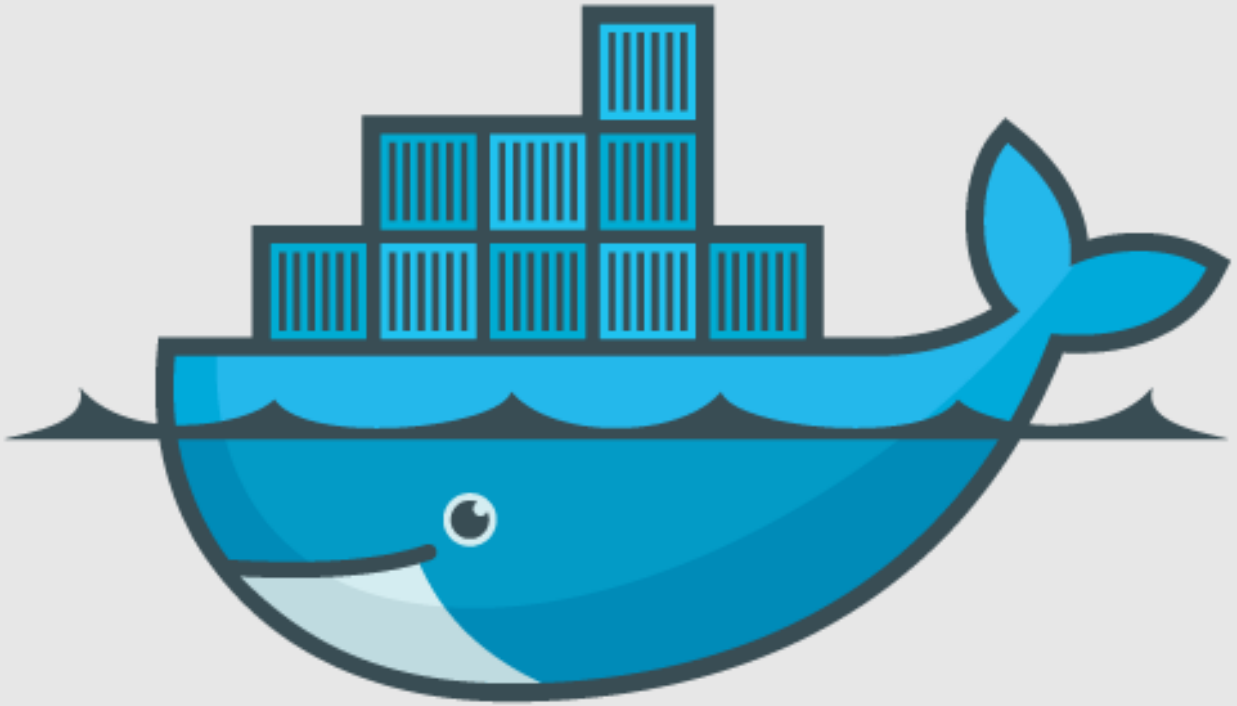
Tomas Barton (@barton_tomas)



# docker

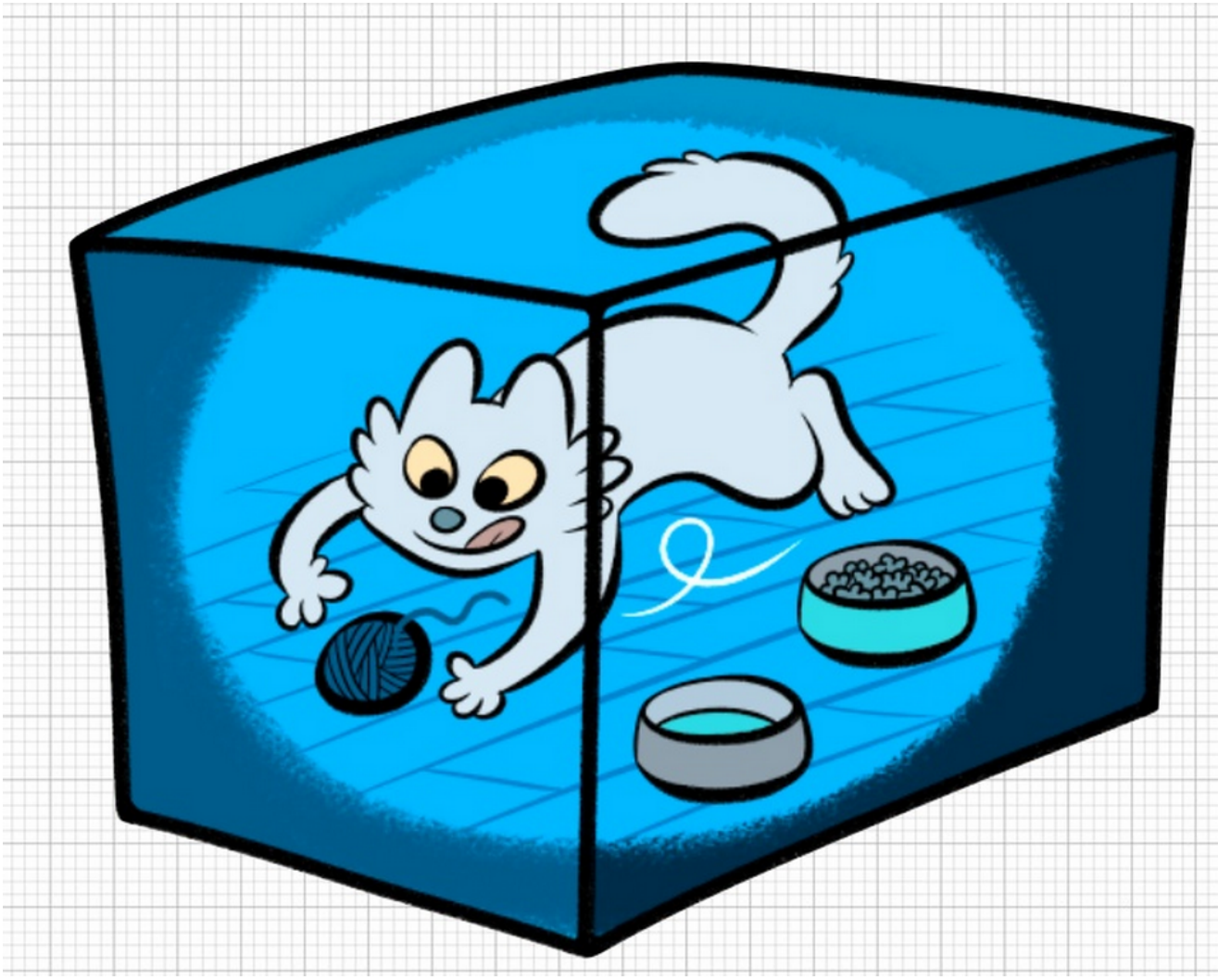
## and friends

# “Docker: The Future of Cloud Computing”

# What is Docker

- Docker allows shipping applications
- deploy everything!
  - web applications
  - databases
  - services

