Kubernetes 101

Oleg Fiksel | Michael Siebertz

Email: oleg@fiksel.info | mail@michaelsiebertz.de
Matrix: @oleg:fiksel.info | @captain.vsan:matrix.org

2019-08-10 FrOSCon 2019

OLEG FIKSEL (DOCKER FUNDAMENTALS)

OLEG FIKSEL (DOCKER FUNDAMENTALS)

- DevOps Engineer
- Passionate about tech
- Like to automate everything

Michael Siebertz
(Kubernetes Fundamentals + Live Demo)

MICHAEL SIEBERTZ (KUBERNETES FUNDAMENTALS + LIVE DEMO)

- DevOps Engineer
- Passionate about tech (IT / Car)
- Trying to automate as much as possible
- Part of Central DevOps Team

MOTIVATION

Our motivation to do this talk

MOTIVATION

Our motivation to do this talk

Provide an overview of container architechture on the example of Docker and Kubernetes.

AGENDA

AGENDA

- Docker Fundamentals

AGENDA

- Docker Fundamentals
- Kubernetes Fundamentals

AGENDA

- Docker Fundamentals
- Kubernetes Fundamentals
- Roadmap from KubeCon 2019

AGENDA

- Docker Fundamentals
- Kubernetes Fundamentals
- Roadmap from KubeCon 2019
- Live Demo

