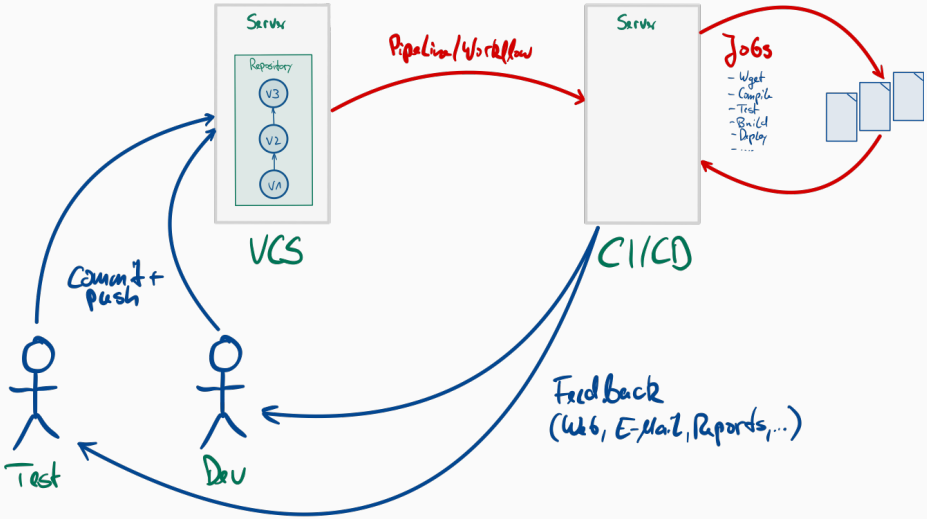
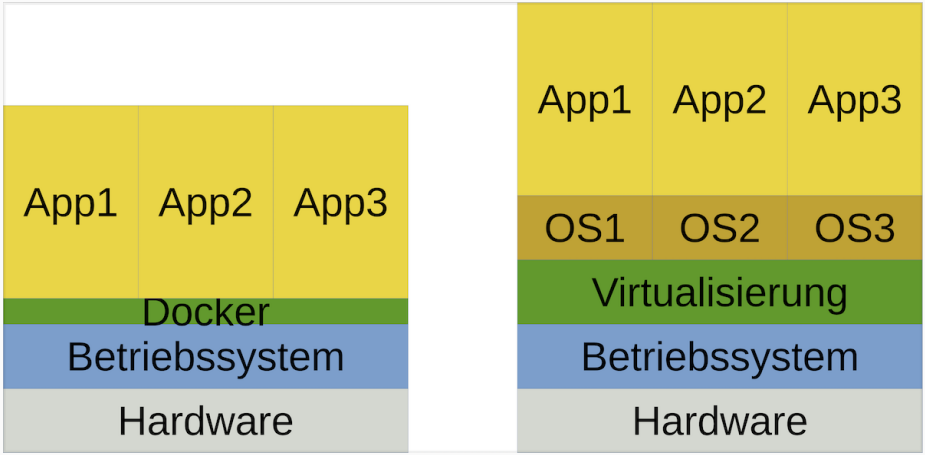
Einführung in Docker

Carsten Gips (HSBI)

Motivation CI/CD: WFM (*Works For Me*)



Virtualisierung: Container vs. VM



Getting started

- DockerHub: fertige Images => hub.docker.com/search

- Image downloaden: `docker pull <IMAGE>`
- Image starten: `docker run <IMAGE>`

Beispiele

```
docker pull debian:stable-slim
docker run  --rm -it  debian:stable-slim  /bin/sh
```

```
docker pull openjdk:latest
docker run  --rm  -v "$PWD":/data -w /data  openjdk:latest  javac Hello.java
docker run  --rm  -v "$PWD":/data -w /data  openjdk:latest  java Hello
```

Demo: Container in der Konsole

Images selbst definieren

```
FROM debian:stable-slim

ARG USERNAME=pandoc
ARG USER_UID=1000
ARG USER_GID=1000

RUN apt-get update && apt-get install -y --no-install-recommends            \
        apt-utils bash wget make graphviz biber                             \
        texlive-base texlive-plain-generic texlive-latex-base               \
    #
    && groupadd --gid $USER_GID $USERNAME                                   \
    && useradd -s /bin/bash --uid $USER_UID --gid $USER_GID -m $USERNAME    \
    #
    && apt-get autoremove -y && apt-get clean -y && rm -rf /var/lib/apt/lists/*

WORKDIR /pandoc
USER $USERNAME
```

`docker build -t <NAME> -f <DOCKERFILE> .`

Beispiel: debian-latex.df

CI-Pipeline (GitLab)

```yaml
default:
    image: openjdk:17

job1:
    stage: build
    script:
        - java -version
        - javac Hello.java
        - java Hello
        - ls -lags
```

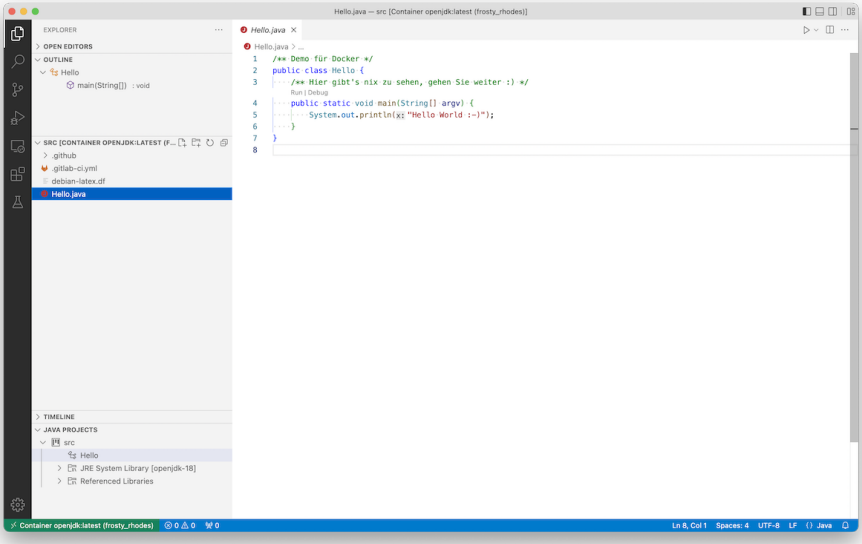
Demo: GitLab CI/CD und Docker

CI-Pipeline (GitHub)

```yaml
name: demo
on:
    push:
        branches: [master]
    workflow_dispatch:

jobs:
    job1:
        runs-on: ubuntu-latest
        container: docker://openjdk:17
        steps:
            - uses: actions/checkout@v4
            - run: java -version
            - run: javac Hello.java
            - run: java Hello
            - run: ls -lags
```

VSCode und das Plugin “Remote - Containers”



Wrap-Up

- Schlanke Virtualisierung mit Containern (kein eigenes OS)
- *Kein* Sandbox-Effekt

- Begriffe: Docker-File vs. Image vs. Container
- Ziehen von vordefinierten Images
- Definition eines eigenen Images
- Arbeiten mit Containern: lokal, CI/CD, VSCode ...

LICENSE

Exceptions

- Website hub.docker.com