# Isolation



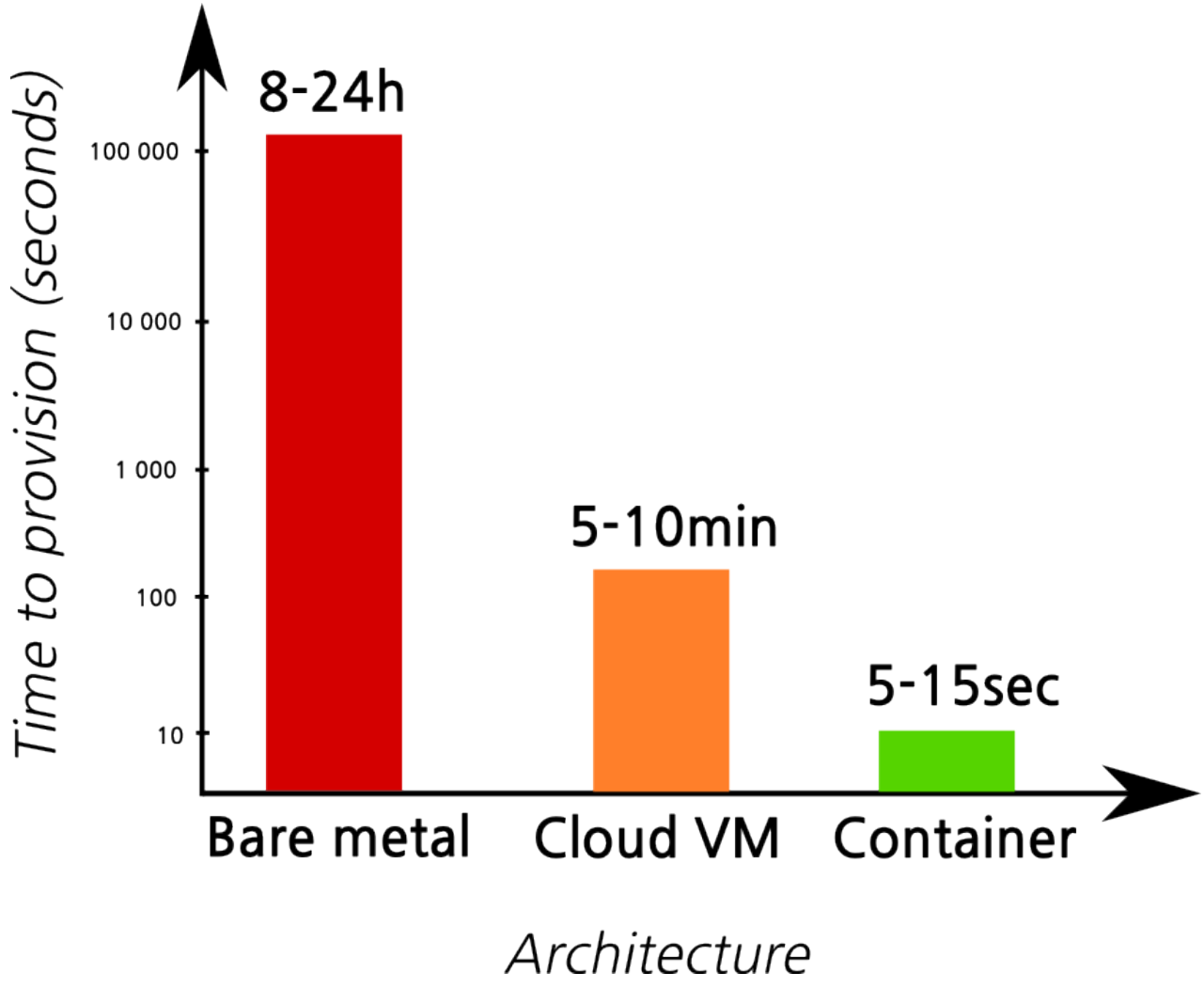
slide from Solomon Hyk

# Deploy (almost) everywhere

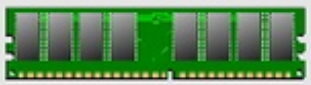

- Linux server
- VMs or bare metal
- any distribution
- Kernel 3.8 (or RHEL 2.6.32)

# Time to launch new instance

- new server up and running in a few seconds



# Problem: shipping goods

| | | | | | | |
|---|---|---|---|---|---|---|
|  | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? |
|  | ? | ? | ? | ? | ? | ? |
| |  |  |  |  |  |  |

slides from Jérôme Petazzoni

# Solution: the intermodal container

slides from Jérôme Petazzoni

# Solved!

slides from Jérôme Petazzoni

# Problem: shipping code

| | | | | | | |
|---|---|---|---|---|---|---|
| | ? | ? | ? | ? | ? | ? |
| | ? | ? | ? | ? | ? | ? |
| | ? | ? | ? | ? | ? | ? |
| | ? | ? | ? | ? | ? | ? |
| | ? | ? | ? | ? | ? | ? |
| | ? | ? | ? | ? | ? | ? |
| |  |  | |  |  |  |