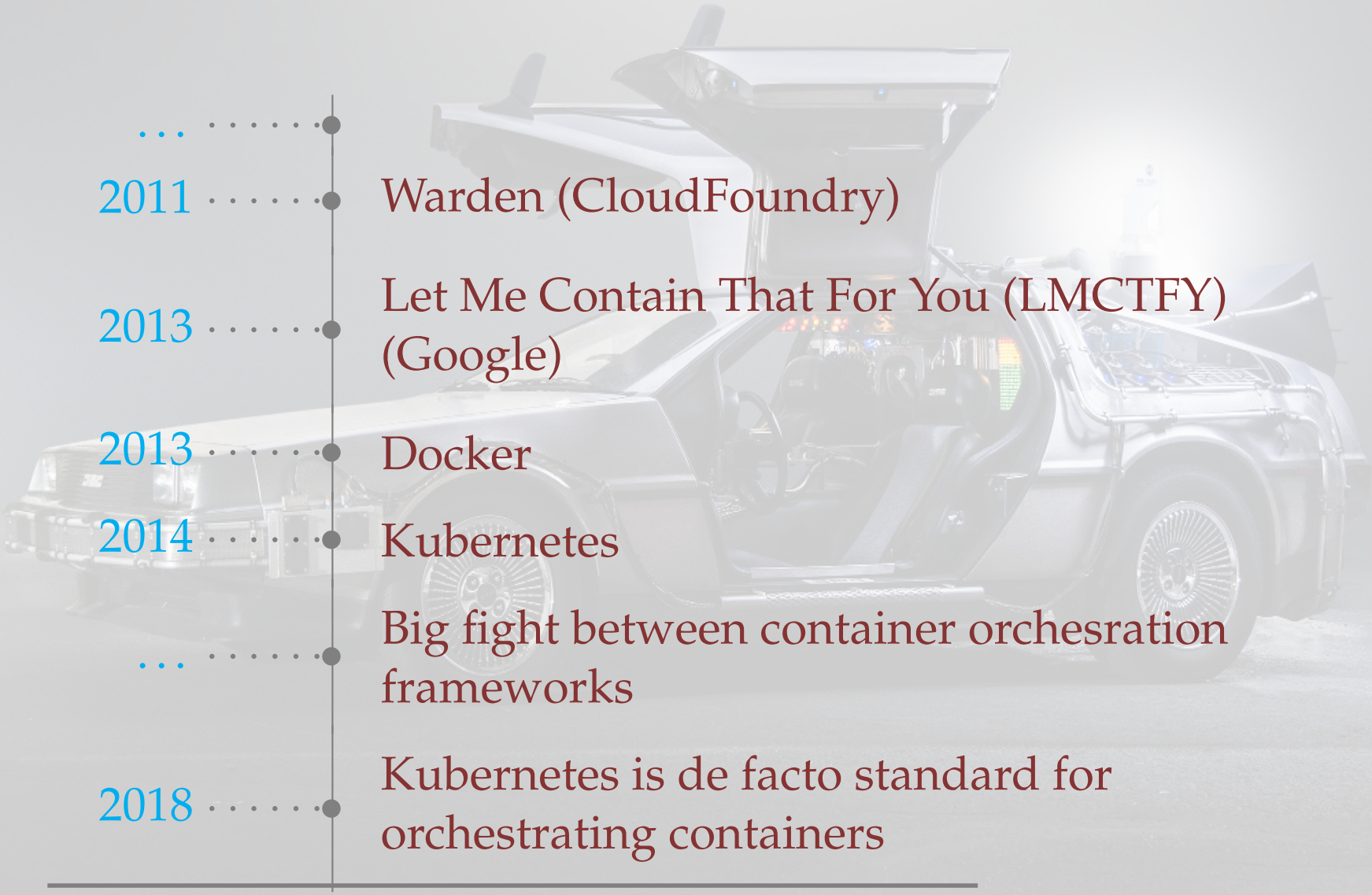
Docker fundamentals

A BIT OF HISTORY

A BIT OF HISTORY 1/2

- 1979 — chroot
- 2000 — FreeBSD Jails
- 2001 — Linux VServer
- 2004 — Solaris Containers
- 2005 — OpenVZ
- 2006 — Process Containers (Google)
- 2008 — LXC
- ... —

