



Docker Docker Docker

Dr. Olaf Flebbe

science + computing ag

IT-Dienstleistungen und Software für anspruchsvolle Rechnernetze

Tübingen | München | Berlin | Düsseldorf

Docker Erklärungsfilm

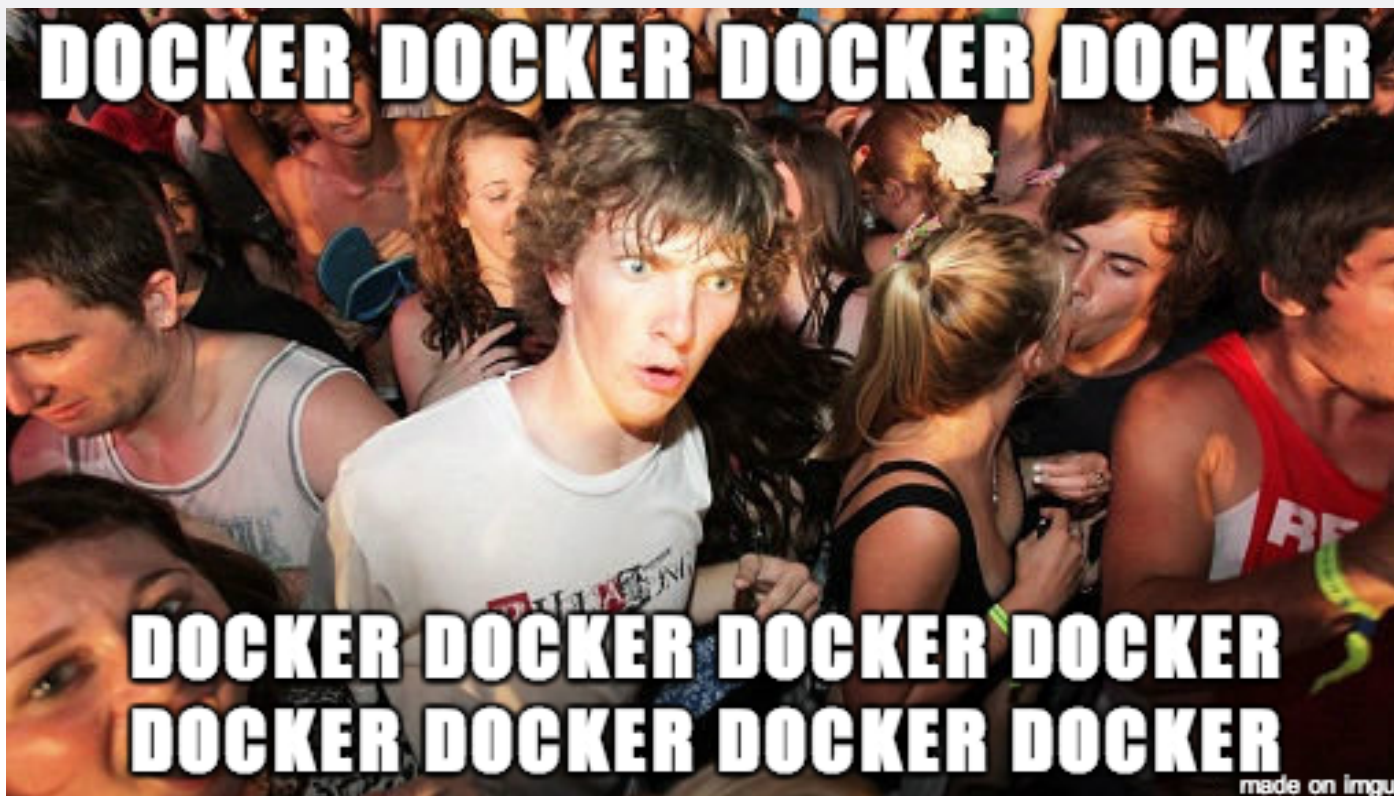
- Linux Container
 - Prozessisolation, Kein Virtualisierungsoverhead.
 - Höhere Packungsdichte auf dem Server
- Standardisierte Laufzeitumgebung
- Deployment vereinfachen