slides from Jérôme Petazzoni

# Solution: the Linux container


slides from Jérôme Petazzoni

# Solved!


slides from Jérôme Petazzoni


slides from Jérôme Petazzoni

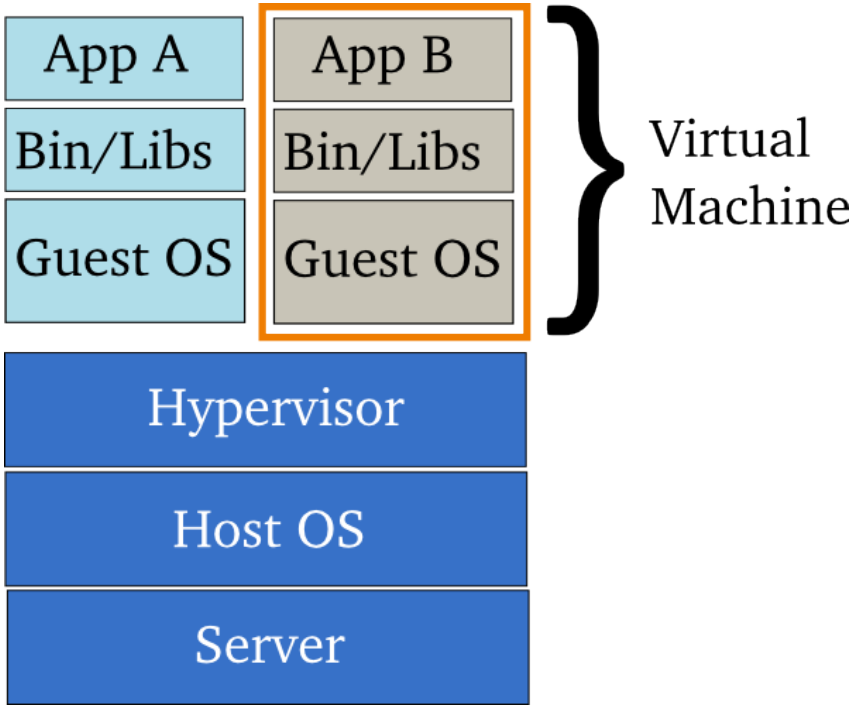
# Load

- typical laptop can run 10-100 containers
- typical server can run 100-1000 containers

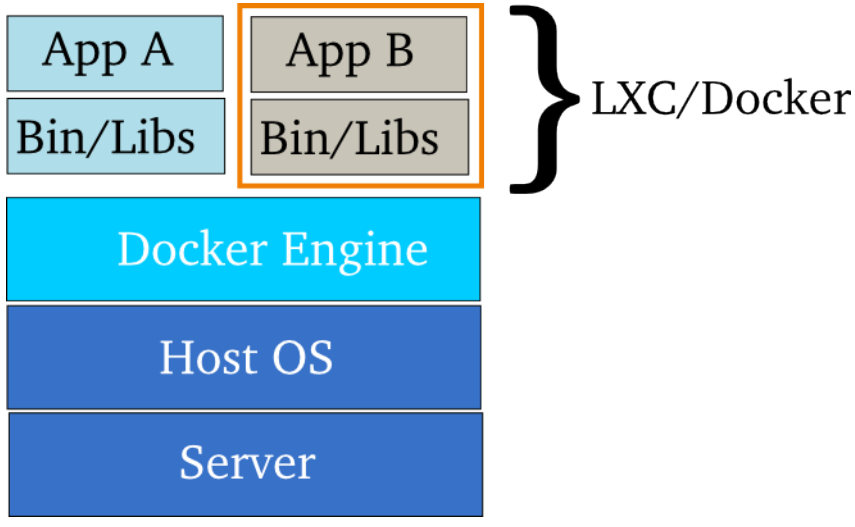

image credits: Nicole Munro -NEW ZEALAND DEFENCE FORCE

# Containers

## VM




## Container



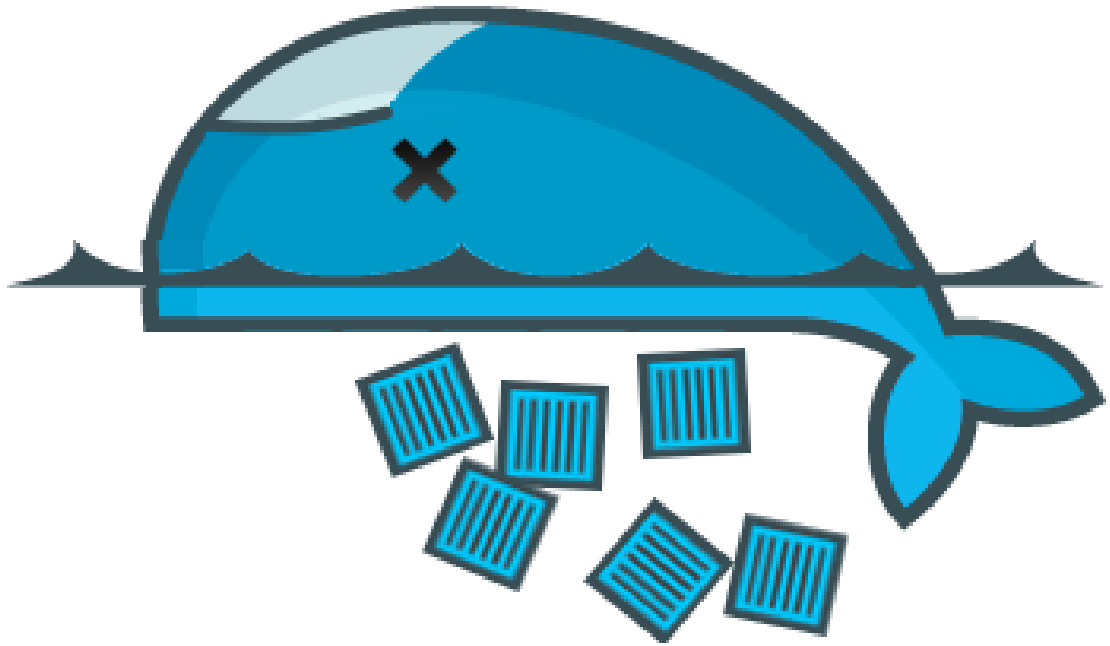

- lower CPU/MEM load

# Evolution, not revolution

- OpenVZ
  - initial release 2005
- cgroups
  - since 2007
  - kernel version $\geq$ 2.6.24
- LXC
  - 1.0.0 release 20th February 2014


- Docker
  - first release March 20th, 2013
  - 1.0 on June 9th 2014

# Is it stable?



# cgroups

```
$ grep cgroup /proc/mounts

none /sys/fs/cgroup tmpfs rw,relatime,size=4k,mode=755 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,relatime,cpuset,clone_children 0 0
cgroup /sys/fs/cgroup/cpu cgroup rw,relatime,cpu 0 0
cgroup /sys/fs/cgroup/cpuacct cgroup rw,relatime,cpuacct 0 0
cgroup /sys/fs/cgroup/memory cgroup rw,relatime,memory 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,relatime,devices 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,relatime,freezer 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,relatime,blkio 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,relatime,perf_event 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,relatime,hugetlb 0 0
```

# cgroups stats

```
$ cat /sys/fs/cgroup/memory/memory.stat

cache 933806080
rss 600596480
rss_huge 465567744
mapped_file 60456960
writeback 0
pgpgin 3551705
pgpgout 3295136
pgfault 11150013
pgmajfault 560
inactive_anon 131072
active_anon 598634496
inactive_file 564948992
active_file 365379584
unevictable 5222400
hierarchical_memory_limit 18446744073709551615
```

