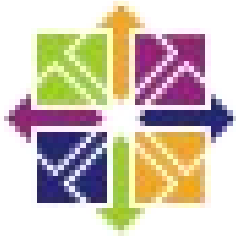
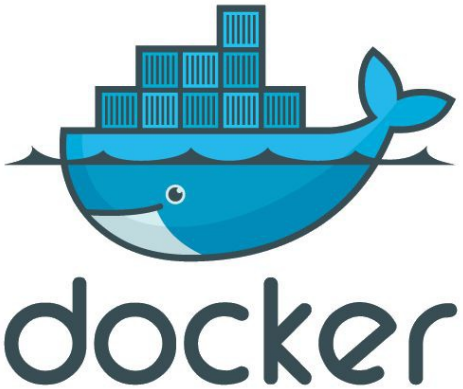





inovex

# docker.io @ CentOS 7

Secure And Portable
Containers Made Easy

Jürgen Brunk

Köln, 04.08.2014

# Agenda

inovex

1. Was ist Docker ?
2. Was sind Container ?
3. Warum Docker ?
4. Architektur
5. Praxis
6. Docker unter CentOS 7 installieren
7. Ein einfaches „Hello World“ Beispiel
8. Grundlegende Docker Befehle
9. Dockerfile

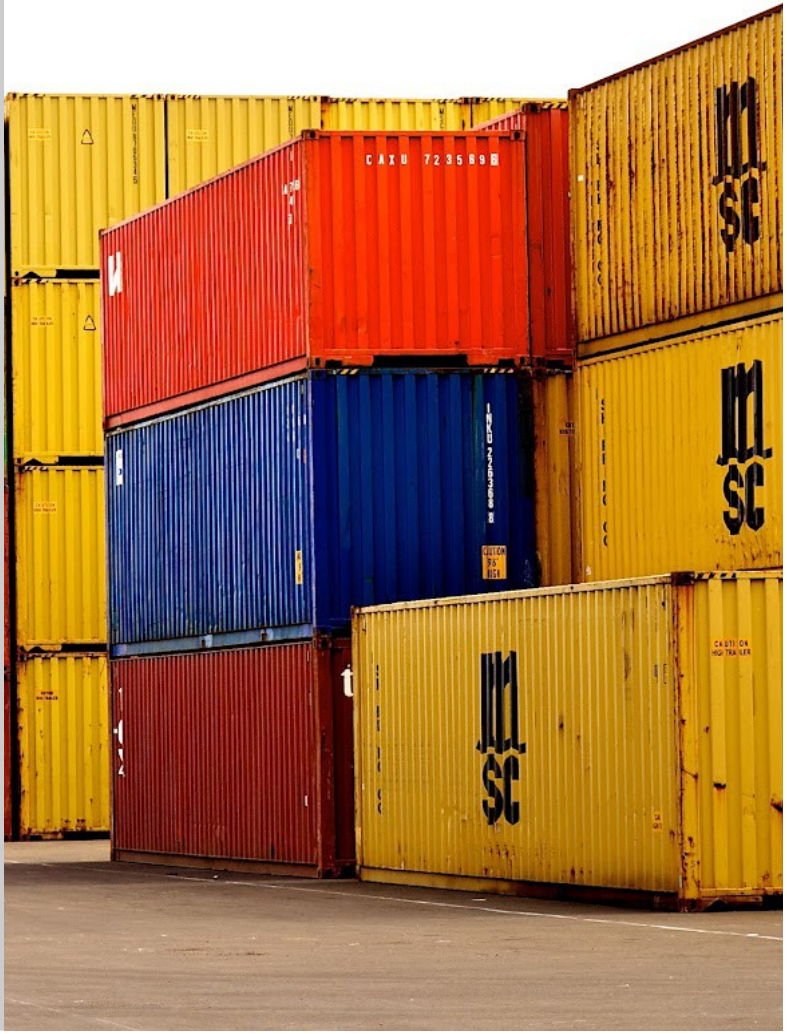
inovex

# Was ist Docker ?

# Was ist Docker ?

inovex

Das **Docker*** Framework erlaubt es (Web-) **Applikationen** in schlanke, autarke und portable Umgebungen, sog. **Container**, zu verpacken




