WHAT IS THIS "DOCKER" ?

Jean-Marc Meessen

I had a dream !

- My own copy of the database
- My own test environment

...And it became true !

HELLO !

- Jean-Marc MEESSEN
- Brussels, Belgium
- "Brol" engineer
  - Senior ESB Java Developer
  - Development Infrastructure Expert
  - Mentor


iglo
CAPTAIN IGLO
iglo
10
Fish Sticks
NATUURLIJKE
BRON VAN
SOURCE NATURELLE DE
SELENIUM
15min
8min
iglo
Forever Food
300g
0:32

AND YOU ?

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

YOU AND DOCKER ?

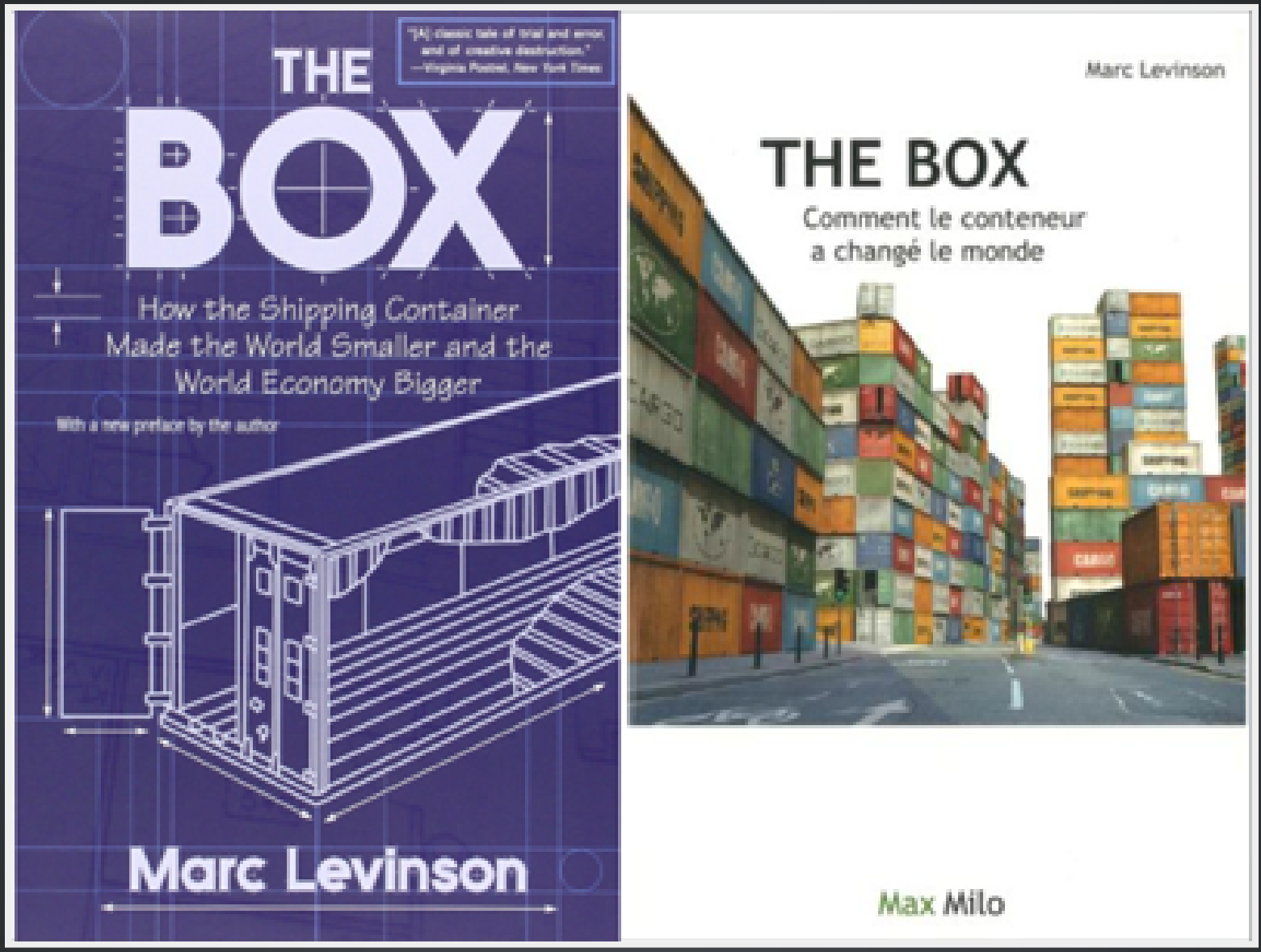
- Never heard about it ?
- Some "Proof of Concept" ?
- Use it every day ?
- In Production ?