- Zutaten: Namespaces
- Dateisystem (chroot)
- Prozessbaum (cgroups)
- Shared Memory, Semaphore, ... (cgroups)
- Netzwerk
- Capabilities einschränken
- (Userids)



- Docker (Docker Engine): Ist der Docker Daemon und die CLI (Kommandozeile)
- Docker Toolbox: Fertige Install Umgebung
- Docker Machine: Management der VM bei MacOSX / Windows

- Image: Eine Blaupause für Container, Dateisystem + Metainfo
- Container: Ein tatsächlicher Zustand eines Systems (Zustand running oder stopped)
- Registry: Depot für Images
- Dockerhub: Öffentliche Registry
- Official Image: Image das von Docker Inc. kuratiert ist



- <http://jamespacileo.github.io/Slides-Dockerize-That-Django-App/img/docker-meme.png>

Installation

- Windows/Mac OS X: Download der Docker Toolbox und Installation <https://www.docker.com/products/docker-toolbox>

Docker Toolbox



The Docker Toolbox is an installer to quickly and easily install and setup a Docker environment on your computer.



Download



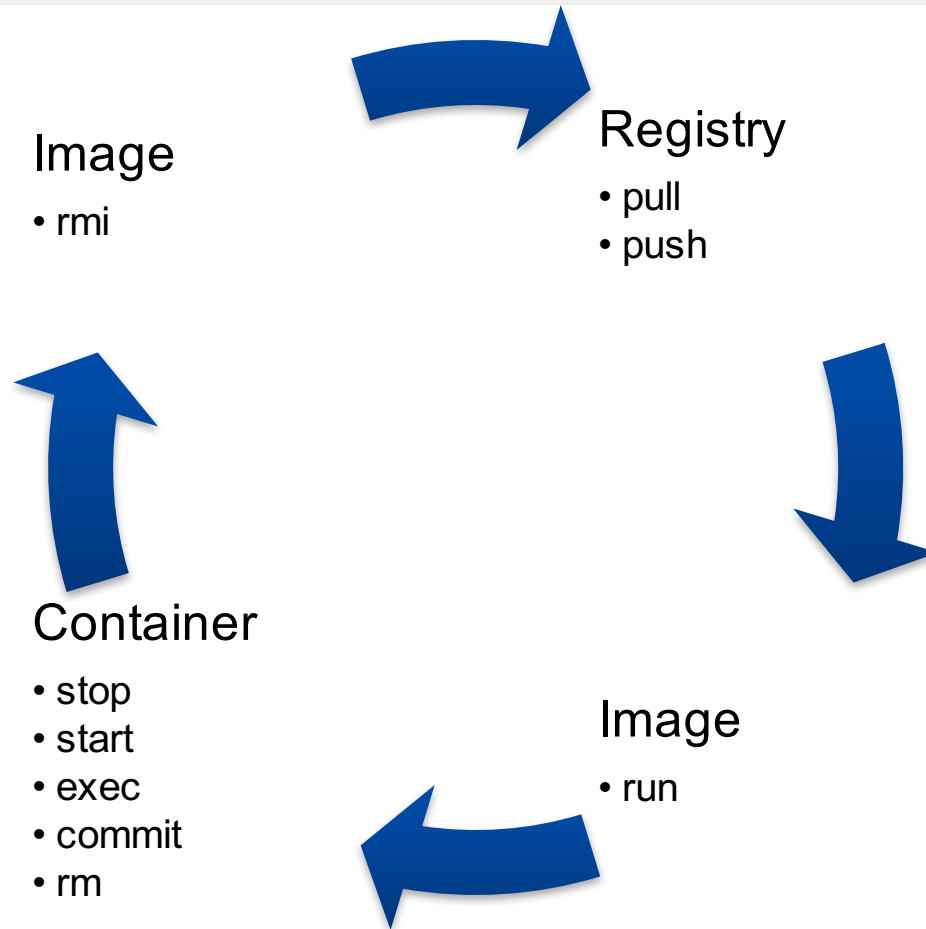
Download

- Docker-machine: Verwaltung der virtuellen Maschine
 - docker-machine ls
 - docker-machine ssh
 - /User gemountet
- VirtualBox
- Boot2docker: Linux VM Image, das docker enthält
- Docker-engine: „Docker“ als CLI auf Host und und boot2Docker und als Server in Boot2docker
- (Kitematic)