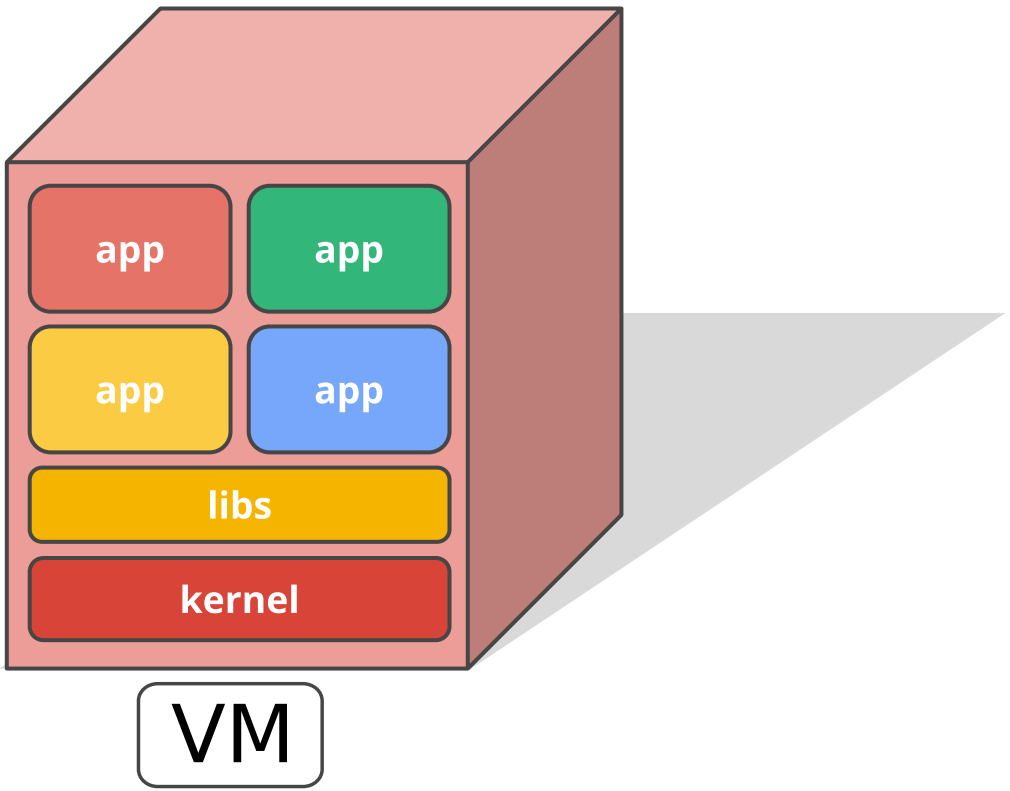
A BIT OF HISTORY 2/2



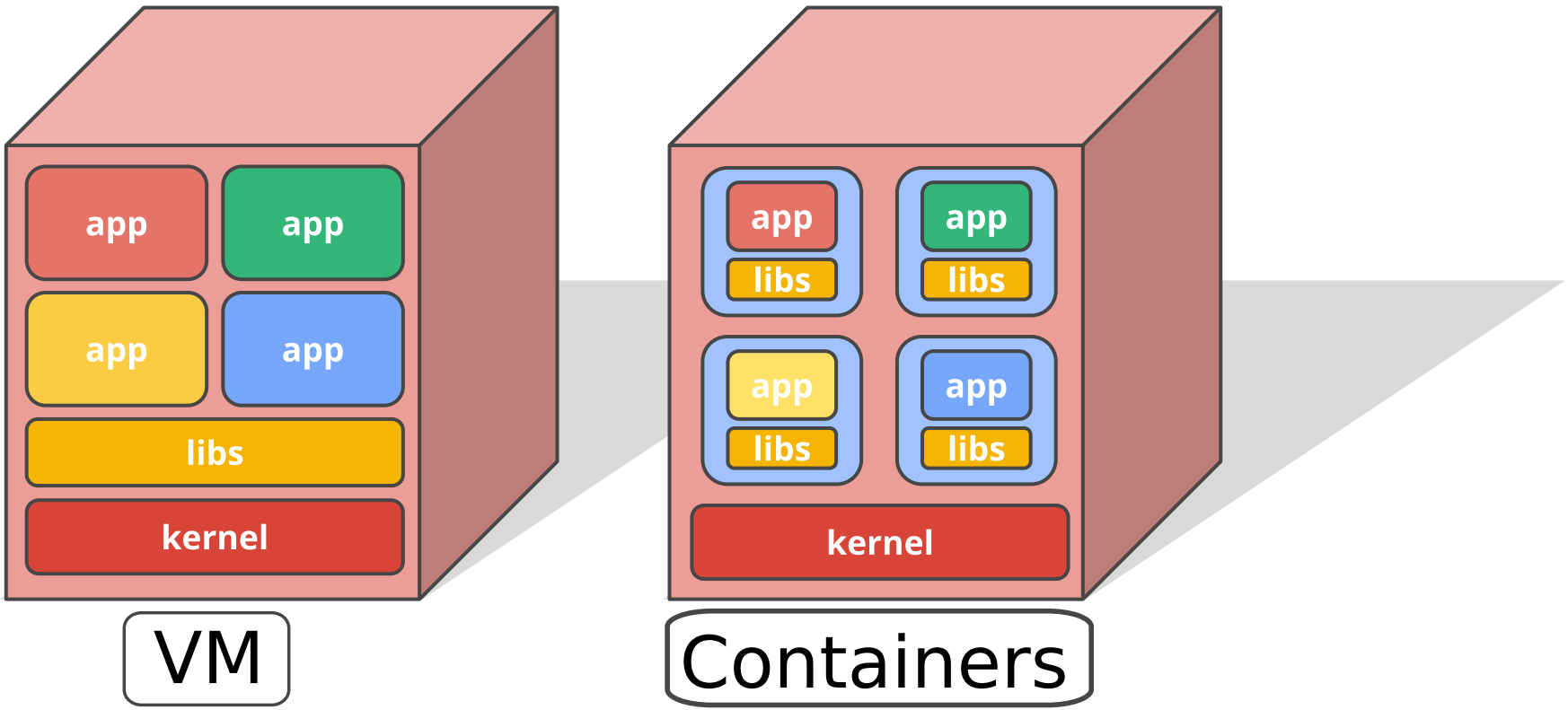
- ...
- 2011 — Warden (CloudFoundry)
- 2013 — Let Me Contain That For You (LMCTFY) (Google)
- 2013 — Docker
- 2014 — Kubernetes
- ... — Big fight between container orchesration frameworks
- 2018 — Kubernetes is de facto standard for orchestrating containers

VMs and Containers

VMs and Containers



VMs and Containers



VMs and Containers

VMs and Containers

- Storage

VMs and Containers

- Storage
  - VM: Binary Image

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation
- Networking

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation
- Networking
  - VM: NIC emulation

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation
- Networking
  - VM: NIC emulation
  - Container: Uses network stack of the host

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation
- Networking
  - VM: NIC emulation
  - Container: Uses network stack of the host
- Runtime