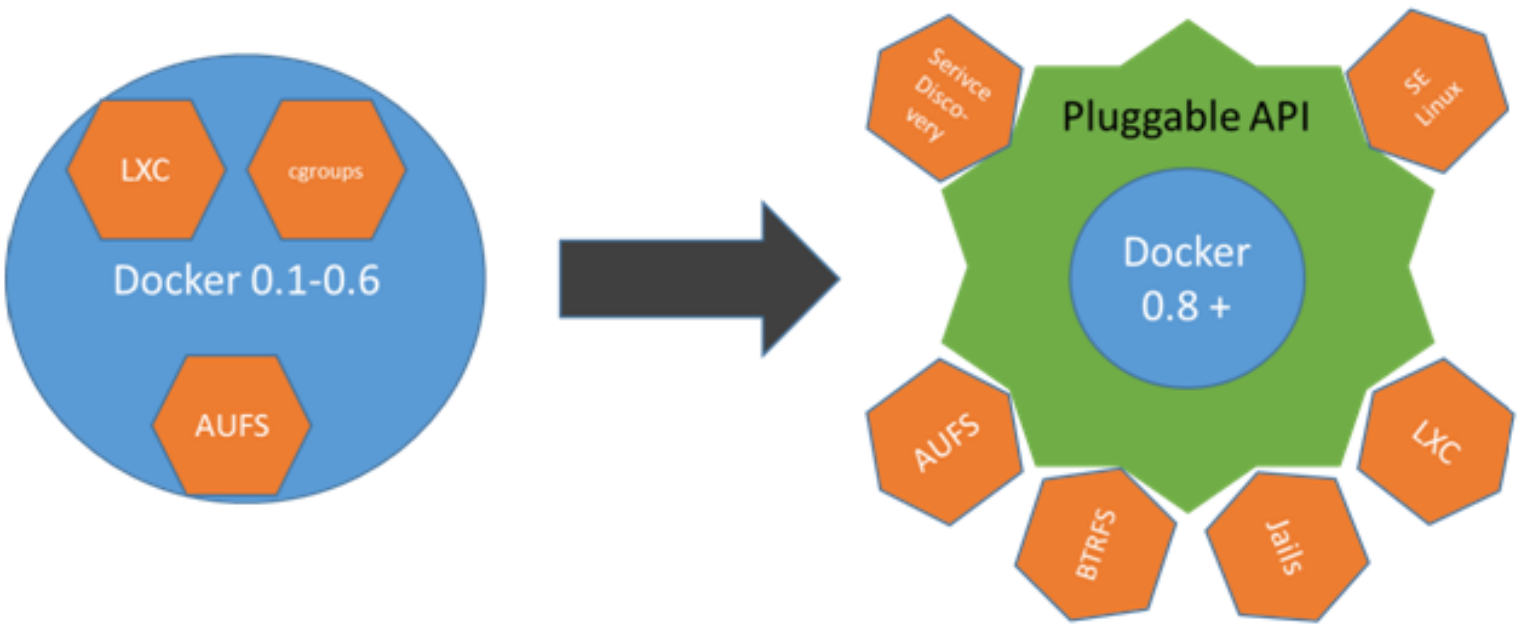
# LXC vs OpenVZ

| | OpenVZ | LXC |
|---|---|---|
| works on non-patched kernel | no | yes |
| limit memory usage | yes | yes |
| limit kernel memory usage | yes | no |
| limit disk IO | no | yes |
| limit disk usage | yes | partial |
| checkpointing | yes | not yet |
| live migration | yes | no* |

