

# CONTAINER / DOCKER FÜR EINSTEIGER\*INNEN

*Dipl. Ing. (FH) Eugen Richter*

# THEMEN

# THEMEN

- Container vs. VM vs. Bare Metall

# THEMEN

- Container vs. VM vs. Bare Metall
- Container / Image

# THEMEN

- Container vs. VM vs. Bare Metall
- Container / Image
- Orchestrierung

# THEMEN

- Container vs. VM vs. Bare Metall
- Container / Image
- Orchestrierung
- Container Hubs / Registries

# THEMEN

- Container vs. VM vs. Bare Metall
- Container / Image
- Orchestrierung
- Container Hubs / Registries
- Docker Geschichte

# THEMEN

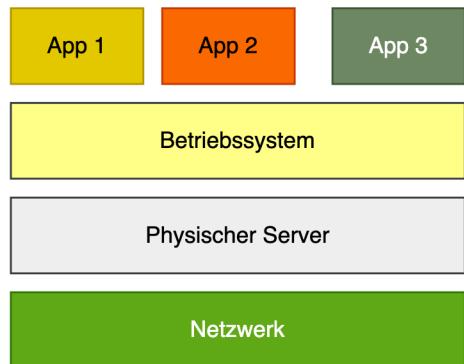
- Container vs. VM vs. Bare Metall
- Container / Image
- Orchestrierung
- Container Hubs / Registries
- Docker Geschichte
- Weiterführende Themen

# THEMEN

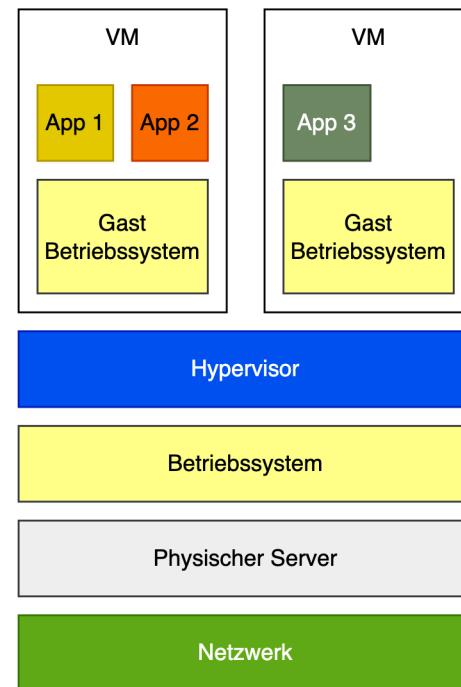
- Container vs. VM vs. Bare Metall
- Container / Image
- Orchestrierung
- Container Hubs / Registries
- Docker Geschichte
- Weiterführende Themen
- Workshop

# CONTAINER VS. VM VS. BARE METALL

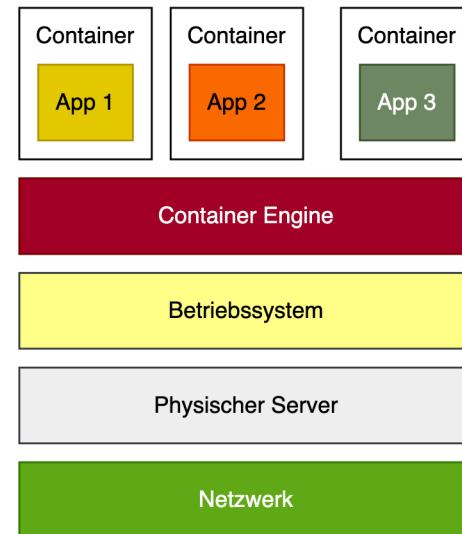
**Bare Metal**



**Virtualisierung**



**Containerisierung**



# BARE METALL

- Pro
  - Alle Ressourcen der Hardware nutzbar
  - Wiederverwendung von Abhängigkeiten
  - Schnelle App Startzeiten
- Kontra
  - Auslastung der Hardware mit einer Applikation
  - HW-Größe basierend am max. Auslastung
  - App Abhängigkeiten
  - Skalierung (neu HW + Installation)