VMs and Containers

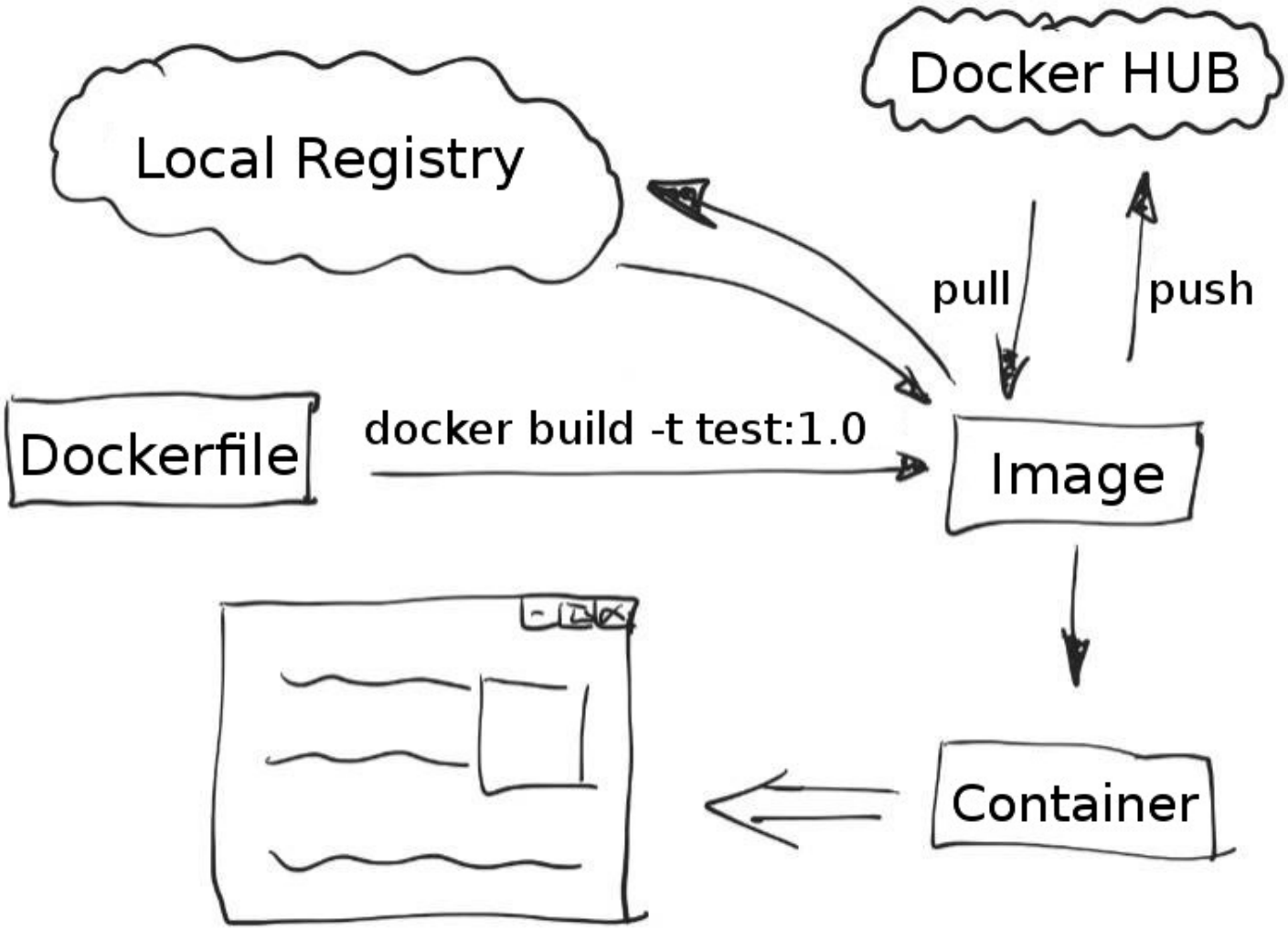
- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation
- Networking
  - VM: NIC emulation
  - Container: Uses network stack of the host
- Runtime
  - VM: stateful

VMs and Containers

- Storage
  - VM: Binary Image
  - Container: Layered plain FS
- Emulation
  - VM: BIOS, Hardware, etc.
  - Container: No emulation, just separation
- Networking
  - VM: NIC emulation
  - Container: Uses network stack of the host
- Runtime
  - VM: stateful
  - Container: stateless

