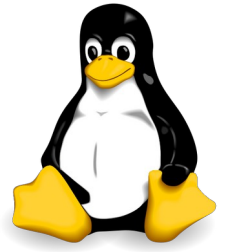
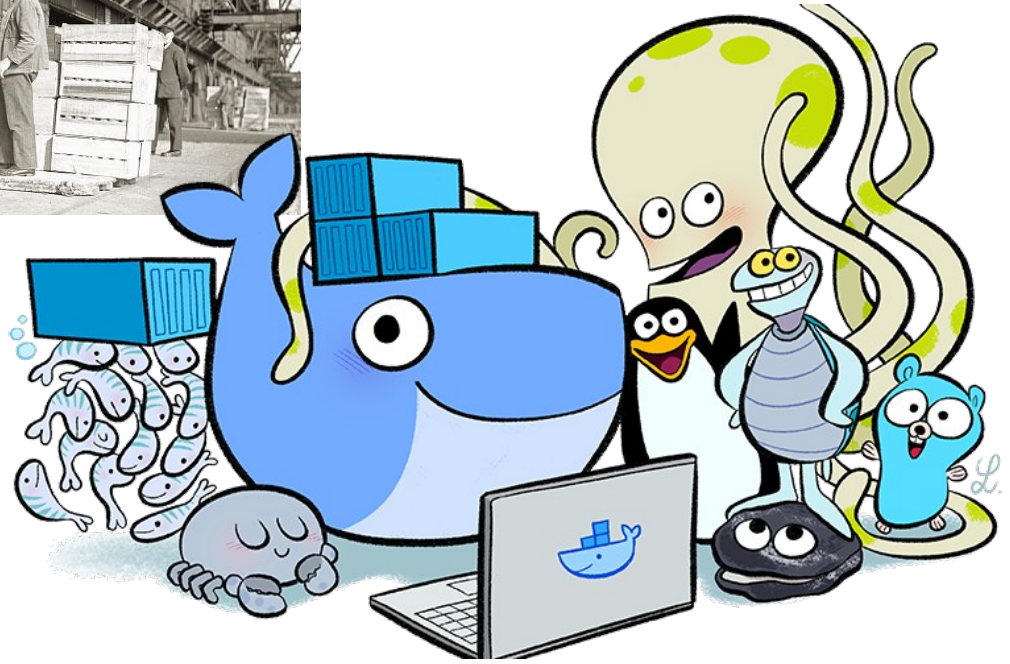
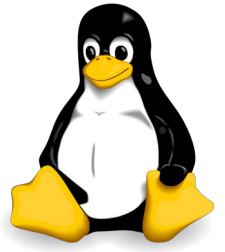
docker



LUG Albtal

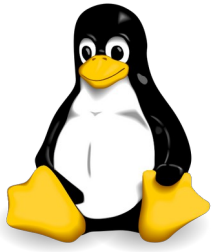




LUG Albtal

Was ist Docker?

- Erstellen, Ausführen, Verwalten von Software-Container
- Open-Source
- läuft unter Linux
- Firma (Docker Inc.)



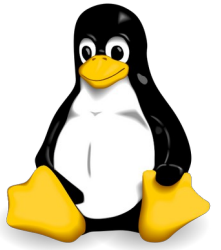
LUG Albtal

Technisch:

Gruppe von Prozessen, die isoliert voneinander ausgeführt werden

→ Software-Container

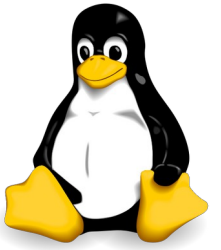
→ **Containervirtualisierung**



LUG Albtal

Basis sind bekannte Techniken:

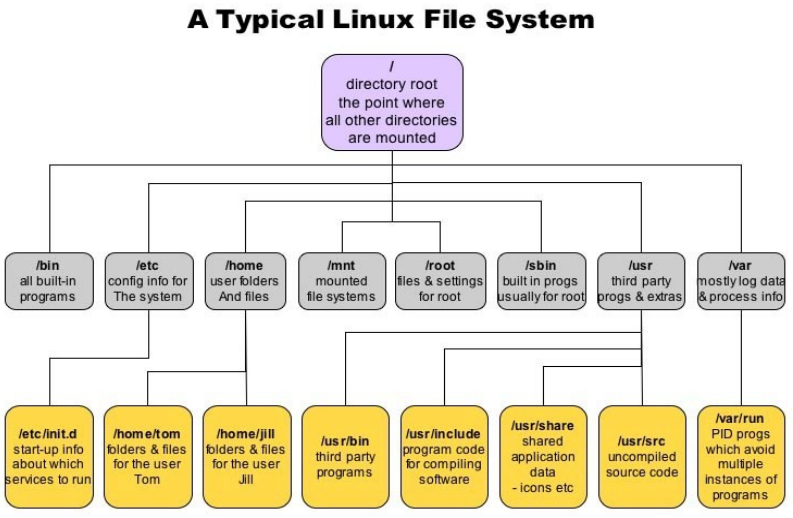
- chroot
- cgroups
- Namespaces
- UnionFS (Union file system)

