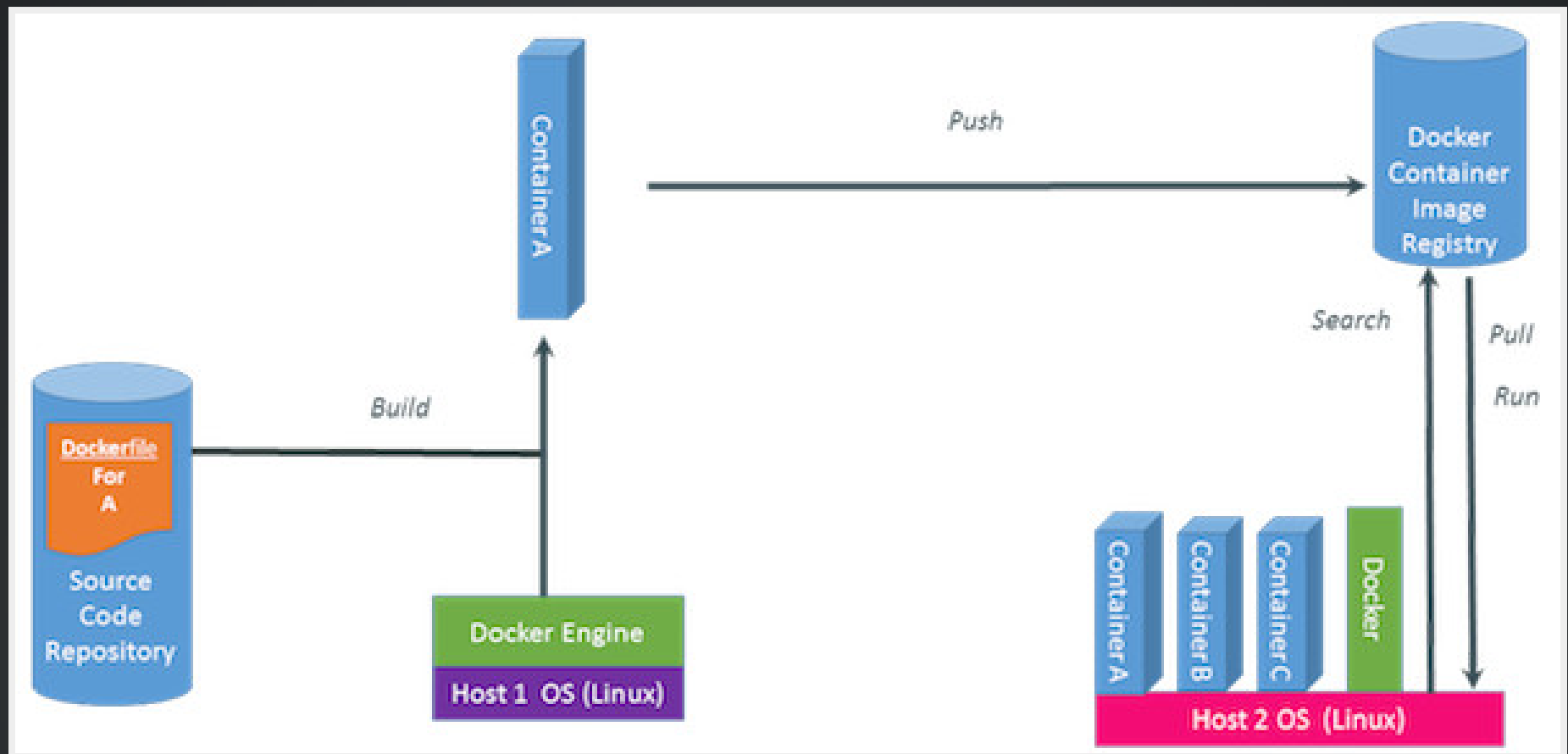
The background of the slide is a dense, repeating pattern of colorful shipping containers in various colors including blue, green, red, yellow, and purple. The containers are stacked in a grid-like fashion, creating a textured, industrial background.

