

LINUX CONTAINERS

Tomáš Tomeček (Developer Experience)

Peter Schiffer (AtomicOpenshift)

Josef Karásek (xPaaS)

ABOUT THIS COURSE

- Docker for Workstation
 - Running and building containers on a single host
 - Managing Docker engine
 - Bit of theory, history, future and alternatives
- Prerequisites
 - Linux knowledge ~ RHCSA
 - Notebook during the course is optional
- github.com/josefkarasek/docker101
- We need your feedback!
 - schiffer.typeform.com/to/lNHegp

SESSION 1

- Intro
- History
- Why containers
- Containers without Docker
- Docker basics

INTRODUCTION TO LINUX CONTAINERS

WHAT IS A CONTAINER?



WHAT IS A CONTAINER?

actually, there is no container

but there are constrained applications*

*application = 1 or more running processes

WHEN WE TALK ABOUT CONTAINERS

we talk about multiple Linux kernel features
configured together for set of processes

HISTORY

- 2000 - [FreeBSD jail](#)
- 2005 - [OpenVZ](#)
- 2007 - [cgroups](#) & [Kernel namespaces](#)
- 2008 - [LXC](#)
- 2011 - [OpenShift](#)
- 2013 - [Docker](#)
- 2014 - [Kubernetes](#)

rhelblog.redhat.com/2015/08/28/the-history-of-containers
youtu.be/wW9CAH9nSLs - first announcement of Docker [5min]