*) engl. Hafenarbeiter

inovex

# Was sind Container ?

# Was sind Container ?

inovex

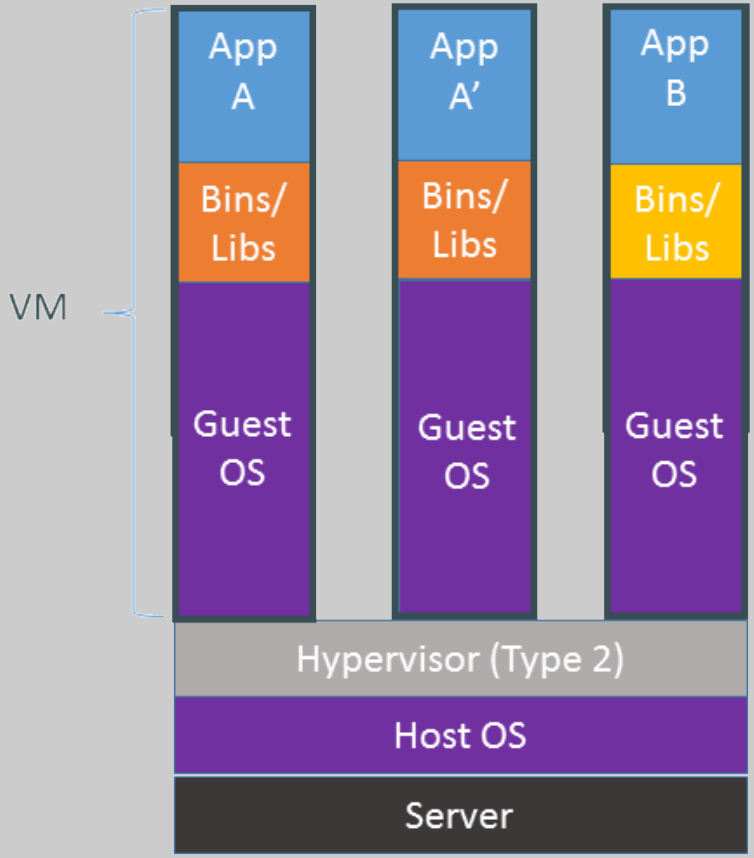
Operating system–level virtualization:
z.B. jails, openvz, lxc, ...

Abgeschottete Teilmenge des Hostsystems
(getrennter Process-, Netzwerk-, I/O-Raum)

Quasi „chroot on Steroids“

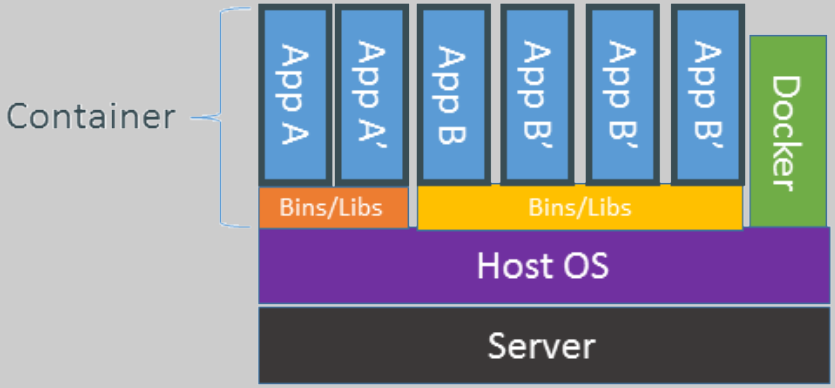
# Container vs. VMs

inovex




Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



inovex

# Warum Docker ?

# Warum Docker? (The Matrix from Hell)

inovex

| | Development VM | QA Server | Single Prod Server | Onsite Cluster | Public Cloud | Contributor's laptop | Customer Servers |
|---|---|---|---|---|---|---|---|
| Static website | ? | ? | ? | ? | ? | ? | ? |
| Web frontend | ? | ? | ? | ? | ? | ? | ? |
| Background workers | ? | ? | ? | ? | ? | ? | ? |
| User DB | ? | ? | ? | ? | ? | ? | ? |
| Analytics DB | ? | ? | ? | ? | ? | ? | ? |
| Queue | ? | ? | ? | ? | ? | ? | ? |

inovex

**Einmal gebaut – läuft überall !**

Saubere, sichere, portable Laufzeitumgebung für die Application

Kein Problem mit Dependencies, Paketen etc. während des Deployments

Jede Application ist ein isolierter Container mit ggf. unterschiedlichen SW-Versionen

