DOCKER IN PRODUCTION ? AND WHAT ABOUT SECURITY ... ?

Jean-Marc Meessen

HELLO !

- Jean-Marc MEESSEN
- Bruxelles, Belgique
- EWC-FPL-Be Middleware developer
- (Development Infrastructure Expert)


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 ?

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.

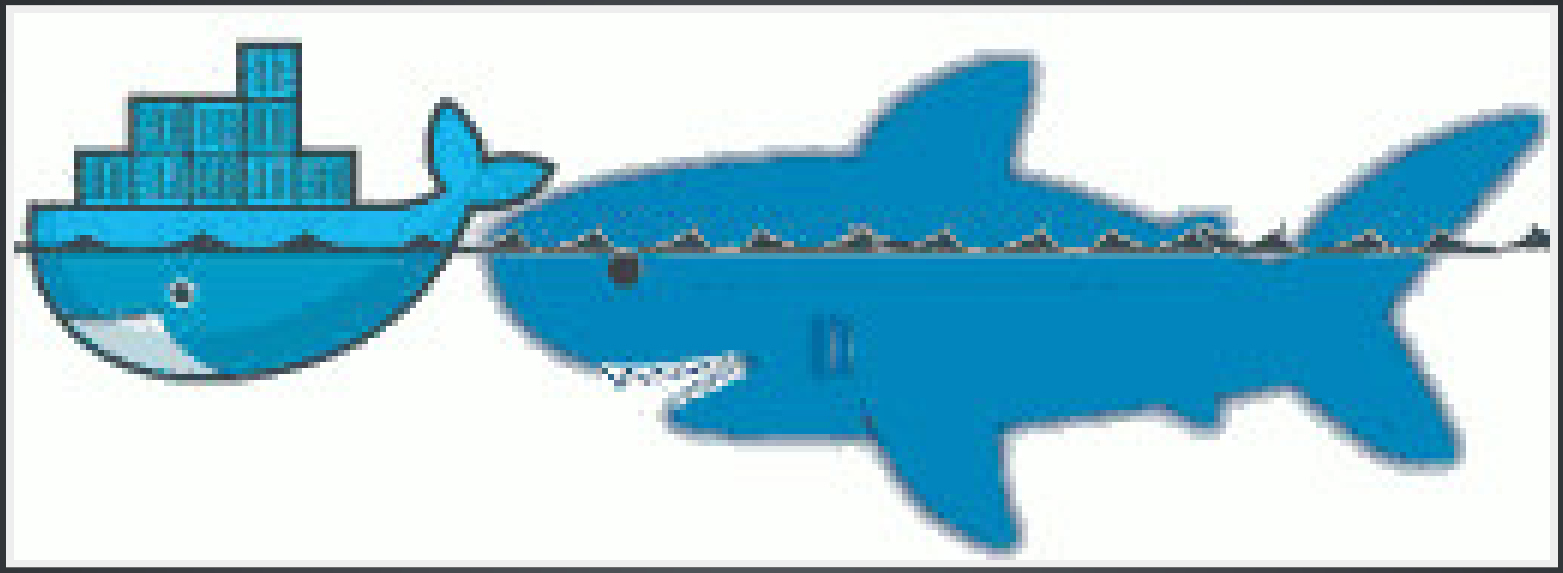
And yet **Security** is important.