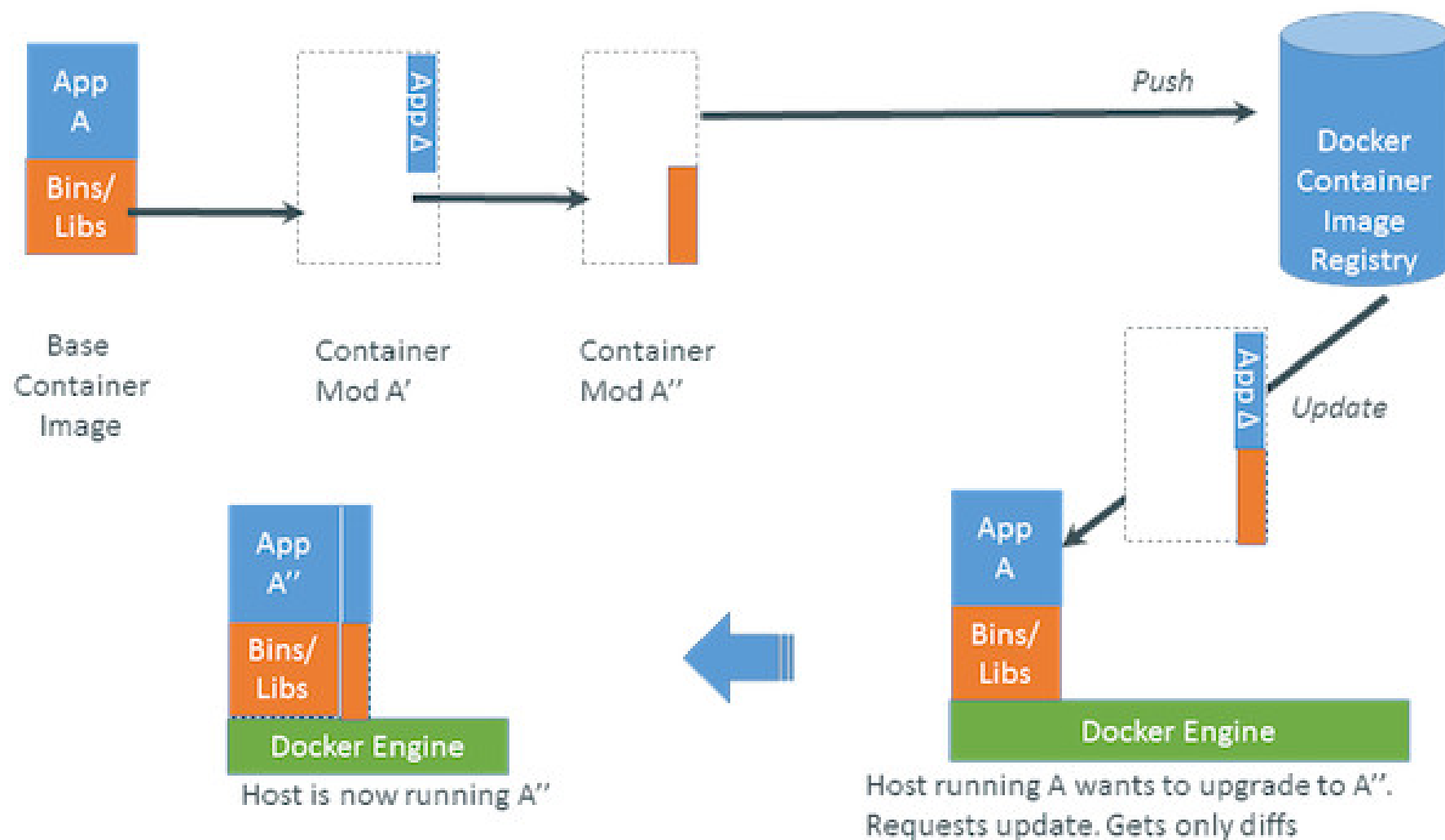
How does it work ?

SOME THEORY....

Docker Filesystem







LET ME SHOW YOU

HOW DO YOU GET IMAGES ?

- Note: an image is immutable
- you get them from
 - DockerHub
 - Corporate Registry
- Or build it yourself