TODAY'S TALK

- What are "containers" ?
- How to start ?
- Where is Docker heading ?


What are "containers"?

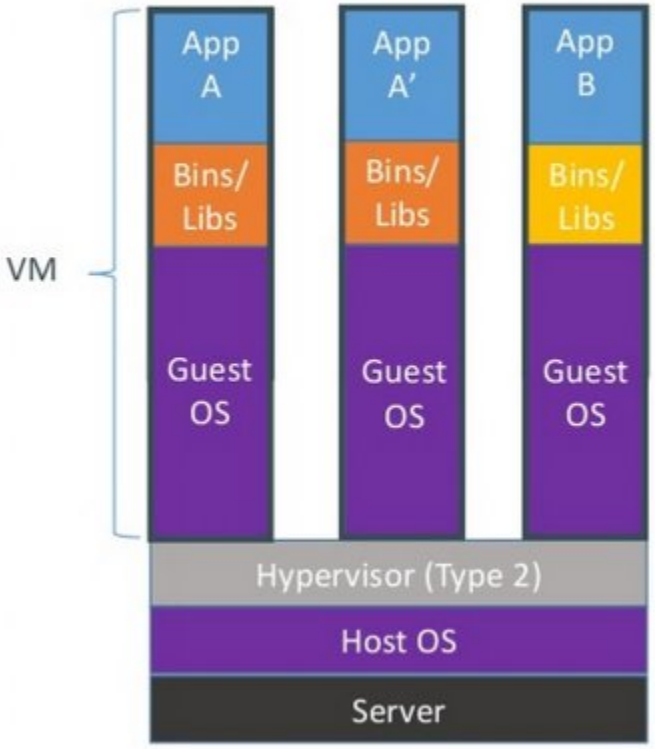

SOU
No. 4 W


THE
BOX
How the Shipping Container
Made the World Smaller and the
World Economy Bigger
With a new preface by the author
Marc Levinson
Marc Levinson
THE BOX
Comment le conteneur
a changé le monde
Max Milo

DOCKER CONTAINERS ARE :

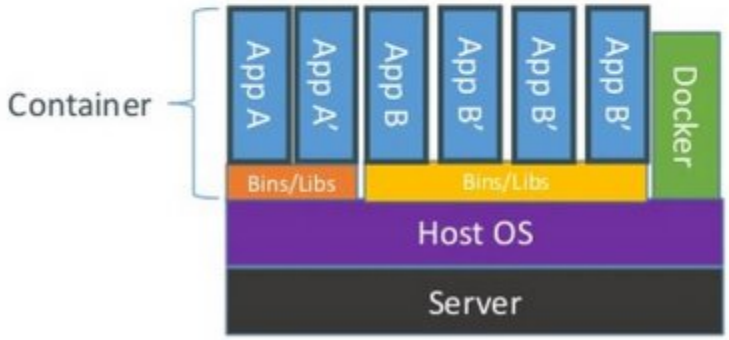
- Isolated namespace (process, users, network, cgroups, etc)
- Lightweight "virtualized servers"
- A new artifact paradigm

