

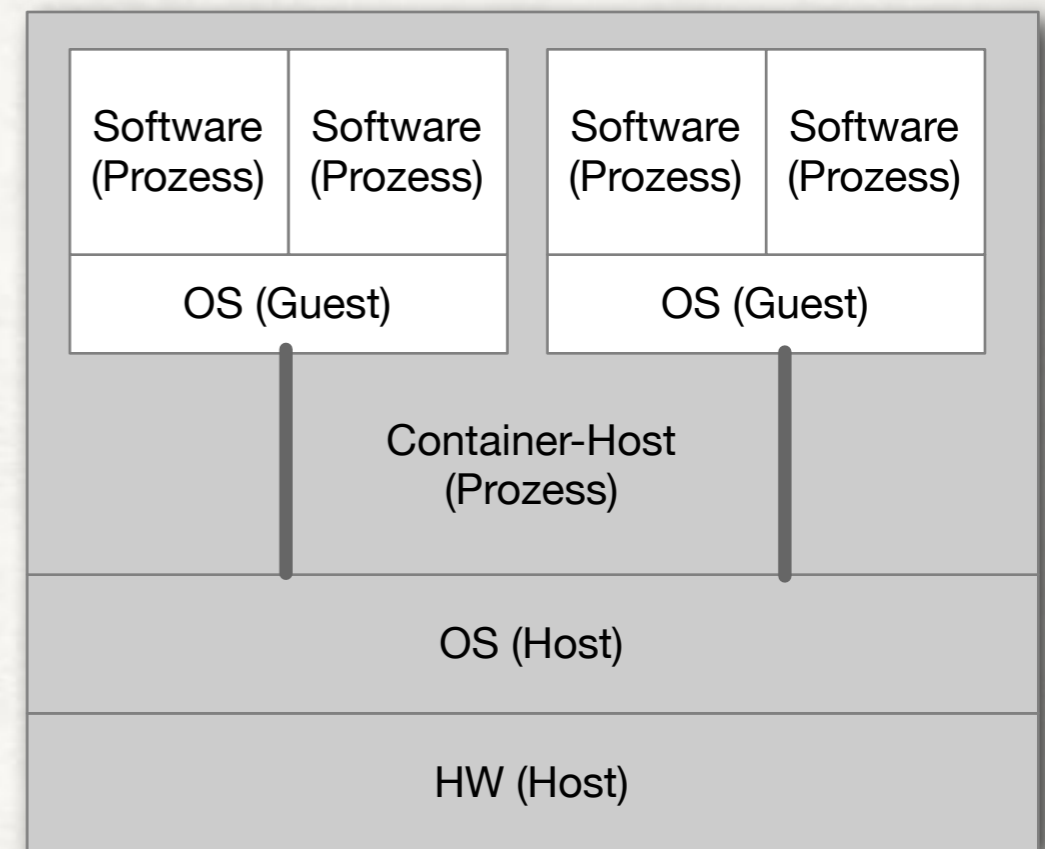
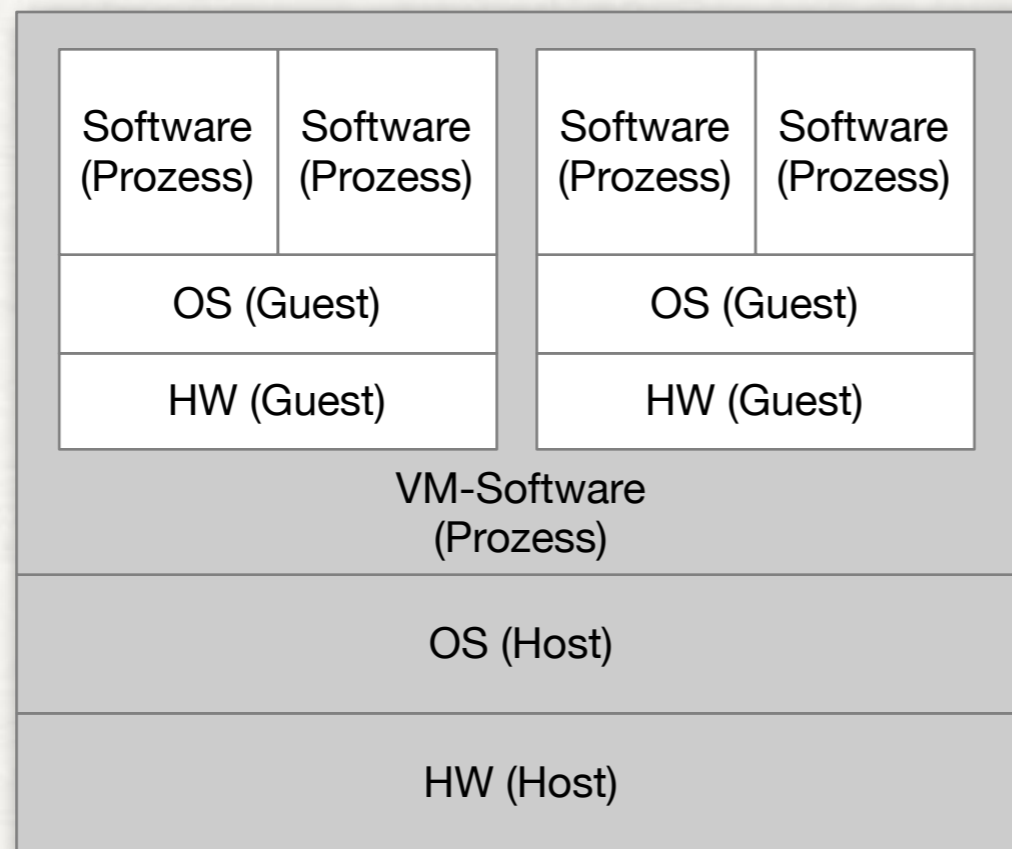
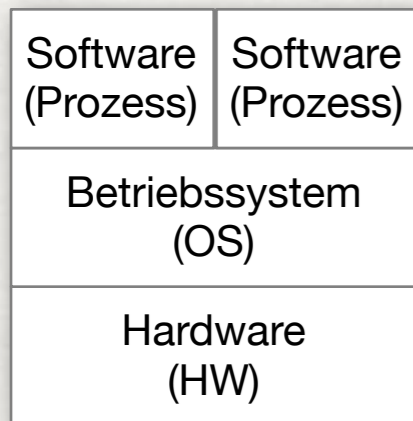
#BCKI18