BUILDING A DOCKER IMAGE

- Described in a Dockerfile

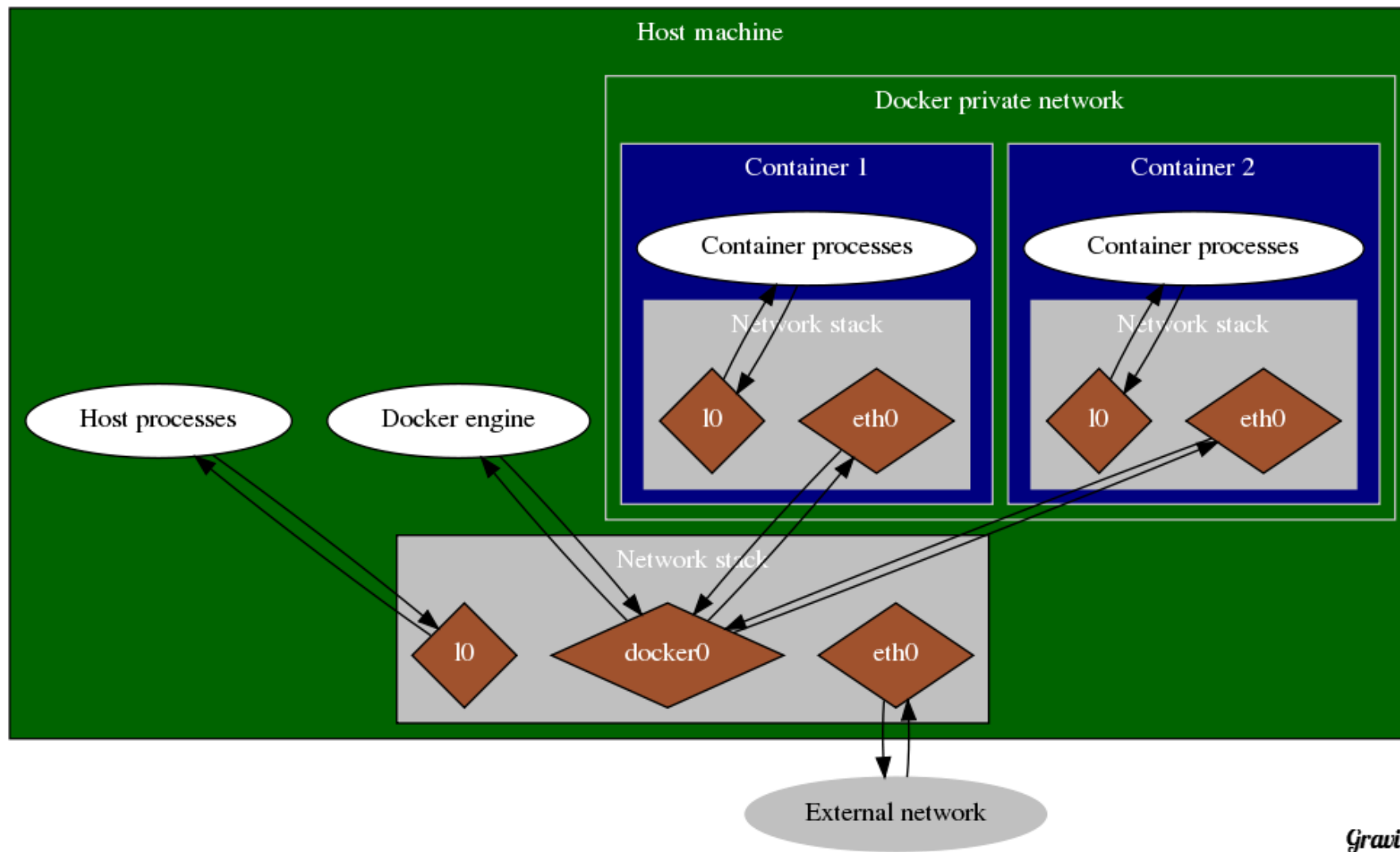
```
FROM ubuntu
MAINTAINER Kimbro Staken
```

```
RUN apt-get install -y software-properties-common python
RUN add-apt-repository ppa:chris-lea/node.js
RUN echo "deb http://us.archive.ubuntu.com/ubuntu/ precise universe" >> /etc/apt/sources.list
RUN apt-get update
RUN apt-get install -y nodejs
#RUN apt-get install -y nodejs=0.6.12~dfsg1-1ubuntu1
RUN mkdir /var/www
```

```
ADD app.js /var/www/app.js
```

```
CMD [ "/usr/bin/node", "/var/www/app.js" ]
```

NETWORKING



STORAGE

- Stateless, statefull, persistence
- Storage drivers
- map a local directory as a volume
- create and share a data volume

Multi-container infra

DESCRIBE A COMPLETE INFRASTRUCTURE

- Complex systems
 - Fuse ESB server
 - MQ series servers
 - Oracle database
- Use "docker-compose"

DOCKER-COMPOSE

- one place to define
 - your components
 - how to (docker) build them
 - what container should start first
 - networks (who can talk to whom)
 - (data) volumes
 - Security restrictions
 - Etc.