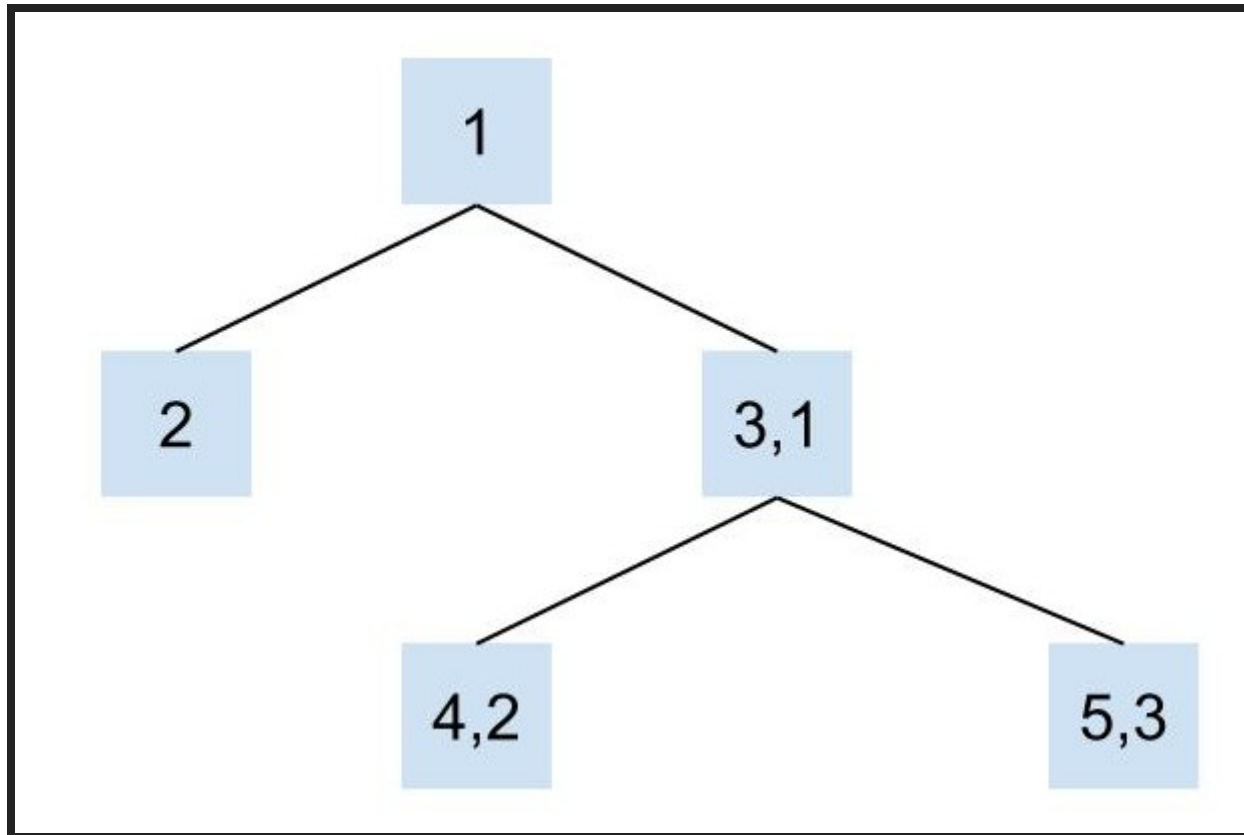
UNDERLYING KERNEL FEATURES

A process with certain properties

- **namespaces** - limit what a process can see (pid, net, user, ...)
- **cgroups** - limit how much a process can use (cpu, mem, i/o)
- **copy-on-write** - instant start, tracking changes (overlayfs, ...)
- **POSIX capabilities** - managing root permissions
- **SELinux** - let containers contain
- **seccomp** - syscall filtering

NAMESPACES

PID namespacing



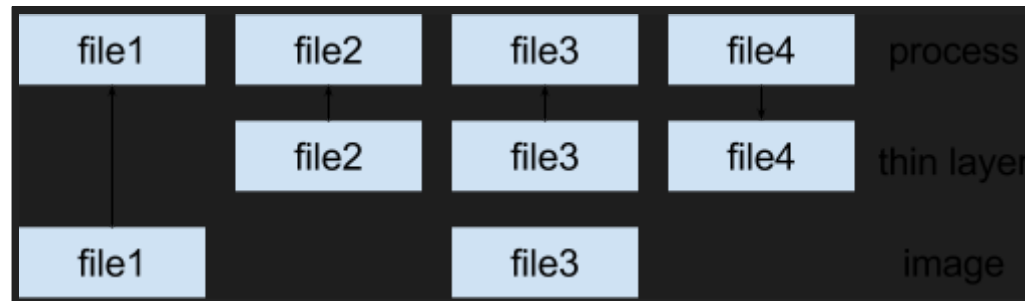
also mnt, net, ipc, uts, user

CGROUPS

- cpu share
- cpu lock
- memory allocation
 - soft vs. hard limits
- read/write speed
- devices cgroup
- freezer cgroup

COPY-ON-WRITE

- Instead of copying whole file system
- Storage keeps track of changes
- BTRFS, ZFS, device mapper, overlayfs, aufs



POSIX CAPABILITIES

- More granular permission checks than the traditional **PRIVILEGED** vs **UNPRIVILEGED** user
- Containers running as UID 0 less harmful
- For more info check man page: `man capabilities`

```
$ getcap /usr/bin/ping  
/usr/bin/ping = cap_net_admin,cap_net_raw+ep
```

SELINUX

- Enforcing container separation
- Labeling system
- Multi level security

SECURE COMPUTING

syscall filtering

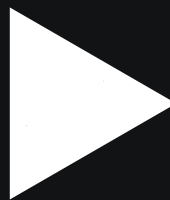
- SECCOMP_SET_MODE_STRICT - only read, write, exit and sigreturn
- SECCOMP_SET_MODE_FILTER - whitelist only some syscalls
- Example: what about **reboot**?
- Some calls are privileged - gated by POSIX capabilities

WHAT IS A CONTAINER GOOD FOR?

- Application with all its dependencies
- Clean environment in fraction of a second
- Isolation mitigates binary interference - distribution and running multiple versions of programs
- Easy deployment - rolling updates, rollback to previous versions

CONTAINERS WITHOUT DOCKER

```
ttomecek at quahog ~/t/redis-container sudo unshare --fork --pid /bin/bash --noprofile --norc
bash-4.3# chroot . /bin/bash
bash-4.3# source /etc/skel/.bashrc
[root@quahog /]# ps aux
Error, do this: mount -t proc proc /proc
[root@quahog /]# mount -t proc proc /proc
```



