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 !

HELLO !

- Jean-Marc MEESSEN
- Brussels, Belgium
- "Brol Engineer" @ Worldline
  - Senior ESB Java Developer
  - Development Infrastructure Expert
  - Mentor
- Starting in January at Cloudbees


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

0:00 / 0:32

AND YOU ?

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

YOU AND DOCKER ?

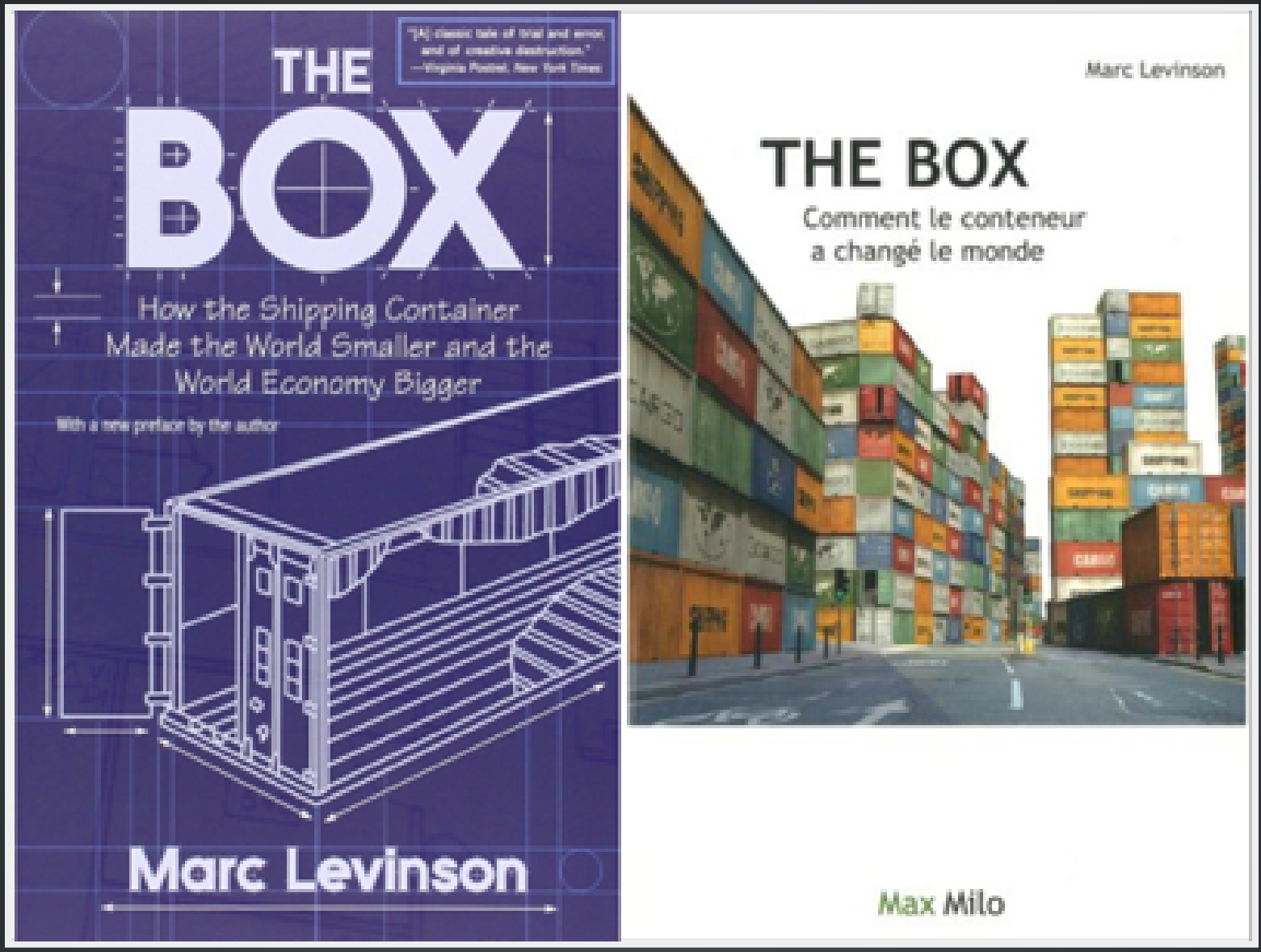
- Never heard about it ?
- Some "Proof of Concept" ?
- Use it every day ?
- In Production ?