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



APPLICATIONS PACKAGED WITH SYSTEM DEPENDENCIES

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

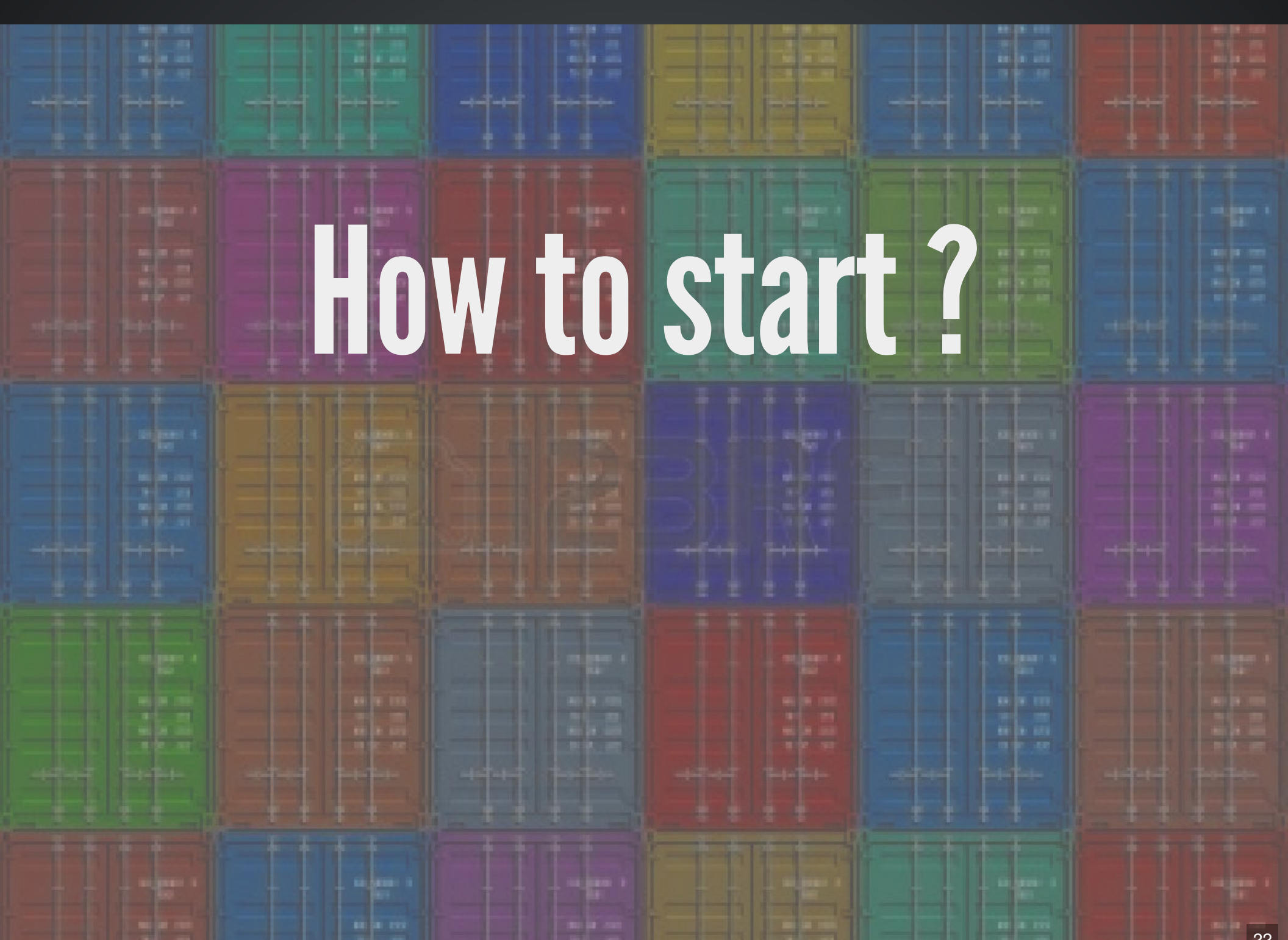
EXECUTION MODE

- Container can run and then exit
- Container can in background (daemon)

DEMO

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"


How to start ?

NEED A CONTAINER ENABLED "KERNEL"

LINUX OR WINDOWS 10 (BETA)

INSTALL THE DOCKER "DAEMON"

DOCKER CLIENT

- Run, stop, halt container
- Remove stopped container
- Examine logs
- etc.