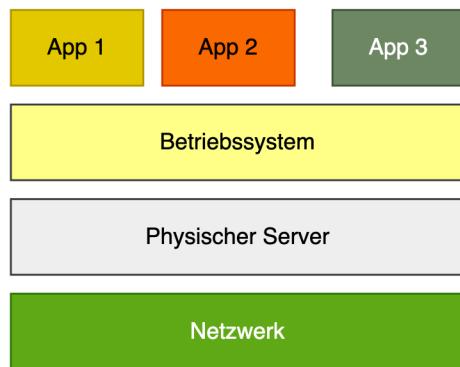
# VIRTUALISIERUNG

- Pro
  - Abhängigkeiten können in der VM isoliert werden
  - Unterschiedliche Betriebssysteme pro VM möglich
  - Vorlagen / Images möglich (für Skalierung)
- Kontra
  - Höher Ressourcenverbrauch pro VM (eigenständiges Betriebssystem)
  - Lange Startzeiten (Hochfahren der VM)

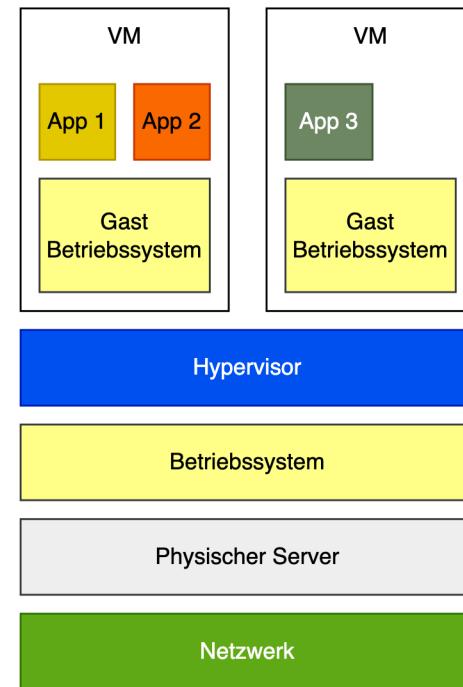
# CONTAINERISIERUNG

- Pro
  - Sehr schnelle Startzeit
  - Kein Verbrauch “unnötiger” Ressourcen
  - Isolierung der Abhängigkeiten pro Container
  - Skalierung
- Kontra
  - Nur Betriebssystem des Host
  - Nicht alle “Features” der HW können direkt genutzt werden

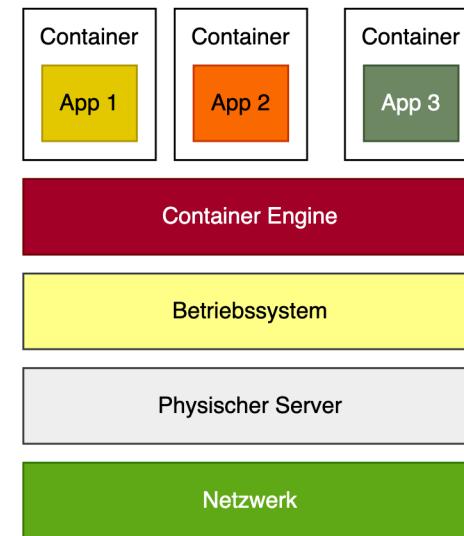
## Bare Metal



## Virtualisierung



## Containerisierung



# CONTAINER / IMAGES

# IMAGE

# IMAGE

- Bauplan für einen Container
- Beschreibt die Software und ihre Abhängigkeiten
- Ist ein Schichten-Dateisystem
- Jede Schicht entspricht einer Änderung an der vorherigen Schicht

# CONTAINER



# CONTAINER

- Instanz, die aus einem Image erstellt wurde
- Letzter Layer im Dateisystem ist beschreibbar
- “Die” laufende Applikation
- Kann beim Start parametrisiert werden

# ORCHESTRIERUNG

# ORCHESTRIERUNG

## PROBLEM

# ORCHESTRIERUNG

## PROBLEM

- Abhängigkeiten zwischen Containern

# ORCHESTRIERUNG

## PROBLEM

- Abhängigkeiten zwischen Containern
- Startreihenfolge

# ORCHESTRIERUNG

## PROBLEM

- Abhängigkeiten zwischen Containern
- Startreihenfolge

## LÖSUNG

# ORCHESTRIERUNG

## PROBLEM

- Abhängigkeiten zwischen Containern
- Startreihenfolge

## LÖSUNG

- Orchestrierung

# ORCHESTRIERUNG

## PROBLEM

- Abhängigkeiten zwischen Containern
- Startreihenfolge

## LÖSUNG

- Orchestrierung
-  Docker Compose

# ORCHESTRIERUNG

## PROBLEM

- Abhängigkeiten zwischen Containern
- Startreihenfolge

## LÖSUNG

- Orchestrierung
-  Docker Compose
-  Kubernetes

# HUBS / REGISTRIES

# HUBS / REGISTRIES

Speicherorte für veröffentlichte Images mit Versionen und meistens einer Beschreibung, wie das Image zu nutzen ist.