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

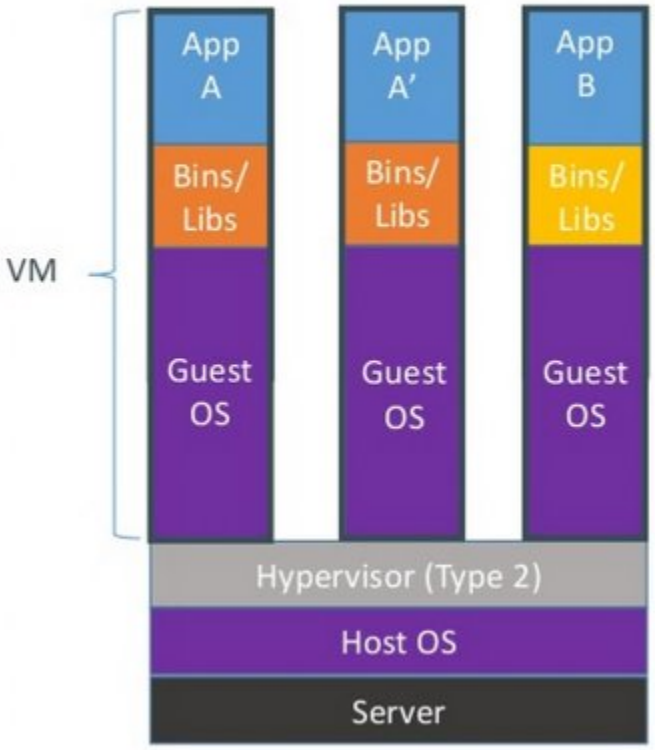
I believe that we have a moral responsibility to remind our managers of the good (security) practices.

THE SITUATION WITH DOCKER



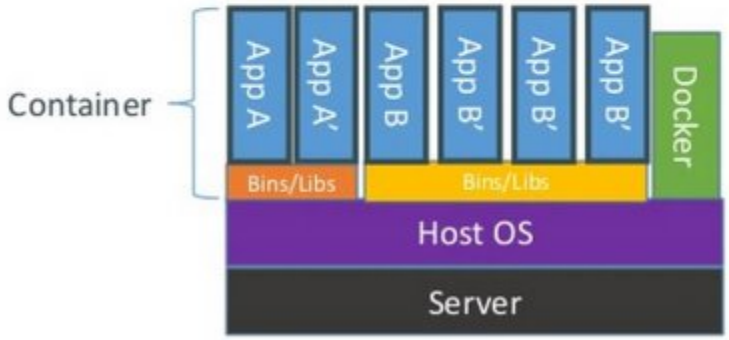
REMINDER

Containers vs. VMs



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

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



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

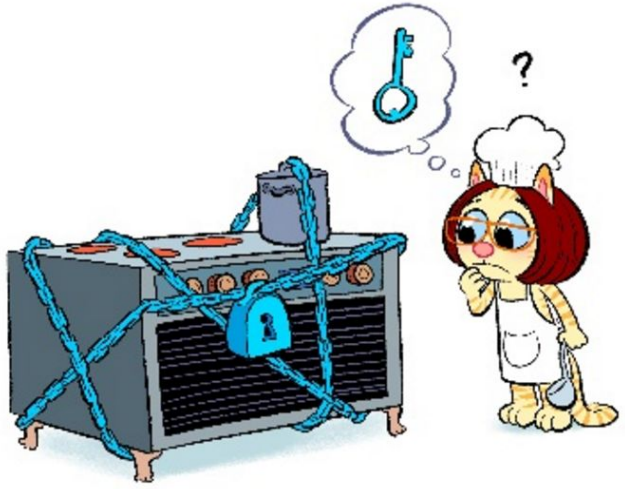
DOCKER, INC AND SECURITY

- 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.