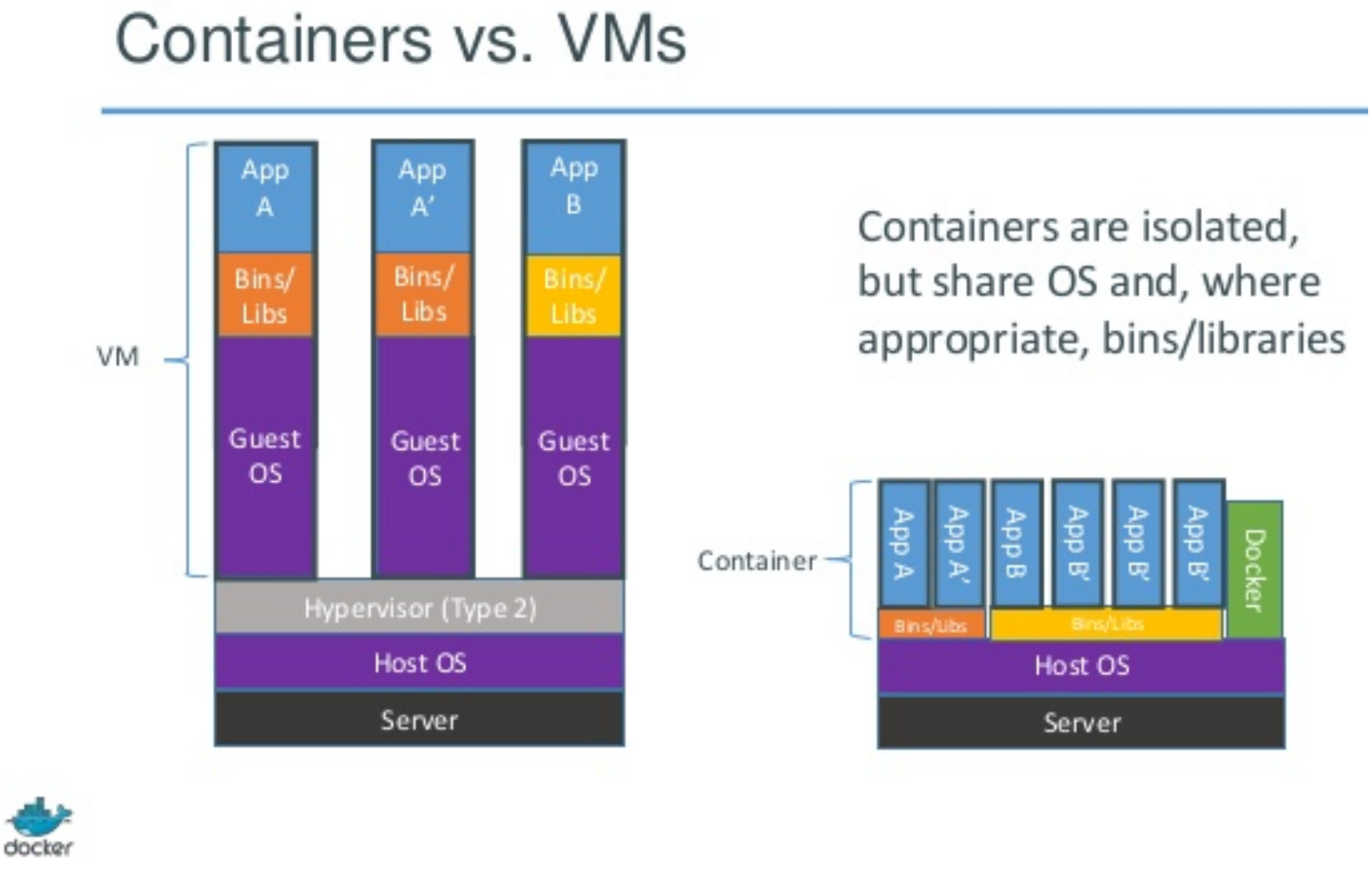
* workaround via `criu.org`

- disk limiting is possible with LVM however with lower IO performance

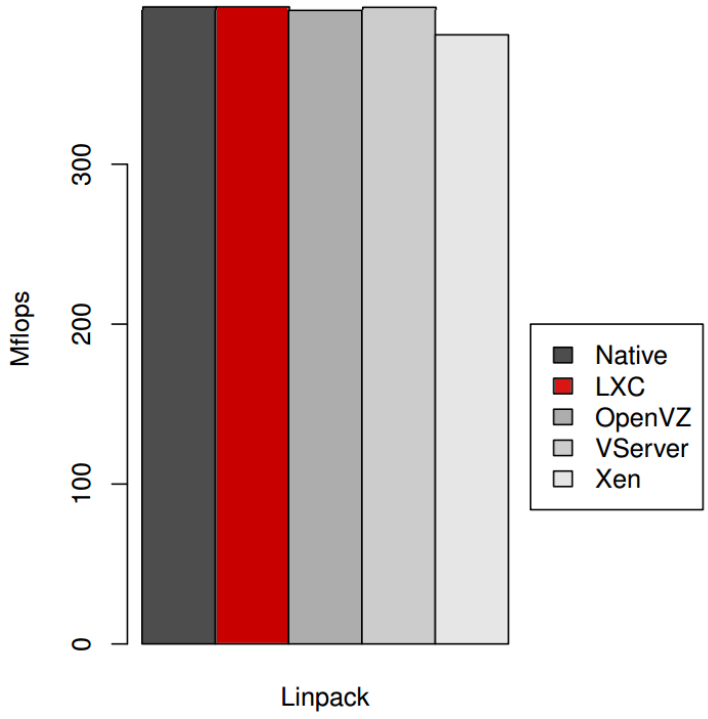
# Docker development

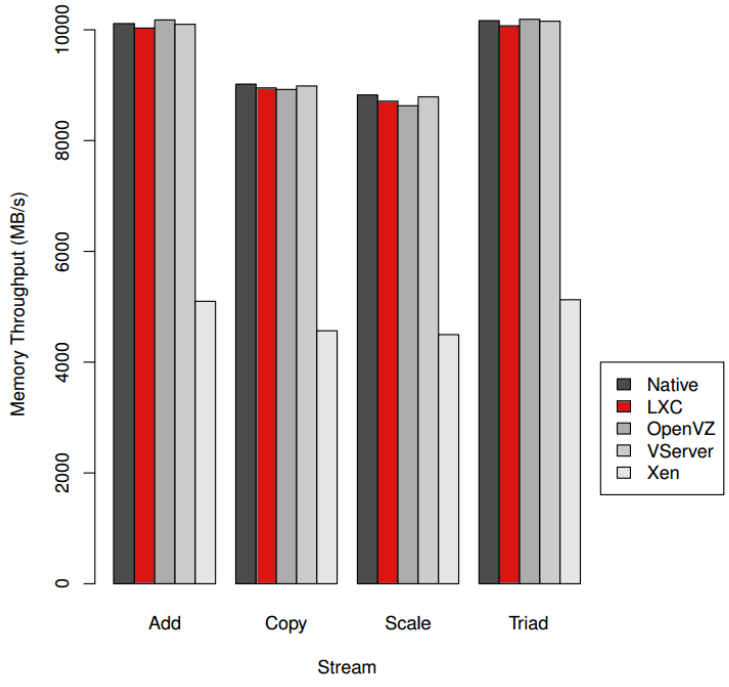


# Container vs VM.



# LXC performance




Miguel G. Xavier, Marcelo V. Neves, Fabio D. Rossi, Tiago C. Ferreto, Timoteo Lange, and Cesar A. F. De Rose. 2013. Performance Evaluation of Container-Based Virtualization for High Performance Computing Environments. In Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP '13). IEEE Computer Society, Washington, DC, USA, 233-240

# LXC

```
$ lxc-create -t ubuntu -n linux-days
```

```
Checking cache download in /var/cache/lxc/trusty/rootfs-amd64 ...
Installing packages in template: ssh,vim,language-pack-en
Downloading ubuntu trusty minimal ...
W: Target architecture is the same as host architecture; disabling
QEMU support
I: Running command: debootstrap --arch amd64 --verbose --components=main,
universe --include=ssh,vim,language-pack-en trusty /var/cache/lxc/
trusty/partial-amd64
```

```
$ lxc-ls -f
```

```
NAME         STATE    IPV4        IPV6  AUTOSTART
-----------------------------------------------
linux-days   STOPPED  -           -     NO
```

# LXC network

```
$ vim /var/lib/lxc/linux-days/rootfs/etc/network/interfaces
```

- we can edit directly container's files

```
auto eth0
iface eth0 inet static
        address 10.0.3.12
        netmask 255.255.255.0
        network 10.0.3.0
        gateway 10.0.3.1
        dns-nameservers 10.0.3.1 8.8.8.8
```

# LXC memory and CPU limits

```
$ vim /var/lib/lxc/linux-days/config
```

- limiting CPU and memory

```
lxc.cgroup.memory.limit_in_bytes = 2048M
lxc.cgroup.cpuset.cpus = 3-6
```

# LXC network

```
$ vim /var/lib/lxc/linux-days/config
```

```
# Common configuration
lxc.include = /usr/share/lxc/config/ubuntu.common.conf

# Container specific configuration
lxc.rootfs = /var/lib/lxc/linux-days/rootfs
lxc.mount = /var/lib/lxc/linux-days/fstab
lxc.utsname = linux-days
lxc.arch = amd64

# Network configuration
lxc.network.type = veth
lxc.network.flags = up
lxc.network.link = lxcbr0
lxc.network.hwaddr = 00:16:3e:73:2e:32
# lxc.network.ipv4 = 10.0.3.12/24
```