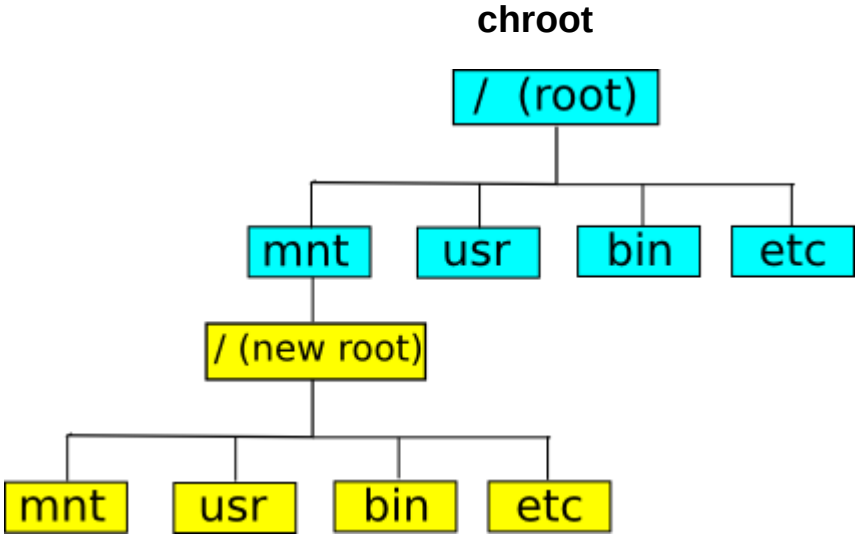


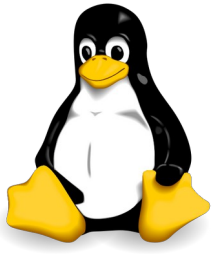
LUG Albtal

chroot

- „change root“: root-Verzeichnis im Unterordner



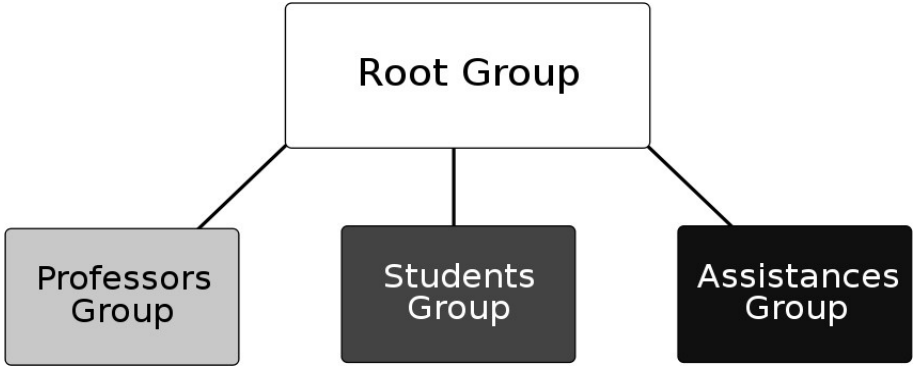


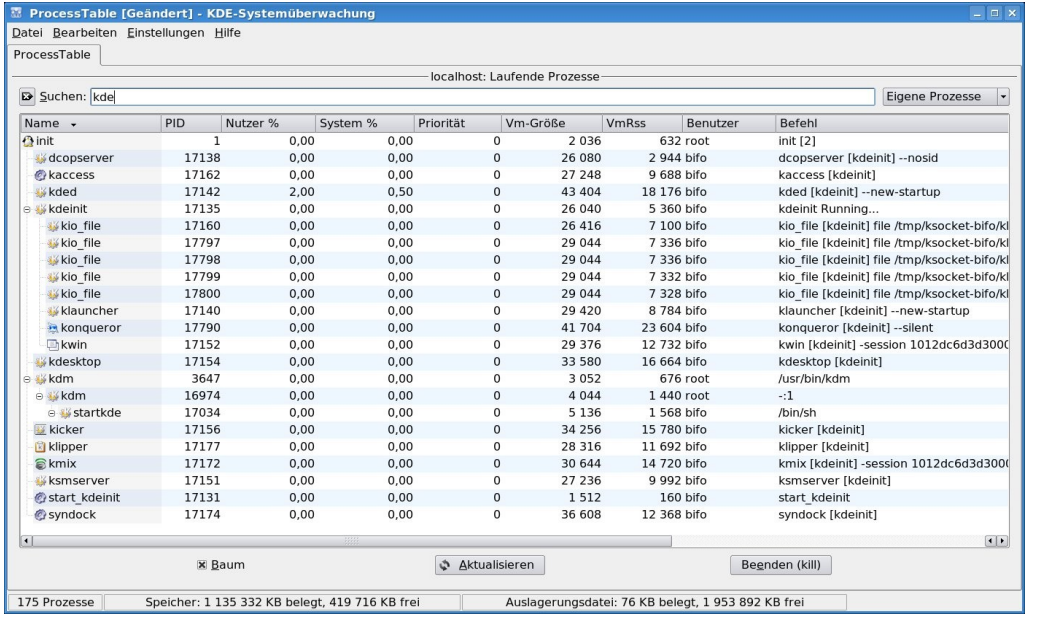


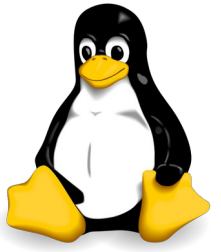
LUG Albtal

cgroups

- Prozessgruppen
- OS verwaltet Ressourcen



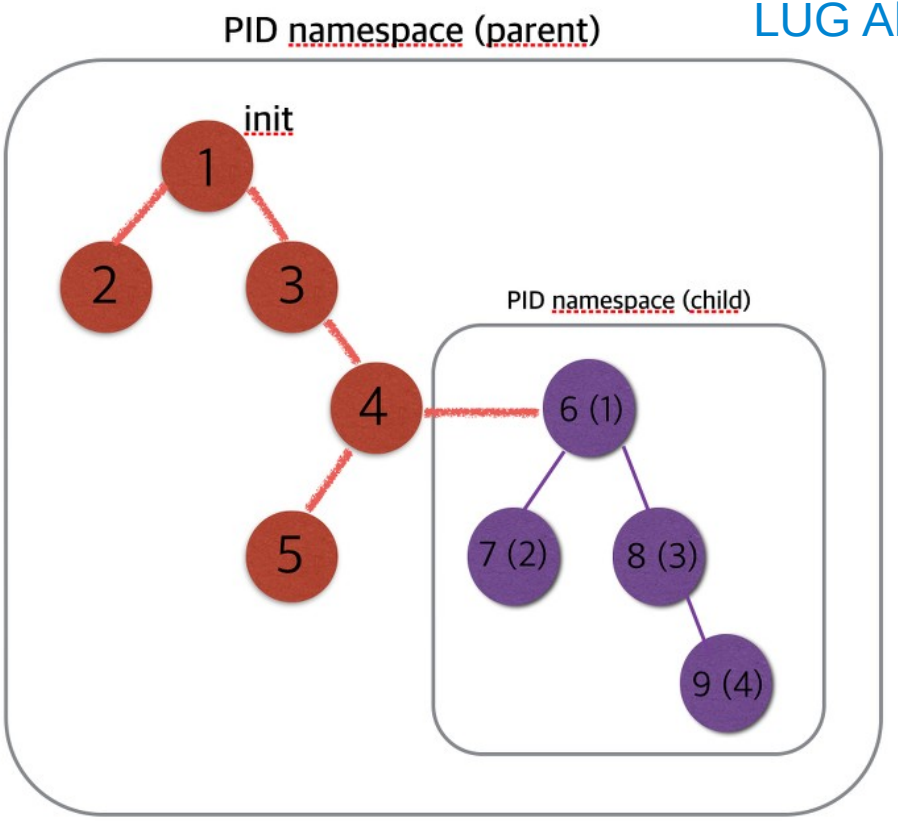


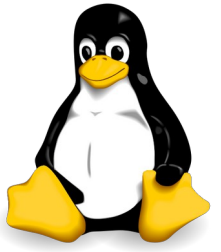