TODAY'S TALK

- What are "containers" ?
- How to start ?
- Docker for the Java Dev


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

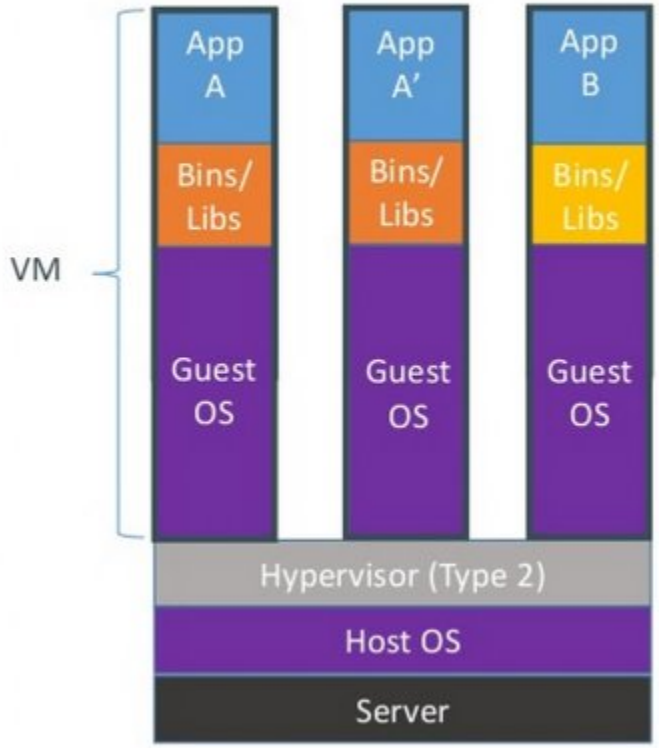
DOCKER CONTAINERS

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

DOCKER CONTAINERS ARE :

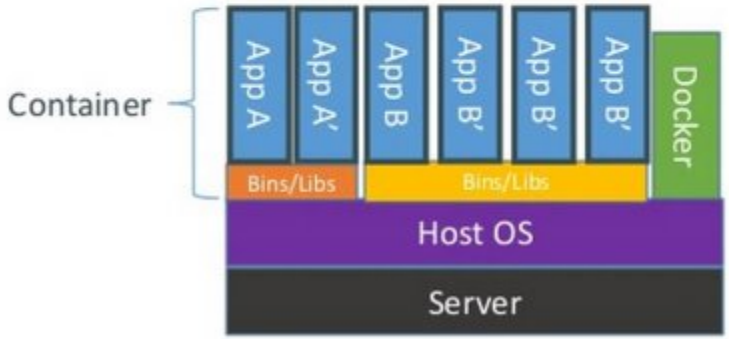
- cgroups (control groups)
  - limits CPU, memory, IOs, etc
- NameSpaces
  - isolates and virtualizes system resources
    - (process, mounts, networking)