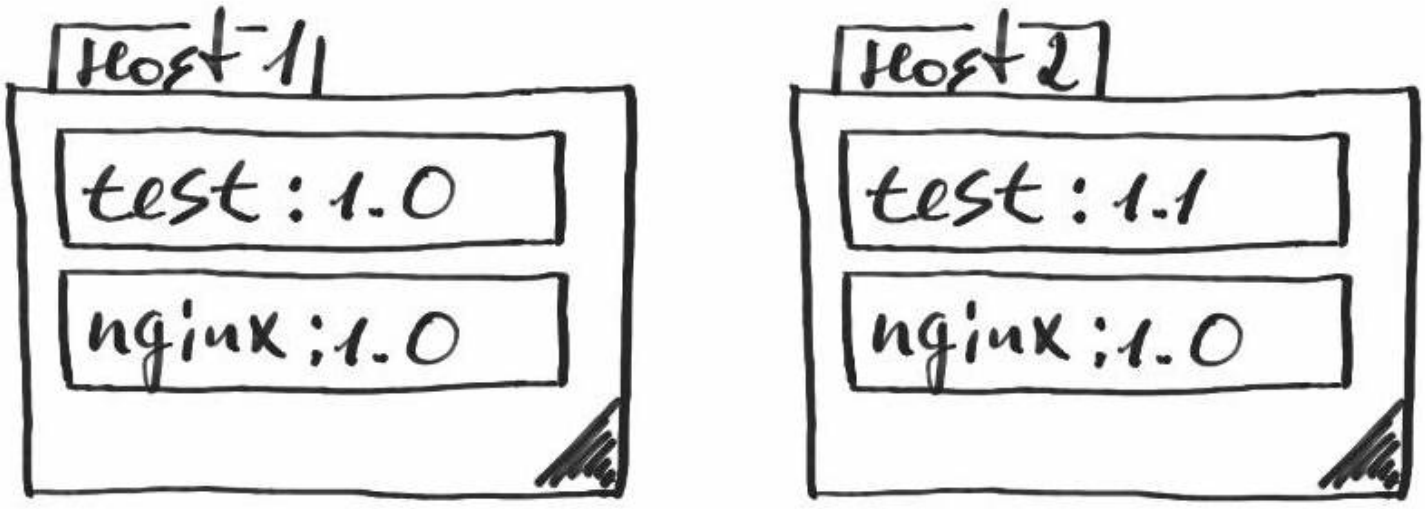
CONTAINER LIFECYCLE - OVERVIEW

CONTAINER LIFECYCLE - OVERVIEW

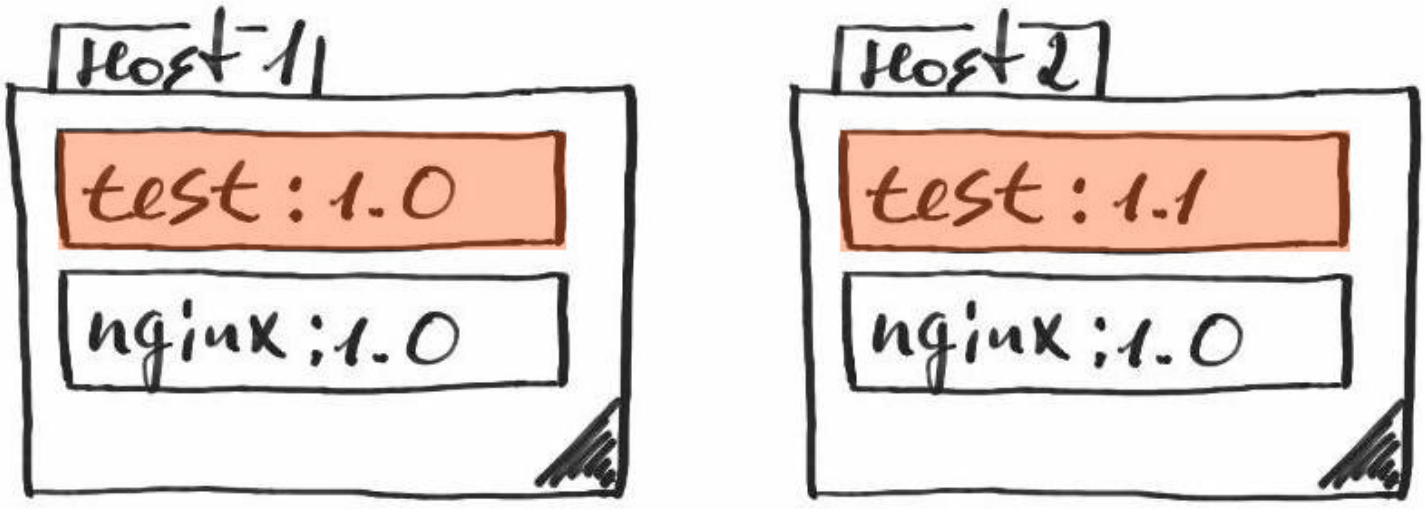


CONTAINER LIFECYCLE - IMAGE VERSIONS

CONTAINER LIFECYCLE - IMAGE VERSIONS

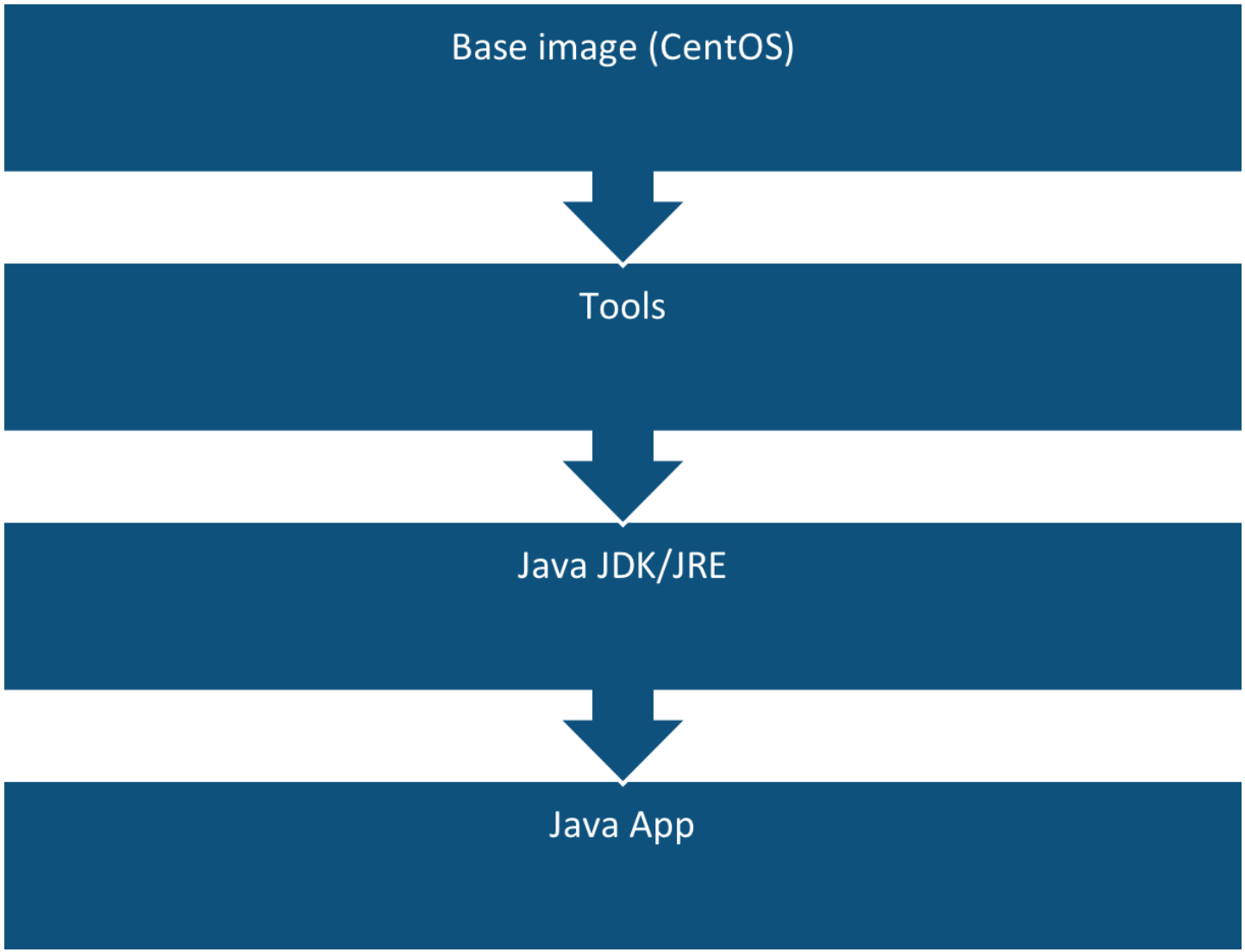


CONTAINER LIFECYCLE - IMAGE VERSIONS

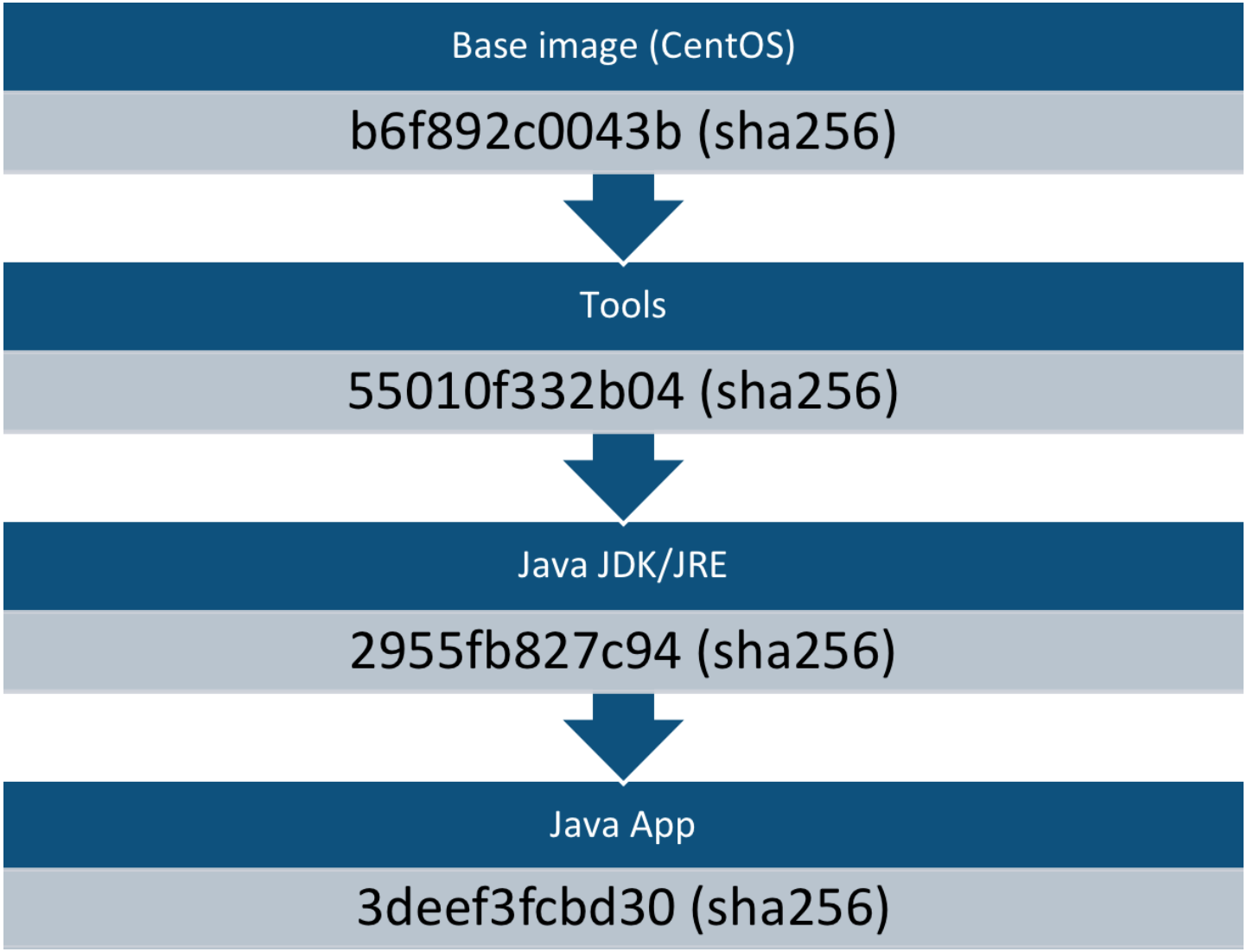


LAYERED FILE SYSTEM

LAYERED FILE SYSTEM



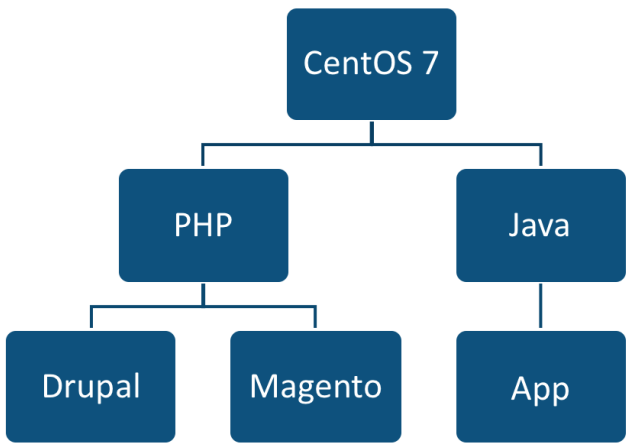
LAYERED FILE SYSTEM



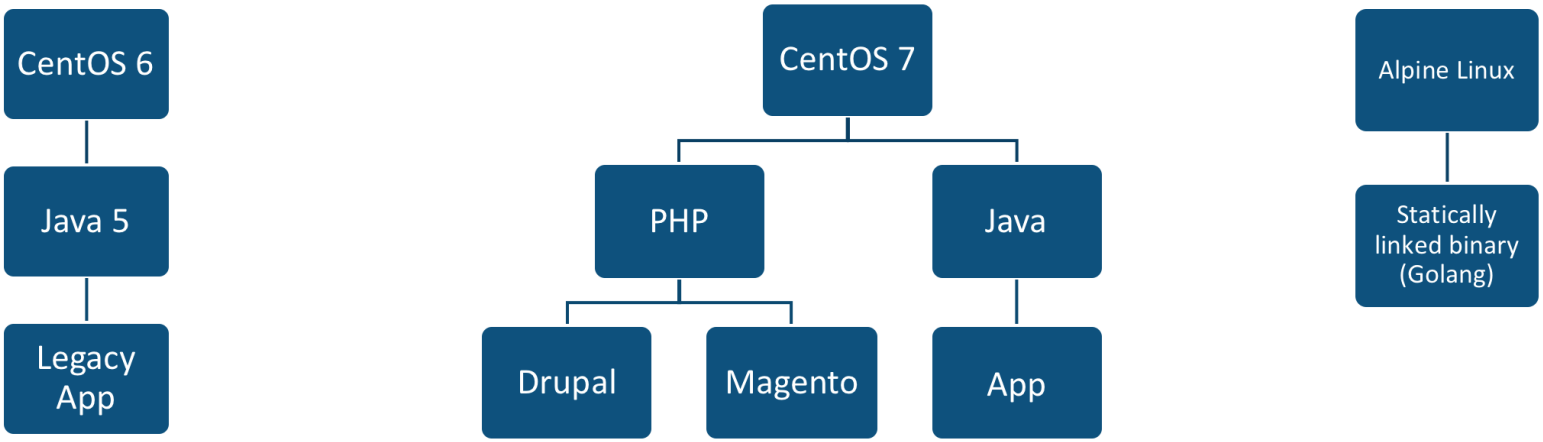
LAYERED FILE SYSTEM - IMAGE DEPENDENCIES

LAYERED FILE SYSTEM - IMAGE DEPENDENCIES

LAYERED FILE SYSTEM - IMAGE DEPENDENCIES



LAYERED FILE SYSTEM - IMAGE DEPENDENCIES