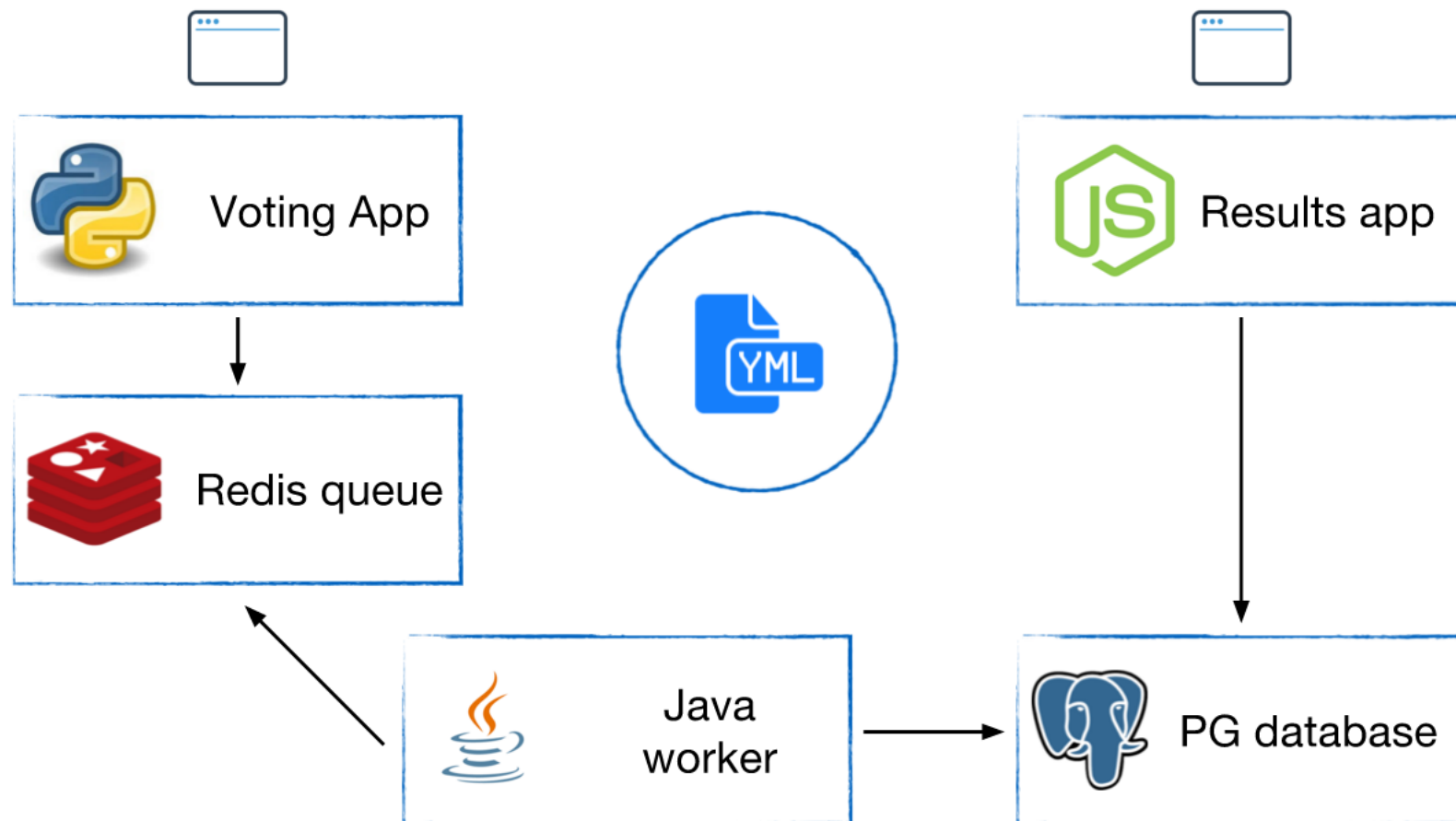
DOCKER-COMPOSE

- three step process
 - dockerfile of individual images
 - Define services
 - Run parameters
 - Network Relationships
 - dependencies
 - `docker-compose up`

DOCKER-COMPOSE

- Commands to manage the whole application lifecycle
 - Start, stop and rebuild services
 - View the status of running services
 - Stream the log output of running services
 - Run a one-off command on a service