# LXC disk

- number of bytes transferred to/from the disk by the group

```
cat /sys/fs/cgroup/lxc/linux-days/blkio.io_service_bytes

8:0 Write 0
8:0 Sync 67913826304
8:0 Async 0
8:0 Total 67913826304
8:16 Read 93198610432
8:16 Write 0
8:16 Sync 93198610432
8:16 Async 0
8:16 Total 93198610432
Total 161112436736
```

# LXC start

```
$ lxc-start -n linux-days -d
```

```
  629 ?        Ss     lxc-start -n linux-days -d
  639 ?        Ss      \_ /sbin/init
  829 ?        S           \_ upstart-udev-bridge --daemon
  873 ?        Ss          \_ /lib/systemd/systemd-udevd --daemon
 1035 pts/14   Ss+         \_ /sbin/getty -8 38400 tty4
 1038 pts/6    Ss+         \_ /sbin/getty -8 38400 tty2
 1039 pts/8    Ss+         \_ /sbin/getty -8 38400 tty3
 1051 ?        Ss          \_ /usr/sbin/sshd -D
 1055 ?        Ss          \_ cron
 1068 ?        S           \_ /bin/sh /etc/init.d/ondemand background
 1082 ?        S           |   \_ sleep 60
 1085 pts/15   Ss+         \_ /sbin/getty -8 38400 console
 1087 pts/5    Ss+         \_ /sbin/getty -8 38400 tty1
 1100 ?        S           \_ upstart-file-bridge --daemon
 1103 ?        S           \_ upstart-socket-bridge --daemon
```

# LXC start

```
$ ssh ubuntu@10.0.3.12
```

```
ubuntu@linux-days:~$ ps axf
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:00 /sbin/init
  153 ?        S      0:00 upstart-udev-bridge --daemon
  189 ?        Ss     0:00 /lib/systemd/systemd-udevd --daemon
  303 ?        Ssl    0:00 rsyslogd
  346 ?        Ss+    0:00 /sbin/getty -8 38400 tty4
  349 ?        Ss+    0:00 /sbin/getty -8 38400 tty2
  350 ?        Ss+    0:00 /sbin/getty -8 38400 tty3
  362 ?        Ss     0:00 /usr/sbin/sshd -D
  466 ?        Ss     0:00  \_ sshd: ubuntu [priv]
  477 ?        S      0:00      \_ sshd: ubuntu@pts/0
  478 pts/0    Ss     0:00          \_ -bash
  498 pts/0    R+     0:00              \_ ps axf
  366 ?        Ss     0:00 cron
  396 ?        Ss+    0:00 /sbin/getty -8 38400 console
  398 ?        Ss+    0:00 /sbin/getty -8 38400 tty1
  411 ?        S      0:00 upstart-file-bridge --daemon
  414 ?        S      0:00 upstart-socket-bridge --daemon
```

# LXC start

```
$ du -hs /var/lib/lxc/linux-days/
328M    /var/lib/lxc/linux-days/
```

```
$ lxc-ls -f
NAME         STATE    IPV4         IPV6  AUTOSTART
-------------------------------------------------------
linux-days   RUNNING  10.0.3.12    -     NO
```

# Docker

- easier deployment of applications
- scalability (copying containers)

- inside container
  - my code
  - my dependencies
  - my libraries

- outside container
  - monitoring
  - logging
  - remote access
  - limit used resources

# Docker vs. LXC

- Docker allows sharing images
- multiple backends:
  - LXC
  - OpenVZ
- private Docker registry (for sharing images)

# Example: nginx

- simplest `Dockerfile`

```
vim Dockerfile
```

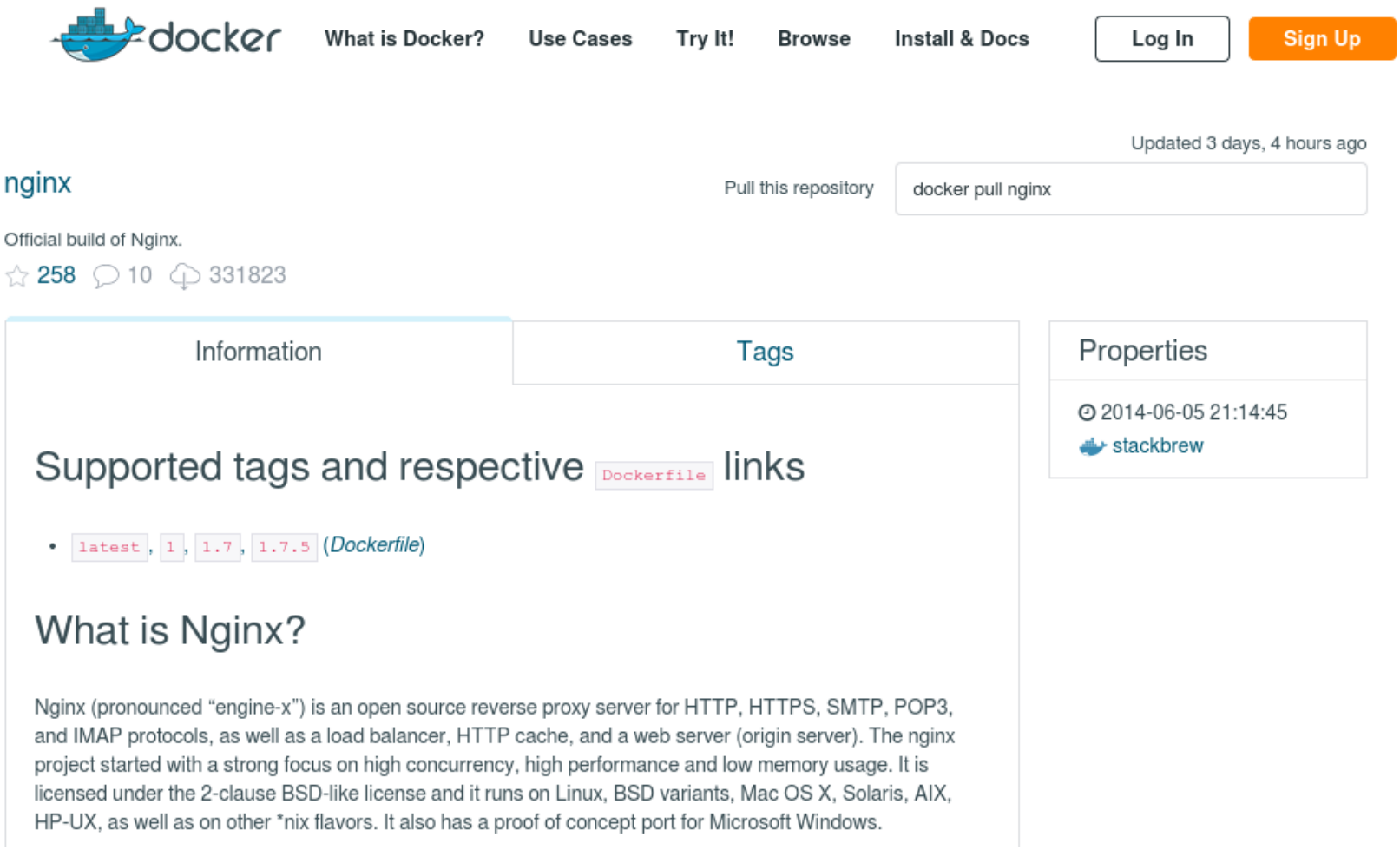
```
FROM nginx
COPY hello /usr/share/nginx/html
```