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

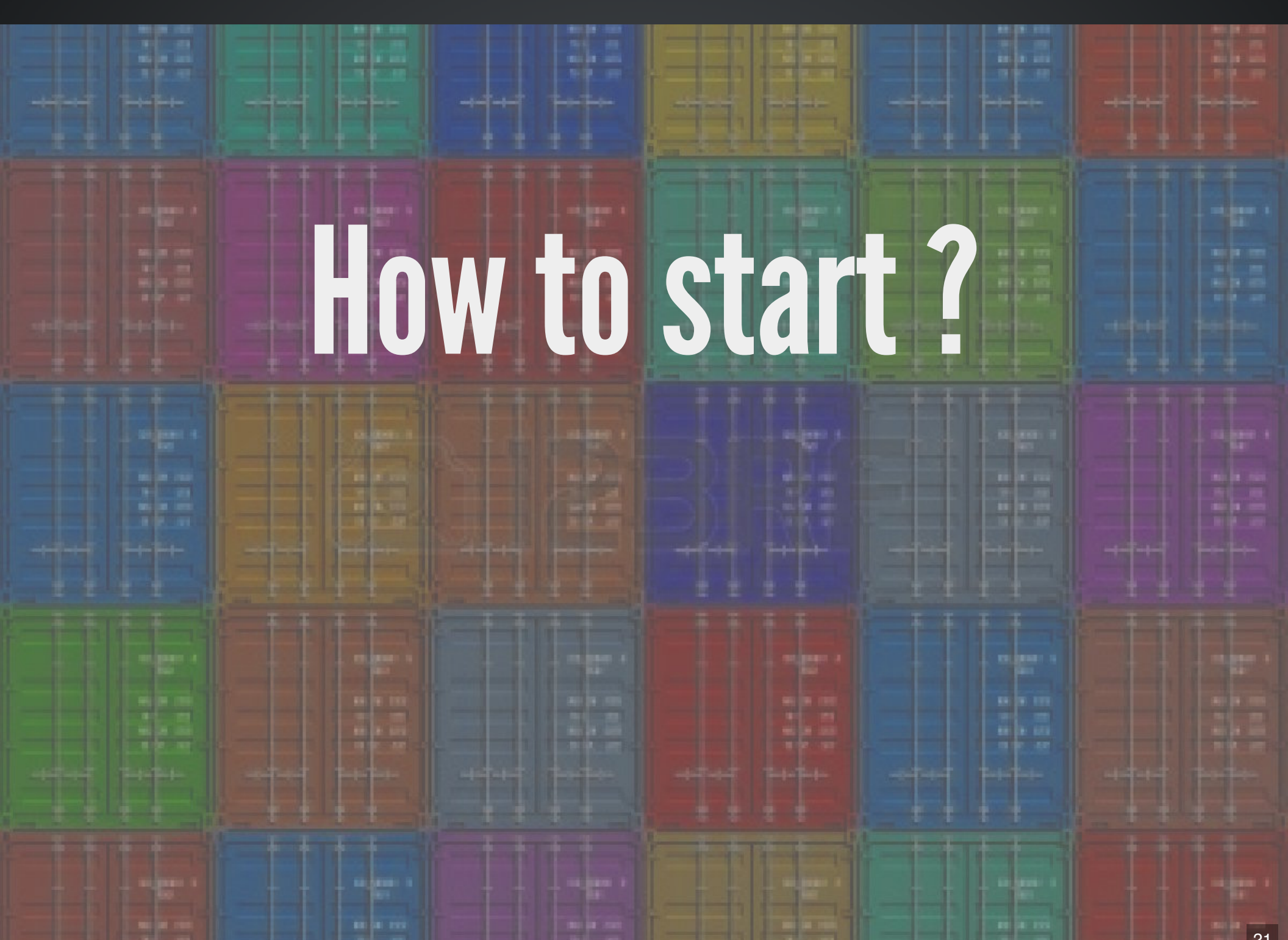


APPLICATIONS PACKAGED WITH SYSTEM DEPENDENCIES

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

WHAT DOCKER SOLVES

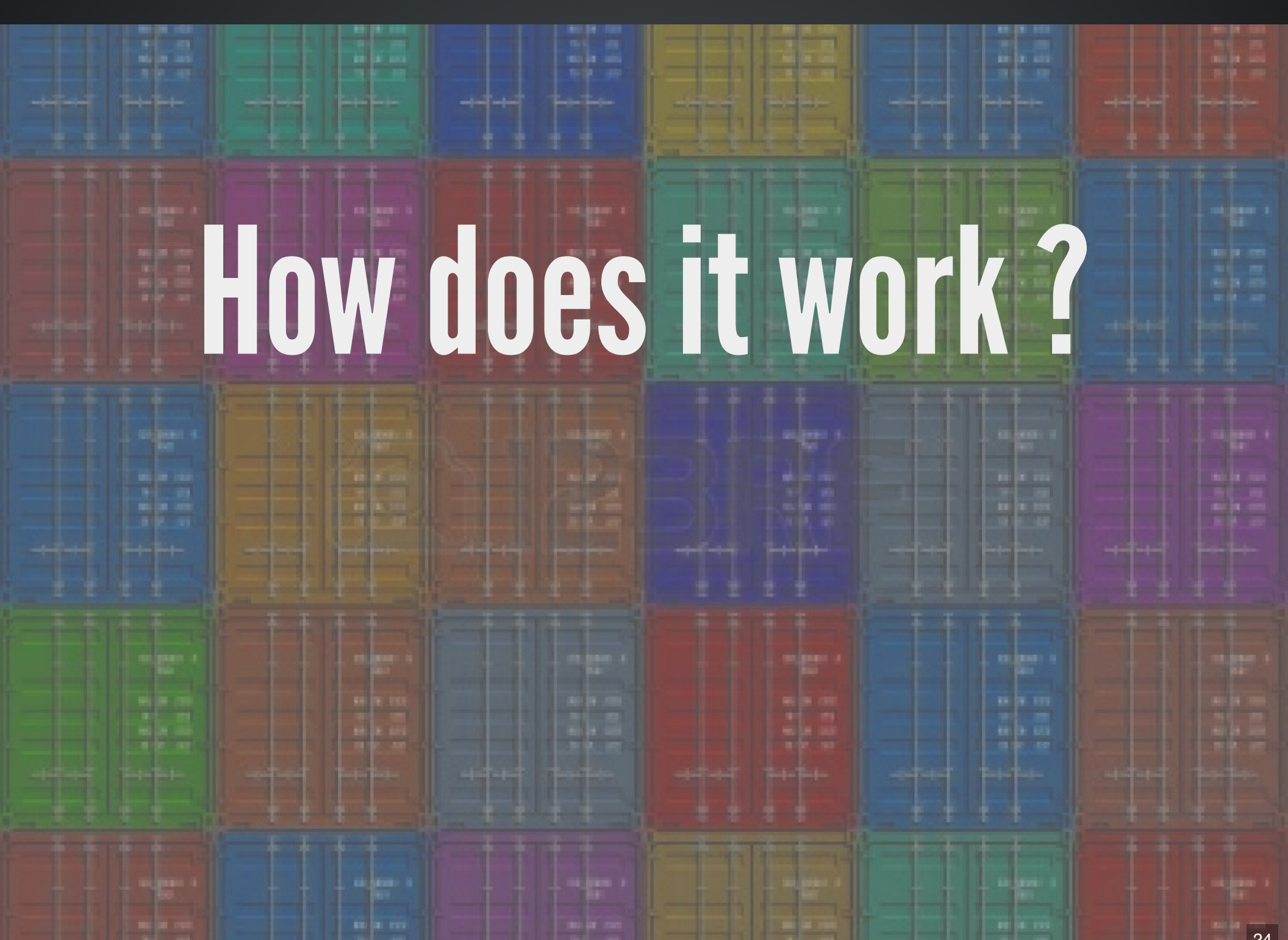
- Escape the dependencies hell
- Fast iterative Infrastructure improvement
- Container "loader" & Container "shipper"
  - (no more "it worked in Dev, now it’s OPS problem")
- easy onboarding of Devs.
- "Own test environment"


How to start ?

NEED A CONTAINER ENABLED "KERNEL"

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


How does it work ?