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

- Docker Hub - [hub.docker.com](https://hub.docker.com)

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

- Docker Hub - [hub.docker.com](https://hub.docker.com)
- Quay.io - [quay.io](https://quay.io)

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

- Docker Hub - hub.docker.com
- Quay.io - quay.io
- GitHub Container Registry

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

- Docker Hub - hub.docker.com
- Quay.io - quay.io
- GitHub Container Registry
- Azure Container Registry

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

- Docker Hub - hub.docker.com
- Quay.io - quay.io
- GitHub Container Registry
- Azure Container Registry
- Google Container Registry

# BEKANNTSTE ANBIETER VON CONTAINER REPOSITORIES

- Docker Hub - hub.docker.com
- Quay.io - quay.io
- GitHub Container Registry
- Azure Container Registry
- Google Container Registry
- Amazon Elastic Container Registry

# DOCKER - GESCHICHTE

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud
- 2013: Ankündigung von Docker (mit RedHat)

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud
- 2013: Ankündigung von Docker (mit RedHat)
- 2014: Partnerschaft mit Microsoft

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud
- 2013: Ankündigung von Docker (mit RedHat)
- 2014: Partnerschaft mit Microsoft
- 2015: OCI Projekt

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud
- 2013: Ankündigung von Docker (mit RedHat)
- 2014: Partnerschaft mit Microsoft
- 2015: OCI Projekt
- 2015: Kubernetes 1.0

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud
- 2013: Ankündigung von Docker (mit RedHat)
- 2014: Partnerschaft mit Microsoft
- 2015: OCI Projekt
- 2015: Kubernetes 1.0
- 2018: GKS, AKS, EKS (Managed Kubernetes)

# DOCKER - GESCHICHTE

- 2008: cgroup kommen in Linux an
- 2008: Gründung als dotCloud
- 2013: Ankündigung von Docker (mit RedHat)
- 2014: Partnerschaft mit Microsoft
- 2015: OCI Projekt
- 2015: Kubernetes 1.0
- 2018: GKS, AKS, EKS (Managed Kubernetes)
- 2021: Änderung der Lizenzierung

# WEITERFÜHRENDE THEMEN

# WEITERFÜHRENDE THEMEN

- Kubernetes (Orchestrierung)

# WEITERFÜHRENDE THEMEN

- Kubernetes (Orchestrierung)
- OCI - Open Container Initiative

# WEITERFÜHRENDE THEMEN

- Kubernetes (Orchestrierung)
- OCI - Open Container Initiative
- CNCF - Cloud Native Computing Foundation

# WEITERFÜHRENDE THEMEN

- Kubernetes (Orchestrierung)
- OCI - Open Container Initiative
- CNCF - Cloud Native Computing Foundation
- Alpine Linux

# WEITERFÜHRENDE THEMEN

- Kubernetes (Orchestrierung)
- OCI - Open Container Initiative
- CNCF - Cloud Native Computing Foundation
- Alpine Linux
- UNRAID (NAS auf Container Basis)

# WEITERFÜHRENDE THEMEN

- Kubernetes (Orchestrierung)
- OCI - Open Container Initiative
- CNCF - Cloud Native Computing Foundation
- Alpine Linux
- UNRAID (NAS auf Container Basis)
- WebAssembly

# WORKSHOP



# WORKSHOP

- Installation

# WORKSHOP

- Installation
- Starten von vorhandenen Images

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)
- Docker Compose (Orchestrierung)

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)
- Docker Compose (Orchestrierung)
- Nutzung als Entwickler (Beispiel Applikation)

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)
- Docker Compose (Orchestrierung)
- Nutzung als Entwickler (Beispiel Applikation)
- Eigenen Code als Image verpacken

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)
- Docker Compose (Orchestrierung)
- Nutzung als Entwickler (Beispiel Applikation)
- Eigenen Code als Image verpacken
- Multi Step Builds

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)
- Docker Compose (Orchestrierung)
- Nutzung als Entwickler (Beispiel Applikation)
- Eigenen Code als Image verpacken
- Multi Step Builds
- *Veröffentlichung der Images*

# WORKSHOP

- Installation
- Starten von vorhandenen Images
- Nutzung als Entwickler (Beispiel Applikation)
- Docker Compose (Orchestrierung)
- Nutzung als Entwickler (Beispiel Applikation)
- Eigenen Code als Image verpacken
- Multi Step Builds
- *Veröffentlichung der Images*
- *Dev / Test Container*