# Vorteile für OPS

inovex

**Einmal konfiguriert – läuft überall !**

Keine Inkonsistenzen mehr zwischen Dev-, QA-, Stage-, Prod-Umgebung

Schnelleres Deployment (continuous deployment / continuous integration)

Schlanke Container – bessere Performance als VM's

# Warum es funktioniert (Trennung der Zuständigkeiten)

inovex

**Entwickler:**

Kümmert sich um das was **innerhalb** des Containers ist:

- sein Code / Daten
- seine Libs / Frameworks
- sein Package Manager

**Alle Linux Server sehen gleich aus**

**Admin:**

Kümmert sich um das was **ausserhalb** des Containers ist:

- Logging / Backup
- Remote Access
- Network Config

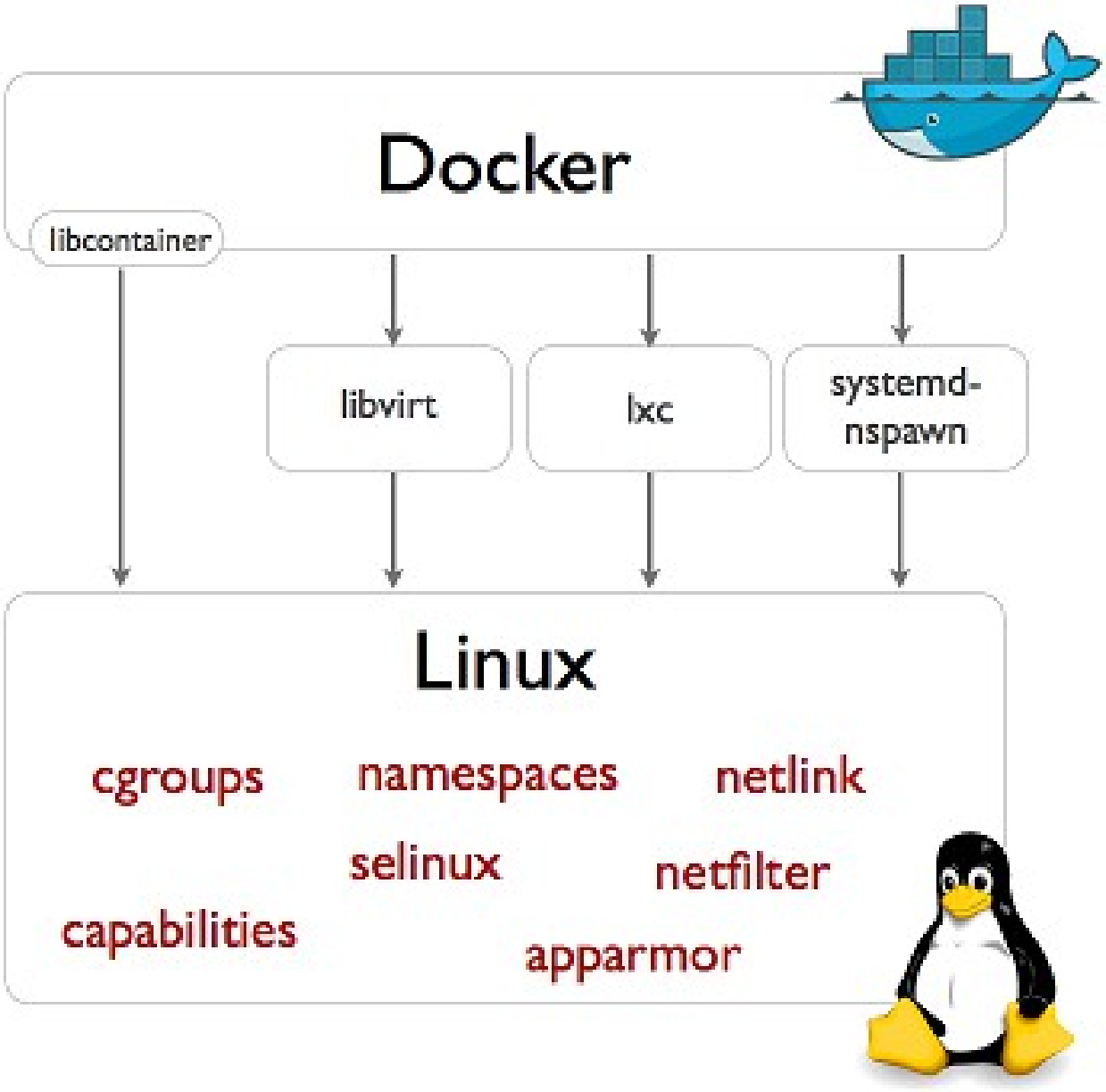
**Alle Container starten und stoppen gleich**