Installation auf Linux

- Linux: (Ver)altet(e) Versionen über die Distributionen:
- Debian8 / ubuntu-16.04 : apt-get install docker.io
- openSuSE 42.1: zypper install docker
- CentOS 7: yum install docker
- Neuere Versionen über Repositories von Docker siehe www.docker.com

Docker Ablauf



Beispiel

- Image auf Dockerhub suchen
`docker search hello-world`
- Image von Dockerhub herunterladen
`docker pull hello-world`
- Image starten
`docker run hello-world`
- Schnell !

Beispiele aus Dockerhub library

- Minimale Betriebssysteme
- Usecase Inhalt einer Datei in jeder Distro anschauen
- ```
for OS in debian:8 centos:7 opensuse:42.1 ; do
 echo "+++ $OS +++"
 docker run --rm $OS cat /etc/os-release
done
```
- (`--rm` entfernt den Container sofort)

# Beispiel tomcat aus Dockerhub library

Bezeichnung der Images

[Username]/name:tag

Hier official (keiner)

OFFICIAL REPOSITORY

tomcat 

Last pushed: 4 days ago

Repo Info

Tags

Short Description

Apache Tomcat is an open source implementation of the Java Servlet and JavaServer Pages technologies

Full Description

## Supported tags and respective Dockerfile links

- [6.0.45-jre7](#), [6.0-jre7](#), [6-jre7](#), [6.0.45](#), [6.0](#), [6](#) ([6/jre7/Dockerfile](#))
- [6.0.45-jre8](#), [6.0-jre8](#), [6-jre8](#) ([6/jre8/Dockerfile](#))
- [7.0.69-jre7](#), [7.0-jre7](#), [7-jre7](#), [7.0.69](#), [7.0](#), [7](#) ([7/jre7/Dockerfile](#))
- [7.0.69-jre8](#), [7.0-jre8](#), [7-jre8](#) ([7/jre8/Dockerfile](#))
- [8.0.35-jre7](#), [8.0-jre7](#), [8-jre7](#), [8.0.35](#), [8.0](#), [8](#), [latest](#) ([8.0/jre7/Dockerfile](#))
- [8.0.35-jre8](#), [8.0-jre8](#), [8-jre8](#) ([8.0/jre8/Dockerfile](#))
- [8.5.2-jre8](#), [8.5-jre8](#), [8.5.2](#), [8.5](#) ([8.5/jre8/Dockerfile](#))
- [9.0.0.M6-jre8](#), [9.0.0-jre8](#), [9.0-jre8](#), [9-jre8](#), [9.0.0.M6](#), [9.0.0](#), [9.0](#), [9](#) ([9.0/jre8/Dockerfile](#))

ImageLayers.io 0 B / 25 Layers

# Tomcat aufrufen

- `docker run -it -p 8080:8080 tomcat:6-jre8`
- `docker run -it -p 8080:8080 tomcat:7-jre8`
- `docker run -it -p 8080:8080 tomcat:8-jre8`
  
- Browser auf localhost:8080 richten.
- (-p 8080:8080 macht einen Port Forward von dem Server im Container an den Host)

# Zusätzliche Zutaten

- Schichtenbasiertes Storage: (graphdriver) aufs, overlays, devicemapper, btrfs
- Images bauen auf einander auf. Schichten haben einen eindeutigen Namen (hash)
- Netzwerk Management. NAT default, Bridge
- Verlinken von Containern
- Volumes: Mountpunkte, die nicht dem Schichten-Storage unterliegen
- Bind Mounts
- Pseudo Terminals
  
- Docker ist in der Programmiersprache „go“ [golang.org](http://golang.org) programmiert.