AND FOR WINDOWS OR MAC OS X ?

- Install a virtual machine (ex VirtualBox)
- Ready made bundles:
  - Docker Toolbox
  - New, better integrated, clients
- Using (corporate) proxies: advanced topic

THEN YOU NEED 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" >> /
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"]
```

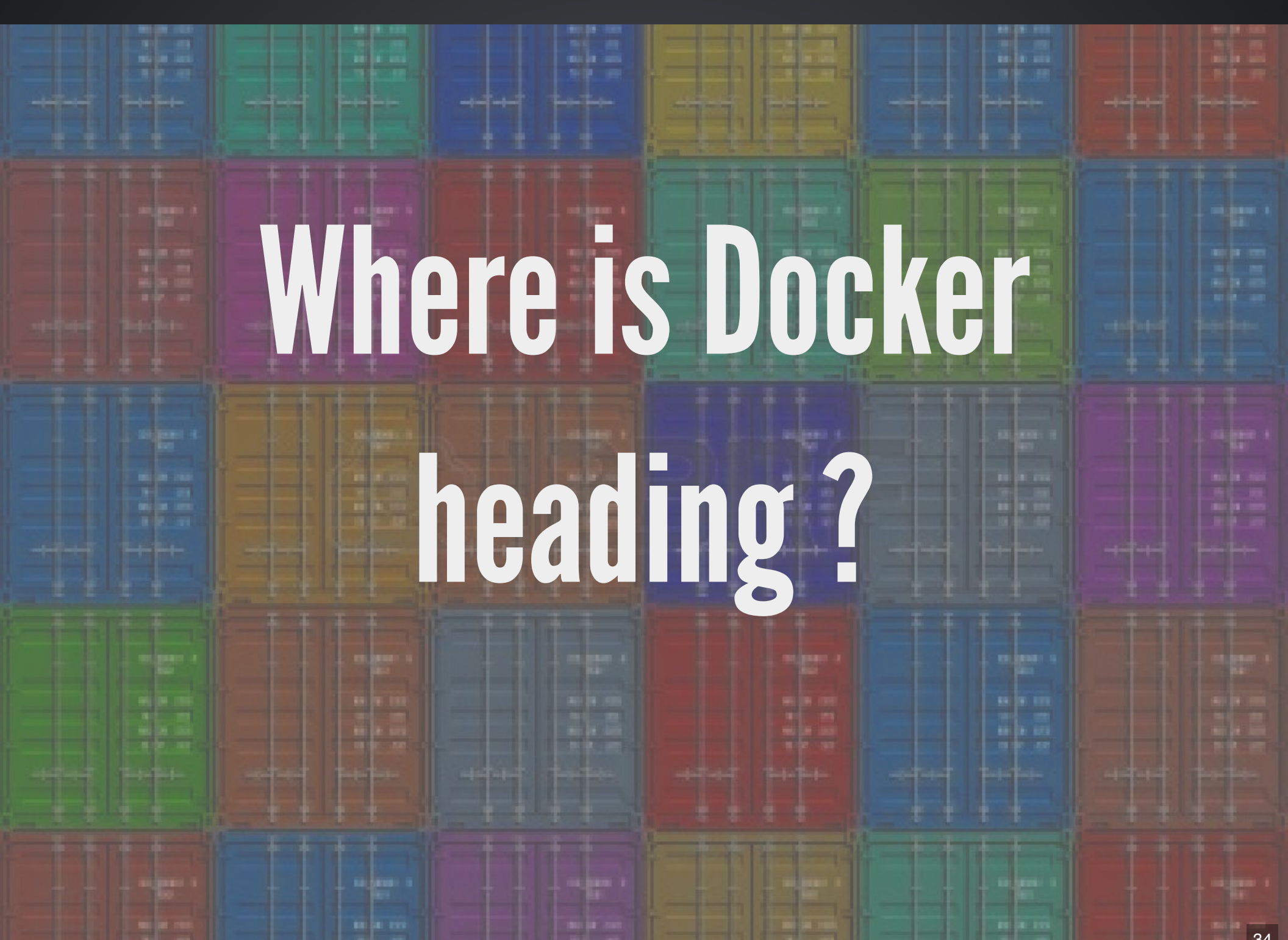
DESCRIBE A COMPLETE INFRASTRUCTURE

- 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.

"BUILD, SHIP AND RUN"


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 hosting world

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

DOCKER IN PRODUCTION ?

This is, in general, the reaction…

THE PROBLEM

- Docker's popularity reflects the quest for less and less friction.
- Its ease of use leads to compromises and to neglect verification.