inovex

# Architektur

# Architektur

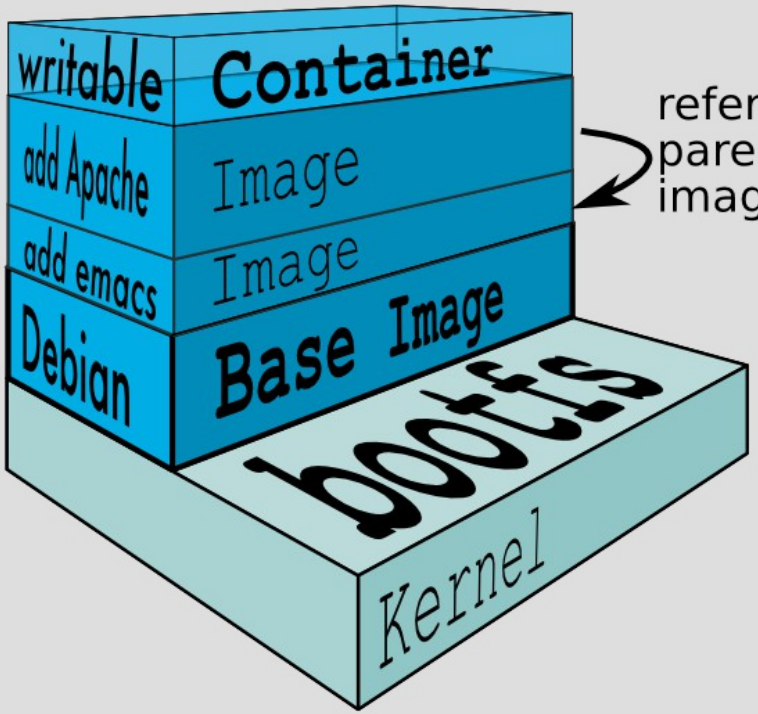
inovex




docker.io

inovex

# **Docker** basiert auf Linux Containern
# Minimaler Overhead (cpu/io/network)
# Verwendet **layered Filesystem***




*)
aufs
btrfs
devicemapper

inovex

Läuft auf jedem System das LXC unterstützt

Ubuntu, Debian, RHEL, CentOS, Fedora, Gentoo, Google Cloud, Rackspace Cloud, Amazon EC2, IBM Softlayer, Arch Linux, FrugalWare, Fedora, openSUSE, CRUX Linux, **CoreOS**, ...