DEMO

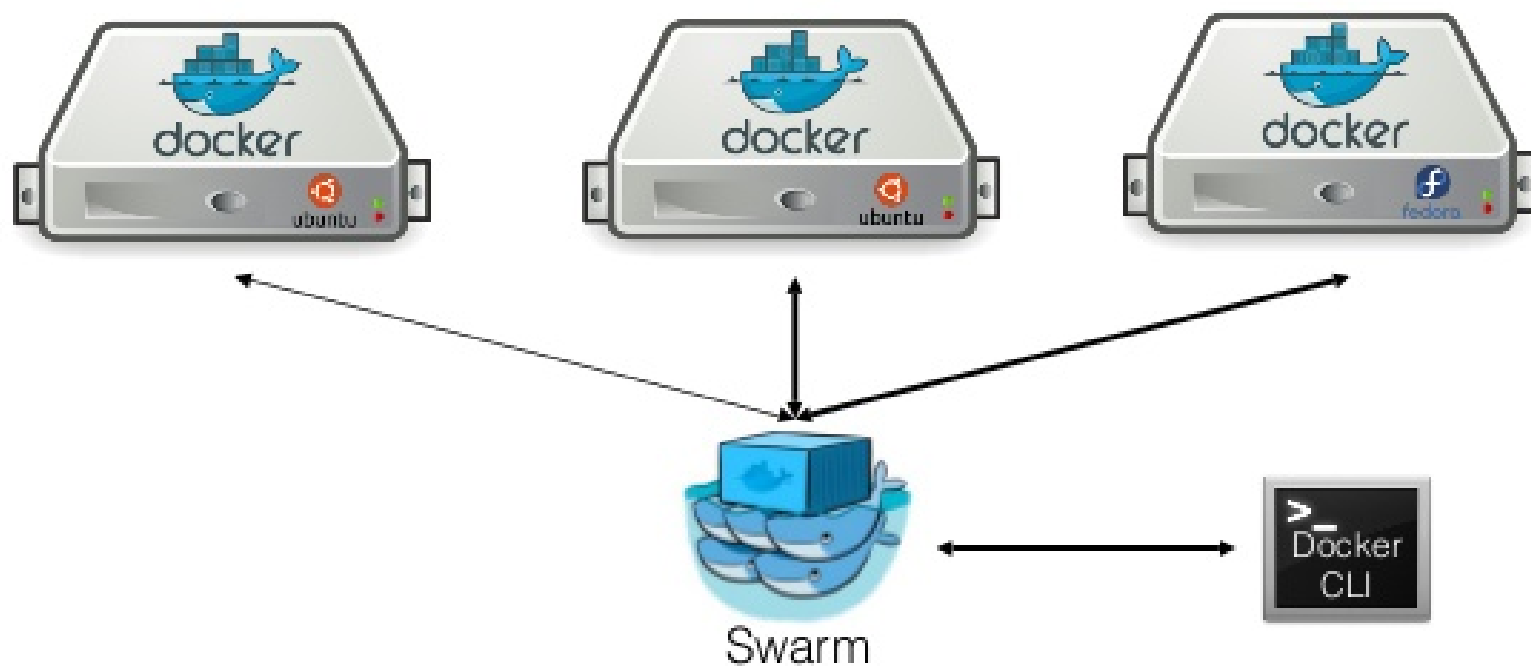


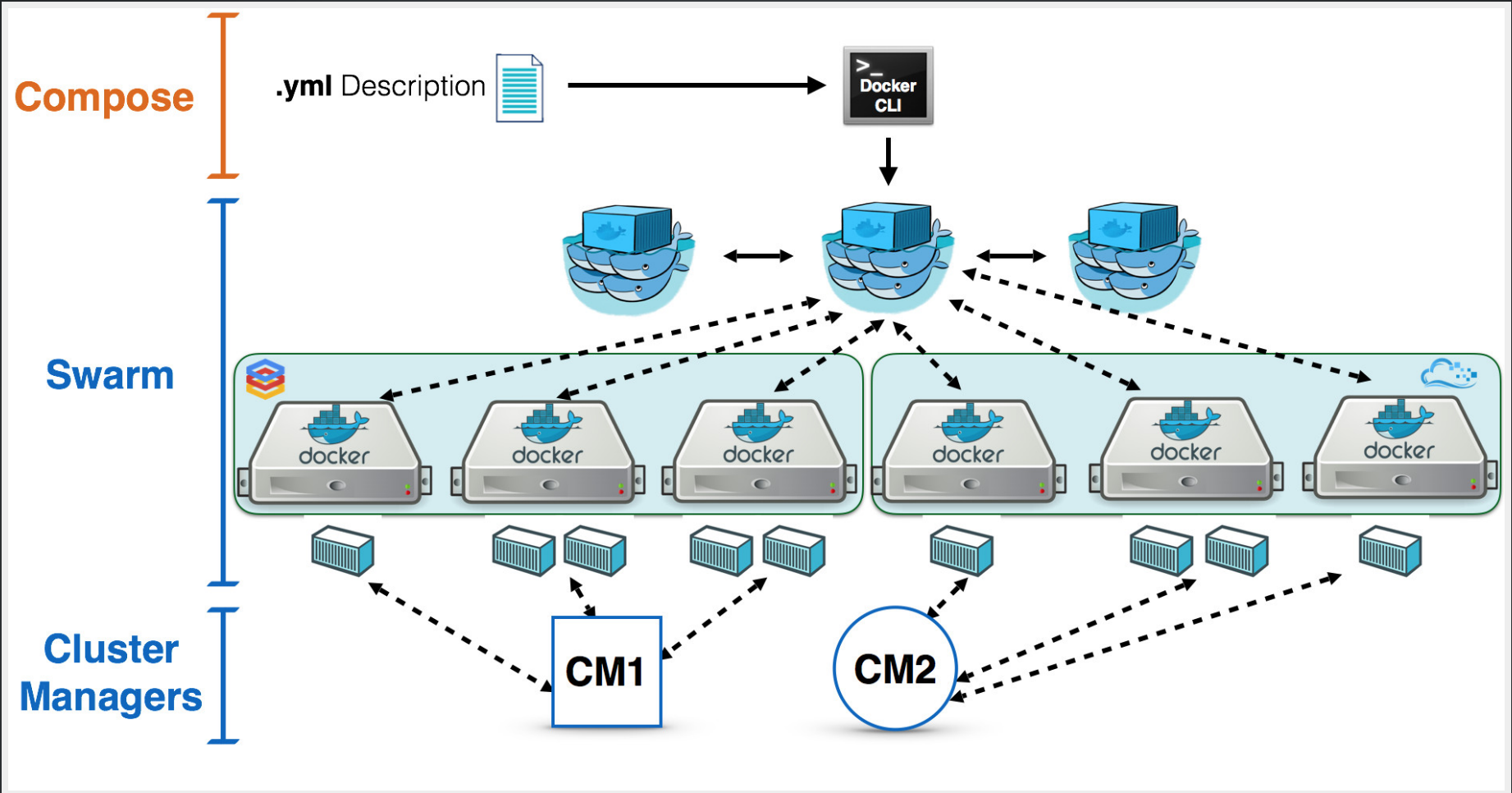
TRY IT

- connect to wifi **PI-LAB** with password **raspberrry**
- in browser, connect to **<http://rpi3-1.local:5000>** to vote.
- connect to **<http://rpi3-1.local:5001>** to view results

Multi-node infra

With Docker Swarm





HOW TO LEARN ?