E 10.34.4.111/23

1* bash

4.78GB/7.68GB :: 0.53 0



00:00



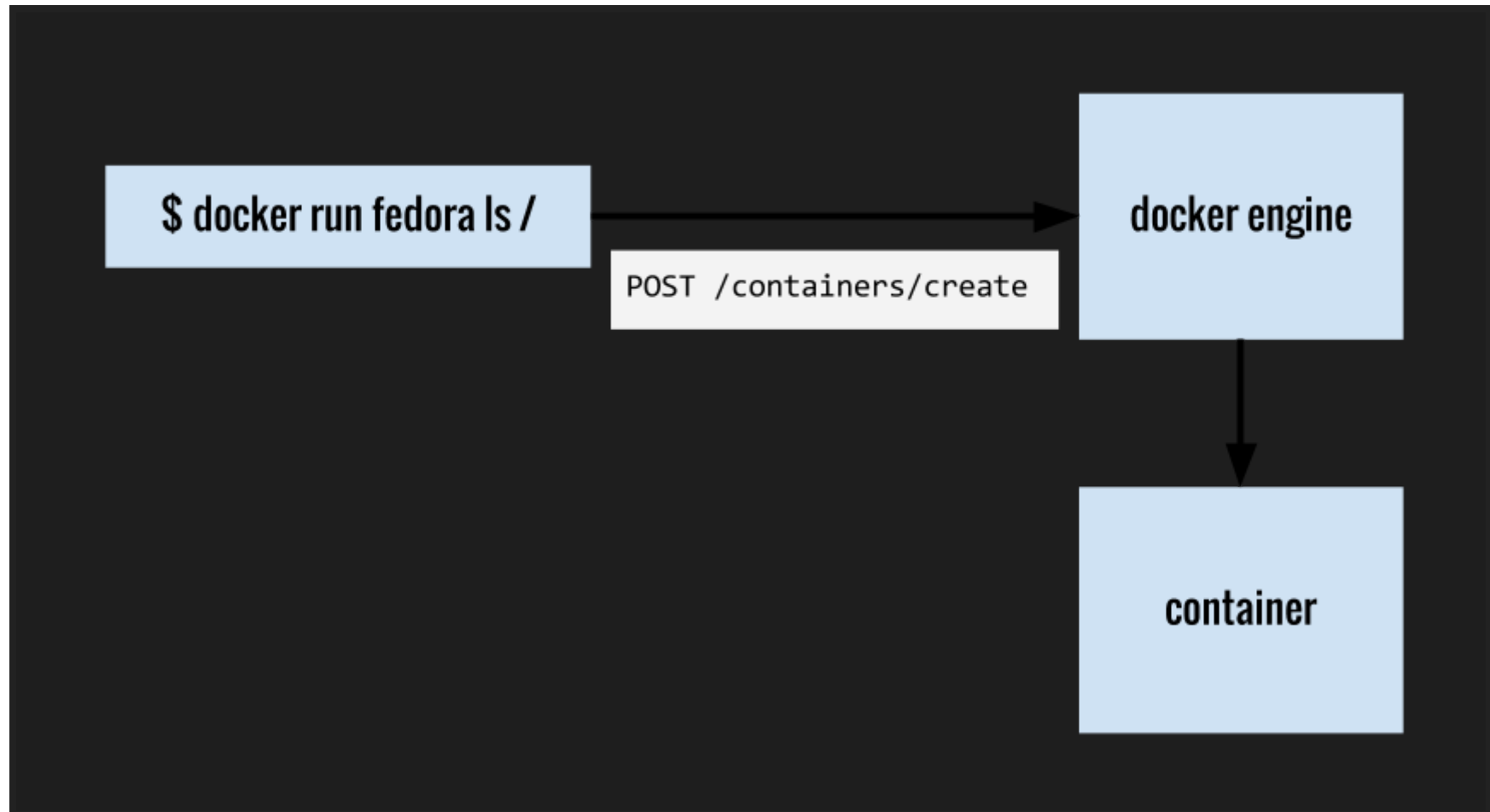
INTRODUCTION TO DOCKER

platform for running, shipping and building containers

BASICS

- Docker image
- Image operations
- Containers operations
- Networking
- Volumes

DOCKER ENGINE ARCHITECTURE



DOCKER IMAGE

An archive containing:

- Minimal OS for installing and running applications
- Application with all dependencies

Image is uniquely identified by:

- Image registry
- Author
- Image name
- Image tag (version)

```
registry.access.redhat.com/jboss-eap-6/eap64-openshift:1.2
```