Microsoft Windows*, Apple OSX*, Raspberry PI*

inovex

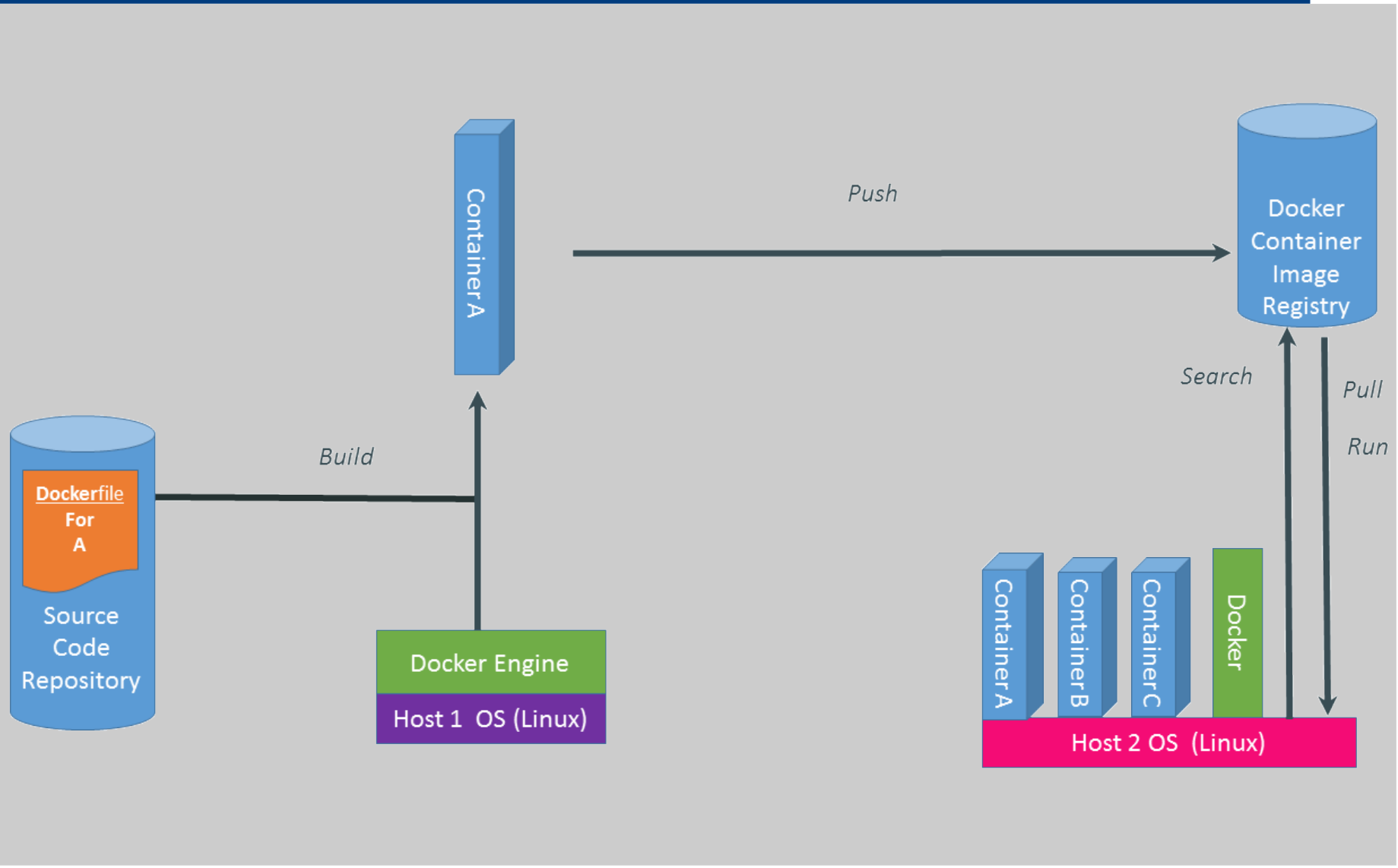
Ein Docker Container enthält alles nötige:

- Minimal Base OS (kein Kernel)
- Libraries / Frameworks
- Application Code + Data

Ein Container kann überall da laufen wo auch Docker installiert werden kann

# Docker Basics

inovex



inovex

# Fragen soweit ?

inovex

# Praxis

inovex

**Wir bauen uns Docker Container**

inovex

# Docker unter CentOS 7 installieren

# Docker unter CentOS 7 installieren

inovex

# EPEL Repo einbinden *)
# **rpm -Uvh http://dl.fedoraproject.org/pub/epel/beta/7/x86_64/epel-release-7-0.2.noarch.rpm**
**# yum clean all && yum makecache**