- Many tutorials available on-line
- <https://training.docker.com/category/self-paced-online>
 - Developer
 - Beginner Linux Containers
 - Beginner Windows Containers
 - Intermediate (both Linux and Windows)
 - Operations
 - Beginner
 - Intermediate

HOW TO LEARN ?

- Docker playground
 - <http://play-with-docker.com/>

The background of the slide is a dense, repeating pattern of colorful shipping containers in various colors including blue, green, red, yellow, and purple. The containers are stacked in a grid-like fashion, creating a textured, industrial background.

Where is Docker heading?

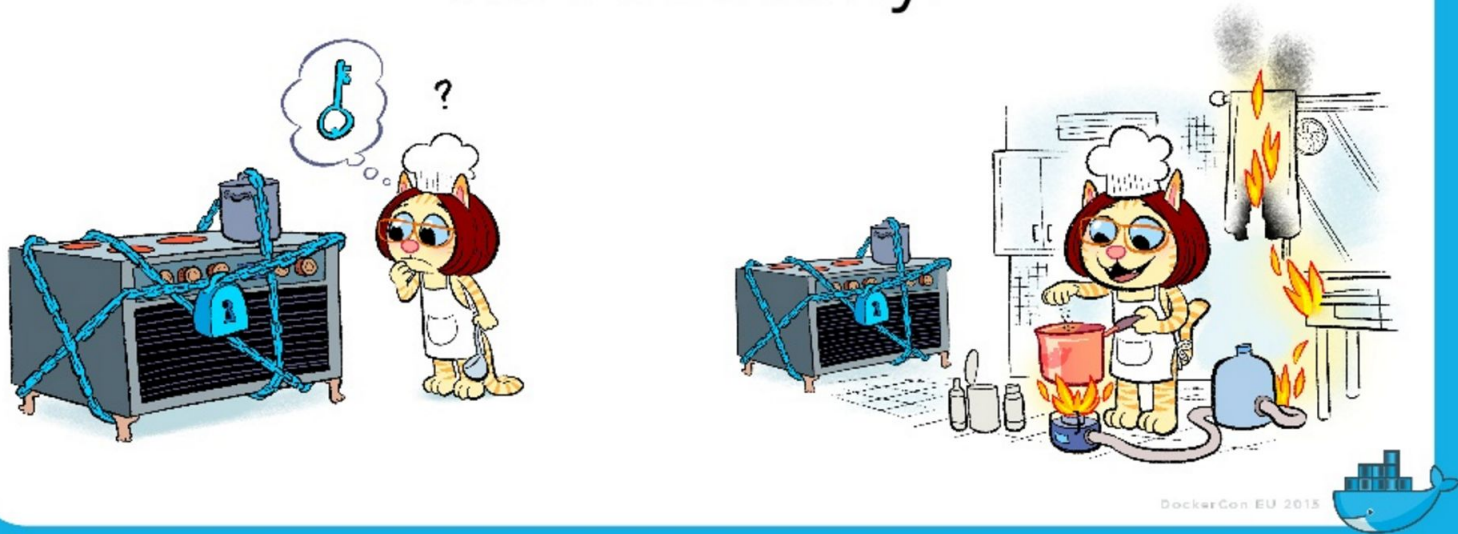
DOCKER INC.

- Docker has been surprised by this techno "flare"
- Very, very lively Open Source community
- "Batteries included"
- Standardization (RunC, etc.)

WELL GROUNDED APPROACH

- Coming from the web hosting world

Unusable security is
not security.