LUG Albtal

Namespaces

- Isoliert Prozessgruppen voneinander

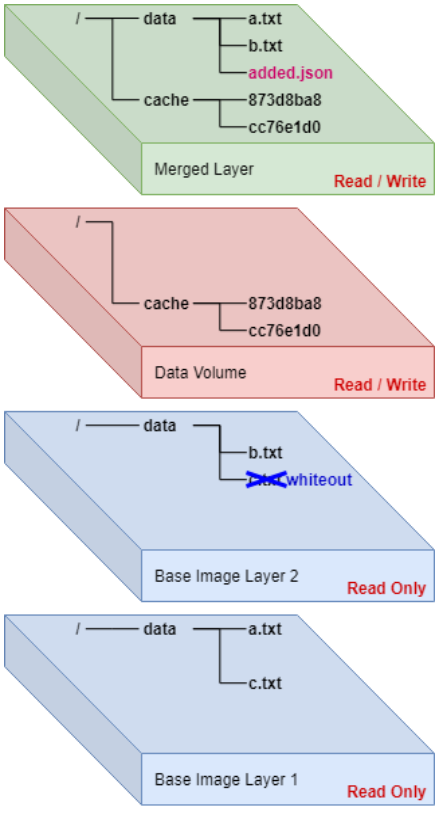


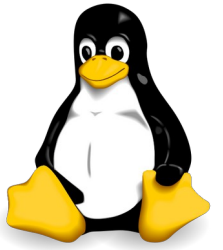


LUG Albtal

UnionFS

- Mehrere Dateisysteme werden vereinigt
- Dateien und Ordner werden überlagert

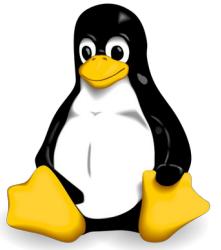




LUG Albtal

Containervirtualisierung ↔ Virtualisierung

- Containervirtualisierung nutzt Kernel und Teile des Gastsystems