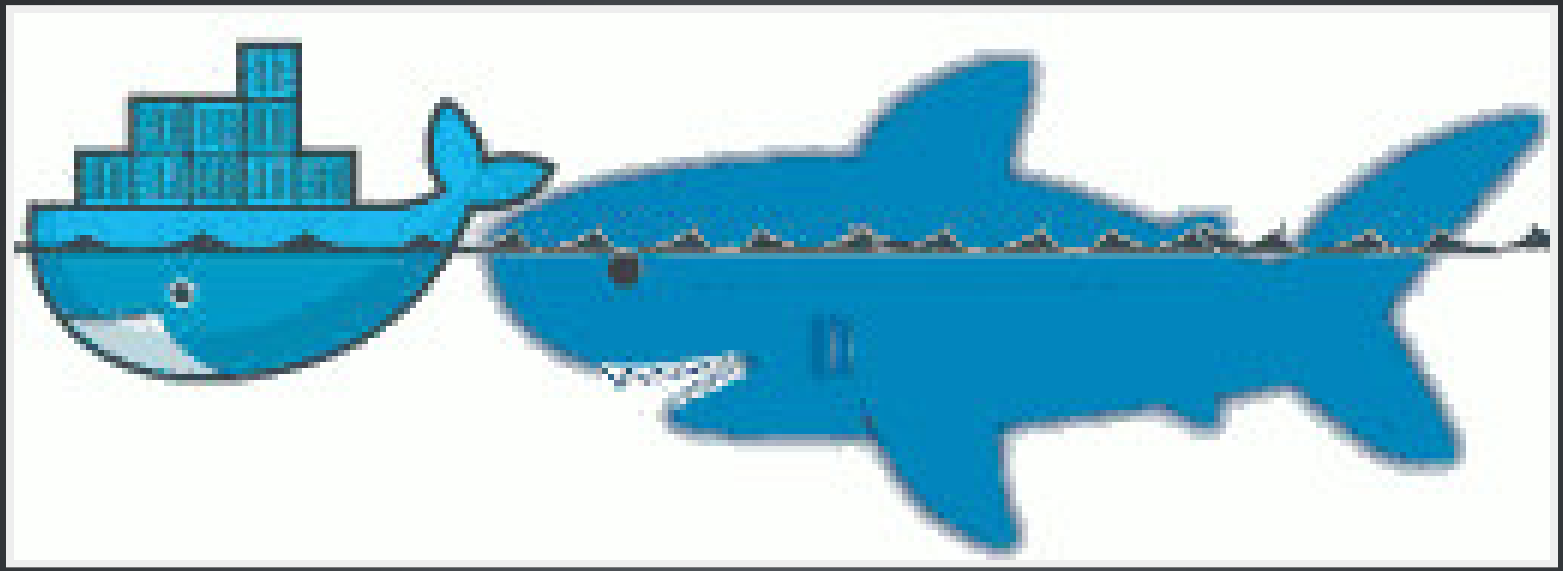
And yet **Security** is important.

AND WHY ?

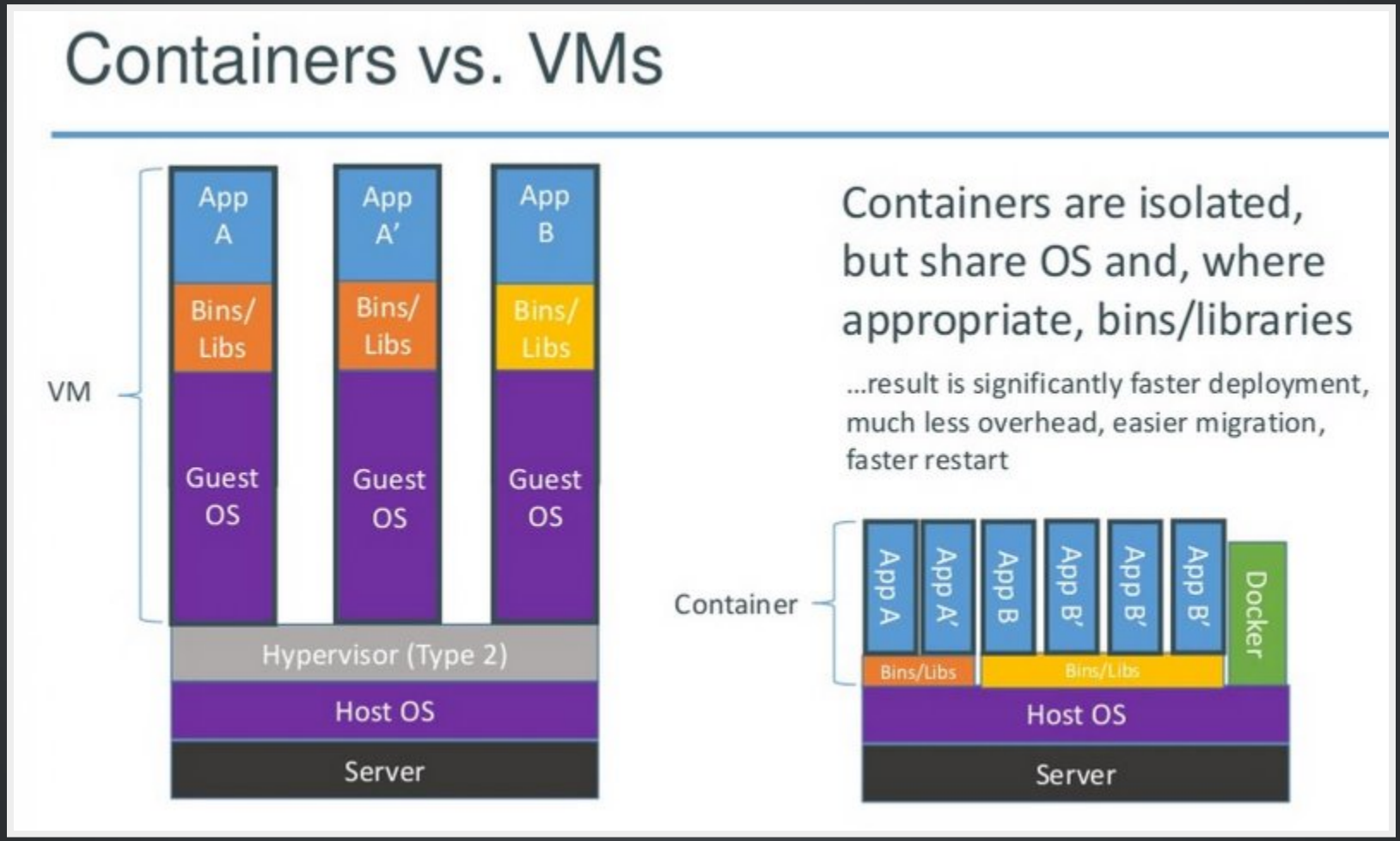
- Our customers entrust us their systems / their data.
- No sanctions for failing Companies
  - security is only seen as a cost
  - no "polluter pays" principle

I believe that we have a moral responsibility to remind our managers of the good (security) practices.

THE SITUATION WITH DOCKER



REMINDER


Containers vs. VMs
VM
App A
App A'
App B
Bins/ Libs
Bins/ Libs
Bins/ Libs
Guest OS
Guest OS
Guest OS
Hypervisor (Type 2)
Host OS
Server
Containers are isolated, but share OS and, where appropriate, bins/libraries
...result is significantly faster deployment, much less overhead, easier migration, faster restart
Container
App A
App A'
App B
App B'
App B'
App B'