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" >> /
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"]
```

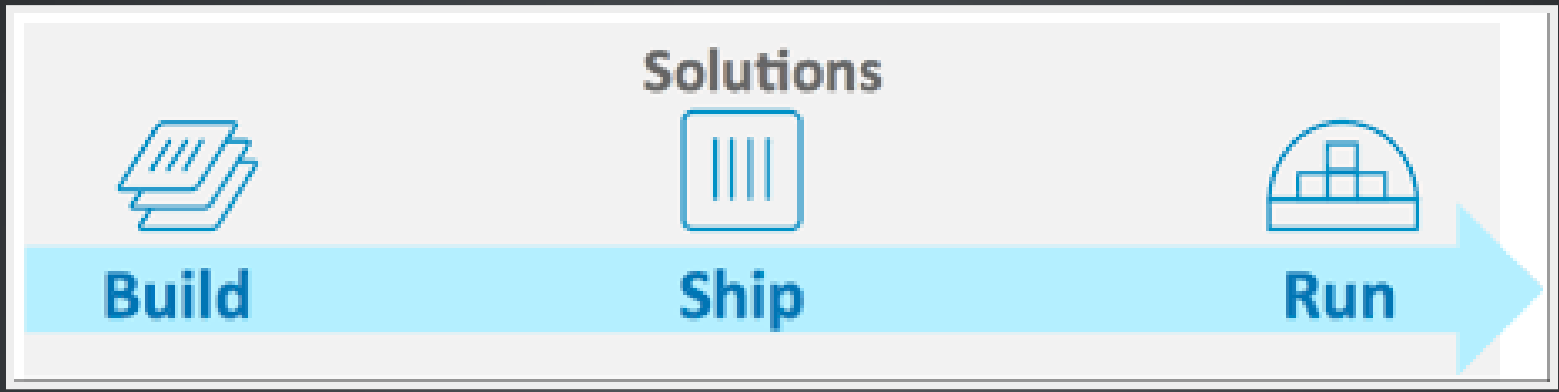
DESCRIBE A COMPLETE INFRASTRUCTURE

- 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"

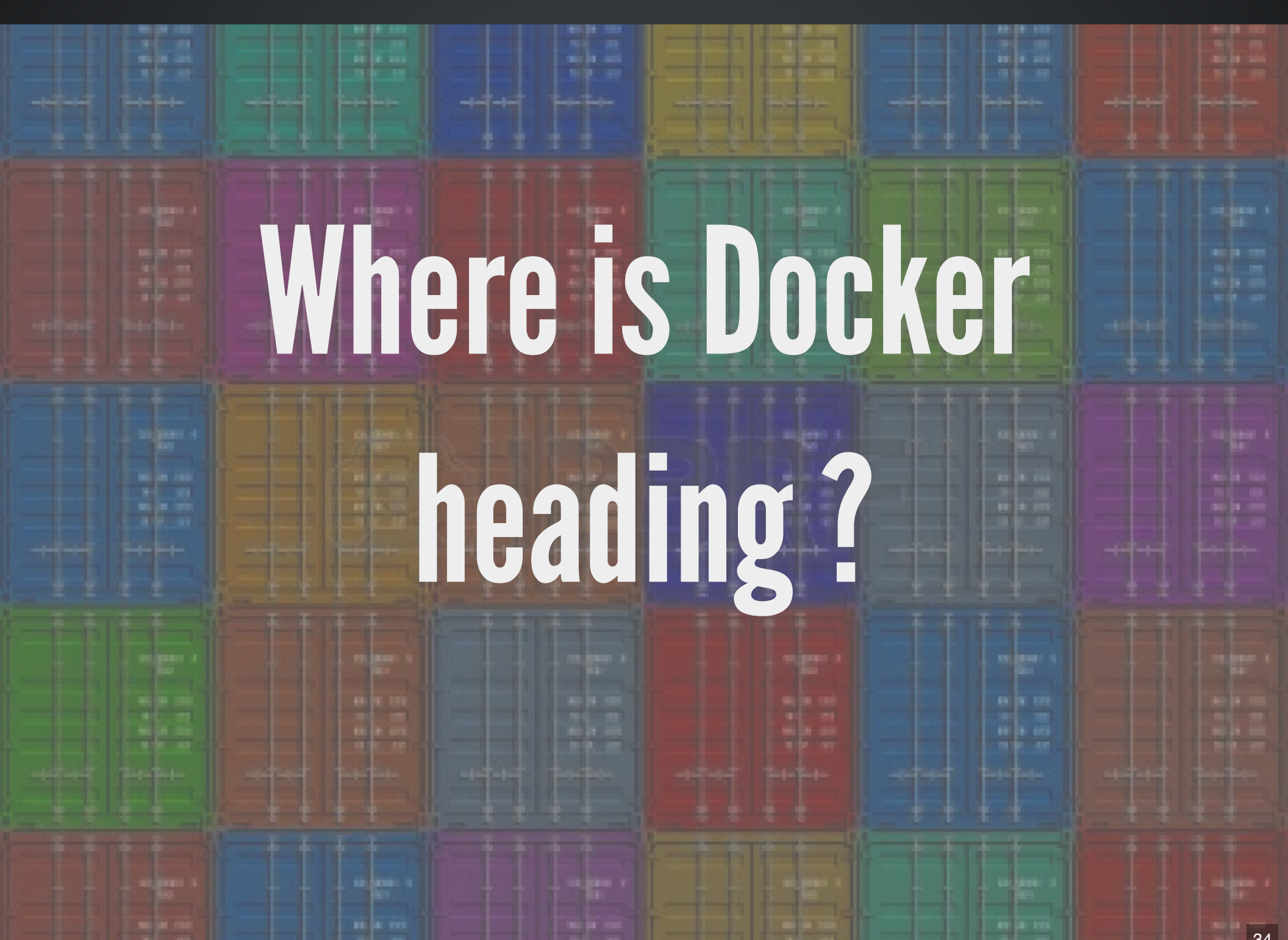


HOW TO LEARN ?

- 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/


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

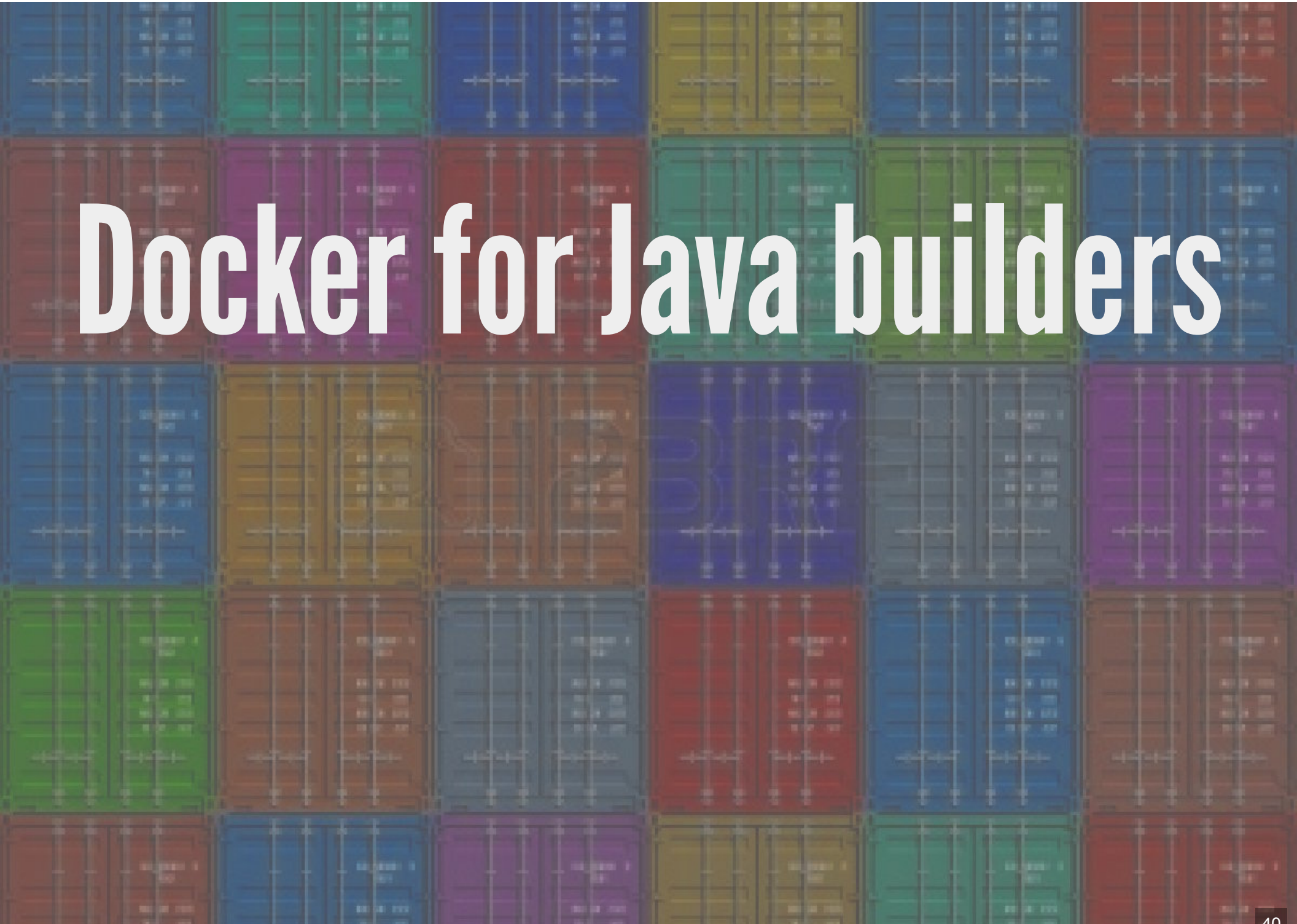
- Coming from the web hosting world

STATUS