- Hypervisoren (KVM, VMware, VirtualBox) sind vollwertige Betriebssysteminstanzen
- Beide Virtualisierungstechniken isolieren Systeme voneinander

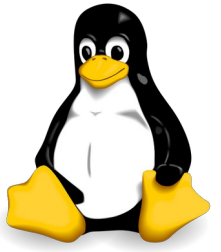


LUG Albtal

Containervirtualisierung ↔ Virtualisierung

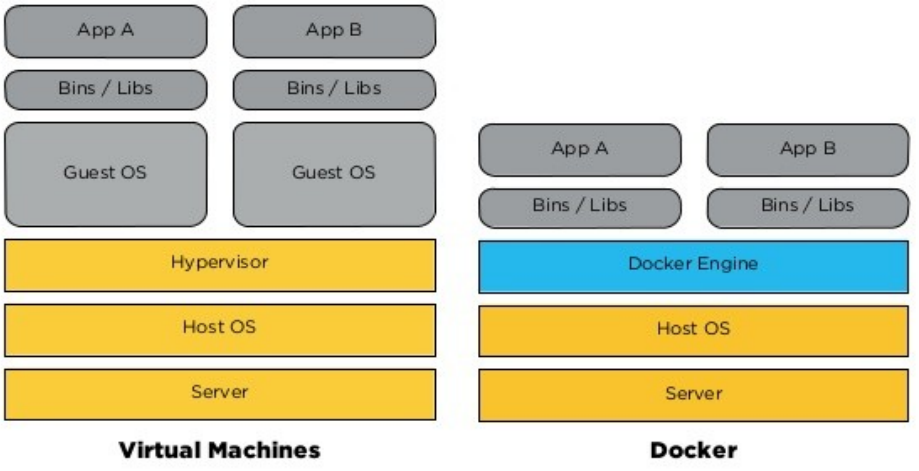
- Virtualisierung ist die Bereitstellung eines gesamten Systems (Betriebssystem + Programme)
- Container sind einzelne, isolierte Programme