DOCKER FÜR DEN HAUSGEBRAUCH

Stefan Winkler

@xpomul

WAS IST EIN CONTAINER? (FÜR MICH...)



Prozess

VM
virtuelle Maschine

Container
virtuelle Maschine
„light“

WOZU DOCKER?

(„NORMALGEBRAUCH“)

- Möglichkeit, Software inkl. ihrer Abhängigkeiten als ein Paket zu verteilen, zu installieren und zu betreiben
- Verwendung auf Servern in Rechenzentren (oder in der „Cloud“)
- leichtgewichtig = ressourcensparend (im Vergleich zu VM)
- standardisiert = es gibt Hosts (Google, IBM, Amazon, Oracle, ...)
- vorkonfiguriert = muss nur auf die Umgebung parameterisiert werden, daher einfach zu deployen und zu skalieren
- Ökosystem vorhanden = es ist möglich, bestehende Images einfach herunterzuladen, anzupassen und zusammenzusetzen

WOZU DOCKER?

(„HAUSGEBRAUCH“)

- Einfacher Weg, Software ohne aufwändige Installation zu verwenden
 - z.B. Datenbanken, um während der Entwicklung mal schnell was auszuprobieren (läuft meine Anwendung auch mit MSSQL, Oracle 12c, ...)
 - z.B. Server-Software (Tomcat, Wordpress, Ghost, ...), um mal schnell lokal was auszuprobieren
 - z.B. gehostete Internetdienste privat nutzen (PlantUML, Sagemath, ...)
 - z.B. LaTeX (spart eine große Installation, die Dateien überall ins System packt)

INSTALLATION

DOCKER CE

- Download unter <https://www.docker.com/community-edition>
 - für Windows (10)
 - Mac
 - Linux
 - PC
 - ARM (Raspberry Pi)
 - ...
- ... installieren, starten, fertig

ERSTE SCHRITTE

COMMAND LINE INTERFACE

- `docker run hello-world` ← Image-Name
 - `docker run ubuntu echo Hello World` ← Image-Name
 - `docker run -it ubuntu:xenial /bin/bash` ← Tag (=Version) - optional (default = latest)
← interactive + attach terminal
- Image = (geschichtetes) fixes Abbild
 - z.B. Ubuntu Base + mysql + java + tomcat
 - Container = gestartetes Image (inkl. Konfiguration)

IMAGES HERUNTERLADEN

DOCKERHUB

- Dockerhub: <https://hub.docker.com/explore/>

- Container suchen und Dokumentation lesen.

- Z.B. mysql:

- `docker run -e MYSQL_ROOT_PASSWORD=geheim -d mysql`

Image



Set environment variable
(vorgegeben durch
Ersteller des Images)

detach
= im Hintergrund ausführen
Gegenstück = -i — interactive

KONFIGURATION

DOCKER RUN - ENVIRONMENT + PORTS + FILESYSTEM

- Environment

- `-e SOME_ENV_VARIABLE=foobar`

- Portmapping

- `-p 12345:8080`
 - host port („außen“)
 - container (guest) port

- Filesystem-Mappings

- `-v /some/local/path:/path/in/image`

- `docker ps` — Listet laufende Container
- `docker stop` — Beendet einen laufenden container (spart CPU)
- `docker ps -a` — Listet alle Container
- `docker rm` — Löscht einen Container (spart HD)
- `docker images` — Listet heruntergeladene Images
- `docker rmi` — Löscht ein heruntergeladenes Image (spart HD)

DOCKER KANN NOCH MEHR

ADVANCED STUFF

- Docker ist ein wenig wie git:
 - docker commit, push, pull
- Docker-Images automatisch bauen mit Dockerfiles und docker build
 - Nimmt ein Basisimage und verändert es automatisch
- ...

NOCHMAL DIE WICHTIGSTEN LINKS

- Docker Download: <https://www.docker.com/community-edition>
- DockerHub: <https://hub.docker.com/explore/>
- CheatSheet: http://files.zereturnaround.com/pdf/zt_docker_cheat_sheet.pdf
 - — oder selbst googeln ;)
- Diese Folien: in Kürze auf <http://www.winklerweb.net>