Docker
Bins/Libs
Bins/Libs
Host OS
Server

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

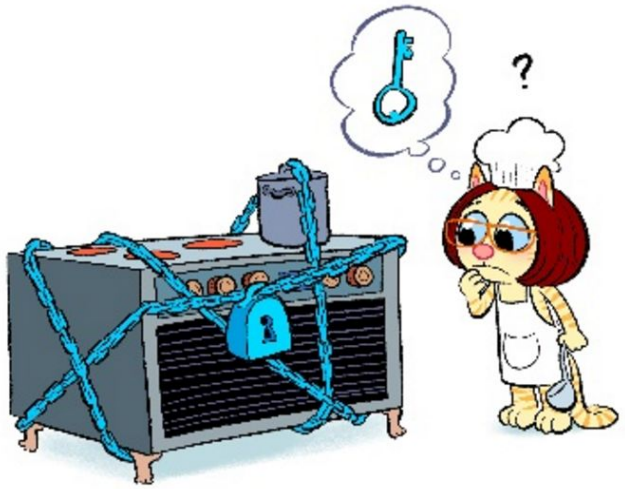
DOCKER, INC AND SECURITY

- Security (= operability) is one of their fundamental preoccupation
- Aware of the youth of the technology
- Very reactive
- Positive attitude in the approach

"How to make developers care about security?"

Wrong question.

Unusable security is not security.



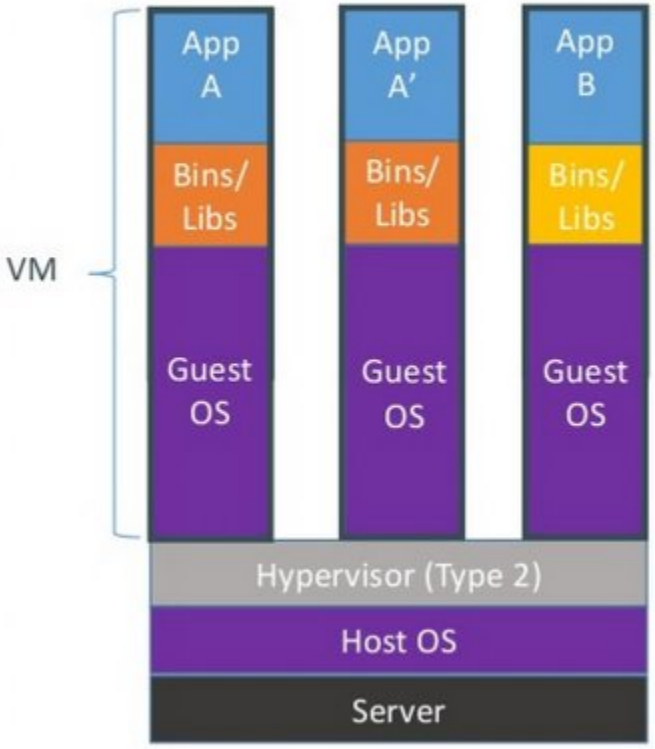
DockerCon EU 2015

"How to give developers usable security?"

"CONTAINER DO NOT CONTAIN !"

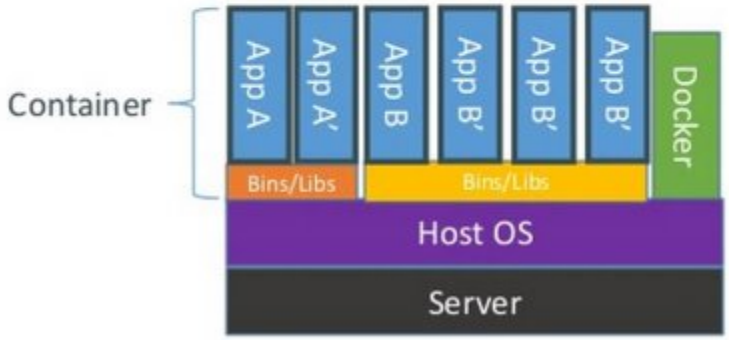
- Wrong perception by the "public"
- Tremendous progress in 3 years
  - but usable…

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



Isolation of Linux containers: it's complicated

- pid namespace
- mnt namespace
- net namespace
- uts namespace
- ipc namespace
- user namespace (new)
- pivot_root
- uid/gid drop
- cap drop
- all cgroups
- selinux
- apparmor
- seccomp

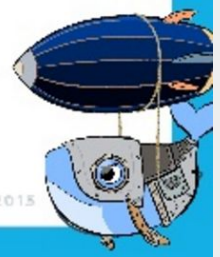
Isolation supported by Docker Engine 0.1 in March 2013

- ✓ pid namespace
- ✓ mnt namespace
- ✓ net namespace
- ✓ uts namespace
- ✓ ipc namespace
- ✗ user namespace (new)

- ✓ pivot_root
- ✓ uid/gid drop
- ✗ cap drop
- ✗ all cgroups
- ✗ selinux
- ✗ apparmor
- ✗ seccomp

Isolation supported in Swarm/Engine experimental

- ☑ pid namespace
- ☑ mnt namespace
- ☑ net namespace
- ☑ uts namespace
- ☑ ipc namespace
- ☑ **user namespace (new)**
- ☑ pivot_root
- ☑ uid/gid drop
- ☑ cap drop
- ☑ all cgroups
- ☑ selinux
- ☑ apparmor
- ☑ **seccomp**



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
```