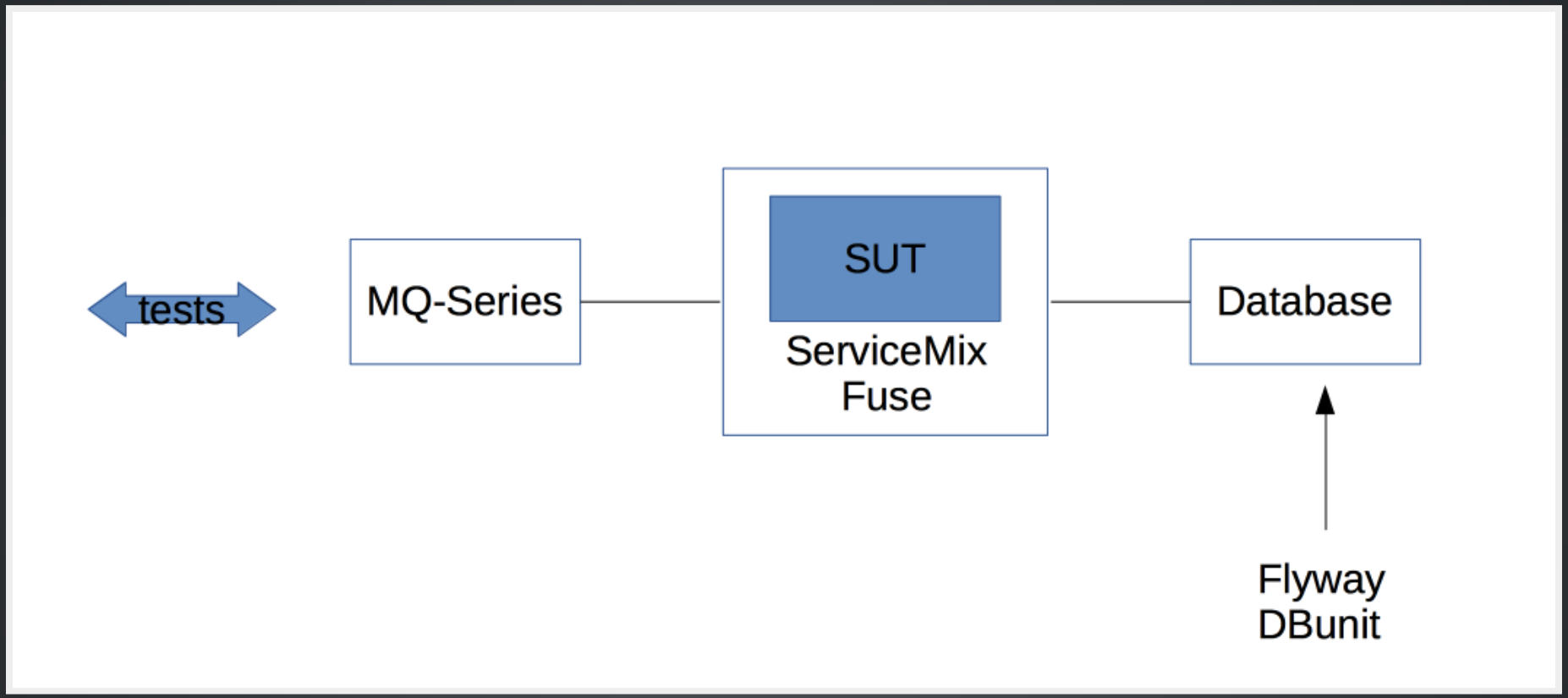
- Was good for development and integration
- Start to be usable for Real Life Run
  - Since December 2015

STATUS

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


Docker for Java builders

INTEGRATION TESTS



MAVEN INTEGRATION

- using `fabric8io/docker-maven-plugin`

MAVEN SEQUENCE

- pre-integration-test
  - execute Docker "build" and "start" goals
  - execute FlyWay migration
- integration-test
  - execute Failsafe "verify" goal
  - integration test make extensive use of DBunit