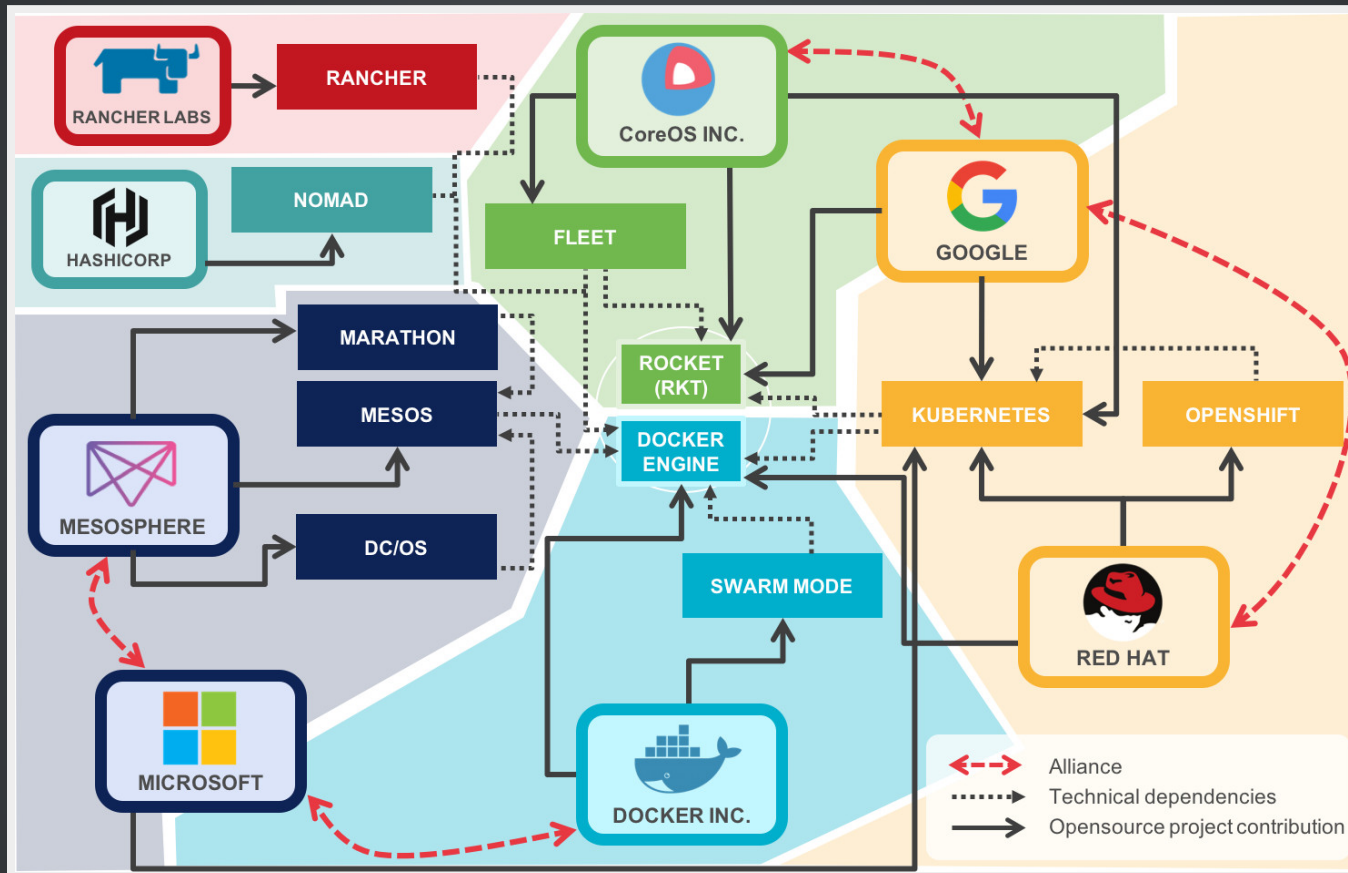
STATUS

- Was good for development and integration
- Start to be usable for Real Life Run
 - Since December 2015

STATUS

- Start to offer enterprise level solutions
 - "Docker Datacenter"
 - Trusted Registry (Image scanning, sig/auth)
 - Docker Universal Control Plane
 - Docker Cloud

CONTAINERUS BELLUM



<http://blog.octo.com/containerus-bellum-ou-la-chronique-des-hostilites-dans-lecosysteme-docker/>

Docker in Production

DOCKER IN PRODUCTION

- you need to have a very good understanding of what you do
- still in the early phase
- Docker works very well for **state less** application
- State full (with databases, etc) still in infancy. Recent announcement very promising