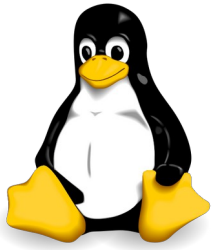
→ *Microservices*



LUG Albtal

Containervirtualisierung ↔ Virtualisierung

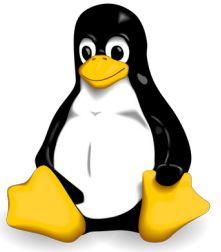




LUG Albtal

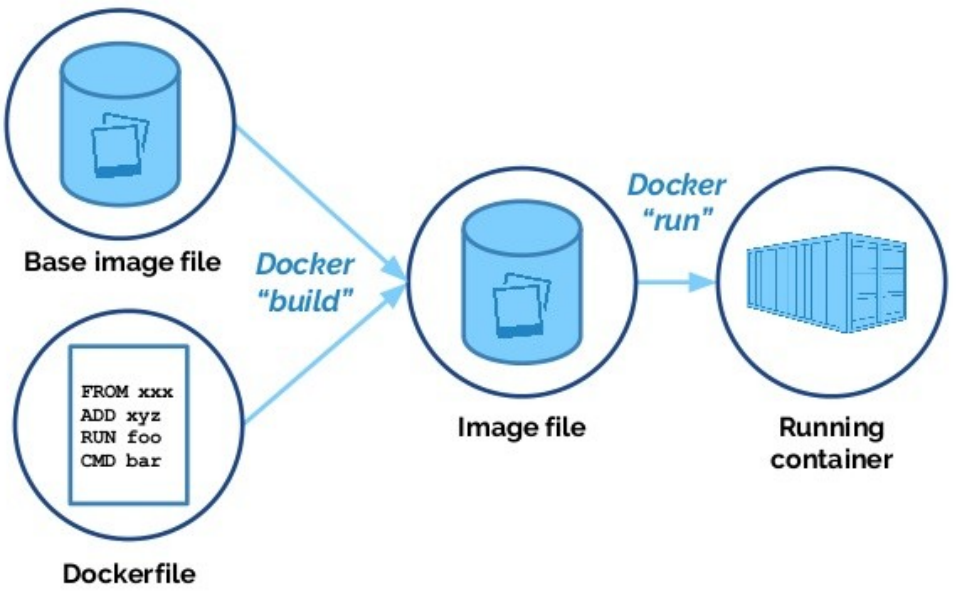
Begriffe:

- **Base Image:** Vorgefertigtes Image. Enthält alles Notwendige aber nicht mehr.
- **Image:** Vorlage für Container. Enthält Bibliotheken und Schnittstellen zum Betriebssystem. Enthält Programme
- **Dockerfile:** Datei, die ein Image erzeugt
- **Container:** Image, das ausgeführt wird



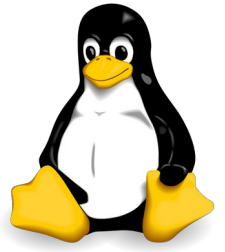
LUG Albtal

Docker images and containers



O'Melveny & Myers LLP

nexB



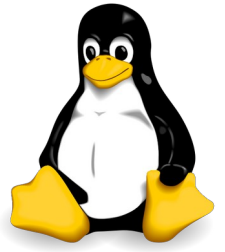
LUG Albtal

Installation von Docker unter Ubuntu:

```
sudo apt-get install docker.io
```

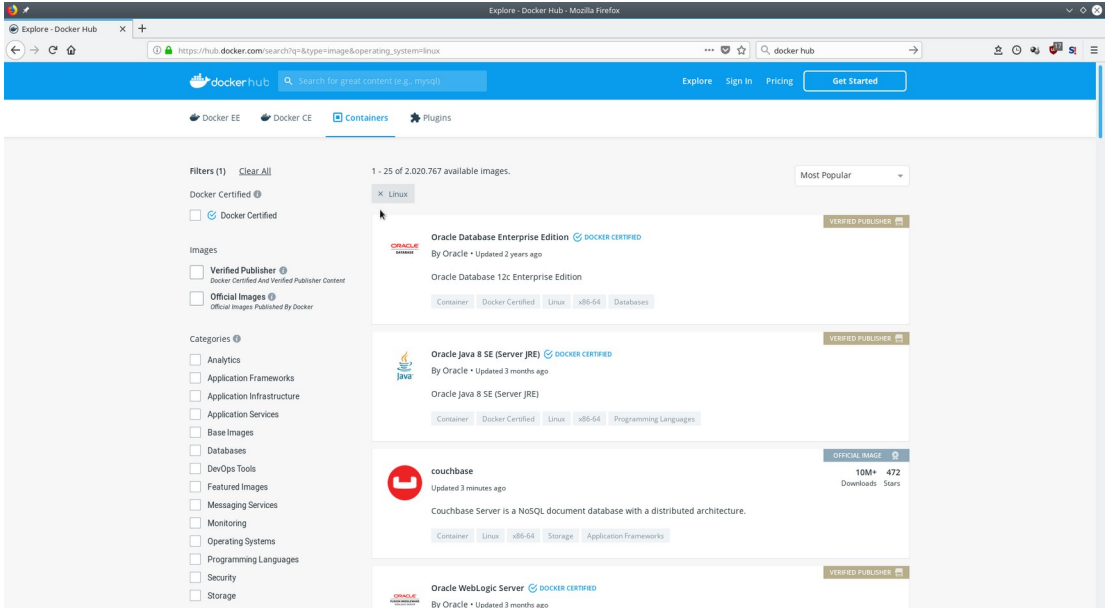
Installation testen:

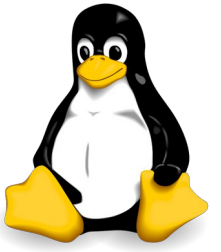
```
sudo docker run hello-world
```



LUG Albtal

Fertige Images von Docker hub laden

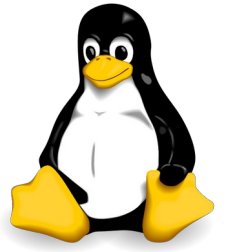




LUG Albtal

Beispiel:

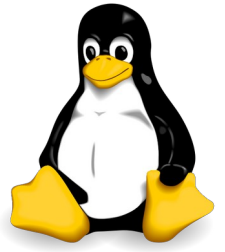
```
volker@volker-VirtualBox:~$ docker run debian echo 'TEST'
Unable to find image 'debian:latest' locally
latest: Pulling from library/debian
4ae16bd47783: Pull complete
Digest: sha256:2f04d3d33b6027bb74ecc81397abe780649ec89f1a2af18d7022737d0482cefe
Status: Downloaded newer image for debian:latest
TEST
volker@volker-VirtualBox:~$
```



LUG Albtal

Aufgabe:
Acrobat Reader als Container

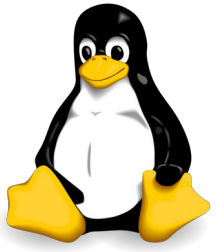
Was wird benötigt:
- Pakete i386 (da Acrobat Reader nur in 32-Bit verfügbar)
- Paket Acrobat Reader
- Paket CUPS (Linux-Druck-System)



LUG Albtal

Ablauf:

1. Basis-Image wählen
2. Ordner `/acroread` erstellen
3. Datei `/acroread/Dockerfile` erstellen
4. Image erstellen



LUG Albtal

Aufbau Dockerfile:

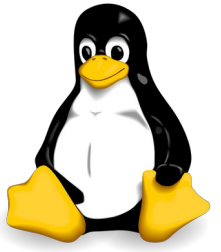
```
<Docker-Kommando> <Linux-Kommando>
```

Beispiel:

```
RUN apt-get update
```

`RUN:` Docker-Kommando

`apt-get update:` Linux-Kommando



LUG Albtal

Datei Dockerfile:

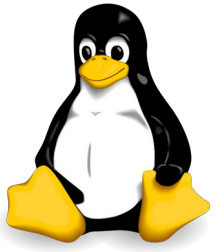
```
FROM ubuntu:18.04
```

→ Basis-Image ist Ubuntu 18.04

```
RUN apt-get update
RUN dpkg --add-architecture i386
RUN apt-get install -y cups cups-client
cups-bsd foomatic-db printer-driver-all
openprinting-ppds hpijs-ppds hp-ppd
```

→ Architektur 32-Bit festlegen

→ CUPS installieren



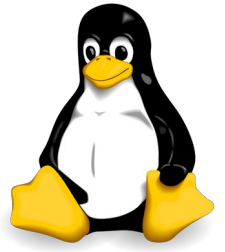
LUG Albtal

Datei Dockerfile:

```
RUN apt-get install -y gdebi-core
libxml2:i386 gtk2-engines-murrine:i386
libcanberra-gtk-module:i386 libatk-adaptor:i386
libgail-common:i386
RUN apt-get install -y wget
RUN wget
ftp://ftp.adobe.com/pub/adobe/reader/unix/9.x/9.5.5/
enu/AdbeRdr9.5.5-1_i386linux_enu.deb
RUN dpkg -i AdbeRdr9.5.5-1_i386linux_enu.deb
```