# Docker installieren
**# yum install docker-io**

# Docker Installation prüfen
$ **sudo docker info**
$ **sudo docker version**

# Docker Verzeichnis – hier liegt alles
$ **sudo ls -l /var/lib/docker/**

# Docker Usage anzeigen lassen
$ **sudo docker**

inovex

# Ein einfaches „Hello World“ Beispiel

# Ein einfaches „Hello World“ Beispiel

inovex

```
# ein fertiges Image aus dem Docker Index ziehen
$ sudo docker pull ubuntu

# alle lokalen Images auflisten
$ sudo docker images

# einen Container erzeugen, Applikation
# „/bin/echo“ laufen lassen und am Ende den
# Container wieder entfernen
$ sudo docker run --rm ubuntu \
/bin/echo „Hello World“
```

inovex

# Grundlegende Docker Befehle

# Grundlegende Docker Befehle

inovex

```
docker search <TERM>
docker start | stop | kill | restart <CID>
docker ps [-a|-s]
docker images
docker pull <IMAGE>[:TAG]
docker run [-i] <IMAGE> [<CMD>]
docker build <PATH> | <URL> | -
docker rm [-f] <CID>
docker rmi [-f] <IMAGE>
docker save <IMAGE>
docker load
```

inovex

# Dockerfile

# Dockerfile

inovex

```
# sshd
#
# VERSION               0.0.1

FROM ubuntu
MAINTAINER Thatcher R. Peskens "thatcher@dotcloud.com"

# make sure the package repository is up to date
RUN echo "deb http://archive.ubuntu.com/ubuntu precise main universe"
> /etc/apt/sources.list
RUN apt-get update

RUN apt-get install -y openssh-server
RUN mkdir /var/run/sshd
RUN echo 'root:screencast' |chpasswd

EXPOSE 22
CMD /usr/sbin/sshd -D
```

# Container aus dem Dockerfile bauen

inovex

```
# Dockerfile erzeugen (Inhalt siehe letzte Folie)
$ vi Dockerfile

# Docker Image bauen, temporäre Zwischenbuilds am Ende verwerfen
$ sudo docker build --rm -t img_sshd .

# lokale Docker Images auflisten
$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
img_sshd            latest              9b8cbe62ff21        2 minutes ago
313.6 MB

# neuen Container aus Image erzeugen und als Daemon starten
$ sudo docker run -d -P --name ct_sshd img_sshd
d25a3b457f1164abc0ab29c30581be3ac7b5594290ceece772bf0f4309c2
28f8
```

# SSH Connect in den Container

inovex

```
# Container auflisten
$ sudo docker ps --no-trunc=true
CONTAINER ID                                                       IMAGE
COMMAND                    CREATED          STATUS          PORTS
              NAMES
d25a3b457f1164abc0ab29c30581be3ac7b5594290ceece772bf0f4309c2
28f8   img_sshd:latest    /bin/sh -c '/usr/sbin/sshd -D'   4 minutes ago
Up 3 minutes       0.0.0.0:49153->22/tcp    ct_sshd

# Container → Host Port Mapping finden
$ sudo docker port ct_sshd 22
0.0.0.0:49153

# SSH Connect via local Port forwarding (passwd = „screencast“)
$ ssh -lroot -p49153 localhost

# SSH Connect via Container IP
$ sudo docker inspect ct_sshd | grep IPAddress
```

inovex

# Noch Fragen ?

inovex

# Quellennachweise und Links

inovex

**Quellennachweise**:

www.docker.io

**Images**:

www.docker.io
www.jundiai.com.br
ruhrnachrichten.de
gist.github.com/simota/9043141
slides.com/stevenborrelli/docker

inovex

**Links:**

Docker Website:
http://www.docker.io/

CoreOS:
http://coreos.com/

Lightweight Linux for Docker:
http://boot2docker.io/

Packer:
http://www.packer.io/

# Vielen Dank für Ihre Aufmerksamkeit

Kontakt

**Jürgen Brunk**
Systems Engineer

inovex GmbH
Office München
Valentin-Linhof Str. 2
D-81829 München

Mobil: 0173 3181 003
Mail: juergen.brunk@inovex.de