DOCKER IMAGE II

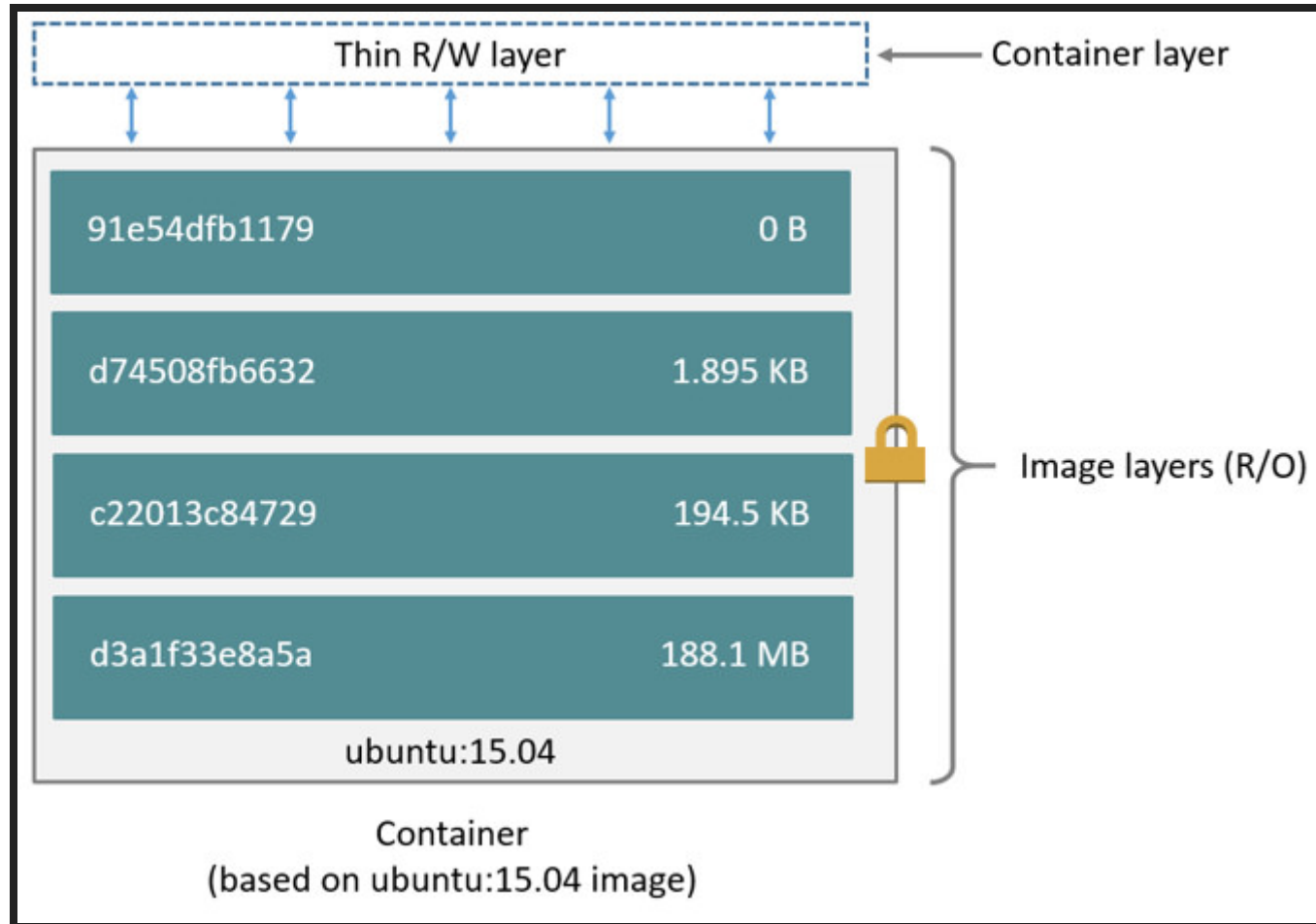


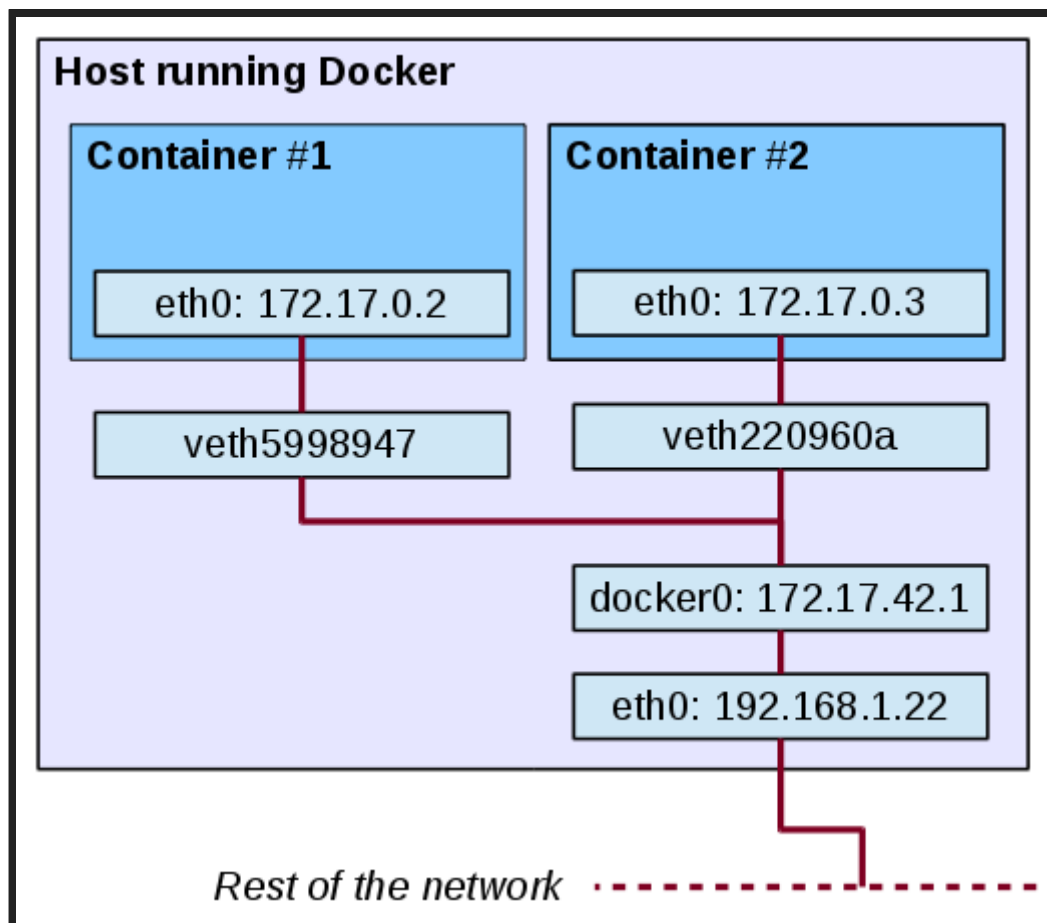
IMAGE OPERATIONS

- `pull` — Downloads and updates Docker images
- `images` — Lists all Docker images on the host
- `rmi` — Remove specified Docker image from the host

CONTAINER OPERATIONS

- `create` — Creates stopped container from the image
- `start` — Starts specified container
- `run` — `create` and `start` in one command
- `ps` — Lists containers available on the host
- `rm` — Remove specified Docker container from the host
- This is a topic of next session

NETWORKING



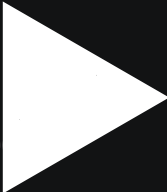
VOLUMES

Containers are ephemeral. Data persistence:

- Named volume
- Unnamed volume
- Bind mount
- Data-only containers

DEMO — DOCKER BASICS

```
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:02",
"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "a45c152c4babad17fd041d6a35b29003043ce1b0ce179692fce5798be0576d56",
    "EndpointID": "b07e6e1ac486423b1ddab737d66ba7633da35bd702ad2a25bee071b6f33f28ed",
    "Gateway": "172.17.0.1",
    "IPAddress": "172.17.0.2",
    "IPPrefixLen": 16,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "02:42:ac:11:00:02"
  }
}
```



```
tt at oat ~/g/docker101 ncat 172.17.0.2 6379
set key welcome
+OK
get key
$7
welcome
tt at oat ~/g/docker101 docker stop ab971a105c46cd050b13d94374eb7d2bfff7779129ae8f01357d875c6e280960d
ab971a105c46cd050b13d94374eb7d2bfff7779129ae8f01357d875c6e280960d
tt at oat ~/g/docker101 docker
```

(gh-pages | +1.)

(gh-pages | +1.)

00:00

DOCKER HUB

- Big value of Docker Inc.
- (Free) repository with images
- Can also build images
- Quality varies
 - vulnerabilities
 - out of date content
- Demo: <https://hub.docker.com/r/pschiffe/docker101-gcc/builds/>

QUESTIONS?

NEXT TIME: SESSION 2

May 17th, 1pm, Tower Big

THANK YOU!

github.com/josefkarasek/docker101

schiffer.typeform.com/to/lNHegp