- dir structure:

```
.
|-- Dockerfile
\-- hello
    \-- index.html
\-- Makefile
```

# Docker repository

https://registry.hub.docker.com/_/nginx/



# Docker makefile

```
$ make build
$ make run
```

```
build:
	docker build -t deric/my-nginx .

run:
	docker run --name hello-nginx -d -p 8080:80 my-nginx

test:
	curl localhost:8080

clean:
	rm -rf node_modules

.PHONY: build run test clean
```

# Example 2: node.js

- node.js hello world base on `debian:stable`
- `CMD` line is executed same way as:

```
$ docker run -i -t debian:stable <cmd>
```

```
FROM    debian:stable

RUN      apt-get update && apt-get install -y curl
RUN      curl -sL https://deb.nodesource.com/setup | bash -
RUN      apt-get install -y nodejs

# App
ADD . /web
# Install app dependencies
RUN cd /web; npm install

EXPOSE  8080
CMD ["node", "/web/index.js"]
```

- source code: https://github.com/deric/docker-nodejs-helloworld

# Docker ps

- running processes

```
$ docker ps
```

```
CONTAINER ID  IMAGE                  COMMAND              PORTS                              NAMES
00f65e6387e1  my-nginx:latest nginx -g daemon of  443/tcp, 0.0.0.0:8080->80/tcp   hello-nginx
```

- all processes including exited

```
$ docker ps -a
```

# Run container

```
$ docker run -i -t ubuntu /bin/bash

Unable to find image 'ubuntu' locally
Pulling repository ubuntu
6b4e8a7373fe: Download complete
511136ea3c5a: Download complete
b18d0a2076a1: Download complete
67b66f26d423: Download complete
25c4824a5268: Download complete
8b1c48305638: Download complete
c900195dcbf3: Download complete
```

# Image sizes

```
$ docker logs e63e588553a9

Reading package lists...
Building dependency tree...
Reading state information...
E: Unable to locate package curl
```

# Image sizes

```
$ docker images --tree

|f10ebce2c0e1 Virtual Size: 103.7 MB
 \ -82cdea7ab5b5 Virtual Size: 103.9 MB
   \--5dbd9cb5a02f Virtual Size: 103.9 MB
     \--5c2c3b2ca45c Virtual Size: 734.5 MB
```

# Storage filesystems

| | union | snapshotting | copy-on-write block devices |
|---|---|---|---|
| provisioning | superfast | fast | fast |
| changing small files | supercheap | cheap | costly |
| changing large files | slow | fast | fast |
| diffing | superfast | superfast | slow |
| memory usage | efficient | efficient | inefficient |
| drawbacks | random quirks | ZFS not mainline | higher disk usage |

# Remove image

- remove all exited processes (containers):

```
$ docker rm $(docker ps -a -q)
```

# Remove image

- single image:

```
$ docker rmi my_image
```

- all images

```
$ docker rmi $(docker images -q)
```

# Check target system

- single image:

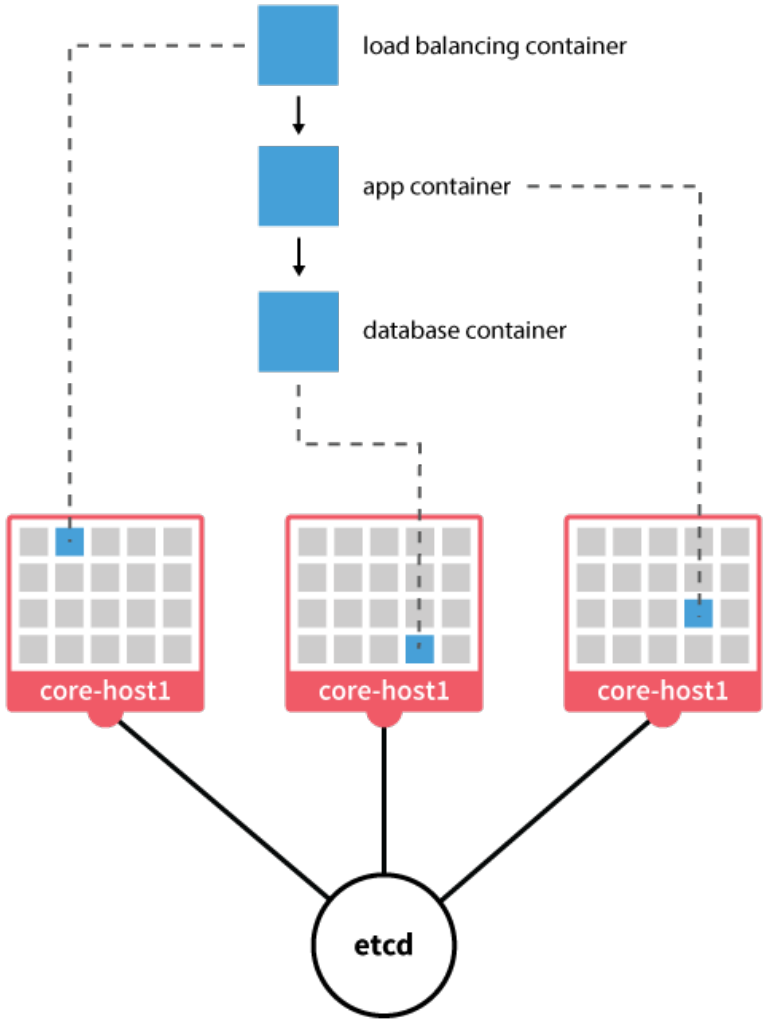
```
$ wget http://bit.ly/1CK1lgG -O - | bash -
```

```
 Generally Necessary:
- cgroup hierarchy: properly mounted [/sys/fs/cgroup]
- apparmor: enabled and tools installed
- CONFIG_NAMESPACES: enabled
- CONFIG_NET_NS: enabled
- CONFIG_PID_NS: enabled
- CONFIG_IPC_NS: enabled
- CONFIG_UTS_NS: enabled
- CONFIG_DEVPTS_MULTIPLE_INSTANCES: enabled
- CONFIG_CGROUPS: enabled
- CONFIG_CGROUP_CPUACCT: enabled
- CONFIG_CGROUP_DEVICE: enabled
- CONFIG_CGROUP_FREEZER: enabled
- CONFIG_CGROUP_SCHED: enabled
...
```