→ Pakete für 32-Bit

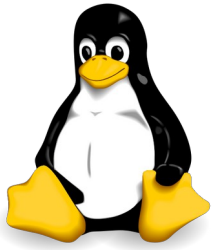
→ Acrobat Reader installieren



LUG Albtal

Datei Dockerfile:

```
CMD service cups start & acroread
```
→CUPS und Acrobat Reader starten



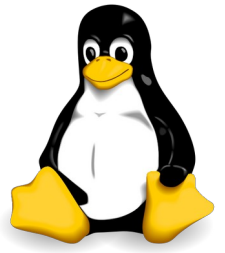
LUG Albtal

Image erstellen

```
cd /acroread
docker build -t acroread .
```

Ablauf:

1. Basis-Paket wird von Docker hub geladen
2. Pakete werden installiert
3. Fertiges Image wird gespeichert



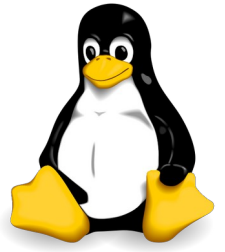
LUG Albtal

Image ausführen

```
docker run --net=host –env="DISPLAY"
--volume="$HOME/.Xauthority:/root/.Xauthority:rw"
--volume="/work/daten:/mnt"
--volume="/work/ppd:/etc/cups/ppd"
--volume="/work/printers.conf:/etc/cups/printers.conf"
acroread
```

Auf Host muß vorhanden sein:

- Ordner /work/daten → PDF-Dateien
- Ordner /work/ppd, Datei /work/printers.conf → Drucker für CUPS

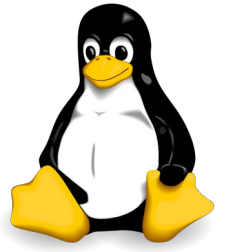


LUG Albtal

Image ausführen

1. Container starten: `docker run`
2. Netzwerk mit Gastsystem aufbauen: `--net=host`
3. Lokaler Bildschirm übergeben: `--env="Display"`
4. Laufwerke/Dateien mit Host verbinden: `--volume`
5. CUPS und Acrobat Reader starten:

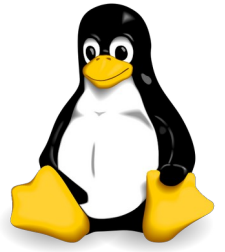
siehe Dockerfile: `CMD service cups start & acroread`



LUG Albtal

Was passiert, wenn das Image gestartet wird:

- Container startet jedes Mal neu → lokale Daten gehen verloren
- Container kann nur über Netzwerk angesprochen werden → verhält sich wie ein eigener Rechner



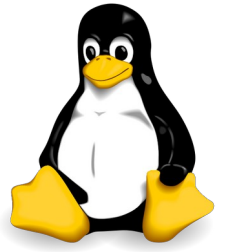
LUG Albtal

Kommandos:

- Info zu aktiven Container: `docker ps`
- Info zu Docker: `docker info`
- Shell im Container starten:

`docker exec -ti <Container> /bin/bash`

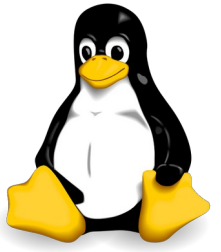
- Image löschen: `docker rm -t <Image>`



LUG Albtal

Weitere Informationen und Quellen:

- https://de.wikipedia.org/wiki/Docker_(Software)
- https://www.informatik-aktuell.de/entwicklung/methoden/containerplattform-lego-fuer-devops.html
- https://entwickler.de/online/development/docker-basics-system-level-virtualisierung-125514.html
- https://www.linuxwiki.de/chroot
- https://www.pro-linux.de/artikel/2/1464/ressourcen-verwaltung-mit-control-groups-cgroups.html
- https://en.wikipedia.org/wiki/Linux_namespaces
- https://de.wikipedia.org/wiki/UnionFS



LUG Albtal

Persönliches Fazit:

- Containervirtualisierung ersetzt nicht Virtualisierung mit Hypervisor.
- ... ist performant.
- ... ist ideal für Server-Dienste.
- ... ist ideal für Softwareentwicklung.
- ... setzt Linux-Systemkenntnisse voraus.

LUG Albtal