- post-integration-test
  - execute Docker "stop" goal

MAVEN INTEGRATION

- additional options
  - (networking, volumes, docker-compose)
- fits nicely in Jenkins
  - especially V2 and pipeline
  - (but this is an other story)

BASE IMAGES

- several base images are available, choose what is best suited
  - openjdk (jdk and jre)
  - openjdk:alpine
  - maven
  - Oracle JDK/JRE is not available from the shelf anymore. Need to build it yourself

DEPLOYMENT MODEL

- app-server
- embedded (self contained JAR)

INSTALLATION MODES/TEST MODES

- add the JAR in the right directory
- execute the app installation script at image build time
- execute the app installation script at container run

SINGLE PURPOSE JVM

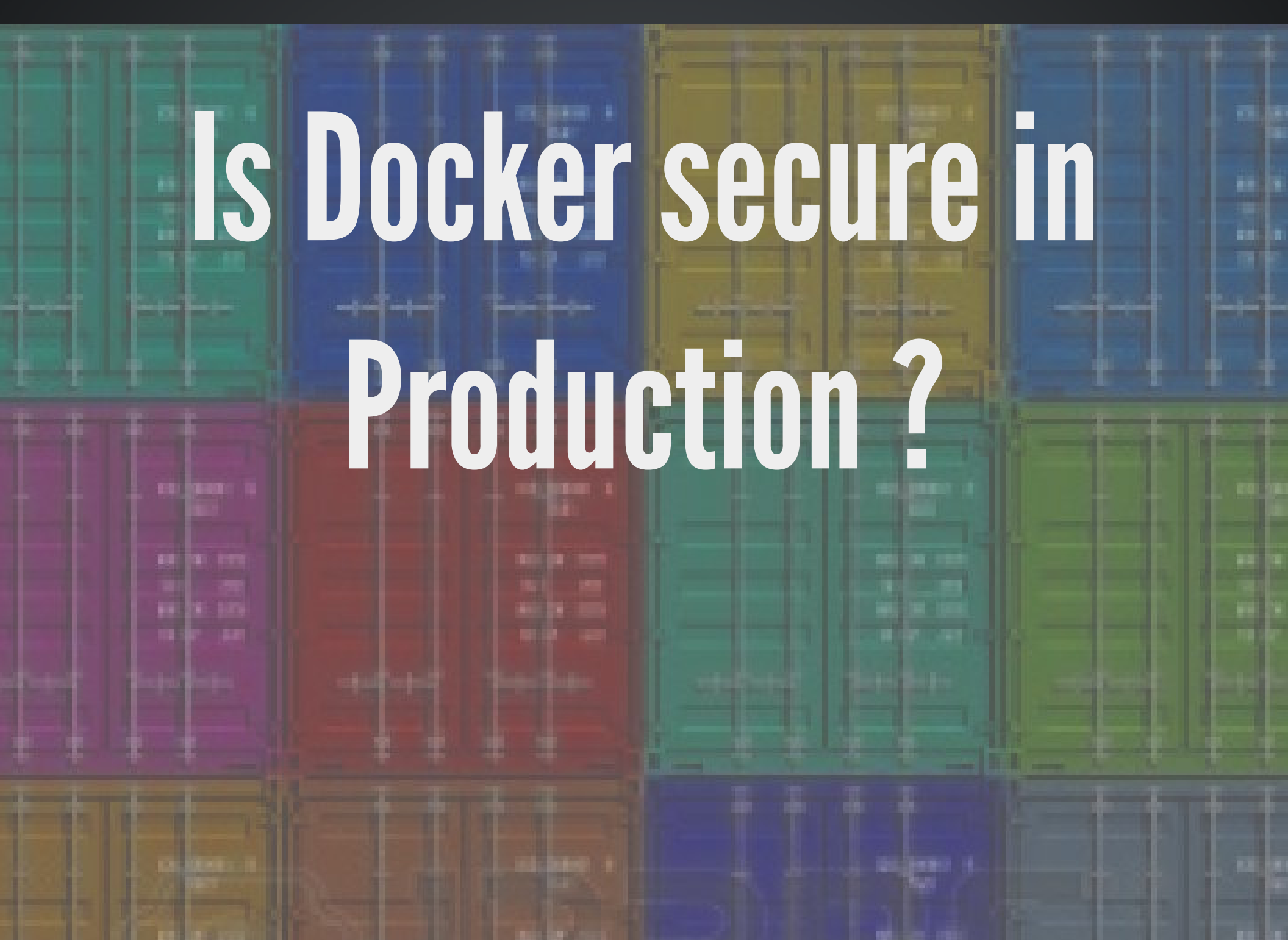
- thinks it is alone in the world
  - JVM defaults based on the machine characteristics
- watch out
  - several JVMs (in Docker) are running on the same machine

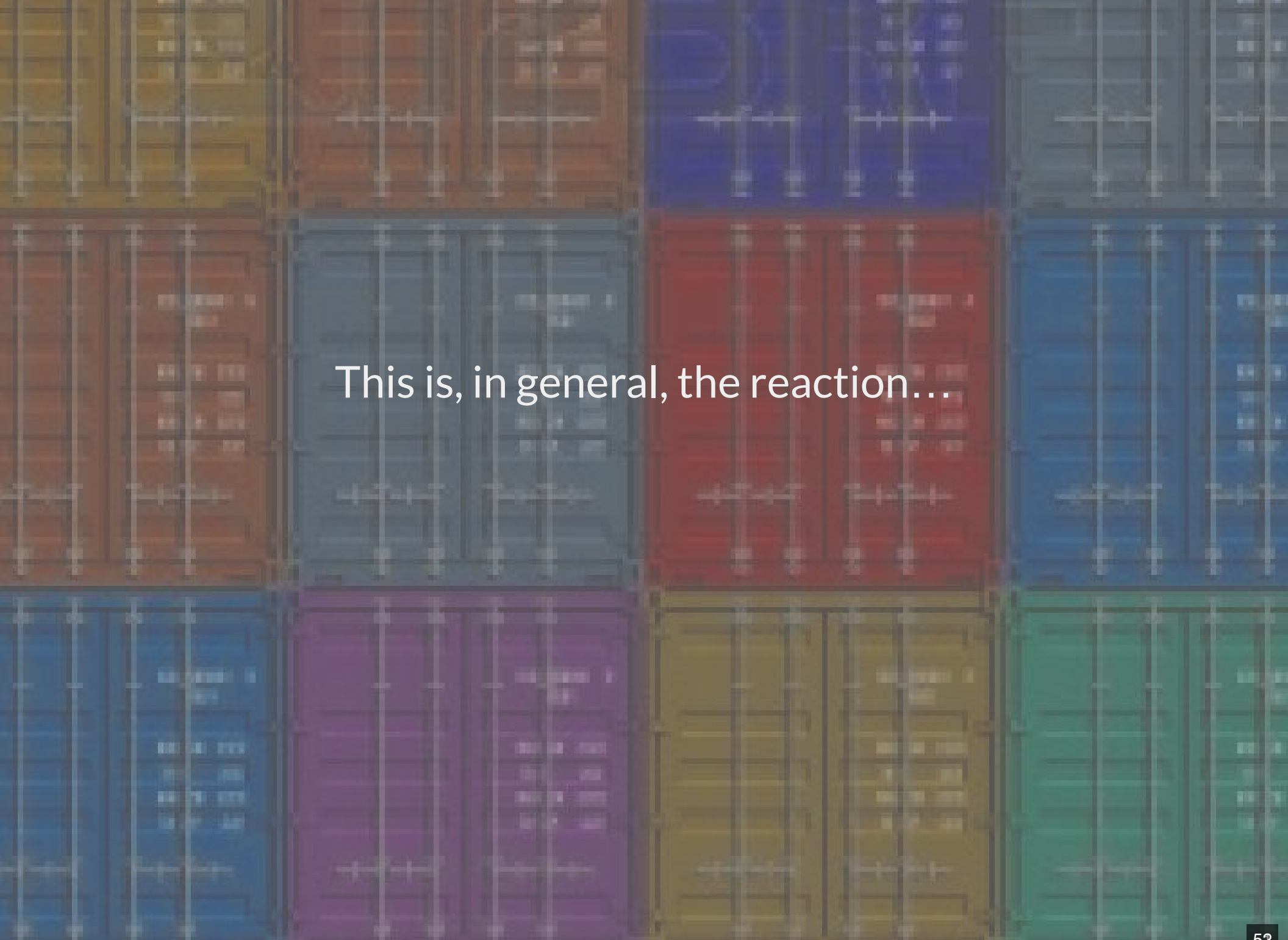
JVM CONFIGURATION

- as on a physical host you need to specifiy/limit the memory/cpu requirement of your JVM