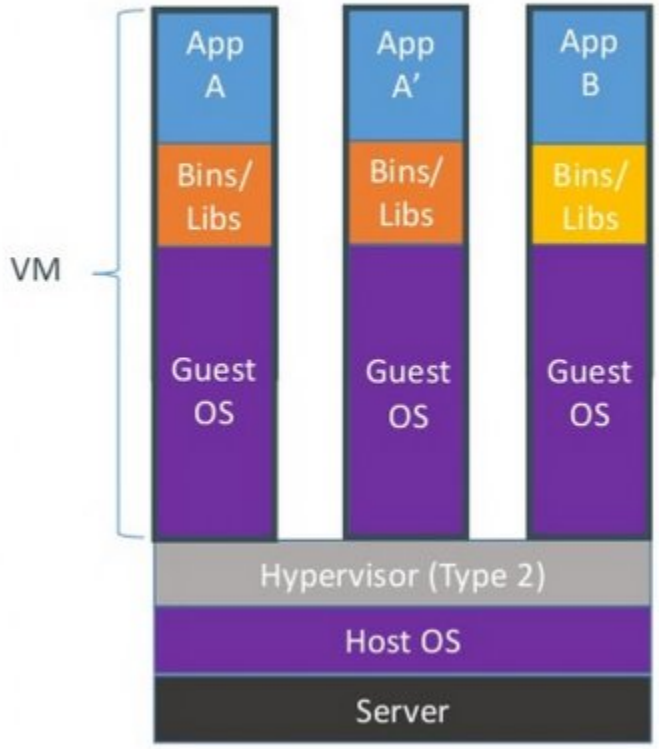
DockerCon EU 2015

"How to give developers usable security?"

"CONTAINER DO NOT CONTAIN !"

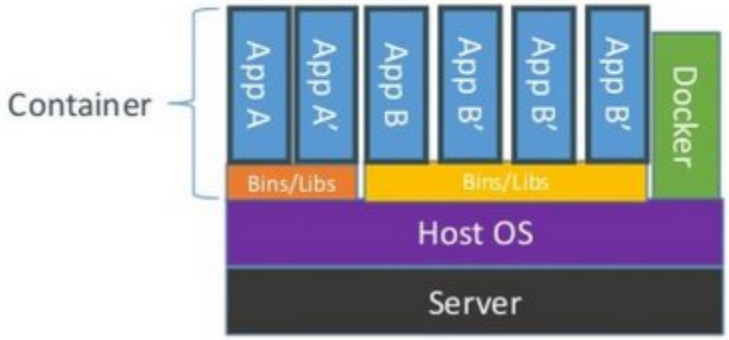
- Wrong perception by the "public"
- Tremendous progress in 3 years
  - but usable…

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

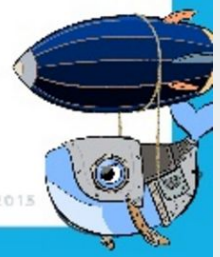
Isolation supported by Docker Engine 0.1 in March 2013

- ✓ 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

WITH USER NAMESPACE

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

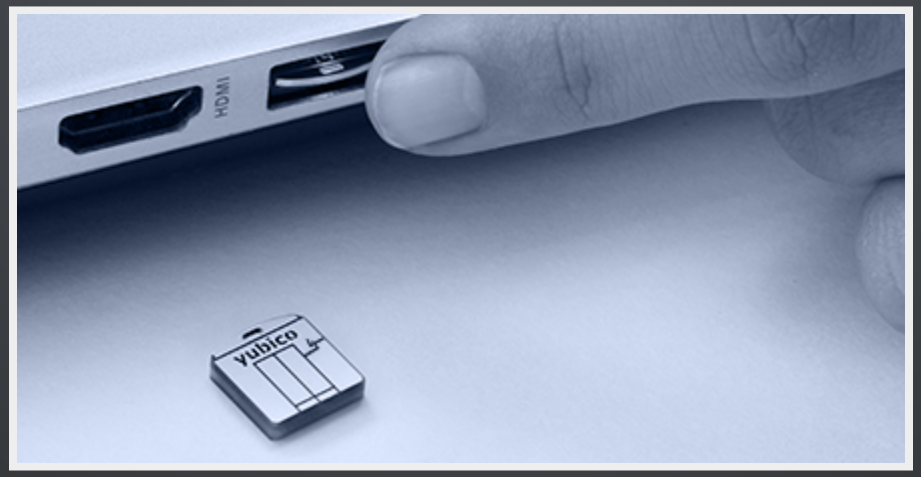
TRUSTED REGISTRY

- 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

CONTACT INFO

- jeanmarc.meessen@worldline.com
- Twitter: @jm_meessen

CRÉDITS PHOTOGRAPHIQUES

- Vidéo Capt. Igloo
- Boite de fishSticks
- Question
- Panique
- Docker shark
- Docker overview

CRÉDITS PHOTOGRAPHIQUES - 2

- Hacker
- Trésor
- Slides de la Keynote de Dockercon 2015 à Barcelone
- Démo
- prévention
- Question