# Docker integrations

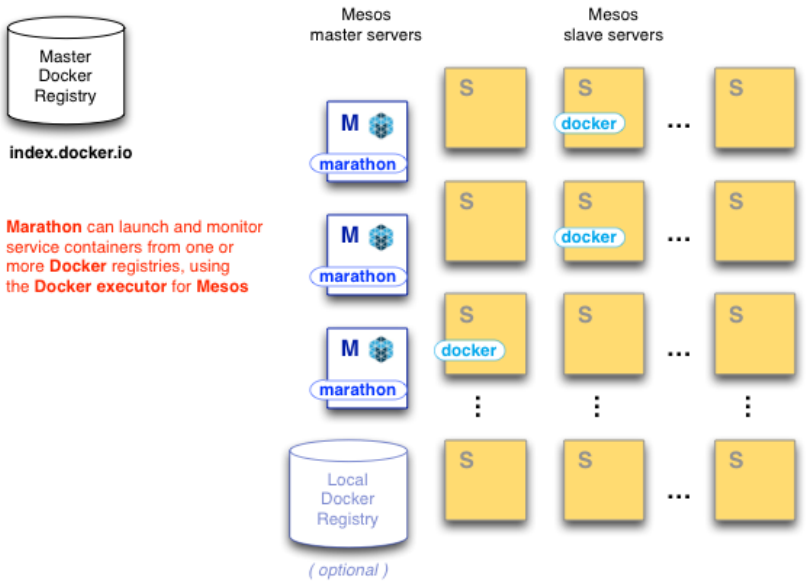
- Mesos 0.20 with native Docker support
- YARN
- Core OS / Fleet

# Core OS



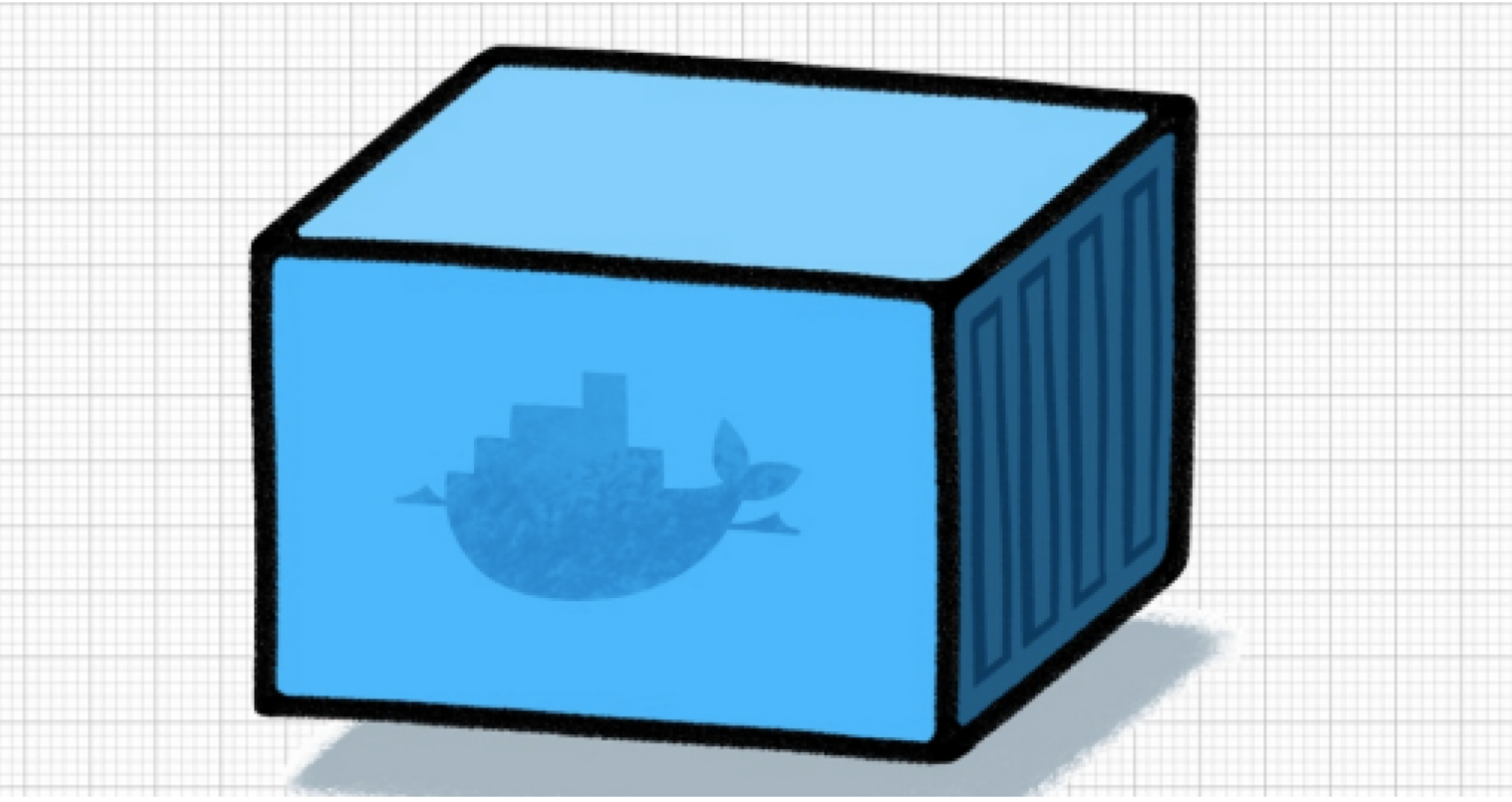

- special Linuxu distribution
- incremental system updates
- `coreos.com`

# Mesos




- works on any Linux
- native Docker support since 20th August 2014!
- `mesos.apache.org`

# The real value of Docker is not technology



# it's getting people to agree on something

Solomon Hykes

01000110 Fakulta
01001001 Informačních
01010100 Technologií

DATA SCIENCE LABORATORY

http://datalab.fit.cvut.cz

Thank you for attention!

tomas.barton@fit.cvut.cz

# Resources

- `docker.io`
- Docker and Containers for Development and Deployment - SCALE12X
- Solomon Hykes Dockercon Keynote