  - Docker offer the same tools. use them
  - orchestrator like mesos or kubernetes can help you for this
- beware of Docker exec in Production

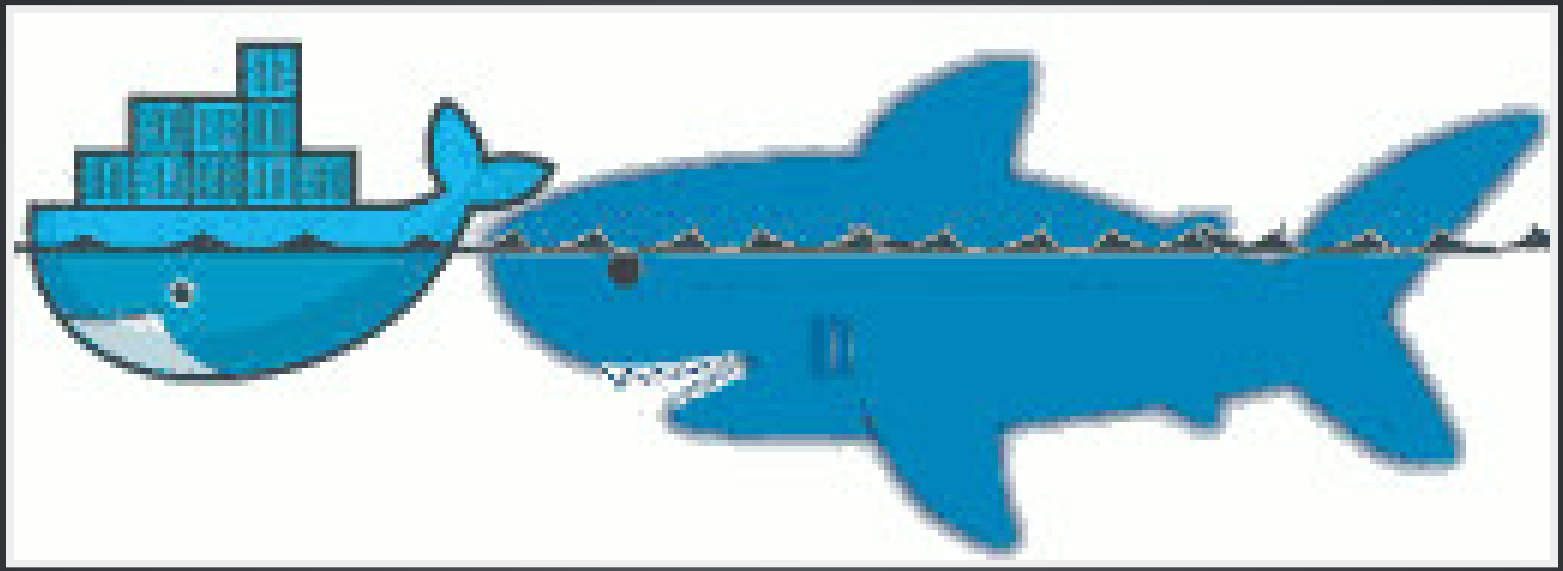
DOCKER IN PRODUCTION

- my point of view of 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 in 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*} 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

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 "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