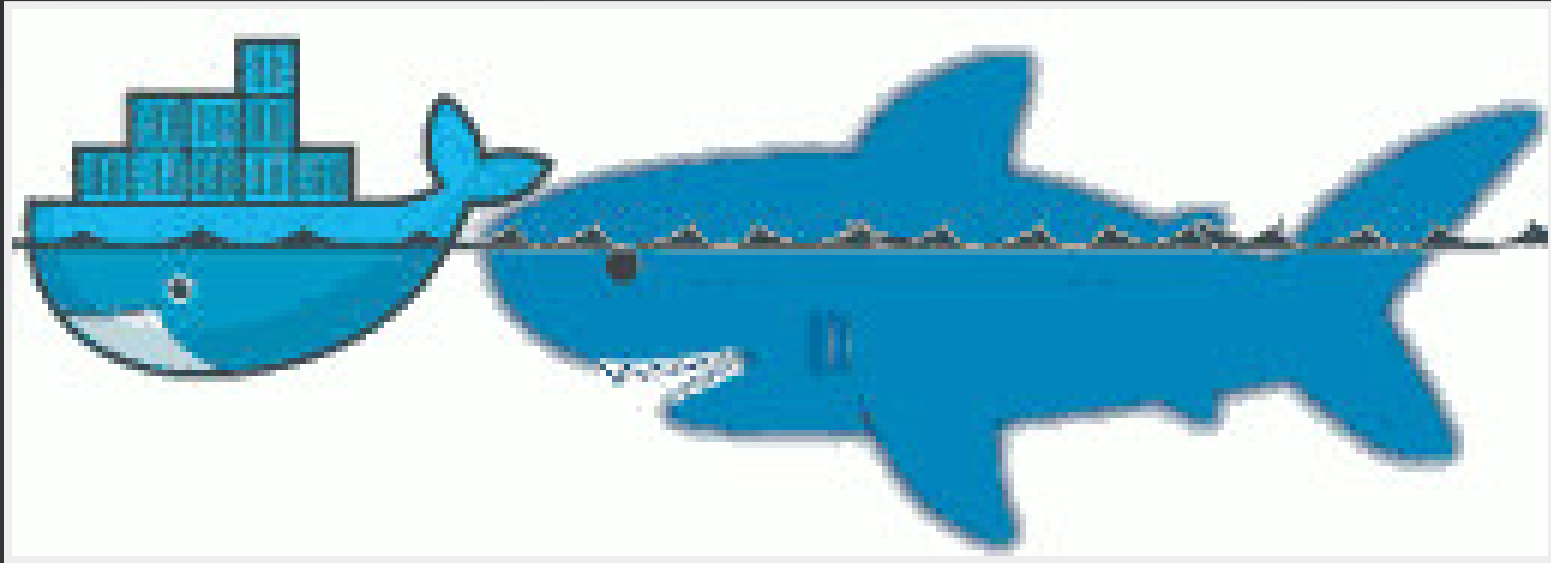
**Is Docker secure
enough for Production ?**



This is, in general, the reaction...



THE SITUATION WITH DOCKER



WHAT IS HE LOOKING FOR?



WHAT IS HE LOOKING FOR?

- (user) Data
- Access other systems
- Privilege elevation



WHAT ARE THE DANGERS WITH DOCKER?

- Kernel exploits
- Denial of service attack
- Container breakout
- Poisoned images
- Compromising Secrets

IS DOCKER "SECURE" ?

- A lot of expectations, of illusions
- "Silver bullet"
- Competition positioning (VM, Configuration Mgt)
- Enviousness

"CONTAINER DO NOT CONTAIN !"

- Wrong perception by the "public"
- Tremendous progress in 3 years
 - but usable...

EQUIPPED WITH SECURITY TOOLS

IN PARTICULAR

- Cap drop
- User namespace
- selinux / apparmor

CAPABILITY DROP

- options to the "Docker run"
- goes beyond the root/non-root dichotomy
- example: container with NTP

```
docker run --cap-drop ALL --cap-add SYS_TIME ntpd
```


SELINUX / APPARMOR

- profiles are called at each "Docker run"
- Allow to go much further in the granularity
 - this program (ex ping) has no access to the network

```
#include <tunables/global>

profile docker-default flags=(attach_disconnected,mediate_deleted) {

    #include <abstractions/base>

    network,
    capability,
    file,
    umount,

    deny @{PROC}/{*,**^[0-9*],sys/kernel/shm*} wx,
    deny @{PROC}/sysrq-trigger rwklx,

    deny mount,

    deny /sys/[^f]*/** wklx,
    deny /sys/f[^sl]*/** wklx
```

A yellow container crane is lifting a red shipping container from the deck of a ship. The container is suspended in the air, and the crane's arm is visible above it. In the background, there are stacks of other shipping containers in various colors like blue, green, and red. The ship's hull is red, and a large ship wheel is visible in the lower left. The sky is blue with some clouds.

"CLEAN" CONTAINERS?

- Malicious contents
- Contains vulnerabilities or bugged applications

TRUSTED REGISTRY

- Systematic use of TLS
- Re-enforcement of the layers integrity
- Upgraded with version 1.10

NOTARY

- System of image signature and its validation
 - Validation of the author and content non alteration
- Protection Against Image Forgery
- Protection Against Replay Attacks
- Protection Against Key Compromise
 - Clever usage of physical key storage

NAUTILUS

- (now called "Docker Security Scanning")
- Docker image scanner
 - vulnerabilities (CVE check)
 - Licence validation
 - Image Optimisation
 - Simplified functional tests

RECOMMENDATIONS



RECOMMENDATIONS

- Keep your host/images up-to-date
- "Bulkheading"
 - Seperate disk partition for Docker
 - Don't run other (non-Docker) applications on the same host
 - Container in a VM ?
- Limit inter-container communications
- log/audit trails
- Access control

RECOMMENDATIONS

- Do not use "privileged" if it is not necessary
- Applicative users in the containers
- Where are my images coming from ? are they up-to-date ?
- Access rights on the files

CONCLUSIONS

- "Is Docker 'secure' ?"
 - No more or less then the door of an apartment
- Security is everyone's business : DevOps + SecOps

THANK YOU !

CONTACT INFO



- jean-marc@meessen-web.org
- Twitter: @jm_meessen