USER NAMESPACE

WITHOUT USER NAMESPACE


End
Demo

WITH USER NAMESPACE


End
Demo

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*} wkx,
  deny @{PROC}/sysrq-trigger rwklx,

  deny mount,

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

"CLEAN" CONTAINERS?

- Malicious contents
- Contains vulnerabilities or bugged applications

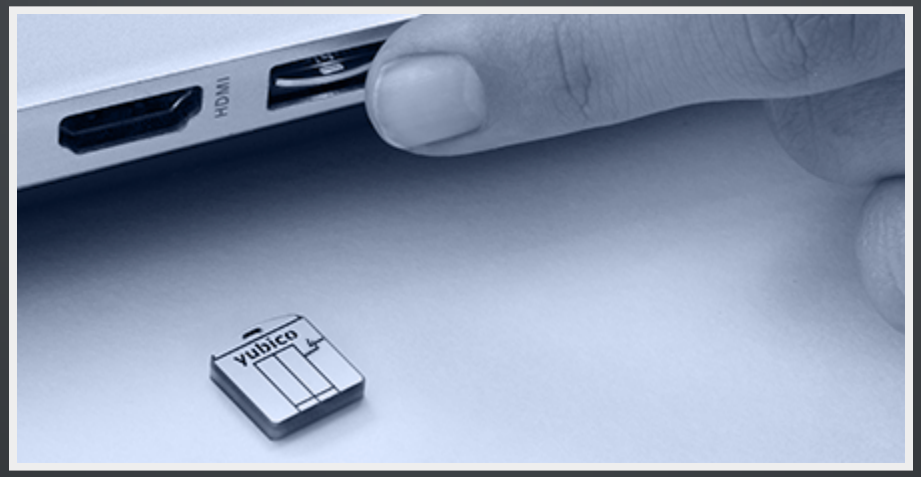
TRUSTED REGISTRY

- 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

YUBIKEY 4



NAUTILUS

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

DOCKER INC'S STRATEGY

- Secure Platform
- Secure Content
- Secure Access

SECURE PLATFORM

- Secure Platform
  - All available isolation containment
  - Default security settings and profiles
  - Docker Bench

SECURE CONTENT

- Secure Content
  - Docker Content Trust
  - Security Scanning

SECURE ACCESS

- Secure Access
  - Role Based Access control
  - AD/LDAP integration
  - Authentication plugins

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 "priviliged" 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