LAYERED FILE SYSTEM - BACKENDS

LAYERED FILE SYSTEM - BACKENDS

- aufs (legacy)
- overlayfs (legacy)
- devicemapper
- overlayfs2
- btrfs

PERSISTENT STORAGE (VOLUMES)

PERSISTENT STORAGE (VOLUMES)

- Containers start without state

PERSISTENT STORAGE (VOLUMES)

- Containers start without state
- State is provided via mounts

PERSISTENT STORAGE (VOLUMES)

- Containers start without state
- State is provided via mounts
  - Binds (local FS)

PERSISTENT STORAGE (VOLUMES)

- Containers start without state
- State is provided via mounts
  - Binds (local FS)
  - Docker volumes

PERSISTENT STORAGE (VOLUMES)

- Containers start without state
- State is provided via mounts
  - Binds (local FS)
  - Docker volumes
  - Docker Storage plugins

SEPARATION AND RESOURCE LIMITING

SEPARATION AND RESOURCE LIMITING

- cgroups

SEPARATION AND RESOURCE LIMITING

- cgroups
  - Linux kernel feature

SEPARATION AND RESOURCE LIMITING

- cgroups
  - Linux kernel feature
  - Limiting: CPU, memory, disk I/O, network

SEPARATION AND RESOURCE LIMITING

- cgroups
  - Linux kernel feature
  - Limiting: CPU, memory, disk I/O, network
- namespaces

SEPARATION AND RESOURCE LIMITING

- cgroups
  - Linux kernel feature
  - Limiting: CPU, memory, disk I/O, network
- namespaces
  - Linux kernel feature

SEPARATION AND RESOURCE LIMITING

- cgroups
  - Linux kernel feature
  - Limiting: CPU, memory, disk I/O, network
- namespaces
  - Linux kernel feature
  - process isolation
    Each container "thinks" it's the only one in the system
    (*ps uax*)

SUMMARY

SUMMARY

- Containers are pretty slick and useful for packaging applications

SUMMARY

- Containers are pretty slick and useful for packaging applications
- Docker is the standard de facto (for now) for running containers

SUMMARY

- Containers are pretty slick and useful for packaging applications
- Docker is the standard de facto (for now) for running containers
- Docker alone is not enough to run containers big time

NEXT UP

Kubernetes Fundamentals