# Was für Interessante Container gibt es?

- Sprache „Go“

- Swift

```
docker pull swiftdocker/swift
docker run --privileged -i -t --name swiftfun
swiftdocker/swift:latest /bin/bash
```

```
echo 'print("Hello, World")' > hello.swift
swiftc hello.swift
```

- „cross“ compile
- Reproduzierbare builds



# Docker Container selber erstellen

- Rezept zum bauen eines Containers
  - Dockerfile
- Docker build <name> <dir>

# Eigene Docker registry

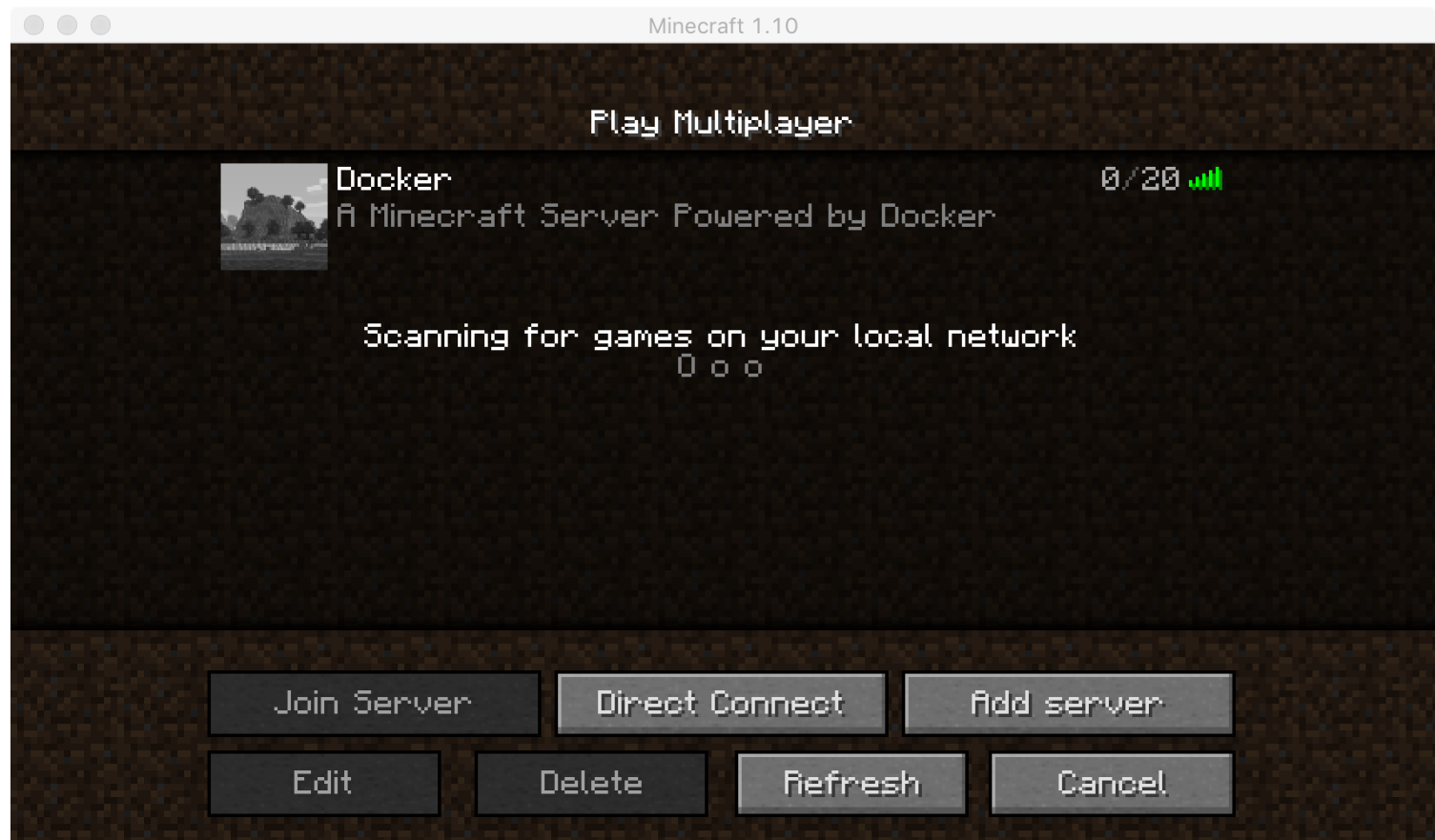
- Registry Starten
  - `docker run -d -p 6000:5000 --name registry registry:2`
- Tag
  - `docker tag hello-world localhost:6000/hello-world`
- Push
  - `docker push localhost:6000/hello-world`
- Pull
  - `docker pull localhost:6000`
- Fail: Löschen von Images, Suchen



# Minecraft

- `docker run -d -it -e EULA=TRUE -p 25565:25565 --name mc itzg/minecraft-server`

- Demo



# StarMade Server

StarMade is a voxel-based 3D sandbox space shooter. Play in your own universe or on a multiplayer server to create, discover, destroy, customize, and design your own worlds.



# StarMade Server (ähnlich Minecraft)

- Benötigt eine Console /PTY (für Chat und Admin Kommandos)
- Aus einer bestehenden Installation zusammengeklaut
- Dockerfile:

```
FROM debian:8
```

```
RUN apt-get update && apt-get install -y
```

```
openjdk-7-jre-headless unzip curl
```

```
COPY StarMade.tar /tmp
```

```
RUN tar xf /tmp/StarMade.tar
```

```
RUN rm /tmp/StarMade.tar
```

```
CMD /StarMade/StarMade-dedicated-
server-linux.sh
```

- `docker build -t starmade:1 .`
- `docker run -name starmadecontainer -it -p4242:4242 starmade`
- **Strg-p Strg-q**
- `docker start/stop/attach starmadecontainer`
- Backup
- `docker cp starmadecontainer:/StarMade starmade.tar`
- **Einfacher als VM!**

# Docker recompile

- git clone <https://github.com/docker/docker.git>
- cd docker
- make
  
- Installiert Entwicklungsumgebung in container
- Compiliert in container

# Nicht genug

- Docker auf anderen Architekturen: PPC64LE (POWER8 64), AARCH64 (ARM 64 bit)
- Docker auf Windows
- Grenzen: Bind Mounts, Filesysteme, Userid Mapping

- Docker Compose
- Demo einer kleinen Compile Umgebung mit Jenkins, Nexus
- Docker-compose up -d

```
version: '2'
services:
 jenkins:
 build: jenkins
 ports:
 - "9080:8080"
 links:
 - "nexus"
 volumes:
 - jenkins_data:/var/jenkins_home
 - /var/run/docker.sock:/var/run/docker.sock
 depends_on:
 - nexus
```

```
nexus:
 image: "sonatype/nexus"
 volumes:
 - nexus_data:/sonatype-work
 ports:
 - "8081:8081"
```

```
volumes:
 jenkins_data:
 driver: local
 nexus_data:
 driver: local
```



# Docker auf Raspi

```
sudo apt-get install -y apt-transport-
https
wget -q
https://packagecloud.io/gpg.key -O - |
sudo apt-key add -
echo 'deb
https://packagecloud.io/Hypriot/Schatz
kiste/debian/ wheezy main' | sudo tee
/etc/apt/sources.list.d/hypriot.list
sudo apt-get update
sudo apt-get install -y docker-hypriot
sudo systemctl enable docker
```



Vielen Dank für Ihre Aufmerksamkeit.

**Vortrag von:**

science + computing ag  
[www.science-computing.de](http://www.science-computing.de)

Telefon: 07071 9457-0

E-Mail: [info@science-computing.de](mailto:info@science-computing.de)

# Bro -> elasticsearch -> Kibana

- `docker run --name elasti elasticsearch`
- `docker run --rm --link=elasti:elasticsearch -p 5601:5601 kibana`
  
- Hardwarezugriff auf Netzwerkkarte
- `Docker run --rm --it`
  
- Workaround mit VMware Fusion
- `/Applications/VMware\ Fusion.app/Contents/Library/vmnet-sniffer -w /tmp/b vmnet8`
  
- `docker run --it`